

Package: IsoFishR (via r-universe)

October 21, 2026

Title Data Reduction and Analysis of Laser-Ablation Strontium Isotope Profiles

Version 1.99.0.9000

Description Reduces $^{87}\text{Sr}/^{86}\text{Sr}$ isotope ratios measured by laser-ablation multi-collector ICP-MS (blank correction, Rb interference and mass bias correction, outlier rejection, smoothing) and supports profile analysis (trimming, regions, changepoints). Version 2 development; the original application is preserved under the v1.1 tag.

License MIT + file LICENSE

URL <https://github.com/MalteWillmes/IsoFishR>

BugReports <https://github.com/MalteWillmes/IsoFishR/issues>

Depends R (≥ 4.1)

Imports bslib ($\geq 0.6.0$), changepoint, dplyr ($\geq 1.1.0$), DT, later, mgcv, plotly, purrr, readr, rlang ($\geq 1.1.0$), shiny ($\geq 1.8.1$), stats, tibble, tidyr, utils, yaml

Suggests htmlwidgets, rstudioapi, tcltk, testthat ($\geq 3.0.0$), withr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://maltewillmes.r-universe.dev>

Date/Publication 2026-10-21 07:42:43 UTC

RemoteUrl <https://github.com/MalteWillmes/IsoFishR>

RemoteRef v2-dev

RemoteSha 4a843fe00edb8876c1dab9d84907b6995ec40344

Contents

add_runfiles	3
apply_edits	4
calibration_model	4
check_settings	5
correct_ratios	6
create_example_project	6
default_reference_materials	7
default_settings	8
export_project	8
export_report	9
find_changepoints	10
find_runfiles	11
import_presets	11
import_v1_settings	12
integrate_cycles	13
merge_settings	13
profile_plot	14
project	15
project_baselines	16
project_data	16
project_profiles	17
project_regions	18
project_runs	18
project_summaries	19
read_acquisition_times	20
read_log	20
read_runfiles	21
reduce_runs	21
reference_material_plot	22
reference_material_results	23
reject_outliers	23
remove_samples	24
report_markdown	24
run_app	25
set_reference_material	25
set_regions	26
set_run_type	27
set_sample_settings	27
set_session	28
settings_io	28
smooth_ma	29
smooth_spline	29
suggest_reference_runs	30
summarise_blank	31
summarise_regions	31
update_project_settings	32

<i>add_runfiles</i>	3
---------------------	---

update_sample	32
-------------------------	----

Index	34
--------------	-----------

add_runfiles	<i>Add runfiles to a project</i>
--------------	----------------------------------

Description

Copies runfiles into the project’s raw/ folder and registers them as new runs, so a project can be built up batch by batch. Each file is read first with the project’s import settings. A file that can’t be read is not added and is reported, without stopping the rest of the batch. If a file’s header doesn’t match the import settings (e.g. lines missing after a bad save on the instrument), the start of its data is found automatically and the file is added with a note.

Usage

```
add_runfiles(  
  project,  
  paths,  
  sample_ids = NULL,  
  source_names = basename(paths),  
  mode = c("add", "replace")  
)
```

Arguments

project	A project from open_project() .
paths	Paths to runfiles.
sample_ids	Sample IDs; default to the file names without extension.
source_names	Original file names to record (useful when paths are temporary upload paths).
mode	"add" or "replace", see above.

Details

Files identical to one already in the project are always skipped. For a file whose name matches a run already in the project but whose content differs (e.g. a re-exported runfile):

- mode = "add" skips it and reports it;
- mode = "replace" gives that run the new file and starts it afresh: its edits (trims, status, flag, comment, changepoint and per-sample settings) and regions are removed. Its run type, reference material and session are kept. The previous file is moved to raw/removed/ and the replacement is logged.

Value

The updated project. A message summarises what happened; the attribute "import_report" holds one row per file with file, sample_id, result ("added", "replaced", "skipped" or "failed") and detail.

apply_edits	<i>Apply trims and profile direction</i>
-------------	--

Description

Apply trims and profile direction

Usage

```
apply_edits(data, edits = NULL)
```

Arguments

data	Reduced data (<code>reduce_runs()</code> \$data).
edits	A data frame with one row per edited sample and columns <code>sample_id</code> , <code>trim_start_s</code> , <code>trim_end_s</code> (seconds removed from the start and end of the profile), <code>reverse</code> (flip the distance axis) and <code>rezero</code> (measure distance from the first kept point). Missing columns and samples without a row use no trim, no reversal and rezero = TRUE.

Value

The edited data, sorted by sample and distance.

calibration_model	<i>Reference material correction model</i>
-------------------	--

Description

Measures every run of the primary reference material (mean $^{87}\text{Sr}/^{86}\text{Sr}$ of its points after trims and outlier rejection, uncorrected) and fits the correction factor accepted / measured through time within each session, using `calibration$fit`:

Usage

```
calibration_model(project)
```

Arguments

project	A project from <code>open_project()</code> .
---------	--

Details

- "regression": a least-squares line through the primary runs' measured means against time (instrument drift); the factor is accepted / predicted. Its uncertainty is the standard error of the regression line at each time (with 2 runs, the runs' own SE).
- "linear": linear interpolation between the primary runs bracketing each point (standard-sample bracketing); constant before the first and after the last primary run of the session.
- "spline": a smoothing spline through the session's primary runs (needs at least 4; with fewer, linear interpolation is used).
- "step": the factor of the most recent primary run ("step forward").
- "mean" (default): one factor per session, the mean of its primary runs.

Value

NULL when no primary reference material is set; otherwise a list with primary, accepted, uncertainty_2s, fit and points (one row per primary run: sample_id, session, time (run midpoint), measured, se, n, factor, u_rel).

check_settings	<i>Validate settings</i>
----------------	--------------------------

Description

Validate settings

Usage

check_settings(settings)

Arguments

settings A settings list.

Value

settings, invisibly; errors describe the first problem found.

correct_ratios	<i>Rb interference and mass bias correction</i>
----------------	---

Description

Mass bias is calculated per cycle from the measured $^{86}\text{Sr}/^{88}\text{Sr}$ ratio using the exponential law, and applied to the Rb-corrected $^{87}\text{Sr}/^{86}\text{Sr}$ ratio. ^{87}Rb is estimated from ^{85}Rb with the natural $^{85}\text{Rb}/^{87}\text{Rb}$ ratio, corrected for the same mass bias.

Usage

```
correct_ratios(data, sr86_sr88_ref = 0.1194, rb85_rb87 = 2.59712)
```

Arguments

data	Blank-corrected cycle data.
sr86_sr88_ref	Reference $^{86}\text{Sr}/^{88}\text{Sr}$ ratio.
rb85_rb87	Natural $^{85}\text{Rb}/^{87}\text{Rb}$ ratio.

Value

data with added ratio columns. sr87_sr86 is the fully corrected ratio; sr84_sr86_mb is $^{84}\text{Sr}/^{86}\text{Sr}$ corrected for mass bias (v1 only reported the uncorrected sr84_sr86).

create_example_project	<i>Create an example project</i>
------------------------	----------------------------------

Description

Creates a project from the example runfiles that come with IsoFishR: five Chinook salmon otolith transects from Putah Creek, California, and one white seabass otolith, measured on a Nu Plasma MC-ICP-MS at UC Davis. It is a quick way to try the app or the package functions without data of your own.

Usage

```
create_example_project(
  path = file.path(tempfile("IsoFishR_"), "Example_project"),
  user = default_user()
)
```

Arguments

path	Folder for the new project. By default a new folder in the R session's temporary directory, which is deleted when R closes.
user	Name recorded in the project log.

Details

The project uses the default settings and the "Nu_UCDavis" import preset, with a run speed of 5 um/s as in the example project of IsoFishR v1.

Value

The open project (see [open_project\(\)](#)).

Examples

```
p <- create_example_project()
p
head(summarise_project_samples(p))
close_project(p)
```

default_reference_materials
<i>Default reference materials</i>

Description

Reference materials IsoFishR recognises in run names, with default accepted $^{87}\text{Sr}/^{86}\text{Sr}$ values set by the IsoFishR authors. They are only used to suggest marking runs (see [suggest_reference_runs\(\)](#)); the value can be changed when the suggestion is applied, and afterwards under Calibration.

Usage

```
default_reference_materials()
```

Value

A tibble with name (also the text looked for in run names) and value.

default_settings	<i>Default project settings</i>
------------------	---------------------------------

Description

Settings are a nested list with four sections: sample (metadata that does not affect calculations), import (runfile layout), reduction (data reduction parameters) and regions (default region labels). Projects store them as project.yml; individual samples only store values that differ from these defaults.

Usage

```
default_settings()
```

Value

A nested list of settings.

export_project	<i>Export project summaries</i>
----------------	---------------------------------

Description

Writes the summary tables to a new time-stamped folder in the project's exports/ folder, together with export_report.md (a record of the settings, calibration and reference materials). Earlier exports are kept.

Usage

```
export_project(  
  project,  
  files = setdiff(EXPORT_FILES, "data"),  
  include_excluded = FALSE  
)  
  
EXPORT_FILES
```

Arguments

- project A project from `open_project()`.
- files Which files to write, any of EXPORT_FILES: "samples", "regions", "changepoints", "profiles", "runs", "reference_materials", "settings", "data". Defaults to all but "data".
- include_excluded Include samples with status "excluded".

Details

Files written:

- samples.csv: `summarise_project_samples()`
- regions.csv: `summarise_project_regions()`
- changepoints.csv: `summarise_project_changepoints()`
- settings.csv: `project_settings_table()`
- profiles.csv: `project_profiles()`, every point of every profile
- runs.csv: `project_runs()`, all runs in acquisition order with type and session
- reference_materials.csv and reference_material_summary.csv (when the project has reference material runs): `reference_material_results()` and `summarise_reference_materials()`
- export_report.md (always): `export_report()` as Markdown, a record of the export: all project settings, the calibration method, the primary and secondary reference materials with their number of runs, mean and SD, the sessions, and per-sample setting overrides.
- data.csv (not by default): all reduction columns per point (raw voltages, mass bias, intermediate ratios), for checking the reduction itself.

Value

The path of the export folder, invisibly.

export_report

Record of an export

Description

Everything needed to understand and reproduce the exported values, as written to `export_report.md` by `export_project()`: the project, IsoFishR version and user; the runs and sessions; the calibration (method, baseline mode, session gap, and for the primary and each secondary reference material its accepted value, number of runs and the mean and SD of its run means before and after correction, overall and per session); all project settings; and the samples with their own setting overrides.

Usage

```
export_report(
  project,
  files = character(),
  include_excluded = FALSE,
  n_samples = NA_integer_
)
```

Arguments

project	A project from <code>open_project()</code> .
files	Names of the files written with it.
include_excluded	Whether excluded samples were exported.
n_samples	Number of samples exported.

Value

A nested list.

find_changepoints	<i>Detect changepoints in a profile</i>
-------------------	---

Description

Runs `changepoint::cpt.mean()` on each sample and adds the segment number and segment mean. Segment means are calculated within each sample (v1 pooled samples that shared a segment number).

Usage

```
find_changepoints(
  data,
  method = "PELT",
  penalty = 5e-05,
  value = "sr87_sr86_value"
)
```

Arguments

data	Reduced (and usually edited) data, sorted by distance.
method	Changepoint method: "PELT", "AMOC", "SegNeigh" or "BinSeg".
penalty	Manual penalty value.
value	Column to analyse.

Value

data with `cp_segment` and `cp_mean` added (NA where the value is missing).

find_runfiles	<i>Find runfiles in a folder</i>
---------------	----------------------------------

Description

Lists files with the runfile extension of the import settings in a folder and all its subfolders.

Usage

```
find_runfiles(dir, import = default_settings()$import)
```

Arguments

dir	Folder to search.
import	The import section of a settings list (for extension).

Value

A tibble with path, file, sample_id, folder (relative to dir) and duplicate (TRUE when another file in the folder tree has the same name; such files can't be told apart and are not added).

import_presets	<i>Instrument import presets</i>
----------------	----------------------------------

Description

A preset is a named runfile layout: header and footer lines to skip, the separator, and which column holds each mass and the time. IsoFishR ships with "Nu_UCDavis"; users can save their own presets, which are stored as YAML files in their IsoFishR config folder (import_preset_dir()) and can be shared by copying the file.

Usage

```
list_import_presets()

get_import_preset(name)

save_import_preset(name, import, description = "", overwrite = FALSE)

delete_import_preset(name)

import_preset_dir()
```

Arguments

name	Preset name (letters, numbers, _, - and .).
import	The import section of a settings list.
description	Optional one-line description.
overwrite	Replace an existing user preset of the same name.

Details

A project records the name of the preset it was set up with, and a full copy of its values, so a project keeps working even where the preset is not installed.

Value

list_import_presets(): a tibble with name, description and builtin.
 get_import_preset(): the import settings of the preset, with preset set to its name.
 save_import_preset(): the path of the preset file, invisibly.
 import_preset_dir(): the folder holding user presets. Set the option isofishr.preset_dir to use a different (e.g. shared) folder.

import_v1_settings	<i>Convert a v1 project settings file</i>
--------------------	---

Description

Reads the <project>_settings.csv written by IsoFishR v1 and returns the equivalent v2 settings list.

Usage

```
import_v1_settings(path)
```

Arguments

path	Path to a v1 _settings.csv file.
------	----------------------------------

Value

A settings list.

integrate_cycles	<i>Average consecutive cycles</i>
------------------	-----------------------------------

Description

Averages blocks of n consecutive cycles within each sample. A shorter final block is kept. Blocks never span two samples (in v1 they could).

Usage

```
integrate_cycles(raw, n = 1L)
```

Arguments

raw	Raw cycle data from read_runfiles() .
n	Number of cycles per block.

Value

A tibble with one row per block and columns sample_id, time_s, raw88 to raw83 and n_cycles.

merge_settings	<i>Combine settings with overrides</i>
----------------	--

Description

Combine settings with overrides

Usage

```
merge_settings(settings, overrides = list())
```

Arguments

settings	A settings list, usually from default_settings() .
overrides	A (partial) nested list of values that replace those in settings.

Value

The combined settings, validated with [check_settings\(\)](#).

profile_plot

Interactive profile plot

Description

The $^{87}\text{Sr}/^{86}\text{Sr}$ profile of one sample against distance, with the smoothed line and its uncertainty band, rejected outliers, regions and changepoint means, plus strips for total Sr, ^{83}Kr and $^{85}\text{Rb}/^{88}\text{Sr}$ below it. Drag along the x axis to select a range.

Usage

```
profile_plot(
  data,
  regions = NULL,
  show = c("points", "outliers", "ma", "spline", "band", "strips"),
  source = "profile",
  lines = numeric(),
  smoothing = NULL
)
```

Arguments

data	Reduced and edited data for one sample (from project_data()).
regions	Regions of the sample (region, start_s, end_s).
show	Elements to show: any of "points", "outliers", "ma" (moving average), "spline", "band", "seawater", "strips". All traces except the strips are always included; those not in show start hidden ("legendonly") so they can be switched on without rebuilding the plot.
source	plotly event source ID.
lines	$^{87}\text{Sr}/^{86}\text{Sr}$ values to mark with labelled horizontal lines (e.g. values of known habitats).
smoothing	The sample's smoothing method (see default_settings()): which smoothed lines to draw. By default worked out from data.

Details

Point traces are drawn with WebGL and values are rounded to the precision shown, which keeps redraws fast for long profiles.

Value

A built plotly object (see [plotly::plotly_build\(\)](#)).

project	<i>IsoFishR projects</i>
---------	--------------------------

Description

A project is a folder chosen by the user. It holds the original runfiles and every choice made about them; reduced data and exports are derived from these and can always be regenerated.

Usage

```
create_project(
  path,
  settings = default_settings(),
  user = default_user(),
  import_preset = NULL
)

open_project(path, user = default_user(), read_only = FALSE, force = FALSE)

close_project(project)
```

Arguments

path	Project folder.
settings	Project settings; defaults to <code>default_settings()</code> .
user	Name recorded in the log and the lock file.
import_preset	Name of an import preset (see <code>list_import_presets()</code>) whose runfile layout replaces the one in settings. NULL keeps settings\$import (by default the "Nu_UCDavis" preset).
read_only	Open without taking the lock; changes are refused.
force	Take the lock even if someone else holds it (e.g. after their session crashed).
project	A project from <code>open_project()</code> .

Details

MyProject/	
project.yml	project settings (see <code>default_settings()</code>)
raw/	copies of the original runfiles, never modified
samples.csv	one row per sample: status, flag, comment, edits, setting overrides
regions.csv	regions per sample, in seconds of analysis time
log.csv	every change: time, user, action, sample, field, old and new value
cache/	reduced data per sample (safe to delete)
exports/	generated tables and plots
.lock	who has the project open for editing

`create_project()` makes a new project folder and opens it. `open_project()` opens an existing one and takes the lock, so only one person edits a project at a time (useful for shared Dropbox folders). `close_project()` releases the lock.

Functions that change a project write to disk immediately, add entries to `log.csv` and return the updated project object.

Value

An `isofishr_project` object (`close_project()` returns `NULL`).

<code>project_baselines</code>	<i>Baseline runs of a project</i>
--------------------------------	-----------------------------------

Description

Mean voltages of each baseline (gas blank) run, after its trims, with its acquisition time and session. Used when `calibration$baseline` is "baseline_runs": the blank subtracted from a run is then interpolated linearly in time between the baseline runs of its session, at the run's start time (and held constant before the first and after the last).

Usage

```
project_baselines(project)
```

Arguments

`project` A project from `open_project()`.

Value

A tibble with `sample_id`, `session`, `acquired`, `n_cycles` and `blk88` to `blk83`.

<code>project_data</code>	<i>Reduced data for project samples</i>
---------------------------	---

Description

Reduces each sample with its own settings and applies its trims and direction. Results are cached in the project's cache/ folder, keyed by the raw file, the sample's import and reduction settings, the run speed and the IsoFishR version. A sample is only reduced again when one of those changes, so adding samples or changing one sample's settings does not reprocess the others.

Usage

```
project_data(project, sample_ids = NULL, edited = TRUE, calibrate = TRUE)
```

Arguments

project	A project from <code>open_project()</code> .
sample_ids	Samples to return; defaults to all runs that are not excluded or baseline runs.
edited	Apply trims, reversal and re-zeroing from <code>samples.csv</code> .
calibrate	Apply the reference material correction (see <code>calibration_model()</code>). The 87Sr/86Sr columns are then corrected, and <code>sr87_sr86_uncorrected</code> , <code>correction_factor</code> , <code>correction_u_rel</code> and <code>correction_status</code> are added. Without a primary reference material the values are unchanged and the factor is NA.

Value

A tibble of reduced data. Any warnings raised during reduction (e.g. no blank cycles) are attached as the "warnings" attribute, a character vector named by sample.

project_profiles	<i>Full profiles of all samples</i>
------------------	-------------------------------------

Description

One row per measurement point for every sample, after trims and reversal: the profile as plotted in the app, in a single table.

Usage

```
project_profiles(project, data = NULL, include_excluded = FALSE)
```

Arguments

project	A project from <code>open_project()</code> .
data	Reduced and edited data from <code>project_data()</code> ; computed if not supplied.
include_excluded	Include samples with status "excluded".

Details

Columns: `sample_id`, `distance_um`, `time_s`, `sr87_sr86` (the point, NA if rejected), `sr87_sr86_outlier` (rejected value), `sr87_sr86_ma` and `sr87_sr86_ma_se` (moving average and its SE) and `sr87_sr86_spline` and `sr87_sr86_spline_se` (spline and its SE), each filled when the sample's smoothing method (in smoothing) uses it, `region` (region name(s) the point falls in, separated by ;), `cp_segment` and `cp_mean` (for samples with changepoints enabled), and the signal columns `total_sr_v`, `kr83_v`, `rb85_sr88` and `sr84_sr86`.

Value

A tibble.

project_regions	<i>Region statistics for project samples</i>
-----------------	--

Description

Region statistics for project samples

Usage

```
project_regions(project, sample_ids = NULL)
```

Arguments

project	A project from open_project() .
sample_ids	Samples to return; defaults to all runs that are not excluded or baseline runs.

Value

One row per sample and region; see [summarise_regions\(\)](#).

project_runs	<i>Runs in acquisition order</i>
--------------	----------------------------------

Description

Lists the runs of a project in the order they were measured, with their type and analytical session. Sessions are found automatically: a new session starts when the gap between two runs is longer than `calibration$session_gap_h` hours (default 6). A session set with [set_session\(\)](#) replaces the automatic one for that run.

Usage

```
project_runs(project)
```

Arguments

project	A project from open_project() .
---------	---

Value

A tibble with `sequence`, `sample_id`, `acquired` (POSIXct, UTC), `run_type`, `reference_material`, `rm_role` ("primary", "secondary" or ""), `status`, `session_auto` and `session`.

project_summaries	<i>Project summaries</i>
-------------------	--------------------------

Description

Summary tables for the samples in a project, as written by `export_project()`.

Usage

```
summarise_project_samples(project, data = NULL, include_excluded = FALSE)

summarise_project_regions(project, data = NULL, include_excluded = FALSE)

summarise_project_changepoints(project, data = NULL, include_excluded = FALSE)

project_settings_table(project, include_excluded = FALSE)
```

Arguments

project	A project from <code>open_project()</code> .
data	Reduced and edited data from <code>project_data()</code> ; computed if not supplied.
include_excluded	Include samples with status "excluded".

Details

- `summarise_project_samples()`: one row per sample with its status, comment, edits, profile statistics and setting overrides.
- `summarise_project_regions()`: one row per sample and region.
- `summarise_project_changepoints()`: one row per changepoint segment, for samples with changepoint detection enabled.
- `project_settings_table()`: the settings each sample was reduced with.

Means, standard deviations and standard errors are calculated from the individual ⁸⁷Sr/⁸⁶Sr points (outliers removed). Smoothed values are autocorrelated, so their standard error would understate the uncertainty; the mean of the smoothed values is reported alongside as `mean_smoothed`.

When a primary reference material is set (see `calibration_model()`), all ⁸⁷Sr/⁸⁶Sr values are corrected, and the tables add `mean_uncorrected`, `correction_factor`, `correction_status` and `se_combined`: the SE combined with the uncertainty of the correction.

Value

A tibble.

read_acquisition_times

Read the acquisition time of runfiles

Description

Reads the date and time a run was started from its header, using the `timestamp_line` and `timestamp_format` of the import settings (for Nu Plasma files: line 45, e.g. "08/12/2017", "04:31:02 PM"). Quotes are removed and commas or tabs between the date and time become spaces before the line is parsed with `base::strptime()` format. Times are treated as the instrument clock and returned in UTC, so daylight saving changes never reorder runs.

Usage

```
read_acquisition_times(paths, import = default_settings()$import)
```

Arguments

<code>paths</code>	Paths to runfiles.
<code>import</code>	The import section of a settings list.

Value

A POSIXct vector (NA where there is no timestamp line or it could not be parsed).

read_log

Read the project session log

Description

Read the project session log

Usage

```
read_log(project)
```

Arguments

<code>project</code>	A project from <code>open_project()</code> .
----------------------	--

Value

A tibble with one row per recorded change.

read_runfiles	<i>Read laser-ablation runfiles</i>
---------------	-------------------------------------

Description

Reads the cycle data from one or more instrument runfiles, using the column layout in the import section of the settings.

Usage

```
read_runfiles(  
  paths,  
  import = default_settings()$import,  
  sample_ids = tools::file_path_sans_ext(basename(paths))  
)
```

Arguments

paths	Paths to runfiles.
import	The import section of a settings list.
sample_ids	Sample IDs, one per file. Defaults to the file names without extension. Pass these explicitly when paths are temporary upload paths.

Value

A tibble with one row per measurement cycle and columns sample_id, cycle, time_s and raw88 to raw83 (volts).

reduce_runs	<i>Reduce laser-ablation Sr isotope data</i>
-------------	--

Description

Runs the full data reduction for one or more samples: integration, blank correction, voltage thresholds, distance, Rb interference and mass bias correction, outlier rejection and smoothing. Every step is applied within each sample.

Usage

```
reduce_runs(raw, settings = default_settings(), blank = NULL)
```

Arguments

raw	Raw cycle data from <code>read_runfiles()</code> .
settings	A settings list (see <code>default_settings()</code>).
blank	Optional blank voltages to subtract instead of each run's own blank: a data frame with <code>sample_id</code> and <code>blk88</code> to <code>blk83</code> , e.g. interpolated from baseline runs (see <code>project_baselines()</code>). When given, the start of each run is not treated as blank; laser-off cycles are removed by the 88Sr threshold.

Value

A list with data (one row per integrated cycle) and blank (mean blank voltages per sample, or NULL without blank correction).

```
reference_material_plot
```

Reference material plot

Description

One panel per reference material, against run order or acquisition time. For the primary reference material: each run's measured (uncorrected) mean with 2 SE, the fitted correction line (by default the linear regression) and the corrected run means. For secondary reference materials: uncorrected and corrected run means, with the corrected session averages as horizontal lines. The accepted values are dashed lines.

Usage

```
reference_material_plot(model, results, x = c("order", "time"), runs = NULL)
```

Arguments

model	A model from <code>calibration_model()</code> (NULL if no primary reference material is set).
results	Output of <code>reference_material_results()</code> .
x	"order" (run number in acquisition order) or "time".
runs	Output of <code>project_runs()</code> , used to draw the correction over all runs of each session (otherwise only between its standards).

Details

On a time axis, sessions that are days or months apart leave each session's lines too short to see; in run order (the default) every run takes the same space.

Value

A plotly object.

reference_material_results	<i>Reference material results</i>
----------------------------	-----------------------------------

Description

One row per reference material run: the measured (uncorrected) and corrected mean $^{87}\text{Sr}/^{86}\text{Sr}$, standard errors with and without the correction uncertainty, and the deviation from the accepted value. For secondary reference materials this is the accuracy check of the correction; primary runs are corrected to their accepted value by construction (exactly so with the linear and step fits).

Usage

```
reference_material_results(project)

summarise_reference_materials(project)
```

Arguments

project A project from `open_project()`.

Value

A tibble.

`summarise_reference_materials()`: one row per reference material and session: number of runs, mean and SD of the run means before and after correction, accepted value and deviation in ppm (of the corrected mean).

reject_outliers	<i>Reject outliers in $^{87}\text{Sr}/^{86}\text{Sr}$</i>
-----------------	--

Description

Flags cycles that deviate from the rolling median by more than `sd` rolling standard deviations. Rejected values move from `sr87_sr86` to `sr87_sr86_outlier`.

Usage

```
reject_outliers(data, sd = 3, window = 20L)
```

Arguments

data Data from `correct_ratios()`.

sd Number of standard deviations; 0 turns rejection off.

window Rolling window width (cycles).

Value

data with `is_outlier` and `sr87_sr86_outlier` added.

<code>remove_samples</code>	<i>Remove samples from a project</i>
-----------------------------	--------------------------------------

Description

Removes samples, their regions and cached results from the project. The raw runfile is not deleted but moved to `raw/removed/`, so a removed sample can be recovered and the same file can be added again later. To keep a sample but leave it out of exports, set its status to "excluded" with `update_sample()` instead.

Usage

```
remove_samples(project, sample_ids)
```

Arguments

<code>project</code>	A project from <code>open_project()</code> .
<code>sample_ids</code>	Samples to remove.

Value

The updated project.

<code>report_markdown</code>	<i>Export record as Markdown</i>
------------------------------	----------------------------------

Description

Renders `export_report()` as a Markdown document (`export_report.md` in every export): readable in any text editor, rendered on GitHub and in RStudio, and easy to copy into a lab notebook or a methods section.

Usage

```
report_markdown(report)
```

Arguments

<code>report</code>	Output of <code>export_report()</code> .
---------------------	--

Value

A character vector of Markdown lines.

run_app

Run the IsoFishR app

Description

Opens IsoFishR in a browser. The app walks through six steps: Project (create or open), Settings, Runfiles, Calibration, Workspace and Export. Every change is saved immediately, and a project reopens on the step it was last worked on.

Usage

```
run_app(project = NULL, user = default_user(), ...)
```

Arguments

project	Optional path of a project to open straight away.
user	Name recorded in the project log.
...	Passed to <code>shiny::shinyApp()</code> options, e.g. port or launch.browser.

Value

A Shiny app object; printing it (or calling at the console) runs it.

set_reference_material

Reference materials of a project

Description

`set_reference_material()` adds a reference material or updates its accepted value. `remove_reference_material()` removes one that no run is assigned to. The primary reference material, used to correct samples, is chosen with `update_project_settings(p, list(calibration = list(primary = name)))`; all others are secondary and are reported as accuracy checks.

Usage

```
set_reference_material(
  project,
  name,
  value,
  uncertainty_2s = NA_real_,
  note = ""
)

remove_reference_material(project, name)
```

Arguments

project	A project from <code>open_project()</code> .
name	Reference material name.
value	Accepted $^{87}\text{Sr}/^{86}\text{Sr}$.
uncertainty_2s	Uncertainty of the accepted value (2 sigma).
note	Optional note, e.g. the source of the value.

Value

The updated project.

set_regions	<i>Set the regions of a sample</i>
-------------	------------------------------------

Description

Replaces all regions of one sample. Regions are given in seconds of analysis time (the `time_s` column of reduced data), so they stay in place if the run speed changes.

Usage

```
set_regions(project, sample_id, regions)
```

Arguments

project	A project from <code>open_project()</code> .
sample_id	One sample ID.
regions	A data frame with columns <code>region</code> , <code>start_s</code> and <code>end_s</code> ; an empty data frame removes all regions.

Value

The updated project.

set_run_type	<i>Set the type of runs</i>
--------------	-----------------------------

Description

Marks runs as samples, baseline (gas blank) runs, or measurements of a reference material. Reference materials must first be defined with [set_reference_material\(\)](#).

Usage

```
set_run_type(project, sample_ids, run_type, reference_material = "")
```

Arguments

project	A project from open_project() .
sample_ids	Runs to change.
run_type	"sample", "baseline" or "reference".
reference_material	Name of the reference material (only for run_type = "reference").

Value

The updated project.

set_sample_settings	<i>Per-sample reduction settings</i>
---------------------	--------------------------------------

Description

`set_sample_settings()` overrides project reduction settings, or the run speed, for one sample (e.g. a different blank time). Pass NULL to go back to the project value. `sample_settings()` returns the settings a sample is reduced with.

Usage

```
set_sample_settings(project, sample_id, ...)
```

```
sample_settings(project, sample_id)
```

Arguments

project	A project from open_project() .
sample_id	One sample ID.
...	Named reduction settings, e.g. blank_time_s = 25, or run_speed_um_s.

Value

set_sample_settings(): the updated project. sample_settings(): a settings list.

set_session	<i>Assign runs to an analytical session</i>
-------------	---

Description

Overrides the automatic session of runs, e.g. to split a long day into two sessions after re-tuning. Corrections are calculated within sessions.

Usage

```
set_session(project, sample_ids, session)
```

Arguments

project	A project from open_project() .
sample_ids	Runs to change.
session	Session name; "" returns the runs to their automatic session.

Value

The updated project.

settings_io	<i>Read and write project settings</i>
-------------	--

Description

Settings files are YAML. Values missing from the file fall back to [default_settings\(\)](#), so older settings files keep working when new settings are added.

Usage

```
read_settings(path)
```

```
write_settings(settings, path)
```

Arguments

path	Path to a settings YAML file.
settings	A settings list.

smooth_ma	<i>Moving average of $^{87}\text{Sr}/^{86}\text{Sr}$</i>
-----------	---

Description

Centred moving average with standard deviation, standard error and 95% confidence interval. Missing values (e.g. rejected outliers) are skipped and the standard error uses the number of values actually in each window (v1 always divided by the full window width).

Usage

```
smooth_ma(data, window = 10L)
```

Arguments

data	Data from <code>reject_outliers()</code> .
window	Window width (cycles).

Value

data with `sr87_sr86_ma`, `_ma_sd`, `_ma_n`, `_ma_se`, `_ma_ci95_low` and `_ma_ci95_up` added.

smooth_spline	<i>Adaptive spline fit of $^{87}\text{Sr}/^{86}\text{Sr}$</i>
---------------	--

Description

Fits a penalised regression spline against distance for each sample, with the amount of smoothing chosen from the data:

Usage

```
smooth_spline(data)
```

Arguments

data	Data from <code>reject_outliers()</code> .
------	--

Details

- The penalty is estimated by restricted maximum likelihood (`mgcv: :bam, method = "fREML"`), so noisy profiles are smoothed more and clean ones less.
- The penalty is adaptive (`bs = "ad"`): it varies along the profile, so the curve stays smooth on plateaus but can follow sharp transitions.
- Points are weighted by their total Sr signal, so low-signal (noisier) stretches pull the curve less.
- The basis size grows with the profile length (one basis function per 10 points, 10 to 80); it only caps how wiggly the curve can be.

Profiles of fewer than 60 points get a thin plate spline (`bs = "tp"`) instead, which needs fewer data. The fitted effective degrees of freedom (how flexible the curve ended up, 1 = a straight line) are in `sr87_sr86_spline_edf`.

Value

data with `sr87_sr86_spline`, `sr87_sr86_spline_se` (standard error of the fitted curve) and `sr87_sr86_spline_edf` added.

`suggest_reference_runs`

Suggest reference material runs from their names

Description

Finds runs whose name contains `pattern` (case-insensitive), e.g. "SBO" for the seabass otolith reference material run at UC Davis (see [default_reference_materials\(\)](#)). `apply_reference_suggestion()` marks them as runs of that reference material, defines it with `value` if it isn't defined yet, and makes it the primary reference material if none is set.

Usage

```
suggest_reference_runs(project, pattern = "SBO")

apply_reference_suggestion(
  project,
  pattern = "SBO",
  name = pattern,
  value = default_rm_value(name)
)
```

Arguments

<code>project</code>	A project from open_project() .
<code>pattern</code>	Text to look for in run names.
<code>name</code>	Reference material name.
<code>value</code>	Accepted 87Sr/86Sr used when the reference material is not yet defined (defaults: default_reference_materials()).

Value

suggest_reference_runs(): sample IDs of matching runs that are still samples. apply_reference_suggestion(): the updated project.

summarise_blank	<i>Mean blank voltages per sample</i>
-----------------	---------------------------------------

Description

Uses the cycles recorded before blank_time_s whose 88Sr signal is below raw88_min_v (i.e. before the laser hits the sample).

Usage

```
summarise_blank(data, blank_time_s, raw88_min_v)
```

Arguments

data	Integrated cycle data.
blank_time_s	Length of the blank period (s).
raw88_min_v	88Sr voltage below which a cycle counts as blank.

Value

A tibble with sample_id, blk88 to blk83 and n_blank.

summarise_regions	<i>Summarise regions of a profile</i>
-------------------	---------------------------------------

Description

Summarise regions of a profile

Usage

```
summarise_regions(data, regions, value = "sr87_sr86_value")
```

Arguments

data	Reduced (and usually edited) data.
regions	A data frame with columns sample_id, region (label), start_s and end_s (analysis time). Regions may overlap.
value	Column to summarise; defaults to the value selected by the smoothing setting.

Value

One row per sample and region with the mean, standard deviation, number of points, standard error and distance range.

update_project_settings	<i>Change project settings</i>
-------------------------	--------------------------------

Description

Change project settings

Usage

update_project_settings(project, overrides)

Arguments

- | | |
|-----------|--|
| project | A project from open_project() . |
| overrides | A nested list of settings to change, e.g. <code>list(reduction = list(ma_window = 15L))</code> . |

Value

The updated project.

update_sample	<i>Change a sample's status, comment or edits</i>
---------------	---

Description

Change a sample's status, comment or edits

Usage

update_sample(project, sample_id, ...)

Arguments

- | | |
|-----------|--|
| project | A project from open_project() . |
| sample_id | One sample ID. |
| ... | Named values for any of status ("new", "reviewed", "excluded"), flag, comment, trim_start_s, trim_end_s, reverse, rezero, cp_enabled, cp_method, cp_penalty. |

update_sample

33

Value

The updated project.

Index

- * **datasets**
 - export_project, 8
- add_runfiles, 3
- apply_edits, 4
- apply_reference_suggestion
 - (suggest_reference_runs), 30
- base::strptime(), 20
- calibration_model, 4
- calibration_model(), 17, 19, 22
- check_settings, 5
- check_settings(), 13
- close_project(project), 15
- correct_ratios, 6
- correct_ratios(), 23
- create_example_project, 6
- create_project(project), 15
- default_reference_materials, 7
- default_reference_materials(), 30
- default_settings, 8
- default_settings(), 13–15, 22, 28
- delete_import_preset(import_presets), 11
- EXPORT_FILES(export_project), 8
- export_project, 8
- export_project(), 9, 19
- export_report, 9
- export_report(), 9, 24
- find_changepoints, 10
- find_runfiles, 11
- get_import_preset(import_presets), 11
- import_preset_dir(import_presets), 11
- import_presets, 11
- import_v1_settings, 12
- integrate_cycles, 13
- list_import_presets(import_presets), 11
- list_import_presets(), 15
- merge_settings, 13
- open_project(project), 15
- open_project(), 3, 4, 7, 8, 10, 16–20, 23, 24, 26–28, 30, 32
- plotly::plotly_build(), 14
- profile_plot, 14
- project, 15
- project_baselines, 16
- project_baselines(), 22
- project_data, 16
- project_data(), 14, 17, 19
- project_profiles, 17
- project_profiles(), 9
- project_regions, 18
- project_runs, 18
- project_runs(), 9, 22
- project_settings_table
 - (project_summaries), 19
- project_settings_table(), 9
- project_summaries, 19
- read_acquisition_times, 20
- read_log, 20
- read_runfiles, 21
- read_runfiles(), 13, 22
- read_settings(settings_io), 28
- reduce_runs, 21
- reference_material_plot, 22
- reference_material_results, 23
- reference_material_results(), 9, 22
- reject_outliers, 23
- reject_outliers(), 29
- remove_reference_material
 - (set_reference_material), 25

remove_samples, 24
report_markdown, 24
run_app, 25

sample_settings (set_sample_settings),
27

save_import_preset (import_presets), 11

set_reference_material, 25
set_reference_material(), 27

set_regions, 26

set_run_type, 27

set_sample_settings, 27

set_session, 28
set_session(), 18

settings_io, 28

shiny::shinyApp(), 25

smooth_ma, 29

smooth_spline, 29

suggest_reference_runs, 30
suggest_reference_runs(), 7

summarise_blank, 31

summarise_project_changepoints
(project_summaries), 19

summarise_project_changepoints(), 9

summarise_project_regions
(project_summaries), 19

summarise_project_regions(), 9

summarise_project_samples
(project_summaries), 19

summarise_project_samples(), 9

summarise_reference_materials
(reference_material_results),
23

summarise_reference_materials(), 9

summarise_regions, 31

summarise_regions(), 18

update_project_settings, 32

update_sample, 32

update_sample(), 24

write_settings (settings_io), 28