

Package: Seahtrue (via r-universe)

October 2, 2026

Type Package

Title Seahtrue revives XF data for structured data analysis

Version 1.7.2

Description Seahtrue organizes oxygen consumption and extracellular acidification analysis data from experiments performed on an XF analyzer into structured nested tibbles. This allows for detailed processing of raw data and advanced data visualization and statistics. Seahtrue introduces an open and reproducible way to analyze these XF experiments. It uses file paths to .xlsx files. These .xlsx files are supplied by the user and are generated by the user in the Wave software from Agilent from the assay result files (.asyr). The .xlsx file contains different sheets of important data for the experiment; 1. Assay Information - Details about how the experiment was set up. 2. Rate Data - Information about the OCR and ECAR rates. 3. Raw Data - The original raw data collected during the experiment. 4. Calibration Data - Data related to calibrating the instrument. Seahtrue focuses on getting the specific data needed for analysis. Once this data is extracted, it is prepared for calculations through preprocessing. To make sure everything is accurate, both the initial data and the preprocessed data go through thorough checks.

biocViews CellBasedAssays, FunctionalPrediction, DataRepresentation, DataImport, CellBiology, Cheminformatics, Metabolomics, MicrotitrePlateAssay, Visualization, QualityControl, BatchEffect, ExperimentalDesign, Preprocessing, GO

License Artistic-2.0

Encoding UTF-8

LazyData false

Suggests rmarkdown, knitr, testthat (>= 3.0.0), tidyxl (>= 1.0.8)

Imports dplyr (>= 1.1.2), readxl (>= 1.4.1), logger (>= 0.2.2), purrr (>= 0.3.5), tidyr (>= 1.3.0), lubridate (>= 1.8.0), stringr (>= 1.4.1), tibble (>= 3.1.8), rlang (>= 1.0.0), glue (>= 1.6.2),

cli ($\geq 3.4.1$), janitor ($\geq 2.2.0$), ggplot2 ($\geq 3.5.0$),
RColorBrewer ($\geq 1.1.3$), colorspace ($\geq 2.1.0$), forcats ($\geq 1.0.0$), ggribes ($\geq 0.5.6$), readr ($\geq 2.1.5$), scales ($\geq 1.3.0$), ggrepel, grid

Depends R ($\geq 4.2.0$)

VignetteBuilder knitr

URL <https://vcjdeboer.github.io/seahtrue/>

BugReports <https://github.com/vcjdeboer/seahtrue/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libicu-dev libx11-dev

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

Date/Publication 2026-10-02 14:49:33 UTC

RemoteUrl <https://github.com/bioc/seahtrue>

RemoteRef HEAD

RemoteSha 981999f90a3392986be819bba847864d2f4dd3ee

Contents

calculate_space	2
glue_xfplates	4
plot_bioenergetic_space	5
plot_bioenergetic_trajectory	6
revive_output_donor_A	7
revive_xfplate	8
sketch_assimilate_rate	9
sketch_assimilate_raw	10
sketch_plate	11
sketch_rate	12
Index	14

calculate_space	<i>Calculate Bioenergetic Indices from OCR and ECAR Data</i>
-----------------	--

Description

Computes bioenergetic parameters (OCR/ECAR) including spare capacity, glycolytic index, and supply flexibility index. Compatible with both oligomycin- and monensin-based protocols.

Usage

```
calculate_space(
  rate,
  param_set_ocr,
  param_set_ecar,
  conversion_model = "mookerjee",
  atp_factors = NULL,
  ug_protein_scaling_factor = 20,
  ocr_var = "J_oxphos",
  ecar_var = "J_glyco"
)
```

Arguments

rate	A tibble from the 'rate_data' list of 'revive_xfplate()'.
param_set_ocr	Named character vector defining OCR injections (e.g. 'init_ocr = "m3"').
param_set_ecar	Named character vector defining ECAR injections (e.g. 'om_ecar = "m6"').
conversion_model	Either "mookerjee" (default) or "agilent".
atp_factors	Optional named list with numeric elements 'Jglyco_ecar_factor', 'Jglyco_ocr_factor', and 'Joxphos_ocr_factor'. When supplied, these custom ATP conversion factors override the 'conversion_model' preset.
ug_protein_scaling_factor	Protein content per well (in ug, default = 20).
ocr_var	Name of column containing OCR values (default = "J_oxphos").
ecar_var	Name of column containing ECAR values (default = "J_glyco").

Value

A tibble with one row per group and multiple bioenergetic indices.

Canonical injection names

'param_set_ocr' and 'param_set_ecar' map measurement columns (e.g. "m3") to *canonical* injection names. Downstream metrics and the plotting functions rely on these names. Use one of 'init', 'fccp', 'amrot', 'om', 'mon' suffixed with '_ocr' or '_ecar' (e.g. 'init_ocr', 'amrot_ecar'). Derived columns ('basal_*', 'max_*', 'spare_*', 'supply_index', 'glyco_index') are computed from these. 'plot_bioenergetic_space()' and 'plot_bioenergetic_trajectory()' require 'basal_ocr', 'fccp_ocr', 'amrot_ocr', 'basal_ecar', 'fccp_ecar', 'amrot_ecar'; a warning is issued if 'calculate_space()' cannot produce them.

Examples

```
data <- revive_xfplate(
  system.file("extdata",
    "20191219_SciRep_PBMCS_donor_A.xlsx", package = "seahtrue")
)
```

```
calculate_space(
  rate = purrr::pluck(data, "rate_data", 1),
  param_set_ocr = c(init_ocr = "m3", fccp_ocr = "m4", amrot_ocr = "m9", mon_ocr = "m12"),
  param_set_ecar = c(init_ecar = "m3", fccp_ecar = "m4", amrot_ecar = "m9", mon_ecar = "m12"),
  conversion_model = "mookerjee")
```

glue_xfplates

Glueing multiple plates from a folder

Description

This function takes a folder path and on the available .xlsx files the `revive_xfplate()` function is run and output in one nested tibble.

Usage

```
glue_xfplates(folderpath_seahorse, arg_is_folder)
```

Arguments

<code>folderpath_seahorse</code>	the path to a folder where the .xlsx files are located or a vector of strings pointing to the path of each individual file
<code>arg_is_folder</code>	either TRUE or FALSE. When the input is a vector of path strings use FALSE, is it points to a folder use TRUE

Value

a nested tibble with all files organized in a row

Examples

```
c(
  system.file("extdata",
    "20191219_SciRep_PBMCs_donor_A.xlsx",
    package = "seahtrue"
  ),
  system.file("extdata",
    "20191219_SciRep_PBMCs_donor_A.xlsx",
    package = "seahtrue"
  )
) |>
glue_xfplates(arg_is_folder = FALSE)
```

`plot_bioenergetic_space`*Plot Bioenergetic Space for Each Group*

Description

Visualizes the metabolic space for each group using rectangles, basal points, and maximal capacity for both ECAR and OCR axes. This function expects output from `'calculate_space()'`.

Usage

```
plot_bioenergetic_space(  
  df,  
  ecar_title = "J ATP glycolysis (pmol/min/ug)",  
  ocr_title = "J ATP oxphos (pmol/min/ug)",  
  legend_title = "Group",  
  palette = NULL,  
  title = NULL  
)
```

Arguments

<code>df</code>	A tibble returned by <code>'calculate_space()'</code> . Must include columns: <code>'group'</code> , <code>'basal_ocr'</code> , <code>'fccp_ocr'</code> , <code>'basal_ecar'</code> , <code>'amrot_ecar'</code> .
<code>ecar_title</code>	Label for the x-axis (ECAR). Default: <code>"ECAR (pmol ATP/min/ug)"</code> .
<code>ocr_title</code>	Label for the y-axis (OCR). Default: <code>"OCR (pmol ATP/min/ug)"</code> .
<code>legend_title</code>	Optional title for the legend. Default: <code>"Group"</code> .
<code>palette</code>	Optional named vector of fill colors (e.g., group names as names).
<code>title</code>	Optional plot title (default <code>'NULL'</code>).

Details

The horizontal extent of the metabolic-space rectangle is drawn from `'amrot_ecar'` (the antimycin A / rotenone ECAR), which represents the glycolytic capacity reached after mitochondrial inhibition. This is by design and differs from the `'max_ecar'` value used inside `[calculate_space()]` for the `'supply_index'` and `'glyco_index_max'` metrics, where `'max_ecar'` is taken from `'mon_ecar'` (or `'om_ecar'`). The plot width and those metrics therefore intentionally reflect different injection points.

Value

A `'ggplot'` object

Examples

```
# Simulate output from calculate_space()
df_space <- tibble::tibble(
  group = c("control", "treated"),
  basal_ocr = c(25, 30),
  fccp_ocr = c(50, 60),
  basal_ecar = c(20, 28),
  amrot_ecar = c(40, 42)
)

# Define group colors
palette <- c("control" = "#1b9e77", "treated" = "#d95f02")

# Plot bioenergetic space
plot_bioenergetic_space(
  df = df_space,
  ecar_title = "J ATP glycolysis (pmol/min/ug)",
  ocr_title = "J ATP oxphos (pmol/min/ug)",
  palette = palette,
  title = "Example Bioenergetic Profiles"
)
```

plot_bioenergetic_trajectory

Plot Metabolic Trajectories in J-space

Description

Converts the wide-format ‘df_space’ tibble into a trajectory plot showing the progression of metabolic states (e.g., Baseline -> FCCP -> AM/Rot) for each group in OCR/ECAR space. Arrows indicate direction of metabolic shifts across conditions.

Usage

```
plot_bioenergetic_trajectory(
  df,
  palette = NULL,
  title = "Metabolic J-space",
  label_map = c(Baseline = "Basal", FCCP = "FCCP", `AM/Rot` = "AM/Rot")
)
```

Arguments

df	A tibble as returned by ‘calculate_space()’. Must contain columns: ‘group’, ‘basal_ocr’, ‘fccp_ocr’, ‘amrot_ocr’, ‘basal_ecar’, ‘fccp_ecar’, ‘amrot_ecar’.
palette	Optional named vector of fill colors (names = groups).
title	Optional title for the plot. Default: “Metabolic J-space”.
label_map	Optional named vector to rename default state labels (e.g., ‘c("Baseline" = "Basal")’).

Details

The final (AM/Rot) point of each trajectory is plotted at 'amrot_ecar' (the antimycin A / rotenone ECAR), which represents the glycolytic capacity reached after mitochondrial inhibition. This is by design and differs from the 'max_ecar' value used inside [calculate_space()] for the 'supply_index' and 'glyco_index_max' metrics, where 'max_ecar' is taken from 'mon_ecar' (or 'om_ecar'). The plotted trajectory and those metrics therefore intentionally reflect different injection points.

Value

A 'ggplot' object showing OCR vs ECAR trajectories with labeled points and arrows.

Examples

```
df_space <- tibble::tibble(
  group = c("control", "treated"),
  basal_ocr = c(25, 30), fccp_ocr = c(50, 60), amrot_ocr = c(15, 18),
  basal_ecar = c(20, 28), fccp_ecar = c(22, 32), amrot_ecar = c(18, 21)
)
palette <- c("control" = "#1f78b4", "treated" = "#33a02c")
plot_bioenergetic_trajectory(df_space, palette = palette)
```

revive_output_donor_A *Dataset PBMC donor A*

Description

A dataset containing output for the "20191219_SciRep_PBMCs_donor_A.xlsx" file. This experiment was part of the study that was published in Janssen et al. (2021) Sci rep 11:1662. This is data from PBMCs that were isolated from buffy coats as described in the material and methods section of Janssen et al. (2021). The .xlsx file is the result of a single XF experiment on one plate. This .xlsx file was used as input for the revive_xfplate() function. The output of that function contains the read, preprocessed, and validated data in the form of a nested tibble. The file '20191219_SciRep_PBMCs_donor_A.xlsx' can be found in the inst/extdata directory.

Usage

```
data(revive_output_donor_A)
```

Format

A data frame with 1 row and 7 variables.

plate_id Barcode plate id of the well plate containing the samples

filepath_seahorse Path, and basename to .xlsx file

date_run Date and time when the plate was run

date_processed Date and time this output from revive_xfplate() was generated

assay_info Meta information from 'Assay Configuration' sheet and 'Calibration' sheet

injection_info Dataframe with information from the 'Operation log' sheet

raw_data Preprocessed raw dataframe from 'Raw' sheet

rate_data Preprocessed rate data from 'Rate' sheet

validation_output A list of all validation information, output, and rules that are used

Source

Janssen et al. 2021 Sci Rep 11:1162 <<https://doi.org/10.1038/s41598-021-81217-4>>

revive_xfplate	<i>Read, Preprocess, and Validate Seahorse XF Excel File</i>
----------------	--

Description

This function reads a Seahorse XF assay result '.xlsx' file (exported from Wave), and processes it through reading, preprocessing, and validation steps. It performs QC validation using either default or user-defined quality control ranges.

Usage

```
revive_xfplate(
  filepath_seahorse,
  my_instrument = "XFe96",
  use_default_qc_ranges = TRUE,
  qc_ranges = list()
)
```

Arguments

filepath_seahorse	Character. Absolute path to the Seahorse Excel file. This file should be exported from Agilent Wave and include the expected sheets.
my_instrument	Character. The type of Seahorse instrument used. Default is "XFe96".
use_default_qc_ranges	Logical. If 'TRUE' (default), built-in QC ranges are used: O2 = [50-180] mmHg, pH = [6.8-7.6].
qc_ranges	Optional list of custom QC ranges, created with [define_qc_ranges()]. This is only used if 'use_default_qc_ranges = FALSE'.

Details

This is the main entry point for working with Seahorse '.xlsx' files. It wraps around lower-level functions like [read_xfplate()], [preprocess_xfplate()], and [validate_preprocessed()]. It also performs basic file and sheet existence checks.

Value

A nested tibble with the processed Seahorse plate data, including:

plate_id Barcode of the plate (if available)
filepath_seahorse Path to the ‘.xlsx’ file
date_run Timestamp when the plate was run
date_processed Timestamp when this object was created
assay_info Metadata from the assay configuration and calibration sheets
injection_info Data frame with injection (operation log) info
raw_data Preprocessed raw data from the Raw sheet
rate_data Preprocessed OCR/ECAR data from the Rate sheet
validation_output List-column with validation outputs and QC flags

See Also

[define_qc_ranges()], [validate_preprocessed()]

Examples

```
# Example using internal test file:
revive_xfplate(
  system.file("extdata", "20191219_SciRep_PBMCs_donor_A.xlsx",
    package = "seahtrue"
  )
)
```

sketch_assimilate_rate

Combine multiple revived xf plates into one plot for rate data

Description

In this plot the OCR or ECAR is plotted per group for each plate in a faceted ggplot

Usage

```
sketch_assimilate_rate(my_df, param = "OCR", my_measurements = c(3, 6, 7, 12))
```

Arguments

my_df	a tibble generated by glue_xfplates() with for each row representing a single xf experiment
param	either "OCR" or "ECAR"
my_measurements	the measurements that needs to be in the plot. For example, c(3,6,7,12) for a typical mito stress test.

Value

a ggplot object

Examples

```
suppressMessages(
  c(
    system.file("extdata",
      "20191219_SciRep_PBMCs_donor_A.xlsx",
      package = "seahtrue"
    ),
    system.file("extdata",
      "20191219_SciRep_PBMCs_donor_A.xlsx",
      package = "seahtrue"
    )
  ) |>
  glue_xfplates(arg_is_folder = FALSE) |>
  sketch_assimilate_rate(
    param = "OCR",
    my_measurements = c(3, 4, 9, 12)
  )
)
```

sketch_assimilate_raw *Combine multiple revived xf plates into one plot for raw data*

Description

In this plot the O2, pH, or its emission value at the very first measurement point plotted for all wells from all xfplates that are provided to the function.

Usage

```
sketch_assimilate_raw(my_df, param = "O2_mmHg")
```

Arguments

my_df	a tibble generated by glue_xfplates() with for each row representing a single xf experiment
param	either "O2_mmHg", "pH", "O2_em_corr" or "pH_em_corr"

Value

a ggplot object

Examples

```
suppressMessages(
  c(
    system.file("extdata",
      "20191219_SciRep_PBMCs_donor_A.xlsx",
      package = "seahtrue"
    ),
    system.file("extdata",
      "20191219_SciRep_PBMCs_donor_A.xlsx",
      package = "seahtrue"
    )
  ) |>
  glue_xfplates(arg_is_folder = FALSE) |>
  sketch_assimilate_raw(param = "O2_mmHg")
)
```

sketch_plate

Make an overview of plate layout, with colored groups

Description

A heatmap style ggplot figure with each well labeled with a color for each group

Usage

```
sketch_plate(xfplate, reorder_legend = FALSE)
```

Arguments

xfplate	This the 'raw_data' or the 'rate_data' tibble that is generated by the 'revive_xfplate()' function
reorder_legend	either 'TRUE' or 'FALSE'. When 'TRUE' the groups are ordered based on the number in the character string of the group. It also adds a "__00" after each character string to make the forcats::refactor(group, parse_number(group)) work.

Value

a ggplot object of a 96 well plate with the group layout

Examples

```
system.file("extdata",
  "20191219_SciRep_PBMCs_donor_A.xlsx",
  package = "seahtrue"
) |>
  revive_xfplate() |>
  purrr::pluck("raw_data", 1) |>
  sketch_plate(reorder_legend = TRUE)
```

sketch_rate

Generate a plot for the rate data

Description

The `sketch_rate()` function uses the `rate_data` from the generated output from the `revive_xfplate()` function. The injection info is annotated in the plot, using the information from the injections provided in the original experiment. Several options are available to plot either ECAR/OCR or normalize the data with the values from the normalization cells in the .xlsx file.

A number of validations are performed to check whether the data can be plotted and whether the layout of the plot will not be ruined...

Usage

```
sketch_rate(
  xf_rate,
  param = "OCR",
  normalize = FALSE,
  normalize_unit = "10000 cells",
  take_group_mean = TRUE,
  reorder_legend = FALSE
)
```

Arguments

<code>xf_rate</code>	The 'rate_data' tibble as generated by 'revive_plate'
<code>param</code>	Either "OCR" or "ECAR"
<code>normalize</code>	Either TRUE or FALSE
<code>normalize_unit</code>	any string that will be pasted in the y-axis label when <code>normalize = TRUE</code>
<code>take_group_mean</code>	Either TRUE or FALSE
<code>reorder_legend</code>	Either TRUE or FALSE. When 'TRUE' the groups are ordered based on the number in the character string of the group. It also adds a "__00" after each character string to make the <code>forcats::refactor(group, parse_number(group))</code> work.

Value

a ggplot object

Examples

```
system.file("extdata",
  "20191219_SciRep_PBMCs_donor_A.xlsx",
  package = "seahtrue"
) |>
  revive_xfplate() |>
```

```

    purrr::pluck("rate_data", 1) |>
    sketch_rate(
      param = "OCR",
      reorder_legend = TRUE
    )

system.file("extdata",
  "20191219_SciRep_PBMCs_donor_A.xlsx",
  package = "seahtrue"
) |>
  revive_xfplate() |>
  purrr::pluck("rate_data", 1) |>
  sketch_rate(
    param = "OCR",
    take_group_mean = FALSE,
    reorder_legend = TRUE
  )

system.file("extdata",
  "20191219_SciRep_PBMCs_donor_A.xlsx",
  package = "seahtrue"
) |>
  revive_xfplate() |>
  purrr::pluck("rate_data", 1) |>
  sketch_rate(
    param = "ECAR",
    normalize = TRUE,
    take_group_mean = TRUE,
    reorder_legend = TRUE
  )

system.file("extdata",
  "20191219_SciRep_PBMCs_donor_A.xlsx",
  package = "seahtrue"
) |>
  revive_xfplate() |>
  purrr::pluck("rate_data", 1) |>
  sketch_rate(
    param = "ECAR",
    normalize = TRUE,
    take_group_mean = FALSE,
    reorder_legend = TRUE
  )

```

Index

* **datasets**

- revive_output_donor_A, [7](#)
- calculate_space, [2](#)
- glue_xfplates, [4](#)
- plot_bioenergetic_space, [5](#)
- plot_bioenergetic_trajectory, [6](#)
- revive_output_donor_A, [7](#)
- revive_xfplate, [8](#)
- sketch_assimilate_rate, [9](#)
- sketch_assimilate_raw, [10](#)
- sketch_plate, [11](#)
- sketch_rate, [12](#)