Package 'ceas'

July 22, 2025

Title Cellular Energetics Analysis Software

Version 1.3.0

Description Measuring cellular energetics is essential to understanding a matrix's (e.g. cell, tissue or biofluid) metabolic state. The Agilent Seahorse machine is a common method to measure real-time cellular energetics, but existing analysis tools are highly manual or lack functionality. The Cellular Energetics Analysis Software (ceas) R package fills this analytical gap by providing modular and automated Seahorse data analysis and visualization using the methods described by Mookerjee et al. (2017) <doi:10.1074/jbc.m116.774471>.

URL https://jamespeapen.github.io/ceas/, https://github.com/jamespeapen/ceas/

BugReports https://github.com/jamespeapen/ceas/issues/

Imports data.table, ggplot2, lme4, readxl, stats

Encoding UTF-8

RoxygenNote 7.3.2

License MIT + file LICENSE

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Rachel House [aut, cre] (ORCID: <https://orcid.org/0000-0003-2176-0431>), James P. Eapen [aut] (ORCID: <https://orcid.org/0000-0001-6016-3598>), Hui Shen [fnd] (ORCID: <https://orcid.org/0000-0001-9767-4084>), Carrie R. Graveel [fnd] (ORCID: <https://orcid.org/0000-0001-7251-5642>), Matthew R. Steensma [fnd] (ORCID: <https://orcid.org/0000-0002-9003-6730>), Van Andel Institute [cph]

Maintainer Rachel House <rachel.house@vai.org>

Repository CRAN

Date/Publication 2024-12-21 00:00:02 UTC

Contents

`atp_plot` *ATP Plot*

Description

Generate the ATP Plot

Usage

```
atp_plot(
  energetics,
  model = "ols",
  error_bar = "ci",
  conf_int = 0.95,
  size = 2,
  shape = 16,
  basal_vs_max = "basal",
  glyc_vs_resp = "glyc",
  group_label = "Experimental group",
  sep_reps = FALSE,
  ci_method = "Wald"
)
```

Arguments

energetics	A table of calculated glycolysis and OXPHOS rates. Returned by get_energetics
model	The linear model used to estimate mean and confidence intervals: ordinary least squares ("ols") or mixed-effects ("mixed")
error_bar	Whether to plot error bars as standard deviation ("sd") or confidence intervals ("ci")
conf_int	The confidence interval percentage. Should be between 0 and 1
size	Size of the points
shape	Shape of the points
basal_vs_max	Whether to plot "basal" or "max" respiration
glyc_vs_resp	Whether to plot glycolysis ("glyc") or respiration ("resp")
group_label	Label for the experimental group to populate the legend title
sep_reps	Whether to calculate summary statistics on the groups with replicates combined. The current default FALSE combines replicates, but future releases will default to TRUE providing replicate-specific summaries.
ci_method	The method used to compute confidence intervals for the mixed-effects model: "Wald", "profile", or "boot" passed to lme4::confint.merMod().

Details

Note: When we use the term 'max' in the package documentation we mean the maximal experimental OCR and ECAR values rather than absolute biological maximums.

Value

a ggplot

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
energetics <- get_energetics(
  partitioned_data,
  ph = 7.4,
  pka = 6.093,
  buffer = 0.1
)
atp_plot(energetics, sep_reps = FALSE)

atp_plot(energetics, basal_vs_max = "max", sep_reps = FALSE)

atp_plot(
  energetics,
  basal_vs_max = "basal",
  glyc_vs_resp = "resp",
```

```
  sep_reps = TRUE
)
# to change fill, the geom_point shape number should be between 15 and 25
atp_plot(
  energetics,
  sep_reps = FALSE
) +
  ggplot2::scale_fill_manual(
    values = c("#e36500", "#b52356", "#3cb62d", "#328fe1")
  )

# to change color, use ggplot2::scale_color_manual
atp_plot(energetics, sep_reps = FALSE) +
  ggplot2::scale_color_manual(
    values = c("#e36500", "#b52356", "#3cb62d", "#328fe1")
  )
```

`bioscope_plot` *Bioenergetic Scope Plot*

Description

Generate the Bioenergetic Scope Plot

Usage

```
bioscope_plot(
  energetics,
  model = "ols",
  error_bar = "ci",
  conf_int = 0.95,
  size = 2,
  basal_shape = 1,
  max_shape = 19,
  group_label = "Experimental Group",
  sep_reps = FALSE,
  ci_method = "Wald"
)
```

Arguments

`energetics`	A table of calculated glycolysis and OXPHOS rates. Returned by `get_energetics`
`model`	The linear model used to estimate mean and confidence intervals: ordinary least squares (`"ols"`) or mixed-effects (`"mixed"`)
`error_bar`	Whether to plot error bars as standard deviation (`"sd"`) or confidence intervals (`"ci"`)
`conf_int`	The confidence interval percentage. Should be between 0 and 1

size	Size of the points
basal_shape	Shape of the points for basal values
max_shape	Shape of the points for max values
group_label	Label for the experimental group to populate the legend title
sep_reps	Whether to calculate summary statistics on the groups with replicates combined. The current default FALSE combines replicates, but future releases will default to TRUE providing replicate-specific summaries.
ci_method	The method used to compute confidence intervals for the mixed-effects model: "Wald", "profile", or "boot" passed to lme4::confint.merMod().
bioscope_plot	Creates a 2D plot visualizing the mean and standard deviation basal and maximal ATP production from glycolysis and OXPHOS for each experimental group

Value

a ggplot

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
energetics <- get_energetics(
  partitioned_data,
  ph = 7.4,
  pka = 6.093,
  buffer = 0.1
)
bioscope_plot(energetics, sep_reps = FALSE)

# to change fill, the geom_point shape should be between 15 and 20.
# These shapes are filled without border and will correctly show on the legend.
bioscope_plot(
  energetics,
  sep_reps = TRUE,
  size = 3,
  basal_shape = 2,
  max_shape = 17 # empty and filled triangle
) +
  ggplot2::scale_fill_manual(
    values = c("#e36500", "#b52356", "#3cb62d", "#328fe1")
  )

# to change color, use ggplot2::scale_color_manual
bioscope_plot(energetics, sep_reps = FALSE) +
  ggplot2::scale_color_manual(
    values = c("#e36500", "#b52356", "#3cb62d", "#328fe1")
  )
```

`energetics_lme_summary` *Get mean and confidence intervals from energetics mixed-effects models*

Description

Runs linear mixed-effects models on the ATP measure columns from `get_energetics` with replicates as the random-effect. Estimates mean and confidence intervals for ATP production from glycolysis and OXPHOS at points defined in `partition_data`

Usage

```
energetics_lme_summary(atp_col, energetics, conf_int, ci_method)
```

Arguments

`atp_col`	The column name of the ATP measure - one of "ATP_basal_resp", "ATP_max_resp", "ATP_basal_glyc", "ATP_max_glyc"
`energetics`	a data.table of Seahorse OCR and ECAR rates (from `get_energetics`)
`conf_int`	The confidence interval percentage. Should be between 0 and 1
`ci_method`	The method used to compute confidence intervals for the mixed-effects model: `"Wald"`, `"profile"`, or `"boot"` passed to `lme4::confint.merMod()`.

Value

a data.table with mean and the confidence interval bounds by experimental group

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
energetics <- get_energetics(
  partitioned_data,
  ph = 7.4,
  pka = 6.093,
  buffer = 0.1
)
# Only for one column. For the full energetics table run
# `get_energetics_summary` with `model = "mixed"`.
energetics_lme_summary(
  "ATP_max_resp",
  energetics,
  conf_int = 0.95,
  ci_method = "Wald"
)
```

`energetics_ols_summary` *Get ordinary least squares mean and confidence intervals from energetics*

Description

Helper function to calculate mean and standard deviation of ATP production from glycolysis and OXPHOS at points defined in `partition_data` and with values calculated using the `get_energetics` function. Should only be called from `get_energetics_summary` as the function itself only operaes on a vector without any of the grouping that `get_energetics` does.

Usage

```
energetics_ols_summary(atp_col, error_metric, conf_int)
```

Arguments

`atp_col`	The column name of the ATP measure - one of "ATP_basal_resp", "ATP_max_resp", "ATP_basal_glyc", "ATP_max_glyc"
`error_metric`	Whether to calculate error as standard deviation (`"sd"`) or confidence intervals (`"ci"`)
`conf_int`	The confidence interval percentage. Should be between 0 and 1

Value

a data.table with mean and the confidence interval bounds by experimental group

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
energetics <- get_energetics(partitioned_data, ph = 7.4, pka = 6.093, buffer = 0.1)
# Only for one row and across all groups and replicates.
# For the full correctly grouped energetics table run
# `get_energetics_summary` with `model = "ols"`.
energetics_ols_summary(energetics$ATP_max_resp, error_metric = "ci", conf_int = 0.95)
```

`get_energetics` *Calculate ATP Production from OXPHOS and Glycolysis*

Description

Calculates ATP production from glycolysis and OXPHOS at points defined in patitioned_data

Usage

```
get_energetics(partitioned_data, ph, pka, buffer)
```

Arguments

`partitioned_data`	a data.table of organized Seahorse OCR and ECAR rates based on timepoints from the assay cycle. Returned by `partition_data`
`ph`	pH value for energetics calculation (for XF Media, 7.5)
`pka`	pKa value for energetics calculation (for XF Media, 6.063)
`buffer`	buffer for energetics calculation (for XF Media, 0.1 mpH/pmol H+)

Details

TODO: check that all symbols are defined

Proton production rate (PPR):

$$\text{PPR} = \frac{\text{ECAR value}}{\text{buffer}}$$

$$\text{PPR}_{\text{mito}} = \frac{10^{\text{pH}-\text{pK}_a}}{1+10^{\text{pH}-\text{pK}_a}} \cdot \frac{\text{H}^+}{\text{O}_2} \cdot \text{OCR}$$

calculates the proton production from glucose during its conversion to bicarbonate and H^+ assuming max $\frac{H^+}{O_2}$ of 1

$$\text{PPR}_{\text{glyc}} = \text{PPR} - \text{PPR}_{\text{resp}}$$

calculates the proton production from glucose during its conversion to lactate + H^+

Joules of ATP (JATP) production:

$$\text{ATP}_{\text{glyc}} = \left(\text{PPR}_{\text{glyc}} \cdot \frac{\text{ATP}}{\text{lactate}}\right) + \left(\text{MITO}_{\text{resp}} \cdot 2 \cdot \frac{\text{P}}{\text{O}_{\text{glyc}}}\right)$$

$$\frac{\text{ATP}}{\text{lactate}} = 1$$

with $\frac{P}{O_{glyc}}$ = 0.167 for glucose (0.242 for glycogen).

$$\text{ATP}_{\text{resp}} = \left(\text{coupled MITO}_{\text{resp}} \cdot 2 \cdot \frac{\text{P}}{\text{O}_{\text{oxphos}}}\right) + \left(\text{MITO}_{\text{resp}} \cdot 2 \cdot \frac{\text{P}}{\text{O}_{\text{TCA}}}\right)$$

with $\frac{P}{O_{oxphos}}$ = 2.486 and $\frac{P}{O_{TCA}}$ = 0.167.

Value

a data.table of glycolysis and OXPHOS rates

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
energetics <- get_energetics(
  partitioned_data,
  ph = 7.4,
  pka = 6.093,
  buffer = 0.1
)
head(energetics, n = 10)
```

---

get_energetics_summary

Calculate ATP Production Mean and Standard Deviation

Description

Calculates mean and standard deviation of ATP production from glycolysis and OXPHOS at points defined in partition_data and with values calculated using the get_energetics function via ordinary least squares or a mixed-effects model

Usage

```
get_energetics_summary(
  energetics,
  model = "ols",
  error_metric = "ci",
  conf_int = 0.95,
  sep_reps = FALSE,
  ci_method = "Wald"
)
```

Arguments

energetics	a data.table of Seahorse OCR and ECAR rates (from get_energetics)
model	The model used to estimate mean and confidence intervals: ordinary least squares ("ols") or mixed-effects ("mixed")
error_metric	Whether to calculate error as standard deviation ("sd") or confidence intervals ("ci")
conf_int	The confidence interval percentage. Should be between 0 and 1

sep_reps	Whether to calculate summary statistics on the groups with replicates combined. The current default FALSE combines replicates, but future releases will default to TRUE providing replicate-specific summaries.
ci_method	The method used to compute confidence intervals for the mixed-effects model: "Wald", "profile", or "boot" passed to lme4::confint.merMod().

Details

To get the means and confidence intervals for experiments with replicates, users can either use sep_reps = TRUE to get replicate-level summary statistics or set model = "mixed" to use a linear mixed-effects model on with replicate as the random-effect. The confidence intervals are generated using confint(method = "Wald").

Value

a list of groups from the data

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
energetics_list <- get_energetics(
  partitioned_data,
  ph = 7.4,
  pka = 6.093,
  buffer = 0.1
)
energetics_summary <- get_energetics_summary(energetics_list, sep_reps = FALSE)
head(energetics_summary[, c(1:5)], n = 10)
head(energetics_summary[, c(1, 2, 6, 7)], n = 10)
```

---

get_rate_summary *Rates summary*

Description

Summarize OCR and ECAR as mean and bounded standard deviations or standard error with confidence intervals

Usage

```
get_rate_summary(
  seahorse_rates,
  measure = "OCR",
  assay,
  model = "ols",
```

```
  error_metric = "ci",
  conf_int = 0.95,
  sep_reps = FALSE,
  ci_method = "Wald"
)
```

Arguments

seahorse_rates	data.table Seahorse OCR and ECAR rates (imported using read_data function)
measure	Whether to calculate summary for "OCR" or "ECAR"
assay	What assay to calculate summary for (e.g. "MITO" or "GLYCO")
model	The model used to estimate mean and confidence intervals:
error_metric	Whether to calculate error as standard deviations ("sd") or confidence intervals ("ci")
conf_int	The confidence interval percentage. Should be between 0 and 1
sep_reps	Whether to calculate summary statistics on the groups with replicates combined. The current default FALSE combines replicates, but future releases will default to TRUE providing replicate-specific summaries.
ci_method	The method used to compute confidence intervals for the mixed-effects model: "Wald", "profile", or "boot" passed to lme4::confint.merMod().

Value

a data.table with means, standard deviations/standard error with bounds around the mean(sd or confidence intervals)

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
combined_reps <- get_rate_summary(
  seahorse_rates,
  measure = "OCR",
  assay = "MITO",
  model = "ols",
  error_metric = "ci",
  conf_int = 0.95,
  sep_reps = FALSE
)
head(combined_reps, n = 10)

# separate replicates
sep_reps <- get_rate_summary(
  seahorse_rates,
  measure = "OCR",
  assay = "MITO",
  model = "ols",
  error_metric = "ci",
```

```
  conf_int = 0.95,
  sep_reps = TRUE
)
head(sep_reps, n = 10)

# mixed effects model
reps_as_random_effects <- get_rate_summary(
  seahorse_rates,
  measure = "OCR",
  assay = "MITO",
  model = "mixed",
  error_metric = "ci",
  conf_int = 0.95,
  sep_reps = FALSE
)
head(reps_as_random_effects, n = 10)
```

`make_bioscope_plot` *Bioenergetic Scope Plot Shortcut*

Description

Wrapper to create a 2D plot visualizing the mean and standard deviation basal and maximal ATP production from glycolysis and OXPHOS for each experimental group Create a Bioenergetic scope plot from input Seahorse Wave export, long-form rates excel files

Usage

```
make_bioscope_plot(rep_list, ph, pka, buffer, sheet = 2)
```

Arguments

`rep_list`	A list of Seahorse Wave excel export files. One file per replicate. Group all replicates for a given experiment in a single folder, and write that folder's path in "seahorse_data". You can use 'list.files("seahorse_data") "full.names=TRUE") to get the paths to the files.
`ph`	pH value for energetics calculation (for XF Media, 7.5)
`pka`	pKa value for energetics calculation (for XF Media, 6.063)
`buffer`	buffer for energetics calculation (for XF Media, 0.1 mpH/pmol H+)
`sheet`	The number of the excel sheet containing the long-form Seahorse data. Default is 2 because the long-form output from Seahorse Wave is on sheet 2

Value

a ggplot

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
make_bioscope_plot(rep_list, ph = 7.4, pka = 6.093, buffer = 0.1)
```

`normalize` *Normalize Seahorse data*

Description

Normalizes input data according to a sample normalization measure (e.g. cell number or μg of protein). It assumes your data is background normalized.

Usage

```
normalize(
  seahorse_rates,
  norm_csv,
  norm_column = "well",
  norm_method = "minimum"
)
```

Arguments

`seahorse_rates`	The seahorse rates table read by the `read_data()` function.
`norm_csv`	A csv file with either well or experimental group in column 1 and the sample normalization measure in column 2. Headers are ignored.
`norm_column`	Whether to normalize by `"well"` or `"exp_group"` group. The first column of the normalization csv provided should match this value.
`norm_method`	How to normalize each well or experimental group (specified by `norm_column`): • by its corresponding row in the `norm_csv` (`"self"`) or • by the minimum of the `measure` column in the provided `norm_csv` (`"minimum"`). See details.

Details

This normalization is distinct from the background normalization done by the Wave software. If the data are not background normalized, `read_data()` will output a warning showing rows with OCR, ECAR and PER values greater than 0.

Normalization Methods:

When `norm_method` is set to `"self"`, each OCR, ECAR, and PER value is divided by the measure it"self". OCR and ECAR values are divided by the corresponding raw value in the "measure" column: an intra-well or experimental group normalization. Each normalized value is then interpreted as pmol/min per measure (e.g. pmol/min/cell or pmol/min/μg of protein.

When set to "minimum", each OCR, ECAR, and PER value is normalized by the minimum value in the norm_csv "measure" column. In this method, every "measure" column's value in the provided CSV file is divided by the lowest of the "measure" values to get a normalization factor for each well or experimental group. The OCR, ECAR, and PER values in each well or experimental group are divided by their corresponding normalization factors. Compared to "self", this is an inter-well/experimental group normalization based on the lowest "measure". The results may be interpreted as pmol/min per minimum of the measure (eg: group cell count or μg of protein.)

Value

a normalized seahorse_rates data.table

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
norm_csv <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "^norm.csv", full.names = TRUE)
read.csv(norm_csv)
seahorse_rates <- read_data(rep_list, sheet = 2)
head(seahorse_rates, n = 10)
# normalize by experimental group based on the minimum cell count or protein quantity
seahorse_rates.normalized <- normalize(
  seahorse_rates,
  norm_csv,
  norm_column = "exp_group",
  norm_method = "minimum"
)
head(seahorse_rates.normalized, n = 10)
```

---

partition_data *Organize Seahorse Data*

Description

Organizes Seahorse OCR and ECAR rates based on defined time points (i.e. the Measurement column) during the experiment. This time point can be specified if you are modifying the Mito and Glyco Stress Test (i.e. from 3 measurements per cycle to X measurements)

Usage

```
partition_data(
  seahorse_rates,
 assay_types = list(basal = "MITO", uncoupled = "MITO", maxresp = "MITO", nonmito =
  "MITO", no_glucose_glyc = "GLYCO", glucose_glyc = "GLYCO", max_glyc = "GLYCO"),
  basal_tp = 3,
  uncoupled_tp = 6,
  maxresp_tp = 8,
  nonmito_tp = 12,
```

```
  no_glucose_glyc_tp = 3,
  glucose_glyc_tp = 6,
  max_glyc_tp = 8
)
```

Arguments

seahorse_rates	A data.table of OCR and ECAR rates returned by read_data
assay_types	A list that configures data partitioning based on the type of assay. See details.
basal_tp	Basal respiration time point. Must be less than uncoupled_tp
uncoupled_tp	ATP-coupled respiration time point. Must be less than maxresp_tp
maxresp_tp	Maximal uncoupled respiration time point. Must be less than nonmito_tp
nonmito_tp	Non-mitochondrial respiration time point. Must be larger than maxresp_tp
no_glucose_glyc_tp	No glucose added acidification time point. Must be less than glucose_glyc_tp
glucose_glyc_tp	Glucose-associated acidification time point. Must be less than max_glyc_tp
max_glyc_tp	Maximal acidification time point. Must be less than twodg_glyc_tp

Details

Note: When we use the term 'max' in the package documentation we mean the maximal experimental OCR and ECAR values rather than absolute biological maximums.

partition_data sets up the rates data for ATP calculations by the get_energetics function. To do this, it takes a list assay_types with the named values basal, uncoupled, maxresp, nonmito, no_glucose_glyc, glucose_glyc, and max_glyc. In the default setting, it is configured for an experiment with both Mito and Glyco assays. However, partitioning can be configured for other experimental conditions.

- Only MITO data:

```
partitioned_data <- partition_data(
  seahorse_rates,
  assay_types = list(
    basal = "MITO",
    uncoupled = "MITO",
    maxresp = "MITO",
    nonmito = "MITO",
    no_glucose_glyc = NA,
    glucose_glyc = "MITO",
    max_glyc = NA
  ),
  basal_tp = 3,
  uncoupled_tp = 6,
  maxresp_tp = 8,
  nonmito_tp = 12,
  no_glucose_glyc_tp = NA,
```

```
  glucose_glyc_tp = 3,
  max_glyc_tp = NA
)
```

Respiratory control ratio (RCR) and glycolytic capacity (GC) assay:

```
partitioned_data <- partition_data(
  seahorse_rates,
  assay_types = list(
    basal = "RCR",
    uncoupled = "RCR",
    maxresp = "RCR,"
    nonmito = "RCR",
    no_glucose_glyc = NA,
    glucose_glyc = "GC",
    max_glyc = "GC"
  ),
  basal_tp = 3,
  uncoupled_tp = 6,
  maxresp_tp = 8,
  nonmito_tp = 12,
  no_glucose_glyc = NA,
  glucose_glyc_tp = 3,
  max_glyc_tp = 9
)
```

- Data according to Mookerjee *et al.* 2017 *J Biol Chem*;*292*:7189-207.

```
partitioned_data <- partition_data(
  seahorse_rates,
  assay_types = list(
    basal = "RefAssay",
    uncoupled = "RefAssay",
    maxresp = NA,
    nonmito = "RefAssay",
    no_glucose_glyc = "RefAssay",
    glucose_glyc = "RefAssay",
    max_glyc = NA
  ),
  basal_tp = 5,
  uncoupled_tp = 10,
  nonmito_tp = 12,
  maxresp = NA,
  no_glucose_glyc_tp = 1,
  glucose_glyc_tp = 5,
  max_glyc = NA
)
```

Also see the vignette.

Value

a list of named time points from each assay cycle

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
partitioned_data <- partition_data(seahorse_rates)
```

---

rate_plot *Rate plot*

Description

Generate OCR and ECAR plots

Usage

```
rate_plot(
  seahorse_rates,
  measure = "OCR",
  assay = "MITO",
  model = "ols",
  error_bar = "ci",
  conf_int = 0.95,
  group_label = "Experimental group",
  linewidth = 2,
  sep_reps = FALSE,
  ci_method = "Wald"
)
```

Arguments

seahorse_rates	data.table Seahorse OCR and ECAR rates (imported using read_data function)
measure	Whether to plot "OCR" or "ECAR"
assay	What assay to plot (e.g. "MITO" or "GLYCO")
model	The model used to estimate mean and confidence intervals:
error_bar	Whether to plot error bars as standard deviation ("sd") or confidence intervals ("ci")
conf_int	The confidence interval percentage. Should be between 0 and 1
group_label	Label for the experimental group to populate the legend title
linewidth	Width for the lines, passed to geom_line()

sep_reps	Whether to calculate summary statistics on the groups with replicates combined. The current default FALSE combines replicates, but future releases will default to TRUE providing replicate-specific summaries.
ci_method	The method used to compute confidence intervals for the mixed-effects model: "Wald", "profile", or "boot" passed to lme4::confint.merMod().

Value

a ggplot

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
rate_plot(
  seahorse_rates,
  measure = "OCR",
  error_bar = "ci",
  conf_int = 0.95,
  sep_reps = FALSE
)
rate_plot(
  seahorse_rates,
  measure = "OCR",
  error_bar = "ci",
  conf_int = 0.95,
  sep_reps = TRUE
)
```

---

read_data *Read Seahorse Wave Excel File*

Description

Reads input seahore data from an excel Seahorse Wave File. It assumes your data is background normalized.

Usage

```
read_data(
  rep_list,
  norm = NULL,
  sheet = 2,
  delimiter = " ",
  norm_column = "exp_group",
  norm_method = "minimum"
)
```

Arguments

rep_list	A list of Seahorse Wave excel export files. One file per replicate. If your data is in a directory called "seahorse_data", use `list.files("seahorse_data", pattern = "*.xlsx", full.names = TRUE)` to make a list of the excel files. Add multiple replicates with care - see details.
norm	A csv file with the experimental groups and their normalization values. Leave unset if normalization is not required. See `normalize()`.
sheet	The number of the excel sheet containing the long-form Seahorse data. Default is 2 because the long-form output from Seahorse Wave is on sheet 2
delimiter	The delimiter between the group name and the assay type in the Group column of the wave output. e.g. "Group1 MITO" would use a space character as delimiter.
norm_column	Whether to normalize by `"Well"` or `"exp_group"` column. The first column of the normalization csv provided should match this value.
norm_method	How to normalize each well or experimental group (specified by `norm_column`): • by its corresponding row in the `norm` csv (`"self"`) or • by the minimum of the `measure` column in the provided `norm` csv (`"minimum"`). See the `normalize()` function for more details.

Details

Although ceas enables integration of multiple biological and/or technical replicates, previous work has reported high inter-plate variation (Yepez et. al 2018). If you don't want your replicate data combined, you can either:

- make sure that the names of the common groups between the replicates are different.
- in downstream analyses (`get_energetics_summary`, `bioscope_plot`, `rate_plot`, `atp_plot`), use `sep_reps = TRUE` to do all calculations and plotting separately for each replicate.

NOTE: to maintain backwards compatibility `sep_reps` is currently `FALSE` by default, but will be set to `TRUE` in a future release.

Value

a seahorse_rates table

References

Yépez *et al.* 2018 *OCR-Stats: Robust estimation and statistical testing of mitochondrial respiration activities using Seahorse XF Analyzer PLOS ONE* 2018;**13**:e0199938. doi:10.1371/journal.pone.0199938

Examples

```
rep_list <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "*.xlsx", full.names = TRUE)
seahorse_rates <- read_data(rep_list, sheet = 2)
head(seahorse_rates, n = 10)
```

```
# normalization by well using raw cell count or protein quantity
norm_csv <- system.file("extdata", package = "ceas") |>
  list.files(pattern = "well_norm.csv", full.names = TRUE)
seahorse_rates.norm <- read_data(
  rep_list,
  norm = norm_csv,
  norm_column = "well",
  norm_method = "self",
  sheet = 2
)
head(seahorse_rates.norm, n = 10)
```

Index