Package: ceas (via r-universe)

August 19, 2024

Title Cellular Energetics Analysis Software

Version 1.0.0

Description Analysis and visualization of cellular energetics data from Agilent Seahorse XF96. Cellular energetics is how cells make, use, and distribute units of energy (primarily ATP). Measuring real-time cellular energetics is essential to understanding a tissue or cell's bioenergetic state and fuel dependencies. The Seahorse machine measures a cell's or matrix's oxygen consumption rate (OCR) – a proxy of oxidative phosphorylation – and extracellular acidification rate – a proxy of glycolysis. This package offers flexible and fast analysis and plotting capabilities for such data 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, readxl, stats

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

License MIT + file LICENSE

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Repository https://jamespeapen.r-universe.dev

RemoteUrl https://github.com/jamespeapen/ceas

RemoteRef HEAD

RemoteSha 2d48eaddd3611e4de9375d8c12eb584d60efb5e2

Contents

`atp_plot` *ATP Plot*

Description

Generate the ATP Plot

Usage

```
atp_plot(
  energetics,
  error_bar = "ci",
  conf_int = 0.95,
  size = 2,
  shape = 21,
  basal_vs_max = "basal",
  glyc_vs_resp = "glyc",
  group_label = "Experimental group"
)
```

Arguments

`energetics`	A table of calculated glycolysis and OXPHOS rates. Returned by `get_energetics`
`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

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)

atp_plot(energetics, basal_vs_max = "max")

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

# to change fill, the geom_point shape number should be between 15 and 25
atp_plot(energetics, shape = 21) + # filled circle
  ggplot2::scale_fill_manual(values = c("#e36500", "#b52356", "#3cb62d", "#328fe1"))

# to change color, use ggplot2::scale_color_manual
atp_plot(energetics) +
  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,
  error_bar = "ci",
  conf_int = 0.95,
  size = 2,
  basal_shape = 1,
  max_shape = 19,
  group_label = "Experimental Group"
)
```

Arguments

energetics	A table of calculated glycolysis and OXPHOS rates. Returned by get_energetics
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
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)

# 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, 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) +
  ggplot2::scale_color_manual(
    values = c("#e36500", "#b52356", "#3cb62d", "#328fe1")
  )
```

`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

Usage

```
get_energetics_summary(energetics, error_metric = "ci", conf_int = 0.95)
```

Arguments

energetics	a data.table of Seahorse OCR and ECAR rates (from get_energetics)
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 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)
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,
  error_metric = "ci",
  conf_int = 0.95
)
```

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")
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

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)
rates <- get_rate_summary(
  seahorse_rates,
  measure = "OCR",
  assay = "MCIO",
  error_metric = "ci",
  conf_int = 0.95
)
head(rates, 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 cell number or μg of protein. It assumes your data is background normalized.

Usage

```
normalize(seahorse_rates, norm_csv)
```

Arguments

seahorse_rates	The seahorse rates table read by the read_data() function.
norm_csv	A csv file with the experimental groups in column 1 and cell count or μg of protein in column 2. Headers are ignored.

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.

Value

a normalzed 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)
seahorse_rates.normalized <- normalize(seahorse_rates, norm_csv)
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",
  error_bar = "ci",
  conf_int = 0.95,
  group_label = "Experimental group"
)
```

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")
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

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)
```

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 = " ")
```

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.
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.

Value

a seahorse_rates table

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

Index