Package ‘CalNetExploreR’

August 21, 2024

Type Package

Title A Package to Explore and Visualize Calcium Live Imaging Data

Version 0.1.0

Author Simone Lenci

Maintainer Simo91 <sim.lenci@gmail.com>

Description
CalNetExploreR provides a suite of functions for the analysis and visualization of calcium imaging data. It includes tools for network analysis, community detection, event frequency calculation, and graph-based visualization to compare and explore biological conditions using calcium live imaging datasets. The package is designed to help researchers gain insights into calcium signaling dynamics and the underlying network structures in biological systems.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports ggplot2,
reshape2,
dendextend,
cowplot,
ggpubr,
igraph,
ggraph,
viridis,
RColorBrewer,
ggdendro,
factoextra,
dplyr

RoxygenNote 7.3.2

Contents

`active_cells_percentage` *Calculate the Percentage of Active Cells Over Time*

Description

This function calculates the percentage of active cells over time in a binarized calcium matrix. It can optionally plot the percentage of active cells over time.

Usage

```
active_cells_percentage(
  calcium_matrix_binarized,
  binarize = FALSE,
  plot = FALSE
)
```

Arguments

`calcium_matrix_binarized`
A binarized matrix where each row represents a cell and each column represents a timepoint. This matrix can be generated using the ‘binarize()‘ function.

`binarize`
A logical value indicating whether to binarize the calcium matrix. If TRUE, the function will apply the ‘binarize()‘ function to the calcium_matrix_binarized before calculation. Defaults to FALSE.

`plot`
A logical value indicating whether to generate a plot of the percentage of active cells over time. Defaults to FALSE.

Value

If ‘plot‘ is FALSE, returns a data frame containing the time points and the percentage of active cells. If ‘plot‘ is TRUE, returns a ggplot object of the percentage of active cells over time.

Examples

```
calcium_matrix <- matrix(runif(1000), nrow = 10)
result <- active_cells_percentage(calcium_matrix, binarize = TRUE)
plot <- active_cells_percentage(calcium_matrix, binarize = TRUE, plot = TRUE)
```

binarize *Binarize Calcium Imaging Data*

Description

This function binarizes the timeseries calcium data for each cell using a specified cutoff function.

Usage

```
binarize(
  calcium_matrix,
  cutoff_func = function(x) {
     th <- 2 * sd(x)
     x[x <= th] <- 0
     x[x > th] <-
    1
     return(x)
 }
)
```

Arguments

calcium_matrix	A matrix where each row represents a cell and each column represents a timepoint.
cutoff_func	A function to determine the threshold for binarizing the data. Default is twice the standard deviation of each cell.

Value

A binary matrix where each cell's timeseries data is converted to 0 or 1 based on the cutoff function.

Examples

```
data <- matrix(runif(100), nrow = 10)
binary_data <- binarize(data)
```

coactive_cells *Calculate and Plot Percentage of Coactive Cells Over Time*

Description

This function calculates the percentage of coactive cells over time from a binarized calcium matrix. It returns a data frame with the time points and corresponding percentage of active cells. Optionally, it can also generate a plot.

Usage

```
coactive_cells(binarized_calcium_matrix, binarize = FALSE, plot = FALSE)
```

Arguments

binarized_calcium_matrix
: A binary matrix where each row represents a cell and each column represents a timepoint.

binarize
: A logical value indicating whether to binarize the calcium matrix. If TRUE, the matrix will be binarized using binarize(). Defaults to FALSE.

plot
: A logical value indicating whether to generate a plot of the percentage of coactive cells over time. Defaults to FALSE.

Value

A data frame showing the percentage of coactive cells at each timepoint.

Examples

```
calcium_matrix <- matrix(runif(1000), nrow = 10)
coactive_cells.df <- coactive_cells(calcium_matrix, binarize = TRUE, plot = TRUE)
```

`events_per_min` *Calculate Event Frequency per Minute*

Description

This function calculates the frequency of events per minute for each cell in a binarized calcium matrix.

Usage

```
events_per_min(binarized_calcium_matrix, frame_rate, mean_all = FALSE)
```

Arguments

binarized_calcium_matrix
: A binarized matrix where each row represents a cell and each column represents a timepoint. Can be created with ‘binarize()‘.

frame_rate
: The frame rate of the calcium imaging data (frames per second).

mean_all
: A logical value indicating whether to return the mean frequency of events per minute for all cells. Defaults to FALSE.

Value

A numeric array representing the event frequency per minute for each cell, or a single numeric value representing the mean frequency if ‘mean_all‘ is TRUE.

Examples

```
binarized_data <- matrix(sample(c(0, 1), 100, replace = TRUE), nrow = 10)
frame_rate <- 30
event_frequency <- events_per_min(binarized_data, frame_rate)
mean_event_frequency <- events_per_min(binarized_data, frame_rate, TRUE)
```

make_network *Create a Network object from a binarized calcium matrix*

Description

This function produces a network based on the maximum cross-correlation between cells' calcium activity. The user can specify the lag for the cross-correlation function and the correlation threshold for filtering edges.

Usage

```
make_network(
  binarized_calcium_matrix,
  lag.max = 1,
  correlation_threshold = "none"
)
```

Arguments

`binarized_calcium_matrix`
A binarized matrix where each row represents a cell and each column represents a timepoint. This matrix can be generated using the 'binarize()' function.

`lag.max` The maximum lag to use in the cross-correlation function (CCF). Defaults to 1.

`correlation_threshold`
The threshold value for filtering edges in the network (Pearson's coefficients go from -1 to +1). Set to "none" to disable filtering. Defaults to "none".

Value

An 'igraph' object representing the network of correlated cells.

Examples

```
binarized_calcium_matrix <- matrix(sample(c(0, 1), 1000, replace = TRUE), nrow = 10)
network <- make_network(binarized_calcium_matrix)
network_no_filter <- make_network(binarized_calcium_matrix, correlation_threshold = "none")
```

normalize *Normalize Calcium Imaging Data*

Description

This function normalizes the timeseries calcium data for each cell.

Usage

```
normalize(calcium_matrix)
```

Arguments

`calcium_matrix` A matrix where each row represents a cell and each column represents a timepoint.

Value

A normalized matrix where each cell's timeseries data is scaled to [0, 1].

Examples

```
data <- matrix(runif(100), nrow = 10)
normalized_data <- normalize(data)
```

`pca` *Perform PCA on Calcium Imaging Data*

Description

This function performs Principal Component Analysis (PCA) on the calcium imaging data and optionally plots the scree plot of the eigenvalues.

Usage

```
pca(calcium_matrix, binarize = TRUE, plot = TRUE)
```

Arguments

`calcium_matrix` A matrix where each row represents a cell and each column represents a timepoint.

`binarize` A logical value indicating whether to binarize the calcium matrix before performing PCA. Defaults to TRUE.

`plot` A logical value indicating whether to plot the scree plot of eigenvalues. Defaults to TRUE.

Value

If plot = FALSE, returns a list containing the PCA results and eigenvalues. If plot = TRUE, displays the scree plot and does not return anything.

Examples

```
calcium_matrix <- matrix(runif(1000), nrow = 10)
pca_results <- pca(calcium_matrix, binarize = TRUE, plot = TRUE)
```

plot_network *Plot a Network Graph from Calcium Data*

Description

This function plots a network graph where nodes represent cells and edges represent connections between them. The nodes can be color-coded based on the selected label, either by community membership or by the frequency of events per minute.

Usage

```
plot_network(
  graph,
  coordinates,
  label = "communities",
  cell_ID = "none",
  reverse_y_scale = FALSE,
  frequency_values = NULL,
  correlation_threshold = 0.3
)
```

Arguments

`graph` An igraph object representing the network. Can be created with ‘make_network()‘.

`coordinates` A data frame containing X and Y coordinates for each cell ID. Must include columns "X", "Y", and "Cell".

`label` A character string indicating what to label the cells with. Options are "communities" or "frequency". Defaults to "communities".

`cell_ID` A dataframe of cell IDs (should contain X and Y columns). If set to "none", the nodes will be labeled with their numbers. Defaults to "none".

`reverse_y_scale` A logical value indicating whether to reverse the Y scale in the plot (useful for matching image coordinates). Defaults to FALSE.

`frequency_values` A numeric vector containing the frequency of events per minute for each cell. Required if ‘label = "frequency"‘.

`correlation_threshold` A numeric value specifying the threshold for filtering edges by weight. Set to "none" to disable filtering. Defaults to 0.3.

Value

A ggplot object representing the network graph.

Examples

```
# Simulate a binarized calcium matrix
binarized_calcium_matrix <- matrix(sample(c(0, 1), 100, replace = TRUE), nrow = 10)

# Generate the network graph
```

```
graph <- make_network(binarized_calcium_matrix)

# Simulate XY coordinates for the cells
posXY <- data.frame(X = runif(10), Y = runif(10), Cell = 1:10)

# Simulate frequency values for the cells
frequency_values <- runif(10, 0, 5)

# Plot the network graph with frequency as the label
plot <- plot_network(graph, coordinates = posXY, label = "frequency", frequency_values = frequency_values)
print(plot)
```

population_activity.plt *Generate Population Activity Plots*

Description

Generates a raster plot and line plot after performing hierarchical clustering on the data to sort similar cells.

Usage

```
population_activity.plt(
  binarized_calcium_matrix,
  binarize = FALSE,
  dendrogram = FALSE
)
```

Arguments

`binarized_calcium_matrix`
A binary matrix where each row represents a cell and each column represents a timepoint.

`binarize` A logical value indicating whether to binarize the calcium matrix. If TRUE, the matrix will be binarized using binarize(). Defaults to FALSE.

`dendrogram` A logical value indicating whether to include the dendrogram plot. Defaults to FALSE.

Value

A combined plot showing the raster plot with hierarchical clustering and a line plot of population activity.

Examples

```
calcium_matrix <- matrix(runif(1000), nrow = 10)
plot <- population_activity.plt(calcium_matrix, binarize = TRUE, dendrogram = TRUE)
```

PSD *Power Spectral Density (PSD) Analysis*

Description

This function performs a Power Spectral Density (PSD) analysis on a calcium imaging matrix. The user can specify whether to binarize the matrix before the analysis, set the frame rate, and choose to either plot the PSD or return the resulting data frame.

Usage

```
PSD(calcium_matrix, binarize = TRUE, frame_rate = 0.5, plot = TRUE)
```

Arguments

`calcium_matrix`	A matrix where each row represents a cell and each column represents a timepoint.
`binarize`	A logical value indicating whether to binarize the calcium matrix before performing the PSD analysis. Defaults to TRUE.
`frame_rate`	The frame rate of the calcium imaging data in frames per second. Defaults to 0.5 Hz (2 seconds per frame).
`plot`	A logical value indicating whether to plot the PSD. Defaults to TRUE.

Value

A data frame containing the PSD values and corresponding frequencies. If ‘plot‘ is TRUE, a PSD plot is also displayed.

Examples

```
calcium_matrix <- matrix(runif(1000), nrow = 10)
PSD_results <- PSD(calcium_matrix, binarize = TRUE, frame_rate = 0.5, plot = TRUE)
```

PSD.plt *Plot Power Spectral Density (PSD)*

Description

This function performs a Power Spectral Density (PSD) analysis on a calcium imaging matrix and returns only the PSD plot. The user can specify whether to binarize the matrix before the analysis and set the frame rate.

Usage

```
PSD.plt(calcium_matrix, binarize = TRUE, frame_rate = 0.5)
```

Arguments

`calcium_matrix`	A matrix where each row represents a cell and each column represents a timepoint.
`binarize`	A logical value indicating whether to binarize the calcium matrix before performing the PSD analysis. Defaults to TRUE.
`frame_rate`	The frame rate of the calcium imaging data in frames per second. Defaults to 0.5 Hz (2 seconds per frame).

Value

A ggplot object representing the PSD plot.

Examples

```
calcium_matrix <- matrix(runif(1000), nrow = 10)
PSD_plot <- PSD.plt(calcium_matrix, binarize = TRUE, frame_rate = 0.5)
print(PSD_plot)
```

Index