Package 'SqueakR'

October 12, 2022

Type Package

Title An Experiment Interface for 'DeepSqueak' Bioacoustics Research

Version 1.3.0

Description Data processing and visualizations for rodent vocalizations exported from
'DeepSqueak'. These functions are compatible with the 'SqueakR' Shiny Dashboard,
which can be used to visualize experimental results and analyses.

URL https://osimon81.github.io/SqueakR/,

https://github.com/osimon81/SqueakR/

License MIT + file LICENSE

Imports dplyr, ggcorrplot, ggeasy, gghighlight, ggplot2, ggpubr,
googlesheets4, MASS, plotly, RColorBrewer, readxl, report,
rlist, rstatix, shiny, shinydashboard, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.2.0

LazyData no

Depends R (>= 2.10)

Config/testthat/edition 3

NeedsCompilation no

Author Simon Ogundare [aut, cre] (<https://orcid.org/0000-0002-6243-2779>)

Maintainer Simon Ogundare <simon.ogundare@columbia.edu>

Repository CRAN

Date/Publication 2022-06-28 09:20:04 UTC

R topics documented:

---

add_timepoint_data   *Add Timepoint Data*

Description

Loads in a specific Excel File, and (optional) selects a time subset of data.

Usage

```
add_timepoint_data(data_path, t1 = "", t2 = "")
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
t1	The start time in the recording (in seconds) to start extracting calls
t2	The end time in the recording (in seconds) to stop extracting calls

Value

An object containing the call data (under time constraints if specified)

Examples

```
## Not run: add_timepoint_data(
  data_path = "~/inst/extdata/Example_Mouse_Data.xlsx",
  t1 = 3, t2 = 12
)
## End(Not run)
```

---

add_to_experiment   *Add Scored Data to Experiment Object*

Description

Adds summarized timepoint data (acquired by running 'add_timepoint_data()', followed by 'score_timepoint_data()') to the created experiment object.

Usage

```
add_to_experiment(experiment, added_data)
```

Arguments

experiment	The experiment object
added_data	The scored data object to be added to the experiment

Value

A list object containing statistics and metadata for the entire experiment. The 'groups' and 'experimenters' field will auto-populate based on added data using the 'update_experiment()' function.

Examples

```
## Not run: add_to_experiment(experiment = experiment, added_data = my_data)
```

---

analyze_factor *Analyze Factor*

Description

Compares data across experimental groups, plotting the data as a bar graph with error bars.

Usage

```
analyze_factor(experiment, analysis_factor)
```

Arguments

experiment   The experiment object

analysis_factor

     A string representing the factor to analyze between groups

Value

A bar graph comparing the analysis_factor between groups

Examples

```
## Not run: analyze_factor(experiment = experiment, analysis_factor = "tonality")
```

---

autosqueakRpipeline *Fully-Automatic Experiment Creation*

Description

A pipeline function which creates a new experiment object, and adds data and metadata referenced from an external Google Sheets document.

Usage

```
autosqueakRpipeline()
```

Value

An object containing the full experiment with all data added.

Examples

```
## Not run: expt <- autosqueakRpipeline()
```

---

create_experiment *Create New Experiment*

Description

Creates an experiment object which will contain all data and metadata. This object will be saved and timestamped by date, so if working with this file over different dates, the object will not be overwritten as a new object will be created (to preserve backups).

Usage

```
create_experiment(experiment_name)
```

Arguments

experiment_name

The name of the experiment

Value

A list object containing statistics and metadata for the entire experiment. The 'groups' and 'experimenters' field will auto-populate based on added data using the 'update_experiment()' function.

Examples

```
create_experiment(experiment_name = "My-Project")
```

---

deepsqueak_data *DeepSqueak Example Data*

Description

Data exported from DeepSqueak after processing 'Example Mouse Recording.flac'

Usage

```
data(deepsqueak_data)
```

Format

A `"data.frame"` with 714 rows and 17 columns. The definitions of these parameters are sourced from the link in the references section.

ID The number of the call

Label The label extracted from DeepSqueak

Accepted Whether the call was accepted (as a call)

Score The call score, from DeepSqueak

Begin Time (s) The time when the call started

End Time (s) The time when the call ended

Call Length (s) The length of the call

Principal Frequency (kHz) The median frequency of the call

Low Freq (kHz) The minimum frequency of the call

High Freq (kHz) The maximum frequency of the call

Delta Freq (kHz) The difference between the maximum and minimum call frequencies

Frequency Standard Deviation (kHz) The standard deviation of the call

Slope (kHz/s) The slope of the call

Sinuosity The call contour path length divided by the distance between start and end of the call

Mean Power (dB/Hz)

Tonality The signal/noise ratio

Peak Freq The frequency at the highest power

Source

([DeepSqueak GitHub](#))

References

([DeepSqueak Wiki: Export to Excel Page](#))

Examples

```
data(deepsqueak_data)
```

---

describe_experiment *Describe Experiment*

Description

Lists a condensed summary of data stored in the experiment object.

Usage

```
describe_experiment(experiment)
```

Arguments

experiment   The experiment object to be saved

Value

A list of information about the experiment

Examples

```
## Not run: describe_experiment(experiment = experiment_object)
```

---

plotClusters *3D Call Clusters (Custom Label) Plot*

Description

Plots call clusters within a data-point in a 3D Plotly graph, with principal frequency on the x-axis, mean power on the y-axis, and call length on the z-axis.

Usage

```
plotClusters(data_path)
```

Arguments

data_path   The path to the raw data

Value

3D scatterplot

Examples

```
## Not run: plotClusters(data_path = "path")
```

---

plotContours   *2D Call Contours*

Description

Plots call density against principal frequency and call length as a contoured heatmap. Histograms for these variables are displayed across their respective axes.

Usage

```
plotContours(data_path)
```

Arguments

data_path   The path to the raw data

Value

2D contour plot

Examples

```
## Not run: plotContours(data_path = "path")
```

---

plotCorrelations   *Plot Correlation Matrix*

Description

Plots a box-plot based on principal frequency, grouped by custom labels designated in DeepSqueak.

Usage

```
plotCorrelations(
  data_path,
  graph_title = "Correlation Matrix",
  graph_subtitle = "Correlation coefficients labeled."
)
```

Arguments

data_path   The full path to the Excel file exported from DeepSqueak

graph_title   The title for the graph (there is a default title if not specified)

graph_subtitle   The subtitle for the graph (there is a default subtitle if not specified)

Value

A ggplot2 visualization of the box-plot shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotCorrelations(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

---

plotDeltaHistogram   *Plot Delta Frequency Histogram*

Description

Plots a distribution of delta frequencies as a histogram.

Usage

```
plotDeltaHistogram(
  data_path,
  graph_title = "Delta Frequency-Labeled Histogram",
  graph_subtitle = "Delta Frequency measures the kHz range of each detected call."
)
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
graph_title	The title for the graph (there is a default title if not specified)
graph_subtitle	The subtitle for the graph (there is a default subtitle if not specified)

Value

A ggplot2 visualization of the histogram shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDeltaHistogram(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

```
plotDensitySplitByCustom
```
Plot Density Graph, Splitting Groups by Custom Label

Description

Plots a density graph, separating each custom label group into a separate plot in the figure.

Usage

```
plotDensitySplitByCustom(
  data_path,
  graph_title = "Call Distribution, Split by Custom Category Labels",
  graph_subtitle = "Calls are split by custom labels designated in DeepSqueak.",
  chosen_group = c()
)
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
graph_title	The title for the graph (there is a default title if not specified)
graph_subtitle	The subtitle for the graph (there is a default subtitle if not specified)
chosen_group	Specifying a particular group to be highlighted in the graph

Value

A ggplot2 visualization of the density graph shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDensitySplitByCustom(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

plotDensitySplitByDuration

Plot Density Graph, Splitting Groups by Duration

Description

Plots a density graph, separating each duration group into a separate plot in the figure.

Usage

```
plotDensitySplitByDuration(
  data_path,
  graph_title = "Call Distribution Grouped by Duration (s)",
  graph_subtitle = "Duration groups are rounded to the nearest 0.01 second.",
  chosen_group = c()
)
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
graph_title	The title for the graph (there is a default title if not specified)
graph_subtitle	The subtitle for the graph (there is a default subtitle if not specified)
chosen_group	Specifying a particular group to be highlighted in the graph

Value

A ggplot2 visualization of the density graph shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDensitySplitByDuration(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

---

plotDensitySplitByFrequency

Plot Density Graph, Splitting Groups by Frequency

Description

Plots a density graph, separating each frequency group into a separate plot in the figure.

Usage

```
plotDensitySplitByFrequency(
  data_path,
  graph_title = "Call Distribution, Split by Frequency Range (kHz)",
  graph_subtitle = "Calls are split by frequency ranges of 10 kHz.",
  chosen_group = c()
)
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
graph_title	The title for the graph (there is a default title if not specified)
graph_subtitle	The subtitle for the graph (there is a default subtitle if not specified)
chosen_group	Specifying a particular group to be highlighted in the graph

Value

A ggplot2 visualization of the density graph shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDensitySplitByFrequency(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

---

plotDensityStackedByCustom

Plot Density Graph Stacked by Custom Labels

Description

Plots a density graph, grouped by custom labels set in DeepSqueak, collapsed into a single graph.

Usage

```
plotDensityStackedByCustom(
  data_path,
  graph_title = "Call Distribution Grouped by Custom Category Labels",
 graph_subtitle = "Calls are grouped by custom categories designated in DeepSqueak.",
  chosen_group = c()
)
```

Arguments

data_path   The full path to the Excel file exported from DeepSqueak

graph_title   The title for the graph (there is a default title if not specified)

graph_subtitle The subtitle for the graph (there is a default subtitle if not specified)

chosen_group   Specifying a particular group to be highlighted in the graph

Value

A ggplot2 visualization of the density graph shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDensityStackedByCustom(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

plotDensityStackedByDuration

Plot Density Graph Stacked by Duration

Description

Plots a density graph, grouped by duration of each call (rounded to the nearest 0.01 second), collapsed into a single graph.

Usage

```
plotDensityStackedByDuration(
  data_path,
  graph_title = "Call Distribution Grouped by Duration (s)",
  graph_subtitle = "Duration groups are rounded to the nearest 0.01 second.",
  chosen_group = c()
)
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
graph_title	The title for the graph (there is a default title if not specified)
graph_subtitle	The subtitle for the graph (there is a default subtitle if not specified)
chosen_group	Specifying a particular group to be highlighted in the graph

Value

A ggplot2 visualization of the density graph shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDensityStackedByDuration(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

plotDensityStackedByFrequency

Plot Density Graph Stacked by Frequency

Description

Plots a density graph, grouped by frequency group collapsed into a single graph.

Usage

```
plotDensityStackedByFrequency(
  data_path,
  graph_title = "Call Distribution Grouped by Frequency Range (kHz)",
  graph_subtitle = "Calls are grouped by frequency ranges of 10 kHz.",
  chosen_group = c()
)
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
graph_title	The title for the graph (there is a default title if not specified)
graph_subtitle	The subtitle for the graph (there is a default subtitle if not specified)
chosen_group	Specifying a particular group to be highlighted in the graph

Value

A ggplot2 visualization of the density graph shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotDensityStackedByFrequency(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

plotEthnogram *Plot Ethnogram*

Description

Plots an ethnogram, representing when detected calls occur through the length of the recording.

Usage

```
plotEthnogram(
  data_path,
  graph_title = "Call Ethnogram",
  graph_subtitle = "Calls are indicated by a vertical line."
)
```

Arguments

data_path   The full path to the Excel file exported from DeepSqueak

graph_title   The title for the graph (there is a default title if not specified)

graph_subtitle   The subtitle for the graph (there is a default subtitle if not specified)

Value

A ggplot2 visualization of the ethnogram shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotEthnogram(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

plotEthnogramSplitByTonality
                 Plot Tonality-colored Ethnogram

Description

Plots a tonality ethnogram, representing tonality (clarity) of calls throughout the recording.

Usage

```
plotEthnogramSplitByTonality(
  data_path,
  graph_title = "Ethnogram Split By Tonality",
  graph_subtitle = "Tonality: Signal/noise"
)
```

Arguments

data_path   The full path to the Excel file exported from DeepSqueak

graph_title   The title for the graph (there is a default title if not specified)

graph_subtitle The subtitle for the graph (there is a default subtitle if not specified)

Value

A ggplot2 visualization of the ethnogram shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotEthnogramSplitByTonality(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

---

plotPrincipalBoxplot   *Plot Principal Frequency Box-Plot*

Description

Plots a box-plot based on principal frequency, grouped by custom labels designated in DeepSqueak.

Usage

```
plotPrincipalBoxplot(
  data_path,
  graph_title = "Principal Frequency-Labeled Box-Plot",
 graph_subtitle = "Main frequencies where calls labeled in DeepSqueak predominate."
)
```

Arguments

data_path   The full path to the Excel file exported from DeepSqueak

graph_title   The title for the graph (there is a default title if not specified)

graph_subtitle The subtitle for the graph (there is a default subtitle if not specified)

Value

A ggplot2 visualization of the box-plot shown in the viewer window, which can be manually exported.

Examples

```
## Not run: plotPrincipalBoxplot(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  graph_title = "myTitle", graph_subtitle = "myDescription"
)
## End(Not run)
```

---

plotSummaryPDF *Export PDF*

Description

Plots a density graph, separating each frequency group into a separate plot in the figure.

Usage

```
plotSummaryPDF(data_path, save_path = tempdir(), experimenter = "Anon")
```

Arguments

data_path	The full path to the Excel file exported from DeepSqueak
save_path	The full path to the directory where the PDF will be saved (default is the current working directory)
experimenter	The name of the experimenter for naming the save file (default is "Anon")

Value

A PDF containing a series of relevant ethnograms and density graphs assessing the recording.

Examples

```
## Not run: plotSummaryPDF(
  data_path = "inst/extdata/Example_Mouse_Data.xlsx",
  save_path = tempdir(), experimenter = "experimenter_name"
)
## End(Not run)
```

---

plotSunburstAnimals   *Sunburst Plot for Animal Distribution*

Description

Plots interactive sunburst plot for animal distribution across experimental groups. Currently only valid for between-groups studies.

Usage

```
plotSunburstAnimals(experiment)
```

Arguments

experiment   The experiment object

Value

Sunburst distribution plotting animal distribution

Examples

```
## Not run: plotSunburstAnimals(experiment = experiment)
```

---

plotSunburstExperimenters

Sunburst Plot for Experimenter Distribution

Description

Plots interactive sunburst plot for experimenter distribution across experimental groups. Currently only valid for between-groups studies.

Usage

```
plotSunburstExperimenters(experiment)
```

Arguments

experiment   The experiment object

Value

Sunburst distribution plotting experimenter contribution

Examples

```
## Not run: plotSunburstExperimenters(experiment = experiment)
```

---

plotSurface *3D Surface Plot for Call Datasets*

Description

Plots interactive 3D plot for density of detected calls against call length (s) and principal frequency (kHz), using bivariate kernel density estimation provided via the MASS package

Usage

```
plotSurface(data_path, res = 60)
```

Arguments

data_path	The path to the call data
res	The resolution to render surface plot (default is 60)

Value

3D surface plot for the selected call dataset

Examples

```
## Not run: plotSurface(data_path = "path")
```

---

remove_experiment_data

Remove Data from Experiment File

Description

Removes the data object of interest from the experiment according to a data_id. Removing the data with data_id = 12 removes the 12th dataset added to the experiment.

Usage

```
remove_experiment_data(experiment, data_id)
```

Arguments

`experiment`	The experiment object
`data_id`	The number of the data file to be removed

Value

The experiment file, with the rejected call data removed.

Examples

```
## Not run: experiment <- remove_experiment_data(experiment = experiment_object, data_id = 12)
```

---

save_experiment   *Save Experiment Locally*

Description

Saves the experiment as an .RData file in a specified location.

Usage

```
save_experiment(experiment, save_path)
```

Arguments

`experiment`	The experiment object to be saved
`save_path`	The full path where the experiment object will be saved locally

Value

The experiment object, saved as "[NAME_OF_EXPERIMENT] ([SAVE_DATE]).RData" to the specified location.

Examples

```
## Not run: save_experiment(experiment = experiment_object, save_path = tempdir())
```

---

score_timepoint_data   *Score Timepoint Data*

Description

Transforms data into a list of mean, standard deviation, and range of several call metrics.

Usage

```
score_timepoint_data(data_subset, group, animal, id, experimenter)
```

Arguments

data_subset	The object created in 'add_timepoint_data()' which will be scored
group	The experimental group (i.e. "Control") these data correspond to
animal	The animal or animal group ID for these data
id	The full name of the experiment, including the extension.
experimenter	The experimenter who recorded these results

Value

A list object containing statistics and metadata for each file.

Examples

```
## Not run: score_timepoint_data(
  data_subset = my_subsetted_data, group = "Control",
  experimenter = "Experimenter 1", animal = "4207", id = "Data_sheet.xlsx"
)
## End(Not run)
```

---

semisqueakRpipeline   *Semi-Automatic Experiment Creation*

Description

A pipeline function which creates a new experiment object and adds data by prompting the user for metadata (for every file that is added).

Usage

```
semisqueakRpipeline()
```

Value

An object containing the full experiment with all data added.

Examples

```
## Not run: expt <- semisqueakRpipeline()
```

---

squeakrANOVA   *Generate ANOVA tables for Detected Calls*

Description

Aggregates raw data tables for all data added to experiment object, and calculates and displays ANOVA statistics.

Usage

```
squeakrANOVA(experiment, analysis_factor)
```

Arguments

experiment   The experiment object

analysis_factor

                    A string representing the factor to analyze between groups

Value

A table comparing the analysis_factor between groups

Examples

```
## Not run: squeakrANOVA(experiment = experiment, analysis_factor = "Delta_Freq")
```

---

squeakRDashboard   *Launch SqueakR Dashboard*

Description

Run the SqueakR Shiny Dashboard locally to interface with experimental data.

Usage

```
squeakRDashboard()
```

Value

A Shiny Dashboard

Examples

```
## Not run: squeakRDashboard()
```

---

squeakrSummary   *Summary Statistics for Experiment*

Description

Aggregates raw data for experiment and summarizes descriptive statistics of the current results, grouped by experimental groups.

Usage

```
squeakrSummary(experiment)
```

Arguments

experiment   The experiment object

Value

Text which summarizes experiment data split by group

Examples

```
## Not run: squeakrSTATS(experiment = experiment)
```

---

unblind_all_ids   *Decode Experiment IDs*

Description

Creates a vector of the original call file names, indexed by the order they are listed in the experiment.
This allows experimenters to unblind themselves to the data they collect

Usage

```
unblind_all_ids(experiment)
```

Arguments

experiment   The experiment object

Value

A vector representing the original call file names

Examples

```
## Not run: decode_experiment_ids(experiment)
```

---

unblind_data_id   *Find Matching Experiment ID*

Description

Finds the index of a dataset matching a supplied file name in the experiment.

Usage

```
unblind_data_id(experiment, filename)
```

Arguments

experiment   The experiment object
filename   The full name of the file, including the extension

Value

A number or numbers representing index or indices where that file appears in the experiment

Examples

```
## Not run: unblind_data_id(experiment, "my_data1.xlsx")
```

---

unblind_data_name   *Find Matching Experiment Name*

Description

Finds the name of a dataset matching a supplied index in the set of data

Usage

```
unblind_data_name(experiment, id)
```

Arguments

experiment   The experiment object

id   The dataset number to be unblinded

Value

The name of the original file, corresponding to the data at the requested index

Examples

```
## Not run: unblind_data_name(experiment, 2)
```

---

update_experiment   *Updates Experiment Object Metadata*

Description

Auto-populates 'groups', 'experimenters', and 'animals' fields in experiment object by checking
experimental data (within the experiment object) for new data.

Usage

```
update_experiment(experiment)
```

Arguments

experiment   The experiment object

Value

A list object containing statistics and metadata for the entire experiment. The 'groups', 'experimenters', and 'animals' fields will auto-populate based on added data using the 'update_experiment()'
function.

Examples

```
## Not run: update_experiment(experiment)
```

Index