Package 'LeadSense'

May 7, 2026

Title Medtronic Brain Sense Local Field Potencial Analysis

Version 0.0.2.0

Description Extracts and creates an analysis pipeline for the JSON data files from Brain Sense sessions using Medtronic's Deep Brain Stimulation surgery electrode implants.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, ggplot2, ggpubr, seewave, tidyr, reshape2, signal

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Paulo Bastos [aut, cre],
Raquel Barbosa [aut]

Maintainer Paulo Bastos <pauloandrediasbastos01@gmail.com>

Repository CRAN

Date/Publication 2025-04-21 18:10:08 UTC

Contents

`brain_sense_spectrogram` *Plot BrainSense Spectrograms and Return Data (with Optional Band Filtering)*

Description

This function generates spectrograms for Medtronic BrainSense time-domain signals across one or more data passes. Optionally, the user can select specific passes to plot, filter by frequency band, save the plots, and extract the underlying spectrogram data.

Usage

```
brain_sense_spectrogram(
  dataset = NULL,
  wl = 512,
  ovlp = 75,
  collevels = seq(-80, 0, by = 0.2),
  save_as = NULL,
  output_dir = getwd(),
  passes = NULL,
  band = NULL
)
```

Arguments

`dataset`	A JSON-like object (e.g., parsed with `jsonlite::fromJSON()`) containing Medtronic BrainSense data. If NULL, attempts to load the default dataset from the LeadSense package.
`wl`	Integer. Window length for FFT. Default is 512.
`ovlp`	Numeric. Overlap percentage between successive windows. Default is 75.
`collevels`	Numeric. A sequence of color levels for the spectrogram image (in dB). Default is `seq(-80, 0, by = 0.2)`.
`save_as`	Character. File format to save plots ("png", "pdf", or "jpeg"). If NULL (default), plots are not saved.
`output_dir`	Character. Path to the directory where plots will be saved. Default is current working directory.
`passes`	Integer vector. Indices of passes to plot (e.g., `c(1,3)`). Default is NULL, which plots all available passes.
`band`	Character. One of "Delta", "Theta", "Alpha", "Beta", "Gamma". If provided, filters signal to this frequency band before generating the spectrogram.

Details

WARNING: This function may be computationally intensive and take significant time to execute. Please wait until all plots are rendered.

Value

A list of data frames (invisible). Each data frame corresponds to one spectrogram and contains:

time Time in seconds

frequency Frequency in Hz

magnitude Spectral power in dB

channel Channel label

pass Pass index (i)

Examples

```
brain_sense_spectrogram(dataset, passes = c(2), band = "Beta")
```

`dataset` *JSON list sample session file*

Description

JSON list sample session file

Usage

```
dataset
```

Format

A Large list obtained using jsonlite::JSON("myJSON_sessionFile.json")

AbnormalEnd AbnormalEnd name

FullyReadForSession FullyReadForSession

FeatureInformationCode FeatureInformationCode

SessionDate SessionDate

SessionEndDate SessionEndDate

ProgrammerTimezone ProgrammerTimezone

ProgrammerUtcOffset ProgrammerUtcOffset

ProgrammerLocale ProgrammerLocale

ProgrammerVersion ProgrammerVersion

PatientInformation PatientInformation

DeviceInformation DeviceInformation

BatteryInformation BatteryInformation

GroupUsagePercentage GroupUsagePercentage

LeadConfiguration LeadConfiguration

Stimulation Stimulation

Groups Groups

BatteryReminder BatteryReminder

MostRecentInSessionSignalCheck MostRecentInSessionSignalCheck

Impedance Impedance

GroupHistory GroupHistory

SenseChannelTests SenseChannelTests

CalibrationTests CalibrationTests

LfpMontageTimeDomain LfpMontageTimeDomain

BrainSenseTimeDomain BrainSenseTimeDomain

BrainSenseLfp BrainSenseLfp

LFPMontage LFPMontage

DiagnosticData DiagnosticData

Source

In-house created

Examples

```
data(dataset)  # Lazy loading (!)
```

`impedance_summary` *Extract and summarize Impedance data if available*

Description

This function extracts impedance data from a JSON-like dataset and computes summary statistics.

Usage

```
impedance_summary(dataset = NULL)
```

Arguments

`dataset`	A JSON object/list loaded into the work environment. If NULL, attempts to load the default dataset from the LeadSense package.

Value

A list containing:

- `combined_impedance_df` - The full impedance dataset (if available).
- `impedance_summary` - Summary of mean impedance values by Hemisphere and Type.

If no valid impedance data is found, a message is printed instead.

Examples

```
impedance_results <- impedance_summary(dataset)
print(impedance_results$impedance_summary)
print(impedance_results$combined_impedance_df)
```

`lfp_data` *Extract and summarize LFP data*

Description

This function extracts and summarizes LFP (Local Field Potential) data from a JSON-like dataset.

Usage

```
lfp_data(dataset = NULL)
```

Arguments

`dataset`	A JSON object/list loaded into the work environment. If NULL, attempts to load the default dataset from the LeadSense package.

Value

A structured LFP dataset including:

- Power in each frequency band
- LFP Frequency vs Magnitude for each electrode
- Time-domain signals for all sequences in the LFP montage

Examples

```
lfp_dataset <- lfp_data(dataset)
print(lfp_dataset$band_power_results)
print(lfp_dataset$structured_lfp_dataset)
```

summary_long *Extract basic session summary information in long format*

Description

Extract basic session summary information in long format

Usage

```
summary_long(dataset = NULL)
```

Arguments

dataset	A JSON object/list loaded into the work environment

Value

Long format table with summary session information

Examples

```
summary_long()
```

Index