Package: biosignalEMG (via r-universe)

August 22, 2024

Type Package

Title Tools for Electromyogram Signals (EMG) Analysis

Version 2.1.0

Date 2018-02-18

Author J.A. Guerrero, J.E. Macias-Diaz

Maintainer Antonio Guerrero <jaguerrero@correo.uaa.mx>

Description Data processing tools to compute the rectified, integrated and the averaged EMG. Routines for automatic detection of activation phases. A routine to compute and plot the ensemble average of the EMG. An EMG signal simulator for general purposes.

License GPL (>= 3)

Imports graphics, stats, utils, signal

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-18 08:11:14 UTC

Contents

---

biosignalEMG-package   *Tools for Electromyogram Signals (EMG) Analysis*

Description

Data processing tools to compute the rectified, integrated and the averaged EMG. Routines for automatic detection of activation phases. A routine to compute and plot the ensemble average of the EMG. An EMG signal simulator for general purposes.

Details

Package:	biosignalEMG
Type:	Package
Version:	2.1.0
Date:	2018-02-18
License:	GPL (>= 3)

Standard processing tools (rectification, integration, average, ensemble-average, onset/offset detection) of raw data of an EMG signal.

Author(s)

J.A. Guerrero, J.E. Macias-Diaz

Maintainer: J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Konrad P. (2005) *The ABC of EMG: A Practical Introduction to Kinesiological Electromyography*. Version 1.0, Noraxon INC. USA

Winter D.A., Rau G., Kadefors R., Broman H., Deluca C.J. (1980) Units, Terms, and Sandards in the Reporting of EMG Research: A Report by the AdHoc Committee of the International Society of Electrophysiology and Kinesiology.

Guerrero J.A., Macias-Diaz J.E. (2014) A computational method for the detection of activation/deactivation patterns in biological signals with three levels of electric intensity. *Math. Biosci.* **248**, 117–127.

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 2))

# Simulate 5 seconds of an EMG
emgx <- syntheticemg(n.length.out = 5000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
plot(emgx, main = "Synthetic EMG")

# Full-wave rectified EMG
emgr <- rectification(emgx, rtype = "fullwave")
plot(emgr, main = "Rectified EMG")

# Integration of the full-wave rectified EMG with reset points every
# 200 samples
emgi <- integration(emgr, reset = TRUE, reset.criteria = "samples", vreset = 200)
plot(emgi, main = "Integrated EMG")

# MA-envelope
emgma <- envelope(emgx, method = "MA", wsize = 60)
# Ensemble-averaged EMG
ea <- eaemg(emgma, runs = emgx$on.off, what = 1, timenormalization = "mean",
    scalem = 1, empirical = TRUE, level = 0.9)
plot(ea, lwd = 2, main = "Ensemble-averaged EMG")

# reset graphical parameters
par(op)
```

---

as.emg	*Coerce to an 'emg' object*

Description

Functions to check if an object is an 'emg' object, or coerce it if possible.

Usage

```
as.emg(x, ...)
## Default S3 method:
as.emg(x, ...)
## S3 method for class 'numeric'
as.emg(x, ...)
## S3 method for class 'matrix'
as.emg(x, ...)
## S3 method for class 'data.frame'
as.emg(x, ...)
is.emg(x)
```

Arguments

x   any R object.

...   additional arguments to be passed to methods (such as the samplingrate or the units of the EMG signal).

Details

`as.emg` is a generic function with many methods to coerce an object to an 'emg' object.

`is.emg` checks if x is an 'emg' object.

Value

`as.emg` returns an 'emg' object.

`is.emg` returns TRUE if its argument is an 'emg' object (that is, has "emg" amongst its classes) and FALSE otherwise.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

See Also

[emg](#)

Examples

```
# Creates 1 sec. of a simulated EMG with only baseline noise
x <- rnorm(1000)
# Coerce a vector into an 'emg' object
emgx <- as.emg(x, sampligrate = 1000)
# Plot the data in emgx
plot(emgx, main = "Synthetic EMG noise")

# Load a data.frame with EMG data
data(emg96627009)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg96627009)
# Plot the data in x
plot(x, main = "EMG of limb muscles of decorticated cat")
```

---

dcbiasremoval   *Remove the DC bias*

Description

Adds a constant to all samples in the EMG to move the baseline noise mean in the presence of DC-offset or DC-bias.

Usage

```
dcbiasremoval(data, channel, baseline, data.name)
```

Arguments

data	an 'emg' object.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
baseline	constant to be substracted to the EMG signal. If not provided, this values is set so the resulting EMG has and overall mean of 0.
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.

Value

An 'emg' object.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

See Also

See also emg

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# Creates a random vector to simulate a baseline noise with a DC-offset
x <- rnorm(1000, 1, 1)
emg1 <- emg(x, samplingrate = 1000, units = "mV", data.name = "")
plot(emg1, main = "Simulated EMG with a DC-offset")
abline(h = mean(emg1$values), col = "red", lwd = 2)  # Show the position of the overall mean

# Remove the baseline noise mean
```

```
emg2 <- dcbiasremoval(emg1)
plot(emg2, main = "Simulated EMG with an overall zero mean")
abline(h = mean(emg2$values), col = "red", lwd = 2)  # Show the position of the overall mean

# reset graphical parameters
par(op)
```

---

eaemg   *Ensemble-averaged EMG*

Description

Creates an ensemble-averaged EMG associated with a cyclical activity.

Usage

```
eaemg(data, channel, runs, what, timenormalization = c("min", "mean", "median", "max"),
    scalem = 1, empirical = TRUE, level = 0.9)

## S3 method for class 'eaemg'
plot(x, ci.lty ="dashed", ...)
## S3 method for class 'eaemg'
print(x, ...)
## S3 method for class 'eaemg'
summary(object, ...)
```

Arguments

data	an 'emg' object, usually rectified, moving-average of a rectified EMG or the computed envelope of an EMG.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
runs	an 'rle' object indicating the lengths of succesive phases (active and inactive) or an atomic vector indicating in each position if the signal corresponds to a silence phase (0) or an active phase (1 or a greater integer).
what	which phases are going to be ensemble-averaged, active phases (1) or silence phases (0).
timenormalization	
	a string that indicates with respect to which value all phase durations should be normalized. It can be the "min" (default), "mean", "median" or "max" duration of the phases.
scalem	a value (>=1) that controls the local detail of the ensemble.
empirical	logical; if TRUE (default), the confidence intervals are computed based on the empirical quantiles; if FALSE, confidence intervals are computed asumming a Gaussian distribution.

`level`	a value in $[0, 1]$ indicating the confidence level.
`x`, `object`	an 'eaemg' object.
`ci.lty`	line type to plot the confidence intervals.
`...`	additional arguments to be passed to the generic function.

Details

Ensemble-Averaged EMG is usually applied to EMG signals associated with cyclical activities. In Ensemble-Averaged EMG the time base for each cycle is normalized and all the profiles are averaged. To successfully obtain the average profile, the data should be a rectified EMG, the moving average of a rectified EMG or the envelope of an EMG signal.

The `runs` argument can be computed throught the `rle` function of a thresholded envelope (see **Examples**) or computed with an onset/offset detection algorithm (see `onoff_singlethres` and `onoff_bonato`).

Value

returns an object of class 'eaemg' which is a list with components:

`intervals`	a matrix with three columns, each corresponding to the lower limit of the interval, the ensemble average and the upper limit of the interval.
`empirical`	the empirical argument
`level`	the `level` argument

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Kumar S., Mital A. (1996) *Electromyography In Ergonomics* Taylor & Francis.

Esquivel M.E., Guerrero J.A., Macias-Diaz J.E. (2010) Activity pattern detection in electroneurographic and electromyogram signals through a heteroscedastic change-point method. *Mathematical Biosciences* **224(2)** 109–117

See Also

`rectification`, `movingaverage`, `envelope`, `onoff_singlethres`, `onoff_bonato`

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# Simulate 10 seconds of an EMG
emgx <- syntheticemg(n.length.out = 10000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
```

```
# MA-envelope
emgma <- envelope(emgx, method = "MA", wsize = 60)
# Ensemble-averaged EMG
ea1 <- eaemg(emgma, runs = rle(emgx$on.off), what = 1, timenormalization = "max",
    scalem = 1, empirical = TRUE, level = 0.9)
plot(ea1, lwd = 2, main = "Ensemble-averaged EMG using apriori onset/offset")

# Determine the onset/offset runs using a thresholded envelope
empiricalonoff <- onoff_singlethres(emgx, t = 0.05)
ea2 <- eaemg(emgma, runs = empiricalonoff, what = 1, timenormalization = "max",
    scalem = 1, empirical = TRUE, level = 0.9)
plot(ea2, lwd = 2, main = "Ensemble-averaged EMG using empirical onset/offset")

# reset graphical parameters
par(op)
```

---

emg   *EMG object*

Description

Creates an 'emg' object to store an EMG signal.

Usage

```
emg(data, samplingrate = 0, units = "", data.name = "")
## S3 method for class 'emg'
plot(x, channels = "all", samples = 0, type = "l",  timeunits = c("samples", "seconds"),
 add = FALSE, ...)
## S3 method for class 'emg'
print(x, ...)
## S3 method for class 'emg'
summary(object, ...)
```

Arguments

data	a vector or a matrix (one column per channel) containing the observed data.
samplingrate	number of samples carried per second, measured in Hertz. A value of 0 indicates that this value is missing or unknown.
units	a character vector specifying the units (usually 'mV') of the EMG data.
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.
x, object	an object of class 'emg'.
type	what type of plot should be drawn. See plot for details.
channels	channels to be considered by the procedure. Possible values: "all" to consider all channels in data, a (vector of) character vector(s) specifying the names(s) of the channel(s) to be considered (case sensitive) or a numeric vector specifying the channels to be considered (number(s) of column(s) in data).

samples	samples to be plotted. Possible values: 0 (all samples), an integer 'n' (to plot the first 'n' samples) and a vectors of 2 integers 'n1' and 'n2' (to plot samples from 'n1' to 'n2').
timeunits	a character vector specifying the units (samples or seconds) for the time axis.
add	logical; if TRUE add to an already existing plot; if FALSE start a new plot.
...	additional arguments to be passed to the generic function.

Value

returns an object of class 'emg' which is a list with components:

values	a copy of data
units	the units argument
samplingrate	the samplingrate argument
data.name	the data.name argument

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Konrad P. (2005) The ABC of EMG: A Practical Introduction to Kinesiological Electromyography. Version 1.0, Noraxon INC. USA.

Winter D.A., Rau G., Kadefors R., Broman H., Deluca C.J. (1980) Units, Terms, and Sandards in the Reporting of EMG Research: A Report by the AdHoc Committee of the International Society of Electrophysiology and Kinesiology.

See Also

iemg, eaemg

Examples

```
# Creates a random vector to simulate an EMG signal
x <- rnorm(10000, 0, 1)
# Creates an 'emg' object to store x
emg1 <- emg(x, samplingrate=1000, units="mV", data.name="")

# Summarize and visualize the data
summary(emg1)
plot(emg1, main="Simulated EMG")
```

---

emg95306000   *Sample EMG data from a decorticate cat*

Description

Activity profile of spinal motoneurons recorded during fictive motion (scratching) evoked in an immobilized decorticate cat.

Usage

```
data("emg95306000")
```

Format

A data frame with 1999 observations on the following variable.

emg95306000   a numeric vector of EMG data

Details

The sampling rate of the recorded data was 1000Hz. The data is in "mV".

References

Esquivel M.E., Guerrero J.A., Macias-Diaz J.E. (2010) Activity pattern detection in electroneurographic and electromyogram signals through a heteroscedastic change-point method. *Mathematical Biosciences* **224(2)**, 109–117

Examples

```
# Load data
data(emg95306000)
# Coerce data into an 'emg' object
x <- as.emg(emg95306000, samplingrate=1000, units="mV")
plot(x, main = "EMG of limb muscles of decorticated cat")
```

---

emg96627009   *Sample EMG data from a decorticate cat (4 channels)*

Description

Activity profiles of spinal motoneurons recorded during fictive motion (scratching) evoked in an immobilized decorticate cat.

Usage

```
data("emg96627009")
```

Format

A data frame with 31979 observations on the following 4 variables.

'ENG-PB' a numeric vector of EMG data of the posterior bicep

'ENG-GM' a numeric vector of EMG data of the medial gastrocnemius

'ENG-FDL' a numeric vector of EMG data of the flexor digitorum longus

MOTON. a numeric vector of ENG data of a spinal motoneuron

Details

This object contains the data in the file 'emg96627009.DAT' (provided within the package) which was recorded using AxoTape(TM). The sampling rate of the recorded data was 2500Hz. The data is in "mV" for all the 4 channels.

References

Esquivel M.E., Guerrero J.A., Macias-Diaz J.E. (2010) Activity pattern detection in electroneurographic and electromyogram signals through a heteroscedastic change-point method. *Mathematical Biosciences* **224(2)**, 109–117

Examples

```
# Load data
data(emg96627009)
# Coerce data into an 'emg' object
x <- as.emg(emg96627009, samplingrate=2500, units="mV")
summary(x)
```

---

envelope	*RMS or MA Envelope of an EMG*

Description

Computes the RMS-envelope, the MA-envelope ot the LE-envelope of an EMG signal.

Usage

```
envelope(data, channel, method = c("MA","RMS","LE"), wsize, cutoff, n = 2, data.name, ...)
```

Arguments

data	an 'emg' object.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).

method	a string, "MA" (default) for MA-envelope, "RMS" for the RMS-envelope and "LE" for the LE-envelope.
wsize	determines the length of the moving window that is used to calculate the local averages of the data in RMS and MA envelopes.
cutoff	cutoff frequency in Hertz for the lowpass filter used in LE-envelope.
n	filter order for the lowpass filter used in LE-envelope.
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.
...	Optional parameters for a MA-envelope: `rtype` for a rectification. See `rectification` for details.

Details

The MA-envelope corresponds to the moving average of the rectified EMG.

The RMS-envelope corresponds to the moving average of

$$y_i = (x_i - \mu_x)^2$$

where x_i is each value of the EMG signal and μ_x is the overall mean of the EMG signal.

LE-envelope or linear envelope is lowpass filtered rectified signal.

Value

An 'emg' object containing the envelope of the EMG.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Konrad P. (2005) The ABC of EMG: A Practical Introduction to Kinesiological Electromyography. Version 1.0, Noraxon INC. USA.

See Also

`emg`, `movingaverage`

Examples

```
# Simulate 5 seconds of an EMG
emgx <- syntheticemg(n.length.out = 5000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")

# MA-envelope
emgma <- envelope(emgx, method = "MA", wsize = 60)
# Superimpose the envelope on the EMG
```

```
plot(emgx, main = "MA-envelope")
plot(emgma, add = TRUE, lwd = 4, col = "red")

# RMS-envelope
emgrms <- envelope(emgx, method = "RMS", wsize = 60)
# Superimpose the envelope on the EMG
plot(emgx, main = "RMS-envelope")
plot(emgrms, add = TRUE, lwd = 4, col = "red")

# RMS-envelope
emgle <- envelope(emgx, method = "LE", cutoff = 10) #10Hz
# Superimpose the envelope on the EMG
plot(emgx, main = "LE-envelope")
plot(emgle, add = TRUE, lwd = 4, col = "red")
```

---

extractchannel   *Extracts a single channel from a multichannel 'emg' object*

Description

Extracts a specific channel from a multichannel 'emg' object.

Usage

```
extractchannel(data, channel, data.name)
```

Arguments

data	an 'emg' object.
channel	the specific channel to be extracted. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data (a channel name in data, if it exists).

Details

if channel is a single channel 'emg' object, that single channel is returned as the output.

Value

an 'emg' object containing a single channel.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

See Also

emg, as.emg

Examples

```
# Load a data.frame with EMG data
data(emg96627009)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg96627009)

# Extract the second channel in x
y <- extractchannel(x, 2)
# Plot the data in y
plot(y, main = "Second channel of emg96627009")

# Extract the channel 'ENG-FDL' in x (third channel)
z <- extractchannel(x, "ENG-FDL")
# Plot the data in z
plot(z, main = "Third channel of emg96627009")
```

---

highpass   *Highpass filter an EMG signal*

Description

Highpass filter an EMG signal to remove signal artifacts and baseline noise contamination.

Usage

```
highpass(data, channel, cutoff = 50, n = 5, data.name)
```

Arguments

data	an 'emg' object.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
cutoff	cutoff frequency in Hertz for the highpass filter. See 'Details' for recommended values.
n	filter order for the highpass filter.
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.

Details

This procedure performs a highpass filtering to the EMG signal in order to remove signal artifacts and baseline noise contamination (such as the DC-bias).

There are various recommendations for the value of the cutoff frequency, for example 5Hz (Merletti, 1999) and 10-20Hz (Freriks et.al., 1999).

Value

An 'emg' object which contains the filtered EMG.

Note

This procedure only uses a butterworth type filter of order n. In order to apply a different type of filter, please see the documentation of the 'signal' package.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Merletti R. (1999) Standards for Reporting EMG data. *Journal of Electromyography and Kinesiology* **9(1)**, III-IV http://www.isek-online.org/standards_emg.html

Freriks B., Stegeman D.F. and Hermens H.J. (1999) European Recommendations Standards for Surface Electromyography for Surface Electromyography, results of the SENIAM project. ISBN: 90-75452-14-4.

See Also

dcbiasremoval, whitening, lowpass

Examples

```
# Load a data.frame with EMG data
data(emg95306000)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg95306000, samplingrate = 1000, units = "mV")

# Filter the fourth channel in x
y <- highpass(x, cutoff = 100)

# change graphical parameters to show multiple plots
op <- par(mfcol = c(2, 2))

# Compute the rectified signals to see the effect of the filter
x_rect <- rectification(x)
y_rect <- rectification(y)

# plot the original channel, the filtered channel and the corresponding
# rectified signals
```

```
plot(x, channel = 1, main = "Original  channel")
plot(y, main = "Highpass filtered channel")
plot(x_rect, main = "Rectified  channel")
plot(y_rect, main = "Rectified highpass filtered channel")

# reset graphical parameters
par(op)
```

---

integration   *Integration of an EMG*

Description

Computes the integrated EMG, with or without reset points.

Usage

```
integration(data, channel, reset = FALSE, reset.criteria = c("samples", "time", "value"),
    vreset, units, data.name)
```

Arguments

data	an 'emg' object, usually rectified, a moving averaged rectified EMG or a computed envelope of an EMG.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
reset, reset.criteria	
	if reset=TRUE, The integrated signal is reset to zero at a regular number of samples (reset.criteria="samples"), at regular intervals of time (reset.criteria="time") or when a voltage level is reached (reset.criteria="value").
vreset	a value to determine the reset points in an EMG, see **Details**.
units	a character vector specifying the units (usually "mV.s") of the integrated EMG data.
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.

Details

If reset=FALSE, the integrated EMG is the cumulative sum of the values in data. If reset=TRUE, the cumulative sum is reset to zero at some points, which are determined based on the value of reset.criteria and vreset:

- reset.criteria = samples : reset points are located every vreset samples.
- reset.criteria = time : reset points are located every vreset seconds.
- reset.criteria = value : reset points appear every time a value vreset is reached in the cumulative sum.

Value

An 'iemg' object storing the integrated EMG.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Winter D.A., Rau G., Kadefors R., Broman H., Deluca C.J. (1980) Units, Terms, and Sandards in the Reporting of EMG Research: A Report by the AdHoc Committee of the International Society of Electrophysiology and Kinesiology.

See Also

iemg

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 2))

# Simulate 5 seconds of an EMG
emgx <- syntheticemg(n.length.out = 5000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
plot(emgx, main = "Synthetic EMG")

# Full-wave rectified EMG
emgr <- rectification(emgx, rtype = "fullwave")

# Integration of the full-wave rectified EMG without reset points
emgi1 <- integration(emgr, reset = FALSE)
plot(emgi1, main = "Integrated EMG (no reset)")

# Integration of the full-wave rectified EMG with reset points every
# 200 samples
emgi2 <- integration(emgr, reset = TRUE, reset.criteria = "samples", vreset = 200)
plot(emgi2, main = "Integrated EMG (with reset)")

# Integration of the full-wave rectified EMG with a reset point each
# time the value 200 is reached
emgi3 <- integration(emgr, reset = TRUE, reset.criteria = "value", vreset = 200)
plot(emgi3, main = "Integrated EMG (with reset)")

# reset graphical parameters
par(op)
```

---

loadABF1 *Imports data in the Axon(TM) ABF format v1.0*

Description

Loads ABFv1.0 files containing data from electrophysiological recordings, as created by Axon Instruments/Molecular Devices software such as AxoTape(TM) and AxoScope(TM).

Usage

```
loadABF1(filename, const = 0.00489615)
```

Arguments

filename   name of the ABF file to load.

const   a multiplier to obtain the value represented in each datum (see 'References').

Details

Loads electrophysiology data from Version 1.0 ABF files, as created by software like AxoTape and AxoScope. The routine only loads the data section and some parts of the header to know how many channels are represented in the data.

Value

Returns an object of class 'emg'.

Note

To load an Axon ABFv2.x file, please use the R package **abf2**.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Axon(TM) Binary File Format (ABF): User Guide. http://mdc.custhelp.com/euf/assets/software/FSP_ABFHelp_2.03.pdf

Half-precision floating-point format (2015, June 26). In Wikipedia, The Free Encyclopedia. Retrieved July 9, 2015, https://en.wikipedia.org/w/index.php?title=Half-precision_floating-point_format&oldid=668770694

See Also

[emg](emg)

Examples

```
## Not run:
## Local path to sample data file
path <- paste(system.file("extdata", package = "biosignalEMG"), "/96627009.DAT",
    sep = "")

## Load and show the data
emg <- loadABF1(path)
plot(emg)

## End(Not run)
```

---

lowpass   *Lowpass filter an EMG signal*

Description

Lowpass filter an EMG signal.

Usage

```
lowpass(data, channel, cutoff = 50, n = 5, data.name)
```

Arguments

data	an 'emg' object.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
cutoff	cutoff frequency in Hertz for the lowpass filter.
n	filter order for the lowpass filter.
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.

Details

This procedure performs a lowpass filtering to the EMG signal with delay correction.

Value

An 'emg' object which contains the filtered EMG.

Note

This procedure only uses a butterworth type filter of order n. In order to apply a different type of filter, please see the documentation of the 'signal' package.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

See Also

[dcbiasremoval](), [whitening](), [highpass]()

Examples

```
# This example computes the LE-envelope using the lowpass routine
# Load a data.frame with EMG data
data(emg95306000)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg95306000, samplingrate = 1000, units = "mV")

# Compute the rectified signal
x_rect <- rectification(x)

# Filter the rectified signal
y <- lowpass(x_rect, cutoff = 100)

# change graphical parameters to show multiple plots
op <- par(mfrow = c(3, 1))

# plot the original channel, the filtered channel and the
# LE-envelope
plot(x, channel = 1, main = "Original  channel")
plot(x_rect, main = "Rectified  channel")
plot(y, main = "LE-envelope")

# reset graphical parameters
par(op)
```

---

movingaverage | *Moving average of an EMG*

Description

Smooths an EMG by taking local averages.

Usage

```
movingaverage(data, channel, wsize, units = c("samples", "time"), data.name)
```

Arguments

`data`	an 'emg' object, usually rectified.
`channel`	in case of multi-channel data, `channel` is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in `data`).
`wsize, units`	determines the length of the moving window that is used to calculate the local averages of the data.
`data.name`	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as `data`.

Details

The moving average is a running average computed over a moving window over the length of the EMG. Usually, the EMG signal is first rectified due that, generally, the mean value of an EMG signal is zero.

The window length is the double of the value of `wsize` in samples. The `units` of the window size could be in number of samples ("samples") or in seconds ("time").

Value

An 'emg' object containing the moving-averaged EMG.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Konrad P. (2005) The ABC of EMG: A Practical Introduction to Kinesiological Electromyography. Version 1.0, Noraxon INC. USA.

See Also

[rectification](), [filter]()

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# Simulate 5 seconds of an EMG
emgx <- syntheticemg(n.length.out = 5000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
plot(emgx, main = "Synthetic EMG")

# Compute the moving average of the rectified EMG
```

```
emgr <- rectification(emgx, rtype = "fullwave")
emgma <- movingaverage(emgr, wsize = 30, units = "samples")
plot(emgma, main = "Moving average of the rectified EMG")

# reset graphical parameters
par(op)
```

on/off detection quality measures

Measure the quality of an automatic on/off detection

Description

These quantities measure different aspects of the accuracy in the determination of activity changes of an electric signal.

Usage

```
ANDP(b, bE)
MNChPD(b, bE)
PCE(b, bE)
PR(b, bE, t)
TD(b, bE, t)
```

Arguments

b	reference vector (target values) indicating for each position if the datum corresponds to a silence phase (0) or an active phase (1 or a greater integer indicating the level of activation).
bE	detected vector (output of an automatic detector) indicating for each position if the datum corresponds to a silence phase (0) or an active phase (1 or a greater integer indicating the level of activation).
t	tolerance value (see **Details**).

Details

These quality measures are defined as

ANDP	The absolute difference of the number of detected phases and the actual number of pahses
MNChPD	The mean of the distances of each detected change-point to the nearest actual change-point
PCE	The percentage of incorrectly classified points (silence-activity)
TD	temporal deviation
PR	computes the true positive ratio (TPR) and the false positive ratio (FPR).

TD and PR depends on the value of t, which is a tolerance for the difference between the calculated and exact changepoints.

See Guerrero et.al. (2014) for details on the computation of these measures.

Value

ANDP, MNChPD, PCE, TD: a numeric value. PR: a list of two numeric values (TPR and FPR).

Note

The parameter `t` should be adjusted in terms of the sampling rate of the EMG.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Guerrero J.A., Macias-Diaz J.E. (2014) A computational method for the detection of activation/deactivation patterns in biological signals with three levels of electric intensity. *Math. Biosci.* **248**, 117–127.

Pistohl T., Schmidt T.S.B., Ball T., Schulze-Bonhage A., Aertsen A., Mehring C. (2013) Grasp detection from human ECoG during natural reach-to-grasp movements. *PLoS ONE* **8**

See Also

[onoff_bonato](), [onoff_singlethres]()

Examples

```
# Simulate 10 seconds of an EMG
emgx <- syntheticemg(n.length.out = 10000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")

# Detect the phases of activation in emgx
b_bonato <- onoff_bonato(emgx, sigma_n = 0.05, m = 10, minL = 30)
b_singlet <- onoff_singlethres(emgx, t = 0.2)

# Compute the quality measures
qm_bonato <- c(ANDP(b_bonato, emgx$on.off), MNChPD(b_bonato, emgx$on.off),
    PCE(b_bonato, emgx$on.off), PR(b_bonato, emgx$on.off, t = 10), TD(b_bonato,
        emgx$on.off, t = 10))

qm_singlet <- c(ANDP(b_singlet, emgx$on.off), MNChPD(b_singlet, emgx$on.off),
    PCE(b_singlet, emgx$on.off), PR(b_singlet, emgx$on.off, t = 10), TD(b_singlet,
        emgx$on.off, t = 10))

res <- as.matrix(cbind(qm_bonato, qm_singlet))
rownames(res) <- c("ANDP", "MNChPD", "PCE", "TPR", "FPR", "TD")
print(res)
```

---

onoff_bonato | *Automatic on/off detection algorithm based on a double threshold scheme*

Description

Automatic detection algorithm to determine the times at which muscles "turn on" (activity periods) and "turn off" (silence periods) in an EMG signal.

Usage

```
onoff_bonato(data, channel, sigma_n, Pfa = 0.05, m = 5, r0 = 1, minL = 15, data.name)
```

Arguments

data | an 'emg' object.
channel | in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
sigma_n | standard deviation of the baseline noise (need to be estimated).
Pfa | false-alarm probability (see 'Details').
m | number of samples observed to perform the detection.
r0 | minimum number of samples out of m which must be over the threshold (see Details) to detect the signal.
minL | minimum length allowed of the detected phases.
data.name | a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.

Details

This technique of detection consists of selecting a initial threshold (such that the probability of false alarm is Pfa) and then observing m successive samples: if at least r0 out of the m successive samples are above the threshold the presence of the signal is acknowledged.

Value

A numeric vector with values 0 (silence) and 1 (activity).

Note

The default values of m and minL where choosen for a sampling rate of 500Hz.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Bonato P., D'Alessio T. and Knaflitz M. (1998) A statistical method for the measurement of muscle activation intervals from surface myoelectric signal during gait. *IEEE Transactions on Biomedical Engineering* **45 (3)**, 287–299. ISSN 0018–9294.

See Also

onoff_singlethres

Examples

```
# Load a data.frame with EMG data
data(emg95306000)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg95306000, samplingrate = 1000, units = "mV")

# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# Estimate 'sigma_n' from the last 200 samples (only baseline noise)
sigma_n <- sd(tail(x$values, 200))

# Detect the phases of activation in x
b <- onoff_bonato(x, sigma_n = sigma_n, m = 10, minL = 30)

# Plot 'x' and the detected phases
plot(x, main = "Sample EMG")
plot(b, type = "l", main = "Detected phases (Bonato's method)")

# reset graphical parameters
par(op)
```

---

onoff_singlethres   *Automatic on/off detection algorithm based on a simple threshold rule*

Description

Automatic detection algorithm to determine the times at which muscles "turn on" (activity periods) and "turn off" (silence periods) in an EMG signal.

Usage

```
onoff_singlethres(data, channel, eemg, t = 0.05, data.name)
```

Arguments

`data`	an 'emg' object.
`channel`	in case of multi-channel data, `channel` is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in `data`).
`eemg`	an 'emg' object containing an envelope.
`t`	a threshold value to determine if a datum reresent activity (above the threshold) or silence (below the threshold) in a signal.
`data.name`	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as `data`.

Details

In this procedure, on and off time estimation is determined by the times at which the envelope of the signal (`eemg` or determined using `envelope`) exceeds a threshold.

Value

A numeric vector with values 0 (silence) and 1 (activity).

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Rose W. (2014) Electromyogram Analysis. *Mathematics and Signal Processing for Biomechanics*. http://www.udel.edu/biology/rosewc/kaap686/

See Also

envelope, onoff_bonato

Examples

```
# Load a data.frame with EMG data
data(emg95306000)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg95306000, samplingrate = 1000, units = "mV")

# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# Detect the phases of activation in x
b <- onoff_singlethres(x, t = 0.1)

# Plot 'x' and the detected phases
plot(x, main = "Sample EMG")
```

```
plot(b, type = "l", main = "Detected phases (single thresholding)")

# reset graphical parameters
par(op)
```

---

phasestats *local statistics of each phase on an EMG (beta)*

Description

Computes local statistics "f" of each phase on an EMG signal. The phases are defined by the contiguos points that have the same class.

Usage

```
phasestats(data,class,f)
```

Arguments

data   an 'emg' object.

class   a vector that indicates the class of each point in data.

f   A function that computes the desired local statistic.

Details

class is usually the result of a detection step (0=not active, 1=active). "f" could be the mean, standard deviaton, sum, length, etc.

Value

An object storing the local statistics and the class of each value.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

None

See Also

onoff_singlethres

Examples

```
# Load a data.frame with EMG data
data(emg95306000)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg95306000, samplingrate = 1000, units = "mV")

# Detect the phases of activation in x
b <- onoff_singlethres(x, t = 0.1)

# Computes the length of the detected phases
plengths<-phasestats(x,b,length)
plengths
# Computes the mean of the values within the detected phases
pmeans<-phasestats(x,b,mean)
pmeans

# Only show the means of the active phases
pmeans$stats[names(pmeans$stats)==1]
```

---

rectification   *Full-wave and half-wave rectification of an EMG*

Description

Obtains the full-wave or half-wave rectification of an EMG.

Usage

```
rectification(data, channel, rtype = c("fullwave", "halfwave"), data.name, ...)
```

Arguments

data	an 'emg' object.
channel	in case of multi-channel data, channel is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in data).
rtype	a string indicating if the rectification is full-wave ("fullwave", default) or half-wave ("halfwave").
data.name	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as data.
...	additional arguments to be passed to the generic function.

Details

A full-wave rectified EMG is obtained by computing the absolute value of the raw signal; In half wave rectification, the positive values are passed, while the negatives values are blocked (set to zero).

Rectification is mainly used as a pre-proccessing step before another process like averaging or computing the envelope of the signal.

Value

An 'emg' object which contains the rectified EMG.

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Konrad P. (2005) The ABC of EMG: A Practical Introduction to Kinesiological Electromyography. Version 1.0, Noraxon INC. USA.

See Also

emg, dcbiasremoval, syntheticemg

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(3, 1))

# Simulate 5 seconds of an EMG
emgx <- syntheticemg(n.length.out = 5000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
plot(emgx, main = "Synthetic EMG")

# Full-wave rectified EMG
emgr1 <- rectification(emgx, rtype = "fullwave")
plot(emgr1, main = "Full-wave rectified EMG")

# Half-wave rectified EMG
emgr2 <- rectification(emgx, rtype = "halfwave")
plot(emgr2, main = "Half-wave rectified EMG")

# reset graphical parameters
par(op)
```

syntheticemg	*Synthetic EMG*

Description

Creates a synthetic EMG based on a simple heteroscedastic model.

Usage

```
syntheticemg(n.length.out = 10000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300,
    off.duration.sd = 20, on.mode.pos = 0.75, shape.factor = 0.5,
    samplingrate = 0, units = "", data.name = "Synthetic EMG")
```

Arguments

n.length.out	number of observations.
on.sd	Standard deviation of the random values during an active phase.
on.duration.mean	
	mean lengths of active phases in number of samples.
on.duration.sd	standard deviation of lengths of active phases in number of samples.
off.sd	standard deviation of the random values during an silence phase (i.e. for the baseline noise).
off.duration.mean	
	mean lengths of silence phases in number of samples.
off.duration.sd	
	standard deviation of lengths of silence phases in number of samples.
on.mode.pos	a value between 0 (beginning of the active phase) and 1 (end of the active phase) indicating the relative position in which the maximum amplitude of an active phase will be located.
shape.factor	shape parameter (recommended values between 0 and 2).
samplingrate	sampling rate associated to the synthetic EMG.
units	units associated to the synthetic EMG.
data.name	a string specifying the name of the variable which will appears on the plots.

Details

A random synthetic EMG signal is generated using an heteroscedastic Gaussian model. There are two parameters that control the amplitude of the active and silence phases (on.sd and off.sd), two that control the (random) length of the active phases (on.duration.mean and on.duration.sd), two that control the (random) length of the silence phases (off.duration.mean and off.duration.sd) and two that control the shape of the burst during an active phase (on.mode.pos and shape.factor). A binary vector stores which samples are associated to an active phase for test purposes.

Value

A derived 'emg' object which also contains a binary vector `on.off` to indicate which samples correspond to an active phase (1) and which to a silence phase (0).

Author(s)

J.A. Guerrero `<jaguerrero@correo.uaa.mx>`

References

Esquivel M.E., Guerrero J.A., Macias-Diaz J.E. (2010) Activity pattern detection in electroneurographic and electromyogram signals through a heteroscedastic change-point method. *Mathematical Biosciences* **224(2)** 109–117

See Also

[emg](emg)

Examples

```
# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# Simulate 10 seconds of an EMG
emg1 <- syntheticemg(n.length.out = 10000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.75, shape.factor = 0.5, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
plot(emg1, main = "Synthetic EMG")

# Simulate 10 seconds of an EMG (pulse trains)
emg2 <- syntheticemg(n.length.out = 10000, on.sd = 1, on.duration.mean = 350,
    on.duration.sd = 10, off.sd = 0.05, off.duration.mean = 300, off.duration.sd = 20,
    on.mode.pos = 0.5, shape.factor = 0, samplingrate = 1000, units = "mV",
    data.name = "Synthetic EMG")
plot(emg2, main = "Synthetic EMG (pulse trains)")

# reset graphical parameters
par(op)
```

---

whitening   *Decorrelates an EMG signal*

Description

Decorrelates an EMG signal using an SVD decomposition of a covariance matrix.

Usage

```
whitening(data, channel, method = c("PCA", "ZCA"), k = 4, r = 1, data.name)
```

Arguments

`data`	an 'emg' object.
`channel`	in case of multi-channel data, `channel` is the specific channel to be considered by the procedure. Possible values: a character vector specifying the name of the channel to be considered (case sensitive) or a numeric value specifying the channel to be considered (number of column in `data`).
`method`	the whitening method to be applied to the EMG signal: "PCA" (default) or "ZCA" (see 'References').
`k`	number of columns (see 'Details') to construct the covariance matrix.
`r`	column to be used as the whitened signal (see 'Details').
`data.name`	a string specifying the name of the variable which will appears on the plots. If empty or not provided is taken from the object given as `data`.

Details

Whitening an EMG signal is a preprocessing step that can improve amplitude estimation, onset activation detection and fecature classification.

A whitening transformation reduces autocorrelation within a signal.

The implemented methods are PCA and ZCA, both based on a SVD decomposition of a covariance matrix. The (artificial) matrix is constructed with k shifted versions of the EMG signal. The result is then an matrix with k columns. The output of the whitening procedure is (usually) the first column (r=1).

Value

An 'emg' object storing the whitened EMG.

Note

The output is not fully whitened. The method decorrelates the columns of the artificial matrix, which contains a shifted version of the signal. At the end only one column (which can be autocorrelated) is taken as the output. Also, it works better when the sampling rate is high (low sampling rates tend to produce decorrelated samples).

Author(s)

J.A. Guerrero <jaguerrero@correo.uaa.mx>

References

Implementing PCA/Whitening http://ufldl.stanford.edu/wiki/index.php/Implementing_PCA/Whitening

See Also

[dcbiasremoval](), [highpass]()

Examples

```
# Load a data.frame with EMG data
data(emg95306000)
# Coerce a data.frame into an 'emg' object
x <- as.emg(emg95306000, samplingrate = 1000, units = "mV")

# Whitening x
y <- whitening(x)

# change graphical parameters to show multiple plots
op <- par(mfrow = c(2, 1))

# plot the original channel and the corresponding whitened channel
plot(x, main = "Original  channel")
plot(y, main = "Whitened channel")

# reset graphical parameters
par(op)
```

Index