Package: PhysioEMG (via r-universe)

May 16, 2026

Title EMG Analysis Functions for PhysioExperiment Objects

Version 0.2.0

Author Yusuke Matsui

Maintainer Yusuke Matsui `<you@example.com>`

Description Provides electromyography (EMG) analysis functions for PhysioExperiment objects. Includes envelope extraction (RMS, Hilbert), muscle activation onset detection (Hodges-Bui, Teager-Kaiser), fatigue analysis (median frequency shift), and muscle synergy decomposition (NMF, PCA, ICA).

Depends R (>= 4.2), PhysioCore

Imports SummarizedExperiment, stats

Suggests signal, testthat (>= 3.2.0), knitr, rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Collate 'emg-amplitude.R' 'emg-data.R' 'emg-onset.R' 'emg-fatigue.R' 'emg-decompose.R' 'emg-network.R' 'zzz.R'

RoxygenNote 7.3.3

Config/pak/sysreqs zlib1g-dev

Repository https://x-biosignal.r-universe.dev

Date/Publication 2026-03-16 11:31:18 UTC

RemoteUrl https://github.com/x-biosignal/PhysioEMG

RemoteRef HEAD

RemoteSha 6ea62b85e5cc5ef448f8ed39b4de257564e810f0

Contents

`emgAmplitudeNormalize` *Normalize EMG Amplitude*

Description

Normalizes EMG amplitude data using MVC or peak normalization. MVC normalization divides each channel by its maximum voluntary contraction value, yielding percentage-of-MVC units. Peak normalization divides by the within-trial peak so each channel ranges from 0 to 1.

Usage

```
emgAmplitudeNormalize(
  x,
  method = c("mvc", "peak"),
  mvc_data = NULL,
  assay_name = NULL,
  output_assay = "normalized"
)
```

Arguments

`x`	A PhysioExperiment object.
`method`	Normalization method: "mvc" (maximum voluntary contraction) or "peak" (peak of the trial).
`mvc_data`	A PhysioExperiment containing MVC trial data (required for "mvc" method). Must have the same number of channels.
`assay_name`	Assay to normalize (default: first assay).
`output_assay`	Output assay name (default: "normalized").

Value

A PhysioExperiment object with an additional assay named `output_assay` containing normalized amplitude values. For "peak" normalization, values range from 0 to 1. For "mvc" normalization, values represent proportion of maximum voluntary contraction.

References

De Luca, C.J. (1997). "The use of surface electromyography in biomechanics." Journal of Applied Biomechanics, 13(2), 135-163. doi:10.1123/jab.13.2.135

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgEnvelope()` for computing amplitude envelopes prior to normalization, `emgFatigue()` for fatigue analysis, `emgOnsetDetect()` for muscle activation onset detection

---

`emgCoherenceNetwork` *EMG Coherence Network Analysis*

Description

Builds a static muscle coordination network from pairwise magnitude-squared coherence between EMG channels.

Usage

```
emgCoherenceNetwork(
  x,
  freq_band = NULL,
  channels = NULL,
  nperseg = 256L,
  noverlap = NULL,
  assay_name = NULL,
  aggregate = c("mean", "max", "median"),
  threshold = NULL
)
```

Arguments

`x`	A PhysioExperiment object.
`freq_band`	Optional numeric vector `c(low, high)` in Hz. If NULL, uses all frequencies.
`channels`	Integer vector of channel indices to include. If NULL, uses all.
`nperseg`	Segment length for Welch estimation (default: 256).
`noverlap`	Overlap length (default: `floor(nperseg / 2)`).
`assay_name`	Input assay name. If NULL, uses default assay.
`aggregate`	Aggregation across frequency bins: "mean", "max", or "median".
`threshold`	Optional threshold for binary adjacency matrix.

Value

A list with:

network Numeric matrix (channel x channel) of coherence strength.

adjacency Logical matrix after thresholding, or NULL.

coherence 3D array (freq x channel x channel).

frequencies Frequency vector (Hz) corresponding to `coherence`.

channel_names Channel labels used in the network.

See Also

`emgDynamicWaveletNetwork()` for time-varying networks, `emgInterpretNetworkKG()` for annotation-aware interpretation.

---

emgCoordinationStructure

EMG Coordination Structure Summary from Network Topology

Description

Quantifies higher-order muscle coordination structure from a weighted network matrix, including module structure, efficiency, and node roles.

Usage

```
emgCoordinationStructure(
  network,
  threshold = NULL,
  n_modules = NULL,
  max_modules = 6L,
  directed = FALSE,
  symmetrize = c("mean", "max", "min"),
  normalize = TRUE
)
```

Arguments

`network`	Numeric square matrix (channels x channels) or a list containing `$network`.
`threshold`	Optional edge-weight threshold. Values below threshold are set to zero before topology estimation.
`n_modules`	Optional number of modules. If NULL, chooses a value automatically by maximizing weighted modularity over candidates.
`max_modules`	Maximum number of candidate modules for automatic search.
`directed`	Logical; set TRUE if `network` is directed/asymmetric.
`symmetrize`	Method to convert directed matrices to undirected form: "mean", "max", or "min".
`normalize`	Logical; if TRUE, rescales weights to `[0, 1]`.

Value

A list with:

network Processed undirected weighted network matrix.

node_metrics Data.frame of node-level topology features.

modules Named integer vector of module assignments.

summary Global network topology summary.

See Also

`emgCoherenceNetwork()`, `emgDynamicWaveletNetwork()`

`emgDirectedGCNetwork` *Directed EMG Network via Pairwise Granger Causality*

Description

Estimates a directed muscle coordination network using pairwise Granger causality in the time domain.

Usage

```
emgDirectedGCNetwork(
  x,
  channels = NULL,
  assay_name = NULL,
  max_lag = 10L,
  score = c("f_stat", "delta_r2"),
  threshold = NULL,
  p_value_cutoff = NULL,
  standardize = TRUE
)
```

Arguments

`x`	A PhysioExperiment object.
`channels`	Integer vector of channel indices to include. If NULL, uses all.
`assay_name`	Input assay name. If NULL, uses default assay.
`max_lag`	Lag order (in samples) for autoregressive modeling.
`score`	Directed edge metric: "f_stat" or "delta_r2".
`threshold`	Optional threshold for adjacency based on selected score.
`p_value_cutoff`	Optional p-value threshold for adjacency.
`standardize`	Logical; if TRUE, z-score each channel before GC.

Value

A list with:

network Directed numeric matrix (source x target).

p_values Directed matrix of GC p-values.

adjacency Logical directed matrix, or NULL.

channel_names Channel labels used in the network.

lag Lag order used for modeling.

`emgDynamicWaveletNetwork`

Dynamic Wavelet Coherence Network for EMG

Description

Builds a time-varying coordination network by computing wavelet coherence between channel pairs and aggregating coherence within sliding windows.

Usage

```
emgDynamicWaveletNetwork(
  x,
  frequencies = seq(5, 120, by = 5),
  freq_band = NULL,
  channels = NULL,
  window_sec = 0.5,
  step_sec = 0.1,
  n_cycles = 7,
  smoothing_cycles = 3,
  assay_name = NULL,
  aggregate = c("mean", "max", "median"),
  threshold = NULL,
  respect_coi = TRUE
)
```

Arguments

`x`	A PhysioExperiment object.
`frequencies`	Numeric vector of wavelet center frequencies (Hz).
`freq_band`	Optional numeric vector `c(low, high)` for aggregation.
`channels`	Integer vector of channel indices to include. If NULL, uses all.
`window_sec`	Sliding window length in seconds.
`step_sec`	Sliding window step in seconds.
`n_cycles`	Number of Morlet cycles (default: 7).
`smoothing_cycles`	Smoothing width in cycles (default: 3).
`assay_name`	Input assay name. If NULL, uses default assay.
`aggregate`	Aggregation across time-frequency bins: "mean", "max", or "median".
`threshold`	Optional threshold for binary adjacency network per window.
`respect_coi`	Logical; if TRUE, masks frequencies below COI before aggregation.

Value

A list with:

network 3D array (window x channel x channel).

adjacency Logical 3D array thresholded from `network`, or NULL.

window_times Window center times in seconds.

static_summary Mean network across windows.

frequencies Frequency vector used for wavelet transform.

coi Cone-of-influence frequency at each time sample.

See Also

`emgCoherenceNetwork()` for static spectral network, `emgInterpretNetworkKG()` for annotation-aware interpretation.

`emgEnvelope` *EMG Envelope Extraction*

Description

Extracts the amplitude envelope from EMG signals using various methods. The RMS method computes root mean square over a sliding window. The Hilbert method uses the analytic signal via the Hilbert transform. The lowpass method rectifies the signal and applies a moving-average lowpass filter.

Usage

```
emgEnvelope(
  x,
  method = c("rms", "hilbert", "lowpass"),
  window_ms = 50,
  cutoff = 6,
  assay_name = NULL,
  output_assay = "envelope"
)
```

Arguments

`x`	A PhysioExperiment object with EMG data.
`method`	Envelope method: "rms" (root mean square), "hilbert" (Hilbert transform), or "lowpass" (rectification + lowpass filter).
`window_ms`	Window size in milliseconds for RMS method (default: 50).
`cutoff`	Cutoff frequency in Hz for lowpass method (default: 6).
`assay_name`	Input assay name (default: first assay).
`output_assay`	Output assay name (default: "envelope").

Value

A PhysioExperiment object with an additional assay named `output_assay` containing the amplitude envelope. The envelope matrix has the same dimensions as the input (time x channels) with non-negative values representing instantaneous signal amplitude.

References

De Luca, C.J. (1997). "The use of surface electromyography in biomechanics." Journal of Applied Biomechanics, 13(2), 135-163. doi:10.1123/jab.13.2.135

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgAmplitudeNormalize()` for normalizing envelope values, `emgOnsetDetect()` for onset detection from envelope data, `emgFatigue()` for fatigue analysis using spectral features

---

`emgFatigue` *EMG Fatigue Analysis*

Description

Tracks median and mean frequency over time to assess muscle fatigue. Decreasing median frequency indicates fatigue due to reduced motor unit conduction velocity. The signal is divided into overlapping windows and the power spectral density is computed via FFT for each window.

Usage

```
emgFatigue(x, window_sec = 1, overlap = 0.5, assay_name = NULL)
```

Arguments

`x`	A PhysioExperiment object with EMG data.
`window_sec`	Analysis window in seconds (default: 1.0).
`overlap`	Overlap fraction between windows (default: 0.5).
`assay_name`	Input assay name (default: first assay).

Value

A data.frame with one row per channel per window, containing columns:

channel Integer channel index.

window Integer window number (1-indexed).

time_sec Start time of the window in seconds.

median_freq Median frequency (Hz) at which 50 percent of the spectral power is below.

mean_freq Power-weighted mean frequency (Hz).

rms_amplitude Root mean square amplitude of the window.

References

De Luca, C.J. (1984). "Myoelectrical manifestations of localized muscular fatigue in humans." Critical Reviews in Biomedical Engineering, 11(4), 251-279.

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgFatigueIndex()` for a summary fatigue metric, `emgSpectralMoments()` for spectral moment analysis, `emgEnvelope()` for amplitude envelope extraction

---

`emgFatigueIndex` *EMG Fatigue Index*

Description

Computes a fatigue index as the ratio of final to initial median frequency. Values less than 1 indicate fatigue (frequency decrease). The signal is internally analyzed with `emgFatigue()` using 0.5-second windows at 50 percent overlap, then the initial and final portions are compared.

Usage

```
emgFatigueIndex(x, initial_pct = 0.2, final_pct = 0.2, assay_name = NULL)
```

Arguments

`x`	A PhysioExperiment object with EMG data.
`initial_pct`	Percentage of signal used for initial estimate (default: 0.2).
`final_pct`	Percentage of signal used for final estimate (default: 0.2).
`assay_name`	Input assay name (default: first assay).

Value

A data.frame with one row per channel, containing columns:

channel Integer channel index.

fatigue_index Ratio of final to initial median frequency. Values less than 1 indicate fatigue.

initial_mdf Mean median frequency (Hz) in the initial portion.

final_mdf Mean median frequency (Hz) in the final portion.

References

De Luca, C.J. (1984). "Myoelectrical manifestations of localized muscular fatigue in humans." Critical Reviews in Biomedical Engineering, 11(4), 251-279.

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgFatigue()` for detailed windowed fatigue tracking, `emgSpectralMoments()` for spectral moment analysis, `emgEnvelope()` for amplitude envelope extraction

`emgInterpretNetworkKG` *Interpret EMG Network with Knowledge-Graph Metadata*

Description

Adds metadata/KG context to high-weight network edges. This function does not require a specific backend and works with user-provided tables exported from PhysioAnnotationHub/physioKG workflows.

Usage

```
emgInterpretNetworkKG(
  network,
  node_metadata = NULL,
  kg_edges = NULL,
  threshold = NULL,
  top_n = 20L,
  window = NULL
)
```

Arguments

`network`	A square matrix (channels x channels) or 3D array (window x channels x channels).
`node_metadata`	Optional data.frame containing at least `channel`. Additional columns (e.g. `kg_node`, `muscle_name`, `muscle_group`) are carried into the edge table.
`kg_edges`	Optional data.frame of KG links. Expected columns are `node_a`, `node_b`, and optional `relation`.
`threshold`	Optional edge threshold. Default is 75th percentile of upper-triangle weights.
`top_n`	Maximum number of edges returned after thresholding.
`window`	Optional window index when `network` is a 3D array. If NULL, uses the mean across windows.

Value

A list with:

edge_table Ranked edge table with optional metadata/KG annotations.

threshold Applied threshold value.

network_matrix Matrix used for interpretation.

summary List of summary statistics.

kg_relation_summary Relation counts from matched KG links, or NULL.

`emgOnsetDetect` *Detect EMG Muscle Activation Onset and Offset*

Description

Identifies when muscle activation begins and ends using threshold-based or energy-based methods. The Hodges-Bui method thresholds the rectified signal at a multiple of baseline standard deviations. The Teager-Kaiser method applies the Teager-Kaiser energy operator before thresholding, which can be more sensitive to sudden onsets.

Usage

```
emgOnsetDetect(
  x,
  method = c("hodges_bui", "teager_kaiser"),
  threshold_sd = 3,
  baseline_sec = 0.2,
  min_duration_ms = 50,
  assay_name = NULL
)
```

Arguments

`x`	A PhysioExperiment object with EMG data.
`method`	Detection method: "hodges_bui" (baseline SD threshold) or "teager_kaiser" (Teager-Kaiser energy operator).
`threshold_sd`	Number of baseline SDs above mean for threshold (default: 3).
`baseline_sec`	Duration of baseline period in seconds from signal start (default: 0.2).
`min_duration_ms`	Minimum activation duration in ms to accept (default: 50).
`assay_name`	Input assay name (default: first assay).

Value

A list with two data.frames:

- **onsets** A data.frame with columns `channel` (integer channel index), `sample` (sample index of onset), and `time_sec` (onset time in seconds).
- **offsets** A data.frame with columns `channel`, `sample`, and `time_sec` for each activation offset. Rows correspond to matching onsets.

If no activations are detected, both data.frames have zero rows.

References

Hodges, P.W. & Bui, B.H. (1996). "A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography." Electroencephalography and Clinical Neurophysiology, 101(6), 511-519. doi:10.1016/S0921-884X(96)95190-5

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgEnvelope()` for computing amplitude envelopes, `emgAmplitudeNormalize()` for amplitude normalization, `emgFatigue()` for fatigue analysis

---

`emgPartialCoherenceNetwork`

EMG Partial Coherence Network Analysis

Description

Builds a static network from pairwise partial coherence, estimated from the inverse cross-spectral density matrix at each frequency.

Usage

```
emgPartialCoherenceNetwork(
  x,
  freq_band = NULL,
  channels = NULL,
  nperseg = 256L,
  noverlap = NULL,
  assay_name = NULL,
  aggregate = c("mean", "max", "median"),
  threshold = NULL,
  ridge = 1e-06
)
```

Arguments

`x`	A PhysioExperiment object.
`freq_band`	Optional numeric vector `c(low, high)` in Hz.
`channels`	Integer vector of channel indices to include. If NULL, uses all.
`nperseg`	Segment length for Welch estimation (default: 256).
`noverlap`	Overlap length (default: `floor(nperseg / 2)`).
`assay_name`	Input assay name. If NULL, uses default assay.
`aggregate`	Aggregation across frequency bins: "mean", "max", or "median".
`threshold`	Optional threshold for binary adjacency matrix.
`ridge`	Ridge regularization added to spectral matrix inversion.

Value

A list with:

network Numeric matrix (channel x channel) of partial coherence.

adjacency Logical matrix after thresholding, or NULL.

partial_coherence 3D array (freq x channel x channel).

frequencies Frequency vector (Hz).

channel_names Channel labels used in the network.

See Also

`emgCoherenceNetwork()`, `emgWPLINetwork()`

`emgSpectralMoments` *EMG Spectral Moments*

Description

Computes spectral moments (M0, M1, M2) over sliding windows. M0 is total power, M1 is the first spectral moment (related to mean frequency), and M2 is the second moment (related to bandwidth). These can be combined to derive the mean frequency (M1/M0) and spectral bandwidth.

Usage

```
emgSpectralMoments(x, window_sec = 1, overlap = 0.5, assay_name = NULL)
```

Arguments

`x`	A PhysioExperiment object with EMG data.
`window_sec`	Analysis window in seconds (default: 1.0).
`overlap`	Overlap fraction (default: 0.5).
`assay_name`	Input assay name (default: first assay).

Value

A data.frame with one row per channel per window, containing columns:

channel Integer channel index.

window Integer window number (1-indexed).

m0 Zeroth spectral moment (total power).

m1 First spectral moment (frequency-weighted power).

m2 Second spectral moment (frequency-squared-weighted power).

References

De Luca, C.J. (1984). "Myoelectrical manifestations of localized muscular fatigue in humans." Critical Reviews in Biomedical Engineering, 11(4), 251-279.

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgFatigue()` for median and mean frequency tracking, `emgFatigueIndex()` for summary fatigue metric, `emgEnvelope()` for time-domain amplitude analysis

---

`emgWPLINetwork` *EMG Weighted Phase-Lag Index (wPLI) Network*

Description

Builds a static network from pairwise weighted phase-lag index (wPLI), which is less sensitive to zero-lag coupling artifacts.

Usage

```
emgWPLINetwork(
  x,
  freq_band = NULL,
  channels = NULL,
  nperseg = 256L,
  noverlap = NULL,
  assay_name = NULL,
  aggregate = c("mean", "max", "median"),
  threshold = NULL,
  debiased = FALSE
)
```

Arguments

`x`	A PhysioExperiment object.
`freq_band`	Optional numeric vector `c(low, high)` in Hz.
`channels`	Integer vector of channel indices to include. If NULL, uses all.
`nperseg`	Segment length for Welch estimation (default: 256).
`noverlap`	Overlap length (default: `floor(nperseg / 2)`).
`assay_name`	Input assay name. If NULL, uses default assay.
`aggregate`	Aggregation across frequency bins: "mean", "max", or "median".
`threshold`	Optional threshold for binary adjacency matrix.
`debiased`	Logical; if TRUE, uses debiased wPLI estimator.

Value

A list with:

network Numeric matrix (channel x channel) of wPLI values.

adjacency Logical matrix after thresholding, or NULL.

wpli 3D array (freq x channel x channel).

frequencies Frequency vector (Hz).

channel_names Channel labels used in the network.

See Also

`emgCoherenceNetwork()`, `emgPartialCoherenceNetwork()`

`make_emg` *Create a Basic EMG PhysioExperiment*

Description

Generates a synthetic multi-channel EMG `PhysioExperiment` object with simulated muscle activity. Each channel contains baseline noise with a burst of higher-amplitude activity in the middle 40 percent of the signal (from 30 to 70 percent), mimicking a typical voluntary contraction.

Usage

```
make_emg(n_time = 2000, n_channels = 4, sr = 1000)
```

Arguments

`n_time`	Number of time points (default: 2000).
`n_channels`	Number of EMG channels (default: 4).
`sr`	Sampling rate in Hz (default: 1000).

Value

A `PhysioExperiment` object with a single `"raw"` assay containing simulated EMG data (time x channels matrix), channel metadata in `colData`, and the specified sampling rate.

References

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`make_emg_contraction()` for EMG with a defined contraction window, `make_emg_fatigue()` for EMG with fatigue progression, `emgEnvelope()` for extracting amplitude envelopes

Examples

```
pe <- make_emg()
pe
dim(SummarizedExperiment::assay(pe, "raw"))
```

`make_emg_contraction` *Create EMG PhysioExperiment with Known Contraction*

Description

Generates a synthetic single-channel EMG `PhysioExperiment` with a clearly defined contraction period. The contraction region has substantially higher amplitude than the baseline, making it suitable for testing onset detection algorithms.

Usage

```
make_emg_contraction(
  n_time = 5000,
  sr = 1000,
  contraction_start = 0.3,
  contraction_end = 0.7,
  baseline_sd = 0.01,
  contraction_sd = 0.5
)
```

Arguments

`n_time`	Number of time points (default: 5000).
`sr`	Sampling rate in Hz (default: 1000).
`contraction_start`	Proportion of signal where contraction begins (default: 0.3).
`contraction_end`	Proportion of signal where contraction ends (default: 0.7).
`baseline_sd`	Standard deviation of baseline noise (default: 0.01).
`contraction_sd`	Standard deviation of contraction activity (default: 0.5).

Value

A `PhysioExperiment` object with a single `"raw"` assay containing a one-channel EMG signal with a known contraction window. The contraction region is defined by `contraction_start` and `contraction_end` as proportions of total signal length.

References

Hodges, P.W. & Bui, B.H. (1996). "A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography." Electroencephalography and Clinical Neurophysiology, 101(6), 511-519. doi:10.1016/S0921-884X(96)95190-5

See Also

`emgOnsetDetect()` for detecting the contraction onset, `emgEnvelope()` for extracting the amplitude envelope, `make_emg()` for basic multi-channel EMG, `make_emg_fatigue()` for EMG with fatigue progression

Examples

```
pe <- make_emg_contraction()
pe
onset <- emgOnsetDetect(pe)
onset$onsets
```

`make_emg_fatigue` *Create EMG PhysioExperiment with Fatigue Progression*

Description

Generates a synthetic single-channel EMG `PhysioExperiment` that simulates a fatiguing isometric contraction. The median frequency of the signal decreases progressively over time (from 80 Hz to approximately 44 Hz across 10 segments), mimicking the spectral compression characteristic of muscle fatigue.

Usage

```
make_emg_fatigue(n_time = 10000, sr = 1000)
```

Arguments

`n_time`	Number of time points (default: 10000).
`sr`	Sampling rate in Hz (default: 1000).

Value

A `PhysioExperiment` object with a single `"raw"` assay containing a one-channel EMG signal exhibiting progressive median frequency decrease across 10 equal-length segments.

References

De Luca, C.J. (1997). "The use of surface electromyography in biomechanics." Journal of Applied Biomechanics, 13(2), 135-163. doi:10.1123/jab.13.2.135

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`emgFatigue()` for tracking median frequency over time, `emgFatigueIndex()` for computing a summary fatigue metric, `emgSpectralMoments()` for spectral moment analysis, `make_emg()` for basic multi-channel EMG

Examples

```
pe <- make_emg_fatigue()
pe
fatigue <- emgFatigue(pe)
head(fatigue)
```

`muscleSynergy` *Muscle Synergy Decomposition*

Description

Decomposes multi-channel EMG into muscle synergies using matrix factorization.

Usage

```
muscleSynergy(
  x,
  n_synergies,
  method = c("nmf", "pca", "ica"),
  max_iter = 200L,
  tol = 1e-04,
  assay_name = NULL
)
```

Arguments

`x`	A PhysioExperiment object with multi-channel EMG.
`n_synergies`	Number of synergies to extract.
`method`	Decomposition method: "nmf" (non-negative matrix factorization), "pca" (principal component analysis), or "ica" (independent component analysis).
`max_iter`	Maximum iterations for NMF (default: 200).
`tol`	Convergence tolerance for NMF (default: 1e-4).
`assay_name`	Input assay name (default: first assay).

Value

A list with:

- `W`: Synergy weight matrix (n_synergies x channels)
- `H`: Activation pattern matrix (time x n_synergies)
- `vaf`: Variance accounted for (0-1)
- `method`: Method used
- `original_data`: Original data matrix for reconstruction

References

De Luca, C.J. (1997). "The use of surface electromyography in biomechanics." Journal of Applied Biomechanics, 13(2), 135-163. doi:10.1123/jab.13.2.135

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`synergyReconstruct()` for reconstructing data from synergies, `synergyCompare()` for comparing synergy solutions, `emgEnvelope()` for amplitude envelope extraction

---

`synergyCompare` *Compare Two Synergy Results*

Description

Computes pairwise correlation between synergy weight vectors from two decompositions. Uses best-match pairing.

Usage

```
synergyCompare(result1, result2)
```

Arguments

`result1`	First result from `muscleSynergy`.
`result2`	Second result from `muscleSynergy`.

Value

A data.frame with columns: synergy1, synergy2, correlation.

References

De Luca, C.J. (1997). "The use of surface electromyography in biomechanics." Journal of Applied Biomechanics, 13(2), 135-163. doi:10.1123/jab.13.2.135

Merletti, R. & Parker, P.A. (2004). "Electromyography: Physiology, Engineering, and Non-Invasive Applications." Wiley-IEEE Press. doi:10.1002/0471678384

See Also

`muscleSynergy()` for computing synergy decompositions, `synergyReconstruct()` for reconstructing data from synergies

---

`synergyReconstruct` *Reconstruct Data from Synergies*

Description

Reconstructs EMG data using a subset of synergies.

Usage

```
synergyReconstruct(synergy_result, n_synergies)
```

Arguments

`synergy_result`	Result from `muscleSynergy`.
`n_synergies`	Number of synergies to use for reconstruction.

Value

A list with:

- `reconstructed`: Reconstructed data matrix
- `vaf`: VAF of the reconstruction

References

De Luca, C.J. (1997). "The use of surface electromyography in biomechanics." Journal of Applied Biomechanics, 13(2), 135-163. doi:10.1123/jab.13.2.135

See Also

`muscleSynergy()` for computing the initial decomposition, `synergyCompare()` for comparing synergy solutions

Index