

Package: PhysioHDEMG (via r-universe)

October 3, 2026

Title High-Density Surface EMG Motor-Unit Decomposition for the Physio Ecosystem

Version 0.2.1

Description Decomposition of high-density surface electromyography (HD-sEMG) electrode-grid recordings into constituent motor-unit discharge patterns by convolutive blind source separation (extension, whitening and a fixed-point contrast maximisation with convolution-kernel-compensation refinement), following Holobar & Zazula (2007) and Negro et al. (2016). Provides motor-unit pulse trains, firing statistics (discharge rate, recruitment / derecruitment, pulse-to-noise ratio and silhouette quality), spatial motor-unit action-potential maps over the electrode grid, and a ground-truth HD-sEMG simulator for validation.

Depends R (>= 4.2)

Imports stats, utils

Suggests PhysioCore, S4Vectors, SummarizedExperiment, rhdf5, knitr, rmarkdown, testthat (>= 3.2.0)

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

URL <https://github.com/x-biosignal/PhysioHDEMG>,
<https://x-biosignal.r-universe.dev/PhysioHDEMG>,
<https://x-biosignal.github.io/PhysioHDEMG/>

BugReports <https://github.com/x-biosignal/PhysioHDEMG/issues>

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

Date/Publication 2026-10-03 13:11:44 UTC

RemoteUrl <https://github.com/x-biosignal/PhysioHDEMG>

RemoteRef HEAD

RemoteSha 1cfd2a23864453fd0856dd4796828fc2e90ca124

Contents

hdEMGDecompose	2
hdemgToPhysioExperiment	4
make_hdemg_sim	5
matchMotorUnits	6
muActionPotentialMaps	7
muFiringStats	8
muPulseTrains	9
readHDEMG	10
Index	12

hdEMGDecompose	<i>Decompose HD-sEMG into motor-unit discharge patterns</i>
----------------	---

Description

Decomposes a high-density surface EMG electrode-grid recording into its constituent motor-unit pulse trains by convolutive blind source separation: the channels are extended by delayed copies and whitened, each candidate source is seeded by a fixed-point contrast (kurtosis) maximisation and then refined by convolution-kernel compensation (the separation vector is iteratively set to the average whitened observation at the estimated discharge times), and discharge times are read off by peak detection with a two-class (spike / noise) split. Sources are accepted on a pulse-to-noise ratio and silhouette quality gate, and duplicate units are merged.

Usage

```
hdEMGDecompose(  
  x,  
  n_units = 8,  
  extension_factor = 16,  
  sampling_rate = NULL,  
  assay_name = NULL,  
  refractory_sec = 0.02,  
  pnr_threshold = 26,  
  sil_threshold = 0.9,  
  max_iter = 100,  
  tol = 1e-04,  
  bandpass = NULL,  
  seed = NULL  
)
```

Arguments

<code>x</code>	HD-sEMG data: an <code>n_time</code> x <code>n_channels</code> numeric matrix, an <code>hdemg_sim</code> object from make_hdemg_sim() , or a <code>PhysioExperiment</code> (requires the <code>PhysioCore</code> package).
<code>n_units</code>	Maximum number of separation vectors to attempt (default 8).
<code>extension_factor</code>	Number of delayed copies per channel used to build the extended observations (default 16).
<code>sampling_rate</code>	Sampling rate in Hz; required when <code>x</code> is a bare matrix (taken from the object otherwise).
<code>assay_name</code>	Assay to use when <code>x</code> is a <code>PhysioExperiment</code> (default: the first assay).
<code>refractory_sec</code>	Minimum inter-discharge interval in seconds used for peak separation (default 0.02).
<code>pnr_threshold</code>	Minimum pulse-to-noise ratio in dB for an accepted unit (default 26).
<code>sil_threshold</code>	Minimum silhouette for an accepted unit (default 0.9).
<code>max_iter</code>	Maximum fixed-point iterations per source (default 100).
<code>tol</code>	Convergence tolerance on the separation vector (default 1e-4).
<code>bandpass</code>	Optional length-2 numeric <code>c(low, high)</code> in Hz to band-pass the signal (zero-phase) before decomposition. Recommended for real HD-sEMG (e.g. <code>c(20, 500)</code>) to remove baseline drift and high-frequency noise; <code>NULL</code> (default) leaves the signal unchanged.
<code>seed</code>	Optional integer random seed for reproducibility.

Details

The default quality gate (`pnr_threshold = 26`, `sil_threshold = 0.9`) is tuned for clean data (e.g. [make_hdemg_sim\(\)](#)). Real HD-sEMG motor units typically have lower quality (PNR ~20-23 dB, silhouette ~0.85-0.92); for real recordings, band-pass the signal (`bandpass = c(20, 500)`) and relax the thresholds (`pnr_threshold ~ 20`, `sil_threshold ~ 0.85`). Validated against `openhdemg` on a real 64-channel recording: `PhysioHDEMg` recovered 4 of 5 reference motor units at rate-of-agreement 0.79-0.99 (lag-aligned).

Value

An `hdemg_decomposition` object with `pulse_trains` (list of discharge sample indices), a quality data frame (`unit`, `n_discharges`, `discharge_rate`, `pnr_db`, `sil`), the separation filters, and the retained `signal/grid/sampling_rate` for downstream firing statistics and action-potential maps.

References

Holobar, A. & Zazula, D. (2007). Multichannel blind source separation using convolution kernel compensation. *IEEE TSP* 55(9):4487-4496. Negro, F. et al. (2016). *J Neural Eng* 13(2):026027.

See Also

[make_hdemg_sim\(\)](#), [muFiringStats\(\)](#), [muActionPotentialMaps\(\)](#)

Examples

```
sim <- make_hdemg_sim(n_units = 2, duration_sec = 3, grid = c(4, 4),
                     noise_sd = 0.005, seed = 1)
dec <- hdEMGDecompose(sim, n_units = 4, extension_factor = 10, seed = 1)
dec$quality
```

hdemgToPhysioExperiment

Wrap an HD-sEMG electrode grid in a PhysioExperiment

Description

Builds a `PhysioExperiment` (from `PhysioCore`) holding a single HD-sEMG assay (`n_time` x `n_channels`) at the given sampling rate, recording the electrode grid layout (`rows`, `cols`) in the channel metadata and object metadata so that `hdEMGDecompose()` can recover the grid for action-potential maps.

Usage

```
hdemgToPhysioExperiment(
  signal,
  sampling_rate,
  grid,
  unit = "uV",
  metadata = list()
)
```

Arguments

<code>signal</code>	An <code>n_time</code> x <code>n_channels</code> matrix of HD-sEMG (channels ordered column-major over the grid: row varies fastest).
<code>sampling_rate</code>	Sampling rate in Hz.
<code>grid</code>	Integer <code>c(rows, cols)</code> electrode-grid layout; <code>rows * cols</code> must equal the number of channels.
<code>unit</code>	Signal unit stored in the channel metadata (default "uV").
<code>metadata</code>	Optional named list stored in the object metadata.

Value

A `PhysioExperiment` object.

See Also

[hdEMGDecompose\(\)](#), [make_hdemg_sim\(\)](#)

Examples

```

sim <- make_hdemg_sim(n_units = 2, duration_sec = 2, grid = c(4, 4), seed = 1)
# PhysioCore supplies the shared PhysioExperiment data model.
if (requireNamespace("PhysioCore", quietly = TRUE) &&
    requireNamespace("S4Vectors", quietly = TRUE)) {
  pe <- hdemgToPhysioExperiment(sim$signal, sim$sampling_rate, grid = c(4, 4))
  pe
}

```

make_hdemg_sim	<i>Simulate a high-density surface EMG recording with known motor units</i>
----------------	---

Description

Generates a synthetic HD-sEMG electrode-grid signal as the linear superposition of a known set of motor units, each firing at a target discharge rate with inter-spike-interval jitter and each with its own spatial action-potential map over the grid. The ground-truth discharge times are returned so that a decomposition can be validated (rate of agreement, discharge-rate error).

Usage

```

make_hdemg_sim(
  n_units = 4,
  duration_sec = 10,
  sampling_rate = 2048,
  grid = c(5, 5),
  firing_rates = NULL,
  muap_length = 24,
  cov_isi = 0.1,
  noise_sd = 0.02,
  seed = NULL
)

```

Arguments

n_units	Number of motor units to simulate (default 4).
duration_sec	Recording duration in seconds (default 10).
sampling_rate	Sampling rate in Hz (default 2048).
grid	Integer c(rows, cols) electrode-grid layout (default c(5, 5)); rows * cols channels are generated.
firing_rates	Optional numeric vector of per-unit mean discharge rates (Hz); defaults to an evenly spaced spread in [8, 18] Hz.
muap_length	MUAP template length in samples (default 24).
cov_isi	Coefficient of variation of the inter-spike interval (firing-time jitter, default 0.1).

noise_sd	Standard deviation of additive white noise, in the same units as the unit-peak MUAPs (default 0.02; smaller = higher SNR).
seed	Optional integer random seed for reproducibility.

Value

An `hdemg_sim` object (a list) with:

signal An `n_time` x `n_channels` matrix of simulated HD-sEMG.

sampling_rate Sampling rate in Hz.

grid The `c(rows, cols)` grid layout.

spikes A length-`n_units` list of integer ground-truth discharge sample indices.

discharge_rates The realised per-unit mean discharge rate (Hz).

muap A length-`n_units` list of `muap_length` x `n_channels` ground-truth action-potential templates.

n_units Number of motor units.

References

Farina, D. et al. (2010). Decoding the neural drive to muscles from the surface electromyogram. *J Physiol* 588(11):1969-1975.

See Also

[hdEMGDecompose\(\)](#)

Examples

```
sim <- make_hdemg_sim(n_units = 3, duration_sec = 2, grid = c(4, 4),
                     seed = 1)

dim(sim$signal)
lengths(sim$spikes)
```

matchMotorUnits	<i>Match decomposed motor units to a reference decomposition</i>
-----------------	--

Description

Scores a decomposition against a reference set of motor-unit discharge patterns with a lag-aware rate-of-agreement (RoA): for each reference unit the best-matching decomposed unit is found (allowing a constant latency of up to `max_lag` samples), and the RoA – the fraction of discharges the two share within `tol` samples, $\text{matched} / (\text{n}_a + \text{n}_b - \text{matched})$ – is reported. This scores cross-tool *recovery* ("did we recover the units a reference finds, and which do we miss") rather than bit-identical agreement, so a constant latency between two decompositions does not penalise a correct match.

Usage

```
matchMotorUnits(decomposition, reference, tol = 1L, max_lag = 0L)
```

Arguments

decomposition	An hdemg_decomposition from hdEMGDecompose() .
reference	The reference decomposition: a list of integer discharge- sample vectors (one per reference unit), or a <code>PhysioExperiment</code> carrying them in metadata <code>reference_mupulses</code> (e.g. from readHDEMG()).
tol	Matching tolerance in samples (default 1).
max_lag	Maximum constant latency in samples to search when matching (default 0 = exact alignment; use a few milliseconds' worth for cross-tool scoring).

Value

A data frame (class `mu_match`), one row per reference unit, with columns `ref_mu`, `n_discharges`, `matched_unit`, `roa_lag_aware`, `roa_no_lag`, `lag`, `pnr_db`, `sil`.

References

Holobar A, Minetto MA, Farina D (2014). Accurate identification of motor unit discharge patterns from high-density surface EMG and validation with a novel signal-based performance metric. *J Neural Eng* 11:016008.

See Also

[hdEMGDecompose\(\)](#), [readHDEMG\(\)](#), [muPulseTrains\(\)](#)

Examples

```
# Score a decomposition against the simulator's known discharge times.
sim <- make_hdemg_sim(n_units = 2, duration_sec = 3, grid = c(4, 4),
  noise_sd = 0.005, seed = 1)
dec <- hdEMGDecompose(sim, n_units = 4, extension_factor = 10, seed = 1)
matchMotorUnits(dec, sim$spikes, tol = 2, max_lag = 5)
```

muActionPotentialMaps *Motor-unit action-potential maps*

Description

Spike-triggered average of the raw HD-sEMG for each motor unit, yielding the multi-channel action-potential waveform and its peak-to-peak amplitude across the electrode grid (the spatial "map" used to localise the unit).

Usage

```
muActionPotentialMaps(decomposition, window_ms = 15)
```

Arguments

`decomposition` An `hdemg_decomposition` from `hdEMGDecompose()`.
`window_ms` Half-window in milliseconds averaged around each discharge (default 15).

Value

A length-`n_units` list; each element has `waveform` ($(2*w+1) \times n_channels$ spike-triggered average), `amplitude` (per-channel peak-to-peak), `map` (the amplitude reshaped to the rows $\times$ cols grid, or NULL if the grid is unknown), and `peak_channel` (the strongest channel).

References

Farina, D. et al. (2016). Reflections on the evolution of surface EMG decomposition. *J Electromyogr Kinesiol* 26:39-46.

See Also

[hdEMGDecompose\(\)](#)

Examples

```
sim <- make_hdemg_sim(n_units = 2, duration_sec = 3, grid = c(4, 4),
                     noise_sd = 0.005, seed = 1)
dec <- hdEMGDecompose(sim, n_units = 4, extension_factor = 10, seed = 1)
maps <- muActionPotentialMaps(dec)
maps[[1]]$peak_channel
```

muFiringStats

Motor-unit firing statistics

Description

Per-unit discharge-rate and recruitment statistics: the mean discharge rate, the mean and coefficient of variation of the inter-spike interval, the recruitment (first discharge) and derecruitment (last discharge) times, and the decomposition quality (pulse-to-noise ratio and silhouette).

Usage

```
muFiringStats(decomposition)
```

Arguments

`decomposition` An `hdemg_decomposition` from `hdEMGDecompose()`.

Value

A data frame with one row per motor unit and columns `unit`, `n_discharges`, `discharge_rate` (Hz), `mean_isi_sec`, `cov_isi`, `recruitment_sec`, `derecruitment_sec`, `pnr_db`, `sil`.

References

De Luca, C.J. et al. (1982). Behaviour of human motor units in different muscles during linearly varying contractions. J Physiol 329:113-128.

See Also

[hdEMGDecompose\(\)](#), [muPulseTrains\(\)](#)

Examples

```
sim <- make_hdemg_sim(n_units = 2, duration_sec = 3, grid = c(4, 4),
                     noise_sd = 0.005, seed = 1)
dec <- hdEMGDecompose(sim, n_units = 4, extension_factor = 10, seed = 1)
muFiringStats(dec)
```

muPulseTrains	<i>Motor-unit discharge pulse trains</i>
---------------	--

Description

Returns the discharge instants of every decomposed motor unit in tidy form.

Usage

```
muPulseTrains(decomposition)
```

Arguments

`decomposition` An `hdemg_decomposition` from [hdEMGDecompose\(\)](#).

Value

A data frame with one row per discharge: `unit` (integer), `discharge` (sample index), and `time_sec` (discharge time in seconds).

See Also

[hdEMGDecompose\(\)](#), [muFiringStats\(\)](#)

Examples

```
sim <- make_hdemg_sim(n_units = 2, duration_sec = 3, grid = c(4, 4),
                     noise_sd = 0.005, seed = 1)
dec <- hdEMGDecompose(sim, n_units = 4, extension_factor = 10, seed = 1)
head(muPulseTrains(dec))
```

readHDEMG

*Read a high-density surface EMG recording from an HDF5 file***Description**

Reads an HD-sEMG electrode-grid recording from an HDF5 (.h5) file into a [PhysioExperiment](#), the ingest counterpart of [hdemgToPhysioExperiment\(\)](#) for on-disk grids. The file must contain a signal dataset (the raw grid signal) and a `sampling_rate` scalar; it may also contain a grid dataset (`c(rows, cols)`) and a `reference_mus` group of per-unit integer discharge-sample vectors – a bundled reference decomposition (e.g. exported from the `openhdemg` sample file) – which are stored in the returned object's metadata as `reference_mupulses` for scoring a decomposition with [matchMotorUnits\(\)](#).

Usage

```
readHDEMG(
  path,
  grid = NULL,
  signal_name = "signal",
  rate_name = "sampling_rate",
  unit = "uV"
)
```

Arguments

<code>path</code>	Path to the HDF5 file.
<code>grid</code>	Optional integer <code>c(rows, cols)</code> electrode-grid layout override; taken from the file's grid dataset when present, otherwise <code>c(1, n_channels)</code> .
<code>signal_name, rate_name</code>	Dataset names for the signal and the sampling rate (defaults "signal" and "sampling_rate").
<code>unit</code>	Signal unit stored in the channel metadata (default "uV").

Details

The signal dataset may be stored `n_time x n_channels` or `n_channels x n_time`; the longer axis is taken as time.

Value

A `PhysioExperiment` holding the HD-sEMG grid, with any bundled reference decomposition in metadata `reference_mupulses`.

See Also

[hdEMGDecompose\(\)](#), [matchMotorUnits\(\)](#), [hdemgToPhysioExperiment\(\)](#)

Examples

```
# Round-trip a tiny grid through a temporary HDF5 file (needs rhdf5).
if (requireNamespace("rhdf5", quietly = TRUE) &&
    requireNamespace("PhysioCore", quietly = TRUE)) {
  sim <- make_hdemg_sim(n_units = 2, duration_sec = 2, grid = c(4, 4), seed = 1)
  f <- tempfile(fileext = ".h5")
  rhdf5::h5createFile(f)
  rhdf5::h5write(sim$signal, f, "signal")
  rhdf5::h5write(sim$sampling_rate, f, "sampling_rate")
  rhdf5::h5write(as.integer(sim$grid), f, "grid")
  rhdf5::H5close()
  pe <- readHDEMG(f)
  unlink(f)
  pe
}
```

Index

hdEMGDecompose, [2](#)
hdEMGDecompose(), [4](#), [6–10](#)
hdemgToPhysioExperiment, [4](#)
hdemgToPhysioExperiment(), [10](#)

make_hdemg_sim, [5](#)
make_hdemg_sim(), [3](#), [4](#)
matchMotorUnits, [6](#)
matchMotorUnits(), [10](#)
muActionPotentialMaps, [7](#)
muActionPotentialMaps(), [3](#)
muFiringStats, [8](#)
muFiringStats(), [3](#), [9](#)
muPulseTrains, [9](#)
muPulseTrains(), [7](#), [9](#)

PhysioExperiment, [10](#)

readHDEMG, [10](#)
readHDEMG(), [7](#)