Package: mrpheus (via r-universe)

July 1, 2026

Title Polysomnography Signal Analysis for Sleep Research

Version 0.1.0

Description Raw polysomnography (PSG) signal analysis for sleep and circadian research. Provides EDF/EDF+ ingestion, artefact detection, spectral analysis, sleep event detection (spindles, slow oscillations), automatic AASM sleep staging via a pre-trained LightGBM model (ported from YASA; Vallat & Walker, 2021), and respiratory and cardiac metrics. Exports staged hypnograms directly to hypnor and PSG-derived metrics to syncR. Part of the Circadia Lab R ecosystem.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://mrpheus.circadia-lab.uk, https://github.com/circadia-bio/mrpheus

BugReports https://github.com/circadia-bio/mrpheus/issues

Depends R (>= 4.1.0)

Imports edfReader (>= 1.2.1), gsignal (>= 0.3.5), pracma (>= 2.4.2), lightgbm (>= 4.0.0), zoo (>= 1.8.0), dplyr (>= 1.1.0), tibble (>= 3.2.0), cli (>= 3.6.0), rlang (>= 1.1.0)

Suggests scales, testthat (>= 3.0.0), knitr, rmarkdown, withr

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository https://circadia-bio.r-universe.dev

Date/Publication 2026-06-29 05:21:25 UTC

RemoteUrl https://github.com/circadia-bio/mrpheus

RemoteRef main

RemoteSha 4ca434630cd5feffe167b8e0277c6895fca7ce6b

Contents

`compute_ahi` *Compute Apnea-Hypopnea Index (AHI)*

Description

Calculates the AHI from respiratory event data and total sleep time.

Usage

```
compute_ahi(events, tst_hours)
```

Arguments

`events`	Output of `detect_apneas()`.
`tst_hours`	Numeric. Total sleep time in hours.

Value

Numeric scalar. AHI (events per hour).

compute_band_power *Compute EEG band power per epoch*

Description

Estimates power spectral density (PSD) using Welch's method and integrates power within standard EEG frequency bands (delta, theta, alpha, sigma, beta, gamma) for each epoch and each specified channel. Mirrors the band-power feature extraction used by YASA's staging pipeline.

Usage

```
compute_band_power(
  psg,
  channels = NULL,
 bands = list(delta = c(0.5, 4), theta = c(4, 8), alpha = c(8, 13), sigma = c(13, 16),
    beta = c(16, 30), gamma = c(30, 40)),
  relative = FALSE,
  window_s = 4,
  overlap = 0.5
)
```

Arguments

psg	An mrpheus_psg object from prepare_psg().
channels	Character vector. EEG channel labels. If NULL (default), all non-bad EEG channels are used.
bands	Named list of length-2 numeric vectors defining frequency bands. Default: `list(delta = c(0.5, 4), theta = c(4, 8), alpha = c(8, 13), sigma = c(13, 16), beta = c(16, 30), gamma = c(30, 40))`
relative	Logical. If TRUE, return relative band power (band / total power in 0.5–40 Hz). Default FALSE.
window_s	Numeric. Welch window length in seconds. Default 4.
overlap	Numeric in [0, 1). Fractional overlap between Welch windows. Default 0.5.

Value

A tibble with columns epoch, channel, one column per band, and total_power. Units are $\mu V^2/Hz$ (or dimensionless if relative = TRUE).

Examples

```
## Not run:
bp <- compute_band_power(psg)
bp <- compute_band_power(psg, channels = "EEG Fpz-Cz", relative = TRUE)

## End(Not run)
```

`compute_hrv_sleep` *Compute HRV metrics across sleep stages*

Description

Extracts R-peak positions from the ECG channel, computes standard time- and frequency-domain heart rate variability (HRV) metrics, and stratifies results by sleep stage.

Usage

```
compute_hrv_sleep(
  psg,
  staging = NULL,
  ecg_channel = NULL,
  min_rr_ms = 300,
  max_rr_ms = 2000
)
```

Arguments

`psg`	An `mrpheus_psg` object from `prepare_psg()`.
`staging`	Tibble from `stage_epochs()` or `NULL`. If `NULL`, HRV is computed across all epochs without stage stratification.
`ecg_channel`	Character or `NULL`. ECG channel label. If `NULL` (default), the first non-bad ECG channel is used.
`min_rr_ms`	Numeric. Minimum physiologically plausible RR interval (ms). Default `300` (200 bpm ceiling).
`max_rr_ms`	Numeric. Maximum physiologically plausible RR interval (ms). Default `2000` (30 bpm floor).

Value

A tibble with one row per sleep stage (or one row if `staging` is `NULL`):

stage Character. AASM stage or `"ALL"`.

n_epochs Integer. Number of epochs in this stage.

mean_rr_ms Numeric. Mean RR interval (ms).

sdnn_ms Numeric. SDNN — SD of all NN intervals.

rmssd_ms Numeric. RMSSD — root mean square of successive differences.

lf_power Numeric. LF band power (0.04–0.15 Hz).

hf_power Numeric. HF band power (0.15–0.4 Hz).

lf_hf_ratio Numeric. LF/HF ratio.

compute_odi *Compute Oxygen Desaturation Index (ODI)*

Description

Calculates the ODI (number of >= 3% SpO2 desaturations per hour of sleep) from the SpO2 channel.

Usage

```
compute_odi(psg, spo2_channel, tst_hours, threshold = 3)
```

Arguments

psg	An mrpheus_psg object from prepare_psg().
spo2_channel	Character. SpO2 channel label.
tst_hours	Numeric. Total sleep time in hours.
threshold	Numeric. Desaturation threshold (percentage points). Default 3.

Value

Numeric scalar. ODI (desaturations per hour).

compute_slow_oscillations *Detect slow oscillations*

Description

Detects slow oscillations (SOs) in EEG channels using a zero-crossing approach in the delta band (0.5–2 Hz), following the algorithm described in Mölle et al. (2002) and implemented in YASA (Vallat & Walker, 2021).

Usage

```
compute_slow_oscillations(
  psg,
  channel = NULL,
  stages = NULL,
  freq_so = c(0.5, 2),
  amp_ptp_threshold_uv = c(75, 500),
  duration_pos_s = c(0.1, 1),
  duration_neg_s = c(0.1, 1.5)
)
```

Arguments

psg	An mrpheus_psg object from prepare_psg().
channel	Character. EEG channel label. If NULL (default), the first non-bad EEG channel is used.
stages	Integer vector or NULL. Epoch indices restricted to N2/N3. If NULL, detection runs across all epochs.
freq_so	Numeric vector of length 2. SO frequency band (Hz). Default c(0.5, 2).
amp_ptp_threshold_uv	Numeric vector of length 2. Min and max acceptable peak-to-peak amplitude (µV). Default c(75, 500).
duration_pos_s	Numeric vector of length 2. Min and max positive half- wave duration (s). Default c(0.1, 1.0).
duration_neg_s	Numeric vector of length 2. Min and max negative half- wave duration (s). Default c(0.1, 1.5).

Value

A tibble with one row per detected slow oscillation:

epoch Integer.

start_s Numeric. Onset (s) relative to epoch start.

end_s Numeric. Offset (s) relative to epoch start.

duration_s Numeric.

neg_peak_uv Numeric. Negative peak amplitude (µV).

pos_peak_uv Numeric. Positive peak amplitude (µV).

ptp_uv Numeric. Peak-to-peak amplitude (µV).

channel Character.

References

Mölle, M., Marshall, L., Gais, S., & Born, J. (2002). Grouping of spindle activity during slow oscillations in human non-rapid eye movement sleep. *Journal of Neuroscience*, 22(24), 10941–10947.

Vallat, R., & Walker, M. P. (2021). An open-source, high-performance tool for automated sleep staging. *eLife*, 10, e70092. doi:10.7554/eLife.70092

`compute_spectrogram` *Compute a time-frequency spectrogram*

Description

Generates a short-time Fourier transform (STFT) based spectrogram for a single PSG channel across all epochs. Returns power (μV²/Hz) as a matrix with time (epochs) on rows and frequency on columns.

Usage

```
compute_spectrogram(
  psg,
  channel = NULL,
  freq_range = c(0, 40),
  window_s = 2,
  overlap = 0.5,
  db = TRUE
)
```

Arguments

`psg`	An `mrpheus_psg` object from `prepare_psg()`.
`channel`	Character. A single channel label.
`freq_range`	Numeric vector of length 2. Frequency range to return (Hz). Default `c(0, 40)`.
`window_s`	Numeric. STFT window length in seconds. Default `2`.
`overlap`	Numeric in [0, 1). Fractional window overlap. Default `0.5`.
`db`	Logical. Return power in decibels (`10 * log10(power)`)? Default `TRUE`.

Value

A list of class `mrpheus_spectrogram` with:

power Numeric matrix. Rows = epochs, columns = frequency bins.

freqs Numeric vector. Frequency bin centres (Hz).

epochs Integer vector. Epoch indices.

channel Character. The channel label.

db Logical. Whether power is in dB.

`compute_spindles` *Detect sleep spindles*

Description

Detects sleep spindles in EEG channels using a band-pass / RMS envelope approach, closely following the algorithm in Lacourse et al. (2019) and implemented in YASA (Vallat & Walker, 2021). Spindles are identified as transient bursts of 11–16 Hz activity during NREM sleep.

Usage

```
compute_spindles(
  psg,
  channel = NULL,
  stages = NULL,
  freq_spindle = c(11, 16),
  rms_window_s = 0.3,
  min_duration_s = 0.5,
  max_duration_s = 3,
  threshold_sd = 1.5
)
```

Arguments

`psg`	An `mrpheus_psg` object from `prepare_psg()`.
`channel`	Character. EEG channel label. If `NULL` (default), the first non-bad EEG channel is used.
`stages`	Integer vector or `NULL`. Epoch indices (1-based) restricted to NREM sleep. If `NULL`, spindles are detected across all epochs.
`freq_spindle`	Numeric vector of length 2. Spindle frequency band (Hz). Default `c(11, 16)`.
`rms_window_s`	Numeric. Moving RMS window length in seconds. Default `0.3`.
`min_duration_s`	Numeric. Minimum spindle duration in seconds. Default `0.5`.
`max_duration_s`	Numeric. Maximum spindle duration in seconds. Default `3.0`.
`threshold_sd`	Numeric. RMS threshold as a multiple of the channel SD (within sigma band). Default `1.5`.

Value

A tibble with one row per detected spindle:

epoch Integer. Epoch in which the spindle was detected.

start_s Numeric. Spindle onset relative to epoch start (seconds).

end_s Numeric. Spindle offset relative to epoch start (seconds).

duration_s Numeric. Spindle duration in seconds.

peak_freq_hz Numeric. Peak frequency within the spindle.

rms_uv Numeric. Mean RMS amplitude within the spindle (µV).

channel Character. Channel label.

References

Lacourse, K., Yetton, B., Mednick, S., & Bhatt, D. L. (2019). *Massive online sleep staging: A polysomnography data repository*. Journal of Sleep Research.

Vallat, R., & Walker, M. P. (2021). An open-source, high-performance tool for automated sleep staging. *eLife*, 10, e70092. doi:10.7554/eLife.70092

`detect_apneas` *Detect respiratory events (apneas and hypopneas)*

Description

Detects apneas and hypopneas in respiratory airflow and effort signals, returning event-level metadata and summary indices. Detection follows AASM 2012/2017 criteria: >= 90% signal reduction for >= 10 s (apnea), or >= 30% reduction with >= 3% SpO2 desaturation or arousal for >= 10 s (hypopnea).

Usage

```
detect_apneas(
  psg,
  airflow_channel = NULL,
  spo2_channel = NULL,
  min_duration_s = 10,
  apnea_threshold = 0.9,
  hypopnea_threshold = 0.3,
  desaturation_threshold = 3
)
```

Arguments

`psg`	An `mrpheus_psg` object from `prepare_psg()`.
`airflow_channel`	Character. Airflow channel label (e.g. nasal pressure or thermistor). If `NULL` (default), the first RESP channel is used.
`spo2_channel`	Character or `NULL`. SpO2 channel for hypopnea scoring.
`min_duration_s`	Numeric. Minimum event duration in seconds. Default `10`.
`apnea_threshold`	Numeric. Fractional reduction required for apnea classification. Default `0.90`.
`hypopnea_threshold`	Numeric. Fractional reduction required for hypopnea classification. Default `0.30`.

desaturation_threshold
: Numeric. SpO2 drop (percentage points) required to confirm hypopnea. Default 3.

Value

A list with:

events Tibble. One row per event: epoch, start_s, end_s, duration_s, type ("apnea" / "hypopnea"), desaturation.

summary Tibble. n_apneas, n_hypopneas, n_events.

See Also

compute_ahi(), compute_odi()

---

detect_artifacts *Detect artefact epochs in a PSG recording*

Description

Flags epochs containing likely artefacts based on amplitude thresholds, excessive high-frequency (muscle) power, and movement contamination. Operates on the EEG channels by default. Returns a logical vector (one value per epoch) and optionally attaches per-epoch artefact metrics for inspection.

Usage

```
detect_artifacts(
  psg,
  channels = NULL,
  amp_threshold_uv = 500,
  hf_band = c(40, 100),
  hf_percentile = 0.99,
  verbose = TRUE
)
```

Arguments

psg
: An mrpheus_psg object from prepare_psg().

channels
: Character vector. Channel labels to evaluate. If NULL (default), all non-bad EEG channels are used.

amp_threshold_uv
: Numeric. Peak-to-peak amplitude threshold in µV. Epochs exceeding this in any selected channel are flagged. Default 500.

hf_band
: Numeric vector of length 2. Frequency band (Hz) considered high-frequency / muscle contamination. Default c(40, 100).

hf_percentile	Numeric. Epochs whose HF power exceeds this percentile (across all epochs) are flagged. Default 0.99.
verbose	Logical. Print summary. Default TRUE.

Value

A tibble with one row per epoch and columns:

epoch Integer. Epoch index (1-based).

artefact Logical. TRUE if the epoch is flagged as artefact.

reason Character. Comma-separated reasons for flagging, or NA.

peak_to_peak_uv Numeric. Maximum peak-to-peak amplitude across selected channels.

hf_power_db Numeric. Mean HF band power (dB) across selected channels.

---

export_hypnogram *Export a staged hypnogram for use with hypnor*

Description

Prepares the staging tibble produced by stage_epochs() for downstream use with the hypnor package. Attaches recording metadata as attributes and returns a tibble of class mrpheus_hypnogram, which hypnor::new_hypnogram() accepts directly once hypnor is installed.

Usage

```
export_hypnogram(
  staging,
  epoch_s = 30,
  start_time = NULL,
  participant_id = NULL
)
```

Arguments

staging	A tibble from stage_epochs() with columns epoch, stage, and optional probability columns.
epoch_s	Numeric. Epoch duration in seconds. Must match the epoch_s used in prepare_psg(). Default 30.
start_time	POSIXct or NULL. Recording start time. Used to compute clock-time axes in hypnor plots. If NULL, epochs are indexed from 0.
participant_id	Character or NULL. Optional identifier passed through to hypnor and syncR.

Value

A tibble of class mrpheus_hypnogram with columns epoch, stage, and any probability columns from the staging model. Metadata (epoch_s, start_time, participant_id, source, resolution) are attached as attributes and forwarded to hypnor::new_hypnogram() when hypnor is available.

See Also

stage_epochs()

---

palette_orpheus *Orpheus mosaic palette*

Description

An 8-colour palette extracted from the Roman mosaic *Orpheus Charming the Animals* (3rd century AD, Palermo Archaeological Museum). The mosaic depicts Orpheus — the mythological figure whose name and story this package honours — surrounded by animals, rendered in warm Mediterranean tesserae.

Usage

```
palette_orpheus
```

Format

A named character vector of 8 hex colour codes:

sand `#CDB992` — warm background tessera
vermillion `#B83E2C` — Orpheus's robe; terracotta
olive `#6A7840` — tree and vegetation
umber `#7C5432` — animal fur and earth
bistre `#3C2212` — shadows and outlines
ochre `#B07C3A` — warm amber accent
slate `#6C8284` — birds; dusty teal-grey
ivory `#EAD6AA` — highlights and light tessera

Details

The palette is intentionally earthy and muted, reflecting the natural pigments of Roman mosaic work: sandy limestone backgrounds, terracotta robes, olive vegetation, warm umber fauna, and the distinctive slate-teal of the birds.

Source

Mosaic: *Orpheus Charming the Animals*, Roman, 3rd century AD. Palermo Archaeological Museum. Image via Wikimedia Commons, https://en.wikipedia.org/wiki/Orpheus.

Examples

```
if (requireNamespace("scales", quietly = TRUE)) {
  scales::show_col(palette_orpheus)
}
```

prepare_psg *Prepare a PSG recording for analysis*

Description

Takes an mrpheus_edf object and segments it into standard epochs, performs a channel inventory (classifying signals by type), and flags channels that appear flat or likely bad. This is the standard entry point before any downstream analyses (spectral, event detection, staging).

Usage

```
prepare_psg(
  edf,
  epoch_s = 30,
  eeg_pattern = "EEG|C3|C4|F3|F4|O1|O2|Fpz|Pz",
  eog_pattern = "EOG|ROC|LOC",
  emg_pattern = "EMG|chin|Chin",
  ecg_pattern = "ECG|EKG",
  resp_pattern = "Thor|Abdo|Flow|SpO2|airflow",
  flat_threshold = 1e-06
)
```

Arguments

edf	An mrpheus_edf object from read_edf().
epoch_s	Numeric. Epoch length in seconds. Default 30 (standard AASM epoch).
eeg_pattern	Character. Regex pattern to identify EEG channels. Default "EEG|C3|C4|F3|F4|O1|O2|Fpz|Pz".
eog_pattern	Character. Regex pattern to identify EOG channels. Default "EOG|ROC|LOC".
emg_pattern	Character. Regex pattern to identify EMG channels. Default "EMG|chin|Chin".
ecg_pattern	Character. Regex pattern to identify ECG/EKG channels. Default "ECG|EKG".
resp_pattern	Character. Regex pattern to identify respiratory channels. Default "Thor|Abdo|Flow|SpO2|airflow".
flat_threshold	Numeric. Variance below this value flags a channel as flat/bad. Default 1e-6.

Value

A list of class mrpheus_psg with components:

edf The original mrpheus_edf object.

epochs List. Each element is one epoch (30 s by default), itself a named list of channel vectors.

n_epochs Integer. Total number of complete epochs.

epoch_s Numeric. Epoch duration in seconds.

channel_map Data frame. Channel label, detected type (EEG/EOG/EMG/ECG/RESP/OTHER), sample rate, and bad flag.

Examples

```
## Not run:
rec  <- read_edf("data/psg_001.edf")
psg  <- prepare_psg(rec)
psg$channel_map

## End(Not run)
```

`read_edf` *Read an EDF or EDF+ recording*

Description

Reads a European Data Format (EDF or EDF+) file and returns a structured list containing signal data, channel metadata, and recording header. Wraps `edfReader::readEdfHeader()` and `edfReader::readEdfSignals()` with consistent output formatting for the mrpheus pipeline.

Usage

```
read_edf(path, channels = NULL, only_header = FALSE)
```

Arguments

`path`	Character. Path to an `.edf` or `.edf+` file.
`channels`	Character vector or `NULL`. Channel labels to import. If `NULL` (default), all channels are imported.
`only_header`	Logical. If `TRUE`, return only the header without reading signal data. Useful for quick channel inspection. Default `FALSE`.

Value

A list of class `mrpheus_edf` with components:

header Data frame. Recording metadata (patient info, start time, number of signals, etc.).

signals Named list of numeric vectors, one per channel.

channels Data frame. Channel-level metadata: label, sample rate, physical min/max, digital min/max, transducer type, prefiltering.

duration_s Numeric. Total recording duration in seconds.

path Character. Resolved path to the source file.

Examples

```
## Not run:
rec <- read_edf("data/psg_001.edf")
rec <- read_edf("data/psg_001.edf", channels = c("EEG Fpz-Cz", "EOG horizontal"))
rec$channels

## End(Not run)
```

stage_epochs *Automatic AASM sleep staging*

Description

Stages each 30-second epoch using a pre-trained LightGBM model originally developed for YASA (Vallat & Walker, 2021) and shipped as a cross-language serialised model in `inst/models/yasa_staging.txt`. Features are computed in R to match the Python feature extraction pipeline exactly; bit-exact parity is validated in the package test suite.

Usage

```
stage_epochs(
  psg,
  eeg_channel = NULL,
  eog_channel = NULL,
  emg_channel = NULL,
  artefacts = NULL,
  model_path = system.file("models", "yasa_staging.txt", package = "mrpheus")
)
```

Arguments

psg	An `mrpheus_psg` object from `prepare_psg()`.
eeg_channel	Character. Central EEG channel (e.g. `"EEG Fpz-Cz"`). If `NULL` (default), the first non-bad EEG channel is used.
eog_channel	Character or `NULL`. EOG channel. If `NULL` (default), the first non-bad EOG channel is used (EOG features are omitted if none found).
emg_channel	Character or `NULL`. Chin EMG channel. If `NULL` (default), the first non-bad EMG channel is used (EMG features are omitted if none found).
artefacts	Tibble or `NULL`. Output of `detect_artifacts()`. Artefact epochs are assigned `NA` stage and excluded from the model. If `NULL`, all epochs are staged.
model_path	Character. Path to the serialised LightGBM model. Defaults to the bundled model at `system.file("models/yasa_staging.txt", package = "mrpheus")`.

Details

Stages returned follow standard AASM nomenclature: `W` (wake), `N1`, `N2`, `N3`, `REM`.

Value

A tibble with one row per epoch:

epoch Integer.

stage Character. AASM stage: `W`, `N1`, `N2`, `N3`, `REM`, or `NA` (artefact).

prob_W, prob_N1, prob_N2, prob_N3, prob_REM Numeric. Posterior class probabilities from the LightGBM model.

References

Vallat, R., & Walker, M. P. (2021). An open-source, high-performance tool for automated sleep staging. *eLife*, 10, e70092. doi:10.7554/eLife.70092

See Also

`export_hypnogram()`

Index