Package: hypnoR (via r-universe)

July 1, 2026

Title Hypnogram Handling, Plotting, and Sleep Architecture Metrics

Version 0.1.0

Description Provides a staging-agnostic layer for hypnogram ingestion, sleep architecture metric computation, cycle segmentation, and transition analysis. Accepts full AASM-staged hypnograms (W / N1 / N2 / N3 / REM) from mrpheus and coarser actigraphy-derived hypnograms (W / Sleep / Quiet sleep) from zeitR; all metric functions degrade gracefully depending on staging resolution. Includes publication-ready plotting via theme_circadia(). Designed as the hypnogram layer of the Circadia Lab ecosystem, feeding into syncR::sync().

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports cli (>= 3.6.0)

Suggests dplyr (>= 1.1.0), forcats (>= 1.0.0), ggplot2 (>= 3.4.0), knitr, lubridate (>= 1.9.0), pkgdown, rlang (>= 1.1.0), rmarkdown, testthat (>= 3.0.0), tibble (>= 3.0.0), tidyr (>= 1.3.0)

Config/testthat/edition 3

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

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

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-06-29 05:20:45 UTC

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

RemoteRef main

RemoteSha 3b0e51c9cb495addbe8425c88ac85d06693e175c

Contents

`compute_cycles` *Detect NREM/REM sleep cycles*

Description

Segments a hypnogram into NREM/REM cycles using either Feinberg & Floyd (1979) rules or a simplified AASM-compatible algorithm. Only applicable to full AASM hypnograms; returns an informative error for coarse actigraphy-derived hypnograms.

Usage

```
compute_cycles(
  hypnogram,
  method = "feinberg_floyd",
  epoch_sec = 30L,
  min_rem_epochs = 5L
)
```

Arguments

`hypnogram`	A tibble returned by `read_hypnogram()`.
`method`	`"feinberg_floyd"` (default) or `"aasm"`.
`epoch_sec`	Epoch duration in seconds. Default `30`.
`min_rem_epochs`	Minimum number of consecutive REM epochs to qualify as a REM period. Default `5` (2.5 min at 30-s epochs).

Value

A tibble with one row per detected cycle:

cycle Integer cycle index.

start_epoch,end_epoch First and last epoch of the cycle.

nrem_min,rem_min Duration of NREM and REM portions (minutes).

cycle_min Total cycle duration (minutes).

References

Feinberg, I., & Floyd, T. C. (1979). Systematic trends across the night in human sleep cycles. *Psychophysiology*, 16(3), 283–291.

Examples

```
## Not run:
hyp <- read_hypnogram("night_001.csv")
compute_cycles(hyp)

## End(Not run)
```

`compute_sleep_architecture` *Compute sleep architecture metrics*

Description

Derives standard sleep architecture summary statistics from a staged hypnogram. All metrics are resolution-agnostic: the function computes every metric that is possible given the available staging levels and silently omits metrics that require stages not present in the hypnogram (e.g. REM latency requires a full AASM hypnogram).

Usage

```
compute_sleep_architecture(
  hypnogram,
  epoch_sec = 30L,
  lights_off = NULL,
  lights_on = NULL
)
```

Arguments

`hypnogram`	A tibble returned by `read_hypnogram()`, or any tibble with at minimum columns `epoch` (integer) and `stage` (factor).
`epoch_sec`	Epoch duration in seconds. Default `30`.
`lights_off, lights_on`	Optional `POSIXct` timestamps for lights-off and lights-on. When supplied, TIB (time in bed) and SE (sleep efficiency) are computed relative to the recording period; otherwise the first and last sleep epoch bound the period.

Value

A one-row tibble with columns:

tst_min Total sleep time (minutes).

tib_min Time in bed (minutes). NA if lights_off/lights_on not supplied.

se_pct Sleep efficiency (percent). NA if TIB unavailable.

sol_min Sleep onset latency (minutes).

waso_min Wake after sleep onset (minutes).

rem_lat_min REM latency (minutes). NA for coarse hypnograms.

sws_lat_min SWS (N3) latency (minutes). NA for coarse hypnograms.

pct_n1,pct_n2,pct_n3,pct_rem Stage percentages of TST. NA for stages absent in the hypnogram.

pct_sleep,pct_quiet_sleep Stage percentages for coarse hypnograms. NA for full AASM hypnograms.

staging_resolution "aasm" or "coarse".

Examples

```
## Not run:
hyp <- read_hypnogram("night_001.csv")
compute_sleep_architecture(hyp)

## End(Not run)
```

---

compute_transitions *Compute stage-transition statistics*

Description

Builds a stage-to-stage transition probability matrix and derives fragmentation indices from a staged hypnogram. Works with both full AASM and coarse actigraphy-derived staging.

Usage

```
compute_transitions(hypnogram, normalise = TRUE)
```

Arguments

hypnogram	A tibble returned by read_hypnogram().
normalise	If TRUE (default), each row of the transition count matrix is divided by its row sum to give transition probabilities. If FALSE, raw transition counts are returned.

Value

A list with two elements:

matrix A square tibble (stages × stages) of transition probabilities or counts. Row = *from* stage, column = *to* stage.

fragmentation A one-row tibble with:

n_transitions Total number of stage transitions.

fragmentation_index Proportion of epochs that are followed by a different stage.

wake_transitions Number of transitions to Wake (proxy for arousal burden).

Examples

```
## Not run:
hyp  <- read_hypnogram("night_001.csv")
trans <- compute_transitions(hyp)
trans$matrix
trans$fragmentation

## End(Not run)
```

---

plot_architecture *Plot sleep architecture as a bar chart*

Description

Renders stage durations or percentages as a horizontal bar chart using `ggplot2` and `theme_circadia()`.

Usage

```
plot_architecture(
  architecture,
  metric = "duration",
  colours = NULL,
  title = NULL
)
```

Arguments

`architecture`	A one-row tibble returned by `compute_sleep_architecture()`, or a multi-row tibble for comparing multiple nights (requires a `night` or `id` grouping column).
`metric`	`"duration"` (minutes, default) or `"percentage"` of TST.
`colours`	Named character vector of stage colours. See `plot_hypnogram()` for defaults.
`title`	Optional plot title.

Value

A `ggplot` object.

Examples

```
## Not run:
hyp  <- read_hypnogram("night_001.csv")
arch <- compute_sleep_architecture(hyp)
plot_architecture(arch)

## End(Not run)
```

`plot_hypnogram` *Plot a hypnogram*

Description

Renders a classic stage-over-time hypnogram using `ggplot2` and `theme_circadia()`. Accepts both full AASM and coarse staging; the y-axis order and colour mapping are set automatically from the staging resolution detected in `hypnogram`.

Usage

```
plot_hypnogram(
  hypnogram,
  epoch_sec = 30L,
  colours = NULL,
  show_cycles = FALSE,
  title = NULL
)
```

Arguments

`hypnogram`	A tibble returned by `read_hypnogram()`.
`epoch_sec`	Epoch duration in seconds, used to construct the time axis. Default `30`.
`colours`	Named character vector mapping stage labels to hex colours. Defaults to the Circadia Lab palette via `circadia::domain_colour_for()`, if available, otherwise a built-in fallback.
`show_cycles`	If `TRUE` and cycle information is available (a `cycle` column is present in `hypnogram`), cycle boundaries are overlaid as vertical dashed lines. Default `FALSE`.
`title`	Optional plot title.

Value

A `ggplot` object.

Examples

```
## Not run:
hyp <- read_hypnogram("night_001.csv")
plot_hypnogram(hyp)

## End(Not run)
```

`plot_transition_matrix` *Plot a stage-transition heatmap*

Description

Renders the transition probability (or count) matrix returned by `compute_transitions()` as a heatmap using `ggplot2` and `theme_circadia()`.

Usage

```
plot_transition_matrix(
  transitions,
  label_values = TRUE,
  digits = 2L,
  title = NULL
)
```

Arguments

`transitions`	The `matrix` element of the list returned by `compute_transitions()`.
`label_values`	If `TRUE` (default), cell values are printed inside each tile.
`digits`	Number of decimal places for cell labels. Default `2`.
`title`	Optional plot title.

Value

A `ggplot` object.

Examples

```
## Not run:
hyp   <- read_hypnogram("night_001.csv")
trans <- compute_transitions(hyp)
plot_transition_matrix(trans$matrix)

## End(Not run)
```

`read_hypnogram` *Read a hypnogram from file*

Description

Imports a staged hypnogram from common formats and returns a tidy tibble with one row per epoch. The resulting object is the standard input for all other hypnoR functions.

Usage

```
read_hypnogram(
  path,
  format = "auto",
  epoch_sec = 30L,
  tz = "UTC",
  quiet = FALSE
)
```

Arguments

`path`	Path to the hypnogram file.
`format`	One of `"csv"`, `"edf_annotations"`, `"yasa"`, `"compumedics"`, `"nox"`, or `"auto"` (default). When `"auto"`, the format is inferred from the file extension and header.
`epoch_sec`	Epoch duration in seconds. Default `30`.
`tz`	Time zone string passed to `lubridate::as_datetime()`. Default `"UTC"`.
`quiet`	Suppress informational messages. Default `FALSE`.

Details

hypnoR distinguishes two staging resolutions:

- **Full AASM** (5-state): supplied by `mrpheus`.
- **Coarse** (3-state): supplied by `zeitR`.

All downstream metric functions are resolution-agnostic: they compute what is possible given the available stages and document which metrics require full AASM staging.

Value

A tibble with columns:

epoch Integer epoch index (1-based).

time `POSIXct` timestamp for the start of each epoch.

stage Ordered factor. Level set depends on staging resolution: full AASM (`W`, `N1`, `N2`, `N3`, `REM`) or coarse actigraphy-derived (`W`, `Sleep`, `Quiet sleep`).

source Character — the originating device / scorer label, if available in the file header.

Examples

```
## Not run:
hyp <- read_hypnogram("night_001.csv")
hyp <- read_hypnogram("night_001.edf", format = "edf_annotations")

## End(Not run)
```

Index