Package: boldR (via r-universe)

July 1, 2026

Title fMRI BOLD Signal Analysis for Circadian and Sleep Research

Version 0.1.0

Description Analysis of Blood-Oxygen-Level-Dependent (BOLD) fMRI data for sleep and circadian research. Receives preprocessed derivatives from fMRIPrep and transforms them into analysis-ready outputs: atlas-agnostic parcellation with an extensible atlas schema, ROI-level timeseries extraction, voxelwise signal metrics (tSNR, GLM contrasts), and functional connectivity matrices. Exports parcellated BOLD data directly to syncR for integration with actigraphy, polysomnography, and questionnaire data in a unified participant-indexed database. Part of the Circadia Lab R ecosystem at Northumbria University. fmriprep gets you clean data. boldR gets you useful data.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Depends R (>= 4.1.0)

Imports RNifti (>= 1.5.0), cli (>= 3.6.0), tibble (>= 3.2.0)

Suggests corpcor, ggplot2 (>= 3.4.0), testthat (>= 3.0.0), knitr, rmarkdown, pkgdown, withr

Config/testthat/edition 3

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

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

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-06-29 05:22:50 UTC

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

RemoteRef main

RemoteSha 0f58a3ab07855e1e3a2d2def8d7198607334ca35

Contents

`compute_fc` *Compute functional connectivity from parcellated BOLD timeseries*

Description

Estimates a ROI-to-ROI functional connectivity matrix from a parcellated BOLD timeseries using Pearson correlation (default) or partial correlation. Returns both the full symmetric matrix and a tidy long-format data frame.

Usage

```
compute_fc(parcellated, method = c("pearson", "partial"), fisher_z = TRUE)
```

Arguments

`parcellated`	A `boldr_parcellated` object from `parcellate()`.
`method`	Character. Connectivity metric. One of:
	`"pearson"` (Default) Pearson product-moment correlation.
	`"partial"` Partial correlation (requires `corpcor` in Suggests).
`fisher_z`	Logical. Whether to apply Fisher r-to-z transformation before returning values. Default `TRUE`. Diagonal is set to `NA`.

Value

A list of class `boldr_fc` with components:

matrix Numeric matrix (n_rois × n_rois). Symmetric FC matrix with `NA` on the diagonal.

long Data frame (long format). Columns: `roi_i`, `roi_j`, `roi_name_i`, `roi_name_j`, `fc`.

method Character. Connectivity method used.

fisher_z Logical. Whether Fisher z was applied.

atlas_name Character. Atlas identifier.

Examples

```
## Not run:
parc <- parcellate(cleaned, atlas)
fc   <- compute_fc(parc)
dim(fc$matrix)
head(fc$long)

## End(Not run)
```

`compute_roi_metrics` *Compute summary metrics per ROI*

Description

Returns a tibble of per-ROI summary statistics from the parcellated timeseries: mean signal, standard deviation, and temporal SNR.

Usage

```
compute_roi_metrics(parcellated)
```

Arguments

`parcellated`	A `boldR_parcellated` object from `parcellate()`.

Value

A tibble with columns `roi`, `mean`, `sd`, `tsnr`, `n_voxels`.

Examples

```
## Not run:
parcel  <- parcellate(bold, atlas)
metrics <- compute_roi_metrics(parcel)
metrics

## End(Not run)
```

compute_tsnr *Compute voxelwise temporal SNR*

Description

Calculates temporal signal-to-noise ratio (tSNR) for every in-mask voxel in a cleaned BOLD image. tSNR is defined as the ratio of the mean signal to its temporal standard deviation across the timeseries. It is a standard quality metric for fMRI acquisitions and preprocessing pipelines.

Calculates temporal signal-to-noise ratio (tSNR) for each voxel in the BOLD image: mean signal divided by standard deviation across timepoints. A standard quality metric for assessing fMRI data quality after preprocessing.

Usage

```
compute_tsnr(bold)

compute_tsnr(bold)
```

Arguments

bold	A boldR_bold object from prepare_bold().

Value

A list of class boldr_tsnr with components:

tsnr 3D numeric array (same x × y × z as input). Contains tSNR values for in-mask voxels and NA elsewhere.

mean_tsnr Numeric. Mean tSNR across in-mask voxels.

median_tsnr Numeric. Median tSNR across in-mask voxels.

n_voxels Integer. Number of in-mask voxels.

A list of class boldR_tsnr with components:

tsnr Numeric array. tSNR map with same x/y/z dimensions as BOLD.

mean_tsnr Numeric. Mean tSNR across in-mask voxels.

median_tsnr Numeric. Median tSNR across in-mask voxels.

bold The input boldR_bold object.

Examples

```
## Not run:
sub      <- read_fmriprep("data/derivatives/fmriprep", "01", task = "rest")
cleaned  <- prepare_bold(sub)
tsnr_out <- compute_tsnr(cleaned)
tsnr_out$mean_tsnr
```

```
## End(Not run)
## Not run:
fprep <- read_fmriprep("data/derivatives/fmriprep", subject = "01")
bold  <- prepare_bold(fprep, tr = 2, drop_volumes = 4)
tsnr  <- compute_tsnr(bold)
tsnr$mean_tsnr

## End(Not run)
```

`export_bold` *Export parcellated BOLD data for syncR*

Description

Packages the output of `parcellate()` into a `boldR_export` object suitable for ingestion by `syncR::sync()`. The export includes the ROI timeseries matrix, functional connectivity matrix (if computed), ROI metrics, and participant metadata.

Usage

```
export_bold(parcellated, fc = NULL, roi_metrics = NULL, participant_id = NULL)
```

Arguments

`parcellated`	A `boldR_parcellated` object from `parcellate()`.
`fc`	A `boldR_fc` object from `compute_fc()`, or `NULL` (default).
`roi_metrics`	A tibble from `compute_roi_metrics()`, or `NULL` (default). If `NULL`, metrics are computed automatically.
`participant_id`	Character or `NULL`. Override the subject label for use in `syncR`. If `NULL`, taken from the `boldR_bold` object.

Value

A list of class `boldR_export` ready for `syncR::sync()`.

Examples

```
## Not run:
parcel <- parcellate(bold, atlas)
fc     <- compute_fc(parcel)
exp    <- export_bold(parcel, fc = fc)

# Pass to syncR
syncR::sync(..., bold = exp)

## End(Not run)
```

list_atlases *List available built-in atlases*

Description

Returns a data frame describing the atlases bundled with boldR. Custom atlases can be loaded via load_atlas().

Usage

```
list_atlases()
```

Value

A tibble with columns name, n_rois, space, and description.

Examples

```
list_atlases()
```

load_atlas *Load a parcellation atlas*

Description

Loads a boldR_atlas object either from the bundled atlas library or from a custom NIfTI label map. The boldR_atlas schema is the extension point for user-defined parcellations.

Usage

```
load_atlas(atlas, labels = NULL, space = NULL)
```

Arguments

atlas	Character. Name of a built-in atlas (see list_atlases()) or a path to a custom NIfTI label image.
labels	Character vector or NULL. ROI labels, one per integer value in the atlas image (excluding 0). If NULL and atlas is a built-in, labels are loaded automatically. Required for custom NIfTI inputs.
space	Character or NULL. Template space of the atlas. Inferred for built-ins; should be specified for custom inputs.

Value

A list of class `boldR_atlas` with components:

name Character. Atlas name or file path.

nifti_path Character. Path to the atlas NIfTI label image.

labels Character vector. ROI label for each integer index.

indices Integer vector. Non-zero integer values in the atlas.

n_rois Integer. Number of ROIs.

space Character. Template space.

Examples

```
## Not run:
atlas <- load_atlas("schaefer100_7n")
atlas

# Custom atlas
atlas <- load_atlas(
  atlas  = "my_parcellation.nii.gz",
  labels = paste0("ROI_", 1:80),
  space  = "MNI152NLin2009cAsym"
)

## End(Not run)
```

---

palette_bold *boldR colour palette*

Description

A named character vector of the eight colours used throughout the boldR package and its pkgdown site.

Usage

```
palette_bold
```

Format

A named character vector of length 8.

`parcellate` *Extract ROI-level timeseries from a BOLD image*

Description

Applies a parcellation atlas to a preprocessed BOLD image, returning mean timeseries per ROI. Works with any `boldR_atlas` object – built-in or custom – as long as the atlas is in the same template space as the BOLD.

Usage

```
parcellate(bold, atlas, summary_fn = mean, min_voxels = 10L)
```

Arguments

`bold`	A `boldR_bold` object from `prepare_bold()`.
`atlas`	A `boldR_atlas` object from `load_atlas()`.
`summary_fn`	Function. Applied across voxels within each ROI at each timepoint. Default `mean`. Common alternatives: `median`.
`min_voxels`	Integer. Minimum number of in-mask voxels an ROI must contain to be included in the output. ROIs below this threshold are returned with `NA` timeseries and a warning. Default `10L`.

Value

A list of class `boldR_parcellated` with components:

timeseries Numeric matrix, timepoints x ROIs. Column names are ROI labels from the atlas.

roi_voxel_counts Integer vector. Number of in-mask voxels per ROI.

atlas The `boldR_atlas` object used.

bold The `boldR_bold` object used.

n_rois Integer. Number of ROIs in output.

n_timepoints Integer. Number of timepoints.

Examples

```
## Not run:
fprep  <- read_fmriprep("data/derivatives/fmriprep", subject = "01")
bold   <- prepare_bold(fprep, tr = 2, drop_volumes = 4)
atlas  <- load_atlas("schaefer100_7n")
parcel <- parcellate(bold, atlas)
dim(parcel$timeseries)  # timepoints x 100

## End(Not run)
```

prepare_bold *Validate and prepare a fMRIPrep object for analysis*

Description

Checks file existence, reads NIfTI headers to confirm dimensionality and TR, and returns an annotated boldR_bold object ready for downstream analysis functions.

Usage

```
prepare_bold(fmriprep, tr = NULL, drop_volumes = 0L)
```

Arguments

fmriprep	A boldR_fmriprep object from read_fmriprep().
tr	Numeric or NULL. Repetition time in seconds. If NULL (default), extracted from the BOLD NIfTI header.
drop_volumes	Integer. Number of initial volumes to discard (dummy scans). Default 0.

Value

A list of class boldR_bold with components:

fmriprep The input boldR_fmriprep object.

dims Integer vector. BOLD array dimensions [x, y, z, t].

tr Numeric. Repetition time in seconds.

n_voxels Integer. Total in-mask voxels.

n_timepoints Integer. Number of volumes after dummy-scan removal.

drop_volumes Integer. Volumes dropped.

Examples

```
## Not run:
fprep <- read_fmriprep("data/derivatives/fmriprep", subject = "01")
bold  <- prepare_bold(fprep, tr = 2, drop_volumes = 4)
bold

## End(Not run)
```

`read_fmriprep` *Read fMRIPrep BIDS derivatives for a participant*

Description

Reads preprocessed fMRI outputs from an fMRIPrep BIDS derivatives directory for a single participant and session, returning a structured `boldR_fmriprep` object containing paths to the BOLD, T1w, T2w, and segmentation files.

Usage

```
read_fmriprep(
  derivatives_dir,
  subject,
  session = NULL,
  task = NULL,
  space = "MNI152NLin2009cAsym",
  resolution = NULL
)
```

Arguments

`derivatives_dir`	Character. Path to the fMRIPrep derivatives root (e.g. `"sub-01/func/"`... or the top-level `derivatives/fmriprep/`).
`subject`	Character. Subject label without the `sub-` prefix (e.g. `"01"`).
`session`	Character or `NULL`. Session label without the `ses-` prefix. If `NULL` (default), assumes a single-session dataset.
`task`	Character or `NULL`. Task label (e.g. `"rest"`). If `NULL`, the first task found is used.
`space`	Character. Template space for the BOLD and mask files. Default `"MNI152NLin2009cAsym"`.
`resolution`	Character or `NULL`. Resolution label (e.g. `"2"`). If `NULL`, resolution is not filtered.

Value

A list of class `boldR_fmriprep` with components:

bold Character. Path to the BOLD NIfTI file.

bold_mask Character. Path to the brain mask NIfTI file.

t1w Character. Path to the T1w NIfTI file in template space.

t2w Character or `NA`. Path to the T2w NIfTI file if available.

dseg Character. Path to the discrete segmentation NIfTI file.

confounds Character. Path to the confounds TSV file.

subject Character. Subject label.

session Character or `NA`. Session label.

task Character. Task label.

space Character. Template space.

derivatives_dir Character. Resolved path to the derivatives root.

Examples

```
## Not run:
fprep <- read_fmriprep(
  derivatives_dir = "data/derivatives/fmriprep",
  subject         = "01",
  task            = "rest"
)
fprep

## End(Not run)
```

`register_atlas` *Register a custom brain atlas*

Description

Validates and registers a NIfTI label map and optional metadata table as a `boldR_atlas` object. The schema is intentionally minimal: any parcellation that provides a 3D integer label image and a label-to-name mapping can be registered and passed to `parcellate()`. Bundled atlas helpers (Schaefer, AAL, Glasser) use this same interface.

Usage

```
register_atlas(
  labels_img,
  metadata,
  name = "custom",
  space = "MNI152NLin2009cAsym"
)
```

Arguments

`labels_img`	A `niftiImage` (3D integer) or a file path to a NIfTI label map. Voxels containing `0` are treated as background.
`metadata`	A data frame with at minimum two columns: `label` Integer. The integer value in `labels_img`. `name` Character. Human-readable ROI name. Additional columns (hemisphere, network, x/y/z centroid, colour, etc.) are stored and carried through to output.
`name`	Character. A short identifier for this atlas (e.g. `"Schaefer200"`, `"AAL3"`, `"custom"`).
`space`	Character. The standard space this atlas is defined in. Default `"MNI152NLin2009cAsym"`. Must match the space used in `read_fmriprep()` / `parcellate()`.

Value

A list of class `boldR_atlas` with components:

labels_img `niftiImage`. The 3D integer label map.

metadata Data frame. ROI metadata.

name Character. Atlas identifier.

space Character. Standard space.

n_rois Integer. Number of unique non-zero labels.

Examples

```
## Not run:
atlas <- register_atlas(
  labels_img = "atlases/Schaefer2018_200Parcels_7Networks_MNI.nii.gz",
  metadata   = read.csv("atlases/Schaefer2018_200Parcels_labels.csv"),
  name       = "Schaefer200",
  space      = "MNI152NLin2009cAsym"
)
atlas

## End(Not run)
```

Index