

Package ‘glscalibrator’

November 5, 2025

Type Package

Title Automated Calibration and Analysis of GLS (Global Location Sensor) Data

Version 0.1.0

Author Fabio Favoretto [aut, cre], Cecilia Soldatini [aut]

Maintainer Fabio Favoretto <favoretto.fabio@gmail.com>

Description Provides a fully automated workflow for calibrating and analyzing light-level geolocation (GLS) data from seabirds and other wildlife. The package auto-discovers birds from directory structures, automatically detects calibration periods from the first days of deployment, processes multiple individuals in batch mode, and generates standardized outputs including position estimates, diagnostic plots, and quality control metrics. Built on proven methods from SGAT, GeoLight, and TwGeos packages.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports GeoLight, magrittr, maps, dplyr, lubridate, stringr, utils, grDevices, graphics, stats, TwGeos

Suggests knitr, SGAT, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/fabbiologia/glscalibrator>

BugReports <https://github.com/fabbiologia/glscalibrator/issues>

NeedsCompilation no

Contents

| | |
|-----------------------------------|---|
| auto_detect_calibration | 2 |
| calibrate_gls_batch | 3 |
| convert_to_glsmerge | 5 |

| | |
|--|----|
| detect_twilights | 6 |
| filter_twilights | 6 |
| glscalibrator_example_metadata | 7 |
| gls_example | 8 |
| list_gls_examples | 9 |
| plot_calibration | 10 |
| plot_track | 10 |
| process_single_bird | 11 |
| read_lux_file | 11 |

auto_detect_calibration

Automatically Detect Calibration Period

Description

Automatically detects a suitable calibration period from the beginning of the light data. This assumes the bird was at a known location (typically the colony) at the start of deployment.

Usage

```
auto_detect_calibration(
  light_data,
  colony_lat,
  colony_lon,
  threshold = 2,
  min_twilights = 2
)
```

Arguments

| | |
|---------------|---|
| light_data | A data.frame with columns Date and Light |
| colony_lat | Numeric latitude of known calibration location |
| colony_lon | Numeric longitude of known calibration location |
| threshold | Light threshold for twilight detection (default: 2) |
| min_twilights | Minimum number of twilights required (default: 2) |

Details

The function tries calibration periods of different lengths (2, 3, 1, 4, 5 days) and returns the first period that yields sufficient twilights with both sunrise and sunset events.

Value

A list with:

| | |
|---------------|-------------------------------------|
| start | POSIXct start of calibration period |
| end | POSIXct end of calibration period |
| twilights | Number of twilights detected |
| duration_days | Duration in days |

Examples

```
# Auto-detect calibration from example data (requires SGAT)
example_file <- gls_example("W086")
light_data <- read_lux_file(example_file)
calib <- auto_detect_calibration(light_data,
                                colony_lat = 27.85,
                                colony_lon = -115.17)

print(calib)
```

| | |
|---------------------|--|
| calibrate_gls_batch | <i>Batch Calibration of Multiple GLS Devices</i> |
|---------------------|--|

Description

Main function for automated batch processing of GLS data. Auto-discovers all birds in a directory, detects calibration periods, performs TwGeos gamma calibration, and generates standardized outputs.

Usage

```
calibrate_gls_batch(
  data_dir,
  output_dir,
  colony_lat,
  colony_lon,
  light_threshold = 2,
  exclude_equinoxes = NULL,
  min_positions = 10,
  create_plots = TRUE,
  verbose = TRUE
)
```

Arguments

| | |
|--------------------------------|---|
| <code>data_dir</code> | Character path to directory containing .lux files |
| <code>output_dir</code> | Character path for output files |
| <code>colony_lat</code> | Numeric latitude of colony/capture location |
| <code>colony_lon</code> | Numeric longitude of colony/capture location |
| <code>light_threshold</code> | Numeric light threshold in lux (default: 2) |
| <code>exclude_equinoxes</code> | List of date ranges to exclude (optional) |
| <code>min_positions</code> | Minimum number of valid positions required (default: 10) |
| <code>create_plots</code> | Logical, whether to create diagnostic plots (default: TRUE) |
| <code>verbose</code> | Logical, print progress messages (default: TRUE) |

Details

This function implements a complete automated workflow:

1. Auto-discovers .lux files in the data directory
2. For each bird:
 - Reads light data
 - Auto-detects calibration period
 - Detects and filters twilights
 - Performs TwGeos gamma calibration
 - Calculates positions using threshold method
 - Generates diagnostic plots
3. Combines results into standardized formats
4. Creates summary statistics and quality control metrics

Value

A list containing:

| | |
|-----------------------------|---|
| <code>summary</code> | data.frame with calibration summary for all birds |
| <code>results</code> | List of position estimates for each bird |
| <code>processing_log</code> | Detailed processing log |

Examples

```
## Not run:
# Basic usage
results <- calibrate_gls_batch(
  data_dir = "data/raw/birds",
  output_dir = "data/processed/calibration",
  colony_lat = 27.85178,
  colony_lon = -115.17390
```

```
)

# With equinox exclusions
equinoxes <- list(
  c("2024-08-24", "2024-10-23"),
  c("2024-02-19", "2024-04-19")
)
results <- calibrate_gls_batch(
  data_dir = "data/raw/birds",
  output_dir = "data/processed/calibration",
  colony_lat = 27.85,
  colony_lon = -115.17,
  exclude_equinoxes = equinoxes
)

## End(Not run)
```

| | |
|---------------------|-----------------------------------|
| convert_to_glsmerge | <i>Convert to GLSmerge Format</i> |
|---------------------|-----------------------------------|

Description

Converts calibrated position data to the standard GLSmerge format used by many researchers and analysis tools.

Usage

```
convert_to_glsmerge(results, bird_id, zenith)
```

Arguments

| | |
|---------|---|
| results | data.frame of position estimates |
| bird_id | Character ID of the bird |
| zenith | Numeric zenith angle used for calibration |

Value

data.frame in GLSmerge format

| | |
|------------------|--|
| detect_twilights | <i>Detect Twilight Times from Light Data</i> |
|------------------|--|

Description

Detects sunrise and sunset times from light intensity data using a threshold-crossing method. This is a proven, simple approach that identifies transitions between day and night.

Usage

```
detect_twilights(light_data, threshold = 2)
```

Arguments

| | |
|------------|---|
| light_data | A data.frame with columns Date (POSIXct) and Light (numeric) |
| threshold | Numeric light threshold in lux for day/night distinction (default: 2) |

Value

A data.frame with columns:

| | |
|----------|---|
| Twilight | POSIXct datetime of twilight event |
| Rise | Logical, TRUE for sunrise, FALSE for sunset |

Examples

```
# Detect twilights from example data
example_file <- gls_example("W086")
light_data <- read_lux_file(example_file)
twilights <- detect_twilights(light_data, threshold = 2)
head(twilights)
```

| | |
|------------------|---------------------------------------|
| filter_twilights | <i>Filter and Clean Twilight Data</i> |
|------------------|---------------------------------------|

Description

Applies quality filters to twilight data to remove spurious detections caused by shading, logger malfunction, or other artifacts.

Usage

```
filter_twilights(twilights, light_data = NULL, threshold = 2, strict = TRUE)
```

Arguments

| | |
|-------------------------|---|
| <code>twilights</code> | A data.frame with columns <code>Twilight</code> and <code>Rise</code> |
| <code>light_data</code> | The original light data (optional, for quality checks) |
| <code>threshold</code> | Light threshold used for twilight detection |
| <code>strict</code> | Logical, if TRUE applies stricter filtering (default: TRUE) |

Details

Filters applied:

- Remove twilights too close together (< 1-2 hours)
- Remove twilights with unusual intervals (far from 12 or 24 hours)
- Optionally check light quality around twilight (if `light_data` provided)

Value

A filtered data.frame of twilights

Examples

```
# Filter twilights from example data
example_file <- gls_example("W086")
light_data <- read_lux_file(example_file)
twilights <- detect_twilights(light_data, threshold = 2)
twilights_clean <- filter_twilights(twilights, light_data, threshold = 2)
nrow(twilights_clean)
```

`glscalibrator_example_metadata`

Metadata for the bundled GLS example datasets

Description

The package ships with three light-level geolocation (.lux) files that are used throughout the documentation, vignettes, and tests. This metadata table records their origin and recommended use so that analysts can reference the contents programmatically.

Named character vector of the filenames stored in `inst/extdata/`, keyed by the identifiers recognised by `gls_example()`.

Usage

`glscalibrator_example_metadata`

`glscalibrator_example_files`

Format

A data frame with 3 rows and 7 variables:

name Short identifier used by `gls_example()`
file Filename stored under `inst/extdata/`
type "real" or "synthetic" dataset
description Summary of the dataset contents
duration_days Approximate deployment duration represented
size_kb Approximate file size in kilobytes
notes Additional guidance for analysis and demonstrations

A data frame (tibble) with metadata for each example dataset.

Named character vector. Use with `system.file("extdata", ...)`.

Details

All files are plain-text .lux exports that can be read directly with `read_lux_file()`. Real datasets were collected from tropical seabirds breeding near 27.85°N, 115.17°W and are approved for demonstration and teaching purposes. The synthetic dataset contains idealised sunrise/sunset curves for rapid testing.

See Also

`gls_example()`, `list_gls_examples()`

| | |
|--------------------------|---------------------------------|
| <code>gls_example</code> | <i>Get Path to Example Data</i> |
|--------------------------|---------------------------------|

Description

Helper function to get the path to example .lux files included with the package. Three example files are available and their metadata is exposed via `glscalibrator_example_metadata`.

Usage

```
gls_example(which = "all")
```

Arguments

which Character string specifying which example file:

- "W086" - See metadata for details
- "W592" - See metadata for details
- "synthetic" - See metadata for details
- "all" - Returns paths to all example files (default)

Value

Character vector of file path(s) to example data

Examples

```
# Inspect available example datasets
list_gls_examples()

# Read the bundled W086 seabird deployment
light_data <- read_lux_file(gls_example("W086"))

# Run calibration on the synthetic dataset (quick demo)
synt_path <- gls_example("synthetic")
synthetic_data <- read_lux_file(synt_path)
twl <- detect_twilights(synthetic_data, threshold = 2)
```

| | |
|-------------------|--|
| list_gls_examples | <i>List Available Example Datasets</i> |
|-------------------|--|

Description

Shows information about example datasets included with the package.

Usage

```
list_gls_examples()
```

Value

A data.frame with columns: name, file, type, description, duration_days, size_kb, notes

Examples

```
list_gls_examples()
```

| | |
|------------------|-------------------------------------|
| plot_calibration | <i>Plot Calibration Diagnostics</i> |
|------------------|-------------------------------------|

Description

Creates diagnostic plots showing light data and detected twilights during the calibration period.

Usage

```
plot_calibration(light_data, twilights, threshold, bird_id, output_dir)
```

Arguments

| | |
|------------|---|
| light_data | data.frame with Date and Light columns |
| twilights | data.frame with Twilight and Rise columns |
| threshold | Numeric light threshold |
| bird_id | Character ID for plot title |
| output_dir | Directory to save plot |

Value

NULL (creates PNG file)

| | |
|------------|------------------------|
| plot_track | <i>Plot Bird Track</i> |
|------------|------------------------|

Description

Creates a map showing the estimated movement track of a bird

Usage

```
plot_track(  
  results,  
  colony_lat,  
  colony_lon,  
  bird_id,  
  output_dir,  
  hemisphere = ""  
)
```

Arguments

| | |
|------------|--|
| results | data.frame with Latitude and Longitude columns |
| colony_lat | Numeric latitude of colony |
| colony_lon | Numeric longitude of colony |
| bird_id | Character ID for plot title |
| output_dir | Directory to save plot |
| hemisphere | Character hemisphere check result |

Value

NULL (creates PNG file)

| | |
|---------------------|------------------------------|
| process_single_bird | <i>Process a Single Bird</i> |
|---------------------|------------------------------|

Description

Internal function to process a single GLS device

Usage

```
process_single_bird(
  lux_file,
  bird_id,
  colony_lat,
  colony_lon,
  light_threshold,
  exclude_equinoxes,
  output_dir,
  fig_dir,
  verbose
)
```

| | |
|---------------|--|
| read_lux_file | <i>Read Light Data from .lux Files</i> |
|---------------|--|

Description

Reads and parses light intensity data from .lux files generated by geolocation loggers. Handles various formats and automatically detects the start of data.

Usage

```
read_lux_file(file_path)
```

Arguments

`file_path` Character string specifying the path to the .lux file

Value

A data.frame with columns:

| | |
|-------|--------------------------------|
| Date | POSIXct datetime in UTC |
| Light | Numeric light intensity in lux |

Examples

```
# Read example data included with package
example_file <- gls_example("W086")
light_data <- read_lux_file(example_file)
head(light_data)
```