The Journal of Open Source Software

glscalibrator: An R Package for Automated Calibration and Analysis of Light-Level Geolocation Data

Fabio Favoretto [1], **Cecilia Soldatini** [2], **Yuri V. Albores-Barajas** [3], and **Gemma Abaunza** [2]

1 Scripps Institution of Oceanography, University of California San Diego, CA United States **2** Centro de Investigación Científica y de Educación Superior de Ensenada (CICIESE), La Paz, Baja California Sur, Mexico **3** Departamento de Ciencias Marinas y Costeras, Universidad Autónoma de Baja California Sur, 23080, La Paz, Baja California Sur, Mexico

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Global Location Sensors (GLS) are miniature archival light-level loggers widely used to track long-distance movements of seabirds, migratory birds, and other wildlife. These devices record ambient light levels over time, which can be analyzed to estimate geographic positions based on day length (latitude) and timing of sunrise/sunset (longitude). However, processing GLS data has traditionally been a manual, time-consuming process requiring expertise in multiple R packages and custom scripting for each study. `glscalibrator` addresses this challenge by providing a fully automated workflow that processes entire datasets with a single command, from raw light data to calibrated position estimates and diagnostic visualizations.

Statement of Need

Light-level geolocation is a crucial tool in movement ecology, enabling researchers to track animals over months to years without the size, cost, and battery limitations of GPS devices (Lisovski et al., 2020). However, the analytical workflow presents significant barriers to entry and reproducibility. Existing R packages such as `SGAT` (Wotherspoon et al., 2016), `GeoLight` (Lisovski & Hahn, 2012), and `TwGeos` (Lisovski, 2019) provide excellent tools for individual components of the analysis, but researchers must:

1. **Manually specify calibration periods** - determining when the device was at a known location requires examining data and ecological knowledge
2. **Write custom scripts for batch processing** - processing multiple individuals requires loops and custom code
3. **Manually create standardized outputs** - combining results and generating consistent visualizations is left to the researcher
4. **Individually troubleshoot failures** - identifying and resolving issues requires examining each device separately

These requirements create several problems:

- **High barrier to entry** for new researchers and students
- **Time-intensive workflow** - processing 25 birds can take several days
- **Reduced reproducibility** - custom scripts vary between studies and researchers
- **Inconsistent quality control** - manual processes are prone to errors and oversights

`glscalibrator` fills this gap by providing an end-to-end automated solution that:

- **Auto-discovers** all GLS devices from directory structures

- **Automatically detects** calibration periods from the first days of deployment
- **Batch processes** multiple individuals without manual intervention
- **Generates standardized outputs** including position estimates, diagnostic plots, and quality control metrics
- **Implements proven methods** from `TwGeos`, `GeoLight`, and `SGAT` packages

This automation transforms a multi-day manual process into a single-command workflow, making GLS analysis more accessible, reproducible, and efficient.

Features and Functionality

Core Workflow

The `calibrate_gls_batch()` function implements a complete automated workflow:

```
results <- calibrate_gls_batch(
  data_dir = "data/raw/birds",
  output_dir = "data/processed/calibration",
  colony_lat = 27.85178,
  colony_lon = -115.17390
)
```

This single command:

1. **Auto-discovers** all `.lux` files in the data directory
2. For each device:
    - Reads and parses light intensity data
    - Auto-detects calibration period (first 1-5 days)
    - Detects twilight times using threshold-crossing method
    - Filters spurious twilights using temporal and quality criteria
    - Performs TwGeos gamma calibration (Lisovski, 2019)
    - Calculates positions using threshold method (Hill & Braun, 1994)
    - Generates diagnostic plots (calibration and track maps)
3. **Combines results** into standardized formats (GLSmergedata.csv)
4. **Creates quality control metrics** including hemisphere checks and summary statistics

Intelligent Auto-Calibration

A key innovation is automatic detection of calibration periods. The function searches the first 1-5 days of data for stable periods where the device was at the known colony location, automatically identifying sufficient twilight events for calibration. This eliminates the need for manual data inspection while ensuring robust calibration.

Quality Control

The package implements multiple quality control steps:

- **Twilight filtering**: Removes events $<$ 1 hour apart and with unusual intervals
- **Position filtering**: Excludes impossible coordinates and optionally removes equinox periods
- **Hemisphere validation**: Checks that positions fall in expected hemisphere
- **Diagnostic visualizations**: Generates plots showing light curves, twilights, and tracks
- **Processing logs**: Records successes, failures, and error messages for troubleshooting

The Journal of Open Source Software

Modular Design

While the batch function provides full automation, individual functions can be used for custom workflows:

- `read_lux_file()`: Parse .lux files
- `detect_twilights()`: Threshold-crossing twilight detection
- `filter_twilights()`: Quality filtering of twilights
- `auto_detect_calibration()`: Automatic calibration period detection
- `convert_to_glsmerge()`: Standardize output format
- `plot_calibration()` and `plot_track()`: Generate visualizations

Implementation and Performance

`glscalibrator` is implemented in R and builds on established packages:

- **TwGeos**: Gamma calibration and light data processing (Lisovski, 2019)
- **GeoLight**: Position estimation via threshold method (Lisovski & Hahn, 2012)
- **SGAT**: Reference implementations for twilight analysis and manual workflows (Wotherspoon et al., 2016)
- **tidyverse**: Data manipulation and workflow management

The package has been validated on datasets of 25+ seabirds, successfully processing 96% of devices (25/26) with appropriate error handling for the remaining cases. Processing time is ~30-60 seconds per bird on a laptop, making batch processing of large datasets practical.

Use Cases and Impact

`glscalibrator` is designed for:

- **Seabird researchers** tracking albatrosses, petrels, shearwaters, and other pelagic species
- **Migration ecologists** studying migratory birds and bats
- **Marine ecologists** investigating animal-environment interactions
- **Students and early-career researchers** learning GLS analysis
- **Large-scale studies** requiring consistent processing of many individuals

The package has been successfully applied to studies of tropical seabirds in the Eastern Pacific, processing deployment and recovery data from multiple years and species. By automating the workflow, researchers can focus on biological interpretation rather than technical implementation.

Comparison with Existing Tools

Feature	SGAT	GeoLight	TwGeos	**glscalibrator**
Twilight detection				
Gamma calibration		-		
Position estimation			-	
Auto-discover birds	-	-	-	
Auto-detect calibration	-	-	-	
Batch processing	Manual	Manual	Manual	**Automated**
Standardized output	Custom	Custom	Custom	**Built-in**
Diagnostic plots	Custom	Custom	Custom	**Automatic**
Quality control	Manual	Manual	Manual	**Automated**

`glscalibrator` complements rather than replaces existing tools, using them internally while adding automation layers.

The Journal of Open Source Software

Availability and Contributions

`glscalibrator` is open source (MIT license) and available at:

- GitHub: https://github.com/fabbiologia/glscalibrator

Contributions are welcome via GitHub issues and pull requests. The package follows standard R package development practices including semantic versioning, continuous integration, and code review.

Acknowledgments

We thank the developers of SGAT, GeoLight, and TwGeos for creating the foundational tools that make this work possible. We also thank the seabird tracking community for feedback on workflow requirements and testing.

References

Hill, C., & Braun, M. (1994). Determining bird migration routes: A comparison of methods using simple models. *The Auk*, *111*(4), 983–989. https://doi.org/10.2307/4088832

Lisovski, S. (2019). TwGeos: Basic data processing for light-level geolocation archival tags. *R Package Version 0.1*. https://github.com/slisovski/TwGeos

Lisovski, S., Bauer, S., Briedis, M., Davidson, S. C., Dhanjal-Adams, K. L., Hallworth, M. T., Karagicheva, J., Meier, C. M., Merkel, B., Ouwehand, J., & others. (2020). Light-level geolocator analyses: A user's guide. *Journal of Animal Ecology*, *89*(1), 221–236. https://doi.org/10.1111/1365-2656.13036

Lisovski, S., & Hahn, S. (2012). GeoLight–processing and analysing light-based geolocator data in r. *Methods in Ecology and Evolution*, *3*(6), 1055–1059. https://doi.org/10.1111/j.2041-210X.2012.00248.x

Wotherspoon, S., Sumner, M., & Lisovski, S. (2016). SGAT: Solar/satellite geolocation for archival tags. *R Package Version 0.1*. https://github.com/SWotherspoon/SGAT