




Athlytics: A Computational Framework for Longitudinal Analysis of Exercise Physiology Metrics from Wearable Sensor Data

Zhiang He¹

¹ Independent researcher

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [julia ferraioli](#) 

Reviewers:

- [@smnlt](#)
- [@brentscott93](#)

Submitted: 15 May 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

Summary

The proliferation of wearable sensors and consumer platforms like Strava has generated vast longitudinal data streams, creating unprecedented opportunities to investigate dynamic human physiological responses to exercise. However, translating this raw data into scientifically rigorous insights for exercise physiology research faces considerable methodological and computational challenges. *Athlytics* is an R [R Core Team (2025)] package designed as a computational framework to streamline the acquisition, analysis, and visualization of key exercise physiology metrics directly from Strava API data. It enables researchers and practitioners to systematically quantify indicators such as Acute:Chronic Workload Ratio (ACWR) [Gabbett (2016); Hulin et al. (2016)], Efficiency Factor (EF) [Allen et al. (2019)], cardiovascular decoupling [Maunder et al. (2021)], and track personal bests (PBs). By providing standardized function interfaces adhering to tidyverse principles [Wickham et al. (2019)], *Athlytics* significantly lowers the technical barrier for conducting complex longitudinal analyses, thereby enhancing the feasibility, efficiency, and reproducibility of quantitative research utilizing widely available physiological sensor data.

Statement of Need

Athlytics targets sports scientists, physiologists, and applied researchers studying longitudinal adaptations to endurance training using consumer wearables and Strava as a data hub. Despite the ubiquity of Strava data access via rStrava [Beck et al. (2024)], there is no standardized, open-source workflow in R that transforms raw API outputs into core physiological indicators (e.g., ACWR, EF, decoupling) with tidy, reproducible interfaces. This methodological gap forces many researchers to spend substantial effort on bespoke scripts for data retrieval, metric calculation, and visualization, limiting analytical scale, efficiency, and methodological transparency [Sanders et al. (2017)].

Athlytics addresses this gap by providing an integrated framework from data acquisition to the calculation and visualization of multiple physiological metrics, using paired `calculate_*` and `plot_*` functions built on the tidyverse [Wickham et al. (2019)]. The package is designed to lower the technical barrier for such analyses while maintaining scientific rigor.

Key Functionalities

Athlytics offers a modular structure, generally featuring paired `calculate_*` and `plot_*` functions for each analytical task, built upon dplyr [Wickham et al. (2023)], ggplot2 [Wickham et al. (2025)], and other tidyverse packages.

- Data Acquisition and Preprocessing:** Leverages rStrava [Beck et al. (2024)] for Strava API authentication and data retrieval. Handles date/time operations using lubridate

[Grolemund & Wickham (2011)] and rolling calculations with zoo [Zeileis & Grothendieck (2005)].

- **ACWR Trend Analysis:**
 - `calculate_acwr`: Computes daily acute (e.g., 7-day) and chronic (e.g., 28-day) rolling loads and their ratio (ACWR) based on user-specified load metrics (duration, distance, or approximated TSS/HRSS).
 - `plot_acwr`: Visualizes the ACWR time series, optionally highlighting risk zones.
- **Load Exposure Analysis:**
 - `calculate_exposure`: Computes daily acute and chronic loads to facilitate a 2D visualization of training state.
 - `plot_exposure`: Generates a scatter plot of acute vs. chronic load, often with ACWR-derived risk zones.
- **Personal Bests (PBs) Tracking:**
 - `calculate_pbs`: Identifies and records best performance times for user-specified distances from Strava activity data (relying on Strava's `best_efforts` field).
 - `plot_pbs`: Visualizes PB progression over time.
- **Efficiency Factor (EF) Analysis:**
 - `calculate_ef`: Computes EF (e.g., average speed/HR or average power/HR) from activity summary data.
 - `plot_ef`: Visualizes EF trends over time, optionally with a smoothing line.
- **Decoupling Analysis:**
 - `calculate_decoupling`: Calculates heart rate/power (or pace/heart rate) decoupling by comparing the efficiency ratio of the first half of an activity to the second half, using detailed activity stream data.
 - `plot_decoupling`: Visualizes decoupling percentages over time.

Comparison with Similar Software

Within the R ecosystem for sports science analytics, *Athlytics* occupies a distinct niche through its integrated analytical workflow designed for Strava API data and a curated set of physiological metrics. The comparison with related tools is summarized below:

Athlytics provides direct Strava API integration with comprehensive physiological metrics (ACWR, Efficiency Factor, aerobic decoupling, personal best tracking) and integrated calculation plus visualization capabilities for reproducible research workflows.

rStrava offers Strava API access only for raw activities and segments, with no built-in analysis layer or physiological metrics.

trackeR handles local files (GPX/TCX/JSON/DB3) without Strava API integration, providing generic performance summaries and basic data structuring with limited visualization.

Single-metric packages (e.g., ACWR packages) focus on individual metrics only, requiring users to combine multiple tools and provide their own data tables.

This design eliminates the need for researchers to piece together multiple single-function packages and supports reproducible longitudinal research directly from a widely-used data source.

Interactive Example

This interactive example demonstrates how to use *Athlytics* to calculate ACWR for multiple athletes and integrate it into a modeling workflow.

```
library(Athlytics)
library(dplyr); library(purrr); library(tidyr); library(lme4)

mod <- read.csv("tokens_access.csv") %>% # 1. Read athlete tokens
```

```
# 2. Calculate ACWR for each athlete using their token
mutate(acw = map(access_token, calculate_acwr,
                 activity_type = "Run",
                 load_metric   = "duration_mins",
                 acute_period   = 7,
                 chronic_period= 28)) %>%

unnest(acw) %>%

# 3. Prepare data for modeling
group_by(athlete_id) %>%
arrange(date) %>%
mutate(lag_ACWR = lag(acwr),           # Predictor: lag-1 ACWR
       perf = run_distance_m) %>%    # Response: run distance
drop_na(perf, lag_ACWR) %>%

# 4. Fit a mixed-effects model
lmer(perf ~ lag_ACWR + (1 | athlete_id), data = .)

# 5. Extract and print model coefficients
print(summary(mod)$coefficients)
```

Acknowledgments

The development of *Athlytics* [He (2025b)], which is now available from the Comprehensive R Archive Network (CRAN), relied upon the R programming language [R Core Team (2025)] and benefited from numerous open-source R packages, including *rStrava* [Beck et al. (2024)], *ggplot2* [Wickham et al. (2025)], *dplyr* [Wickham et al. (2023)], *tidyr* [Wickham et al. (2024)], *lubridate* [Grolemund & Wickham (2011)], *zoo* [Zeileis & Grothendieck (2005)], *purrr* [Wickham & Henry (2025)], and *rlang* [Henry & Wickham (2025)]. Access to data was made possible by the Strava API. We also acknowledge the preprint of this work on bioRxiv [He (2025a)].

References

- Allen, H., Coggan, A. R., & McGregor, S. (2019). *Training and Racing with a Power Meter* (3rd ed.). VeloPress. ISBN: 9781937715939
- Beck, M. W., Villarroel, P., Padfield, D., Gaborini, L., & Maltzahn, N. von. (2024). *rStrava: Access the 'Strava' API*. <https://doi.org/10.32614/CRAN.package.rStrava>
- Gabbett, T. J. (2016). The training—injury prevention paradox: should athletes be training smarter and harder? *British Journal of Sports Medicine*, 50(5), 273–280. <https://doi.org/10.1136/bjsports-2015-095788>
- Grolemund, G., & Wickham, H. (2011). Dates and Times Made Easy with lubridate. *Journal of Statistical Software*, 40(3), 1–25. <https://doi.org/10.18637/jss.v040.i03>
- He, Z. (2025a). *Athlytics: A Computational Framework for Longitudinal Analysis of Exercise Physiology Metrics from Wearable Sensor Data*. *bioRxiv*. <https://doi.org/10.1101/2025.05.01.651597>
- He, Z. (2025b). *Athlytics: Advanced Sports Performance Analysis for 'Strava' Data*. <https://doi.org/10.32614/CRAN.package.Athlytics>
- Henry, L., & Wickham, H. (2025). *rlang: Functions for Base Types and Core R and 'Tidyverse' Features*. <https://doi.org/10.32614/CRAN.package.rlang>
- Hulin, B. T., Gabbett, T. J., Lawson, D. W., Caputi, P., & Sampson, J. A. (2016). The

- 111 acute: chronic workload ratio predicts injury: high chronic workload may decrease injury
112 risk in elite rugby league players. *British Journal of Sports Medicine*, 50(4), 231–236.
113 <https://doi.org/10.1136/bjsports-2015-094817>
- 114 Maunders, E., Seiler, S., Mildenhall, M. J., Kilding, A. E., & Plews, D. J. (2021). The
115 Importance of 'Durability' in the Physiological Profiling of Endurance Athletes. *Sports*
116 *Medicine*, 51(8), 1619–1628. <https://doi.org/10.1007/s40279-021-01459-0>
- 117 R Core Team. (2025). *R: A Language and Environment for Statistical Computing*. R
118 Foundation for Statistical Computing. <https://www.R-project.org/>
- 119 Sanders, D., Abt, G., Hesselink, M. K. C., Myers, T., & Akubat, I. (2017). Methods of
120 Monitoring Training Load and Their Relationships to Changes in Fitness and Performance
121 in Competitive Road Cyclists. *International Journal of Sports Physiology and Performance*,
122 12(5), 668–675. <https://doi.org/10.1123/ijspp.2016-0454>
- 123 Wickham, H., Averick, M., Bryan, J., Chang, W., D'Agostino McGowan, L., François, R.,
124 Grolemond, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E.,
125 Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani,
126 H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686.
127 <https://doi.org/10.21105/joss.01686>
- 128 Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K.,
129 Yutani, H., & Dunnington, D. (2025). *ggplot2: Create elegant data visualisations using*
130 *the grammar of graphics*. <https://doi.org/10.32614/CRAN.package.ggplot2>
- 131 Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A Grammar*
132 *of Data Manipulation*. <https://doi.org/10.32614/CRAN.package.dplyr>
- 133 Wickham, H., & Henry, L. (2025). *purrr: Functional Programming Tools*. <https://doi.org/10.32614/CRAN.package.purrr>
- 134
- 135 Wickham, H., Vaughan, D., & Girlich, M. (2024). *tidyr: Tidy Messy Data*. <https://doi.org/10.32614/CRAN.package.tidyr>
- 136
- 137 Zeileis, A., & Grothendieck, G. (2005). zoo: S3 Infrastructure for Regular and Irregular Time
138 Series. *Journal of Statistical Software*, 14(6), 1–27. <https://doi.org/10.18637/jss.v014.i06>