




Athlytics: A Reproducible Framework for Endurance Data Analysis

Zhiang He  ¹

¹ Joint Laboratory of Biomedical Artificial Intelligence, Shanghai East Hospital, Tongji University, Shanghai, China

DOI: [10.21105/joss.09075](https://doi.org/10.21105/joss.09075)

Software

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

Editor: [Arfon Smith](#)  

Reviewers:

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

Submitted: 15 May 2025

Published: 01 July 2026

License

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

Summary

Athlytics is an R package for reproducible, offline analysis of endurance-training data exported from Strava. It reads local ZIP and CSV exports and provides a pipeline for import, quality control, cohort summaries, and visualization of metrics including acute-to-chronic workload ratios (ACWR) ([Gabbett, 2016](#)), aerobic efficiency, and cardiovascular decoupling (pa:hr). The package is designed for analyses that need to be repeatable without OAuth credentials, API quotas, or changing service availability.

Statement of Need

Researchers and coaches working with wearable data often need to combine API clients, file parsers, quality-control scripts, and metric-specific code before analysis can begin. This is especially fragile for cohort-scale studies, where authentication, rate limits, and inconsistent export formats can obstruct reproducible workflows. Athlytics fills this gap by offering one research-oriented R workflow for local Strava export archives, from raw XML/FIT/CSV-derived activity data to physiological indicators and cohort reference bands. Its primary audience is sports scientists, sports epidemiologists, and endurance coaches who need auditable, programmatic analyses across one or more athletes.

Related Work

We provide a direct feature comparison against the most relevant existing R tools — rStrava ([Beck et al., 2024](#)), trackeR ([Frick & Kosmidis, 2017](#)), activatr ([Schafer, 2024](#)), the ACWR package ([Fernandez-Santos, 2024](#)), and injurytools ([Olaskoaga & Lee, 2023](#)) — to highlight the capabilities essential for reproducible, cohort-scale research.

Feature (research-relevant)	Athlytics	rStrava	trackeR	activatr	ACWR	injurytools
Offline archives; No OAuth/tokens/quotas	Yes	No (API)	Yes	Yes	Yes (tabular)	Yes (tabular)

Feature (research-relevant)	Athlytics	rStrava	trackeR	activatr	ACWR	injurytools
API-limited (OAuth, scope, rate-limits)	No	Yes	No	No	No	No
End-to-end pipeline (Import→QC→Models→Plot)	Yes	No	Partial (parsing/viz)	Partial (parsing/pace)	No	No
Built-in metrics (ACWR/EF/decoupling)	Yes	No	No	No	Partial (ACWR only)	No
Steady-state guards & HR-coverage checks	Yes	No	No	No	No	No
Uncertainty (ACWR-EWMA confidence bands)	Yes	No	No	No	No	No
Cohort benchmarking (percentile bands)	Yes	No	Partial (summaries only)	No	No	Partial (for injury/exposure)
Diagnostic outputs (status codes/fields)	Yes	No	No	No	No	No

Compared with existing R tools, Athlytics combines local Strava export ingestion, quality-control checks, ACWR/EF/decoupling workflows, uncertainty summaries, and cohort reference bands in a single offline workflow.

Software Description

- **Offline Data Parsing:** Operates directly on local Strava ZIP exports. Using .tcx and .gpx parsers through xml2 plus optional .fit parsing through FITfileR (Smith, 2024), activity streams are loaded on demand.
- **Physiological & Load Metrics:** Supports multiple load tracking algorithms including HRSS (TRIMP-based) and TSS approximations. Calculates core metrics such as cardiovascular decoupling (pa:hr), Efficiency Factor (EF), and automatically tracks Personal Bests (PBs) using distance-based best-effort search over activity streams.
- **Signal Processing & Quality Control:** Automatically filters implausible HR, power, and velocity samples and identifies steady-state output segments using a rolling coefficient of variation (CV) algorithm to support valid physiological comparisons.

- **Uncertainty Quantification:** Provides confidence intervals for EWMA-based ACWR models using a moving-block bootstrap (Künsch, 1989), partly preserving short-range temporal dependence in training loads. This uncertainty reporting acknowledges the ongoing conceptual debates surrounding ACWR as a predictive tool (Impellizzeri et al., 2020, 2021).
- **Cohort Benchmarking & Visualization:** Generates population-level percentile reference bands (`calculate_cohort_reference()`) that can be layered onto publication-ready ACWR plots (e.g., `plot_acwr_enhanced()`).
- **Diagnostics & Transparency:** Functions return **diagnostic fields** (e.g., `status`, `quality_score`, `hr_coverage`) when inputs are insufficient, making the workflow transparent and debuggable.

Example

The following example demonstrates a common cohort analysis workflow: using pre-computed sample ACWR data to construct a synthetic three-athlete cohort, deriving cohort-wide reference bands, and plotting one athlete against those bands.

```
library(Athlytics)
library(dplyr)

# 1. Use built-in sample data to simulate a cohort of athletes
data("sample_acwr", package = "Athlytics")
set.seed(42)
cohort_acwr <- bind_rows(
  sample_acwr %>% mutate(athlete_id = "A1"),
  sample_acwr %>% mutate(
    athlete_id = "A2",
    acwr_smooth = acwr_smooth * runif(n(), 0.9, 1.1)
  ),
  sample_acwr %>% mutate(
    athlete_id = "A3",
    acwr_smooth = acwr_smooth * runif(n(), 0.85, 1.15)
  )
)

# 2. Generate cohort-wide percentile reference bands
reference_bands <- calculate_cohort_reference(
  cohort_acwr,
  metric = "acwr_smooth",
  by = character(0),
  min_athletes = 2
)

# 3. Extract individual data
individual_acwr <- cohort_acwr %>% filter(athlete_id == "A1")

# 4. Plot an individual's ACWR against the cohort reference using the enhanced plotting
plot_acwr_enhanced(
  acwr_data = individual_acwr,
  reference_data = reference_bands,
  show_ci = FALSE,
  show_reference = TRUE,
  highlight_zones = TRUE
)
```

Acknowledgements

The author thanks Benjamin S. Baumer and Iztok Fister Jr. for their insightful feedback during early development, and rOpenSci handling editor Emily Riederer and reviewers Eunseop Kim and Simon Nolte for their peer-review guidance.

The author acknowledges institutional support from the Joint Laboratory of Biomedical Artificial Intelligence, jointly established by the Institute of Software, Chinese Academy of Sciences, and Shanghai East Hospital.

This work received no external funding.

References

- 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>
- Fernandez-Santos, J. R. (2024). *ACWR: Acute chronic workload ratio calculation*. <https://doi.org/10.32614/CRAN.package.ACWR>
- Frick, H., & Kosmidis, I. (2017). trackeR: Infrastructure for running and cycling data from GPS-enabled tracking devices in r. *Journal of Statistical Software*, 82(7), 1–29. <https://doi.org/10.18637/jss.v082.i07>
- 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>
- Impellizzeri, F. M., Tenan, M. S., Kempton, T., Novak, A., & Coutts, A. J. (2020). Acute:chronic workload ratio: Conceptual issues and fundamental pitfalls. *International Journal of Sports Physiology and Performance*, 15(6), 907–913. <https://doi.org/10.1123/ijssp.2019-0864>
- Impellizzeri, F. M., Woodcock, S., Coutts, A. J., Fanchini, M., McCall, A., & Vigotsky, A. D. (2021). What role do chronic workloads play in the acute to chronic workload ratio? Time to dismiss ACWR and its underlying theory. *Sports Medicine*, 51(3), 581–592. <https://doi.org/10.1007/s40279-020-01378-6>
- Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3), 1217–1241. <https://doi.org/10.1214/aos/1176347265>
- Olaskoaga, L. Z., & Lee, D.-J. (2023). *Injurytools: A toolkit for sports injury and illness data analysis*. <https://doi.org/10.32614/CRAN.package.injurytools>
- Schafer, D. (2024). *Activatr: Utilities for parsing and plotting activities*. <https://doi.org/10.32614/CRAN.package.activatr>
- Smith, M. (2024). *FITfileR: Read FIT files in r*. GitHub repository. <https://github.com/grimbough/FITfileR>