The Journal of Open Source Software

AquaFetch: A Unified Python Interface for Water Resource Dataset Acquisition and Harmonization

Ather Abbas [1], **Sara Iftikhar** [1], **and Hylke E. Beck** [1¶]

1 King Abdullah University of Science and Technology, Thuwal, Saudi Arabia ¶ Corresponding author

DOI: 10.21105/joss.08051

Software

- Review
- Repository
- Archive

Editor: Ethan White

Reviewers:

- @tamnva
- @jeiloh

Submitted: 21 January 2025
Published: 23 August 2025

Summary

AquaFetch is a Python package designed for the automated downloading, parsing, cleaning, and harmonization of freely available water resource datasets related to rainfall-runoff processes, surface water quality, and wastewater treatment. The package currently supports 70 datasets, downloading and transforming raw data into consistent, easy-to-use analysis-ready data. This allows users to directly access and utilize the data without labor-intensive and time-consuming preprocessing.

The package comprises three submodules, each representing a different type of water resource data: `rr` for rainfall-runoff processes, `wq` for surface water quality, and `wwt` for wastewater treatment. The `rr` submodule offers data for 47,291 catchments from 36 sources worldwide, encompassing both dynamic and static features for each catchment (Figure 1). The dynamic features consist of observed streamflow and meteorological time series, averaged over the catchment area, available at daily or hourly time steps. Static features include constant parameters such as land use, soil, topography, and other physiographical characteristics in tabular format, along with catchment boundaries as shapefiles. This submodule not only provides access to established rainfall-runoff datasets such as CAMELS (Addor et al., 2017) and LamaH (Klingler et al., 2021) but also introduces 10 new datasets compiled for the first time from publicly accessible online data sources. The `wq` submodule offers access to 16 surface water quality datasets, each containing various water quality parameters measured across different spaces and times (Figure 2). The `wwt` submodule provides access to 22,471 experimental measurements related to wastewater treatment techniques such as adsorption, photocatalysis, and sonolysis.

The development of AquaFetch was inspired by the growing availability of diverse water resource datasets in recent years. As a community-driven project, the codebase is structured to allow contributors to easily add new datasets, ensuring the package continues to expand and evolve to meet future needs.

Statement of need

The rapid increase in publicly available spatio-temporal datasets in various domains, including water resources, is mainly driven by advances in computation and storage as well as demand to develop accurate data-driven solutions for challenges like climate change, water scarcity, and environmental pollution. However, importing them into Python is cumbersome, requiring researchers to navigate multiple sources and handle inconsistent formats and units. Many datasets also need extensive preprocessing, making its acquisition a complex and skill-intensive task before it can be used for analysis or modeling.

These challenges highlight the need for a unified, consistent, automated, and reusable framework for extracting hydrological and environmental data. The AquaFetch package addresses this

gap by leveraging data-handling tools such as Pandas (McKinney, 2010), NumPy (Harris et al., 2020), xarray (Hoyer & Hamman, 2017), and netCDF4 (NSF Unidata et al., 2024) to offer a streamlined workflow for automatic data extraction from multiple sources in various formats.

Several other packages also aim to facilitate the acquisition and processing of water resource data. For example, tools like dataretrieval (Hodson & Horsburgh, 2023), hydrofunctions (Roberge, 2018), and harmonize-wq (Bousquin & Mullin, 2024) provide streamlined access to hydrological and water quality data, though they are primarily limited to data from U.S. Geological Survey (USGS) and Environmental Protection Agency (EPA) stations. Data Retriever (Senyondo et al., 2017) offers access to over 200 parameters, including water quality, but lacks comprehensive rainfall-runoff datasets and wastewater treatment data. Meanwhile, tsp (Brown, 2022) focuses on managing ground temperature data, and the HyRiver suite (Taher Chegini1 & Leung, 2021) provides access to hydro-climate data exclusively from USGS's National Water Information Service (NWIS). Geodata-Harvester (Haan et al., 2023) offers reusable and automated workflows for extracting diverse geospatial and environmental data for Australia.

The Caravan initiative (Kratzert et al., 2023) stands out as a global platform for hydro-meteorological data with a distinct focus. While it aims to create large-sample hydrology datasets globally, AquaFetch facilitates access to a broad range of data via a unified interface, including processed rainfall-runoff datasets from countries like Japan, Thailand, Ireland, and Poland — regions not covered by Caravan. Additionally, Caravan emphasizes cloud-based processing using the proprietary Google Earth Engine and does not provide a standalone Python package, requiring users to upload catchment boundaries for extracting static features. This highlights the distinct operational focuses of Caravan and AquaFetch.

Ultimately, we hope this package will foster the development of benchmark datasets in hydrological and environmental sciences, enhancing the comparability and reproducibility of data-driven solutions for water resource management.

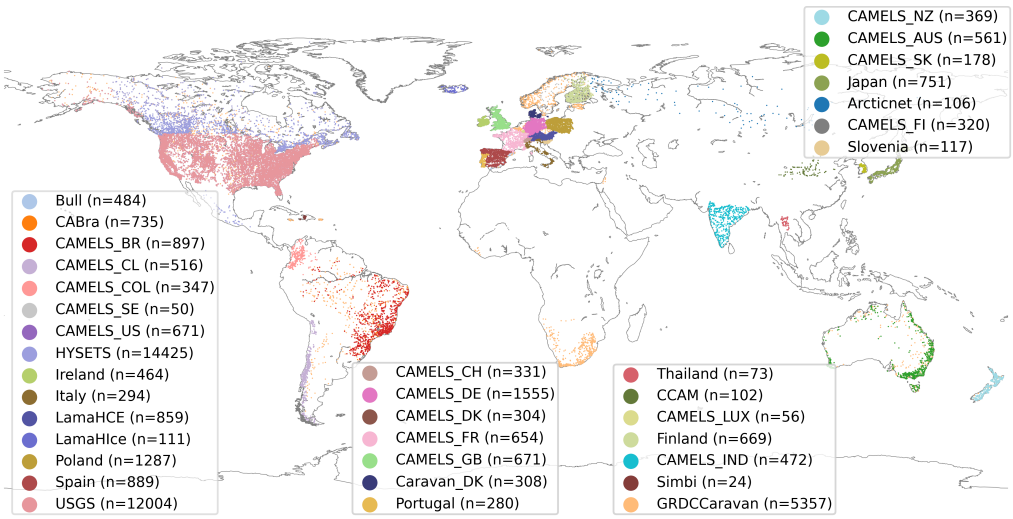

Figure 1: Locations of catchment gauge stations covered by each of the 36 rainfall-runoff datasets

Implementation and Architecture

The AquaFetch package is structured using both functional and Object-Oriented Programming (OOP) designs. The OOP design is employed for handling more complex datasets within the `rr` and `wq` submodules, while simpler datasets are managed through a functional interface. All datasets in the rr submodule can be accessed via the RainfallRunoff class, thus offering a

unified interface. The package's code is logically organized, with `rr`, `wq`, and `wwt` subdirectories present in both the source code (aqua_fetch) and tests directories. All public classes and functions are accessible from the parent directory, allowing for straightforward imports as shown below:

```
from aqua_fetch import RainfallRunoff
from aqua_fetch import SWatCh
from aqua_fetch import mg_degradation
```

Datasets are downloaded upon the first call to the respective function or class. The data is saved locally and will not be redownloaded unless the user explicitly requests it to be overwritten. The package also leverages parallel processing to expedite the downloading and parsing of large datasets in the `rr` submodule, significantly speeding up data retrieval when multiple CPU cores are available.

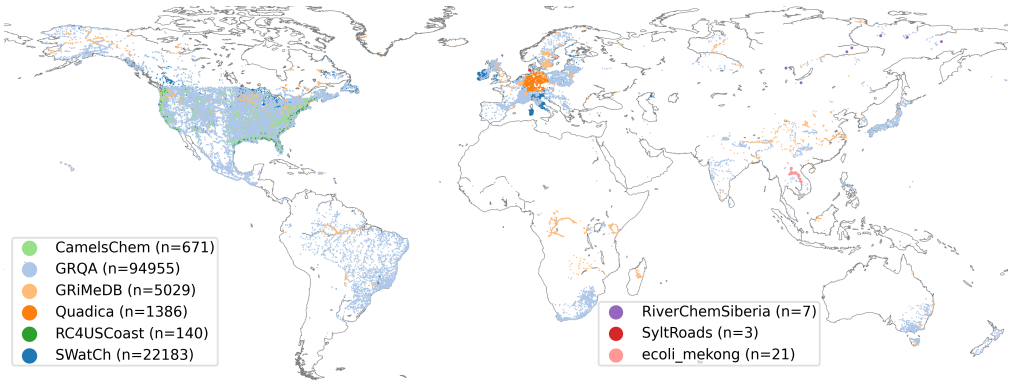

Figure 2: Locations of measuring stations of water quality datasets

Testing and dependencies

The package's core dependencies include requests (Reitz, 2018), NumPy, and Pandas. xarray is utilized for saving data in netCDF4 (NSF Unidata et al., 2024) format, which is efficient for handling large datasets; however, this step is an optional dependency. Other optional dependencies include Matplotlib (Hunter, 2007) for plotting and visualization, openpyxl (Gazoni & Clark, 2024) for parsing Microsoft Excel files, along with Fiona (Sean et al., 2024) for processing shapefiles. Adhering to the 'unit test' protocol, comprehensive testing has been implemented for all data classes and functions. Since downloading the datasets is time-consuming, the tests are conducted offline under the assumption that the datasets are already downloaded. These unit tests verify the number and types of parameters returned by the data functions.

Acknowledgements

For part of the analysis, we utilized the Shaheen~III supercomputer, managed by the Supercomputing Core Laboratory at King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia. Part of the research was supported by the KAUST/MEWA Strategic Partnership Agreement (SPA) for Water, under award numbers 6110 and 6111.

References

Addor, N., Newman, A. J., Mizukami, N., & Clark, M. P. (2017). The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, *21*(10), 5293–5313. https://doi.org/10.5194/hess-21-5293-2017

Bousquin, J., & Mullin, C. A. (2024). harmonize-wq: Standardize, clean and wrangle Water Quality Portal data into more analytic-ready formats. *Journal of Open Source Software*, *9*(102), 7305. https://doi.org/10.21105/joss.07305

Brown, N. (2022). tsp ("Teaspoon"): A library for ground temperature data. *Journal of Open Source Software*, *7*(77), 4704. https://doi.org/10.21105/joss.04704

Gazoni, E., & Clark, C. (2024). *openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files*. https://foss.heptapod.net/openpyxl/openpyxl.

Haan, S., Harianto, J., Butterworth, N., & Bisho, T. (2023). Geodata-Harvester: A Python package to jumpstart geospatial data extraction and analysis. *Journal of Open Source Software*, *8*(89), 5205. https://doi.org/10.21105/joss.05205

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hodson, H., T. O., & Horsburgh, J. S. (2023). *dataretrieval (Python): a Python package for discovering and retrieving water data available from U.S. federal hydrologic web services: U.S. Geological Survey*. https://github.com/DOI-USGS/dataretrieval-python. https://doi.org/10.5066/P94I5TX3

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Klingler, C., Schulz, K., & Herrnegger, M. (2021). LamaH-CE: LArge-SaMple DAta for Hydrology and Environmental Sciences for Central Europe. *Earth System Science Data*, *13*(9), 4529–4565. https://doi.org/10.5194/essd-13-4529-2021

Kratzert, F., Nearing, G., Addor, N., Erickson, T., Gauch, M., Gilon, O., Gudmundsson, L., Hassidim, A., Klotz, D., Nevo, S., Shalev, G., & Matias, Y. (2023). Caravan - A global community dataset for large-sample hydrology. *Scientific Data*, *10*(61). https://doi.org/10.1038/s41597-023-01975-w

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56). https://doi.org/10.25080/Majora-92bf1922-00a

NSF Unidata, Davis, G., Rew, R., Heimbigner, D., Hartnett, E., Fisher, W., & Others, M. (2024). *NetCDF-C* (Version 1.7.2). https://doi.org/10.5065/D6H70CW6

Reitz, K. (2018). *requests: A simple, yet elegant, HTTP library*. https://github.com/psf/requests.

Roberge, M. (2018). *hydrofunctions*. https://github.com/mroberge/hydrofunctions.

Sean, G., Rene, B., Joshua, A., Mike W, T., Kevin, W., Alan D, S., Micah, C., & Others, M. (2024). *Fiona* (Version 1.10.0). https://github.com/Toblerity/Fiona

Senyondo, H., Morris, B. D., Goel, A., Zhang, A., Narasimha, A., Negi, S., Harris, D.

J., Digges, D. G., Kumar, K., Jain, A., Pal, K., Amipara, K., & White, E. P. (2017). Retriever: Data Retrieval Tool. *Journal of Open Source Software*, *2*(19), 451. https://doi.org/10.21105/joss.00451

Taher Chegini1, Hong-Yi Li1, & Leung, L. R. (2021). HyRiver: Hydroclimate Data Retriever. *Journal of Open Source Software*, *6*(66), 3175. https://doi.org/10.21105/joss.03175