The Journal of Open Source Software

hf_hydrodata: A Python package for accessing hydrologic simulations and observations across the United States

Amy Defnet [1,2], **William Hasling** [1,2], **Laura Condon** [3], **Amy Johnson** [3,4], **Georgios Artavanis** [1,2], **Amanda Triplett** [3], **William Lytle** [3], **and Reed Maxwell** [2,5,6]

1 Research Software Engineering, Princeton University, USA **2** Integrated GroundWater Modeling Center, Princeton University, USA **3** Department of Hydrology and Atmospheric Sciences, University of Arizona, USA **4** CyVerse, USA **5** Department of Civil and Environmental Engineering, Princeton University, USA **6** High Meadows Environmental Institute, Princeton University, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 16 February 2024
Published: unpublished

Summary

The field of hydrologic modeling, or modeling of the terrestrial hydrologic cycle, is very data intensive. Models require many inputs to define topography, geology and atmospheric conditions. Additionally, in situ observations such as streamflow rate and depth to groundwater, can be used to evaluate model outputs and calibrate input parameters. There are many public organizations and research groups in the United States which produce and make freely available parts of this required data. However, the data have a wide range of spatiotemporal resolutions, file types, and methods of access. This makes finding and accessing all the data required for analysis a very time-consuming part of most hydrologic studies. The `hf_hydrodata` package is designed to simplify this data acquisition process by providing access to a broad array of variables, all of which have been pre-processed for consistency.

Statement of need

`hf_hydrodata` is a Python package that provides a streamlined, uniform syntax for accessing hydrologic data. Spanning the Continental United States, `hf_hydrodata` serves both gridded data and site-level point observations from the HydroData catalog. The package allows users to add filters to obtain data for only certain geographic areas and/or time periods of interest. This package was developed with hydrologists in mind, to facilitate the collection of domain-specific model inputs and validation data.

This package includes three main types of data. First we provide access to point observations that are compiled from public sources. Sources include the United States Geological Survey (USGS), the Snow Telemetry Network (SNOTEL), Soil Climate Analysis Network (SCAN), AmeriFlux, and the National Oceanic and Atmospheric Administration (NOAA). All point observation data are continuously updated to the HydroData Database and are pre-processed for consistency.

We also provide access to a national geofabric of hydrologically processed topography, land cover and hydrogeology land cover datasets that were developed from the national ParFlow model (i.e. the ParFlow CONUS model, e.g. Maxwell & Condon (2016); O'Neill et al. (2021); Yang et al. (2023)). Simulation outputs generated from the first (ParFlow CONUS1.0) and second (ParFlow CONUS2.0) generation of the ParFlow CONUS model are also available through this interface.

The Journal of Open Source Software

The HydroData catalog also contains atmospheric forcing datasets that can be used to drive hydrologic models. These large gridded datasets can be difficult to download and use in their entirety. Our interface makes it possible to easily subset just the forcings needed for a local simulation without ever downloading the entire dataset.

The aim of the `hf_hydrodata` package is to provide a "one-stop shop" for all of a hydrologists' data needs and to eliminate the burden of each researcher needing to learn multiple syntaxes in order to obtain the data relevant for their study area. It also aims to facilitate the sharing of open-source hydrologic data across research groups. `hf_hydrodata` requires a simple yet flexible set of parameters to be able to include a new offering.This keeps the barrier to entry low for members of the hydrologic community to add additional data sources to the package and keeps `hf_hydrodata` relevant as new datasets are created.

Functionality

Complete documentation of the `hf_hydrodata` package including available datasets, example workflows, and the full API reference is available on Read the Docs.

The `hf_hydrodata` API contains distinct modules for accessing gridded data and site-level point observations. The output data structure is designed to align with the data type: gridded data gets returned as a NumPy array (Walt et al., 2011) while point data gets returned in a pandas DataFrame (The pandas development team, 2020) (to connect site identifiers to time series in a straightforward manner). However the API is structured to take in compatible input parameters (where applicable), to make the data querying process as seamless as possible across the different data types.

For example, if a user wanted to obtain gridded ParFlow CONUS1 daily simulated water table depth data for the latitude/longitude bounding box of [38.749, -106.207, 41.485, -100.695] for October 1, 2003 - May 1, 2004, they would use the following syntax to get relevant data and metadata.

```python
import hf_hydrodata

gridded_parameters = {'dataset': 'conus1_baseline_mod',
                      'variable': 'water_table_depth',
                      'temporal_resolution': 'daily',
                      'aggregation': 'mean',
                      'grid': 'conus1',
                      'latlng_bounds': [38.749, -106.207, 41.485, -100.695],
                      'start_time': '2003-10-01', 'end_time': '2004-05-01'
                      }

gridded_data = hf_hydrodata.get_gridded_data(gridded_parameters)
gridded_metadata = hf_hydrodata.get_catalog_entry(gridded_parameters)
```

If they also wanted to query observational water table depth data from USGS wells for the same geography and time period, they would use the following syntax. A subset of each of the output DataFrames produced is shown in Figure 1.

```python
import hf_hydrodata

point_parameters = {'dataset': 'usgs_nwis',
                    'variable': 'water_table_depth',
                    'temporal_resolution': 'daily',
                    'aggregation': 'mean',
                    'latitude_range': (38.749, 41.485),
                    'longitude_range': (-106.207, -100.695),
```

```
                                'date_start': '2003-10-01', 'date_end': '2004-05-01'
                                }

point_data = hf_hydrodata.get_point_data(point_parameters)
point_metadata = hf_hydrodata.get_point_metadata(point_parameters)
```

	date	393358103454800	393902103554000	393902103554001	393902103554003
0	2003-10-01	2.941320	3.392424	3.617976	3.557016
1	2003-10-02	2.932176	3.386328	3.605784	3.547872
2	2003-10-03	2.950464	3.392424	3.605784	3.557016
3	2003-10-04	2.953512	3.389376	3.608832	3.553968
4	2003-10-05	2.956560	3.392424	3.605784	3.553968

	site_id	site_name	site_type	agency	state	latitude	longitude	first_date_data_available	last_date_data_available	record_count
0	393358103454800	SC00605703BAB DTX5 BEAVER CREEK	groundwater well	USGS	CO	39.566111	-103.763333	2000-02-24	2005-06-13	1923
1	393902103554000	SC00505806BBD DTX9 MUDDY CREEK	groundwater well	USGS	CO	39.650556	-103.927778	2000-04-26	2014-09-29	4104
2	393902103554001	SC00505806BBD1 DTX10A MUDDY CREEK	groundwater well	USGS	CO	39.650556	-103.927778	2000-04-26	2024-01-27	6944
3	393902103554003	SC00505806BBD3 DTX11 MUDDY CREEK	groundwater well	USGS	CO	39.650556	-103.927778	2000-04-26	2014-09-29	4126

Figure 1: Image of example site-level point observations DataFrame and select site-level attributes, as returned by the provided example function calls.

This streamlined syntax showcases the advantage of the `hf_hydrodata` package, to allow users to access a wide variety of hydrologic data from a simple Python interface.

Acknowledgements

This research has been supported by the U.S. Department of Energy Office of Science (DE-AC02-05CH11231) and the US National Science Foundation Office of Advanced Cyberinfrastructure (OAC- 2054506 and OAC-1835855).

References

Maxwell, R. M., & Condon, L. E. (2016). Connections between groundwater flow and transpiration partitioning. *Science*, *353*(6297), 377–380. https://doi.org/10.1126/science.aaf7891

O'Neill, M. M. F., Tijerina, D. T., Condon, L. E., & Maxwell, R. M. (2021). Assessment of the ParFlow–CLM CONUS 1.0 integrated hydrologic model: Evaluation of hyper-resolution water balance components across the contiguous united states. *Geoscientific Model Development*, *14*(12), 7223–7254. https://doi.org/10.5194/gmd-14-7223-2021

The pandas development team. (2020). Pandas-dev/pandas: pandas. In *Zenodo repository*. Zenodo. https://doi.org/10.5281/zenodo.3509134

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*, 22–30. https://doi.org/10.1109/MCSE.2011.37

Yang, C., Tijerina-Kreuzer, D. T., Tran, H. V., Condon, L. E., & Maxwell, R. M. (2023). A high-resolution, 3D groundwater-surface water simulation of the contiguous US: Advances in the integrated ParFlow CONUS 2.0 modeling platform. *Journal of Hydrology*, *626*, 130294. https://doi.org/10.1016/j.jhydrol.2023.130294