










HydroMT: Automated and reproducible model building and analysis

Dirk Eilander ^{1,2}, H el ene Boisgontier¹, Laur ene J. E. Bouaziz ¹, Joost Buitink¹, Ana is Couasnon ^{1,2}, Brendan Dalmijn¹, Mark Hegnauer¹, Tjalling de Jong¹, Sibren Loos¹, Indra Marth¹, and Willem van Verseveld ¹

¹ Deltares, The Netherlands ² Institution for Environmental Studies (IVM), Vrije Universiteit Amsterdam, The Netherlands  Corresponding author

DOI: [10.21105/joss.04897](https://doi.org/10.21105/joss.04897)

Software

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

Editor: [Jayaram Hariharan](#)  

Reviewers:

- [@JannisHoch](#)
- [@mcflugen](#)
- [@LejoFlores](#)

Submitted: 28 September 2022

Published: 07 March 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Many impact assessment studies rely on hydrological and hydrodynamic (hydro) models. These models typically require a large set of parameters derived from different datasets and hence manual setup can be time consuming and hard to reproduce. HydroMT (Hydro Model Tools) is an open-source Python package that aims to make the process of building model instances and analyzing model results automated and reproducible. The package provides a common interface to data and model instances, workflows to transform data into models based on (hydrological) GIS and statistical methods, and various methods to analyze model results. The user can describe a full model instance including its forcing in a single configuration file based on a sequence of workflows, making the process reproducible, fast, and modular. The package has been designed with an iterative, data-centered modeling process in mind. First-order model schematizations can be built for any location in the world by leveraging open global datasets. These can later be improved by updating the input datasets with detailed local datasets. This iterative process enables the user to quickly get an initial result to then make informed decisions about the most relevant improvements and/or required data collection and to kick-start discussions with stakeholders. Furthermore, model parameter maps or forcing data can easily be modified for sensitivity analysis or calibration to support these robust modeling practices. HydroMT is successfully being used for several model software through a plugin infrastructure that allow for model specific functionality, such as readers and writers of model data formats, and can easily be extended to new model software. Currently supported model software include the distributed rainfall-runoff model Wflow, the hydrodynamic flood model SFINCS, the water quality models D-Water Quality and D-Emissions and the flood impact model Delft-FIAT.

Statement of need

Hydrological and hydrodynamic (hydro) models are crucial to understand water systems and perform impact assessment studies (e.g., [UNDRR, 2022](#)). These models require different accurate input data. For instance, hydrodynamic models typically require bathymetric and/or topographic data, surface roughness data and boundary forcing data (e.g., water level time series). However, sufficiently accurate datasets are often not readily available and need to be converted into model-specific, sometimes idiosyncratic, file formats. For instance, many Digital Elevation Models (DEMs) require ample preprocessing as these typically represent the elevation of the top of buildings, trees or water surface, while in the model the terrain and riverbed elevation are required ([Hawker et al., 2022](#); [Neal et al., 2021](#)). Therefore, hydro models require various steps to process raw input data to model data which, if done manually, makes the process time consuming and hard to reproduce. Furthermore, it makes it hard to

improve model instances based on new or updated (large) datasets, such as (global) DEMs and land use maps, potentially slowing down the uptake of such datasets for geoscientific modeling. Hence, there is a clear need for automated setup of model instances for increased transparency and reproducibility in hydro modeling (Hall et al., 2022; White et al., 2020).

Several open-source packages have been proposed to facilitate automated and reproducible model building. For the Soil and Water Assessment Tool (SWAT) model (Arnold et al., 2012), the SWAT+AW package provides scripted automated workflows (AW) that are compatible with the SWAT GUI to improve reproducibility (Chawanda et al., 2020). For groundwater modeling the FloPy (Bakker et al., 2016) and iMOD-Python (Engelen et al., 2022) packages provide scripting interfaces to automatically setup instances of the groundwater model MODFLOW (Harbaugh, 2005). Similarly, LFPtools (Sosa et al., 2020) provides scripts to setup instances of the hydrodynamic model LISFLOOD-FP (Bates et al., 2010). While these packages can provide automated and reproducible modeling, their functionality is model software specific and often requires scripting skills to be used.

HydroMT (Hydro Model Tools) is an open-source Python package that aims to make the process of building a model instance and analyzing model results automated and reproducible for any hydro software. As such it contributes towards the FAIR principles of research software (FAIR4RS) which aim to increase the transparency, reproducibility, and reusability of research (Chue Hong et al., 2021). Compared to other packages for automated model building, it is data- and model-agnostic meaning that data sources can easily be interchanged without additional coding and the generic model interface can be used for different model software. This makes it possible to reuse workflows to prepare input for different model software that require the same parameter (e.g., Manning roughness derived from landuse maps) or workflows used for postprocessing of results (e.g., calculating model skill statistics), thereby providing an environment for controlled model experiments. Furthermore, it provides a high-level command line interface making it accessible to modelers without Python scripting experience.

Several recent applications have already used HydroMT. It has been used to build instances of the hydrological Wflow model (Imhoff et al., 2020; Verseveld et al., 2022) for climate change impact studies (Sperna Weiland et al., 2021); for a large sample assessment of spatial scales in hydrological modeling (Aerts et al., 2021); to estimate water balance components based on global data in a data-scarce area (Rusli et al., 2021); and to build instances of the hydrodynamic SFINCS model (Leijnse et al., 2021) to simulate compound flood hazard from (global) datasets in any coastal delta globally (Eilander et al., 2022). While the package is sufficiently general to support model software outside the domain of hydro models, the focus of many workflows is on typical hydrological data and parameters and hence hydro models benefit most from the tool.

Functionality

The functionality of HydroMT can be broken down into three components, which are around input data, model instances, and methods and workflows. Users can interact with HydroMT through a high-level command line interface (CLI) to build model instances from scratch, update existing model instances or analyze model results (in progress). Furthermore, a Python interface is available that exposes all functionality for experienced users. A schematic overview of the package is provided in Figure 1 and each HydroMT component is discussed below.

- **Input Data.** HydroMT is data-agnostic through the Data Adapter, which reads a wide range of data formats and unifies the input data (e.g., on-the-fly renaming and unit conversion). Datasets are listed and passed to HydroMT in a user defined data catalog yaml file. HydroMT also provides several predefined data catalogs with mostly global datasets that can be used as is, although not all datasets in these catalogs are openly accessible. Currently, four different types of input data are supported and represented by a specific Python data object: gridded datasets such as DEMs or gridded spatially

distributed rainfall datasets (represented by RasterDataset objects, a raster-specific type of Xarray Datasets); tables that can be used to, for example, convert land classes to roughness values (represented by Pandas DataFrame objects); vector datasets such as administrative units or river centerlines (represented by Geopandas GeoDataFrame objects); and time series with associated geolocations such as observations of discharge (represented by GeoDataset objects, a geo-specific type of Xarray Datasets).

- **Models.** HydroMT defines any model instance through the model-agnostic Model API based on several components: maps, geometries, forcings, results, states, and the model simulation configuration. For different types of general model classes (i.e., gridded, lumped, mesh and network models) additional model components have been defined. Each component is represented with a specific Python data object to provide a common interface to different model software. Model instances can be built from scratch, and existing instances can be updated based on a pipeline of methods defined in a model configuration ini file. While HydroMT provides several general model classes that can readily be used, it can also be tailored to specific model software through a plugin infrastructure. These plugins have the same interface, but with model-specific file readers, writers and workflows.
- **Methods and workflows.** Most of the heavy work in HydroMT is done by methods and workflows, indicated by the gear wheels in [Figure 1](#). Methods provide the low-level functionality such as GIS rasterization, reprojection, or zonal statistics. Workflows combine several methods to transform data to a model layer or postprocess model results. Examples of workflows include the delineation of hydrological basins (watersheds), conversion of landuse-landcover data to model parameter maps, and calculation of model skill statistics. Workflows are implemented for the data types mentioned above to allow reusing common workflows between HydroMT plugins for different model software.

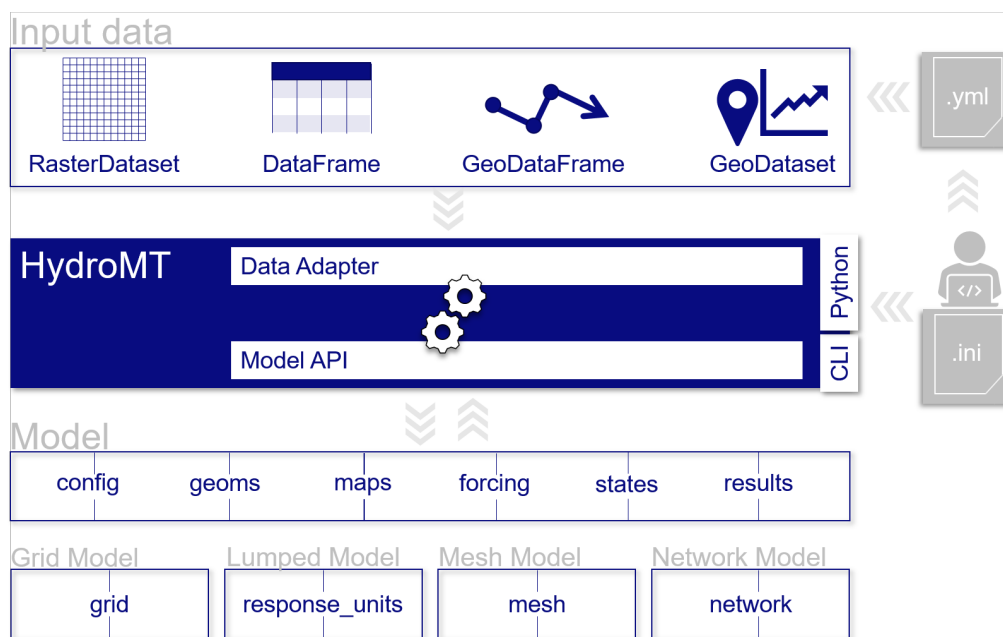


Figure 1: Schematic overview of HydroMT design.

Example usage

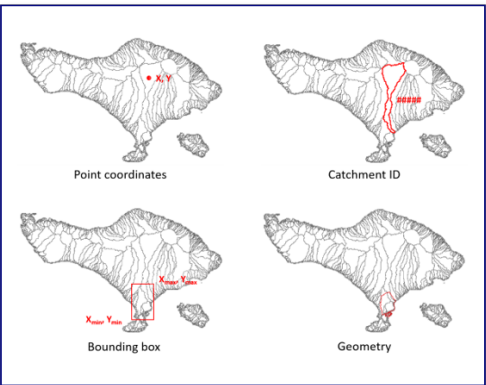
Here, a high-level example of how to build a model instance using HydroMT is provided. Building a model instance from scratch with HydroMT involves the following four generic steps, see [Figure 2](#). First, the user defines the input data in a *yaml* data catalog file or selects

available datasets from a pre-defined data catalog. Second, the user defines the region that describes the area of interest. The region can be based on a simple bounding box or geometry, but also a (sub)(inter)basin that is delineated on-the-fly based on available hydrography data. Third, the user configures the model instance in an *ini* configuration file. This file represents a model software specific recipe to build a model instance and this is where the modeler can make choices how to best setup the model instance. It lists the workflows and their arguments in order of execution and in combination with the data catalog file provides a reproducible recipe to build a model instance. Available workflows (i.e., model setup methods) and their arguments are described in the documentation. Finally, the user runs the HydroMT build method from either command line or Python.

1. Prepare data catalog

```
merit_hydro_v1.0:
  crs: 4326
  data_type: RasterDataset
  driver: raster
  kwargs:
    chunks:
      x: 6000
      y: 6000
  meta:
    category: topography
    paper_doi: 10.1029/2019WR024873
    paper_ref: Yamazaki et al. (2019)
    source_license: CC-BY-NC 4.0 or ODbL 1.0
    source_url: http://hydro.iis.u-
    tokyo.ac.jp/~yamada/MERIT_Hydro
    source_version: 1.0
    path: topography/merit_hydro/*.vrt
```

2. Define model region



3. Configure model setup

```
[setup_config] # options parsed to wflow :
starttime = 2010-01-01T00:00:00
endtime = 2010-03-31T00:00:00
timestepsecs = 86400
input.path_forcing = inmaps-era5-2010.nc

[setup_basemaps]
hydrography_fn = merit_hydro # source hydrography dat
basin_index_fn = merit_hydro_index # source of basin inde
upscale_method = ihu # upscaling method for flow

[setup_rivers]
hydrography_fn = merit_hydro # source hydrography d:
river_geom_fn = rivers_lin2019_v1 # river source data w:
river_upa = 30 # minimum upstream are:
rivdph_method = powlaw # method to estimate d:
min_rivdph = 1 # minimum river depth |
min_rivwth = 30 # minimum river width |
slope_len = 2000 # length over which tp
smooth_len = 5000 # length over which to
```

4. Run HydroMT

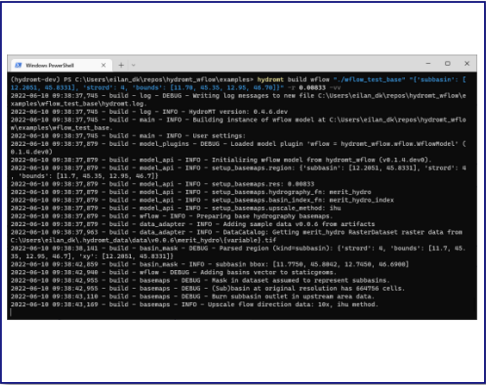


Figure 2: Workflow of building a model from scratch with HydroMT.

Audience

HydroMT is intended for students, researchers and modelers who would like to automatically build reproducible hydro models, anywhere in the world and based on the input data of their choice. For users, no Python knowledge beyond installing the package is required as HydroMT provides a CLI for common high-level functionalities. However, with Python knowledge, experienced users can directly use methods and workflows of HydroMT and interact their with model instances.

Dependencies

HydroMT builds on many open source (Python) packages including Xarray (Hoyer & Hamman, 2017), dask (Dask Development Team, 2016), rasterio (Gillies, 2013), rioxarray (Snow et al., 2022), pandas (The pandas development team, 2022), geopandas (Jordahl et al., 2021), pyflwdir (Eilander, 2022), and GDAL (GDAL/OGR contributors, 2022).

Acknowledgements

The developments have been sponsored by Deltares Strategic Research funding, the Deltares Hydrology Product Line, and the Netherlands Organization for Scientific Research (NWO) in the form of a VIDI grant (Grant No. 016.161.324).

References

- Aerts, J. P. M., Hut, R. W., Giesen, N. C. van de, Drost, N., Verseveld, W. J. van, Weerts, A. H., & Hazenberg, P. (2021). Large-sample assessment of spatial scaling effects of the distributed wflow_sbm hydrological model shows that finer spatial resolution does not necessarily lead to better streamflow estimates. In *Hydrol. Earth Syst. Sci.* (pp. 1–28). <https://doi.org/10.5194/hess-2021-605>
- Arnold, J. G., Moriasi, D. N., Gassman, P. W., Abbaspour, K. C., White, M. J., Srinivasan, R., Santhi, C., Harmel, R. D., Van Griensven, A., Van Liew, M. W., & Others. (2012). SWAT: Model use, calibration, and validation. *Transactions of the ASABE*, 55(4), 1491–1508.
- Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fienen, M. N. (2016). Scripting MODFLOW model development using python and FloPy. *Ground Water*, 54(5), 733–739. <https://doi.org/10.1111/gwat.12413>
- Bates, P. D., Horritt, M. S., & Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *J. Hydrol.*, 387(1-2), 33–45. <https://doi.org/10.1016/j.jhydrol.2010.03.027>
- Chawanda, C. J., George, C., Thiery, W., Griensven, A. van, Tech, J., Arnold, J., & Srinivasan, R. (2020). User-friendly workflows for catchment modelling: Towards reproducible SWAT+ model studies. *Environmental Modelling & Software*, 134, 104812. <https://doi.org/10.1016/j.envsoft.2020.104812>
- Chue Hong, N. P., Katz, D. S., Barker, M., Lamprecht, A.-L., Martinez, C., Psomopoulos, F. E., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., Honeyman, T., Struck, A., Lee, A., Loewe, A., Werkhoven, B. van, Jones, C., Garijo, D., Plomp, E., Genova, F., ... Wg, R. F. (2021). *FAIR principles for research software (FAIR4RS principles)*. Research Data Alliance. <https://doi.org/10.15497/RDA00068>
- Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. Dask Development Team.
- Eilander, D. (2022). *pyFlwDir: Fast methods to work with hydro- and topography data in pure python*. <https://doi.org/10.5281/zenodo.6108034>
- Eilander, D., Couason, A., Leijnse, T., Ikeuchi, H., Yamazaki, D., Muis, S., Dullaart, J., Winsemius, H. C., & Ward, P. J. (2022). A globally-applicable framework for compound flood hazard modeling. *EGU Sphere*, 1–40. <https://doi.org/10.5194/egusphere-2022-149>
- Engelen, J. van, Delsman, J., & Bootsma, H. (2022). *Reproducible large-scale groundwater modelling projects using the iMOD python package* (No. EGU22-5510). Copernicus Meetings. <https://doi.org/10.5194/egusphere-egu22-5510>

- GDAL/OGR contributors. (2022). *GDAL/OGR geospatial data abstraction software library*. Open Source Geospatial Foundation. <https://doi.org/10.5281/zenodo.5884351>
- Gillies, S. (2013). *Rasterio: Geospatial raster I/O for python programmers*.
- Hall, C. A., Saia, S. M., Popp, A. L., Dogulu, N., Schymanski, S. J., Drost, N., Emmerik, T. van, & Hut, R. (2022). A hydrologist's guide to open science. *Hydrol. Earth Syst. Sci.*, 26(3), 647–664. <https://doi.org/10.5194/hess-26-647-2022>
- Harbaugh, A. W. (2005). *MODFLOW-2005 : The U.S. Geological survey modular ground-water model—the ground-water flow process*. US Geological Survey. <https://doi.org/10.3133/tm6a16>
- Hawker, L., Uhe, P., Paulo, L., Sosa, J., Savage, J., Sampson, C., & Neal, J. (2022). A 30 m global map of elevation with forests and buildings removed. *Environ. Res. Lett.*, 17(2), 024016. <https://doi.org/10.1088/1748-9326/ac4d4f>
- Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in python. *J. Open Res. Softw.*, 5(1), 10. <https://doi.org/10.5334/jors.148>
- Imhoff, R. O., Verseveld, W. J. van, Osnabrugge, B. van, & Weerts, A. H. (2020). Scaling Point-Scale (pedo)transfer functions to seamless Large-Domain parameter estimates for High-Resolution distributed hydrologic modeling: An example for the rhine river. *Water Resour. Res.*, 56(4), 1–28. <https://doi.org/10.1029/2019WR026807>
- Jordahl, K., Van den Bossche, J., Fleischmann, M., McBride, J., Wasserman, J., Badaracco, A. G., Gerard, J., Snow, A. D., Tratner, J., Perry, M., Farmer, C., Hjelle, G. A., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Ward, B., Caria, G., Taves, M., ... Wasser, L. (2021). *Geopandas/geopandas: v0.10.2*. <https://doi.org/10.5281/zenodo.5573592>
- Leijnse, T., Ormond, M. van, Nederhoff, K., & Dongeren, A. van. (2021). Modeling compound flooding in coastal systems using a computationally efficient reduced-physics solver: Including fluvial, pluvial, tidal, wind- and wave-driven processes. *Coast. Eng.*, 163, 103796. <https://doi.org/10.1016/j.coastaleng.2020.103796>
- Neal, J., Hawker, L., Savage, J., Durand, M., Bates, P., & Sampson, C. (2021). Estimating river channel bathymetry in large scale flood inundation models. *Water Resour. Res.*, 57(5). <https://doi.org/10.1029/2020wr028301>
- Rusli, S. R., Weerts, A. H., Taufiq, A., & Bense, V. F. (2021). Estimating water balance components and their uncertainty bounds in highly groundwater-dependent and data-scarce area: An example for the upper citarum basin. *Journal of Hydrology: Regional Studies*, 37, 100911. <https://doi.org/10.1016/j.ejrh.2021.100911>
- Snow, A. D., Brochart, D., Raspaud, M., Bell, R., RichardScottOZ, Chegini, T., Amici, A., Braun, R., Annex, A., Hoese, D., Bunt, F., GBallesteros, Hamman, J., Zehner, M., Cordeiro, M., Henderson, S., Miller, S., Badger, T. G., Augspurger, T., ... Pmallas. (2022). *Corteva/rioxarray: 0.11.1 release*. Zenodo. <https://doi.org/10.5281/zenodo.6478182>
- Sosa, J., Sampson, C., Smith, A., Neal, J. C., & Bates, P. D. (2020). A toolbox to quickly prepare flood inundation models for LISFLOOD-FP simulations. *Environ. Modell. Softw.*, 123(August 2019), 104561. <https://doi.org/10.1016/j.envsoft.2019.104561>
- Sperna Weiland, F. C., Visser, R. D., Greve, P., Bisselink, B., Brunner, L., & Weerts, A. H. (2021). Estimating regionalized hydrological impacts of climate change over europe by performance-based weighting of CORDEX projections. *Front. Water*, 3. <https://doi.org/10.3389/frwa.2021.713537>
- The pandas development team. (2022). *Pandas-dev/pandas: pandas (Version v1.5.2)*. Zenodo. <https://doi.org/10.5281/zenodo.7344967>

UNDRR. (2022). *Global assessment report on disaster risk reduction 2022: Our world at risk: Transforming governance for a resilient future*. United Nations Office for Disaster Risk Reduction. ISBN: 9789212320281

Verseveld, W. J. van, Weerts, A. H., Visser, M., Buitink, J., Imhoff, R. O., Boisgontier, H., Bouaziz, L., Eilander, D., Hegnauer, M., Velden, C. ten, & Russell, B. (2022). Wflow_sbm v0.6.1, a spatially distributed hydrologic model: From global data to local applications. In *Geoscientific Model Development Discussions*. <https://doi.org/10.5194/gmd-2022-182>

White, J. T., Foster, L. K., Fienen, M. N., Knowling, M. J., Hemmings, B., & Winterle, J. R. (2020). Toward reproducible environmental modeling for decision support: A worked example. *Front. Earth Sci.*, 8. <https://doi.org/10.3389/feart.2020.00050>