Multiorder Hydrologic Position for Europe as a Set of Metrics for Hydrologic Modelling and Groundwater Mapping

Maximilian Nölscher[*, 1]**, Michael Mutz**[2]**, and Stefan Broda**[1]

[1]Federal Institute for Geosciences and Natural Resources (BGR), Berlin, 13593, Germany
[2]independent researcher
[*]corresponding author: Maximilian Nölscher (maximilian.noelscher@bgr.de, max-n@posteo.de)

ABSTRACT

The presented dataset EU-MOHP v013.1.0 provides cross-scale information on the multiorder hydrologic position (MOHP) of a geographic point within its respective river network and catchment as gridded maps. More precisely, it comprises the three measures “lateral position” (LP) as a relative measure of the position between the stream and the catchment divide, “divide stream distance” (DSD) as sum of the distances to the nearest stream and divide and “stream distance” (SD) as an absolute measure of the distance to the nearest stream. These three measures are calculated for several hydrologic orders to reflect different spatial scales. Its spatial extent covers major parts of the European Economic Area (EEA39) which also largely coincides with physiographical Europe. Although there might be many potential use cases, this dataset serves predominantly as valuable static environmental predictor variable for hydrogeological and hydrological modelling such as mapping or regionalization tasks using machine learning.

Background & Summary

In recent years, data science tools such as machine learning are increasingly applied to and specifically developed for hydro(geo)logical challenges and research questions[1,2]. In the field of hydrogeology, machine learning has been used successfully for groundwater level prediction and a variety of mapping tasks[3–10]. Since machine learning models – except for hybrid- or physics-guided models – are purely based on data without any built-in knowledge of physical processes, it is important to provide as many features (also called predictor variables or explanatory variables) as possible that have an impact on the target variable to potentially enable the machine learning algorithm to approximate the underlying process. For surface and near-surface processes, this criterion can be more or less fulfilled by the availability of remote sensing data, whereas for modelling sub-surface processes such as in hydrogeology, this poses a serious challenge.

The key motivation for this dataset is to partially close this gap by providing a set of features that introduce hydrological context to machine learning models regarding the horizontal position of a point within its catchment. Therefore, it serves as a proxy for multiple geophysical characteristics of a hydrologic system. and complements commonly available datasets such as land-use and land-cover, geological or soil maps. This dataset is strongly inspired by Belitz et al. (2019)[11] and adapts their ideas and methods to the “EU-Hydro - River Network Database”[12] but — in contrast — using free open-source software and a strong focus on reproducibility. For more detailed background, we refer to Belitz et al. (2019)[11].

In their study, Belitz et al. (2019)[11] also provide results from case studies to prove that the multiorder hydrologic position is a valuable feature when mapping diverse geophysical targets using machine learning. Its benefit to the performance of machine learning models has also been acknowledged by several other studies[7,13,14].

Being a static geophysical catchment attribute, the gridded maps of the EU-MOHP dataset[15] can be used as features in any machine learning task in the domain of hydrology and hydrogeology. This dataset can be applied at multiple spatial scales – from local via regional to continental scales. Examples of use cases might be the mapping of hydrogeochemical parameters or hydraulic variables like depth to groundwater, the prediction of groundwater levels or catchment classification tasks using unsupervised machine learning methods.

The EU-MOHP v013.1.0 dataset[15] comprises the 3 measures

- lateral position (LP)
- divide stream distance (DSD) and
- stream distance (SD)

for each of the 9 hydrologic orders which leads to $n_{measures} \cdot n_{hydrologic\,orders} = 27$ different metrics to be used as features. Spatially, the dataset covers major parts of physiographical Europe and all of the 39 countries in the European Economic Area (EEA39). More precisely, it covers the 10 largest contiguous land masses of the EEA39 (Figure 1).

Conceptually, the three measures LP, DSD and SD of EU-MOHP[15] are based on the idea that the location in hydrologic systems matters[11]. A location can be e.g. close to the confluence of two large rivers or in another extreme close the catchment boundary of headwater streams. The location or hydrologic position refers to the position of a point between a stream and its catchment divide. Thiessen divides are used as catchment boundaries instead of divides that are generated from digital elevation models (DEM). A Thiessen divide is the outline of a Thiessen catchment which is the area that contains all points to which a stream is closer than any other stream[16]. One major advantage is that Thiessen divides can be calculated purely based on the river network itself while avoiding issues such as closed lows in the resulting metrics[11]. This advantage outweighs the numerous minor problems associated with DEM-based catchments especially due to the uncertain correspondence of the subsurface catchment to the surface catchment. A detailed discussion on the preference of Thiessen divides over topographic divides is provided in Belitz et. al. (2019), section 2.2.[11].

The overall concept also includes the spatial scale that the role or importance of different hydrologic processes can depend upon. This principle is addressed by calculating the EU-MOHP[15] measures for different hydrologic orders. The hydrologic orders are based on the stream orders of the river network. For a specific hydrologic order i only streams with a stream order $>= i$ are used (e.g. for stream order 2, all streams with stream order 2, 3, 4 and greater, compare Figure 2A and B). This involves stepwise pruning of the smallest streams from the river network for each hydrologic order, which subsequently represent different spatial scales. Here, the stream orders are defined according to Strahler (1957)[17] where all streams between the headwaters and the first confluence are assigned to the first stream order. The stream order downstream of a confluence increases by 1 if the upstream stream orders are equal. If the stream orders are not equal, it inherits the greater stream order.

Based on the river network and the Thiessen divides, the EU-MOHP[15] measures are calculated with

$$\mathrm{LP}_i = \frac{\mathrm{DS}_i}{\mathrm{DS}_i + \mathrm{DD}_i} \tag{1}$$

$$\mathrm{DSD}_i = \mathrm{DS}_i + \mathrm{DD}_i \tag{2}$$

$$\mathrm{SD}_i = \mathrm{DS}_i \tag{3}$$

where i is the hydrologic order, DS is the horizontal distance from a point to the nearest stream (Figure 2) and DD is the horizontal distance to the nearest Thiessen divide under the condition that the divide is on the same side of the stream or in other words (Figure 2).

Examples of the generated EU-MOHP v013.1.0[15] maps are shown in Figure 3 for the two hydrologic orders 3 and 4 for Sardinia.

Methods

Most processing and calculation steps are done in the R programming language (Figure 4C)[18]. Due to the memory size of this dataset as well as for the sake of computational speed, a PostgreSQL database with PostGIS extension is used for some processing steps of vector data and a GRASS GIS database is used for all final raster based calculations of the EU-MOHP[15] metrics (Figure 4D and E). For reproducibility and programming reasons, all processing steps including the databases are tracked and executed through a data processing pipeline using the targets package in R (Figure 4C)[19]. More details can be found in the following sections or in the code itself (see Code availability). This processing or targets pipeline can be seen as programming script that tracks each step and skips processing steps that are still up-to-date when re-running the script after changes in the code.

To better refer to the code during the description of the methods, each processing step provides the name of its related target in the targets pipeline and the file containing this target. Targets titled as *helper target* in the code are not described here because they are not relevant to the description of the methods and exist only for technical reasons. For consistency, all column names are changed to lowercase, e.g. *OBJECT_ID* to *object_id*. The usage of “processing step” translates to a “target” in the targets pipeline which is also called “processing pipeline.”

Directory & File Structure

The directory and file structure of the project folder containing all code and files to generate this dataset is summarized in Figure 5 in a tree structure. Files and directories that are not relevant for describing the methods are not shown here. The project folder as the top level directory is the working directory. The file *config.yml* (line 2) contains definitions of variables that are meant to be set by a user before running the targets pipeline. The most relevant variable is *cellsize* which sets the spatial resolution of the resulting EU-MOHP gridded maps[15]. Another important variable is *area* to switch between a test study area and the complete study area for all EEA39. The test study area represents a small fraction of the study area. This reduces the runtime of the pipeline for testing purposes. The folder *grassdata* (line 3) is used for writing the GRASS GIS databases to. The folder *input_data* (line 4) contains all required input data. Firstly, the sub-folder *data* (line 5) comprises the river network data as one single folder per basin as it is derived after unzipping the downloaded "EU-Hydro – River Network Database" data[12] (see Underlying Dataset). The second sub-folder *EUHYDRO_Coastline_EEA39_v013* (line 6) contains the coastline data (see Underlying Dataset). The third sub-folder *studyarea_test* (line 7) contains a test study area as Shape file for pipeline testing purposes only (see Code availability). The file *macro_mohp_feature.Rproj* (line 8) is the R project file. The folder *output_data* (line 9) contains three sub-directories where the final EU-MOHP gridded maps[15] are written to. These directories are created by the pipeline if they don't already exist. *R* (line 13) contains R scripts where custom functions and constants are defined. *renv* (line 21) and the file *renv.lock* (line 27) are related to the R package renv that tracks versions of package dependencies[20]. The R script *run_pipeline.R* (line 28) contains code to execute the targets pipeline that does all the data processing and calculations. *targets* (line 29) contains the definition of all targets or processing steps of the pipeline. For overview reasons, it is split thematically across multiple files. *_targets* (line 35) is used by the targets package internally. The file *_targets.R* (line 38) sets up the targets pipeline and loads all dependencies.

Underlying Dataset

The generation of this dataset is based on two data products, the "EU-Hydro – River Network Database" version v013[12] and "EU-Hydro – Coastline" version v013[21] with the advantage that data dependencies are low. Therefore, it is possible to transfer the methodology to other regions with only little effort. Table 4 provides an overview of the input data layers necessary for generating the dataset. These input data layers are all derived from layers of GeoPackages or Shape files of the two previously mentioned data products.

The "EU-Hydro – River Network Database"[12] as well as the "EU-Hydro – Coastline"[21] has been manually downloaded from the Copernicus - Land Monitoring Service website (Figure 4A) in GeoPackage (*.gpkg*) and Shapefile (*.shp*) file format, respectively (Figure 4B)[12,21]. The river network data is provided as two .gpkg files in one folder with the suffix *_GPKG* in the folder name for each of the 35 major river basins in the EEA39 countries. All files have a total size of approximately 14GB when unzipped. The single *.shp* file containing the coastline has a size of 288MB. For instructions on accessing this underlying data, see Code availability.

Data Import

The river network comprises the layers *Canals_l*, *Ditches_l* and *River_Net_l* in the "*euhydro_<name of the river basin>_v013*" named .gpkg file of all river basins of the river network data. They are imported with the target *river_networks* in *import_targets.R* (Figure 5, line 31) from the directory *input_data/data*. The layer *Canals_l* contains canals, which are defined as "an artificial waterway with no flow, or a controlled flow, usable or built for navigation"[22]. The layer *Ditches_l* contains ditches, which are defined as "an artificial waterway with no flow, or a controlled flow, usually unlined, used for draining or irrigating land"[22]. The layer *Rivers_l* contains rivers, which are defined as "a naturally flowing watercourse"[22].

The surface water bodies are derived from the layer *InlandWater*. This layer is imported by the target *inland_waters* in *import_targets.R* and contains inland water defined as "a large body of water entirely surrounded by land."[22]. In the same step, all geometries with an area smaller than $4 \cdot \text{cellsize}^2 = 4 \cdot (30\,\text{m})^2 = 3600\,\text{m}^2$, where cellsize is the spatial resolution, are removed to exclude geometries that only have a negligible impact on the result.

The river basins are based on the layer "*<name of river basin>_eudem2_basins_h1*" in the "*drainage_network_<name of the river basin>_public_beta_v009*" named .gpkg file. This layer is imported with the target *river_basins* in *studyarea_targets.R* (Figure 5, line 34) and contains polygon geometries of all sub-basins of each major basin. The spatial coverage of the river basins is the basis for the study area. The study area itself delineates the area for which the EU-MOHP[15] metrics will be calculated.

The fourth required input data layer is the coastline. It is imported with the target *coastline_grouped* in *studyarea_targets.R*.

In this study, the coordinate reference system (CRS) ETRS89-extended / LAEA Europe with the EPSG code 3035 is used. Therefore, all data is reprojected to this CRS after importing if necessary.

Preprocessing

River Basins/ Study Area

The preprocessing steps related to the river basins are described first because they aim at the determination of a study area, which is required for subsequent processing steps. The EU-MOHP[15] measures are then calculated for this area as a last step. The following steps refer to targets that can be found in the file *studyarea_targets.R* (Figure 5, 34). After the previously mentioned import, the sub-basins are unioned basin-wise (target: *river_basins_unioned*). Then, all polygons belonging to European oversea territories such as the French islands in the Caribbean are removed (target: *river_basins_subset*). The resulting polygon geometries are unioned in the PostgreSQL database (target: *river_basins_subset_union_in_db*) to reduce the runtime of this union. Previous attempts to perform this union in R have shown too long runtimes, because the polygon geometries have a large amount of nodes due to the high details in the digitized coastline. Subsequently, out of these polygons of contiguous land masses the 10 largest polygons by area are chosen as study area (target: *river_basins_region_name*). Lastly, names are automatically assigned to each of the polygons (target: *selected_studyarea*). These names are mainly used to generate the output file names at the end of the pipeline.

River Network

After the data import of the river network data, the next step is the filtering of linestring geometries from the river network based on the attribute columns *dfdd* and *hyp* (target: *river_networks_non_dry_selected_streamtypes* in the file *preprocessing_targets.R*; Figure 5, line 30). *dfdd* classifies the geometries into BH140 (river), BH020 (canal) and BH030 (ditch). Canals and ditches are removed from the river network for several reasons through filtering out the geometries with the value BH140. Many of the canal and ditch geometries have missing stream order values, which is required for the following processing steps. Another reason is the assumption that canals might be hydraulically disconnected to the natural hydrologic system through walls with low permeability. Lastly, the overall importance of canals and ditches is low when comparing the number of geometries to rivers as shown in Figure 6. The column *hyp* classifies the geometries into the following degrees of hydrologic persistence: 1 (Perennial), 2 (Intermittent), 3 (Ephemeral) and 4 (Dry)[22]. Geometries with the value 4 (Dry) are removed. Then, missing and invalid stream order values are imputed with the value 1 to include them in the first hydrologic order (target: *river_networks_imputed_streamorder_canals_as_1*). The river network geometries are clipped to the study area by only keeping geometries that intersect with the study area (target: *db_river_networks_strahler_studyarea*).

The next processing step implements a method to obtain linestring geometries that represent the mainstems of the rivers and its tributaries (target: *rivernetworks_merged_per_streamorder*). This is achieved by merging the geometries by a column containing an unique id for each mainstem. The mainstem is defined here as the longest path from the head water to the most distant river mouth (see geometries with the same *levelpath_id* in Figure 7B). As the underlying river network data has no column providing this information, it is necessary to first generate this id column by which we will conduct the merging of the lines (Belitz et al. (2019)[11] made use of the already existing column *levelpathi* from their underlying NHDPlusV2 river network dataset). For doing so, it is now required to first derive a river network for each hydrologic order separately. This is achieved by keeping only geometries with a stream order equal or greater than the specific hydrologic order as described in Background & Summary. The river network of each hydrologic order is then sorted by the column *longpath* in descending order:

```r
river_network_path <-
  river_network %>%
  as_tibble() %>%
  arrange(-longpath) %>%
  select(object_id, nextdownid)
```

The column *longpath* indicates the length of the path from the start node of a linestring geometry to the end node of the most downstream geometry of the river network. Starting with the topmost geometry, all downstream geometries that constitute the longest path to the river mouth are identified by making use of the columns *object_id* and *nextdownid* and the R package *igraph*. This is the start of a loop over the sorted geometries. The column *object_id* provides an unique ID for every linestring geometry and *nextdownid* indicates the *object_id* of the next downstream geometry. Based on this information the function `subcomponent` of the *igraph* package identifies the *object_id*s of all geometries that belong to the longest path. These identified geometries are then removed from the river network for the next iteration of the loop. The subsequent iteration identifies all geometries downstream of the current topmost geometry of the remaining river network. This is repeated until the river network has no geometries left:

```r
longestpaths_list <- list()
i <- 1
while (nrow(river_network_path) > 0) {
  longestpaths_list[[i]] <-
```

```
    river_network_path %>%
    graph.data.frame(directed = TRUE) %>%
    subcomponent(1, mode = "out") %>%
    as.vector() %>%
    slice(river_network_path, .)

  river_network_path <-
    river_network_path %>%
    filter(!(object_id %in% longestpaths_list[[i]]$object_id))

  i <- i + 1
}
```

Subsequently, a column *levelpath_id* is added as a unique ID for all geometries belonging to the same mainstem (Figure 7B). The geometries of the respective river network is then merged based on this column (see difference in linestring geometries between Figure 7B and C). This results in a river network for each hydrologic order separately with a reduced number of geometries as multiple geometries are now summarised into mainstems.

The next step addresses the occurrence of flow splits in the river network (target: *river_networks_treated_brackets*). A flow split or divergence is defined here as junction of linestring geometries with more than one linestring geometry representing out-flowing streams (orange marks in Figure 8). To transfer the methods from Belitz et al. (2019)[11] for the calculation of EU-MOHP[15], it is required to remove minor flow paths that originate from such divergences from the river network for all hydrologic orders except for the first order. A classification of linestring geometries into major and minor flow paths is not directly provided by any column in the underlying river network dataset. Belitz et al. (2019)[11] used the column *divergence* for removing all minor flow paths. In order to reproduce this removal of minor paths from all hydrologic orders greater than 1 without having this information, a more conservative approach is implemented. Therefore, all linestring geometries that intersect with the same other linestring geometry at their start and end node are removed from the river networks. Other minor paths that result from more complex divergences remained in the river network for further calculations (see *feature_id* in Figure 8).

Then, the river networks are sorted by the length of the linestring geometries in descending order and provided with an unique ID for each geometry in the column *feature_id* (target: *rivernetworks_feature_id*; see *feature_id* in Figure 7C).

Surface Water Bodies

Besides the import, there is only one other step in the targets pipeline that preprocesses the surface water bodies (target: *db_inland_waters_strahler* in the file *preprocessing_targets.R*; Figure 5, line 30). A filter is applied to only keep those geometries of surface water bodies that intersect with the river network. In order to assign a hydrologic order to the surface water bodies, the flow order of the river network geometries intersecting them is used, although the endowment of the surface water bodies with a flow order is not relevant for further processing, since the intersection is performed with the already generated hydrological orders.

Coastline

The coastline geometries consist of a vast number of nodes which slows down many geometry operations and calculations. Therefore, many processing steps are parallelised across smaller batches of the coastline data which lead to many helper targets. First, the imported polygon geometries are unioned basin by basin (target: *coastline_unioned*). Contrary to the assumption based on the name “coastline,” the geometry type of the coastline is polygons. Then, all geometries that don’t intersect with the previously derived study area are removed (target: *coastline_filtered*). This is mainly to reduce the large number of geometries contained in the data caused by islands. A buffer of 3000 m is added to the remaining polygon geometries to compensate for inaccuracies of the match between the study area and the coastline contours. The value of 3000 m results from visual inspection of discrepancies between the coastline and the study area boundaries. This is relevant for the second next step. First, the buffered polygon geometries are unioned to a single multipolygon geometry in the PostGIS database for reducing the runtime of calculating the union operation (target: *coastline_buffer_unioned*). Now, the multipolygon geometry that represents the coastline is intersected with the study area as linestring geometry (target: *studyarea_as_coastline*). This intersection ensures that the shoreline lies exactly over the study area. Similarly, the next step determines the parts of the study area that are not coastline meaning where the contour of the coastline touches land instead of the ocean. This is achieved by calculating the difference of the study area and the same coastline geometry as before (target: *coastline_watershed*). All these targets can be found in the *studyarea_targets.R* file (Figure 5, line 34).

EU-MOHP Calculation

After preprocessing all required data layers as described previously, the next and last processing step comprises multiple smaller steps with the final goal to calculate and export the EU-MOHP[15] metrics. This core step is implemented in the target *db_objects_to_grass* which performs the calculation for the hydrologic orders separately in succession. Because the processing is analogous for all hydrologic orders, this step is described only once in general terms. This step also outsources all heavy raster based calculations to a GRASS GIS database. It starts with initiating a GRASS GIS database. Then, the linestring geometries of the river network are read from the PostGIS database. The linestring geometries of the coastline are provided with a column *feature_id* to uniquely identify each geometry. The counter of this *feature_id* starts after the highest *feature_id* of the river network to avoid duplicate values in this column when adding the coastline geometries to the river network in the following. The geometries from the river network and the coastline are merged to also include the coastline in the calculation of the Thiessen watersheds. After combining these geometries, they are written into the GRASS GIS database where they are converted into the raster layer *river_network_raster* (rasterized) using the GRASS command `v.to.rast`:

```
execGRASS(
  cmd = "v.to.rast",
  input = "river_network",
  output = "river_network_raster",
  type = "line",
  use = "attr",
  attribute_column = "feature_id",
  flags = c("overwrite", "d"),
  memory = GRASS_MAX_MEMORY
)
```

This results in a raster layer, where cell values represent the *feature_id* of the linestring geometries rasterized to raster features. The GRASS command `r.neighbors` is used to ensure that mainstems of the river network in the raster layer are not interrupted by cells representing tributaries:

```
execGRASS(
  cmd = "r.neighbors",
  input = "river_network_raster",
  selection = "river_network_raster",
  output = "river_network_raster",
  method = "minimum",
  flags = c("overwrite", "c")
)
```

This command replaces a cell value with the minimum value of its neighboring cells by setting the parameter *method* to *minimum*. As the *feature_id* is added as continuous counter starting at 1 after sorting the river network by the linestring geometry length in descending order, cell values are replaced in favor of the mainstems.

Subsequently, the polygon geometries of the surface water bodies are imported into R from the PostGIS database, written into the GRASS GIS database, rasterized and added to the raster layer *river_network_raster* using the GRASS command `r.patch`.

All further calculations are performed separately for each of the 10 polygon geometries of the study area that has intersecting streams of the respective hydrologic order to avoid unnecessary calculations. After setting the region to the spatial extent of the respective study area polygon, the study area polygon is written into the GRASS GIS database. From this polygon, a raster mask is created to limit all further raster calculations to the study area. Then, the distance from a raster grid cell center to the nearest stream (see DS in Eq. (1), (2) and (3) or Figure 2) is calculated using the GRASS command `r.grow.distance` with

```
execGRASS(
  cmd = "r.grow.distance",
  input = "river_network_raster",
  distance = "river_network_distance_raster",
  value = "river_network_value_raster",
  flags = c("overwrite", "m")
)
```

This command creates the two raster layers *river_network_distance_raster* and *river_network_value_raster*. The former contains the horizontal distance to the nearest linestring geometry of the river network and the coastline, the latter represents the value of the *feature_id* of the nearest raster feature. The raster layer *river_network_value_raster* represents the Thiessen catchments. For deriving the Thiessen divides, this raster layer is converted into a vector layer of polygon geometries. The associated occurrence of dangling polygon outlines is reduced using the GRASS command *v.clean*. Subsequently, the rasterized outlines of these polygons are used as Thiessen divides. To calculate the distance the nearest Thiessen divide with the restriction to not cross a stream (see DD in Eq. (1), (2) and (3) or Figure 2), the GRASS command `r.walk` is used as follows:

```
execGRASS(
  cmd = "r.walk",
  elevation = "river_network_distance_raster",
  friction = "friction",
  output = "thiessen_catchments_distance_raster",
  start_raster = "thiessen_catchments_lines_raster_thin",
  walk_coeff = "1,0,0,0",
  lambda = 1,
  memory = GRASS_MAX_MEMORY,
  flags = c("overwrite")
)
```

Through adjusting the parameters *walk_coeff* and *lambda*, this command calculates the horizontal distance between every cell and the nearest Thiessen divide in the raster layer *thiessen_catchments_lines_raster_thin* while being aware of the defined restriction. This restriction is taken into account by additionally providing the raster layer *friction* that represents friction costs. The *friction* raster layer is created by assigning a value of 1 billion to all non-empty cells of the *river_network_raster*. This value is greater then the maximum possible distance. Thus, for the calculation of the nearest divide by `r.walk` it is now ensured, that crossing a river is not an option leading to a preference of divides that lie on the same side of the stream as the respective cell. The resulting distances are stored in the raster layer *thiessen_catchments_distance_raster*. A discussion on the limitations of this implementation is provided in Technical Validation.

Now, the EU-MOHP[15] measures are calculated using the GRASS command `r.mapcalc` and the two raster layers *river_network_distance_raster* and *thiessen_catchments_distance_raster* containing the cell values for DS and DD respectively. The EU-MOHP[15] measure DSD is calculated according to Eq. (2) with

```
execGRASS(
  cmd = "r.mapcalc",
  expression = glue::glue(
    "{FEATURE_NAMES[1]} = (river_network_distance_raster + thiessen_catchments_distance_raster)"
  ),
  flags = c("overwrite")
)
```

where *`FEATURE_NAMES[1]`* is the raster layer name *divide_stream_distance* for DSD. LP is calculated according to Eq. (1) with

```
execGRASS(
  cmd = "r.mapcalc",
  expression = glue::glue(
    "{FEATURE_NAMES[2]} = round((river_network_distance_raster/{FEATURE_NAMES[1]})*10000)"
  ),
  flags = c("overwrite")
)
```

where *`FEATURE_NAMES[2]`* is the raster layer name *lateral_position* for LP. In order to be able to write the raster layer as integer data type with two decimals, the result of the division is multiplied by a factor of 10.000 and rounded. The data type integer reduces storage space compared with float. For the same reason, the previously calculated raster layer *divide_stream_distance* is rounded, too.

As last measure, SD is calculated according to Eq. (3) with

```
execGRASS(
  cmd = "r.mapcalc",
  expression = glue::glue(
    "{FEATURE_NAMES[3]} = round(river_network_distance_raster)"
  ),
  flags = c("overwrite")
)
```

where `FEATURE_NAMES[3]` is the raster layer name *stream_distance* for SD. Its calculation is simply performed by rounding the raster layer "river_network_distance_raster."

Lastly, the resulting raster layers for LP, DSD and SD are exported from the GRASS GIS database. Therefore, the directory *output_data* with the sub-directories *divide_stream_distance*, *lateral_position* and *stream_distance* is created. The raster layers are written into these sub-directories in the GeoTIFF (*.tif*) file format.

Data Descriptor

To ensure reproducibility of the data descriptor itself, it is generated as part of the targets pipeline (target: *data_descriptor*). Also all tables and some figures are created from within the pipeline (see all targets in file *visualizations_data_descriptor_targets*). The data descriptor is written using the R package *rmarkdown* in the file *main.Rmd*. From there it is rendered as LaTeX (.tex) and PDF (.pdf) file format using the *knitr* package and exported to *data_descriptor/tex/*[23,24].

Hardware

The computations to generate the presented dataset were performed on a DELL PowerEdge C4140 Server with an Intel Xeon Gold 6240R CPU and 384 GB installed RAM. The installed operation system is Microsoft Windows Server 2019 Standard, version 10.0.17763 Build 17763. The total runtime of the pipeline as well as of individual targets is summarised in Table 1.

Data Records

The presented EU-MOHP v013.1.0 dataset[15] is available on the Hydroshare cloud-platform at https://doi.org/10.4211/hs.0f02af18e5344ae7a65dfa7fe1444f34. The dataset represents gridded spatial maps and is divided into multiple GeoTIFF files with a *.tif* file extension. Each file represents data on one of the three EU-MOHP[15] measures – LP, DSD, and SD – for one hydrologic order for a different study area polygon (spatial coverage). The file names are structured according to the file naming scheme "*mohp_europe_<region name for spatial coverage>_<abbreviation of the EU-MOHP measure>_<hydrologic order>_<spatial resolution>.tif*." The placeholders including "<" and ">" can be theoretically replaced by any combination of the values summarized in Table 2. But not all study area polygons have a river network for each of the 9, 8, 7, 6, 5, 4, 3, 2, 1 hydrologic orders. For example, the study area polygon for the island of Sardinia only has rivers up to a maximum streamorder of 6 and therefore only a maximum hydrologic order of 6. This means that there are no GeoTIFF files for Sardinia for hydrologic orders 7 - 9. Therefore, the total number of files is $n_{measures} \cdot \sum_{i=1}^{n_{hydrologic\,orders}} n_{study\,area\,polygons,\,i} = 3 \cdot \sum_{i=1}^{9} n_{study\,area\,polygons,\,i} = 189$.

The GeoTIFF files derived in section EU-MOHP Calculation, were uploaded to Hydroshare as separately compressed files with the file extension *.7z* using the free and open-source file archiver program 7-Zip. Each *.7z* file corresponds to one *.tif* file.

On Hydroshare you have the option to either select all *.7z* files and download them as a zipped bagit archive or download a custom selection of files if your are only interested in a specific region (area of interest) or specific hydrologic orders. For creating a user defined selection you can use the search bar to filter the files for a spatial coverage or a hydrologic order as described on Hydroshare website of this dataset. If you want to check more precisely whether your area of interest is covered by this dataset at all or which files are relevant, please see the interactive map on Github (https://mxnl.github.io/macro_mohp_feature/).

The presented EU-MOHP dataset[15] has version v013.1.0 The version is generated as a composition of the "EU-Hydro – River Network Database"[12] version (v013) and a major and a minor version number (1.0) that are related to the methods of this dataset.

Technical Validation

The EU-MOHP dataset[15] consists of calculated values based on a hydrological concept and therefore cannot be validated by observations or measurements. As a first approximation, the statistical summary is used. Table 3 provides the mean, median, minimum and maximum value of the three measures across all hydrologic orders. ??? Continue here. Figure 9

As the generation of this dataset is based on the "EU-Hydro – River Network Database," its accuracy and validity depends strongly on the quality of this underlying dataset. The "EU-Hydro – River Network Database"[12] has been generated through a combination of photo interpretation of very high resolution imagery and drainage modelling based on the EU DEM with 25 m resolution. According to our search, there is no comprehensive quality assessment or validation for the used version v013. From visual inspection, the following error becomes evident. A confusion of the classification of the linestring geometries into canals, ditches and rivers occurs frequently. An example for such confusion is shown in Figure 10. Here, some relatively straight shaped linestring geometries are classified as river (value BH140 in column *dfdd*), whereas meandering geometries are classified as canal (value BH020 in column *dfdd*). Other errors might be introduced through the limitation of the spatial resolution of the photo imagery and the EU DEM. This potentially affects the detection and of smaller rivers, canals and ditches. Nevertheless, the "EU-Hydro – River Network Database"[12] is a valuable dataset that made this dataset possible. It might also be further improved in the future.

The accuracy of this dataset may also be reduced near the boundaries that run over land rather than along the coast. This includes the regions that are close to the borders in the South and East of Turkey, in the East of continental Europe and in the East of Finland. Here, the boundaries of the underlying dataset, and thus this dataset, follow administrative borders instead of basin outlines. Therefore, calculated distances to the nearest stream in these regions may be inaccurate because another stream not included in the dataset could be closer to a raster cell center. The width of these potentially inaccurate regions along the margins increases with hydrologic order. Because the stream locations of adjacent stream networks are unknown, it is not possible to delineate this region or quantify its width. To address this issue when applying this dataset to such a region, a conservative option would be to truncate or mask these regions by shifting the corresponding boundaries inward by the maximum value in the stream distance map of the respective hydrologic order.

Another inaccuracy is introduced by the method to calculate DD. This inaccuracy only affects a narrow area near headwaters. As described previously, the GRASS GIS command `r.walk` is used to calculate DD. The command `r.walk` originally aims at a different purpose than the one it is used for here. It calculates the cumulative costs for moving between two geographic locations based on topographic map and a map that represents friction costs. Because of the applied setting of the command parameters, it calculates the horizontal distance from a cell to the nearest Thiessen divide while preferring a path without crossing a stream. This behavior is usually achieved everywhere except for areas near headwaters where "walking" around the stream becomes an option. To illustrate this, following case is considered. If a linestring geometry representing a stream is closer to one side of the Thiessen divide than to the other side, `r.walk` calculates an incorrect distance around the start of the linestring as it cheaper to "walk" around the stream than walking a straight path from the more distant but correct side of the Thiessen divide. Thus, the straight path from this mistakenly nearest side of the Thiessen divide crosses the stream. Whereas the required and correct behaviour would be to calculate the distance as the length of a straight line to the Thiessen divide that does not cross the stream (Figure 11).

The method for calculating DD also causes missing (NA) values for cells that are located in lakes. This only affects the DSD raster maps ("*<abbreviation of the EU-MOHP measure> = dsd*").

As stated below, we encourage readers and users of this dataset to report errors in the methods or the code in the mentioned Github repository.

Usage Notes

As described previously, the presented dataset can be used as features in any machine learning task in the domain of hydrology and hydrogeology across many scales. Due to the widely used GeoTIFF file format, the dataset can be processed and visualized through any GIS Software. For the sake of reproducibility in science, it is recommended to use programming languages instead of point-and-click software such as ArcGIS or QGIS. The programming languages R or Python provide a variety of tools to import, process and visualize GeoTIFF data but also offer flexibility from a machine learning perspective. The R packages *raster* and *stars* cover most common operations on raster data[25, 26]. To crop the GeoTIFF files to your custom study area or area of interest, the function `st_crop()` from the *stars* package offers a fast cropping without having to read the large GeoTIFF files into memory. To do so, it's required to read in the GeoTIFF files as *stars_proxy* objects with `read_stars(<path to GeoTIFF file>, proxy = TRUE)` before applying `st_crop()`. For a fast raster cell value extraction based on polygons, the R package *exactextractr* (https://github.com/isciences/exactextractr) is recommended.

To decompress the *.7z* files after download you can use the free and open-source file archiver program 7-Zip.

The raster cell values of all GeoTIFF files are stored as integers in the *INT32* data type to reduce storage size. Cell values of files that represent LP ("*<abbreviation of the EU-MOHP measure> = lp*") must be divided by 100 to obtain percentages with two decimals. The cell values of all other files represent a distance in meters and can be used as is. All files are stored using the coordinate reference system (CRS) ETRS89-extended / LAEA Europe with the EPSG code 3035.

Code availability

All processing and analysis is conducted using free open-source software and free data. The code[27] can be found on Hydroshare (https://doi.org/10.4211/hs.bfdfd782ffc74c42b0347690ae543961) as a static code repository. The actively developed code can be found in on Github (https://github.com/MxNl/macro_mohp_feature). We encourage interested users of this dataset to report errors in the code or to give hints on further methodological or programming improvements through opening an issue in the Github repository.

The used software comprises R (version 4.0.3), PostgreSQL (version 13) database with the PostGIS (version 3.1.0) extension and GRASS GIS (version 7.8.5-2). R package dependencies are managed with the *renv* package. The versions of used R packages can be found in the *renv.lock* file.

To reproduce this dataset, the subsequent steps are required. They have been tested under Windows as operating system (see Hardware), therefore deviations under Linux or MacOS are possible:

1. Install the R language, PostgreSQL, PostGIS and GRASS GIS in their previously described versions. Furthermore, install the latest version of RStudio. RStudio is a free integrated development environment for R.

2. Create a PostgreSQL database with the name "postgis" or, alternatively, choose a different name and change the variable *database_name* in the *config.yml* file later. Independently from the database name, change the setting of the PostgreSQL database to not request a password for connection.

3. Download the project repository containing all required code and scripts from above mentioned static code repository.

4. Download the required input data "EU-Hydro – River Network Database"[12] and "EU-Hydro – Coastline"[21] from the links below and store it in the directory *input_data* as described in Underlying Dataset and Directory & File Structure to make it match the file structure of the *input_data* (Figure 5, line 4 - 7). For downloading the data a free user account is required. Alternatively, if you want to keep the data at another directory, e.g. on a remote server, you need to change the file paths in the file *constants.R*.

5. Navigate to the project directory and open the file *macro_mohp_feature.Rproj* with RStudio.

6. Install the package *renv* by running following command in the console

```
install.packages("renv")
```

7. Install all package dependencies with the subsequent line (Note that under Linux and MacOS some R-packages have system dependencies, such as the package *sf*, which depends on `libgeos-dev`, among others. Please consult the respective documentation when facing an issue.)

```
renv::restore()
```

8. Before running the pipeline on the full spatial coverage of the EEA39 countries, we recommend to test the pipeline with the smaller test study area by setting the variable *area* in the file *config.yml* to "test." The runtime will be around 20 min. The content of the *config.yml* should look like this (Note the empty line in line 6):

```
area: test
cellsize: 30
database_name: postgis
exclude_scandinavian_basins: FALSE
simplify_polygons: FALSE
data_descriptor_only: FALSE
parallel: TRUE
```

If the pipeline works in "test" mode, you can change the variable *area* back to "europe."

9. Start the processing pipeline by running the file *run_pipeline.R* with

```
source("run_pipeline.R")
```

10. If you encounter any problems, please contact the corresponding author or preferably open a Github issue. Errors can probably be caused by incorrect directories and file paths. If the available memory is insufficient, one option is to run the pipeline sequentially rather than in parallel. To do this, change the variable *parallel* in the file *config.yml* from `TRUE` to `FALSE`.

11. To reproduce the data descriptor itself, you can execute the pipeline after a successful run by setting the variable *data_descriptor_only* in the file *config.yml* to "TRUE."

The required underlying datasets “EU-Hydro – River Network Database”[12] version v013 can be downloaded from the Copernicus Land Monitoring Service (https://land.copernicus.eu/imagery-in-situ/eu-hydro/eu-hydro-river-network-database?tab=download) as well as the “EU-Hydro – Coastline”[21] version v013 (https://land.copernicus.eu/imagery-in-situ/eu-hydro/eu-hydro-coastline?tab=download). In order to maximize and simplify reproducibility, we currently plan to set up a docker container. For availability updates, please visit the mentioned Github repository. For transferring the presented methods to another custom region, equivalent input data to Table 4 is required.

Acknowledgements

The generation of this dataset would not have been possible without all the free open-source packages for R. Therefore, a special thanks goes to their developers, especially to Will Landau who quickly provided answers and solutions regarding the *targets* package. The developers of all used packages can be found in the references of the respective package. We were also grateful for discussions and hint by the colleagues at BGR.

Author contributions statement

M.N. was involved in all phases and steps of the generation of this dataset including investigations and visualizations. M.M. contributed to software development in R and PostGIS and set up the docker container. M.M. also contributed to the methodology and validation. S.B. contributed to the conceptualization of the dataset, but also led the supervision, project administration and funding acquisition. All authors reviewed and edited the manuscript.

Competing interests

The authors declare no competing interests related to the presented dataset, its generation or data descriptor.

Figures & Tables

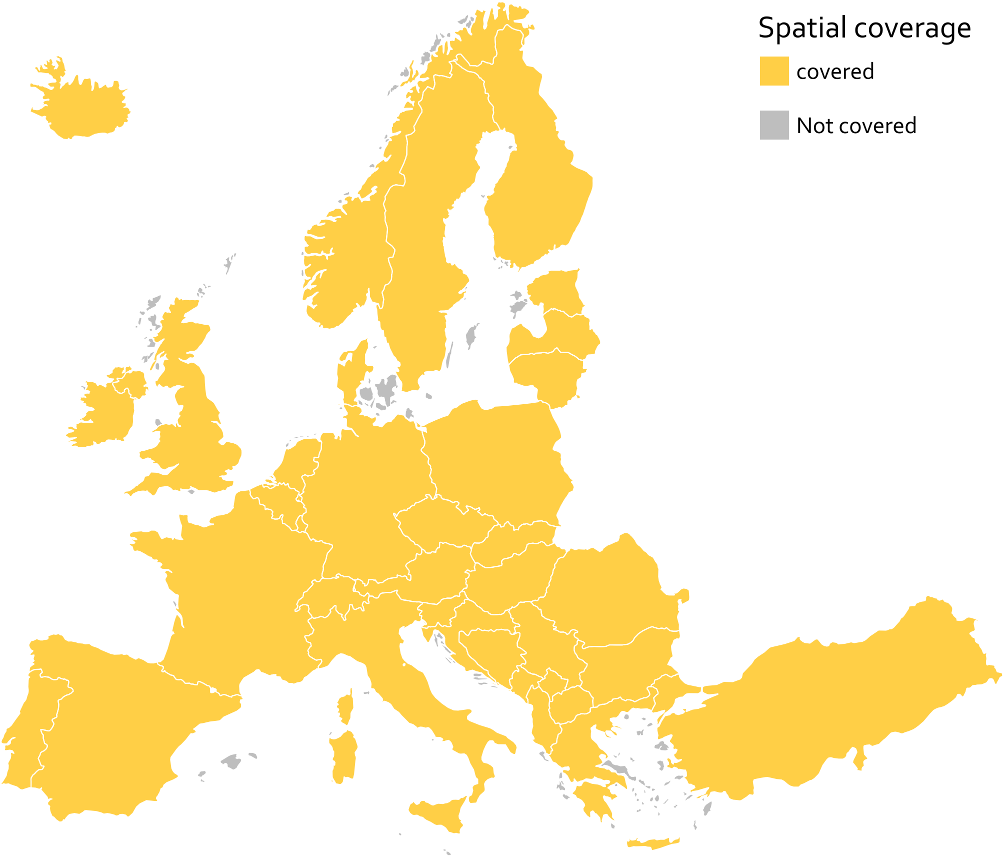

Figure 1. Spatial coverage of the dataset which is determined by the study area data layer.

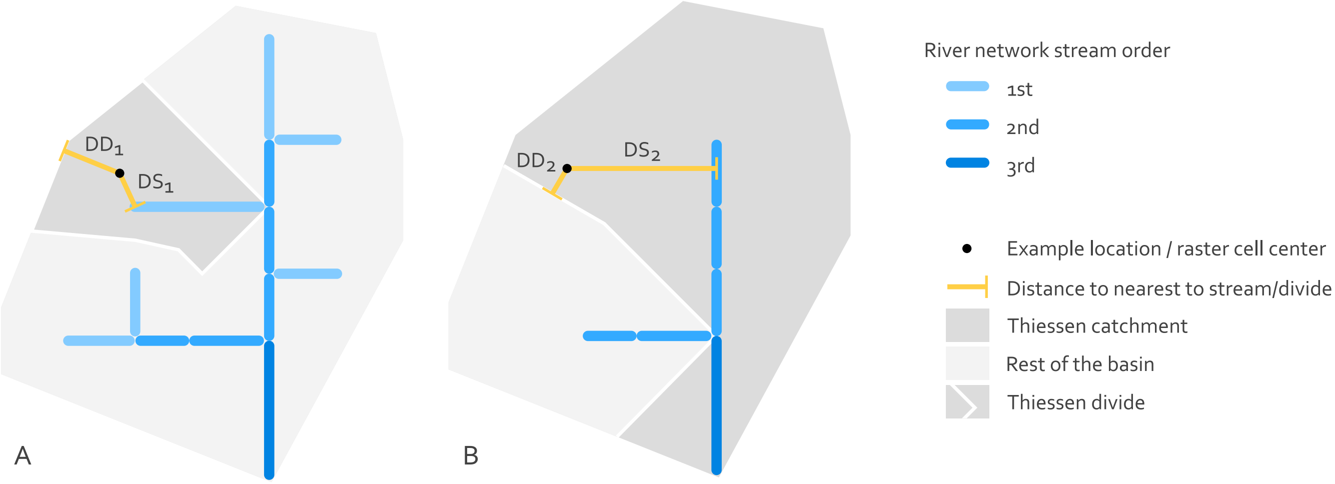
DD_1
DS_1
A
DD_2
DS_2
B
River network stream order
1st
2nd
3rd
Example location / raster cell center
Distance to nearest to stream/divide
Thiessen catchment
Rest of the basin
Thiessen divide

Figure 2. Schematic representation of MOHP measures using two examples for the hydrologic orders 1 (A) and 2 (B). DS is the horizontal distance to the nearest stream and DD is the horizontal distance to the nearest Thiessen divide under the condition that the divide is on the same side of the stream as the raster cell center (black point).

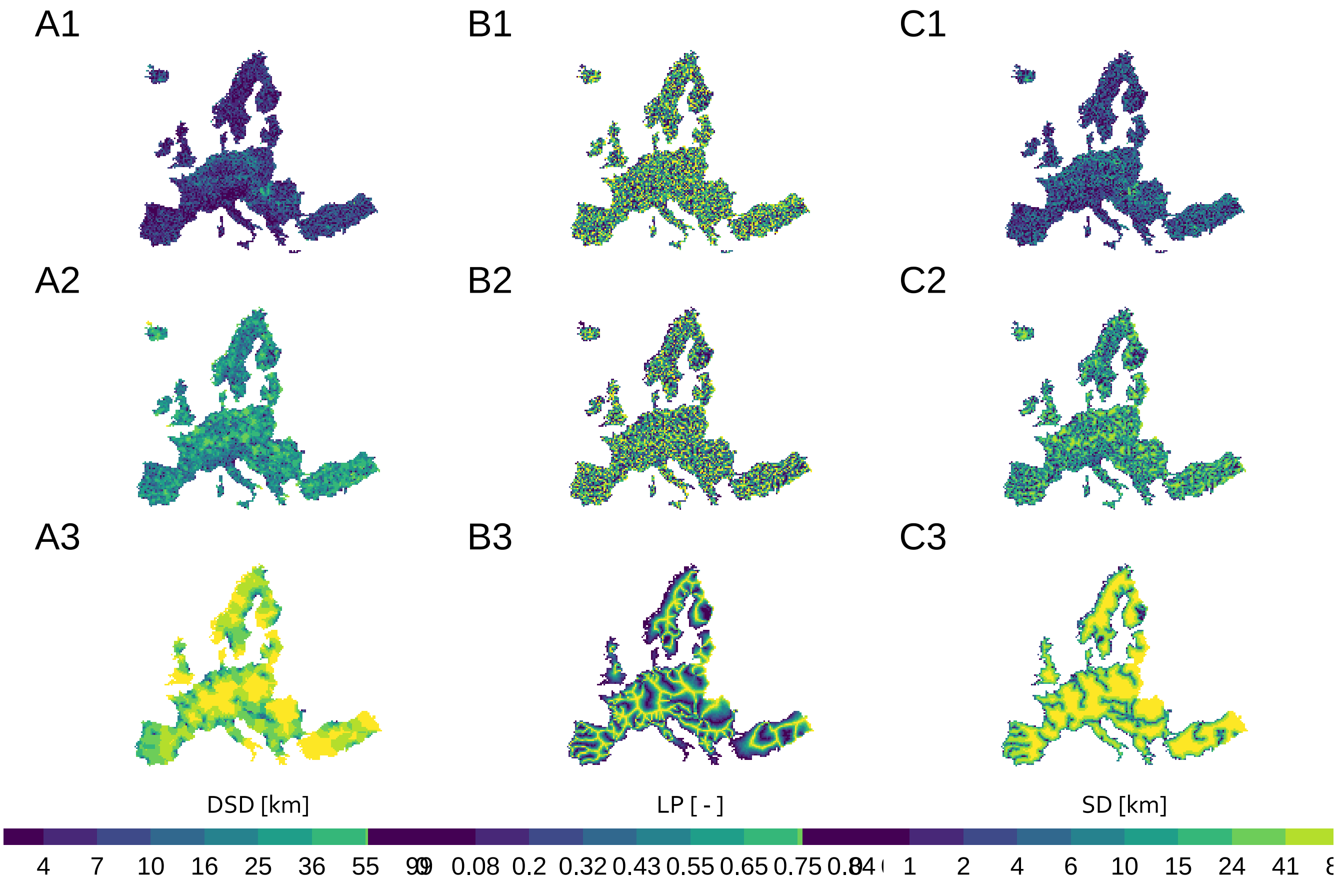

Figure 3. Resulting maps of the three EU-MOHP measures divide stream distance (A), lateral position (B), and stream distance (C) in the columns exemplary for the three hydrologic orders 3 (1), 5 (2) and 7 (3) in the rows. Note the binned colour scale with breaks based on quantiles.

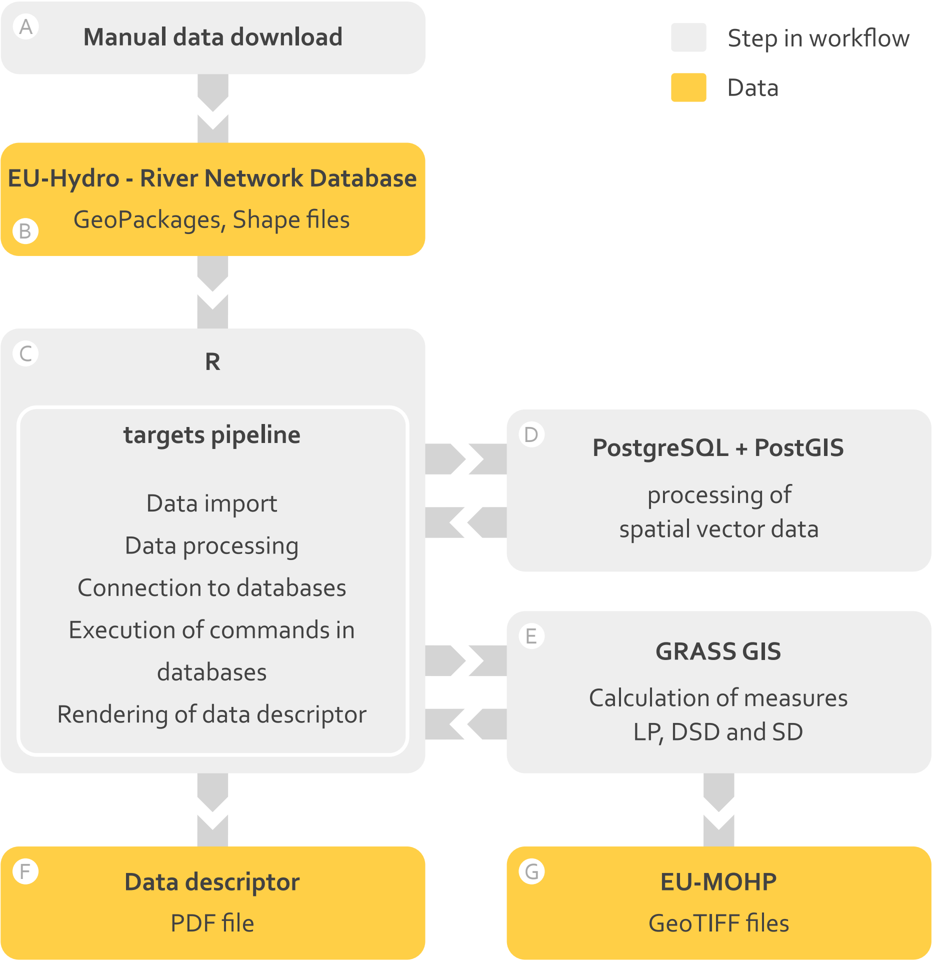

Figure 4. Workflow of the data processing in different software.

```
.
+-- config.yml
+-- grassdata
+-- input_data
|   +-- data
|   +-- EUHYDRO_Coastline_EEA39_v013
|   \-- studyarea_test
+-- macro_mohp_feature.Rproj
+-- output_data
|   +-- divide_stream_distance
|   +-- lateral_position
|   \-- stream_distance
+-- R
|   +-- constants.R
|   +-- database_functions.R
|   +-- export_functions.R
|   +-- grass_functions.R
|   +-- import_functions.R
|   +-- postgis_functions.R
|   \-- preprocessing_functions.R
+-- renv
|   +-- activate.R
|   +-- library
|   +-- local
|   +-- settings.dcf
|   \-- staging
+-- renv.lock
+-- run_pipeline.R
+-- targets
|   +-- export_targets.R
|   +-- import_targets.R
|   +-- mohpcalculation_targets.R
|   +-- preprocessing_targets.R
|   \-- studyarea_targets.R
+-- _targets
|   +-- meta
|   \-- objects
\-- _targets.R
```

Figure 5. Directory tree of the project directory; only relevant subdirectories and files are listed here.

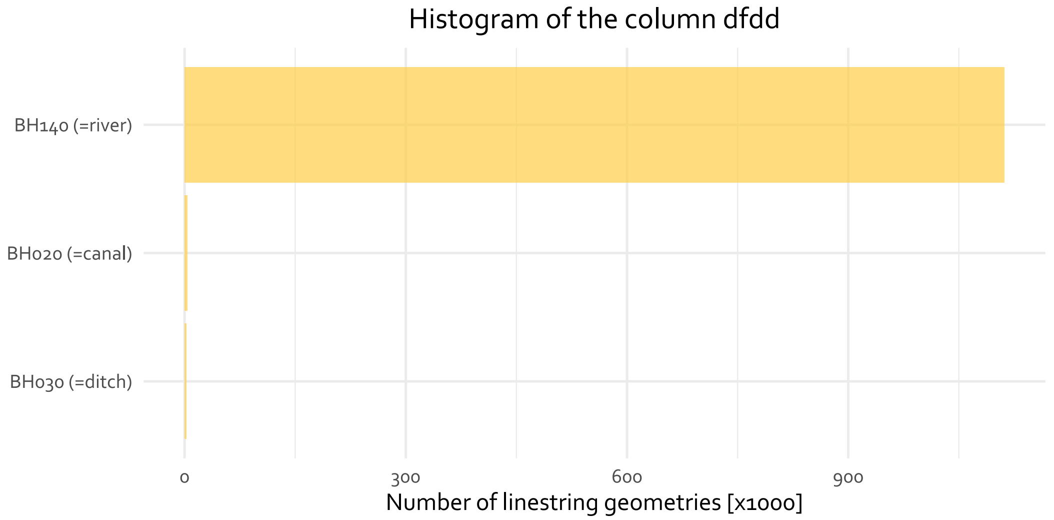

Figure 6. Distribution of the values in the attribute column dfdd.

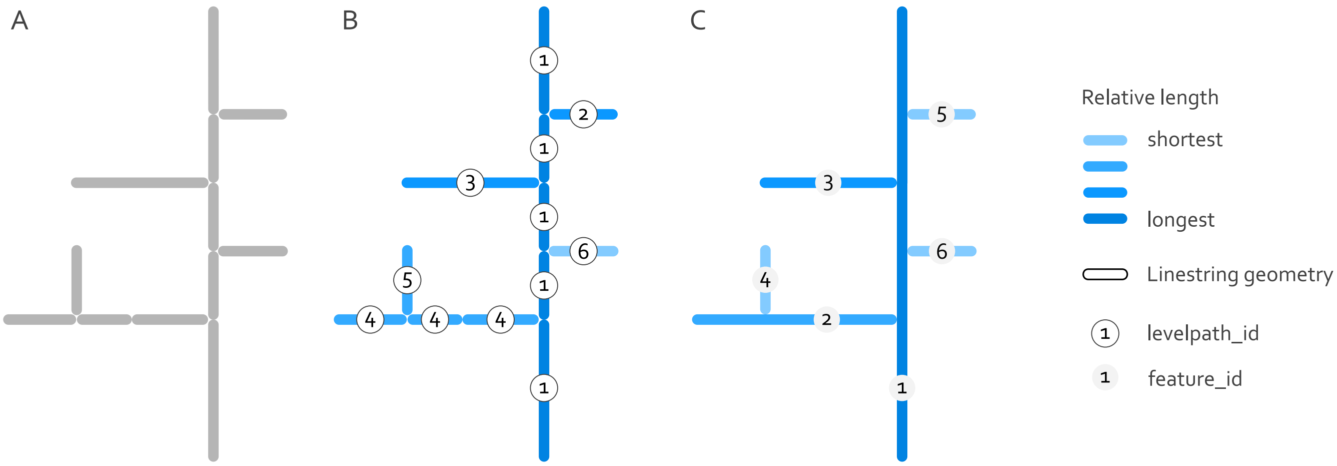

Figure 7. Schematic representation of the river network and its linestring geometries after import (A), after the identification of mainstems including the column levelpath_id (B) and after merging the linestring geometries by this column and adding a feature_id column (C).

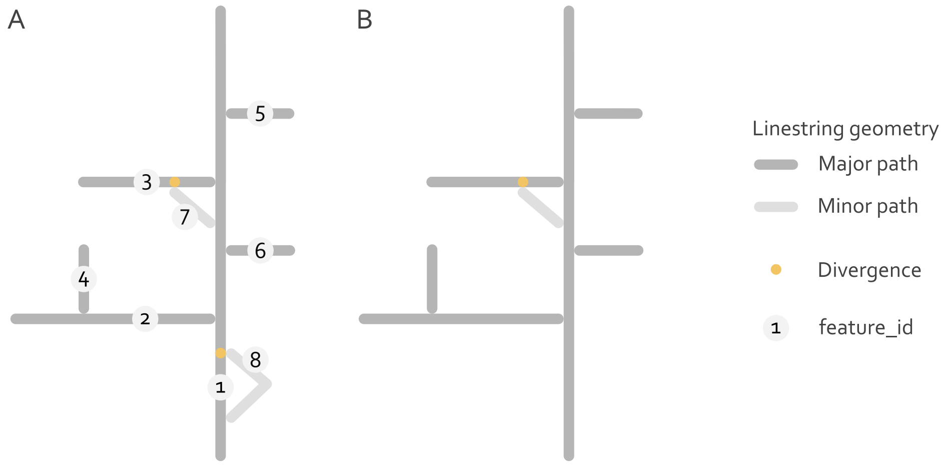

Figure 8. Schematic representation of the river network and its linestring geometries including divergences before (A) and after (B) the removal of minor paths under the condition that they intersect with the same linestring geometry at their start and end node. The linestring geometry with the feature_id = 8 has been removed from the river network in B, because it intersects the linestring geometry with feature_id = 1 at the start and end node. Whereas linestring geometry with feature_id = 7 remains in the river network, because it intersects with two different linestring geometries at its start and end node.

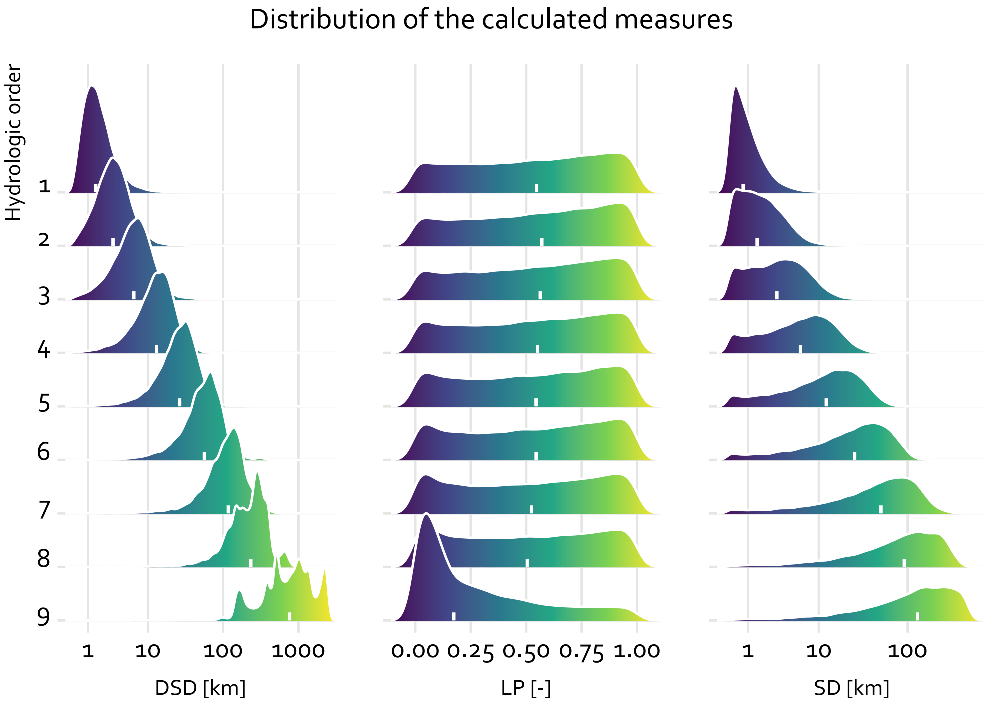

Figure 9. Ridgelines showing the distribution of the three measures DSD, LP and SD for all nine hydrologic orders.

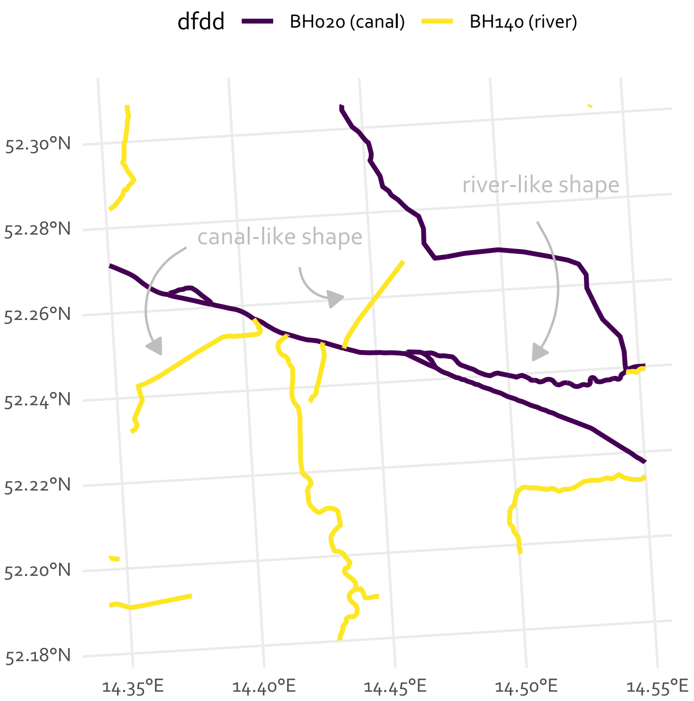

Figure 10. Example of the river network data showing the confusion between the values BH140 (river), BH020 (canal) and BH030 (ditch) of the attribute column dfdd.

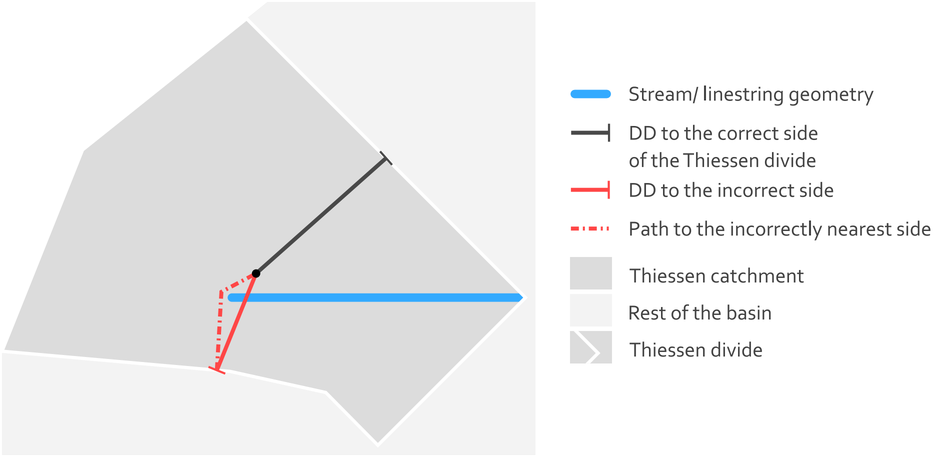
Stream/ linestring geometry
DD to the correct side of the Thiessen divide
DD to the incorrect side
Path to the incorrectly nearest side
Thiessen catchment
Rest of the basin
Thiessen divide

Figure 11. Schematic example showing the source of inaccurate of DD in areas near headwaters caused by the applied method to calculate DD. The red distance as DD is incorrect, because it crosses the stream and therefore does not fulfill the defined condition. The correct DD would be the dark grey distance. The path to the correct side is equal to the correct DD (dark grey solid line) and therefore not drawn on the schematic map.

Table 1. Overview of the runtime and data size of all targets or processing steps in descending order.

	Runtime				Data size
Target name	Seconds	Minutes	Hours	Days	Mb
db_objects_to_grass	1199656.6	19994.3	333.2	13.9	0.0
rivernetworks_merged_per_streamorder	155852.7	2597.5	43.3	1.8	2002.6
eumohp_files_compression	127353.4	2122.6	35.4	1.5	0.0
db_inland_waters_strahler	1741.6	29.0	0.5	0.0	0.0
river_basins_unioned	1296.2	21.6	0.4	0.0	87.8
coastline_unioned	653.4	10.9	0.2	0.0	84.2
coastline_buffer	586.0	9.8	0.2	0.0	7.7
river_basins_subset_union_in_db	451.0	7.5	0.1	0.0	76.0
coastline_filtered	363.3	6.1	0.1	0.0	76.5
river_networks	235.3	3.9	0.1	0.0	1112.0
db_river_networks_merged_per_streamorder	835.3	13.9	0.2	0.0	0.0
db_river_networks_clean	129.9	2.2	0.0	0.0	0.0
inland_waters	101.4	1.7	0.0	0.0	183.0
db_river_networks_strahler_studyarea	41.3	0.7	0.0	0.0	0.0
river_networks_clean	38.0	0.6	0.0	0.0	1051.0
db_inland_waters	25.6	0.4	0.0	0.0	0.0
river_networks_non_dry_selected_streamtypes	24.9	0.4	0.0	0.0	1053.7
rivernetworks_feature_id	126.3	2.1	0.0	0.0	2070.1
river_basins	20.3	0.3	0.0	0.0	147.3
river_basins_subset	8.7	0.1	0.0	0.0	84.0
streamorders	5.7	0.1	0.0	0.0	0.0
coastline_grouped	5.4	0.1	0.0	0.0	92.8
quantile_breaks	5.1	0.1	0.0	0.0	0.0
config	4.8	0.1	0.0	0.0	0.0
filepath_coastline	4.8	0.1	0.0	0.0	0.0
studyarea_as_coastline	4.5	0.1	0.0	0.0	28.1
directory_river_networks	4.5	0.1	0.0	0.0	0.0
coastline_watershed	4.2	0.1	0.0	0.0	29.8
db_selected_studyarea	3.5	0.1	0.0	0.0	0.0
coastline_buffer_unioned	3.1	0.1	0.0	0.0	7.5
selected_studyarea	2.0	0.0	0.0	0.0	29.8
major_path_ids	1.9	0.0	0.0	0.0	4.4
bracket_start_ids	1.7	0.0	0.0	0.0	0.0
river_basins_region_name	1.7	0.0	0.0	0.0	29.8
distinct_streamorders_in_riverbasins	1.1	0.0	0.0	0.0	0.0
river_networks_imputed_streamorder_canals_as_1	0.1	0.0	0.0	0.0	1053.7
rivernetworks_merged_per_streamorder_grouped	0.0	0.0	0.0	0.0	2003.1
river_networks_files	0.0	0.0	0.0	0.0	0.0
river_basins_grouped	0.0	0.0	0.0	0.0	147.4
river_basins_files	0.0	0.0	0.0	0.0	0.0
river_basin_names	0.0	0.0	0.0	0.0	0.0
coastline_regrouped	0.0	0.0	0.0	0.0	84.2
river_networks_clip	0.0	0.0	0.0	0.0	1112.0
Total	**1489595.3**	**24826.6**	**413.7**	**17.2**	**12658.5**

Table 2. Overview of the output file naming scheme and its placeholder values. Files for any combination of the placeholder values exists except for those study area polygons (<region name for spatial coverage>) that have no streams for certain hydrologic orders. The values are inserted for the respective placeholder in "mohp_europe_<region name for spatial coverage>_<abbreviation of the EU-MOHP measure>_<hydrologic order>_<spatial resolution>.tif". For example, selecting the first value of each placeholder results in the file name "mohp_europe_europemainland_dsd_hydrologicorder1_30m.tif". The spatial coverage of the values for "<region name for spatial coverage>" is shown in the mentioned interactive map in the Github repository.

Placeholder in output file name	Value	Description
<region name for spatial coverage>	europemainland	Raster data covers the contiguous land area of continental Europe, ...
	finland-norway-sweden	...the Scandinavian countries Finland, Norway and Sweden
	france	...Corsica
	greece	...Creta
	iceland	...Iceland
	italy1	...Sicily
	italy2	...Sardinia
	turkey	...Turkey
	unitedkingdom	...United Kingdom
	unitedkingdom-ireland	Ireland and Northern Ireland
<abbreviation of the EU-MOHP measure>	dsd	Divide stream distance
	lp	Lateral Position
	sd	Stream distance
<hydrologic order>	streamorder1	Hydrologic order
	streamorder2	
	streamorder3	
	streamorder4	
	streamorder5	
	streamorder6	
	streamorder7	
	streamorder8	
	streamorder9	
<spatial resolution>	30m	Spatial resolution

Table 3. Statistical summary of the calculated measures DSD, LP and SD across all hydrologic orders.

	DSD				LP				SD			
Hydrologic order	min	median	mean	max	min	median	mean	max	min	median	mean	max
1	0.00	1.57	2.00	55.70	0	0.55	0.53	1	0	0.73	1.08	35.18
2	0.00	3.19	3.73	55.70	0	0.57	0.54	1	0	1.56	2.08	35.18
3	0.00	6.42	7.24	76.62	0	0.56	0.54	1	0	3.12	3.98	45.77
4	0.00	13.05	14.48	187.04	0	0.55	0.53	1	0	6.11	7.67	48.95
5	0.03	26.57	28.77	207.69	0	0.54	0.52	1	0	12.14	14.96	97.97
6	0.00	56.53	61.49	406.36	0	0.54	0.52	1	0	25.42	30.99	189.69
7	1.06	117.40	128.75	542.97	0	0.52	0.51	1	0	50.23	62.27	329.87
8	1.27	233.65	247.43	861.56	0	0.51	0.50	1	0	91.81	112.69	416.79
9	4.36	764.12	920.84	2526.04	0	0.17	0.28	1	0	129.17	159.96	561.65

Table 4. Overview of the required input data to reproduce this dataset.

No	Data layer	Data source	Layers in .gpkg files	Geometry type	Description
1	river network	EU-Hydro – River Network Database	Canals_l, Ditches_l, River_Net_l	linestring	representing stream lines of rivers
2	surface water bodies	EU-Hydro – River Network Database	InlandWater	polygon	representing lakes, ponds and wide rivers
3	river basins/ study area	EU-Hydro – River Network Database	_eudem2_basins_h1	linestring	required to set the area for which the EU-MOHP measures are calculated for
4	coastline	EU-Hydro – Coastline	-	linestring	representing the coastline

References

1. Zounemat-Kermani, M. *et al.* Neurocomputing in surface water hydrology and hydraulics: A review of two decades retrospective, current status and future prospects. *J. Hydrol.* **588**, 125085, https://doi.org/10.1016/j.jhydrol.2020.125085 (2020).

2. Sit, M. *et al.* A Comprehensive Review of Deep Learning Applications in Hydrology and Water Resources. preprint, Engineering (2020). https://doi.org/10.31223/OSF.IO/XS36G.

3. DeSimone, L. A., Pope, J. P. & Ransom, K. M. Machine-learning models to map pH and redox conditions in groundwater in a layered aquifer system, Northern Atlantic Coastal Plain, eastern USA. *J. Hydrol. Reg. Stud.* **30**, 100697, https://doi.org/10.1016/j.ejrh.2020.100697 (2020).

4. Knoll, L., Breuer, L. & Bach, M. Large scale prediction of groundwater nitrate concentrations from spatial data using machine learning. *Sci. The Total. Environ.* **668**, 1317–1327, https://doi.org/10.1016/j.scitotenv.2019.03.045 (2019).

5. Knoll, L., Breuer, L. & Bach, M. Nation-wide estimation of groundwater redox conditions and nitrate concentrations through machine learning. *Environ. Res. Lett.* **15**, 064004, https://doi.org/10.1088/1748-9326/ab7d5c (2020).

6. Mueller, J. *et al.* Surrogate Optimization of Deep Neural Networks for Groundwater Predictions. *http://arxiv.org/abs/1908.10947* (2019). ArXiv: 1908.10947.

7. Stackelberg, P. E. *et al.* Machine Learning Predictions of pH in the Glacial Aquifer System, Northern USA. *Groundwater* **59**, 352–368, https://doi.org/10.1111/gwat.13063 (2021).

8. Wang, B., Oldham, C. & Hipsey, M. R. Comparison of Machine Learning Techniques and Variables for Groundwater Dissolved Organic Nitrogen Prediction in an Urban Area. *Procedia Eng.* **154**, 1176–1184, https://doi.org/10.1016/j.proeng.2016.07.527 (2016).

9. Wunsch, A., Liesch, T. & Broda, S. Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). *J. Hydrol.* **567**, 743–758, https://doi.org/10.1016/j.jhydrol.2018.01.045 (2018).

10. Wunsch, A., Liesch, T. & Broda, S. Groundwater Level Forecasting with Artificial Neural Networks: A Comparison of LSTM, CNN and NARX. preprint, Groundwater hydrology/Modelling approaches (2020). https://doi.org/10.5194/hess-2020-552.

11. Belitz, K., Moore, R. B., Arnold, T. L., Sharpe, J. B. & Starn, J. J. Multiorder Hydrologic Position in the Conterminous United States: A Set of Metrics in Support of Groundwater Mapping at Regional and National Scales. *Water Resour. Res.* **55**, 11188–11207, https://doi.org/10.1029/2019WR025908 (2019).

12. EU-Hydro - River Network Database - Copernicus Land Monitoring Service (2021).

13. Degnan, J. R., Lindsey, B. D., Levitt, J. P. & Szabo, Z. The relation of geogenic contaminants to groundwater age, aquifer hydrologic position, water type, and redox conditions in Atlantic and Gulf Coastal Plain aquifers, eastern and south-central USA. *Sci. The Total. Environ.* **723**, 137835, https://doi.org/10.1016/j.scitotenv.2020.137835 (2020).

14. Knierim, K. J., Kingsbury, J. A., Haugh, C. J. & Ransom, K. M. Using Boosted Regression Tree Models to Predict Salinity in Mississippi Embayment Aquifers, Central United States. *JAWRA J. Am. Water Resour. Assoc.* **56**, 1010–1029, https://doi.org/10.1111/1752-1688.12879 (2020).

15. Nölscher, M., Mutz, M. & Broda, S. EU-MOHP v013.1.0 Dataset. *hydroshare* https://doi.org/10.4211/hs.0f02af18e5344ae7a65dfa7fe1444f34, 10.4211/hs.0f02af18e5344ae7a65dfa7fe1444f34 (2021).

16. Johnston, C. M. *et al.* Evaluation of Catchment Delineation Methods for the Medium-Resolution National Hydrography Dataset. Scientific Investigations Report, U.S. Geological Survey (2009).

17. Strahler, A. N. Quantitative analysis of watershed geomorphology. *Eos, Transactions Am. Geophys. Union* **38**, 913–920 (1957).

18. R Core Team. *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, Austria, 2020).

19. Landau, W. M. The targets R package: a dynamic Make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. *J. Open Source Softw.* **6**, 2959 (2021).

20. Ushey, K. *renv: Project Environments* (2021).

21. EU-Hydro - Coastline - Copernicus Land Monitoring Service (2021).

22. Gallaun, H., Dohr, K., Puhm, M., Stumpf, A. & Hugé, J. EU-Hydro - River Net User Guide 1.3 (2019).

23. Allaire, J. J. *et al. rmarkdown: Dynamic Documents for R* (2021).

24. Xie, Y. knitr: A Comprehensive Tool for Reproducible Research in R. In Stodden, V., Leisch, F. & Peng, R. D. (eds.) *Implementing Reproducible Computational Research* (Chapman and Hall/CRC, 2014).

25. Hijmans, R. J. *raster: Geographic Data Analysis and Modeling* (2020).

26. Pebesma, E. *stars: Spatiotemporal Arrays, Raster and Vector Data Cubes* (2021).

27. Nölscher, M., Mutz, M. & Broda, S. EU-MOHP v013.1.0 Code. *hydroshare* https://doi.org/10.4211/hs.bfdfd782ffc74c42b0347690ae543961, 10.4211/hs.bfdfd782ffc74c42b0347690ae543961 (2021).

28. Wickham, H. *et al.* Welcome to the tidyverse. *J. Open Source Softw.* **4**, 1686, https://doi.org/10.21105/joss.01686 (2019).

29. Pebesma, E. Simple Features for R: Standardized Support for Spatial Vector Data. *The R J.* **10**, 439–446, https://doi.org/10.32614/RJ-2018-009 (2018).

30. Fischetti, T. *assertr: Assertive Programming for R Analysis Pipelines* (2021).

31. Francois, R. *bibtex: Bibtex Parser* (2021).

32. Aust, F. *citr: 'RStudio' Add-in to Insert Markdown Citations* (2019).

33. R Special Interest Group on Databases (R-SIG-DB), Wickham, H. & Müller, K. *DBI: R Database Interface* (2021).

34. Chang, W. *extrafont: Tools for using fonts* (2014).

35. Hester, J. & Wickham, H. *fs: Cross-Platform File System Operations Based on 'libuv'* (2020).

36. Vaughan, D. & Dancho, M. *furrr: Apply Mapping Functions in Parallel using Futures* (2021).

37. Hester, J. *glue: Interpreted String Literals* (2020).

38. Müller, K. *here: A Simpler Way to Find Your Files* (2020).

39. Baumgartner, J. & Dinnage, R. *hues: Distinct Colour Palettes Based on 'iwanthue'* (2019).

40. Csardi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal* **Complex Systems**, 1695 (2006).

41. Firke, S. *janitor: Simple Tools for Examining and Cleaning Dirty Data* (2021).

42. Pebesma, E. *lwgeom: Bindings to Selected 'liblwgeom' Functions for Simple Features* (2020).

43. Pedersen, T. L. *patchwork: The Composer of Plots* (2020).

44. McLean, M. W. RefManageR: Import and Manage BibTeX and BibLaTeX References in R. *The J. Open Source Softw.* https://doi.org/10.21105/joss.00338 (2017).

45. Bivand, R., Keitt, T. & Rowlingson, B. *rgdal: Bindings for the 'Geospatial' Data Abstraction Library* (2021).

46. Bivand, R. & Rundel, C. *rgeos: Interface to Geometry Engine - Open Source ('GEOS')* (2020).

47. Bivand, R. *rgrass7: Interface Between GRASS 7 Geographical Information System and R* (2021).

48. Teucher, A. & Russell, K. *rmapshaper: Client for 'mapshaper' for 'Geospatial' Operations* (2020).

49. South, A. *rnaturalearth: World Map Data from Natural Earth* (2017).

50. Wickham, H., Ooms, J. & Müller, K. *RPostgres: 'Rcpp' Interface to 'PostgreSQL'* (2021).

51. Cooley, D. *sfheaders: Converts Between R Objects and Simple Feature Objects* (2020).

52. Qiu, Y. & details, a. o. t. i. s. S. f. A. f. *showtext: Using Fonts More Easily in R Graphs* (2021).

53. Walthert, L. & Müller, K. *styler: Non-Invasive Pretty Printing of R Code* (2021).

54. Landau, W. M. *tarchetypes: Archetypes for Targets* (2021).

55. Tennekes, M. tmap: Thematic Maps in R. *J. Stat. Softw.* **84**, 1–39, https://doi.org/10.18637/jss.v084.i06 (2018).

56. Knoll, L., Breuer, L. & Bach, M. Large scale prediction of groundwater nitrate concentrations from spatial data using machine learning. *Sci. The Total. Environ.* **668**, 1317–1327, 10.1016/j.scitotenv.2019.03.045 (2019).