Multiorder Hydrologic Position in Europe as a Set of Metrics in Support of Machine Learning Based Cross-scale 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

This dataset (EU-MOHP v013.1.0) provides 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" as a relative measure of the position between the stream and the catchment boundary/ watershed, "divide stream distance" as an absolute distance measure that serves as a proxy for the position within the catchment and "stream distance" as an absolute measure of the distance to the nearest stream. These three measures were calculated for several hydrologic orders to reflect different spatial scales. Its spatial extent covers major parts of physiographical Europe or the European Economic Area (EEA39). Although there might be many potential use cases, this dataset serves predominantly as valuable static environmental predictor variable for hydrogeological and hydrological mapping or regionalization tasks using machine learning.

1 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]. In the field of hydrogeology, machine learning has been used successfully for groundwater level prediction and a variety of mapping tasks[2–9]. Since machine learning models – except for hybrid- or physically-based models – are purely based on data with no built-in knowledge of physical processes, it is important to provide as many features (synonyms: predictor variables, 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 subsurface processes such as in hydrogeology, this poses a serious challenge.

The key motivation for this dataset is to provide 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. It complements commonly available datasets such as land-use and land-cover, geological or soil maps and tackles the above mentioned challenge. This dataset is strongly inspired by Belitz et al. (2019)[10] and adapts their ideas and methods to the "EU-Hydro - River Network Database" as underlying river network 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)[10].

In their study, Belitz et al. (2019)[10] also provide the 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[6, 11, 12].

Being a static geophysical catchment attribute, the EU-MOHP dataset can be used as features in any machine learning task in the domain of hydrology and hydrogeology. Because of the calculation based on the different stream orders, 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 comprises the 3 measures

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

for each of the 6 stream orders which leads to $n_{measures} \cdot n_{streamorders} = 18$ 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 2 largest coherent land masses of the EEA39 (Fig. 3).

Conceptually, the three measures LP, DSD and SD of EU-MOHP are based on the idea that the location in hydrologic systems matters[10]. 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 divide/catchment boundary. Thiessen divides were used as catchment boundaries instead of divides that are generated from digital elevation models. 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[10]. 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[13]. In other words, a Thiessen divide contains all points with equal distance to the two nearest streams.

The overall concept also includes the spatial scale that the role of different hydrologic processes can depend upon. This issue is addressed by calculating the EU-MOHP measures for different hydrologic orders which are derived from the stream orders of the river network. For a specific hydrologic order i, only streams with a stream order $>= i$ were used (e.g. for stream order 2, all streams with stream order 2, 3, 4 and greater, compare Fig. 5A and B). Here, the stream orders are defined according to[14] where all streams without tributaries are assigned to the first stream order (headwater streams). 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. Therefore, each hydrologic order reflects a different scale.

Based on the river network and the Thiessen divides, the EU-MOHP measures (LP, DSD, SD) can be calculated as follows:

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

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

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

where i is the hydrologic order, DS is the horizontal distance from a point to the nearest stream (Fig. 5) 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: the line that connects the nearest Thiessen divide and the point must not cross a stream (Fig. 5). This condition arises from the fact that water flows towards the topographically lowest points namely the streams.

Examples of the generated EU-MOHP v013.1.0 maps are shown in Fig. 1 for the two hydrologic orders 3 and 4.

Methods

Most processing and calculation steps were done in the R programming language (Fig. 2C)[15]. Due to the memory size of this dataset as well as for the sake of computational speed, a PostgreSQL database with PostGIS extension was used for some processing steps of vector data and a GRASS GIS database was used for all final raster based calculations of the EU-MOHP metrics (Fig. 2D and E). For reproducibility and programming reasons, all processing steps including the databases were tracked and executed through a data processing pipeline using the targets package in R (Fig. 2)[16]. 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.

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 were 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 and file structure

The directory and file structure of the project folder containing all code and files to generate this dataset is summarized in Fig. 4 as tree. Files and directories that are not relevant for describing the methods are not shown here. The file *config.yml* (line !!1) contains definitions of variables that are meant to be changed by a user before running the script. The most relevant variable is *cellsize* which sets the raster cell width of the resulting EU-MOHP gridded maps. Another important variable is *area* to switch

between a test study area and the complete study area for all EEA39. The test study area reduces the calculation time for pipeline testing purposes. The folder *grassdata* (line !!2) contains the GRASS GIS databases. The file *macro_mohp_feature.Rproj* (line !!3) is the R project file. *output_data* (line !!5) contains sub-directories where the final EU-MOHP gridded maps are written to. *R* (line 9) contains R scripts where custom functions and constants are defined. *renv* (line !!19) and the file *renv.lock* (line !!25) are related to the R package renv that tracks package dependencies and versions. The R script *run_pipeline.R* contains code to execute the targets pipeline that does all the data processing and calculations. *targets* (line !!27) 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 !!39) sets up the targets pipeline and loads all dependencies.

The resulting EU-MOHP metrics (Fig. 2G) are written to the sub-directories *divide_stream_distance*, *lateral_position* and *stream_distance* of *output_data* as gridded maps in the GeoTiff (.tif) file format. The data descriptor was written to *output_data* in the LaTeX (.tex) and PDF (.pdf) file format.

Underlying dataset

The generation of this dataset is based on two basic data products, the "EU-Hydro – River Network Database" version v013 and "EU-Hydro – Coastline" version v013 with the advantage that data dependencies are low[17,18]. Therefore, it is possible to transfer the methodology to other regions with only little effort. The Tab. 1 provides an overview of the input data layers necessary for calculating the dataset.

The "EU-Hydro – River Network Database" as well as the "EU-Hydro – Coastline" were manually downloaded from the Copernicus - Land Monitoring Service website (Fig. 2A) in GeoPackage (.gpkg) and in Shapefile (.shp) file format, respectively (Fig. 2B)[17,17]. The river network data is split into 2 .gpkg files for each of 35 the major river basins. All unzipped files have a total size of approximately 14GB. The single .shp file containing the coastline has a size of 288MB.

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* (Fig. 4, line !!29). 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"[19]. 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"[19]. The layer *Rivers_l* contains rivers, which are defined as "a naturally flowing watercourse"[19].

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."[19]. In this step, all geometries with an area smaller than $4 \cdot \text{cellsize} = 3600m^2$, where cellsize is the area of raster cells, are removed to exclude geometries that only have a negligible impact on the result. The polygon geometries of wider river parts were not included in the surface water bodies.

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* (Fig. 4, line !!32) and contains polygon geometries of all sub-basins of a 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 metrics will be calculated.

The fourth required data layer is the coastline. It is imported with the target *coastline_grouped* in *studyarea_targets.R* (Fig. 4, line !!32).

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

1.1 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 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* (Fig. 4, line !!32). After the previously mentioned import, the sub-basins were unioned basin-wise (target: *river_basins_unioned*). Then, all polygons belonging to European oversea territories such as the French islands in the Caribbean were removed (target: *river_basins_subset*). The resulting polygon geometries were unioned in the PostgreSQL database (target: *river_basins_subset_union_in_db*) to reduce the run-time of this union. Previous attempts to perform this union in R have shown too long run_times, 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 were chosen as study area (target: *river_basins_region_name*). Lastly, names were automatically assigned to each of the polygons (target: *selected_studyarea*).

River Network

After the data import, the next step is the filtering of linestring geometries from the river network based on attribute columns *dfdd* and *hyp* (target: *river_networks_non_dry_selected_streamtypes* in *preprocessing_targets.R* file; Fig. 4, line 30). *dfdd* classifies the geometries into BH140 (river), BH020 (canal) and BH030 (ditch). Canals and ditches were removed from the river network by only keeping the geometries with the value BH140 for several reasons. 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 Fig. 7. !!The section Technical Validation provides a discussion on this issue. The column *hyp* classifies the geometries into the following degrees of hydrologic persistence: 1 (Perennial), 2 (Intermittent), 3 (Ephemeral) and 4 (Dry). Geometries with the value 4 (Dry) were 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 were 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 Fig. 8B). 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 (!![10] used the already existing column *levelpathi* from their underlying 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 networks of each individual hydrologic order was then sorted by column *longpath* in descending order.

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

longpath indicates the length of the path from the start node of a 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 the the longest path. These identified geometries were 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.

```
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* was added as a unique ID for all geometries belonging to the same mainstem. The geometries of the respective river network was then merged based on this column (see difference in linestring geometries between Fig. 8B 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 (Fig. 9) that starts at the divergence. To transfer the methods from Belitz et al. (2019)[10] for the calculation of EU-MOHP, 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 main and minor flow paths is not provided by any column directly. Belitz et al. (2019)[10] used the column *divergence* for removing all minor flow paths. The assignment of the main flow paths results from the *object_id* of the column *nextdownid* of the upstream linestring. Consequently, all other other linestring geometries starting at the divergence can be defined as minor flow paths. This differentiation was already implicitly implemented in the previous step where all consecutive linestring geometries were merged based on the attribute columns *object_id* and *nextdownid*. In this step, all linestring geometries that intersect with the same other linestring geometry at their start and end node were removed from the river networks of all hydrologic orders except for the first order. Other minor paths that result from more complex divergences remain in the river network for further calculations (see *feature_id* in Fig. 9).

Then, the river networks were 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 Fig. 8C).

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 *preprocessing_targets.R* file; Fig. 4, line 30). A filter was applied to only keep those geometries of surface water bodies that intersect with the river network of a specific hydrologic order. To assign a stream order to the surface water bodies, the stream order of the river network geometries intersecting them was used although providing the surface water bodies with a stream order is not relevant for further processing.

Coastline

The coastline geometries consist of a vast number of nodes which slows down many geometry operations and calculations. Therefore, many processing steps were parallelised across smaller batches of the coastline data which lead to many helper targets. First, the imported polygon geometries were basins-wise unioned (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 the previously derived study area were removed (target: *coastline_filtered*). This is mainly to reduce the large number of geometries contained in the data caused by islands. A buffer of 3000m was added to the remaining polygon geometries to compensate for inaccuracies of the match between the study area and the coastline contours. This is relevant for the second next step. First, the buffer added polygon geometries were unioned to a single multipolygon geometry in the PostGIS database for reducing the run-time of calculating the union (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 (Fig. 4, line 31).

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 metrics. This core step is implemented in the target db_objects_to_grass which splits the processing for the hydrologic orders. 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. This merge of the geometries from the river network and the coastline is necessary to also include the coastline in the calculation of the Thiessen watersheds. After combining these geometries, they were written into the GRASS GIS database where they were 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` was 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* was 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 were 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 were performed separately for each of the 10 polygon geometries of the study area. After setting the region to the spatial extent of the respective study area polygon, the study area polygon was written into the GRASS GIS database. From this polygon, a raster mask was created to limit the all further raster calculations to the study area. Then, the distance to the nearest stream (see DS in eq. (1), (2) and (3)) was 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" already represents the Thiessen catchments. For deriving the Thiessen divides, this raster layer was converted into a vector layer of polygon geometries. The associated occurrence of dangling polygon outlines was reduced using the GRASS command *v.clean*. Subsequently, the rasterized outlines of these polygons were used as Thiessen divides. To calculate the distance the nearest Thiessen divide (see DD in eq. (1), (2) and (3)) with the restriction to not cross a stream, the GRASS command `r.walk` was 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 was created by assigning a value of 1 billion to all non-empty cells of the *river_network_raster*. For the calculation of the nearest divide by `r.walk`, the crossing of a river now results in high costs, which leads 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."

Now, the EU-MOHP measures were 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 measure DSD was calculated according to (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 was calculated according to (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 was multiplied by a factor of 10000 and rounded. The data type integer reduces storage space compared with float. For the same reason, the previously calculated raster layer *divide_stream_distance* was rounded, too.

As the last measure, SD was calculated according to (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 were exported from the GRASS GIS database. Therefore, a the sub-directory *output_data* with further sub-directories *divide_stream_distance*, *lateral_position* and *stream_distance* is created. The raster layers were written into these sub-directories in the GeoTiff (.tif) file format.

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 run-time of the pipeline as well as of individual targets is summarised in Tab. 2.

Data Records

The presented EU-MOHP v013.1.0 dataset is available in the hydroshare data portal at !!linktoDOI. The dataset represents gridded spatial information and is split into separate GeoTIFF files with a .tif file ending. Each file represents data on one of the three EU-MOHP measures LP, DSD and SD for one hydrologic order for different a spatial coverage. The file names are composed following 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 replaced by any combination of the values summarized in Tab. 3. The combinations of all placeholder values results in a total number of $n_{measures} \cdot n_{hydrologicorders} \cdot n_{studyareapolygons} = 3 \cdot 6 \cdot 2 = 36$ files. Files of the same EU-MOHP measure are stored together in the respective sub-directory "divide_stream_distance," "lateral_position" or stream_distance.

If you want to check more precisely whether your study area or area of interest is covered by this dataset, please see the webmap at !!link to github readme.

The presented EU-MOHP dataset has version v013.1.0 It is composed of the "EU-Hydro – River Network Database" version (v013) and a major and a minor version number (1.0) that are related to the methods of this dataset.

Technical Validation

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" was 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 a 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 a confusion is shown in Fig. 6. 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" 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 edges that run over land rather than along the coast. This includes the regions that are close tho the edges in the South and East of Turkey, in the East of continental Europe and in the East of Finland. Here, the edges of the underlying dataset, and thus this dataset, follow administrative boundaries instead of basin outlines. Therefore, calculated distances to the nearest stream in these regions may be inaccurate because another stream not in the dataset could be closer to a cell. 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 edges 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 was 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. To illustrate this, an 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 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 (see Fig. 10).

The method for calculating DD also causes NA cells 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. 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`[20,21] cover most common operations on raster data. For a fast raster cell value extraction based on polygons, the R package `exactextractr` is recommended.

The raster cell values of all GeoTIFF files were 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.

For transferring the presented methods to another custom region, equivalent input data to Tab. 1 is required.

Code availability

All processing and analysis was conducted using free open source software and free data. The code as a static code repository can be found at !!link to code in hydroshare. The actively developed code can be found in the github repository at this link. 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.

!!In order to maximize reproducibility, a docker container can be found at !!link to docker container. To run this docker container it is required to install the latest version of docker from this link and run the container with !!command.

!!If you want to reproduce this dataset, it is necessary to set the directories of the downloaded river network and coastline data in the file `constants.R` to the local directories.

The required underlying datasets "EU-Hydro – River Network Database" version v013 and "EU-Hydro – Coastline" version v013 can be downloaded at this link and here respectively.

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. M.M supported to R and PostGIS code development !!and set up the docker container. S.B. contributed to the conceptual design of the dataset. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Table 1. Overview of the required input data required to reproduce this dataset or, alternatively, transfer the methods to another custom region.

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

Figures & Tables

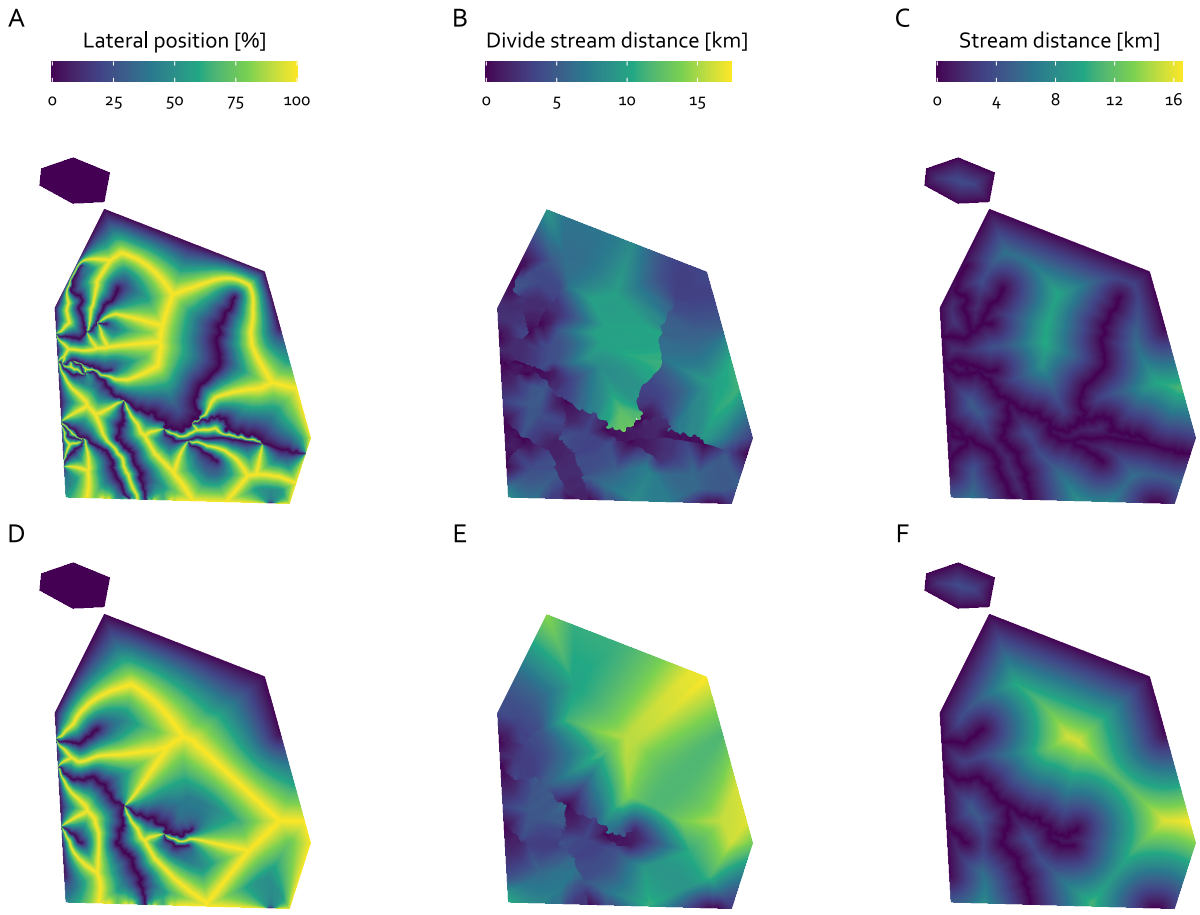

Figure 1. Resulting maps of the three EU-MOHP measures lateral position (A, D), divide stream distance (B, E) and stream distance (C, F) in the columns exemplary for the two hydrologic orders 3 (A, B, C) and 4 (D, E, F) in the rows. The colour gradients in the legend represent the mapped values of all plots in their column.

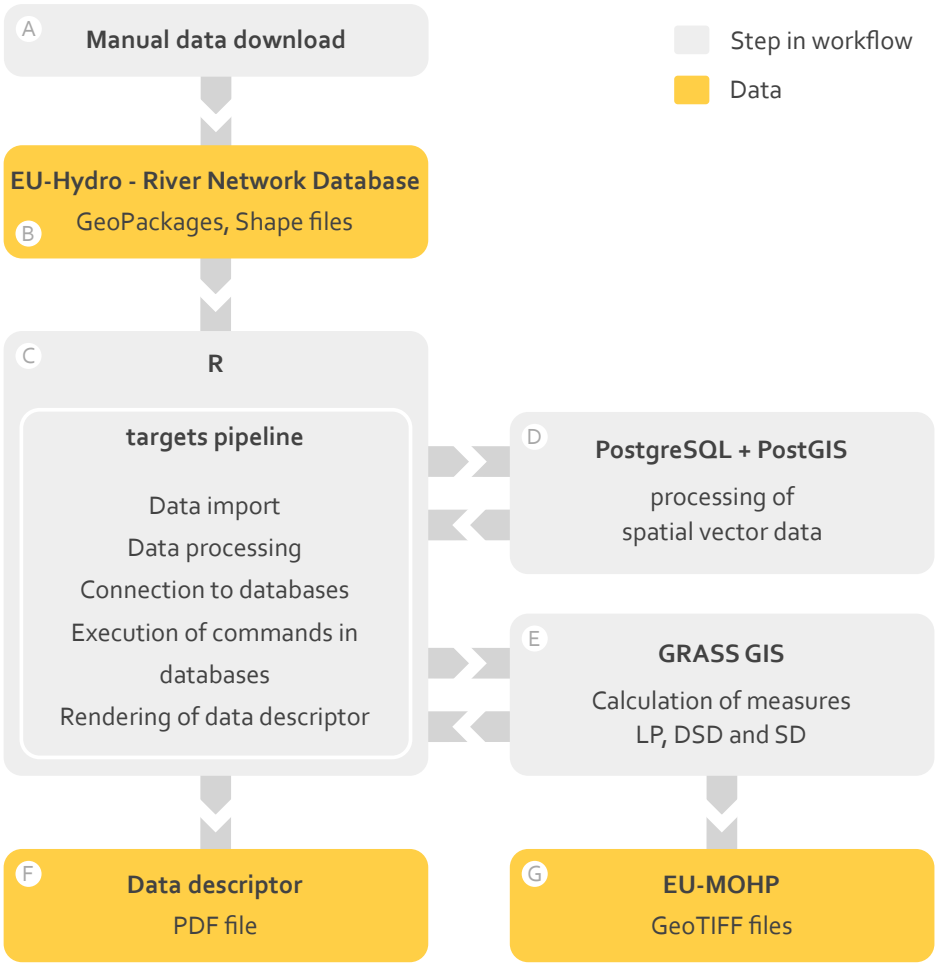

Figure 2. Workflow of the data processing in different software.

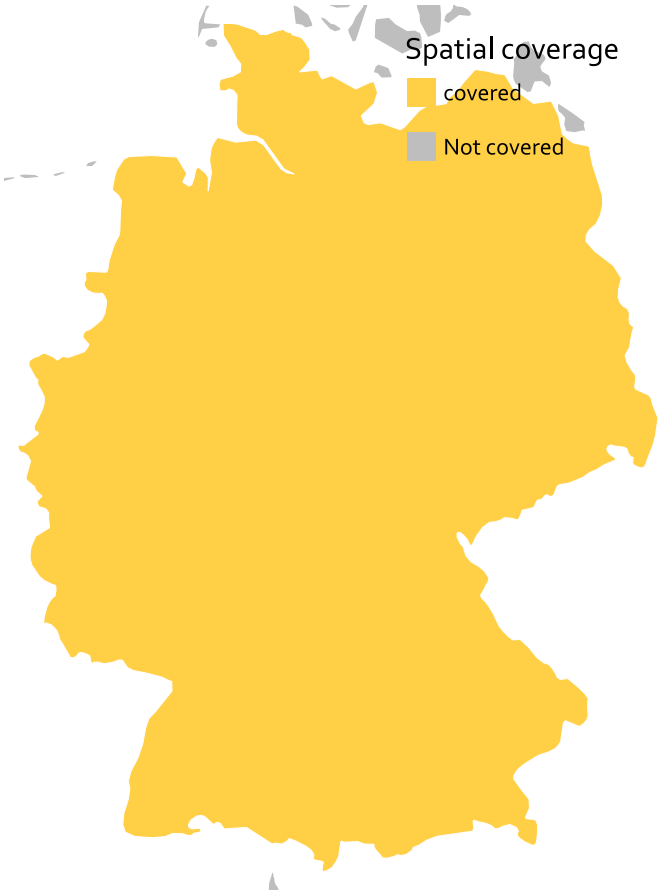

Figure 3. Spatial coverage of the dataset which is determined by the study area data layer.

```
.
+-- config.yml
+-- grassdata
+-- input_data
|   +-- data
|   +-- EUHYDRO_Coastline_EEA39_v013.shp
|   \-- macro_datapreparation_pipeline_test_studyarea_island.shp
+-- macro_mohp_feature.Rproj
+-- output_data
|   +-- divide_stream_distance
|   +-- lateral_position
|   \-- stream_distance
+-- R
|   +-- constants.R
|   +-- database_functions.R
|   +-- directory_functions.R
|   +-- export_functions.R
|   +-- grass_functions.R
|   +-- import_functions.R
|   +-- plot_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
|   +-- utility_targets.R
|   \-- visualization_targets.R
+-- test_files
+-- _targets
|   +-- meta
|   +-- objects
|   \-- scratch
\-- _targets.R
```

Figure 4. Directory tree of the project directory; only relevant subdirectories and files are listed here.

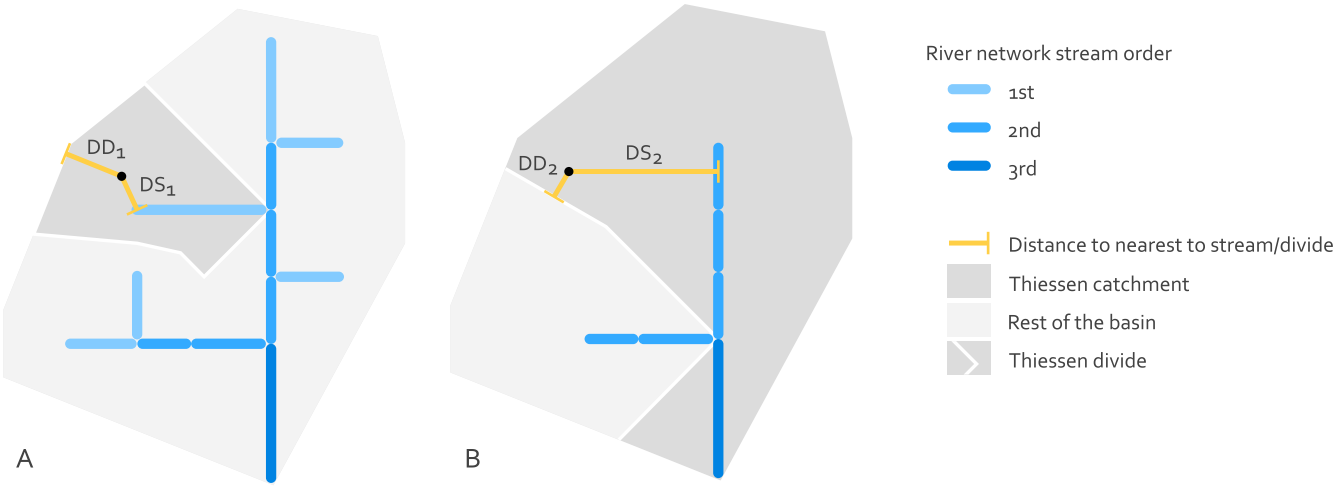

Figure 5. Schematic representation of MOHP measures using two examples for the hydrologic orders 1 (A) and 2 (B). DS is the horizontal distance to the nearest stream and DD is the horizontal distance to the nearest Thiessen divide under the condition that the divide is on the same side of the stream.

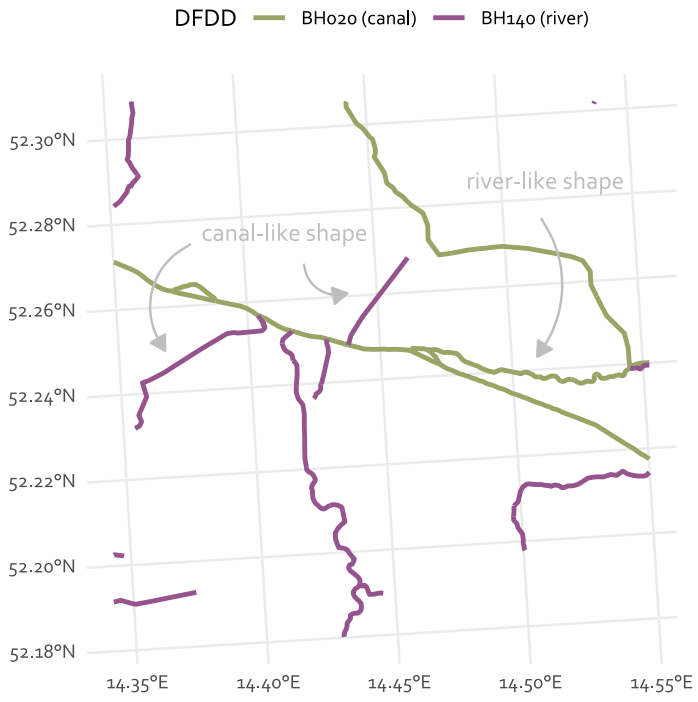

Figure 6. Example of the river network data showing the confusion between the classes BH140 (river), BH020 (canal) and BH030 (ditch).

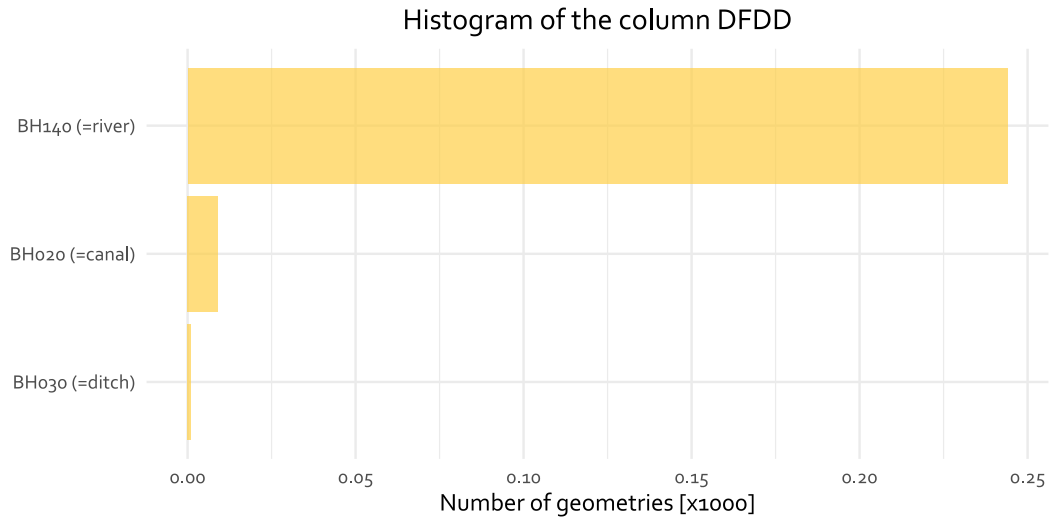

Figure 7. Distribution of classes BH140 (river), BH020 (canal) and BH030 (ditch) of the attribute column "DFDD".

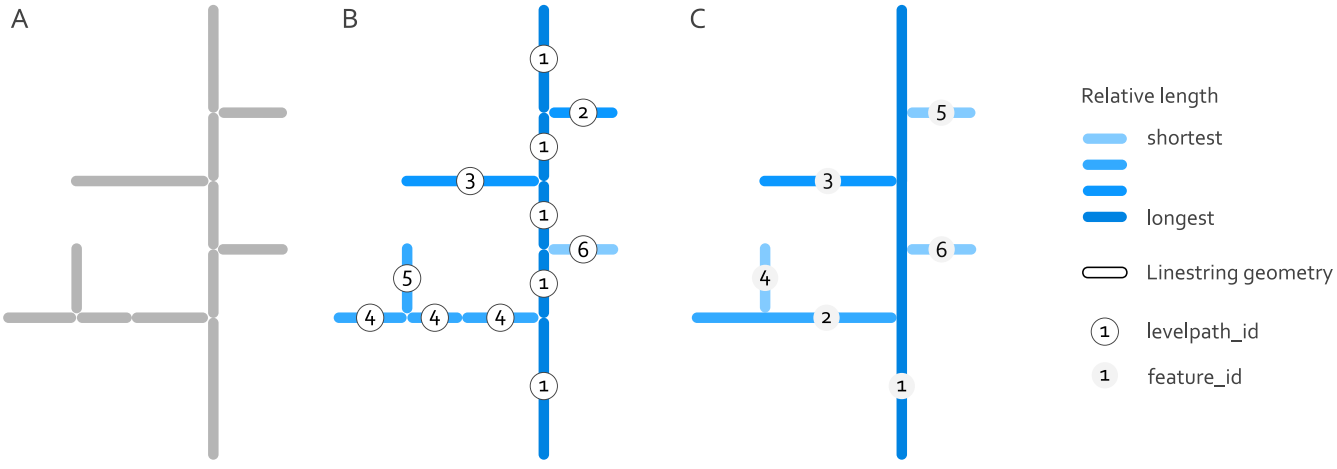

Figure 8. Schematic representation of the river network and its linestring geometries after import (A), after the identification of mainstems including the column ‘Levelpath ID‘ (B) and after merging the linestring geometries by this column and adding a ‘Feature ID‘ column (C).

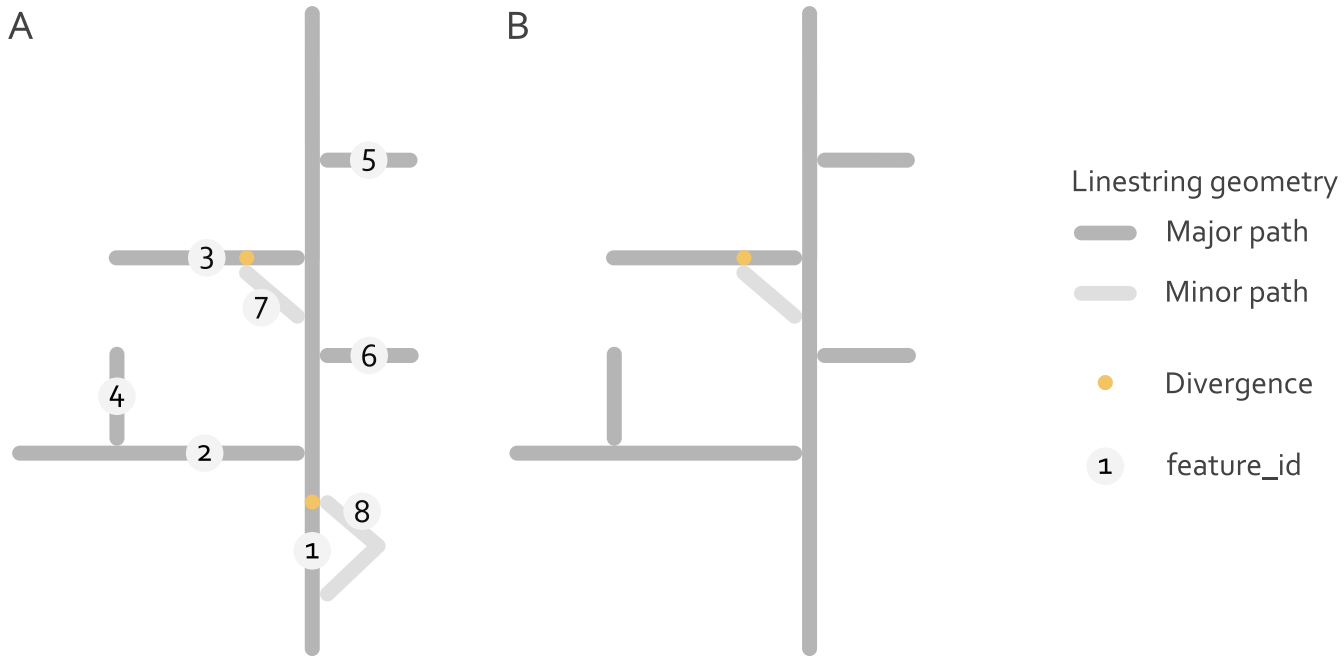

Figure 9. Schematic representation of the river network and its linestring geometries inlcuding divergences before (A) and after (B) the removal of minor paths under the condition that they intersect with the same linestring geometry at the start and end node. The linestring geometry with the "Feature ID" 8 is being removed from the river network in B, because it intersects the linestring geometry with the ID 1 the start and end node. Whereas linestring geometry with the ID 7 remains in the river network, because it intersects with two different linestring geometries at start and end node (ID 3 and ID 1).

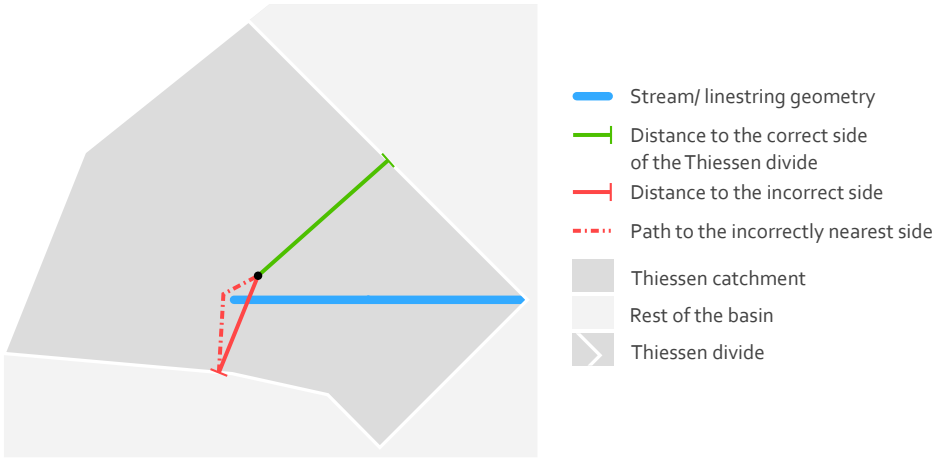

Figure 10. Schematic example showing the source of inaccurate of DD in areas near headwaters caused by the applied method to calculate DD. The red distance as DD is incorrect, because it crosses the stream and therefore does not fulfill the defined condition The correct DD would be the green distance.

Table 2. Overview of the run-time and data size of all targets or processing steps in descending order.

	Run-time				Data size
Target name	Seconds	Minutes	Hours	Days	Mb
db_objects_to_grass	273.9	4.6	0.1	0	0.0
river_networks	123.1	2.1	0.0	0	1112.0
inland_waters	273.1	4.6	0.1	0	183.0
data_descriptor	45.6	0.8	0.0	0	1.9
river_canal_confusion_plot	40.0	0.7	0.0	0	1112.2
river_networks_clip	39.9	0.7	0.0	0	0.3
db_inland_waters	21.7	0.4	0.0	0	0.0
rivernetworks_merged_per_streamorder	7.3	0.1	0.0	0	0.7
dataset_map_overview_plot	8.4	0.1	0.0	0	71.4
output_data_table	8.0	0.1	0.0	0	0.0
workflow_figure	7.7	0.1	0.0	0	0.0
directory_river_networks	7.6	0.1	0.0	0	0.0
input_data_table	7.3	0.1	0.0	0	0.0
selected_hydrologic_orders	7.2	0.1	0.0	0	0.0
river_networks_streamorderone	7.1	0.1	0.0	0	0.3
distinct_streamorders_in_riverbasins	5.6	0.1	0.0	0	0.0
directory_tree	5.4	0.1	0.0	0	0.1
config	5.2	0.1	0.0	0	0.0
river_basin_names	4.4	0.1	0.0	0	0.0
db_river_networks_merged_per_streamorder	5.8	0.1	0.0	0	0.0
db_inland_waters_strahler	1.9	0.0	0.0	0	0.0
studyarea_figure	1.1	0.0	0.0	0	0.2
db_selected_studyarea	0.7	0.0	0.0	0	0.0
rivernetworks_feature_id	1.4	0.0	0.0	0	0.7
selected_studyarea	0.6	0.0	0.0	0	0.0
db_river_networks_strahler_studyarea	0.4	0.0	0.0	0	0.0
db_river_networks_clean	0.4	0.0	0.0	0	0.0
coastline_watershed	0.4	0.0	0.0	0	0.0
river_networks_files	0.2	0.0	0.0	0	0.0
dfdd_stats_bar_plot	0.0	0.0	0.0	0	0.4
river_networks_greater_one_grouped	0.0	0.0	0.0	0	0.4
streamorders	0.0	0.0	0.0	0	0.0
river_networks_grouped	0.0	0.0	0.0	0	0.7
river_networks_non_dry_selected_streamtypes	0.0	0.0	0.0	0	0.3
river_networks_clean	0.0	0.0	0.0	0	0.3
river_networks_treated_brackets	6.0	0.1	0.0	0	0.0
river_networks_imputed_streamorder_canals_as_1	0.0	0.0	0.0	0	0.3
filepath_studyarea_pipelinetest	0.0	0.0	0.0	0	0.0
Total	**917.4**	**15.3**	**0.2**	**0**	**2485.2**

Table 3. Overview of the output file naming scheme and its placeholder values. Files for any combinations of the placeholder values exists. 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_streamorder1_30m.tif". The spatial coverage of the values for "<region name for spatial coverage>" is shown in Fig. 1.

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	
<spatial resolution>	30m	Spatial resolution

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, 10.1016/j.jhydrol.2020.125085 (2020).
2. 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, 10.1016/j.ejrh.2020.100697 (2020).
3. 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).
4. Knoll, L., Breuer, L. & Bach, M. Nation-wide estimation of groundwater redox conditions and nitrate concentrations through machine learning. *Environ. Res. Lett.* **15**, 064004, 10.1088/1748-9326/ab7d5c (2020).
5. Mueller, J. *et al.* Surrogate Optimization of Deep Neural Networks for Groundwater Predictions. *arXiv:1908.10947 [cs, math, stat]* (2019). ArXiv: 1908.10947.
6. Stackelberg, P. E. *et al.* Machine Learning Predictions of pH in the Glacial Aquifer System, Northern USA. *Groundwater* **59**, 352–368, 10.1111/gwat.13063 (2021).
7. 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, 10.1016/j.proeng.2016.07.527 (2016).
8. Wunsch, A., Liesch, T. & Broda, S. Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). *J. Hydrol.* **567**, 743–758, 10.1016/j.jhydrol.2018.01.045 (2018).
9. Wunsch, A., Liesch, T. & Broda, S. Groundwater Level Forecasting with Artificial Neural Networks: AComparison of LSTM, CNN and NARX. preprint, Groundwater hydrology/Modelling approaches (2020). 10.5194/hess-2020-552.
10. 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, 10.1029/2019WR025908 (2019).
11. 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).
12. 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, 10.1111/1752-1688.12879 (2020).
13. 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).
14. Strahler, A. N. Quantitative analysis of watershed geomorphology. *Eos, Transactions Am. Geophys. Union* **38**, 913–920 (1957).
15. R Core Team. *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, Vienna, Austria, 2020).
16. 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).
17. EU-Hydro - River Network Database - Copernicus Land Monitoring Service (2021).
18. EU-Hydro - Coastline (2021).
19. Gallaun, H., Dohr, K., Puhm, M., Stumpf, A. & Hugé, J. EU-Hydro - River Net User Guide 1.3 (2019).
20. Hijmans, R. J. *raster: Geographic Data Analysis and Modeling* (2020).
21. Pebesma, E. *stars: Spatiotemporal Arrays, Raster and Vector Data Cubes* (2021).
22. Wickham, H. *et al.* Welcome to the tidyverse. *J. Open Source Softw.* **4**, 1686, 10.21105/joss.01686 (2019).
23. Allaire, J. J. *et al. rmarkdown: Dynamic Documents for R* (2021).
24. Pebesma, E. Simple Features for R: Standardized Support for Spatial Vector Data. *The R J.* **10**, 439–446, 10.32614/RJ-2018-009 (2018).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

48. ? *styler: Non-Invasive Pretty Printing of R Code* (2021).

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

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

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