BasinMaker

Release 3.1.0

Hydrology research group at the University of Waterloo

Jun 22, 2023

CONTENTS:

User Documentation and examples.

CHAPTER

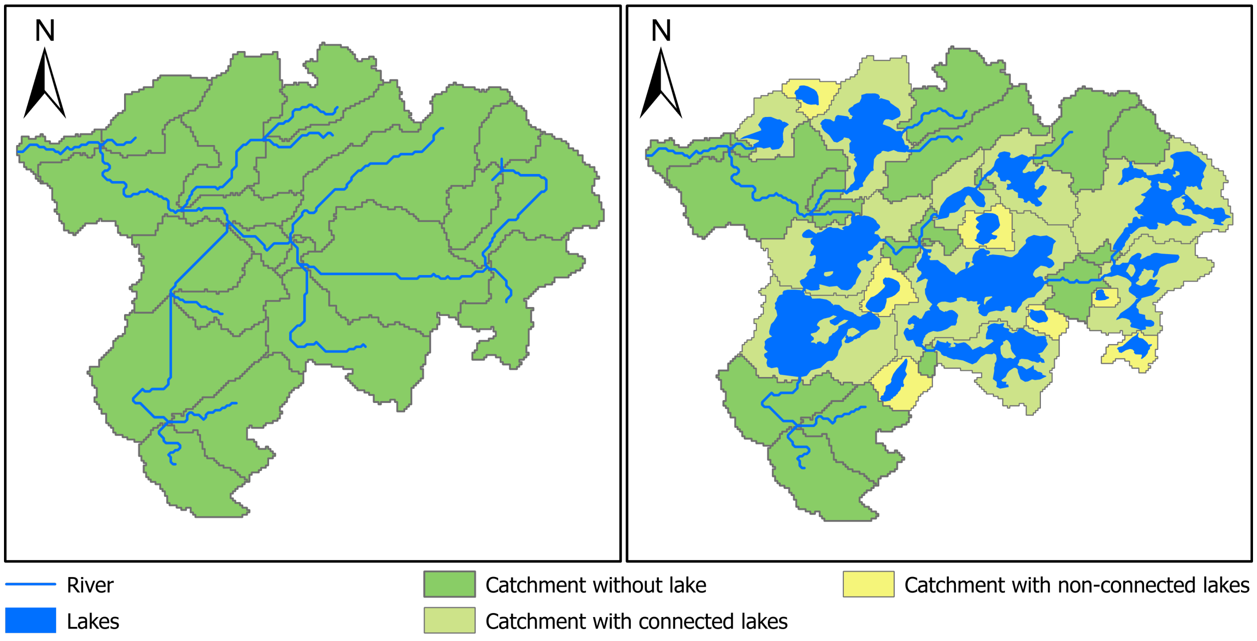
ONE

OVERVIEW

Before using the BasinMaker or routing product, the differences between routing structure without considering lake and routing structure with lakes defined by BasinMaker and routing product will be introduce in this section.

A routing structure without considering lakes is showed in Figure A. Catchments in a routing structure without considering lakes are only determined by river reaches (Figure A). When a semi hydrological model build with this routing structure, the lakes inflow and outflow cannot be simulated, and thus the impact of lake on the routing process modeling such as flow attenuation cannot be correctly modeled. The reason is that the streamflow is only explicitly simulated at the outlet of each catchment in semi-distributed hydrological models. It is not explicitly simulated at the mid of the river segment or any point inside each catchment. An example of routing structure with lakes represented by the BasinMaker or routing product is shown in Figure B. Lakes are divided into two categories: (1) connected lakes (CL), which indicates lakes outlets are explicitly connected to a downstream non-zero length river channel/lake in the routing product; and (2) non-connected lakes (NCL), which denotes lakes is not explicitly connected to the downstream routing network. The connection is more implicit (Figure B). Both connected lakes (CLs) and NCLs within a watershed are considered to be contributing areas of the watershed. As such, both CLs and NCLs will drain to the outlet of the watershed. Both CLs and NCLs are represented by a lake catchment (Figure B). A lake catchment is defined by the following rules:1) The extent of the lake catchment will fully cover the lake; 2) the outlet of the lake catchment is the same as the outlet of the lake; 3) each lake's inlets are treated as a catchment outlet. In this way, both inflow and outflow of each lake can be explicitly simulated by hydrologic routing models.

The only difference between CLs and NCLs is that CLs always drain into an explicitly represented river channel that is connected to the lake outlet while NCLs do not. NCLs exist because for smaller catchments, flow accumulation threshold settings can sometimes suppress the creation of a river channel at the lake outlet. As such, NCLs should drain directly, via a zero length flow path, into the next downstream catchment. Users need to ensure their hydrologic routing model accomplishes this. Specific hydrologic routing logic is as follows for NCLs: NCL catchment outflows need to be delivered to the next downstream river channel and if that river channel has a zero length (only possible if downstream catchment is also a lake catchment), that water must be delivered directly to the lake in this downstream catchment.


N
N
River
Lakes
Catchment without lake
Catchment with connected lakes
Catchment with non-connected lakes

CHAPTER

TWO

ROUTING PRODUCTS FROM BASINMAKER

- The North American Lake-River Routing Product v2.1 developed using BasinMaker is available at here.
- BasinMaker generated Ontario Lake-River Routing Product v1 is available at here. .

CHAPTER

THREE

BASINMAKER ON GOOGLE COLAB

A post-processing example via google colab can be found at here here. The google colab is an online python notebook dose not require installation. This example will show you how to discretize, simplify, and revise the provided routing product for your purposes.

CHAPTER

FOUR

VERSION UPDATE NOTES

We are excited to announce the release of version 3.1.0 of our software, which includes both major and minor updates.

4.1 Major updates in version 3.1.0 include:

- Fix the lake inflow subbasin bug in the previous version.

4.2 Major updates in version 3.0.3 include:

- Developed BasinMaker delineation functions under the ArcGIS Pro python environment to improve the user experience.
- Added a new function to the BasinMaker post-processing functions called *Add_Point_Of_Interest_Sites_In_Routing_Product*, which allows users to define the point of interest in the developed routing product.

4.3 Minor updates in version 3.0.3 include:

- Add an new parameter to function *Remove_Small_Lakes* and *Decrease_River_Network_Resolution* to allow users remove tiny subbasins in the routing network.
- Fixed the observed bugs in the previous version.

CHAPTER FIVE

AUTHORS

BasinMaker and the associated river and lake routing product was developed by the hydrology research group at the University of Waterloo. Primary Contributors are Ming Han, Hongren Shen, Bryan A. Tolson, James R. Craig and Juliane Mai. Secondary contributors are Simon Lin, Nandita B. Basu and Frezer Awol.

CHAPTER SIX

CITATION

- Han, M., Shen, H., Tolson, B. A., Craig, J. R., Mai, J., Lin, S. G. M., Basu, N. B., Awol, F. S. (2023). BasinMaker 3.0: A GIS toolbox for distributed watershed delineation of complex lake-river routing networks. Environmental Modelling and Software, 105688. https://doi.org/10.1016/J.ENVSOFT.2023.105688.

CHAPTER

SEVEN

LICENSE

BasinMaker is open-source under the Artistic License 2.0. This sofware is freely distributed ’as is’ without warranties or conditions of any kind, either express or implied, including, without limitation, any warranties or conditions of title, non-infringement, merchantability, or fitness for a particular purpose.

CHAPTER

EIGHT

INSTALLATION V3.1.0

8.1 Overview

BasinMaker is a python package work on several existing GIS platforms. So, the installation of BasinMaker includes two steps: 1) setup the python environment for the dependent GIS platforms; and 2) install BasinMaker itself.

Two installation modes (light installation and full installation) are available. The light installation will allow user to use BasinMaker to post process an existing routing product, such as the Basinmaker derived North American lake-river routing product. But it cannot be used to delineate a lake-river routing structure from DEM. The combination of BasinMaker light installation and the North American lake-river routing product could generate lake-river routing structures satisfying many user modeling demands. While the full installation of BasinMaker enables users to delineate a new lake-river routing structure from a user specified DEM.

For light installation (recommended unless users know for sure they need to delineate their watershed from scratch from a DEM), only geopandas or ArcGIS pro is needed. The python environment for both geopandas can be easily compiled within the anaconda environment under windows OS environment. The instruction about light installation procedure can be found in *Light installation*. Note that users can install both the light version and the full version on the same OS system.

Users, who want to use BasinMaker post processing functions without a Windows OS system, could try BasinMaker on Google Colab in *BasinMaker on Google Colab*

For full installation, BasinMaker can work under two python enviroments: the ArcGIS Pro and GRASS/QGIS GIS python enviroments. The instruction to install BasinMaker for each python enviroments are provided at *Full installation*. It is quite a challenge to setup a python environment for QGIS and GRASS together. Here, two procedures are provided for Windows OS systems, respectively. The procedure has been tested on several machines. But we can't guarantee install procedures work on every machine. If you run into a problem create an issue on the GitHub and, time permitting, we will try to help. If you managed to do the full installation on a different operating system, we would be grateful if you could document and share the detailed installation procedure that was successful (email: m43han@uwaterloo.ca).

8.2 Updating BasinMaker to v3.1

For existing users of a BasinMaker version before 3.1 who want to update to version v3.1, users should simply reinstall BasinMaker 3.1 to a new working environment (e.g., called 'Basinmaker3_1')

8.3 BasinMaker on Google Colab

A post-processing example via google colab (no installation on your local machine necessary!) can be found at here here. The google colab is an online python notebook dose not require installation. This example will show you how to discretize, simplify, and revise the provided routing product for your purposes.

8.4 Light installation

8.4.1 Geopandas with anaconda

1. Install anaconda

   The installer of anaconda can be installed from here. Note for windows system, please activate the 'Register Anaconda3 as my default python 3.9'

2. Create an empty python environment and then active it

   For windows system, search and open "Anacoda Prompt" (Windows) to active a conda command line. **Users must make sure:**

   - They have the proper privileges to create environment variables (e.g., run Anacoda Prompt as administrator will work)
   - DO NOT USE Anaconda Powershell Prompt

   Then

   ```
   conda create --name <any_name_for_env> python==3.11.3
   conda activate <any_name_for_env>
   ```

3. Install packages (may take some time)

   ```
   conda config --append channels conda-forge
   ```

   ```
   conda install gdal
   ```

   ```
   python -m pip install  pytest  simpledbf netCDF4 joblib jupyter requests␣
   ↪wget gdown pandas rasterstats geopandas rasterio scipy
   ```

4. Install BasinMaker

   ```
   python -m pip install https://github.com/dustming/basinmaker/archive/master.
   ↪zip
   ```

5. Test validation

   Please download the test data and scripts from here. and unzip it to a folder, the path of this folder will refer as path_test_data in following section. Then

   Please ignore following output messages

   - PyTables is not installed. No support for HDF output.
   - SQLalchemy is not installed. No support for SQL output.
   - Warnings

```
cd path_test_data/test
python test_light_installation_qgis.py
(... some messages)
####################################
BasinMaker is successfully installed
####################################
```

6. Users must active this conda environment when they wish to use functionalities from BasinMaker.

8.5 Full installation

The BasinMaker watershed delineation mode can be used under both ArcGIS Pro and GRASS GIS environments. We recommend using BasinMaker under the ArcGIS Pro environment. However, installation instructions for both Python environments are provided in the following two sections.

8.5.1 ArcGIS Pro in Windows

1. Install ArcGIS Pro

   BasinMaker has been tested with ArcGIS Pro version 3.0.3. To use the software, please ensure that you have installed this version of ArcGIS Pro. If you need assistance with installing ArcGIS Pro, please contact your IT department for detailed instructions.

2. Setup the python environment for BasinMaker in ArcGIS Pro

   Below are the key steps to create an ArcGIS Python environment. For detailed instructions, please refer to this link .

   - Open ArcGIS Pro and click on the "Settings/Project" icon in the upper left corner of the ArcGIS Pro window.
   - Click the "Package Manager" tab. And then click the "Manage Environments" button in the upper right corner of the window.
   - Clone the "ArcGIS Pro" environment and name it <any_name_for_env>. The clone process will take a few minutes.
   - Select and active the newly created environment and restart ArcGIS Pro for the changes to take effect

3. Install BasinMaker in ArcGIS Pro

   - Open the ArcGIS Pro Python command prompt. To open the ArcGIS Pro Python command prompt, navigate to the Windows program directory: Programs > ArcGIS > Python Command Prompt.
   - Install BasinMaker and related pacakges using the following command:

```
> python -m pip install https://github.com/dustming/basinmaker/archive/
↪master.zip
```

4. Install dependent packages

```
> python -m pip install pytest simpledbf netCDF4 joblib jupyter requests␣
↪wget gdown geopandas rasterstats
```

5. Validate the installation with the package of test files.

    - Please download the test data and script and unzip it to a folder, the path of this folder will refer as path_test_data in following section. Then
    - Open the ArcGIS Pro Python command prompt and run the following command:

```
cd path_test_data
python test_full_delineation_arcgis.py
(... some messages)
####################################
BasinMaker is successfully installed
####################################
```

    - Please ignore following output messages

      PyTables is not installed. No support for HDF output.

      SQLalchemy is not installed. No support for SQL output.

      Warnings

6. Users must Open the ArcGIS Pro Python command prompt every time they wish to use functionalities from BasinMaker.

8.5.2 QGIS and GRASS in Windows

1. Installation of QGIS and GRASS using OSGEO4W:

   For the Windows system, we can install both GRASS and QGIS within OSGEO4W environment.

   The OSGeo4W is a binary distribution of a broad set of open source geospatial software for Windows environments, including both GRASS GIS and QGIS.

   The OSGeo4W installer can be downloaded from here.

   Please use the advanced install option and keep the default selection in all pop up pages, except in the 'select package page'.

   In the select package:

    - In the Desktop group, please select 1) grass: GRASS GIS 7.8; 2) qgis: QGIS DESKTOP; 3)qt5_tools:Qt5 tools (development); 4)saga:SAGA(. . .)
    - In the Libs group, please select 1)python3-geopandas; 2)python3-rtree; 3)python3-rasterstats

   We would suggest to

    - Install QGIS and GRASS outside the **C/:Program Files**. Better to install them into a folder path without space in the folder name.
    - Run the downloaded installation file

2. Setup GRASS and QGIS python environment

   The python environment for QGIS and GRASS GIS in Windows can be set up by modifying the following `basinmaker.bat.txt`.

    - Please rename 'basinmaker.bat.txt' to 'basinmaker.bat'.
    - Please change OSGEO4W_ROOT to your OSGEO4W installation folder at line 2.
    - Please change the grass78.* in line 8 and 10 to your GRASS GIS version number.

- Please double check the paths defined in the basinmaker.bat file exist in your machine
- Save the modified basinmaker.bat to a handy directory. Run basinmaker.bat every time before using basinmaker.

3. Install BasinMaker (do not activate anaconda)

```
>basinmaker.bat
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. All rights reserved
>
>python -m pip install https://github.com/dustming/basinmaker/archive/
↪master.zip
```

4. Validate the GRASS and QGIS python environment

- Please check if the python executable comes from the OSGeo4W installation folder by typing following commands after run basinmaker.bat. If the output is not similar to the output showed in following output block. Please go back to step 2 and check the basinmaker.bat file

```
>where python
C:\OSGeo4W\apps\Python37\python.exe
```

- Check if all dependent QGIS and GRASS libraries can be imported in current python environment by type following commands.

```
>python
>>>from qgis.core import *
>>>import qgis
>>>from qgis.analysis import QgsNativeAlgorithms
>>>from qgis.PyQt.QtCore import *
>>>from qgis import processing
Application path not initialized
>>>from processing.core.Processing import Processing
>>>from processing.tools import dataobjects
>>>import grass.script as grass
>>>from grass.script import array as garray
>>>from grass.script import core as gcore
>>>import grass.script.setup as gsetup
>>>from grass.pygrass.modules.shortcuts import general as g
>>>from grass.pygrass.modules.shortcuts import raster as r
>>>from grass.pygrass.modules import Module
>>>quit()
```

5. Install dependent packages

```
python -m pip install simpledbf grass_session scipy joblib wget gdown
python -m pip install --upgrade pip
python -m pip install geopandas -U
python -m pip install rasterstats -U
```

6. **Install GRASS GIS addons**

   Install following GRASS GIS addons:

   - r.accumulate
   - r.clip

   - r.stream.basins
   - r.stream.snap

   For new GRASS users, see how to install GRASS GIS addon here.

   If you want to learn how to use GRASS for more than BasinMaker, this site. may help you.

7. Test validation

   - Please download the test data and scripts from here. and unzip it to a folder, the path of this folder will refer as path_test_data in following section. Then
   - run basinmaker.bat
   - Please ignore following output messages

     PyTables is not installed. No support for HDF output.

     SQLalchemy is not installed. No support for SQL output.

     Warnings

   ```
   cd path_test_data/test
   python test_full_installation.py
   (... some messages)
   ####################################
   BasinMaker is successfully installed
   ####################################
   ```

8. Users must run basinmaker.bat every time they wish to use functionalities from BasinMaker.

CHAPTER NINE

BASINMAKER DEVELOPER DOCUMENTATION

9.1 BasinMaker - postprocessing tools

9.1.1 Extract the region of interest

`basinmaker.basinmaker.postprocess.`**`Select_Subregion_Of_Routing_Structure`**`(`*path_output_folder*, *routing_product_folder*, *gis_platform*, *most_down_stream_subbasin_ids=[]*, *most_up_stream_subbasin_ids=[]*`)`

Select subregion of hydrologic routing network based on provided subbasin IDs

Parameters

- **`path_output_folder`** (*`string`*) – is the folder path that stores generated outputs
- **`routing_product_folder`** (*`string`*) – is the folder path where the input hydrologic routing network is stored
- **`gis_platform`** (*`string`*) – is the parameter indicating which gis platform is used. It can be either “qgis” or “arcgis”.
- **`most_down_stream_subbasin_ids`** (*`list`*) – A list of subbasin ID, the subbasin IDs in this list should be the most downstream subbasin ID of each interested watershed.
- **`most_up_stream_subbasin_ids`** (*`list`*) – A list of subbasin ID, the subbasins that drainage to the SubID in this list will be excluded. Value [-1] is required to indicate no upstream subbasin needs to be removed.

Notes

This function has no return values, The extracted hydrological routing network will be generated in the path_output_folder including following files:

finalcat_info.shp
[shapefile] The finalized hydrologic routing network. the GIS layer containing subbasin polygons which respect the lake inflow and outflow routing structures. This layer contains all the necessary information for hydrologic routing through the lake-river network.

finalcat_info_riv.shp
[shapefile] The finalized hydrologic routing network. the GIS layer containing river network polylines in the routing network.

catchment_without_merging_lakes.shp
: [shapefile] The GIS layer containing subbasin polygons of an incomplete hydrologic routing network. In this incomplete hydrologic routing network subbasin polygons covered by the same lake are not merged into one lake subbasin yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

river_without_merging_lakes.shp
: [shapefile] The GIS layer containing river polylines of an incomplete hydrologic routing network. In this incomplete hydrologic routing network, the river polylines covered by the same lake are not merged into one river segment yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

sl_connected_lake.shp
: [shapefile] the GIS layer containing the lake polygons of lakes that are connected by the river_without_merging_lakes.shp

sl_non_connected_lake.shp
: [shapefile] the GIS layer containing the lake polygons of lakes that are not connected by the river_without_merging_lakes.shp

poi
: [shapefile] It is the point shapefile that represent the point of interest after snap to river network.

Examples

9.1.2 Filter lakes

`basinmaker.basinmaker.postprocess.`**`Remove_Small_Lakes`**(*path_output_folder*, *routing_product_folder*, *gis_platform*, *connected_lake_area_thresthold=-1*, *non_connected_lake_area_thresthold=-1*, *selected_lake_ids=[]*, *area_thresthold=0.009*)

This function is to simplify the hydrologic routing network by removing lakes.

Parameters

- **`path_output_folder`** (*`string`*) – is the folder path that stores generated outputs
- **`routing_product_folder`** (*`string`*) – is the folder path where the input hydrologic routing network is stored
- **`gis_platform`** (*`string`*) – is the parameter indicating which gis platform is used. It can be either "qgis" or "arcgis".
- **`connected_lake_area_thresthold`** (*`float (optional)`*) – is a lake area thresthold for connected lakes in km2. Connected lake with lake area below which lake will be removed
- **`non_connected_lake_area_thresthold`** (*`float (optional)`*) – is a lake area thresthold for non-connected lakes in km2 Non connected lake with lake area below below which lake will be removed
- **`selected_lake_ids`** (*`list (optional)`*) – A list of lake IDs from in the hydrologic routing network (Column 'HyLakeId'). Lakes with their lake ID in this list will be kept by the BasinMaker even if their area smaller than the lake area threstholds
- **`area_thresthold`** (*`float (optional)`*) – This parameter sets a subbasin area threshold in square kilometers (km^2). Subbasins with an area below this threshold will be merged with

downstream or upstream subbasins, depending on their location within the river network. This can be useful for simplifying the river network and reducing computational requirements. The parameter is optional and has a default value of 0.009 km2. If the parameter is set to 0, no subbasin merging will be performed.

Notes

This function has no return values, The simplified hydrological routing network will be generated in the path_output_folder including following files:

finalcat_info.shp
: [shapefile] The finalized hydrologic routing network. the GIS layer containing subbasin polygons which respect the lake inflow and outflow routing structures. This layer contains all the necessary information for hydrologic routing through the lake-river network.

finalcat_info_riv.shp
: [shapefile] The finalized hydrologic routing network. the GIS layer containing river network polylines in the routing network.

catchment_without_merging_lakes.shp
: [shapefile] The GIS layer containing subbasin polygons of an incomplete hydrologic routing network. In this incomplete hydrologic routing network subbasin polygons covered by the same lake are not merged into one lake subbasin yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

river_without_merging_lakes.shp
: [shapefile] The GIS layer containing river polylines of an incomplete hydrologic routing network. In this incomplete hydrologic routing network, the river polylines covered by the same lake are not merged into one river segment yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

sl_connected_lake.shp
: [shapefile] the GIS layer containing the lake polygons of lakes that are connected by the river_without_merging_lakes.shp

sl_non_connected_lake.shp
: [shapefile] the GIS layer containing the lake polygons of lakes that are not connected by the river_without_merging_lakes.shp

poi
: [shapefile] It is the point shapefile that represent the point of interest after snap to river network.

Examples

9.1.3 Increase catchment area

`basinmaker.basinmaker.postprocess.`**`Decrease_River_Network_Resolution`**(*path_output_folder*, *routing_product_folder*, *gis_platform*, *minimum_subbasin_drainage_area*, *area_thresthold*)

This function is to simplify the hydrologic routing network by removing subbasins/river reaches with their drainage area below user provided drainage area thresthold.

Parameters

- `path_output_folder` (*string*) – is the folder path that stores generated outputs
- `routing_product_folder` (*string*) – is the folder path where the input hydrologic routing network is stored
- `gis_platform` (*string*) – is the parameter indicating which gis platform is used. It can be either "qgis" or "arcgis".
- `minimum_subbasin_drainage_area` (*float*) – is a subbasin drainage area thresthold, subbasin with their drainage area smaller than this thresthold will be removed.
- `area_thresthold` (*float (optional)*) – This parameter sets a subbasin area threshold in square kilometers (km^2). Subbasins with an area below this threshold will be merged with downstream or upstream subbasins, depending on their location within the river network. This can be useful for simplifying the river network and reducing computational requirements. The parameter is optional and has a default value of 0.009 km2. If the parameter is set to 0, no subbasin merging will be performed.

Notes

This function has no return values, The simplified hydrological routing network will be generated in the path_output_folder including following files:

finalcat_info.shp
[shapefile] The finalized hydrologic routing network. the GIS layer containing subbasin polygons which respect the lake inflow and outflow routing structures. This layer contains all the necessary information for hydrologic routing through the lake-river network.

finalcat_info_riv.shp
[shapefile] The finalized hydrologic routing network. the GIS layer containing river network polylines in the routing network.

catchment_without_merging_lakes.shp
[shapefile] The GIS layer containing subbasin polygons of an incomplete hydrologic routing network. In this incomplete hydrologic routing network subbasin polygons covered by the same lake are not merged into one lake subbasin yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

river_without_merging_lakes.shp
[shapefile] The GIS layer containing river polylines of an incomplete hydrologic routing network. In this incomplete hydrologic routing network, the river polylines covered by the same lake are not merged into one river segment yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

sl_connected_lake.shp
[shapefile] the GIS layer containing the lake polygons of lakes that are connected by the river_without_merging_lakes.shp

sl_non_connected_lake.shp
[shapefile] the GIS layer containing the lake polygons of lakes that are not connected by the river_without_merging_lakes.shp

poi
[shapefile] It is the point shapefile that represent the point of interest after snap to river network.

Examples

9.1.4 Modify point of interest in the routing product

`basinmaker.basinmaker.postprocess.`**`Add_Point_Of_Interest_Sites_In_Routing_Product`**`(`*path_output_folder*, *routing_product_folder*, *gis_platform*, *clean_exist_pois*, *area_thresthold*`)`

This function allows the user to modify point of interest (POI) sites in a pre-existing BasinMaker-generated user input hydrologic routing network/product. Specific modifications this function can make are to add new POI sites, remove existing POI sites, or modify the location of existing POI sites in the input routing network. This function will use the location of the provided POI and link them to the subbasins in the routing network. At least one POI should be provided in the point shapefile.

Parameters

- **path_output_folder** (*string*) – is the folder path that stores generated outputs
- **routing_product_folder** (*string*) – is the folder path where the input hydrologic routing network is stored
- **path_to_points_of_interest_points** (*string*) – is the path to the point shapefile that contains the point of interest (POI) sites. The shapefile must have an attribute table that includes the following columns:
  - Obs_NM (string): This column should contain the name or ID of the POI site. When the provided POI has the same Obs_NM as the existing POI in the routing product, this function will assume that the user wants to relocate the existing POI. Otherwise, the clean_exist_pois variable below controls whether the new POI sites fully replace existing POI sites or augment the existing POI sites.
  - DA_Obs (float): This column should contain the drainage area of the POI site.
  - SRC_obs (string): This column should contain the source of the POI site.
  - Type (string): This column should contain the type of the POI site. Each POI can only have one type. The following types are currently supported: "Lake": the POI is located on a lake waterbody surface. "River": the POI is located on a river channel. Note that the "Lake" type POI should be located within a lake subbasin and the "River" type POI should be located within a non-lake subbasin. The POI located in the wrong subbasin will be ignored and not added to the routing product. The "River" type POI will be linked to a non-lake subbasin that contains the POI. The "Lake" type POI will be linked to the lake subbasin that contains the POI.
- **gis_platform** (*string*) – is the parameter indicating which gis platform is used. Currenly, only "purepy" is allowed for this parameter.
- **clean_exist_pois** (*boolean*) – Indicates if the user wants to remove all existing POI in the input routing product. When it is true, all existing POI will be removed, and only the POI sites provided in the path_to_points_of_interest_points will be added to the input routing network.

Notes

This function does not return any values. If the user only wants to add POI to the routing product, they should set clean_exist_pois to False. This will keep the existing POI and add the provided POI to the routing product. If the user wants to remove all existing POI and add new POI, they should set clean_exist_pois to True. This will remove all existing POI from the routing product and then add the provided POI to the routing product. The modified hydrological routing network will be generated in the path_output_folder and will include the following files:

finalcat_info.shp
: [shapefile] The finalized hydrologic routing network. the GIS layer containing subbasin polygons which respect the lake inflow and outflow routing structures. This layer contains all the necessary information for hydrologic routing through the lake-river network.

finalcat_info_riv.shp
: [shapefile] The finalized hydrologic routing network. the GIS layer containing river network polylines in the routing network.

catchment_without_merging_lakes.shp
: [shapefile] The GIS layer containing subbasin polygons of an incomplete hydrologic routing network. In this incomplete hydrologic routing network subbasin polygons covered by the same lake are not merged into one lake subbasin yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

river_without_merging_lakes.shp
: [shapefile] The GIS layer containing river polylines of an incomplete hydrologic routing network. In this incomplete hydrologic routing network, the river polylines covered by the same lake are not merged into one river segment yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

sl_connected_lake.shp
: [shapefile] the GIS layer containing the lake polygons of lakes that are connected by the river_without_merging_lakes.shp

sl_non_connected_lake.shp
: [shapefile] the GIS layer containing the lake polygons of lakes that are not connected by the river_without_merging_lakes.shp

poi
: [shapefile] It is the point shapefile that represent the point of interest after snap to river network.

Examples

9.1.5 Generate HRUs

`basinmaker.basinmaker.postprocess.`**`Generate_HRUs`**(*path_output_folder*, *gis_platform*, *path_subbasin_polygon*, *path_landuse_info*, *path_soil_info*, *path_veg_info*, *prjected_epsg_code='EPSG:3573'*, *path_connect_lake_polygon='#'*, *path_non_connect_lake_polygon='#'*, *path_landuse_polygon='#'*, *path_soil_polygon='#'*, *path_vegetation_polygon='#'*, *path_other_polygon_1='#'*, *area_ratio_thresholds=[0, 0, 0]*, *path_to_dem='#'*)

This function is to generate HRU map based on subbasin polygon, lake polygon (optional), Land use polygon (optional), soil type polygon(optional), vegetation polygon (optional), and two other user defined polygons (optional).

Parameters

- `path_output_folder` (*string*) – is the folder path that stores generated outputs
- `gis_platform` (*string*) – is the parameter indicating which gis platform is used. It can be either "qgis" or "arcgis".
- `path_subbasin_polygon` (*string*) – is the path of the subbasin polygon, which is generated by BasinMaker.
- `path_landuse_info` (*string*) – Path to a csv file that contains landuse information, including the following attributes:

  Landuse_ID (integer) – the landuse ID in the landuse polygon,-1 for lake
  LAND_USE_C (string) – the landuse class name for each landuse type

- `path_soil_info` (*string*) – is the path to a csv file that contains soil information, including following attributes:

  Soil_ID (integer) – the soil ID in the soil polygon,-1 for lake
  SOIL_PROF (string) – the soil profile name for each soil profile type

- `path_veg_info` (*string*) – is the path to a csv file that contains vegetation information, including following attributes:

  Veg_ID (integer) – the vegetation ID in the vegetation polygon,-1 for lake
  VEG_C (string) – the vegetation class name for each vegetation Type

- `projected_epsg_code` (*string (optional)*) – is a EPSG code to indicate a projected coordinate system. If the routing network generated by basinmaker is under a projected system, please assgin the EPSG code of that projected system to projected_epsg_code. If the routing network generated by basinmaker is under a geographic system, please use EPSG code from any projected system. The coordinate system of routing network from basinmaker is determined by the coordinate system of input DEM.
- `path_connect_lake_polygon` (*string (Optional)*) – is the path to the connected lake's polygon
- `path_non_connect_lake_polygon` (*string (Optional)*) – is the path to the non connected lake's polygon
- `path_landuse_polygon` (*string (Optional)*) – is the path to the landuse polygon. when path_landuse_polygon is not provided. The Landuse ID in path_landuse_info should be 1: land, -1: lake
- `path_soil_polygon` (*string (Optional)*) – is the path to the soil polygon. when soil polygon is not provided. The Soil ID in path_soil_info should be the same as Landuse ID.

- **path_vegetation_polygon** (*string (Optional)*) – is the path to the vegetation polygon. when vegetation polygon is not provided. The Veg ID in path_veg_info should be the same as Landuse ID.
- **path_other_polygon_1** (*string (Optional)*) – is the path to the other polygon that will be used to define HRU, such as elevation band, or aspect.
- **DEM** (*string (optional)*) – is the path to a raster elevation dataset, that is used to calcuate average apspect, elevation and slope within each HRU. if no data is provided, subbasin averaged value will be used for each HRU.
- **area_ratio_thresholds** (*list (optional)*) – It is a list of HRU area thresthold for landuse, soil and other_1 layer respectively. In BasinMaker HRU calculation, each layer will firstly be overlaid to the subbasin map. Attributes falling into each subbasin with their area ratios (i.e., the intersected area to the subbasin area) smaller than the defined threshold values will then be dissolved into the largest part. For example, if forest area ratio in a subbasin, say subbasin #10, is 0.05, while we set the area threshold for land cover is 0.1. The forest polygons will then be dissolved to the largest land cover type in subbasin #10.

Notes

This function has no return values, but a HRU map saved in the path_output_folder

Examples

9.1.6 Generate Raven input files

basinmaker.basinmaker.postprocess.**Generate_Raven_Model_Inputs**(*path_output_folder*, *path_hru_polygon*, *aspect_from_gis*, *model_name='test'*, *subbasingroup_nm_channel=['Allsubbasins']*, *subbasingroup_length_channel=[-1]*, *subbasingroup_nm_lake=['AllLakesubbasins']*, *subbasingroup_area_lake=[-1]*)

This function is to generate Raven input files. A subfolder 'RavenInput' in path_output_folder is created to sotre resultant outputs.

Parameters

- **path_output_folder** (*string*) – is the folder path that stores generated Raven input fiels
- **path_hru_polygon** (*string*) – is path of the output HRU shapefile from BasinMaker which includes all required hydrologic attributes and parameters; Each row in the attribute table of this shapefile represent a HRU.
- **aspect_from_gis** (*string*) – is a string indicating how aspect of each HRU in path_hru_polygon is caculated. "grass" represent the aspect is calcuated by GRASS GIS; "arcgis" represent the aspect is calcuated by ArcGIS.
- **model_name** (*string (optional)*) – is the The Raven model base name.
- **subbasingroup_names_channel** (*list (optional)*) – is a list of names for subbasin groups, which are grouped based on channel length of each subbsin. Should at least has one name

- **subbasingroup_length_channel** (*list (optional)*) – is a list of channel length thresthold in meter, that divides subbasin into different groups. For example, [1,10,20] divides subbasins into four groups:

  1) group 1 with channel length range from (0,1];
  2) group 2 with channel length range from (1,10];
  3) group 3 with channel length range from (10,20];
  4) group 4 with channel length range from (20,Max channel length].

- **subbasingroup_name_lake** (*list (optional)*) – is a list of names for subbasin groups, which are grouped based on area of lakes in lake subbasins. Should at least has one name
- **subbasingroup_area_lake** (*list (optional)*) – is a list of lake area thresthold in m2, that divides subbasin into different groups. For example, [1,10,20] divides subbasins into four groups:

  1) group 1 with lake area range from (0,1];
  2) group 2 with lake are range from (1,10],
  3) group 3 with lake are range from (10,20],
  4) group 4 with lake are range from (20,max_lake_area].

Notes

The following ouput files will be generated in "<path_output_folder>/RavenInput"

modelname.rvh - contains subbasins and HRUs
Lakes.rvh - contains definition and parameters of lakes
channel_properties.rvp - contains definition and parameters for channels

Examples

9.2 BasinMaker - delineate lake-river routing product tools

In order to fully delineate a lake-river routing network from scratch, users need to sequentially apply five BasinMaker functions (ND1 to ND5) described in this section.

The overview of the workflow is summarized in the Figure below.

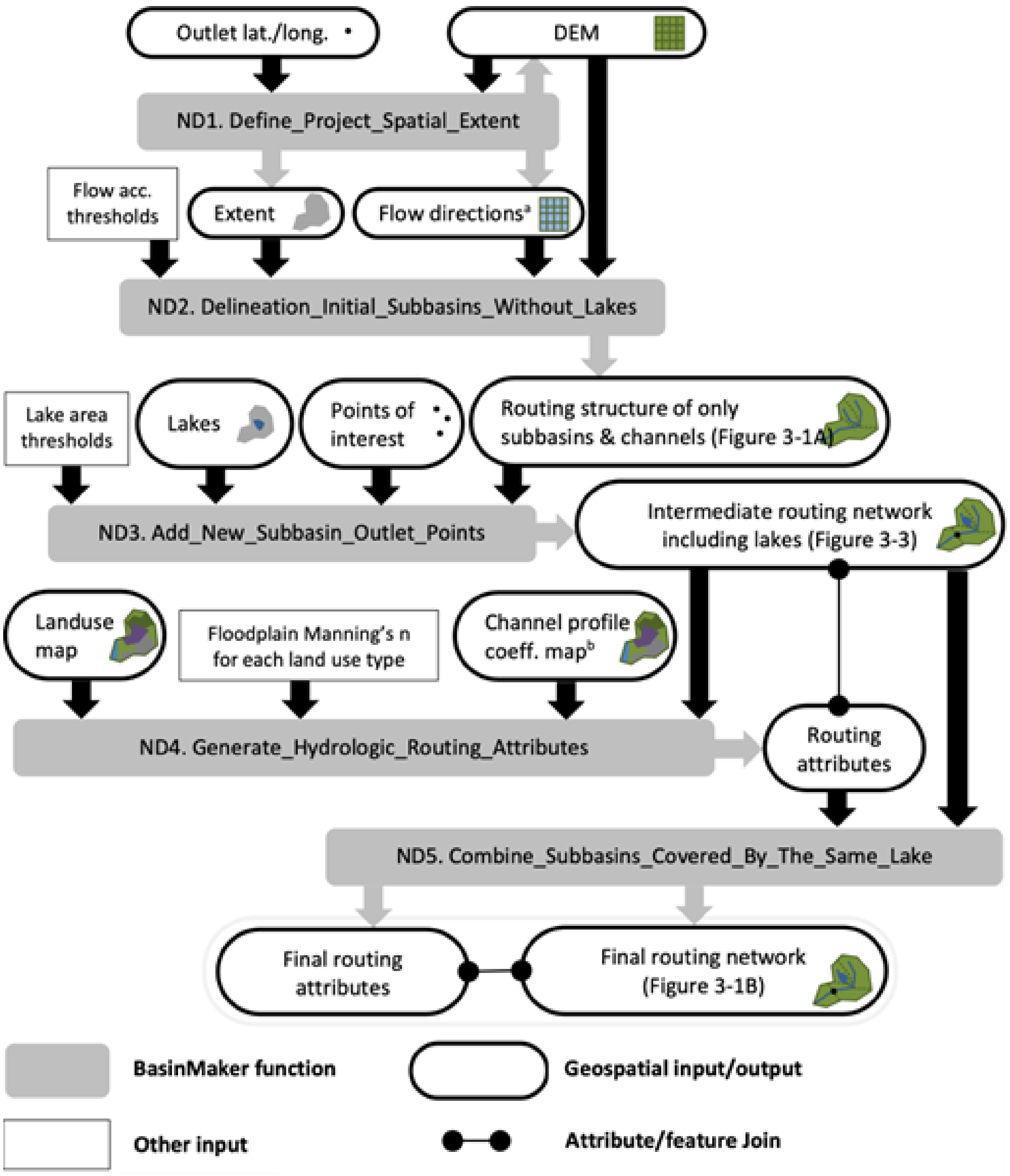

Outlet lat./long.
DEM
ND1. Define_Project_Spatial_Extent
Flow acc. thresholds
Extent
Flow directions^a
ND2. Delineation_Initial_Subbasins_Without_Lakes
Lake area thresholds
Lakes
Points of interest
Routing structure of only subbasins & channels (Figure 3-1A)
ND3. Add_New_Subbasin_Outlet_Points
Intermediate routing network including lakes (Figure 3-3)
Landuse map
Floodplain Manning's n for each land use type
Channel profile coeff. map^b
ND4. Generate_Hydrologic_Routing_Attributes
Routing attributes
ND5. Combine_Subbasins_Covered_By_The_Same_Lake
Final routing attributes
Final routing network (Figure 3-1B)
BasinMaker function
Geospatial input/output
Other input
Attribute/feature Join

Figure Caption: This is Figure 3 from han et al. (2023). Workflow, including inputs and outputs, of the BasinMaker network delineation mode functions for generating a hydrological routing network with lakes and rivers given a DEM and a lake polygon layer. Notes: 1. Flow direction raster dataset is optional user input here. 2. The channel profile coefficient map to define bankfull channel widths and depths across the extent can alternatively be replaced by coefficients that are constant across the spatial extent of the project.

9.2.1 Define project spatial extent

`basinmaker.basinmaker.delineate.`**`Define_Project_Spatial_Extent`**(*mode*, *path_to_dem_input*, *watershed_outlet_coordinates=[-1, -1]*, *path_to_spatial_extent_polygon='#'*, *buffer_distance=0.0*, *path_to_hydrobasin_polygon='#'*, *hydrobasin_id_of_watershed_outlet=-1*)

This function is to define project spatial extent (PSE). Domain ouside of the PSE is not processed by BasinMaker functions.

Parameters

- **`mode`** (*`string (required)`*) – is a string indicating which to define PSE

  ’using_dem’ : the extent of input dem is used

  ’using_hybasin’ : the extent is defined by subbasins that are drainage to the provided watershed outlet subbasins ID in HydroBASINS product

  ’using_outlet_pt’ : the extent is defined by the watershed generated from input dem and the watershed outlet coordinates

  ’using_provided_ply’ : the extent of provided polygon is used

- **`path_to_dem_input`** (*`string (required)`*) – is the path to input dem
- **`watershed_outlet_coordinates`** (*`list (optional)`*) – is a list that indicate the outlet coordinates of the region of interest in [lat, lon]. It is needed when mode = ‘using_outlet_pt’.
- **`path_to_spatial_extent_polygon`** (*`string (optional)`*) – is the path of a polygon shapefile, the extent of which will be used as PSE.
- **`buffer_distance`** (*`float (optional)`*) – is a float number to enlarge the PSE. It is needed when mode = ‘using_hybasin’ or mode = ‘using_provided_ply’. It is the distance around the PSE from HydroBASINS or provided polygons that will be buffered. The unit is the same with the spatial unit of input DEM.
- **`path_to_hydrobasin_polygon`** (*`string (optional)`*) – is a path to the HydroBASINs product. It is needed when mode = ‘using_hybasin’
- **`hydrobasin_id_of_watershed_outlet`** (*`int (optional)`*) – is a HydroBASINS subbasin ID of the watershed outlet. It is needed when mode = ‘using_hybasin’

Notes

Outputs are following files in GRASS GIS database loacated in os.path.join(path_working_folder,’grassdb’)

MASK.*
: [raster/shp] it is a mask raster stored in grass database, which indicate the PSE.

dem.*
: [raster/shp] it is a dem raster stored in grass database, which is has the same extent with MASK.

Examples

9.2.2 Delineate routing structure without lakes

`basinmaker.basinmaker.delineate.`**`Delineation_Initial_Subbasins_Without_Lakes`**(*fac_thresold*, *mode='using_dem'*, *path_flow_dirction_in='#'*, *max_memroy=4096*)

Function that used to generate a initial routing structure with user provied flow accumulation thresthold without considering lake.

Parameters

- **`fac_thresold`** (*`float`*) – is the flow accumulation thresthold, used to determine subbsains and river network. Increasing of this paramter will increase the size of generated subbasins, reduce the number subbasins and reduce the number of generated stream reaches
- **`mode`** (*`string (required)`*) – is a string indicate which dataset will be used to delineate watershed.

  'using_dem' : dem is used for initial subbasin delineation

  'using_fdr' : flow direction data is used for subbasin delineation

- **`path_flow_dirction`** (*`string (optional)`*) – is a path indicating the path of flow direction input dataset only needed when mode = 'using_fdr'
- **`max_memroy`** (*`integer`*) – is the maximum memeory that allow to be used in MB.

Notes

Outputs are following files in GRASS GIS database loacated in os.path.join(path_working_folder,'grassdb')

fdr_grass
[raster] is a raster represent flow direction dataset, which is using 1 - 8 to represent different directions

fdr_arcgis
[raster] is a raster represent flow direction dataset, which is using 1,2,4,...64,128 to represent different directions

str_v
[vector] is a river network in vector format

str_r
[raster] is a river network in raster format

cat_no_lake
[raster] is the raster represent the delineated subbasins without considering lakes

acc
[raster] is the raster represent the flow accumulation

Examples

9.2.3 Add lake control points and points of interest

basinmaker.basinmaker.delineate.**Add_New_Subbasin_Outlet_Points**(*path_lake_polygon='#', lake_attributes=[], connected_lake_area_thresthold=0, non_connected_lake_area_thresthold=0, path_point_of_interest='#', point_of_interest_attributes=[], max_memroy=4096*)

Update the subbasin delineation result by adding lake inflow and outflow points and observation gauges as a new subbasin outlets. The output is not the final delineation result. because:

1) Hydrologcial related attributes for each subbasin are not calcuated yet.

2) Some lakes may cover several subbasins. The output needs to be finalized by combing those subbasins with the same lake as one subbasin only.

Parameters

- **path_lake_polygon** (*string (optional)*) – is a path of the lake polygon shapefile
- **lake_attributes** (*list (optional)*) – the columns names in the input lake polygon that indicate following items (mandatory). It is needed only when path_lake_polygon_in is provided. Columns (2-4 in following list) in lake poylon can be fill with any number, when these infomation is not avaiable.

  1) column name for the unique Id of each lake, datatype is integer

  2) column name for type of the lake, datatype is integer

  3) column name for the volume of lakes in km3, datatype is float

  4) column name for the average depth of lakes in m, datatype is float

  5) column name for the area of lakes in km2, datatype is float

- **connected_lake_area_thresthold** (*float (optional)*) – is a lake area thresthold for connected lakes in km2. Connected lake with lake area below this value will not be considerd
- **non_connected_lake_area_thresthold** (*float (optional)*) – is a lake area thresthold for non-connected lakes in km2 Non connected lake with lake area below this value will not be considered
- **path_point_of_interest** – is a path of the point shapefile that indicate points of interest,which can include different observation gauges.
- **point_of_interest_attributes** (*list (optional)*) – the columns names in the point of interest shapefile that indicate following items (mandatory). It is needed only when path_point_of_interest is provided. Columns (2-4 in following list) in point shapefile can be fill with any value, when these infomation is not avaiable.

  1) column name for the unique Id of each observation point, datatype is integer

  2) column name for the unique name of each observation point, datatype is string

3) column name for the drainage area of each observation point in km3, datatype is float

4) column name for the source of the observation point: 'CA' for observation in canada; 'US' for observation in US, or any user-provided names, datatype is string

- **max_memroy** (*integer (optional)*) – is the maximum memeory that allow to be used in MB.

Notes

Output raster and vector files that will be used by next step are list as following. All files are saved in GRASS GIS database loacated in os.path.join(path_working_folder,'grassdb')

selected_lakes
[raster] it is a raster represent all lakes that are selected by two lake area threstholds

sl_nonconnect_lake
[raster] it is a raster represent all non connected lakes that are selected by lake area threstholds

sl_connected_lake
[raster] it is a raster represent all connected lakes that are selected by lake area threstholds

river_without_merging_lakes
[raster/vector] it is the updated river segment for each subbasin

catchment_without_merging_lakes
[raster/vector] it is a raster represent updated subbasins after adding lake inflow and outflow points as new subbasin outlet.

snapped_obs_points
[raster/vector] it is a name of the point gis file represent successfully sanpped point of interest points

Examples

9.2.4 Guidance on Points of Interest input layer preparation

For those running BasinMaker with QGIS & GRASS, points of interest will be snapped by this function automatically to the closest delineated river channel and as such, no input layer preparation is strictly required. However, in our experience this auto-snapping approach is only modestly successful. Please inspect snapped results carefully. If users ensure that points of interest associated with lake levels are located within a lake polygon, then this function will include the point of interest at the lake (and lake subbasin) outlet.

For those running BasinMaker in ArcGIS pro, the user needs to carefully prepare the points of interest shapefile as input. Specifically, the location of each non-lake point of interest should be carefully snapped to the the appropriate eventual delineated river channel. To do this, we recommend users build a temporary channel raster and snap to that. We also recommend users ensure that points of interest associated with lake levels are located within a lake polygon. These POI will be ignored by this function and must be added separately from this function using the function called "Add_Point_Of_Interest_Sites_In_Routing_Product".

A robust new function that can help users automatically snap points of interest to delineated river channels/lakes is coming in the next version of BasinMaker.

9.2.5 Add hydrology related attributes

`basinmaker.basinmaker.delineate.`**`Generate_Hydrologic_Routing_Attributes`**(*path_output_folder*, *projected_epsg_code='EPSG:3573'*, *path_bkfwidthdepth_polyline='#'*, *bkfwd_attributes=[]*, *k=-1*, *c=-1*, *path_landuse='#'*, *path_landuse_and_manning_n_table='#'*, *lake_attributes=[]*, *point_of_interest_attributes=[]*)

Calculate hydrological paramters for each subbasin.

Parameters

- **`path_output_folder`** (*`string`*) – The path to a folder to save outputs
- **`projected_epsg_code`** (*`string (optional)`*) – is a EPSG code to indicate a projected coordinate system. If the routing network generated by basinmaker is under a projected system, please assgin the EPSG code of that projected system to projected_epsg_code. If the routing network generated by basinmaker is under a geographic system, please use EPSG code from any projected system. The coordinate system of routing network from basinmaker is determined by the coordinate system of input DEM.
- **`path_bkfwidthdepth_polyline`** (*`string (optional)`*) – is a path of the polyline shapefile that contains bankfull width (w) and depth (d) data.Following the methodology in Andreadis et al. (2013), the w, and d of each subbasin can be calculated by these two equation w= 7.2 Q**0.5 and d=0.27Q**0.39 , respectively. And the the bankfull discharge Q of each subbasin can be estimated using the relationship between Q and the drainage area (DA in [km2]), which is the Q=k×DA**c. If the bankfull width and depth polyline data is provided, basinmaker will use it to estimate the k and c for the entire watershed.
- **`bkfwd_attributes`** (*`list (optional)`*) – the columns names that indicate following items (mandatory).It is only needed with path_bkfwidthdepth is provided

  1) column name for the Bankfull width in m;
  2) column name for the Bankfull depth in m;
  3) column name for the annual mean discharge in m3/s;
  4) column name for the drainage area in km2;

- **`k`** (*`float (optional)`*) – the coefficient in Q=k×DA**c.if k and c is provided, the path_bkfwidthdepth_polyline will not be used
- **`c`** (*`float (optional)`*) – the coefficient in Q=k×DA**c.if k and c is provided, the path_bkfwidthdepth_polyline will not be used
- **`path_landuse`** (*`string (optional)`*) – is a path of the landuse raster.It is used to estimate the floodplain Manning's coefficient. Require the same projection with the DEM data and formatted in ".tif".
- **`path_landuse_and_manning_n_table`** (*`string (optional)`*) – is a path of the table in '.csv' format.The table describe the floodplain Manning's coefficient correspond to a given landuse type. The table should have two columns:

RasterV: is the landuse value in the landuse raster for each land use type

MannV: is the roughness coefficient value for each landuse type.

- **lake_attributes** (*list (optional)*) – the columns names in the input lake polygon that indicate following items (mandatory). It is needed only when path_lake_polygon_in is provided. Columns (2-4 in following list) in lake poylon can be fill with any number, when these infomation is not avaiable.

  1) column name for the unique Id of each lake, datatype is integer
  2) column name for type of the lake, datatype is integer
  3) column name for the volume of lakes in km3, datatype is float
  4) column name for the average depth of lakes in m, datatype is float
  5) column name for the area of lakes in km2, datatype is float

- **point_of_interest_attributes** (*list (optional)*) – the columns names in the point of interest shapefile that indicate following items (mandatory). It is needed only when path_point_of_interest is provided. Columns (2-4 in following list) in point shapefile can be fill with any value, when these infomation is not avaiable.

  1) column name for the unique Id of each observation point, datatype is integer
  2) column name for the unique name of each observation point, datatype is string
  3) column name for the drainage area of each observation point in km3, datatype is float
  4) column name for the source of the observation point: 'CA' for observation in canada; 'US' for observation in US, or any user-provided names, datatype is string

Notes

Five vector files will be generated in the output folder. these files can be further finalized as hydrologcial routing network by function "combine_catchments_covered_by_the_same_lake" or be used as input for BasinMaker post processint tools. The attributes included in each GIS file can be found in http://hydrology.uwaterloo.ca/basinmaker/index.html

catchment_without_merging_lakes.shp
[shapefile] The GIS layer containing subbasin polygons of an incomplete hydrologic routing network. In this incomplete hydrologic routing network subbasin polygons covered by the same lake are not merged into one lake subbasin yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

river_without_merging_lakes.shp
[shapefile] The GIS layer containing river polylines of an incomplete hydrologic routing network. In this incomplete hydrologic routing network, the river polylines covered by the same lake are not merged into one river segment yet. This incomplete hydrologic routing network is only intended as input to customize the routing network with our BasinMaker GIS toolbox (for example by defining new lake area thresholds and/or a new catchment minimum drainage area threshold)

sl_connected_lake.shp
[shapefile] the GIS layer containing the lake polygons of lakes that are connected by the river_without_merging_lakes.shp

sl_non_connected_lake.shp
[shapefile] the GIS layer containing the lake polygons of lakes that are not connected by the river_without_merging_lakes.shp

poi
[shapefile] It is the point shapefile that represent the point of interest after snap to river network.

Examples

9.2.6 Combine catchment covered by the same lake

`basinmaker.basinmaker.delineate.`**`Combine_Subbasins_Covered_by_The_Same_Lake`**`(`*routing_product_folder*, *gis_platform='qgis'*`)`

Finalize a incomplete hydrologic routing network by merging subbasin polygons that are covered by the same lake.

Parameters

- **`routing_product_folder`** (*`string`*) – is the folder where the input routing product is stored
- **`gis_platform`** (*`string`*) – It is the parameter indicate which gis platform is used. It can be either “qgis” or “arcgis”.

Notes

This function has no return values, two vector files will be generated in the routing_product_folder

finalcat_info.shp
[shapefile] The finalized hydrologic routing network. the GIS layer containing subbasin polygons which respect the lake inflow and outflow routing structures. This layer contains all the necessary information for hydrologic routing through the lake-river network.

finalcat_info_riv.shp
[shapefile] The finalized hydrologic routing network. the GIS layer containing river network polylines in the routing network.

Examples

CHAPTER

TEN

USEFUL LINKS

- Raven hydrological modeling framework.
- RavenR.
- OSTRICH - Optimization Software Toolkit.
- PAVICS.
- Ravenpy.

INDEX