QRaven Documentation

Release 0.1

Francis Lapointe

Mar 19, 2024

CONTENTS

You will find all the information about QRaven, from its installation to contributing to its development. If you find any errors or typos, please report them.

CHAPTER

ONE

INSTALLATION

1.1 Dependencies

To use QRaven, first install a recent version of QGIS (3.20 and above): https://qgis.org/en/site/forusers/download.html

Some features, such as the BasinMaker tools and the GridWeight Generator, require a containerization software to be installed. If you are not planning on using those features, you can skip this step. Otherwise, you can pick between Podman and Docker.

Get Podman : https://podman.io/getting-started/installation

Get Docker : https://docs.docker.com/get-docker

Warning: Linux users who choose Docker must run it as a non root user. This can be done with the following command:

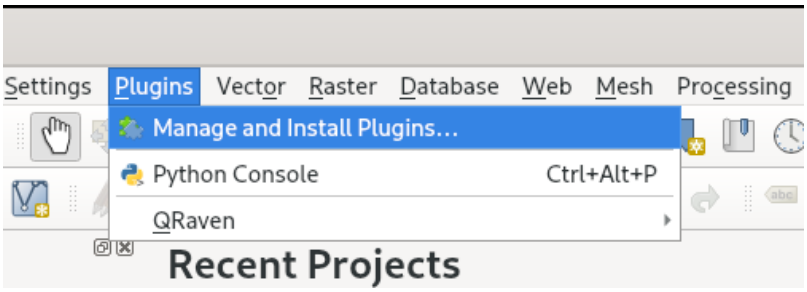
```
sudo usermod -aG docker $USER
```

For more information, go to: https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user

1.2 Plugin installation

1.2.1 Method 1 (recommended)

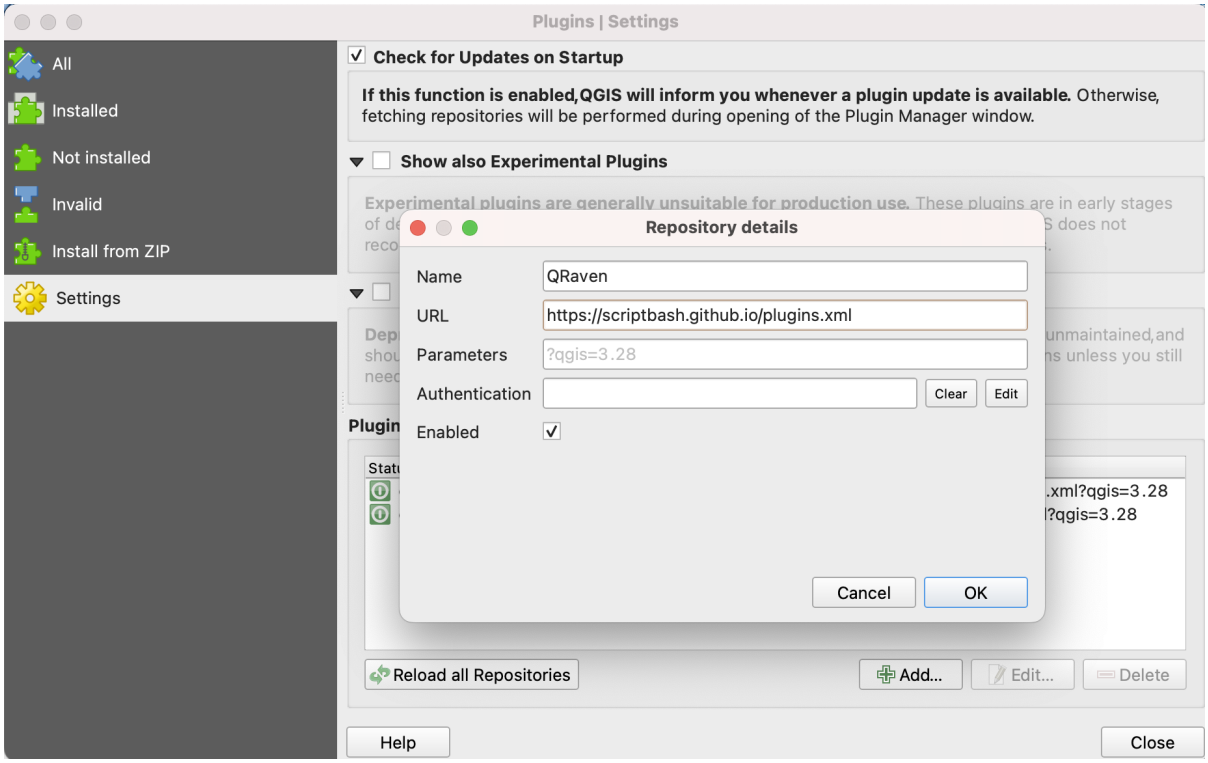
Open QGIS and go to the "Plugins" menu. Click on "Manage and Install Plugins".



Click on the "Settings" tab and click on the "Add" button to add a new plugin repository.

In the window that just opened, enter a name, such as "QRaven", enter the link below in the URL field and leave the remaining options to their defaults:

https://scriptbash.github.io/plugins.xml



After connecting to QRaven's repository, click on the "All" menu and search for QRaven. Click on QRaven and click on "Install".

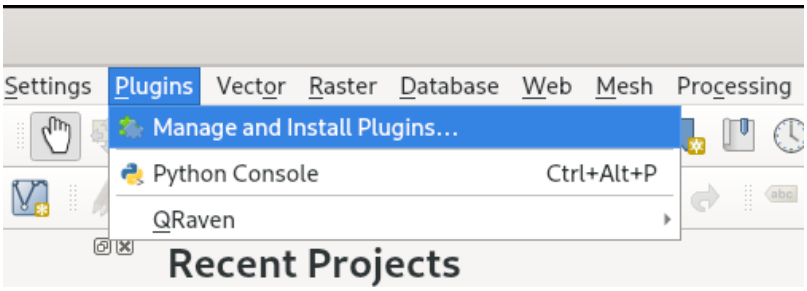
To update QRaven, simply head back to the plugins manager, search for QRaven and click on "Upgrade Plugin".

1.2.2 Method 2 (manual)

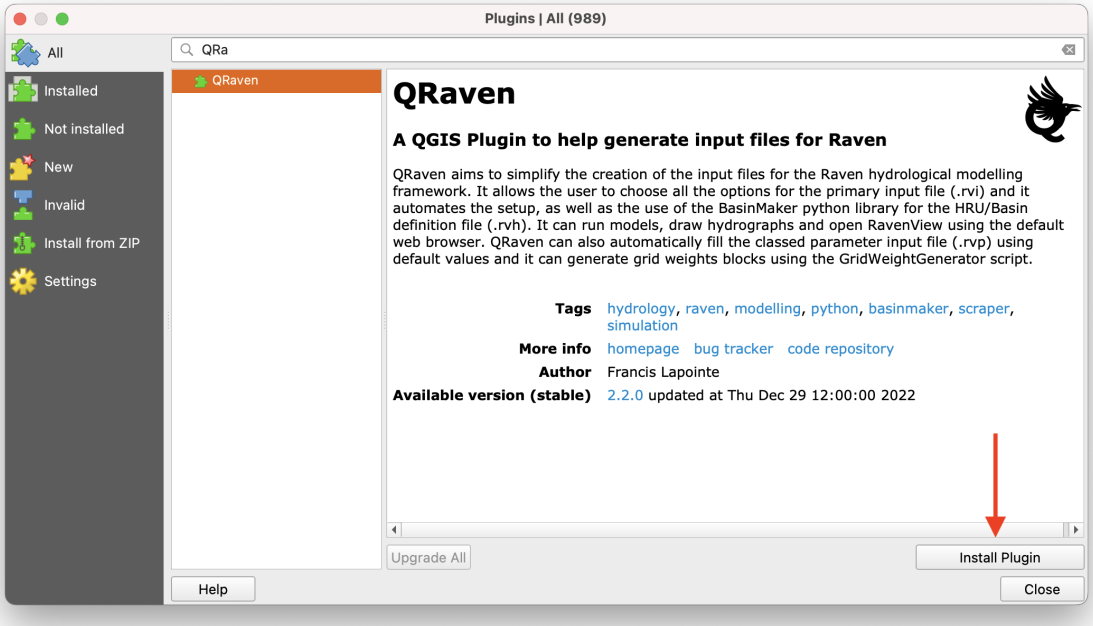
Download the latest release of QRaven here: https://github.com/Scriptbash/QRaven/releases/latest/download/qraven.zip

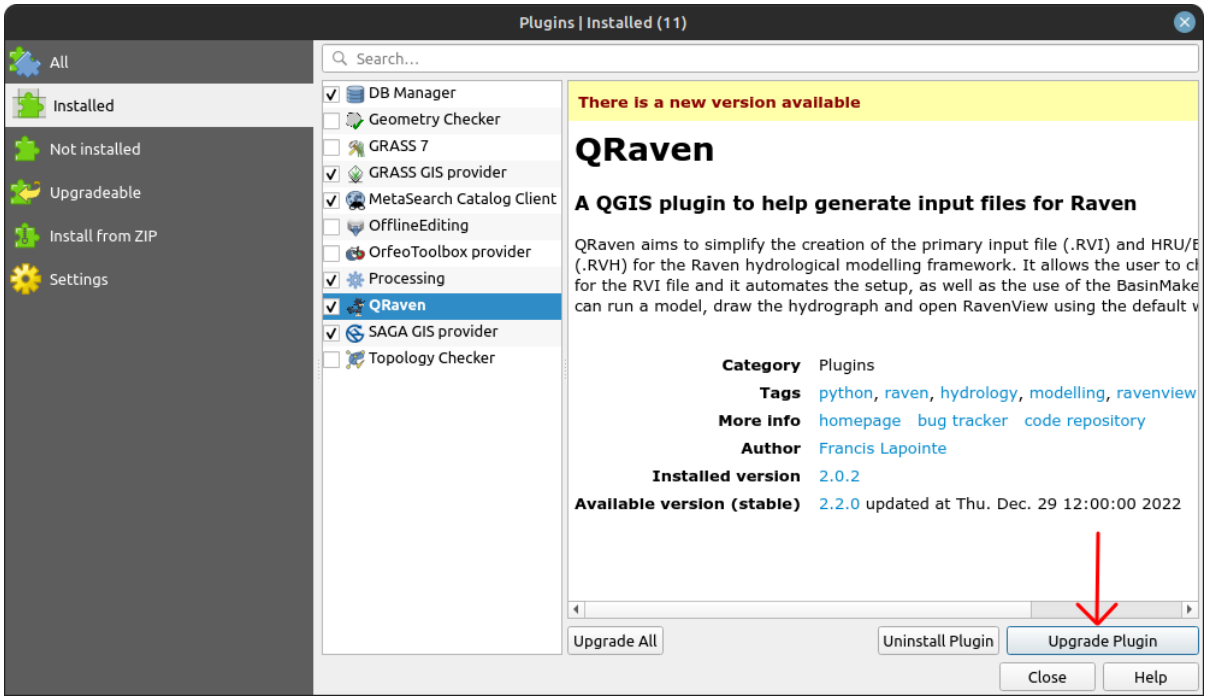
Alternatively, you can view all of QRaven versions here : https://github.com/Scriptbash/QRaven/releases

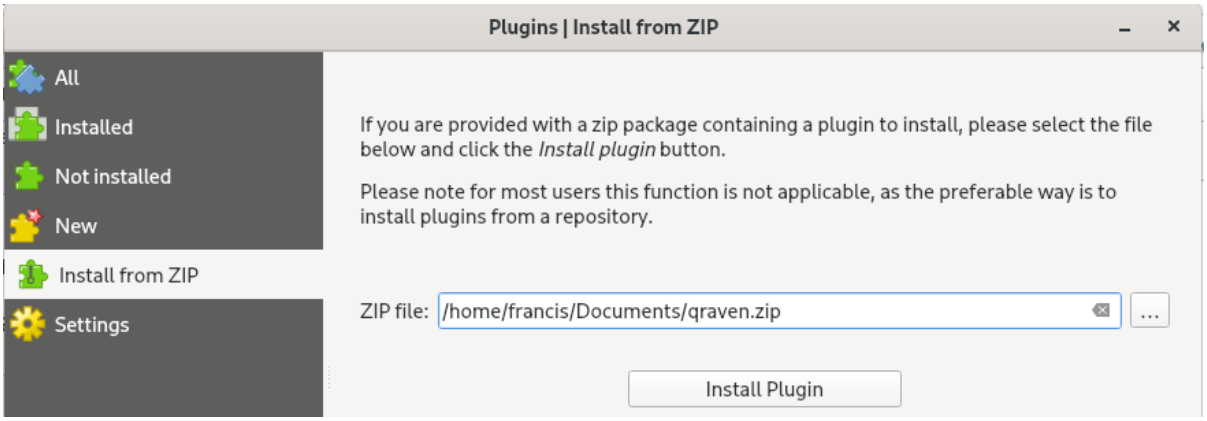
Next, open QGIS and go to the "Plugins" menu. Click on "Manage and Install Plugins".



Finally, click on "Install from ZIP", select the downloaded qraven.zip file and click on "Install".


Plugins | All (989)
All
Installed
Not installed
New
Invalid
Install from ZIP
Settings
QRa
QRaven
QRaven
A QGIS Plugin to help generate input files for Raven
QRaven aims to simplify the creation of the input files for the Raven hydrological modelling framework. It allows the user to choose all the options for the primary input file (.rvi) and it automates the setup, as well as the use of the BasinMaker python library for the HRU/Basin definition file (.rvh). It can run models, draw hydrographs and open RavenView using the default web browser. QRaven can also automatically fill the classed parameter input file (.rvp) using default values and it can generate grid weights blocks using the GridWeightGenerator script.
Tags hydrology, raven, modelling, python, basinmaker, scraper, simulation
More info homepage bug tracker code repository
Author Francis Lapointe
Available version (stable) 2.2.0 updated at Thu Dec 29 12:00:00 2022
Upgrade All
Install Plugin
Help
Close


Plugins | Installed (11)
All
Installed
Not installed
Upgradeable
Install from ZIP
Settings
Search...
DB Manager
Geometry Checker
GRASS 7
GRASS GIS provider
MetaSearch Catalog Client
OfflineEditing
OrfeoToolbox provider
Processing
QRaven
SAGA GIS provider
Topology Checker
There is a new version available
QRaven
A QGIS plugin to help generate input files for Raven
QRaven aims to simplify the creation of the primary input file (.RVI) and HRU/E
(.RVH) for the Raven hydrological modelling framework. It allows the user to cl
for the RVI file and it automates the setup, as well as the use of the BasinMake
can run a model, draw the hydrograph and open RavenView using the default v
Category Plugins
Tags python, raven, hydrology, modelling, ravenview
More info homepage bug tracker code repository
Author Francis Lapointe
Installed version 2.0.2
Available version (stable) 2.2.0 updated at Thu. Dec. 29 12:00:00 2022
Upgrade All
Uninstall Plugin
Upgrade Plugin
Close
Help



To update QRaven, you will need to download the latest .zip file and repeat the same steps as above.

Note: The plugin will look for an update each time QGIS is started. If it finds one, you will have a notification in the notification bar and inside the "Settings" menu of QRaven.

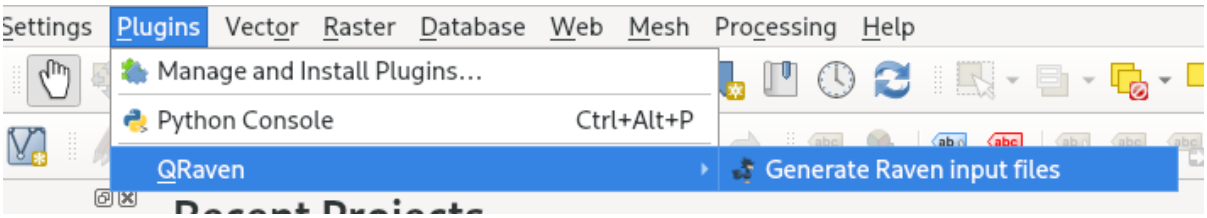
CHAPTER

TWO

HOW TO USE QRAVEN

2.1 Open QRaven

Click on the QRaven icon  in your toolbar or go in the "Plugins" menu, select the QRaven option and click on "Generate Raven input files"



You will have seven main menus

- *Raven RVI*
- *BasinMaker RVH*
- *GridWeights*
- *Streamflow*
- *GIS*
- *Run Model*
- *Settings*

2.2 Create a RVI file

QRaven all started with this feature in mind. This tab can be used to create a rvi file from scratch or from a template. The templates available are UBCWM, HBV-EC, HBV-Light, GR4J, Canadian Shield, MOHYSE, HMETS, HYPR, HYMOD and AWBM. Those models structures were taken directly from the Raven official documentation.

This section of the plugin is pretty straight forward to use. Check/uncheck/ options, select entries in drop down lists, etc. When ready, click on the "Write" button to write the rvi file. Otherwise, you can click on "Reset" to revert the options back to their default values.

The parameters are separated in different sections. You can find an overview of the options below;

- **Model info**

  - You can find templates buttons. Simply click on the one needed and the interface will load its configuration.
  - Basic information like the name of the model, the start/end date, time step, etc.

- **Sim. parameters**
  - Simulation parameters such as the catchment route, the routing method, evaporation, etc.
- **Hydro. processes**
  - A table that allows to set up the hydrologic processes.
  - Click on "Add process" to add a new row or "Remove selected process" to remove the selected row.
  - After selecting a process, the available algorithms for that process will be inside the algorithm drop down list. The from and to compartments drop down list will contain the available compartments for the select process and algorithm.
- **Transport cmd**
  - This work the same way as the hydrologic processes, but for the transport commands.
- **Optional I/O**
  - All kind of optional options like CreateRVPTemplate, evaluation metrics, debug mode, etc.
- **Custom output**
  - This is similar to the hydrologic processes table, but for custom outputs.

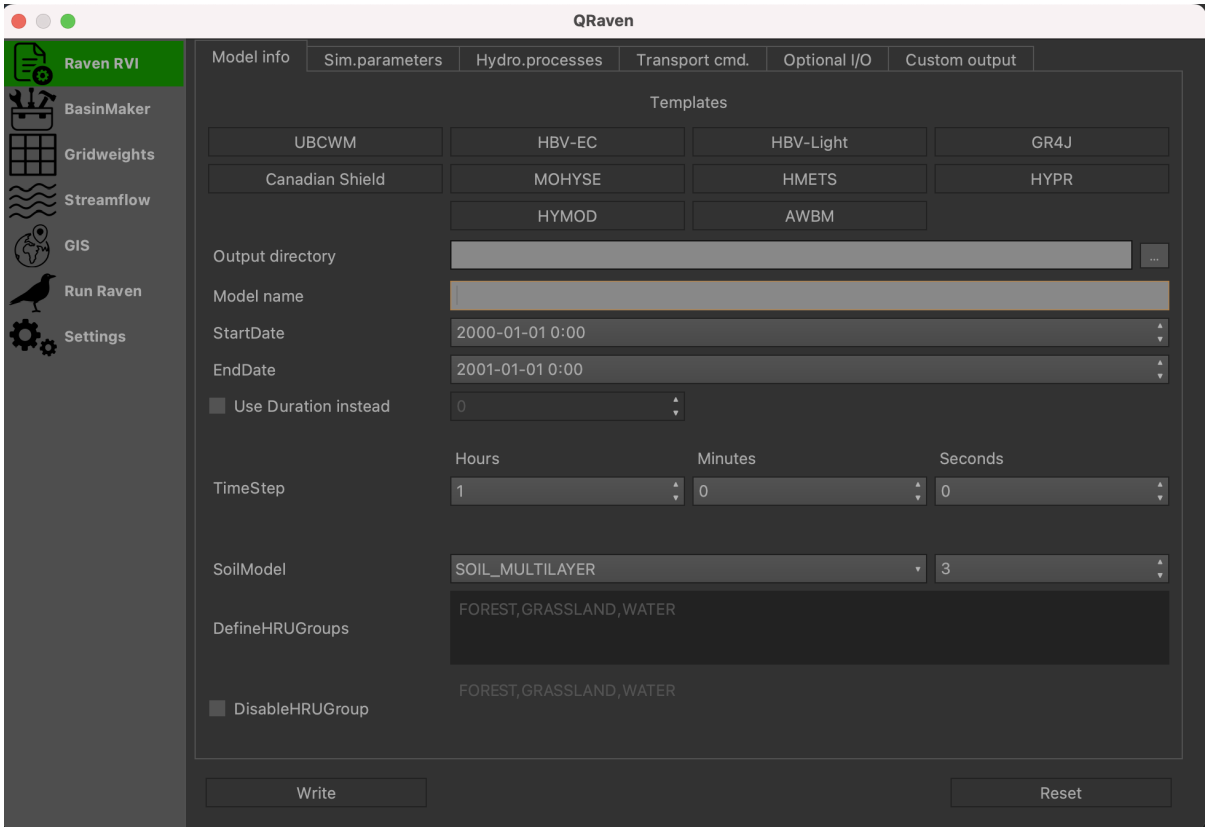
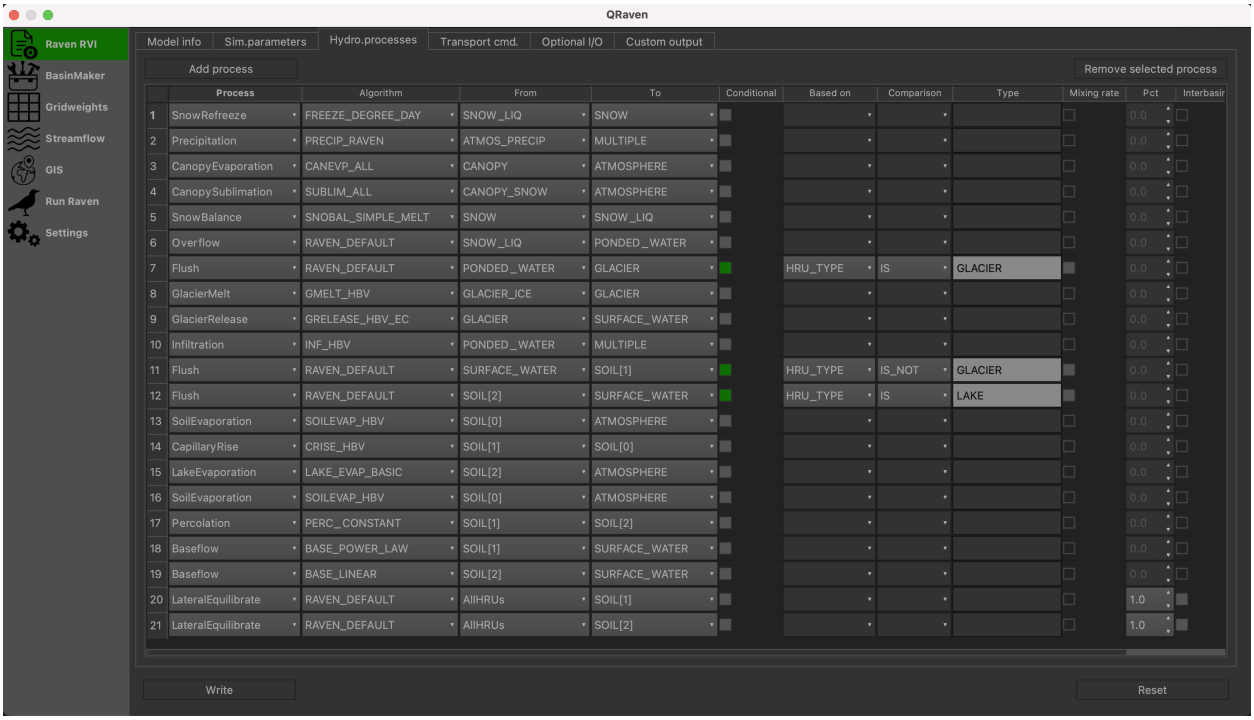


Fig. 1: The model info section.



	Process	Algorithm	From	To	Conditional	Based on	Comparison	Type	Mixing rate	Pct	Interbasir
1	SnowRefreeze	FREEZE_DEGREE_DAY	SNOW_LIQ	SNOW						0.0	
2	Precipitation	PRECIP_RAVEN	ATMOS_PRECIP	MULTIPLE						0.0	
3	CanopyEvaporation	CANEVP_ALL	CANOPY	ATMOSPHERE						0.0	
4	CanopySublimation	SUBLIM_ALL	CANOPY_SNOW	ATMOSPHERE						0.0	
5	SnowBalance	SNOBAL_SIMPLE_MELT	SNOW	SNOW_LIQ						0.0	
6	Overflow	RAVEN_DEFAULT	SNOW_LIQ	PONDED_WATER						0.0	
7	Flush	RAVEN_DEFAULT	PONDED_WATER	GLACIER		HRU_TYPE	IS	GLACIER		0.0	
8	GlacierMelt	GMELT_HBV	GLACIER_ICE	GLACIER						0.0	
9	GlacierRelease	GRELEASE_HBV_EC	GLACIER	SURFACE_WATER						0.0	
10	Infiltration	INF_HBV	PONDED_WATER	MULTIPLE						0.0	
11	Flush	RAVEN_DEFAULT	SURFACE_WATER	SOIL[1]		HRU_TYPE	IS_NOT	GLACIER		0.0	
12	Flush	RAVEN_DEFAULT	SOIL[2]	SURFACE_WATER		HRU_TYPE	IS	LAKE		0.0	
13	SoilEvaporation	SOILEVAP_HBV	SOIL[0]	ATMOSPHERE						0.0	
14	CapillaryRise	CRISE_HBV	SOIL[1]	SOIL[0]						0.0	
15	LakeEvaporation	LAKE_EVAP_BASIC	SOIL[2]	ATMOSPHERE						0.0	
16	SoilEvaporation	SOILEVAP_HBV	SOIL[0]	ATMOSPHERE						0.0	
17	Percolation	PERC_CONSTANT	SOIL[1]	SOIL[2]						0.0	
18	Baseflow	BASE_POWER_LAW	SOIL[1]	SURFACE_WATER						0.0	
19	Baseflow	BASE_LINEAR	SOIL[2]	SURFACE_WATER						0.0	
20	LateralEquilibrate	RAVEN_DEFAULT	AllHRUs	SOIL[1]						1.0	
21	LateralEquilibrate	RAVEN_DEFAULT	AllHRUs	SOIL[2]						1.0	

Fig. 2: The hydrologic process table.

2.3 Create a RVH file

Warning: The Docker daemon must be running to use this feature. Podman users don't need to worry, as Podman is daemonless.

*To-do

2.4 Associate a NetCDF grid to the HRUs

Warning: The Docker daemon must be running to use this feature. Podman users don't need to worry, as Podman is daemonless.

1. **NetCDF file** : The NetCDF file to process (inluding the file extension).
2. **Shapefile attribute** (Optional) : Only needed if the Netcdf file is a shapefile. It is the attribute containing the numbering of the subbasins.
3. **Dim name longitude (x)** : The dimension name for the longitude (e.g. rlon).
4. **Dim name latitude (y)** : The dimension name for the latitude (e.g. rlat).
5. **Var name longitude (x)** : The variable name for the longitude (e.g. lon).
6. **Var name latitude (y)** : The variable name for the latitude (e.g. lat).

7. **HRUs file** : The final shapefile created by the BasinMaker tools.
8. **Use gauge ID** and **Use subbasins ID** : Either use a gauge ID or subbasins ID. The ID must be entered manually in the field below these options.
9. **Output path** : The path and file name of the results.

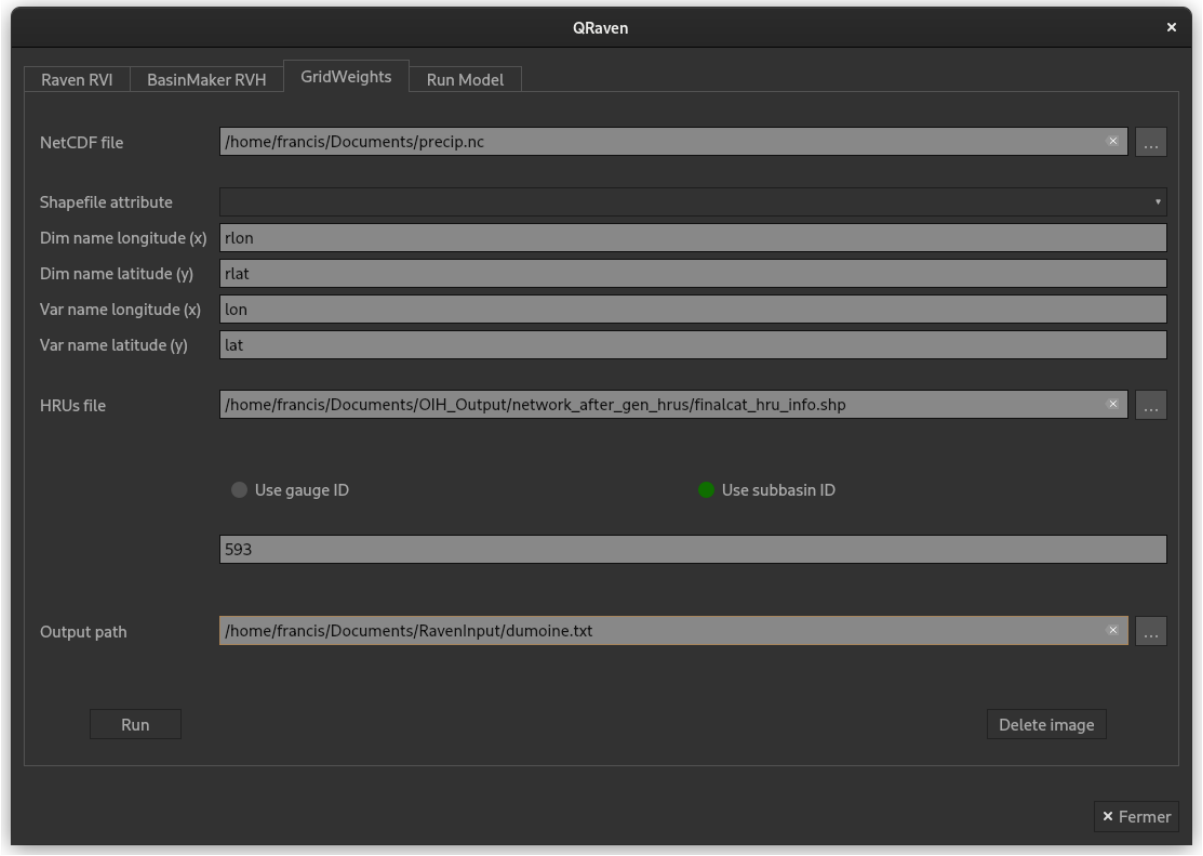


Fig. 3: Example of the gridweights generator interface.

2.5 Download streamflow data

QRaven can fetch hydrometric data from two providers at this time, which are the direction principale des prévisions hydriques et de la cartographie (DPPHC) and the Water office. Not only can it fetch data automatically, it can also generate rvt files from the data. Only flow data is supported, level data is unsupported.

Both data scrapers work the same way, but their search criterias differ a little bit. This documentation will only cover the Water office scraper.

- **Search a station**
    1. Select either “Station name” or “Province”.
    2. If “Station name” is selected, type in the full or patial name of the station. If “Province” is selected, select a province in the drop down list.
    3. Use the “Regulation” and “Station status” drop down lists to refine your search if needed.
    4. Click on “Search”.

5. Results will show up in the text area above the “Search” button.

Note: The station ID is always the first information in the search results. Simply copy/paste an ID into the “Station ID” field in the download section.

- **Download hydrometric data**

1. In the “Station ID” field, type in the hydrometric station ID from which the data will be downloaded.
2. In the “Output file” field, select a directory and name for the output file. The extension will always be “.rvt”.
3. Click the “Download” button.

If you have already downloaded data from one of the two providers, use the following option.

- **Process a local file**

1. In the “Input file” field, select the file you want to process.
2. In the “Output file” field, select a directory and a name for the rvt file.
3. Click on the “Process” button.

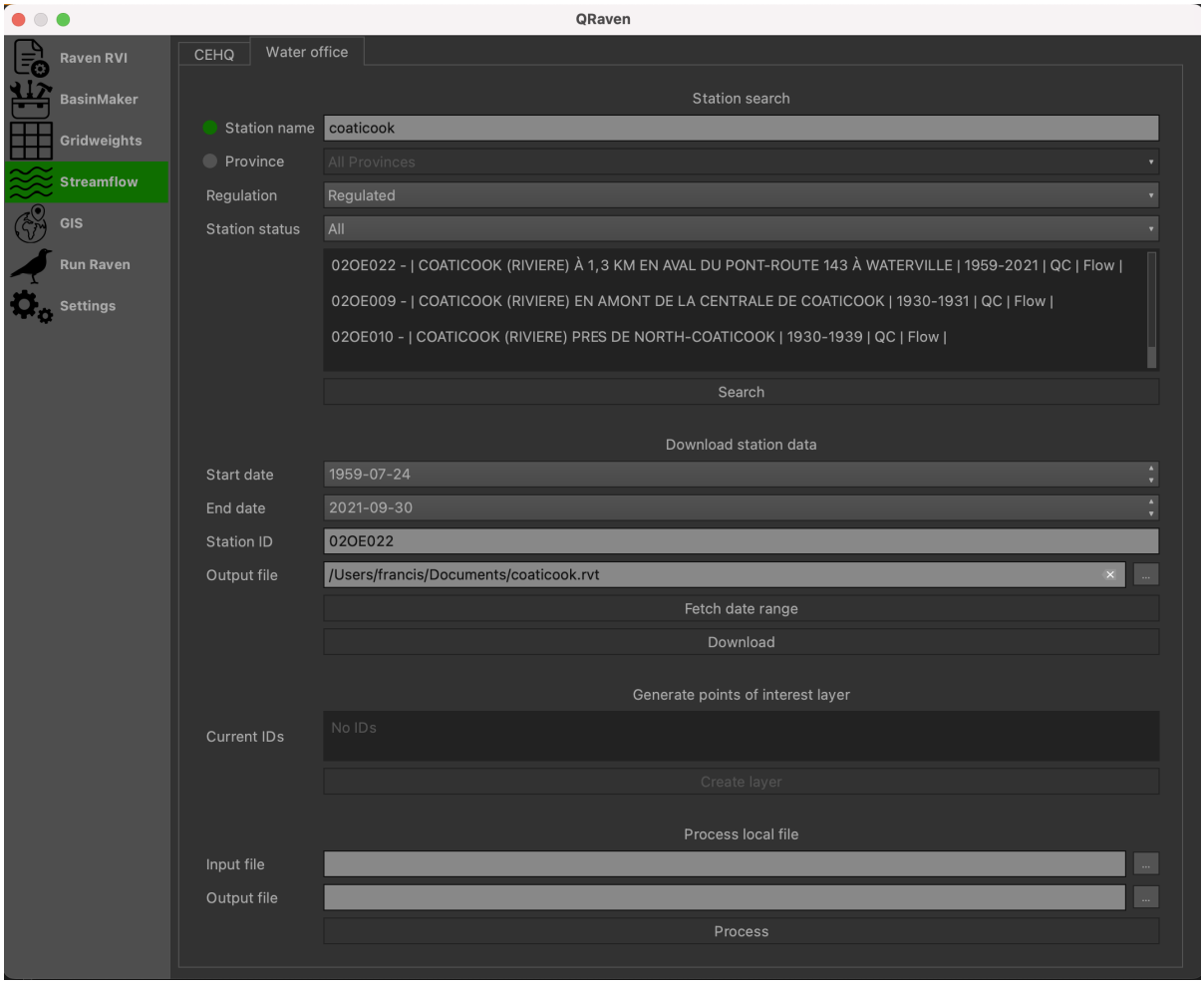


Fig. 4: Example of the Water office UI

2.6 Download GIS data

To write.

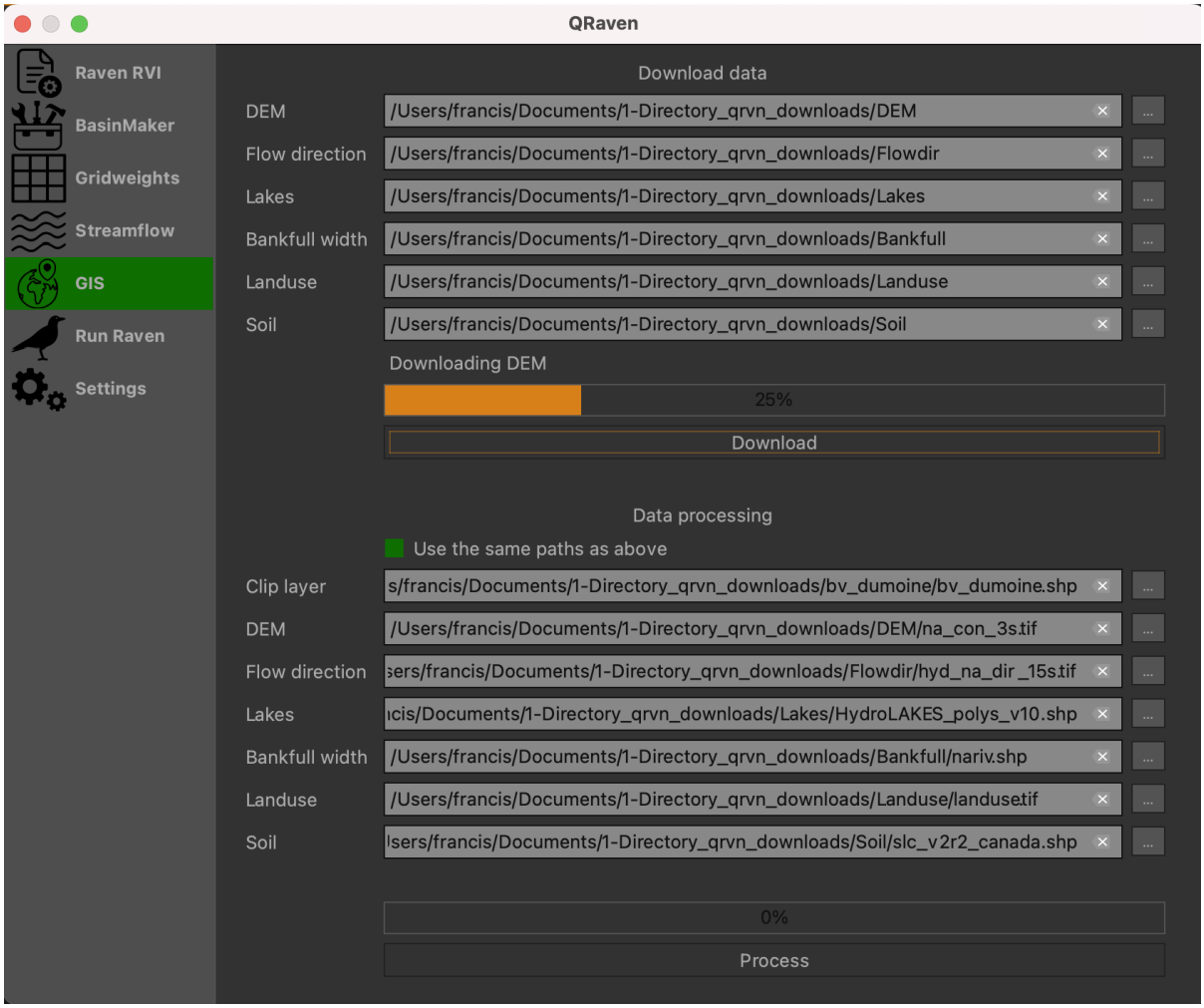


Fig. 5: Example of the GIS UI

2.7 Run a Raven model

To run a Raven model, you need to provide three information.

1. **Input directory** : The directory containing your Raven model files.
2. **Output directory** : The directory where the results of the simulation will be saved.
3. **Raven executable location** : The path to the Raven.exe file (including the filename).

Two other fields are also available, but they should be automatically filled by reading the .rvi file of your model. If an error occurs and they are not filled automatically, please submit a bug report.

1. **Filename prefix** : The name of the .rvi file (without the file extension)
2. **RunName** : The text following the command ":RunName" if used in the .rvi file.

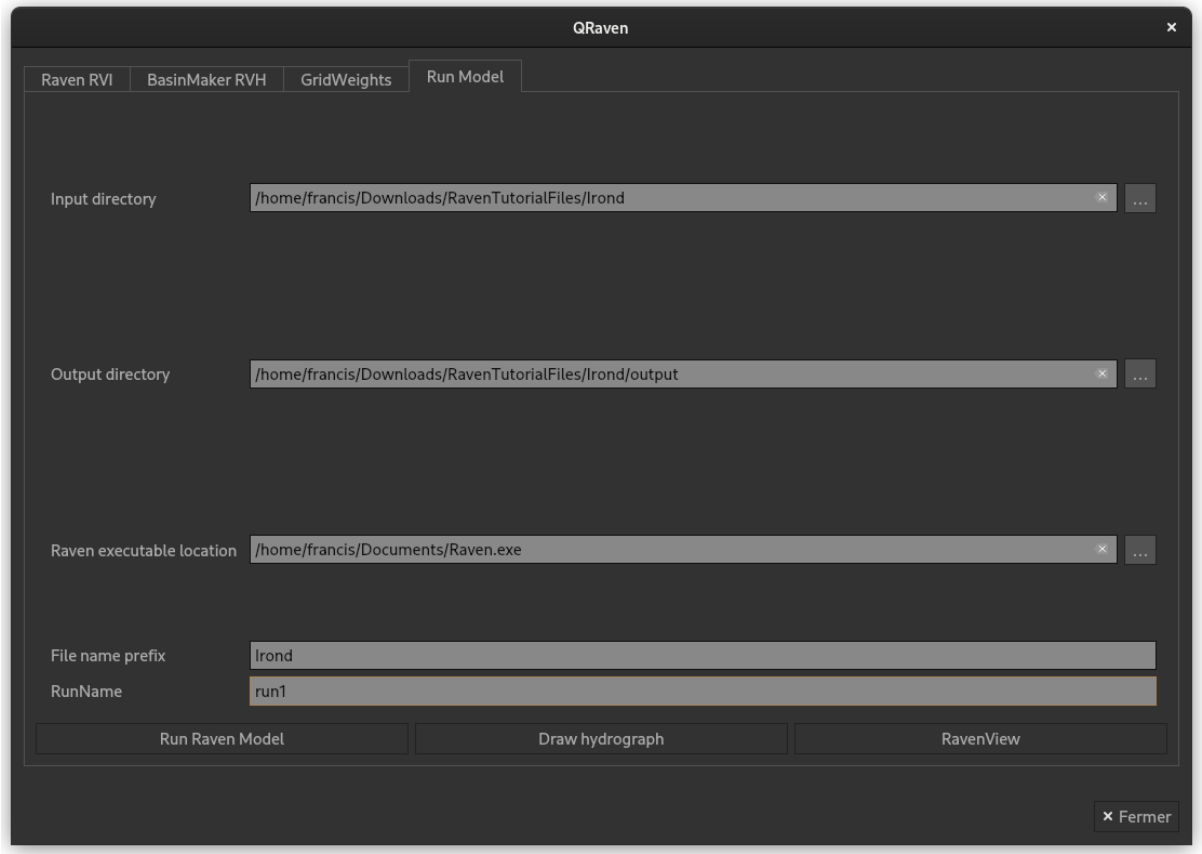


Fig. 6: Example of the Run Model interface.

2.8 Draw the hydrograph

After running a Raven model successfully (with or without QRaven), you will be able to draw the resulting hydrograph. To do so, all that is required is the "Output directory" field and a click on the "Draw hydrograph" button.

In the graph window, multiple buttons are available. They will allow you to zoom in and out, modify the graph size, customize the axis and export the graph as an image.

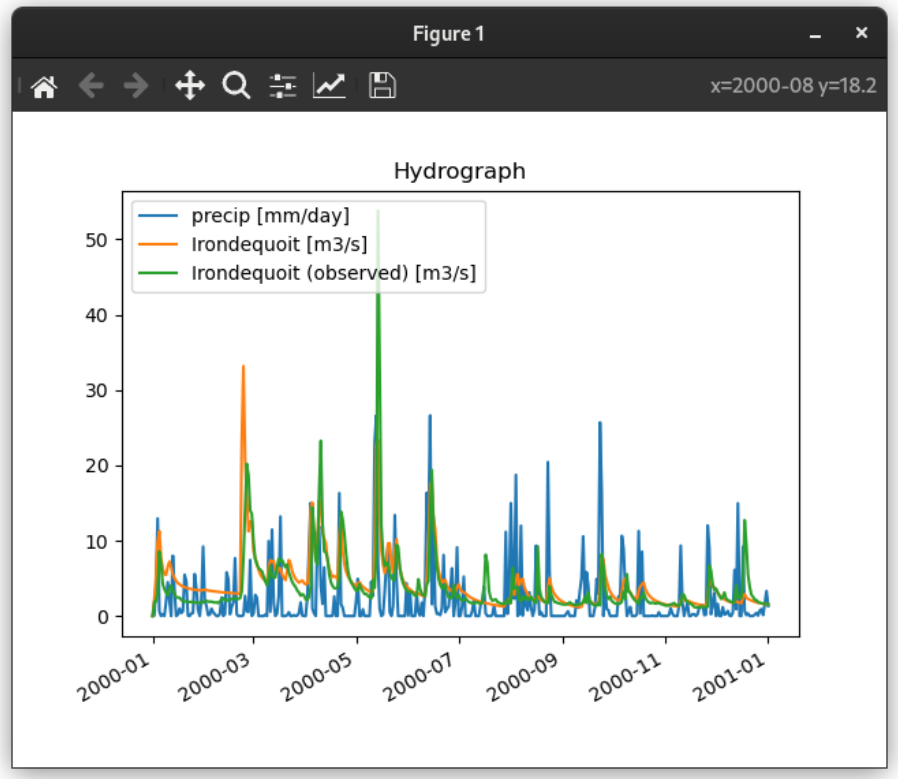


Fig. 7: Example of an hydrograph produced by QRaven.

2.9 Autofill a .rvp template file

A cool feature based on a RavenR function is available to attempt to automatically fill a .rvp template file.

In order to use this feature, a few steps are required.

1. Make sure the ":CreateRVPTemplate" command is used inside the .rvi file.
2. Run the model.
3. Click on the "Auto fill rvp template".
4. Review the generated .rvp file for any values that do not have a default value. Those values will show as "0.12345".

Note: This feature needs more testing and could have many oversights. To help improve it, please submit any problems you encounter by opening a new issue. If possible, also send your Raven model so the issue can be easily reproduced.

2.10 Settings

The Settings menu allows some slight customization of QRaven for the moment. More settings may come later on.

As per now, you can select which containerization software and which image you want to use. You can also select a side menu style.

Containerization software: By default, Docker is selected, but if you want to use Podman, simply select the Podman option in the drop down list.

Image: This option lets you pick between the regular container image and the ARM based image. By default, MacOS will be running the ARM based image. If your MacBook is Intel based, switch to the regular image. Linux and Windows both default on the regular image.

Menu bar style: Allows you to choose between the default menu (icons with text) or a collapsed menu (icons only).

> **Warning:** After making changes to the settings, do not forget to click on the "Save" button. Otherwise, your changes will be lost after closing QGIS.

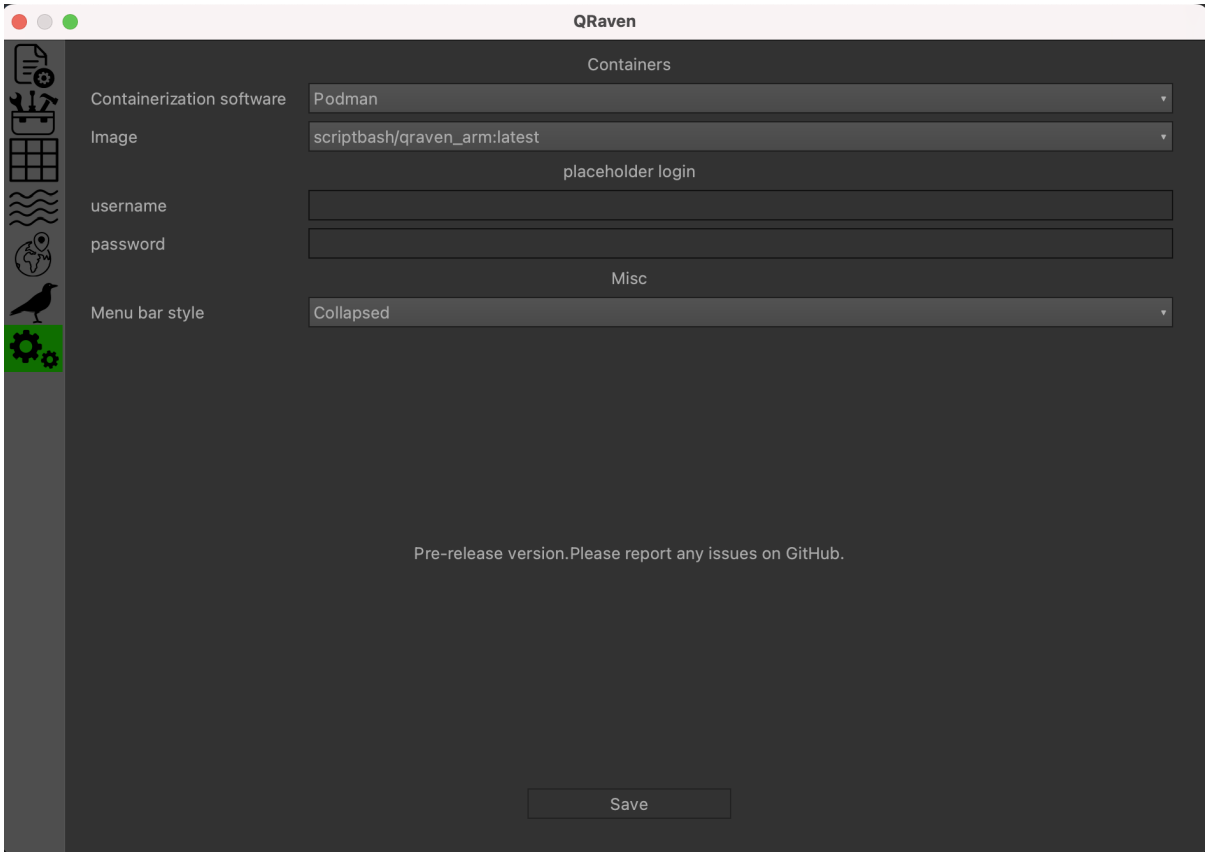


Fig. 8: The settings menu as of version 2.3.0

CHAPTER

THREE

HOW TO CONTRIBUTE TO QRAVEN

There are many ways you can contribute to QRaven and not only by coding!

3.1 Bug reporting

Reporting bugs and problems is very important, since even though I try to test as much as possible the plugin, I can't run into every problems possible.

If you find any possible bugs, errors, typos, please open an issue on GitHub.

3.2 Feature requests

I'm trying to add as many useful features to create Raven models and also ease the use of Raven in general. If there are features you would like or you think would be useful, please submit your request by either opening an issue or a discussion.

3.3 Development

If you want to contribute to the code, by adding new features, fixing bugs, optimizing the code, you're welcome to do so!

QRaven is written in Python and uses PyQt for the graphical user interface. Here is a small guide on how to setup your environment :

3.3.1 Setting up the environment

Note: The folder "QRaven" is not the QGIS plugin, it is the repository name. The folder "qraven" is the plugin. If it is zipped, it can be installed directly in the QGIS extension manager.

Before starting, find where the QGIS plugins folder is.

1. Open QGIS.
2. Click on the "**Settings**" menu.
3. Hover "**User Profiles**" and click on "**Open Active Profile Folder**".
4. Go inside the "**Python**" folder, then inside "**Plugins**".

5. Remember this path, as this is where the "qraven" folder needs to be.
6. Clone QRaven's repository.

```
git clone https://github.com/Scriptbash/QRaven.git
```

You now have two options, either copy/paste the qraven folder into the QGIS plugins folder or create a symbolic link.

I prefer to create a symlink, as it allows to:

- modify the code.
- reload the plugin.
- see the changes.

as opposed to:

- modify the code.
- copy and paste the qraven folder into the plugins folder.
- reload the plugin.
- see the changes.

1. So, to create a symlink, open a "**terminal**" or "**command prompt**" depending on your operating system.
2. Type:

   for Linux and MacOS

   ```
   ln -s <path to qraven directory> <path to qgis plugins folder>
   ```

   for Windows

   ```
   mklink /D <path to qraven directory> <path to qgis plugins folder>
   ```

Now that the development version of QRaven is installed in QGIS, we will install another QGIS plugin. The plugin will allow to reload QRaven after changes are made to it, without the need to close QGIS and reopen it.

1. In QGIS, go to the "**Plugins**" menu.
2. Click on "**Manage and Install plugins**"
3. Click on "**All**" and search for "**Plugin Reloader**"
4. Open the "**Plugin Reloader**" and set it to reload QRaven

You are now good to go! Make changes to the code, reload QRaven with the plugin reloader and submit pull requests!

3.3.2 Files explanation

Not sure where to start? No worries, here is a quick overview of the important files :

- Inside the root folder of the repository, there is **Dockerfile** and **create_RVH.py**
  - **Dockerfile** is the file used to create the Docker image used for BasinMaker and the GridWeights generator
  - **create_RVH.py** file is the script that lies inside the Docker container. In other words, it is the script that runs the BasinMaker functions.
- In the "qraven" folder, you will find many important files.
  - **qraven.py** is the main python file used for the plugin.

  - **qraven_dialog_base.ui** is the graphical user interface of QRaven. You can open it with QtDesigner.

- In the "modules" folder, there are several python files that are imported inside the "qraven.py" file.

CHAPTER

FOUR

CREATE A RAVEN MODEL STEP BY STEP

This tutorial will explain how to build a Raven model from scratch using QRaven. The HBV-EC template will be used to model the watershed of Dumoine river.

4.1 Get the required data

The data needed can be separated for two uses;

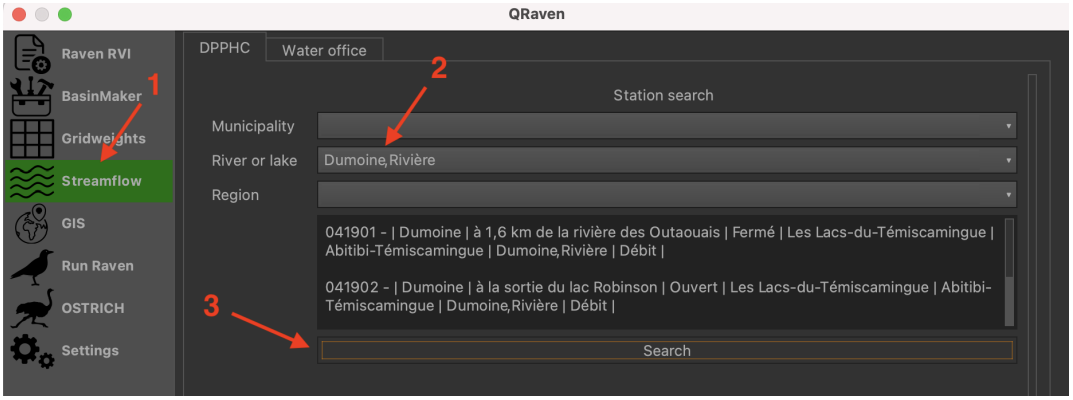
1. for Raven (Temperature, precipitations, streamflow, etc.)
2. for BasinMaker (Landuse polygons, rivers network, DEM, etc.)

4.1.1 Data for Raven

Streamflow

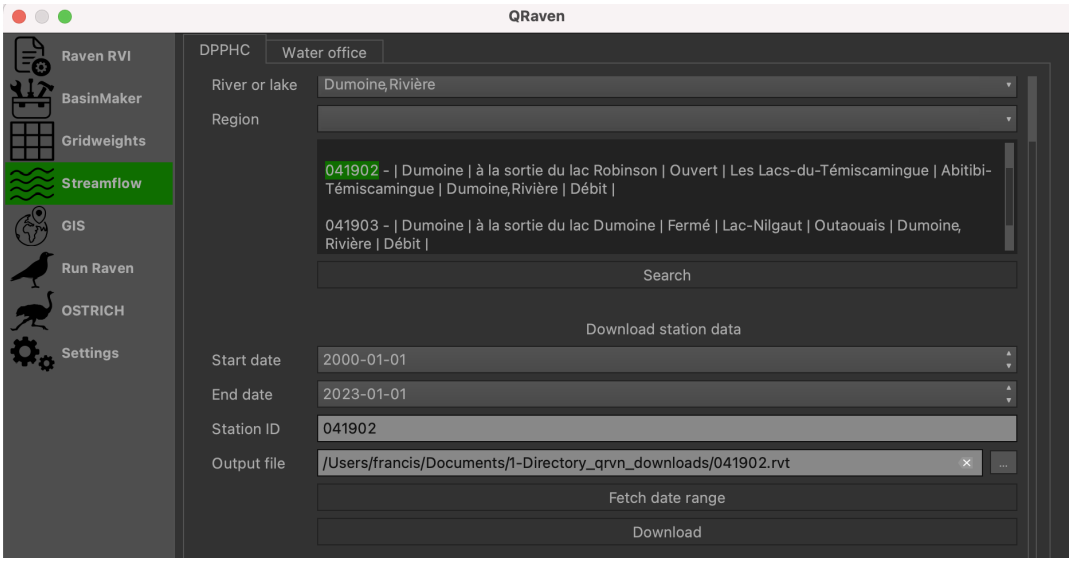
Downloading streamflow data for Canada is quite easy in QRaven.

1. Click on the Streamflow menu.
2. Since the watershed is in the province of Quebec, we will use the DPPHC data scraper. Make sure the DPPHC tab is active.
3. Search for "Dumoine,Rivière" in the "River or lake" field.
4. Click on the "Search button".



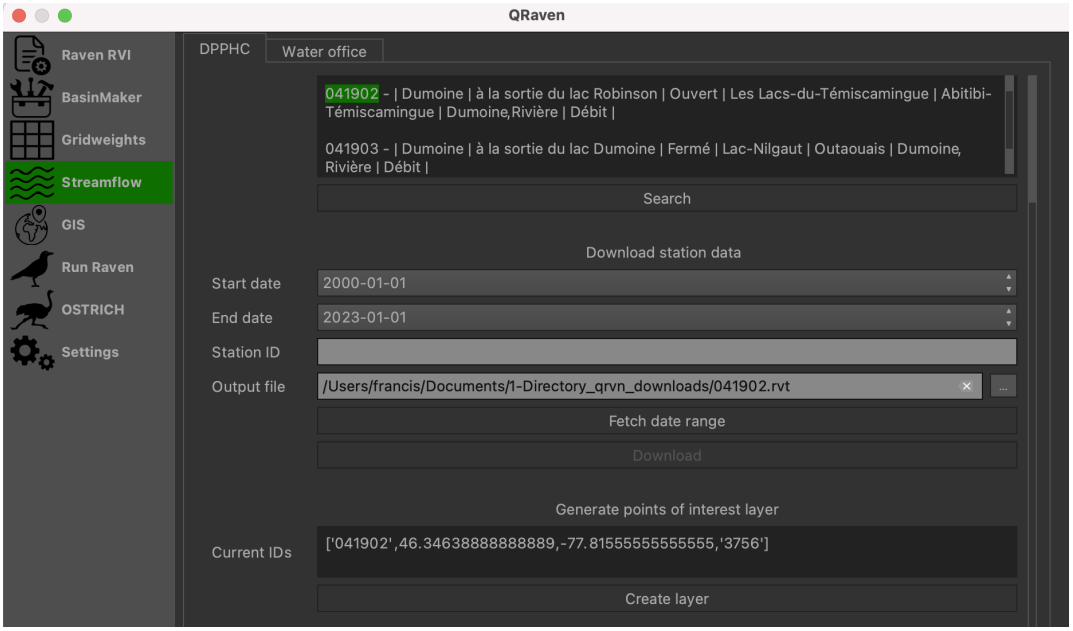
5. Next, copy the station ID of the only operational station.
6. Paste the ID into the "Station ID" field

7. Click on fetch date range. This will set the start date and end date widgets with the first and last date of observation data available.
8. By clicking on fetch date range, the download button will be made available. Select a path where to save the data and click on Download.



Note: The streamflow will already be transformed into a .rvt file. We will only need to edit the basin/HRU ID later. You will also notice some information about the station has been added below the download button. Each time you download data from a station, its information will be added there.

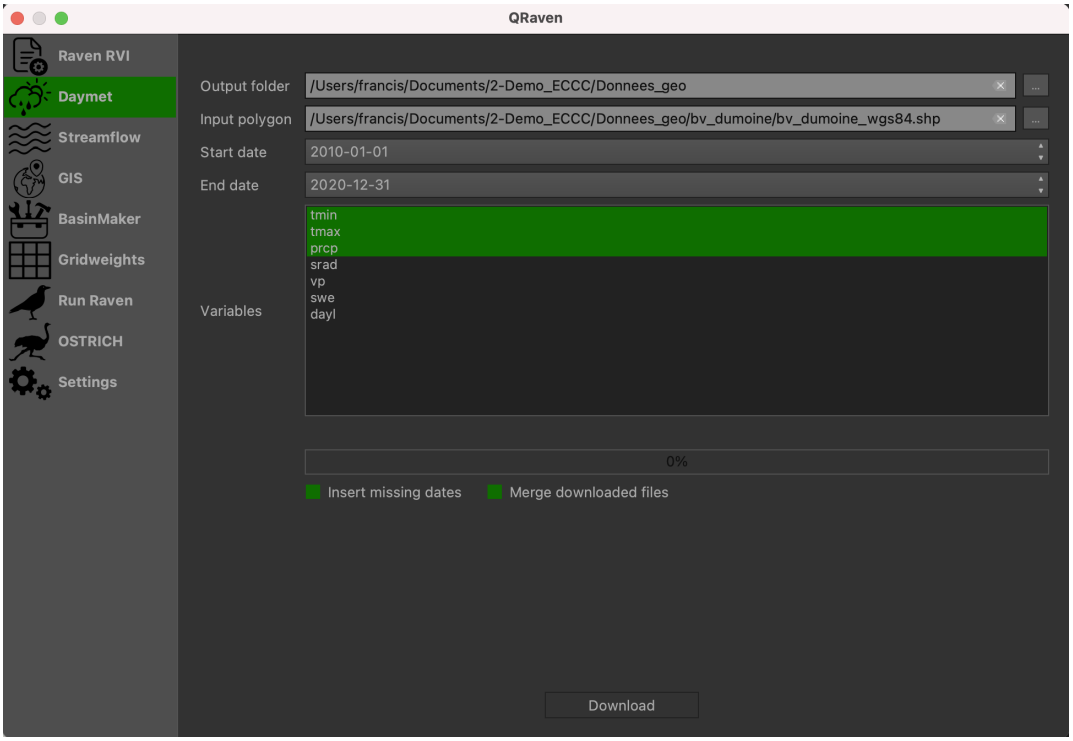
9. Click on "Create layer". This will generate a points layer to be used with BasinMaker.



Precipitations and Temperature

For this tutorial we will only use data coming from the Daymet services. To do so, we will use the Daymet component of QRaven.

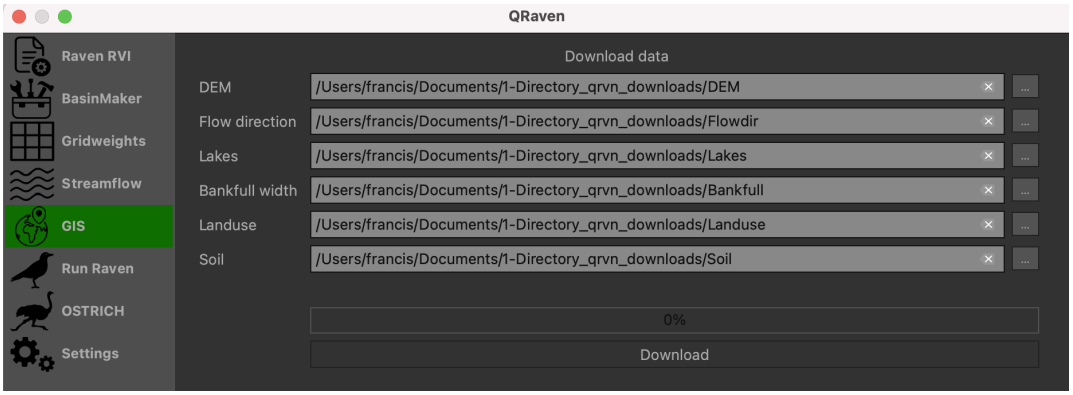
1. Download and extract the shapefile of the Dumoine river here: https://github.com/Scriptbash/QRaven/raw/main/bv_dumoine.zip
2. Click on the "Daymet" menu.
3. Select an output folder where the files will be saved.
4. In the "Input polygon" field, select the extracted shapefile from step 1.
5. Set the start date to 2010-01-01 and the end date to 2020-12-31.
6. In the variable list, select "tmin", "tmax" and "prcp".
7. Leave both the "Insert missing dates" and "Merge downloaded files" checkboxes checked. Since Daymet strips december 31st from the NetCDF files during leap years to include february 29, this will automatically fix the problem for us.
8. Click on the "Download" button and wait for the process to finish. It could take quite a while to finish.



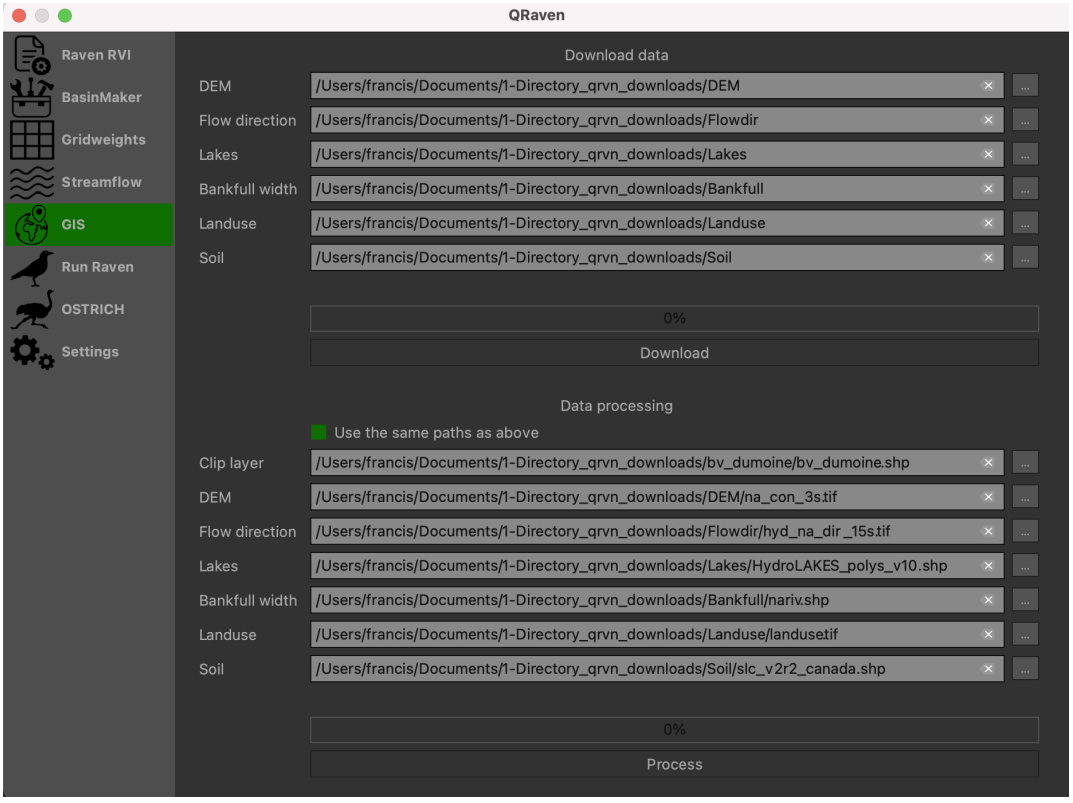
4.1.2 Data for BasinMaker

Data needed to run BasinMaker can be fetch automatically by QRaven (Canada only for now).

1. Click on the GIS menu
2. Select a path where to save the files. Do this for all of the Data.
3. Click on “Download”. This could take a while depending on the files being downloaded.



4. Once the download is finished, check the “Use the same paths as above” checkbox. This tells QRaven where the files to process are.
5. In the “Clip layer” field, select the watershed’s polygon shapefile.
6. Click on “Process”



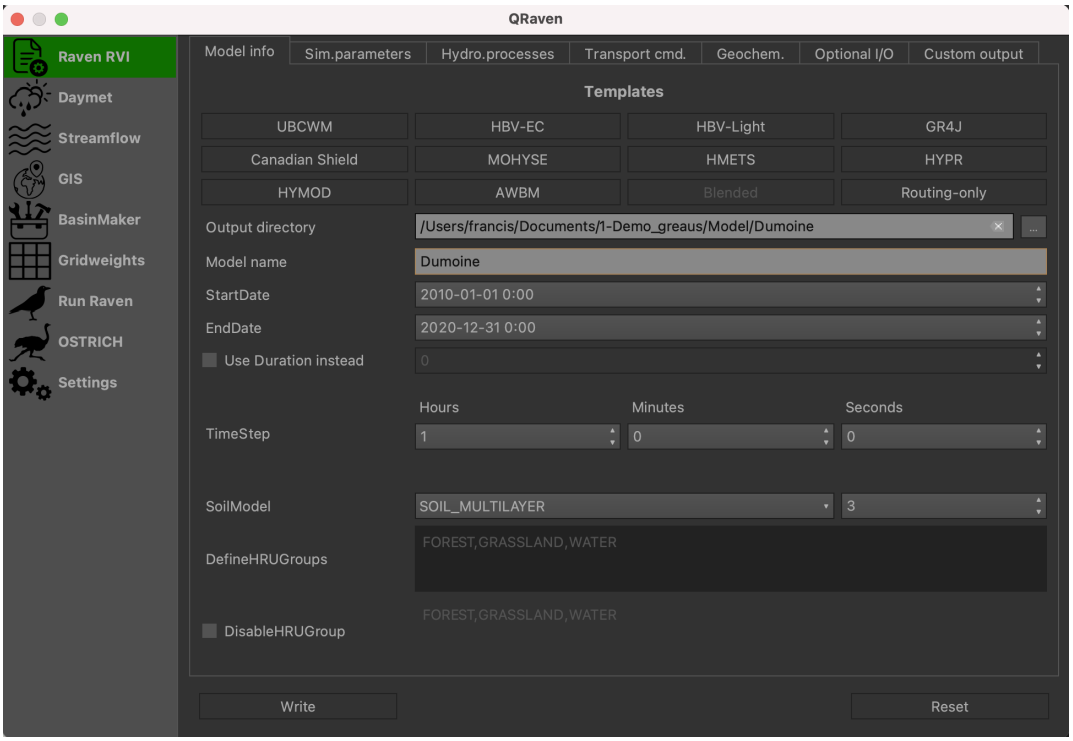
7. The results will be saved inside each data folder and inside a folder named “Results”.

4.2 Setup the Raven files

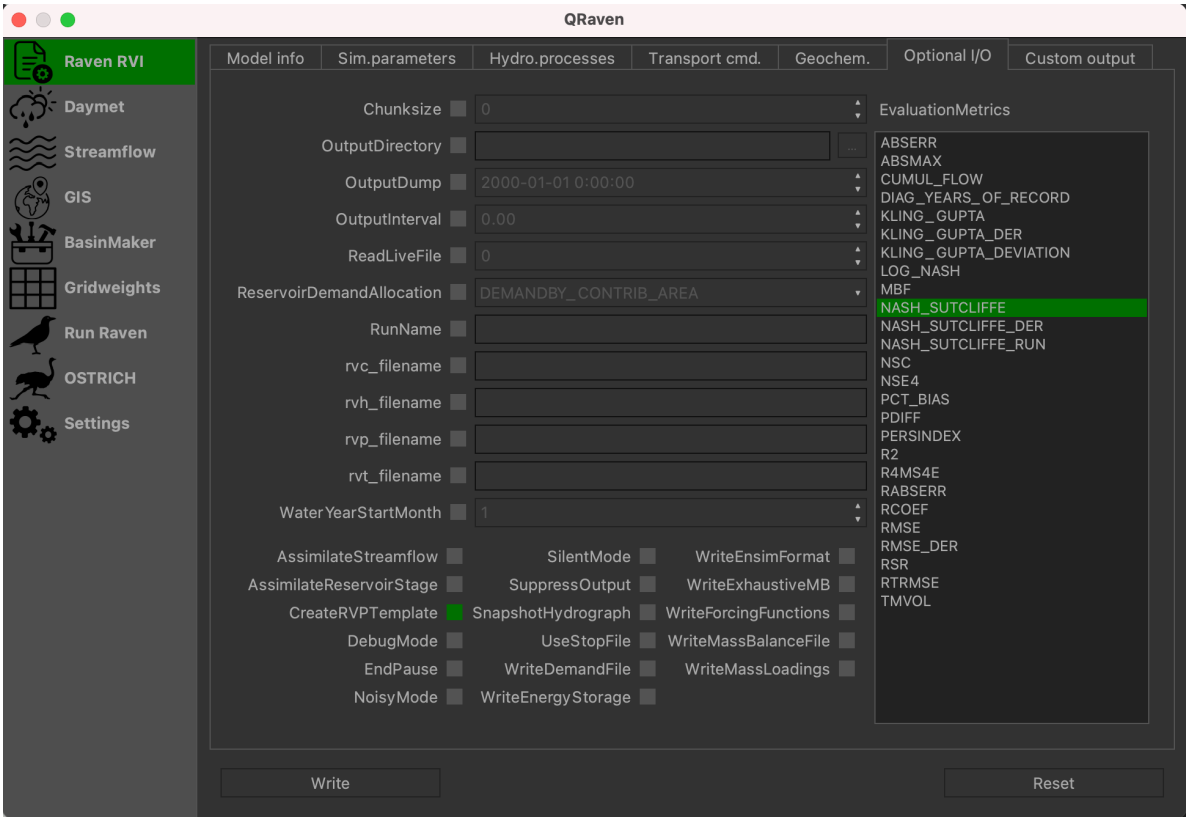
Now that we have all the required data and some model files, we can start setting up the Raven model files.

4.2.1 Generate a .rvi file

1. Click on the "Raven RVI" menu.
2. Make sure the selected tab is "Model info".
3. Click on the "HBV-EC" template button. This will load a basic template with the HBV-EC structure.
4. Select an output directory where the generated .rvi file will be saved.
5. In the model name, type "Dumoine".
6. Set the simulation start date to 2010-01-01 and the end date to 2020-12-31. Leave the hours to 0:00:00.
7. Set the "TimeStep" to 1 hour.



8. Next, click on the "Optional I/O" tab.
9. Check the "CreateRVPTemplate" checkbox. This will allow us to generate an .rvp file with the required parameters for HBV-EC when we will first run the model.
10. While we are in the "Optional I/O", we will select an evaluation metric. Let's select Nash-Sutcliffe.



11. Click on the “Custom output” tab.
12. Click on the “Add output” button to add a new row.
13. Select the proper options to get a custom output that will be “DAILY AVERAGE SNOW BY_HRU”.
14. Generate the .rvi file by clicking on the “Write” button.

4.2.2 Run BasinMaker to create a .rvh file

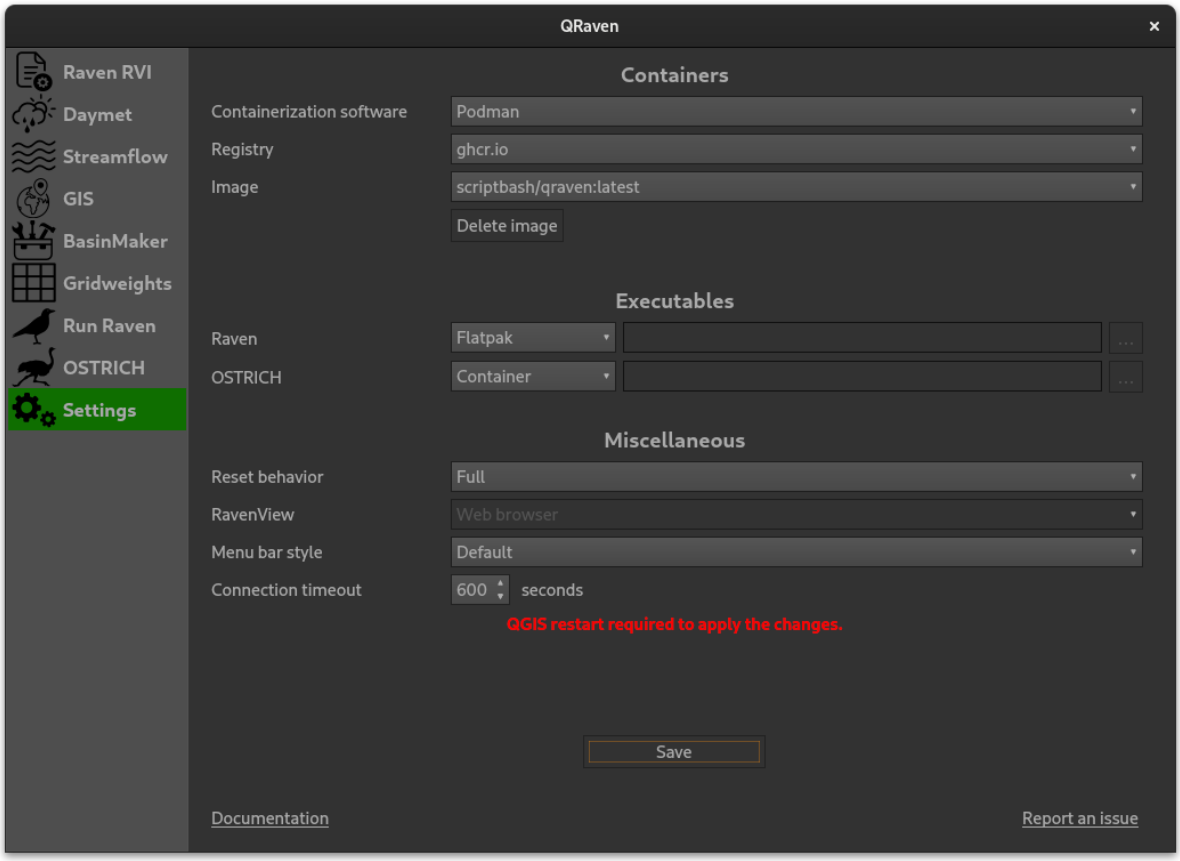
Before using BasinMaker, we will verify the QRaven settings. Since this component uses a containerization software, we will need to make sure the proper options are selected.

1. Click on the “Settings” menu.
2. Select the software you have installed on your computer (either Docker or Podman).

> **Warning:** If you are using Docker, make sure it’s running in the background. If you are on Linux, You will need to take an extra step to run Docker without sudo. Please refer to the *Dependencies* section.

3. You can leave the “Registry” option to the default “ghcr.io”. The choice of the registry should not have any impacts unless one of the website is down.
4. Make sure the “Image” is set on “scriptbash/qraven:latest”, as the unstable version is used for development and may have breaking changes.
5. Click on the “Save” button when you are ready.

Note: If you have made changes to the containers settings, you will need to close QRaven and restart QGIS for the changes to apply.

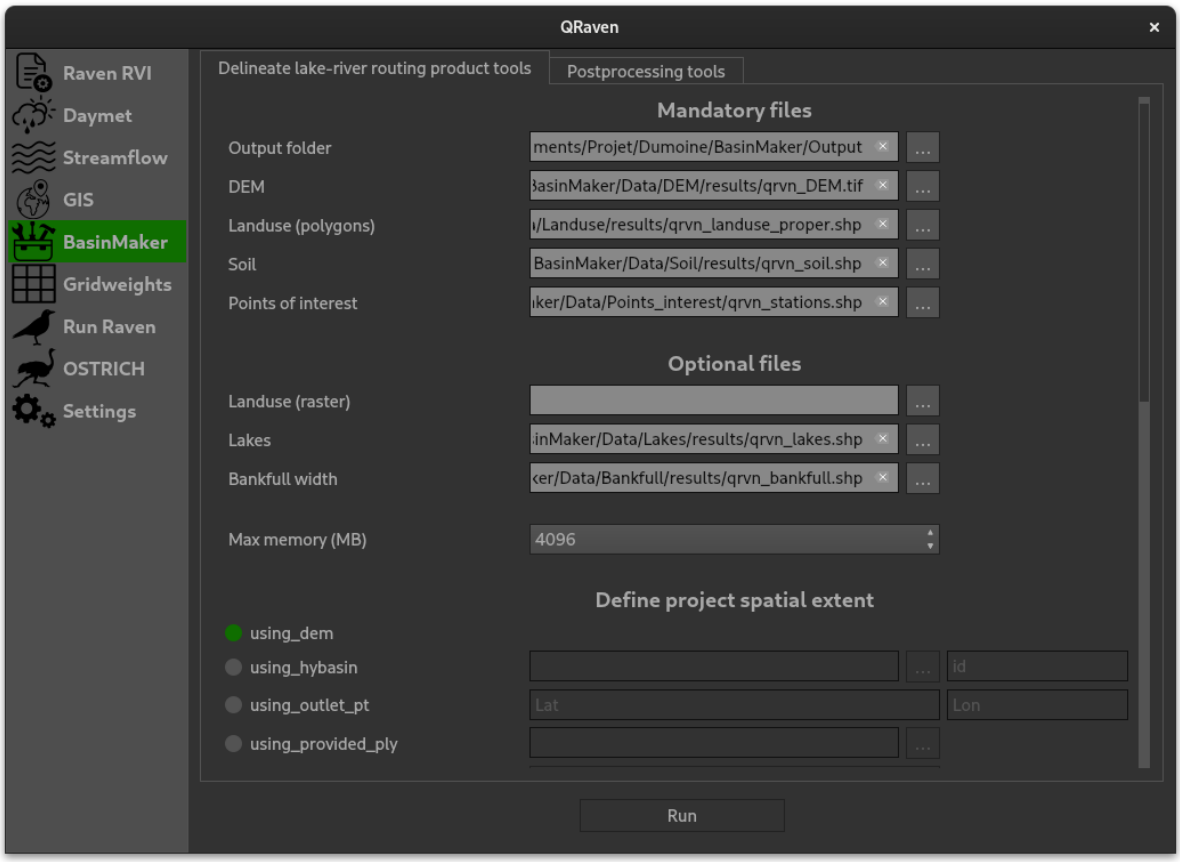


Now that the settings are properly set up, let's head back to the "BasinMaker" menu.

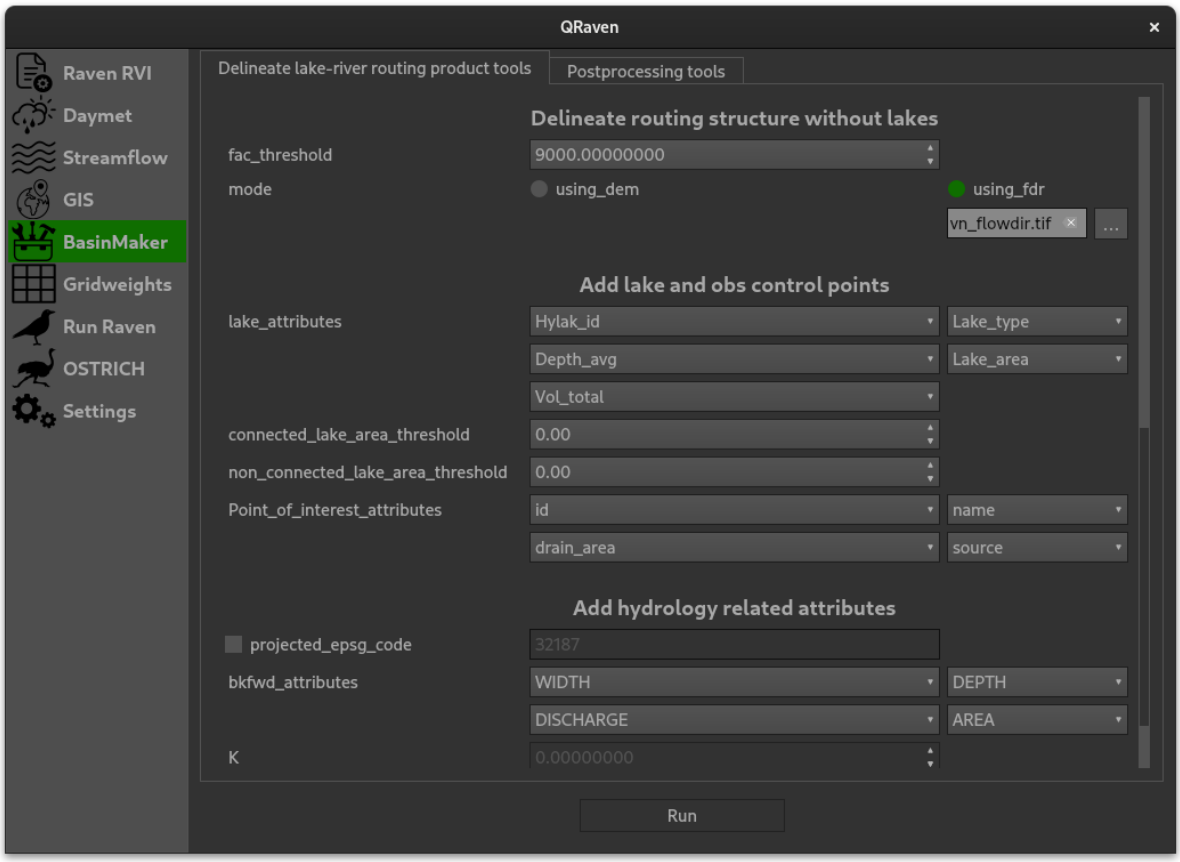
1. Select an output folder where the BasinMaker results will be saved.
2. In the "DEM", "Landuse (polygons)", "Soil", "Lakes" and "Bankfull width" widgets, select the corresponding file previously downloaded in the "GIS" menu.

Note: The files should be in a folder named "Results" and the files should have "qrvn" as a prefix.

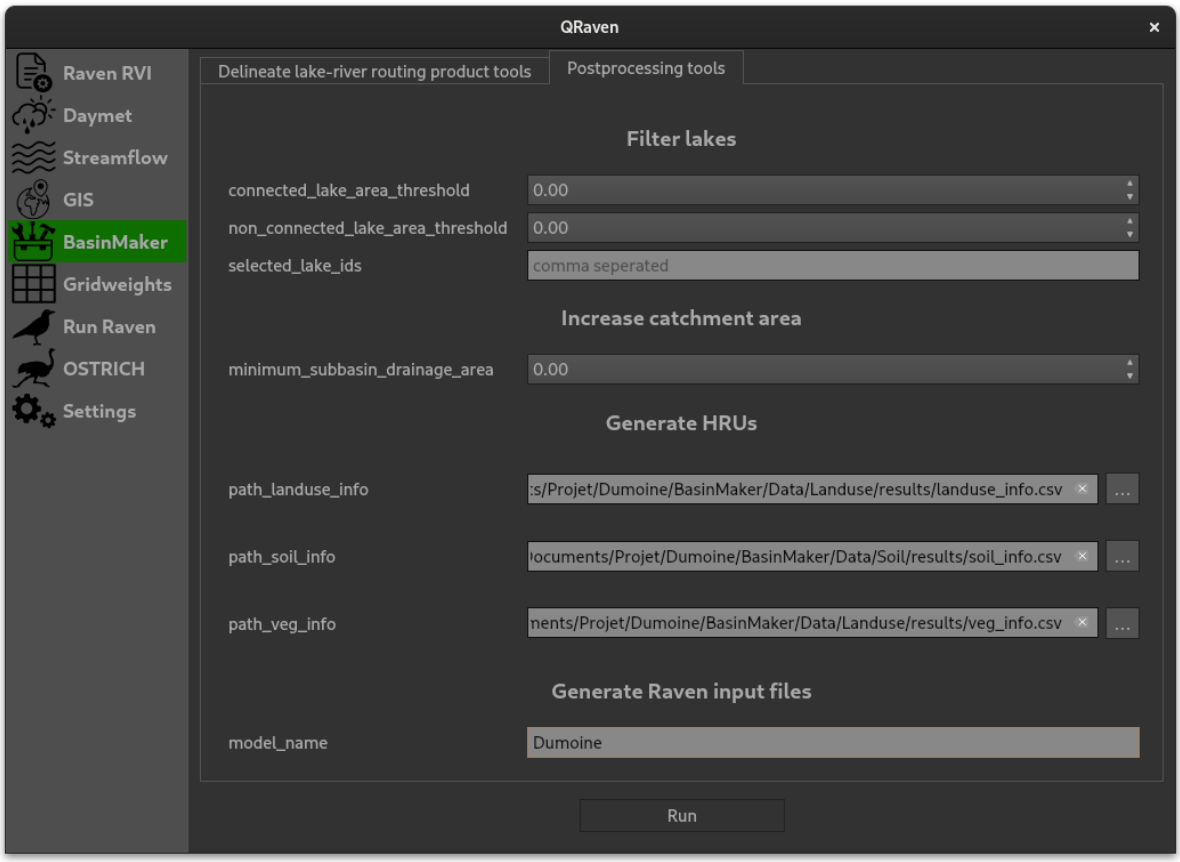
3. For the "Points of interest", we will select the file created after streamflow data was acquired.
4. You can leave the "Max memory (MB)" to the default value.
5. Under "Define project spatial extent", verify that the "using_dem" button is checked.



6. Leave the “fac_threshold” to the default value (9000).
7. Select “using_fdr” as the mode for “Delineate routing structure without lakes”. The field under the option will unlock. Select the flow direction file.
8. For the lake_attributes, select the corresponding shapefile attributes.
9. Just like for the lakes, select the corresponding shapefile attributes for the “point_of_interest_attributes”.
10. Select the corresponding “bkfwd_attributes” for the bankfull width shapefile.



11. Click on the “Postprocessing tools” tab.
12. In the “path_landuse_info, select the “landuse_info.csv” file which should be inside the folder with the qrvn_landuse shapefile.
13. In the “path_soil_info, select the “soil_info.csv” file which should be inside the folder with the qrvn_soil shapefile.
14. In the “path_veg_info, select the “veg_info.csv” file which should be inside the folder with the qrvn_landuse shapefile.
15. Enter “Dumoine” in the “model_name” text field.
16. Finally, click on the “Run” button

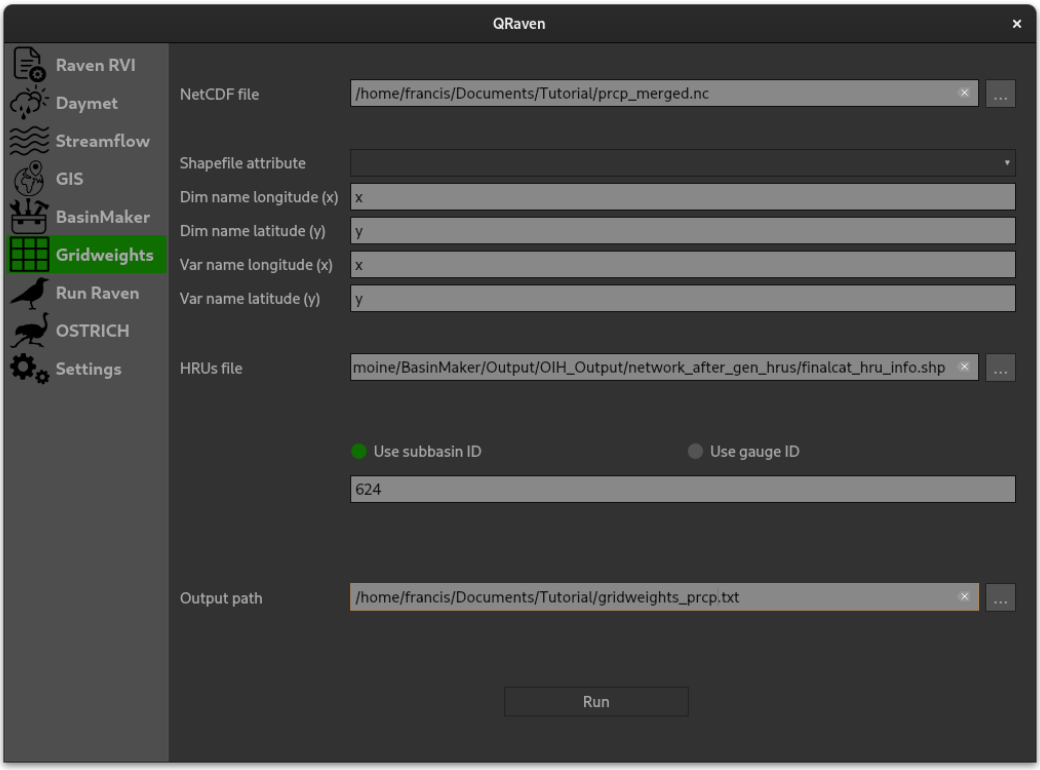


Note: It will take a significant amount of time to complete. At times, it may look like QGIS is completely frozen, but it's not actually the case. Please do not force close QGIS, wait for the process to finish by itself. If you must absolutely stop BasinMaker, you can stop the container instead. The process should end shortly after and QGIS will unfreeze.

4.2.3 Generate grid weights

Since the format of the data taken from Daymet is netCDF, we need to use the GridWeightsGenerator.

1. Click on the "Gridweights" menu.
2. Select one of the netCDF files.
3. Enter the dimensions and variables names. It should be x and y.
4. In the "HRUs file" widget, select the BasinMaker final output. The file name should be "finalcat_hru_info.shp".
5. Check the "Use subbasin ID" button.
6. Below the button we just checked, we need to enter the ID of the most downstream subbasin. This information can be found in the attributes table of the "finalcat_hru_info.shp" shapefile. The column should be called "SubId".
7. Select an output file.
8. Click on the "Run" button.
9. repeat the previous steps for the other netCDF files.



4.2.4 Finishing up the files setup

We now have almost all the required files to make the Raven model. The file structure of the model should look like this for now :

```
Dumoine_model
├── Dumoine.rvi
├── Dumoine.rvh
├── Lakes.rvh
├── Dumoine.rvp
├── channel_properties.rvp
├── 041902.rvt
├── tmin_merged.nc
├── tmax_merged.nc
├── prcp_merged.nc
├── gridweights_tmin.txt
├── gridweights_tmax.txt
└── gridweights_prcp.txt
```

Before being able to run the model, we will need to make some manual changes.

First, we need to update the HRU Id in the 041902.rvt file.

1. Open the file 041902.rvt
2. Replace the text <Basin_ID or HRU_ID> in the first line for the HRU Id in which the station is located. In my case it is 602.
3. The first line of the file should now be like this :ObservationData HYDROGRAPH 602 m3/s

Next, we must create an .rvt file that will tell Raven where to look for the observations and the gridded data.

1. Create a file called Dumoine.rvt
2. Enter the following text:

```
:GriddedForcing           precipitations
    :ForcingType          PRECIP
    :FileNameNC           prcp_merged
    :VarNameNC            prcp
    :DimNamesNC           x y time
    :RedirectToFile       gridweights_prcp.txt
:EndGriddedForcing

:GriddedForcing             Min_temp
    :ForcingType            TEMP_MIN
    :FileNameNC             tmin_merged.nc
    :VarNameNC              tmin
    :DimNamesNC             x y time
    :RedirectToFile         gridweights_tmin.txt
:EndGriddedForcing

:GriddedForcing             Max_temp
    :ForcingType            TEMP_MAX
    :FileNameNC             tmax_merged.nc
    :VarNameNC              tmax
    :DimNamesNC             x y time
    :RedirectToFile         gridweights_tmax.txt
:EndGriddedForcing

:RedirectToFile                 041902.rvt
```

4.3 Run the raven model

Before being able to run the model, we will need to create a .rvp file. Since we checked the option ":CreateRVPTemplate" in the .rvi file, Raven will generate a template file the first time we run the model.

Warning: The Raven executable path must be set the in the "Settings" menu. If you want to use a container or Flatpak, you must select those options before running the model.

1. Click on the "Run Raven" menu.
2. In the "Input directory", select the folder that contains all the model files.
3. Select an output directory.
4. The "File name prefix" and "RunName" fields will be filled automatically.
5. Click on the "Run Raven model" button.

We should now have a template file called "Dumoine.rvp_temp.rvp". Since this file only contains the required parameters without their values, we can automatically fill it using QRaven. To do so, click on "Auto fill rvp template". Now that we have a .rvp template, click on "Overwrite :CreateRVPTemplate". This will remove the command from the .rvi file and will allow us to run the actual model.

Since BasinMaker already generated an .rvp file, we will need to add its content to the new .rvp file. Open the new file file and add the following line:

```
:RedirectToFile channel_properties.rvp
```

The model is now ready, click on the "Run Raven model" button once again.

4.4 Calibration with OSTRICH

to-do