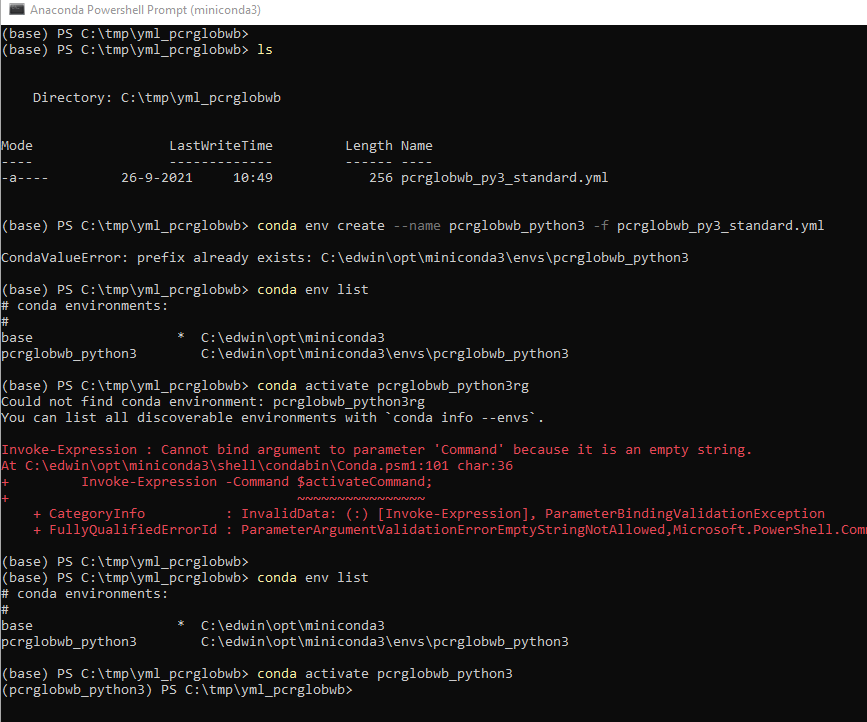
**Cooking recipes**

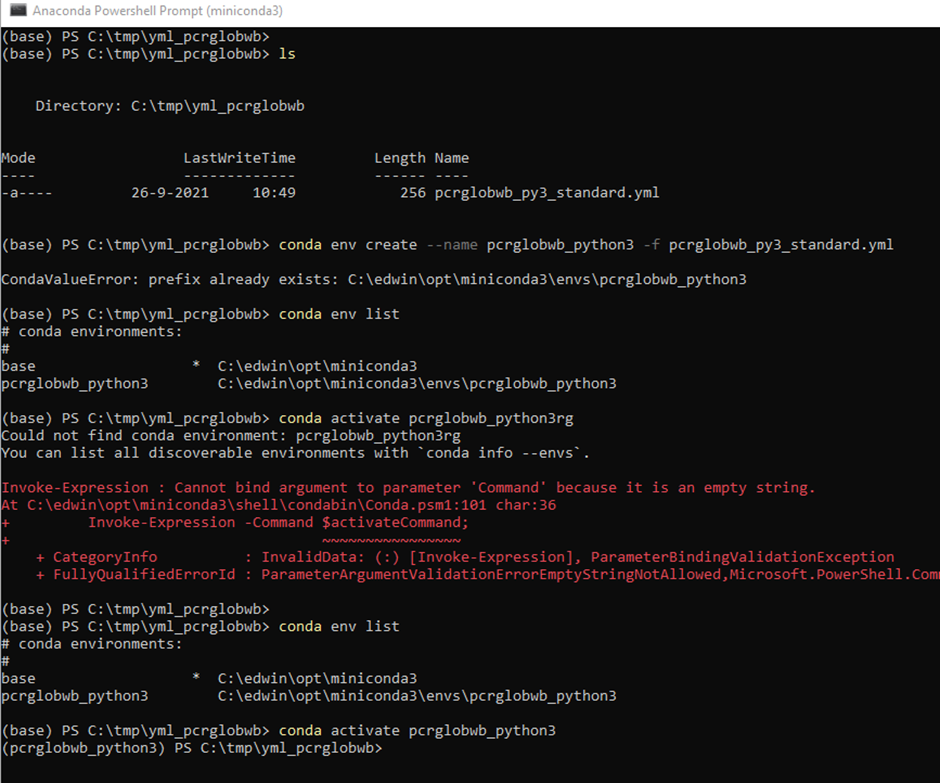
**Making PCR-GLOBWB runs using your own clone/landmask maps (your own study areas)**

Exercise:

1. Choose a study area: a river basin/sub-catchment with the size less than ~250,000 km2. In this exercise, we will make a PCR-GLOBWB run at 5 arcmin (~10 km) resolution for this study area and we expect that the calculation should be finished within 24 hours.
2. We will start by making the clone and landmask map files (at 5 arcmin resolution) for the region that you choose. The map files must be on PCRaster format. Follow the instructions below. I suggest to make these maps on your local computer. You may also want to use your remote computing machine (e.g. velocity), but connections may make the processes, especially visualization, difficult.

* Please install PCRaster (if you have not installed it). It is recommended to use Miniconda, particularly for Python 3 (<https://docs.conda.io/en/latest/miniconda.html#latest-miniconda-installer-links>), and create the conda environment that is used for running PCR-GLOBWB, see e.g. <https://github.com/UU-Hydro/PCR-GLOBWB_model/blob/master/conda_env/pcrglobwb_py3_standard.yml>, which already includes PCRaster (see the following screenshots that illustrate the installation under Miniconda3 on a windows machine).





* You may also just use the existing clone and landmask maps for the Rhine-Meuse basin that are already provided on <https://github.com/UU-Hydro/PCR-GLOBWB_model/raw/master/clone_landmask_maps/clone_landmask_examples.zip> (please do not use the ones for Mississippi).

1. We will have to use the ldd map (drainage direction map). You can find the ldd map from the following opendap server location (ps: you can also find this file on “velocity”).

# download the file from the opendap server

$ wget <https://opendap.4tu.nl/thredds/fileServer/data2/pcrglobwb/version_2019_11_beta/pcrglobwb2_input/global_05min/routing/ldd_and_cell_area/lddsound_05min.nc>

The ldd file (somehow) has a netcdf format (such that it can be supported by the netcdf opendap protocol). For our purpose now, to further working with this file on this exercise, you have to convert it to a pcraster ldd file. You can use the following gdal and pcraster commands.

# convert a netcdf file to pcraster format

$ gdal\_translate -of PCRaster lddsound\_05min.nc lddsound\_05min.nc.map

# the above lddsound\_05min.nc.map file has scalar values

$ mapattr -p lddsound\_05min.nc.map

mapattr version: 4.4.0 (linux/x86\_64)

attributes lddsound\_05min.nc.map

rows 2160

columns 4320

cell\_length 0.0833333

**data\_type scalar**

cell\_repr single

projection yb2t

angle(deg) 0

xUL -180

yUL 90

min\_val 1

max\_val 9

version 2

file\_id 0

native y

attr\_tab n

# convert a pcraster scalar map to a pcraster ldd type

$ pcrcalc lddsound\_05min.map = "lddrepair(ldd(lddsound\_05min.nc.map))"

PS: You may want to check the description for the pcraster functions ldd and lddrepair, see <https://pcraster.geo.uu.nl/pcraster/4.4.0/documentation/pcraster_manual/sphinx/op_ldd.html> and <https://pcraster.geo.uu.nl/pcraster/4.3.3/documentation/pcraster_manual/sphinx/op_lddrepair.html>.

1. Try to visualize the map (using aguila) and try to zoom in to your chosen study area. You may want to get some overview about a ldd type pcraster map: <https://pcraster.geo.uu.nl/pcraster/4.4.0/documentation/pcraster_manual/sphinx/secdatbase.html#formldd>

$ aguila lddsound\_05min.map

1. We will calculate “strahler order” of these river network (and then visualize it). For some explanation about this function, see: <https://pcraster.geo.uu.nl/pcraster/4.4.0/documentation/pcraster_manual/sphinx/op_streamorder.html>. For the definition of “strahler order”, see e.g. <https://en.wikipedia.org/wiki/Strahler_number>.

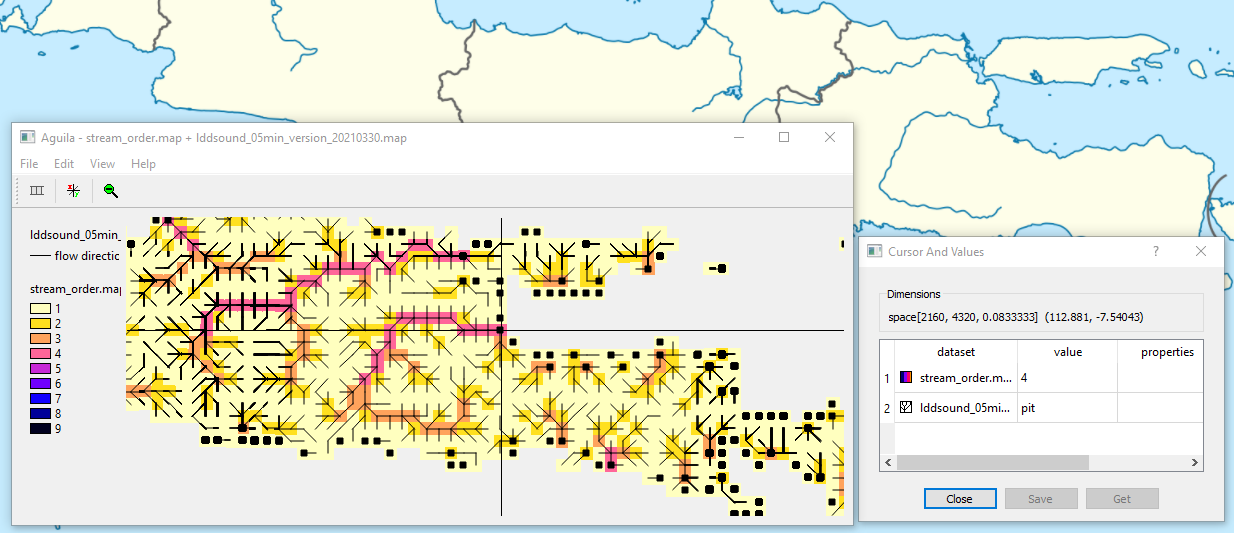
$ pcrcalc stream\_order.map = "streamorder(lddsound\_05min.map)"

$ aguila stream\_order.map

1. Try to visualize your stream\_order.map and your lddsound\_05min.map at the same time using the following command.

$ aguila stream\_order.map + lddsound\_05min.map

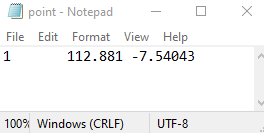
1. Then, try to zoom into your visualization/aguila in order to identify the most downstream pixel of your study area. Click the menu “View” and then “Show Cursor and Values”. You will get an extra window showing some information of the pixel (where the cursor is currently being clicked or active/located), including its coordinate (longitude and latitude values). Write this coordinate on a piece of paper.



1. For the above example, the coordinates that are identified: x: 112.881; y: -7.54043. We will make a point map for this coordinate.
2. Use your text editor (e.g. notepad, geany) to make a text (.txt) file consisting of 1 line and 3 columns:

*1 <your\_x\_longitude\_coordinate> <your\_y\_latitude\_coordinate>*

An example is given below:

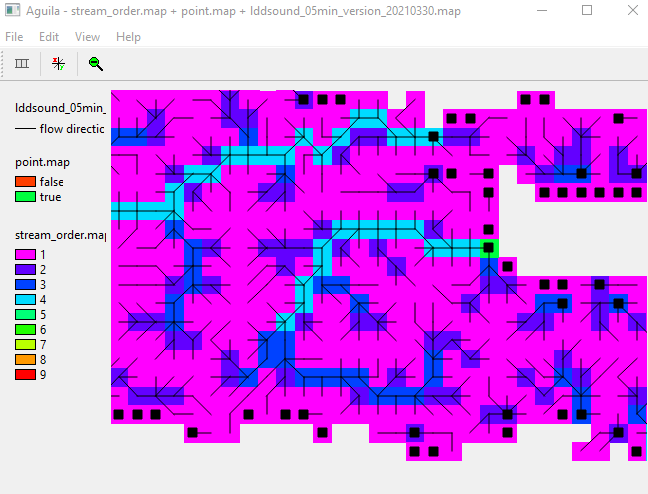


1. We will rasterize this point.txt using the col2map command (see also <https://pcraster.geo.uu.nl/pcraster/4.4.0/documentation/pcraster_manual/sphinx/app_col2map.html>)

$ col2map --clone lddsound\_05min.map -B -m mv -x 2 -y 3 -v 1 point.txt point.map

1. Try to visualize this point map and check if the point is in the location/raster/pixel that you expect.

$ aguila stream\_order.map + point.map + lddsound\_05min.map



1. We will have to make a catchment map based on the point map.

$ pcrcalc catchment.map = "catchment(lddsound\_05min.map, point.map)"

$ pcrcalc catchment.map = "if(catchment.map eq 1, boolean(1))"

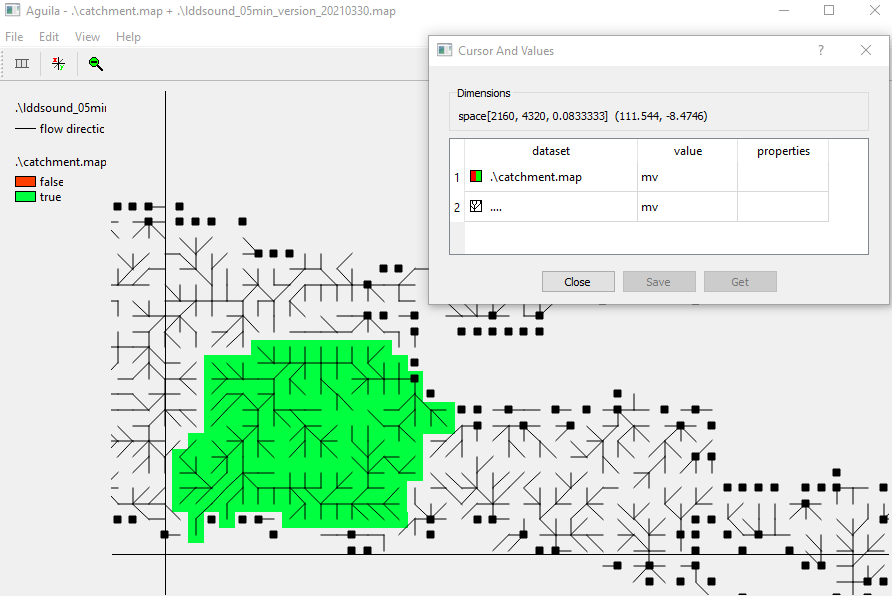
For the definition of the pcraster function “catchment”, see: <https://pcraster.geo.uu.nl/pcraster/4.4.0/documentation/pcraster_manual/sphinx/op_catchment.html>

1. Visualize the “catchment.map” using aguila.

$ aguila catchment.map + lddsound\_05min.map

1. Next, using aguila, we will try to identify a rectangular cropped area/box that can capture/bound the catchment area that you choose, see the next figure that includes an “imaginary” bounding box colored in blue (PS: The blue box is just an imaginary one, you will not be able to draw and see it). You will have to identify the coordinates of four corners of your bounding box. Please click the menu “View” and then “Show Cursor and Values” so that you will get an extra window showing some information of the pixel (where the cursor is currently being clicked or active/located), including its coordinate (longitude and latitude values).

Note that for clone maps used in PCR-GLOBWB, there is an important condition for the corner coordinates: **The corner coordinates (from all four corners) of the clone map must be nicely-rounded (integer) values (without decimals)**. This condition is needed by resampling processes used in PCR-GLOBWB, particularly to avoid rounding errors. For example, the lower left corner coordinate from the bounding box shown in the screenshot below is 111.0, -9.0 (rounded from 111.544, -8.4746).



1. For the example above, the coordinates of the clone map are (x, y or longitude, latitude):

upper left corner : 111.0, -7.0 upper right corner : 114.0, -7.0

lower left corner : 111.0, -9.0 lower right corner : 114.0, -9.0

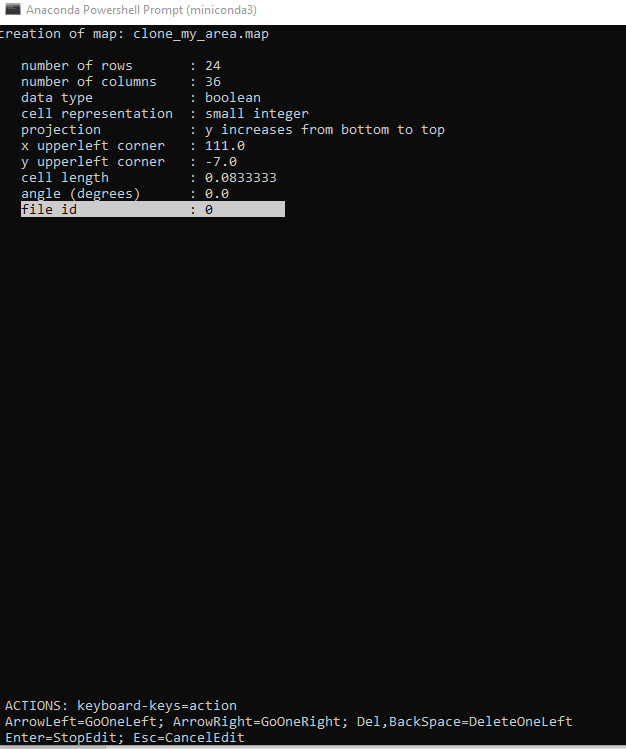
1. Then, given the model resolution (5 arc minutes = 5/60 arc degrees = 0.08333333333 degrees), you can calculate the number of rows and columns:

number of rows : [-9.0 – (-7.0)] / [5/60] = 24

number of columns : [114.0 – 111.] / [5/60] = 36

1. Given all information about the corner coordinates, numbers of rows and columns, as well model cell size or resolution, we can make the clone map using the command mapattr, which will invoke an interactive window/menu. Please fill in the information needed to the interactive menu. Note that for PCR-GLOBWB clone map, please keep the default values for cell representation (“small integer”), projection (“y increases from bottom to top”), angle (“0.0”). For more information about mapattr, please see <https://pcraster.geo.uu.nl/pcraster/4.4.0/documentation/pcraster_manual/sphinx/app_mapattr.html>

$ mapattr clone\_my\_area.map

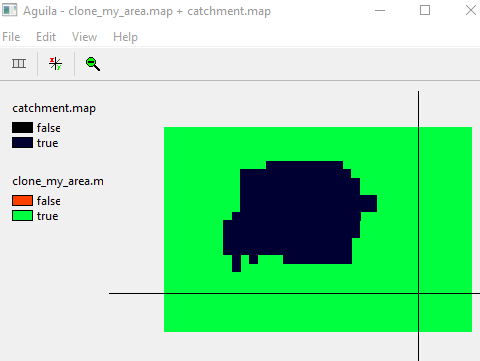


After you finish with filling in or editing the information about map properties, press “q” (to quit) and then “y” (to confirm to create the map).

1. Now you should have your clone map. Using aguila, please check whether the clone map (clone\_my\_area.map) captures/bounds the entire study area (catchment.map).

$ aguila clone\_my\_area.map + catchment.map

Note that to able to distinguish both maps, you have to change the color/legend of your clone\_my\_area.map by double clicking the legend of the catchment.map and then choosing the latest (lowermost) color scheme option.



1. Next, we will have to resample/crop the catchment.map, that still has a global extent, to the extent of your clone (e.g. clone\_my\_area.map). You may want to check <http://pcraster.geo.uu.nl/pcraster/4.1.0/doc/manual/app_resample.html>

$ resample --clone clone\_my\_area.map catchment.map landmask\_my\_area.map

1. Copy (or rename) the file to a different name that indicates the (e.g. using the river name).

$ cp landmask\_my\_area.map landmask\_brantas\_05min.map

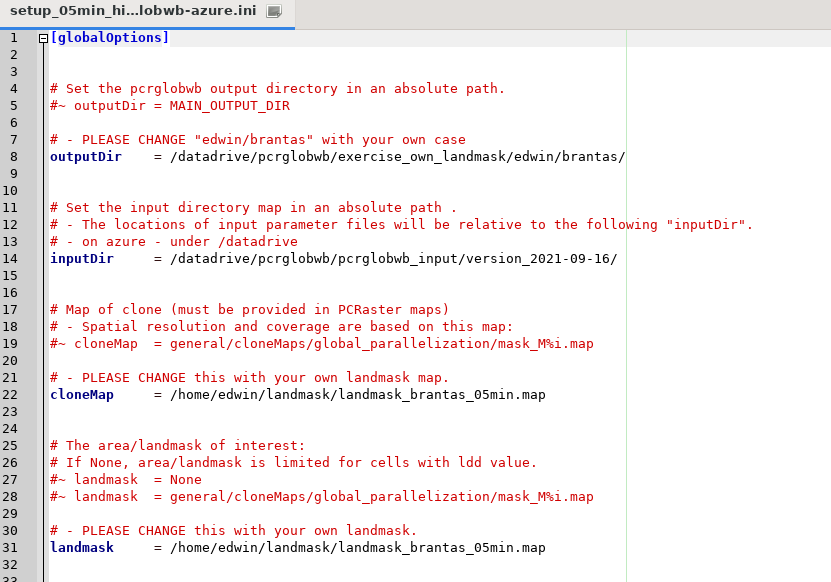
1. The file will be used as the **cloneMap** and **landmask** in the **globalOptions** of your PCR-GLOBWB configuration (.ini) file.
2. If you create the clone and landmask map files on your local laptop, you will have to copy the files to your computing machine (e.g. velocity). For this, you can use the following “scp” command.

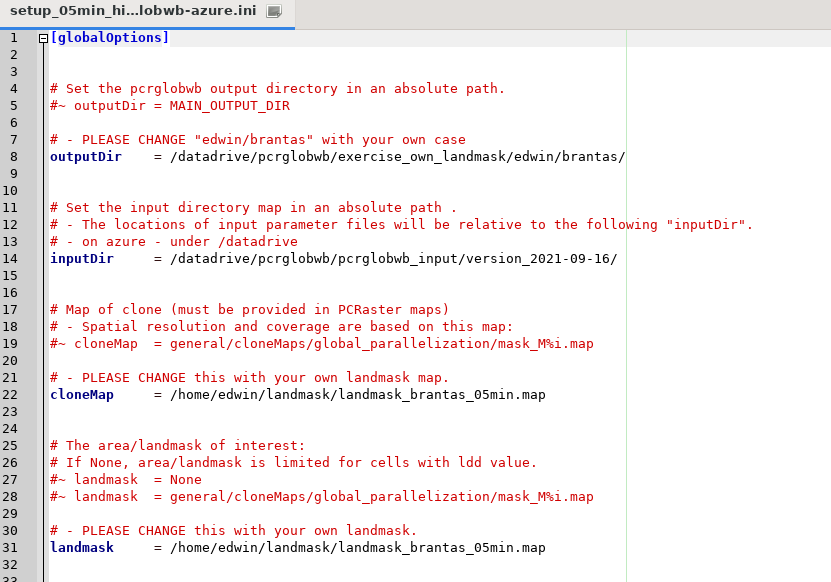
# on your local terminal:

# scp <source\_location> <target\_location>

$ scp landmask\_brantas\_05min.map sutan101@velocity.geo.uu.nl:/home/sutan101/landmask

1. Up to this point, you should already have your clone and landmask maps and have uploaded them to your computing machine. Next, we will use a template of a configuration (.ini) file that been provided on <https://github.com/UU-Hydro/PCR-GLOBWB_model/blob/feature/exercise/config/setup_05min.ini>
2. Open the .ini file, and please modify the fields “cloneMap”, and “landmask”, see below for an example/illustration.





1. Do not forget to adjust some other fields, such as outputDir, inputDir, etc, for your own needs.
2. After you edit the .ini file, please save it. Then, you will be able to perform your PCR-GLOBWB. You may want to start a screen session before running your model, as you may want to make several runs with different configuration.

# go to the sub-folder “model” within “PCR-GLOBWB\_model”

$ cd /home/edwin/PCR-GLOBWB\_model/model

$ pwd

/home/edwin/PCR-GLOBWB\_model/model

# the basic command to run PCR-GLOBWB is the following

# python deterministic\_runner.py <your\_configuration\_file> debug

$ python deterministic\_runner.py /home/edwin/PCR-GLOBWB\_model/config/setup\_05min.ini debug

1. During your model run, you can already explore the model output. Please go to the directory that you set for the field “outputDir” in your .ini file. Then, go to the folder “netcdf”, and perform some “ncview” commands.

# go to the output folder

$ cd /datadrive/pcrglobwb/exercise\_own\_landmask/edwin/brantas

# go to the folder netcdf

$ cd netcdf

# visualize the simulated discharge variable

$ ncview discharge\_dailyTot\_output.nc

**Performing a naturalized run (without human water abstraction and without reservoirs):**

1. We will try to perform a simulation under a naturalized condition (without anthropogenic factors). For this exercise, we will perform a PCR-GLOBWB run for the clone map that you have chosen.
2. To set/configure the simulation under a naturalized condition, you have to set the following in your ini file:

[landSurfaceOptions]

…

includeIrrigation = False

includeDomesticWaterDemand = False

includeIndustryWaterDemand = False

includeLivestockWaterDemand = False

[routingOptions]

…

# do not include reservoirs

onlyNaturalWaterBodies = True

For this run, please do the following:

* DO NOT FORGET TO CHANGE THE **outputDir** FOLDER so that you will not overwrite your other runs. Also, you may want to save your ini file under a different name (so that you will not be confused).

**Do you still want to execute more runs?**

Below are some inspirations:

* You may want to make some other runs with different forcing data. Please ask Edwin for their locations.
* You can also increase/reduce water demand by modifying the default water demand files and/or irrigation areas (e.g. multiplying by 10). Ask Edwin how to do this.