ISIMIP Flood Toolchain

Jan Volkholz

Potsdam-Institut für Klimafolgenforschung

Berlin, 2 / 14 / 2022

Introduction

We have a tool chain that can process ISIMIP runoff data and translate it to flooded areas/flood depths globally. This uses the CaMa-Flood model.

1. process the models' runoff with CaMa-Flood
2. translate discharge to return intervals of a baseline scenario
3. compare projections to prescribed protection levels
4. determine flood depths via a look up table
5. determine flooded areas via downscaling

The tool chain was built by Sven Willner.

Tool chain

- everything is controlled through a `.yml` file
- per model/driver/soc scenario/CO_2 scenario, 4 jobs are submitted

    1. `camaflood`
    2. `discharge`
    3. `mean_daily_discharge_mmpd`
    4. `downscaling`

- of these, 1. and 4. do most of the work, 2. and 3. are mainly house keeping and conversions
- the jobs and their dependencies are handled by a python program (`jobsched`)

1. `camaflood`

- for a given time slice, process all runoff (`qtot`) from the specified hyd. models/scenarios
- for spin up, the first year is repeated four times
- we turn floodplain routing on (output variable is `fldout`)
- this produces flood depth (`flddph`), river discharge (`rivout`) and total discharge (`outflw=rivout+fldout`), all on a $0.25° \times 0.25°$ grid
- `outflw` is the most important one of these, everything that follows depends on it
- currently used version of CaMa-Flood is v3.6.2
- implemented as a C++ wrapper around CaMa-Flood's Fortran Code

2. discharge and 3. mean_daily_discharge_mmpd

2. discharge
   - from the CaMa-Flood output `fldout`, generate time lines of the annual maximal discharge and the annual mean discharge
   - this generates the files `discharge/dis_annual_max.nc` and `discharge/dis_mean_daily.nc`
   - shell script `create_discharge.sh` using `cdo`

3. mean_daily_discharge_mmpd
   - spreads the mean daily discharge `outflw` across the grid cell
   - generates the file `discharge/dis_mean_daily_mmpd.nc`
   - needed to identify areas where the fit doesn't work
   - python program `mean_daily_discharge_mmpd.py`

4. `downscaling` (i)

- the job `downscaling` relates discharges to return intervals, applies protection levels, and downscales to high resolutions
- assumption: models can only be compared in terms of return intervals, see figure on slide 9
- consequently, everything is done in terms of return intervals (of the baseline time slice / scenario)
  - protection levels are given in return intervals
  - projected discharges are translated to return intervals
  - impacts, i.e. flood depths, are determined from a look up table in terms of return intervals
- a final downscale step also builds flooded areas
- the program is called `flood-processing` and implemented in C++

4. downscaling (ii)

1. from the baseline scenario, use `dis_annual_max.nc` to establish a relationship

$$\text{discharge} \longleftrightarrow \text{return level}$$

   - the fit can be done by either a GEV or a Gumbel distribution
   - all areas with a discharge $< 0.1\,\mathrm{mm/d}$ are set to zero, since the fit doesn't work properly there

2. translate the projected `dis_annual_max.nc` with the relationship to return intervals

3. apply protection levels

   - wherever the projected return interval is lower than the protection level, set values to zero; locations with return intervals above the protection remain untouched, protection is all or nothing

4. `downscaling` (iii)

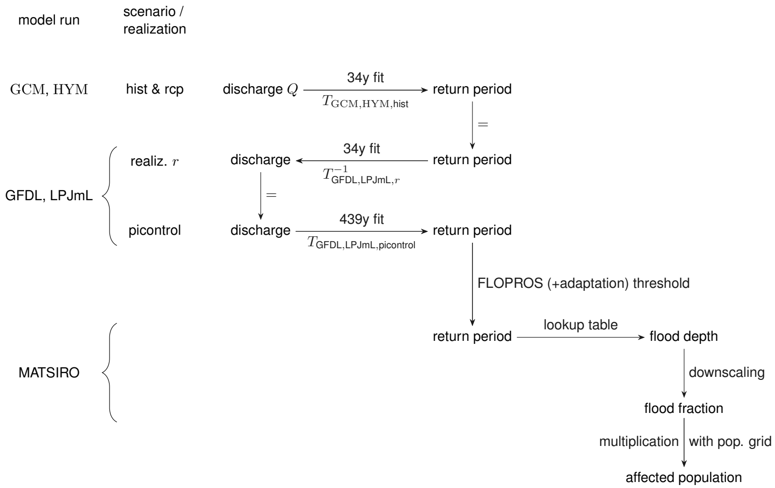
4. translate map of return intervals to map flood depths
   - `lookup-table.nc` provides a map of flood depth return levels
   - global map at 0.25°×0.25°, return intervals run from 2 to 1000 years
   - same as in Hirabayashi (2013)?
     "An inundation area associated with a specific return period was therefore calculated from a retrospective simulation of a land surface model, MATSIRO, forced by climate variables obtained from gauges and reanalysis data between 1979 and 2010 with the CaMa-Flood river routing."

     Supplements to Hirabayashi, Y., et al. (2013) *Global Flood Risk under Climate Change.* Nature Climate Change, 3, 816-821. https://doi.org/10.1038/nclimate1911

5. downscale to fine resolution
   - uses maps provided by CaMa-Flood
   - reimplementation of CaMa-Flood code
   - the resolution of these maps is 0.005°×0.005°, our target resolution must be lower than this
   - this process generates the flooded areas

4. downscaling (iv)

setup in a similar calculation



from communication with S. Willner