USER MANUAL

FLOMART

(FLOod MAp in Real Time)

Coupling the operational hydro-meteorological forecasting chain with the hydraulic flood modelling 2D (abacus of flood scenarios) for the creation of real-time flood maps

Date:
01/07/2022
Version:
2.0.4
Developed by:
CIMA Research Foundation

Authors:
Matteo DARIENZO[1],
Fabio DELOGU[1],
Simone GABELLANI[1],
Rocco MASI[1],
Francesco SILVESTRO[1],
Lorenzo CAMPO[1],
Flavio PIGNONE[1],
Giacomo FAGUGLI[1],
Andrea LIBERTINO[1]

[1]CIMA Research Foundation, Savona, Italy

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 Purpose of Flomart

Software FLOMART (**FLOod MAp in Real Time**) provides flood hazard map in real time. Its main objective is to integrate the operational hydro-meteorological forecasting and monitoring chain (e.g. FloodPROOFS) with flood hazard maps for the generation of forecasting or near-real-time flood maps. To reduce computational time of hydraulic model and support Civil Protection activities the actual hazard maps are created through an abacus of hazard maps.

1.1.1 Info

FLOMART has been developed by **CIMA Research Foundation**. Source codes of the current version v2.0.4 are written in Python (main scripts) and Matlab (preprocessing scripts) and are available at the following public repository:

https://github.com/c-hydro/flomart

For any question or feedback please contact us at fabio.delogu@cimafoundation.org or matteo.darienzo@cimafoundation.org.

1.1.2 Context and main objective

The aim of FLOMART is to couple a hydro-meteorological chain with hydraulic modeling through an abacus of flood hazard maps to provide flood maps in "real time" for flood forecasting and flood monitoring. These maps can be used to assess impacts on the territory, on the population and on infrastructures.

Hydraulic modeling by its nature requires significant costs in terms of computational time and resources, thus it would be very challenging to achieve high resolution 2D-modeling on the basis of observations or forecasts in times compatible with the operational needs of the alert.

To overcome this issue, before the real-time application, FLOMART proposes to create an abacus of flood scenarios with hydraulic 2D modelling, which aims to represent all hydraulic conditions (flood levels and areas for increasing values of discharge) for the studied river. Then, in 'real time', this abacus is coupled with the stream-flow forecasts at the river sections of interest and/or the hydrometric level observations, to create predicted scenarios of flooded areas (hence, the maximum expected flooding).

This method can be operationally used, since the processing times are compatible with the operational needs of the decision makers in the forecasting field. It has been initially developed and tested for different basins in Liguria Region in Italy (e.g. the Entella basin) showing promising results with good performance in simulating the extent of the flood. After further development, it is now operationally applied in Liguria and also to the Foglia and Chienti basins (Marche Region).

As mentioned, FLOMART method is retrospectively initialized with the creation of the abacus of flood hazard maps (e.g., of 495 maps). In particular, the method begins with the creation of a limited number of maps with an assigned return time T (e.g., 12 maps) obtained from 2D hydraulic modeling, for the percentiles, for example, from 80 (return time 5 years) to 99.8 (return time 500 years). All other maps (for statistical percentiles not calculated with hydraulic simulations) are then reconstructed with an interpolation procedure with a 1-year step.

FLOMART initialization also requires the identification, within the analyzed area, of the control sections (for which measurements and/or forecasts are available in real time) and the consequent subdivision of the basin into sub-basins, hence the areas controlled by each section.

In real time, the main steps computed by FLOMART are summarized here-after:

1. Get the hydrological forecast (flood peak) for each control section.

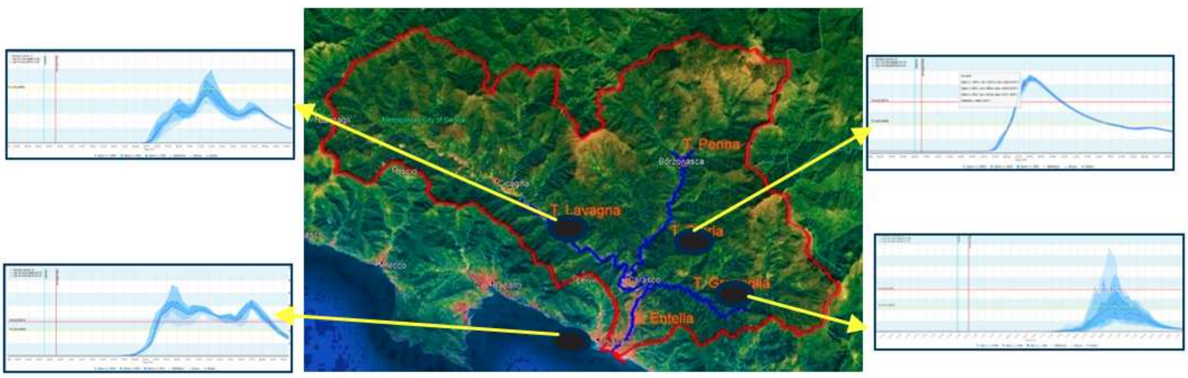


Figure 1.1: Step 1 of Flomart real-time.

2. For each section (or for each river branch), selection from the flood maps abacus of the closest map to the forecasted value.

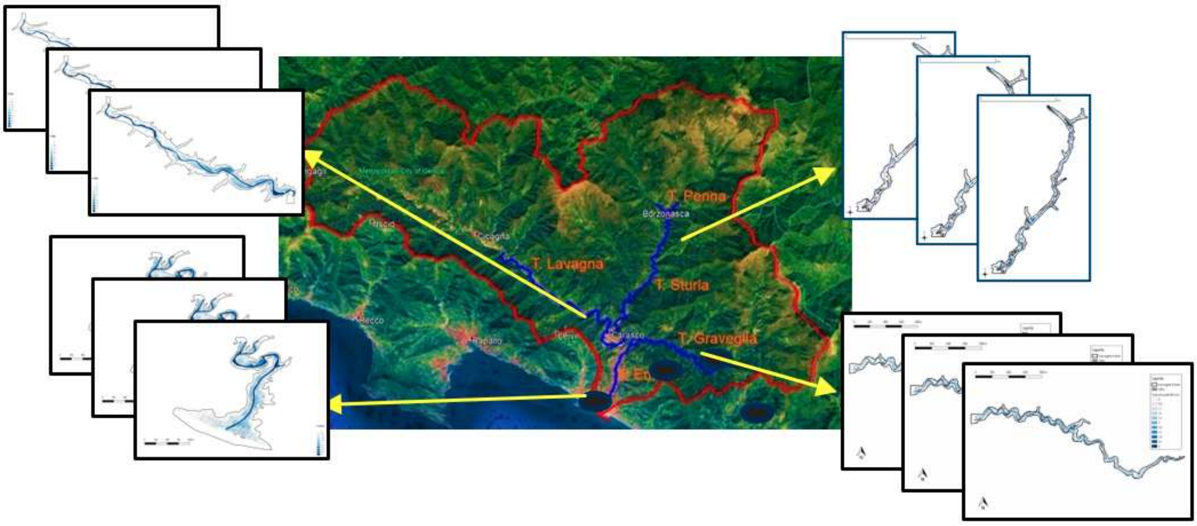


Figure 1.2: Step 2 of Flomart real-time.

3. Merge the selected maps and create the final forecasted flood scenario.

Figure 1.3: Step 3 of Flomart real-time.

Flomart is designed to be implemented in existing operational chains. But it can also be used retrospectively to analyze past flooding events.

The final output of the method is the flood map, which can be provided at different time scale, depending on operational purposes, in jpeg and .tiff formats. Moreover, the essential information on the selected flood map (e.g., return period and maximum discharge for each studied section, and subtended area) is provided through a *json* file.

Figure 1.4 illustrates an example of the .tiff output file of Flomart applied to the Foglia river, in Marche region, Italy.

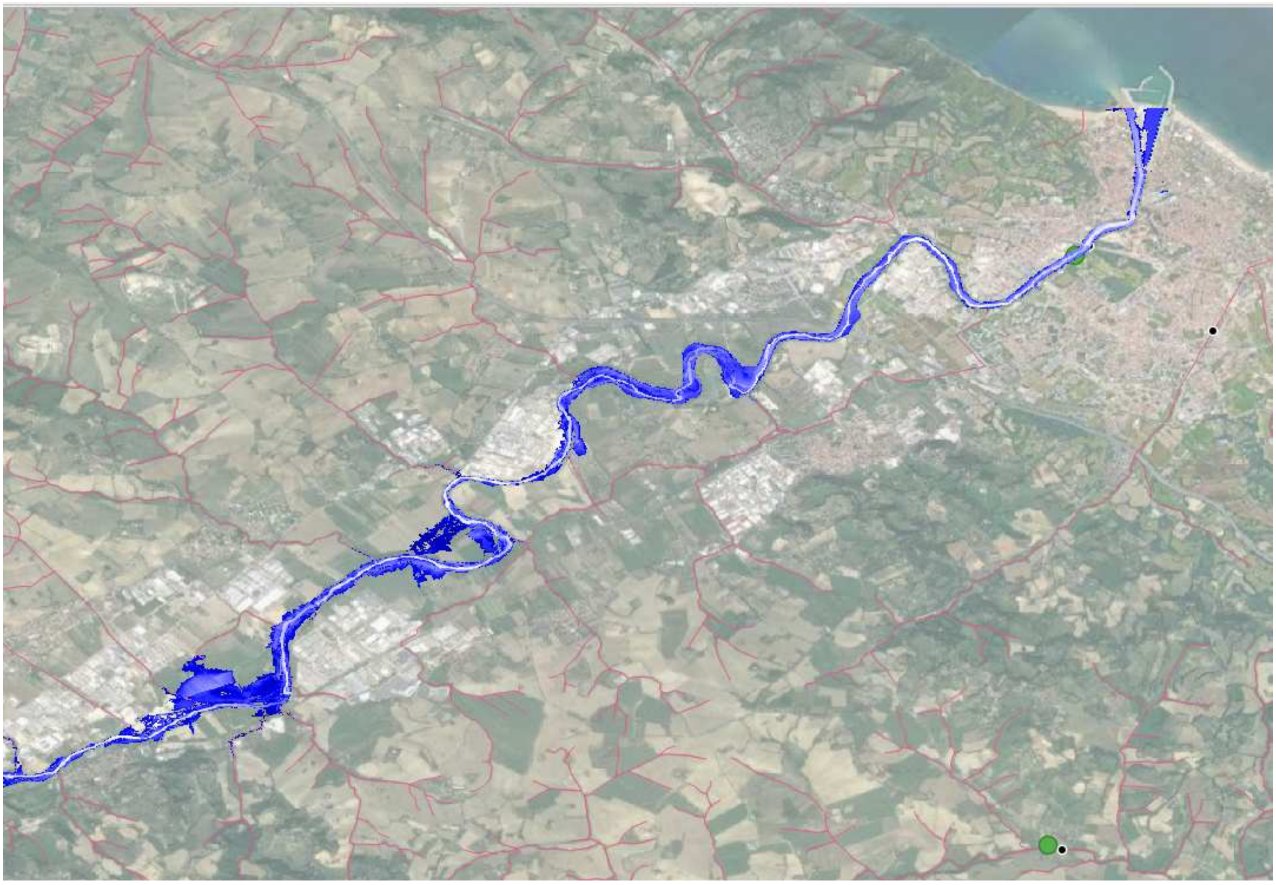
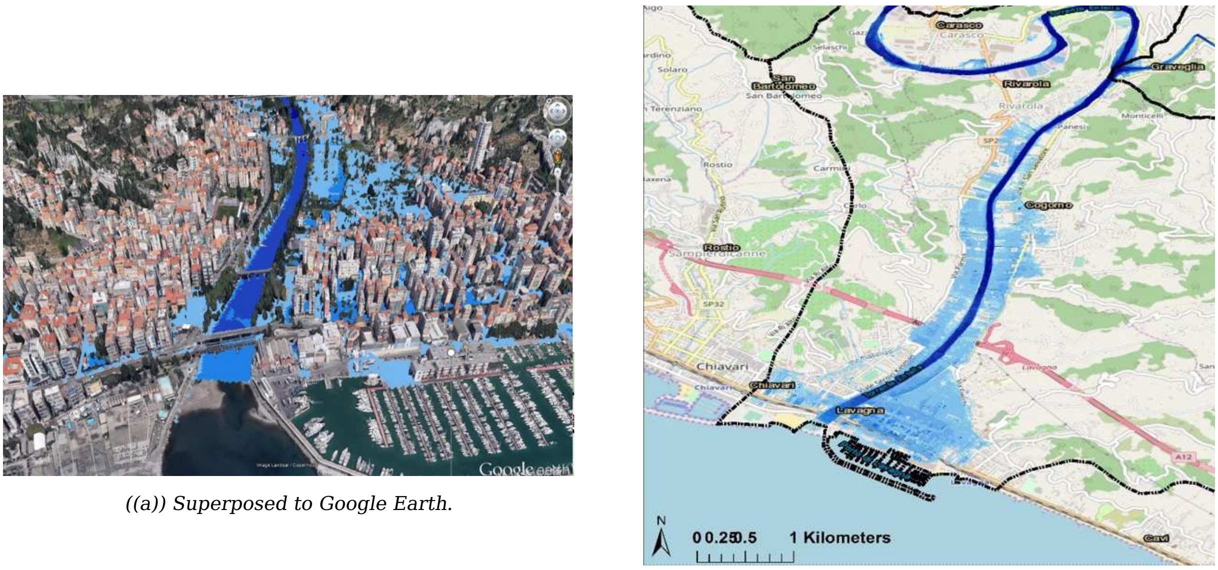


Figure 1.4: Application of Flomart to Foglia Basin (Marche Region), superposed to OpenStreetMap.

Output maps can be used for different purposes and visualizations. For example the obtained rasters can be superposed to existing georeferenced layers 2d or 3d, as illustrated in Figure 1.5.



((a)) Superposed to Google Earth.

((b)) Superposed to OpenStreetMap.

Figure 1.5: Flomart applied to Entella river (Liguria, Italy).

1.1.3 Operational uses

On demand, the outputs (the flood maps) can eventually be visualized through the existing web platforms. As an example, the results for the Foglia and Chienti rivers are now available on the **MyDewetra web platform** (version STAGING), developed by CIMA Foundation and funded by Italian Civil Protection DPC, at the following link (credentials required):

https://https://staging.mydewetra.org/apps/dewetra2/index_mydewetra.html#!#%3Fskin=3

1.1.3.1 Login

Platform MyDewetra needs initial login (see Figure 1.6). After that, the user 'hat' must be selected. Flomart is available for Marche region under the following hats:

- *user cima*
- *cf_centrale*
- *user regione Marche*

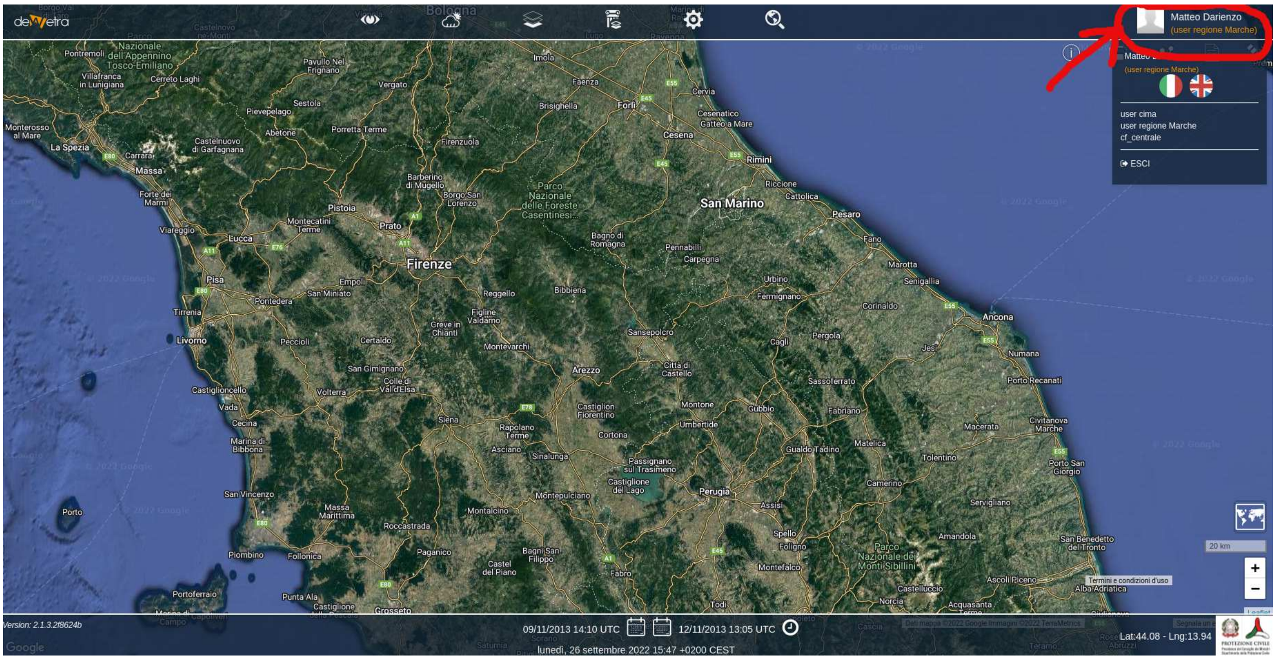


Figure 1.6: Dewetra platform main page. Login is required on the top-right corner.

1.1.3.2 Define time window

Once you are logged in, you are asked to define at the bottom of MyDewetra page (see example in Figure 1.7) the **time window** where to search for the desired Flomart map.

You will find two adjacent buttons with calendars. On the left calendar select the timing from which you want to start the search. On the right one, select the timing to which you want to end the search.

This will represent the current time of the analysis.

Moreover, by clicking on the clock symbol (on the right) it is possible to locate the ending time to the actual real time and the search starting time to 24 h earlier.

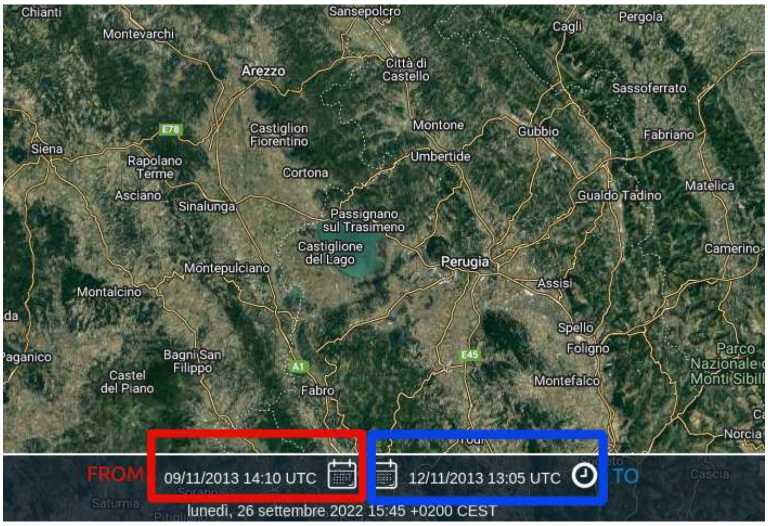


Figure 1.7: Definition of the time window where to search for Flomart map: "from" (choose time in the calendar on the left) "to" (choose time in the calendar on the right).

Please, notice that times are intended as **UTC**. As an example, Figure 1.8 summarizes the relation between utc and local time (Central European Time and Central European Summer Time) for Marche region as for 2022.

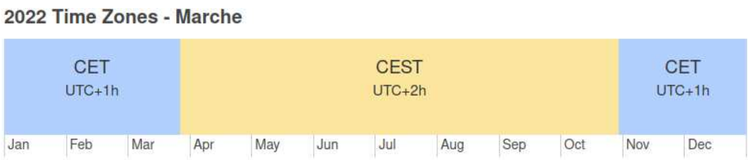


Figure 1.8: Local time respect to UTC in Marche region (taken from https://www.timeanddate.com/time/zone/@3174004).

1.1.3.3 Upload Flomart layers

Flomart maps are located within the Forecast ("PREVISIONI") dynamic layers, available from the menu on the top of the screen:

Once you have clicked on PREVISIONI, several panels appear. You will need to click on the panel "HAZARD MARCHE" (see red arrow in Figure 1.9).

Moreover, Flomart uses as input hydrographs the results of the hydrological forecasts of Marche operational chain, which are located within the panel "MODELLI IDRO MARCHE DETERMINISTICI" (circles in blue in the same figure).

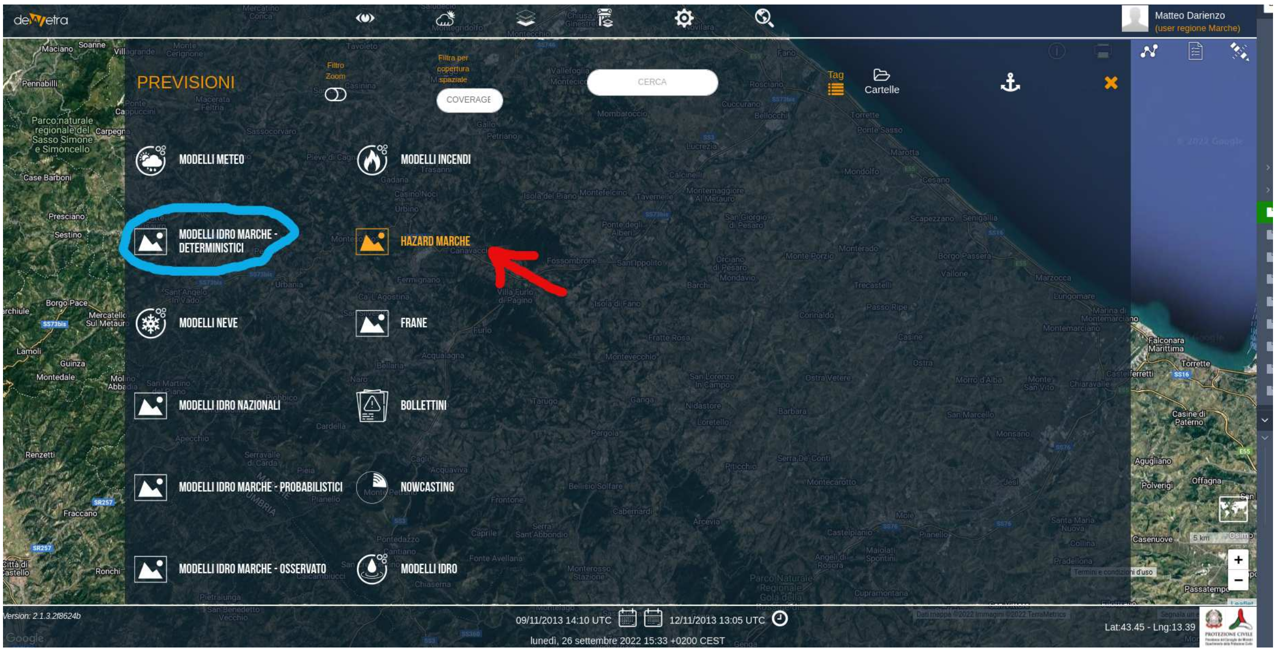


Figure 1.9: Screenshot from MyDewetra platform where the dynamic layers with the forecasts are located (click on HAZARD MARCHE for Flomart maps for Marche region).

In particular, for Marche region there are two forecasting chains (Flood Proofs Marche):

- **Deterministic Cosmo/Lami:**

  FP-DET.COSMOI2

- **Deterministic Ecmwf:**

  FP-DET.ECMWF

Each chain computes one forecast per day (usually in the early morning, around 04:00 - 09:00 utc).

Then, a bit later in the morning, the results of both forecasts (e.g., simulated hydrographs at the control sections) are used as dynamic inputs for Flomart application. Thus, a different map is produced for each chain. As the hydrological forecast is daily implemented, also Flomart maps are generated once per day.

For instance, two case studies have been tested and operationally implemented for Marche region: **the Chienti river** and **the Foglia river**. Each river has its own Flomart map. Thus, in total four maps are available every day within the panel "HAZARD MARCHE".

Flomart maps layers Marche (see Figure 1.10):

- FP FLOMART - MARCHE - CHIENTI - DET. COSMO I2
- FP FLOMART - MARCHE - CHIENTI - DET. ECMWF
- FP FLOMART - MARCHE - FOGLIA - DET. COSMO I2
- FP FLOMART - MARCHE - FOGLIA - DET. ECMWF

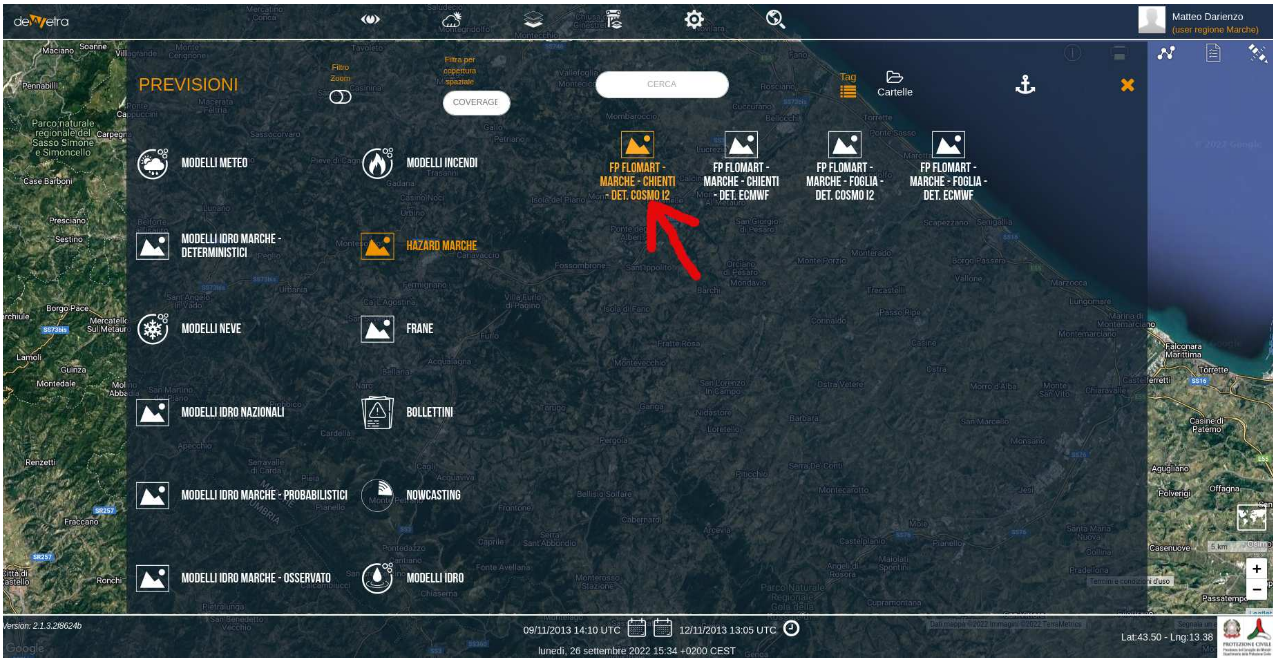


Figure 1.10: Screenshot from MyDewetra platform where the four maps Flomart for Marche region are located.

When opening one of the available layers (Figure 1.11), e.g., layer FP FLOMART - MARCHE - FOGLIA - DET. COSMO I2, the panel of Figure 1.10 with the list of the four layers disappears. Instead, on the top left, a panel with the name of the selected layer (and a few additional settings) appears.

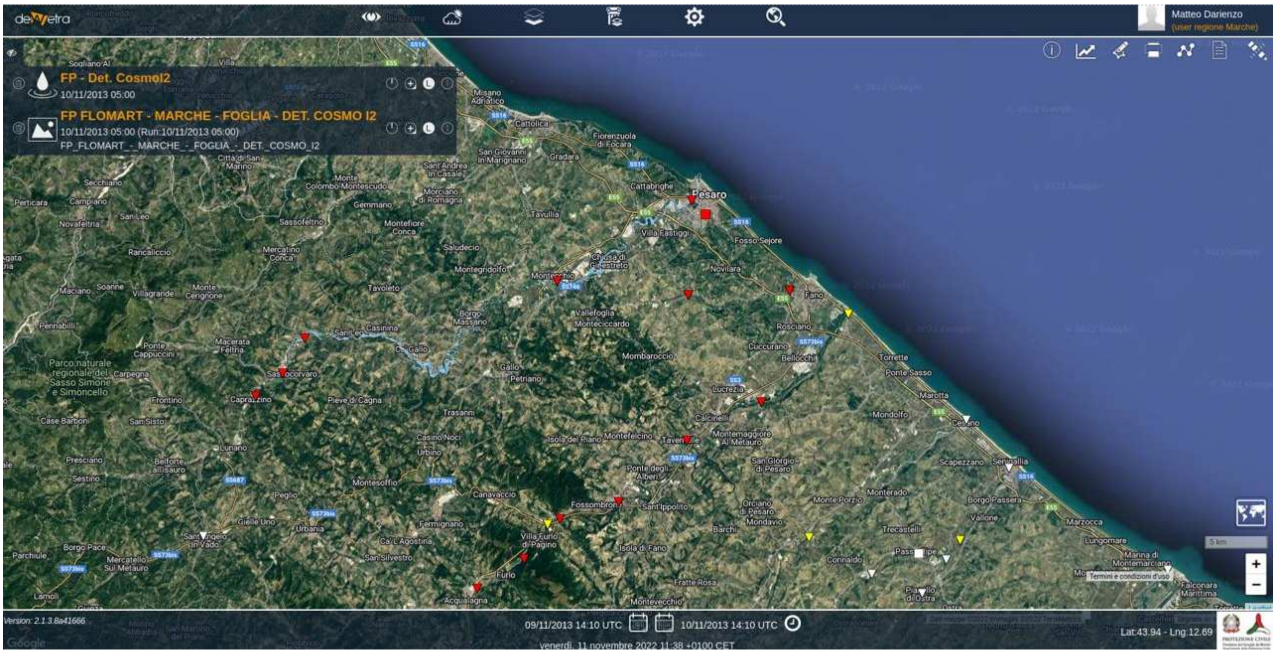


Figure 1.11: Screenshot from MyDewetra platform when uploadind one Flomart map (in this case, FP FLOMART - MARCHE - FOGLIA - DET. COSMO I2).

1.1.3.4 Zoom in

The Flomart map will not probably be visible unless zooming in to the interest zone. To do that, you can zoom in by means of the dewetra zoom tool, by clicking on the symbol "+" on the layer's panel (see Figure 1.12). This will zoom to the Flomart map extent (Figure 1.13).

Figure 1.12: Screenshot from MyDewetra platform with the layer Flomart Marche and the zoom tool.

Figure 1.13 illustrates an example of a map computed by Flomart for Foglia river during the event of November 2013.

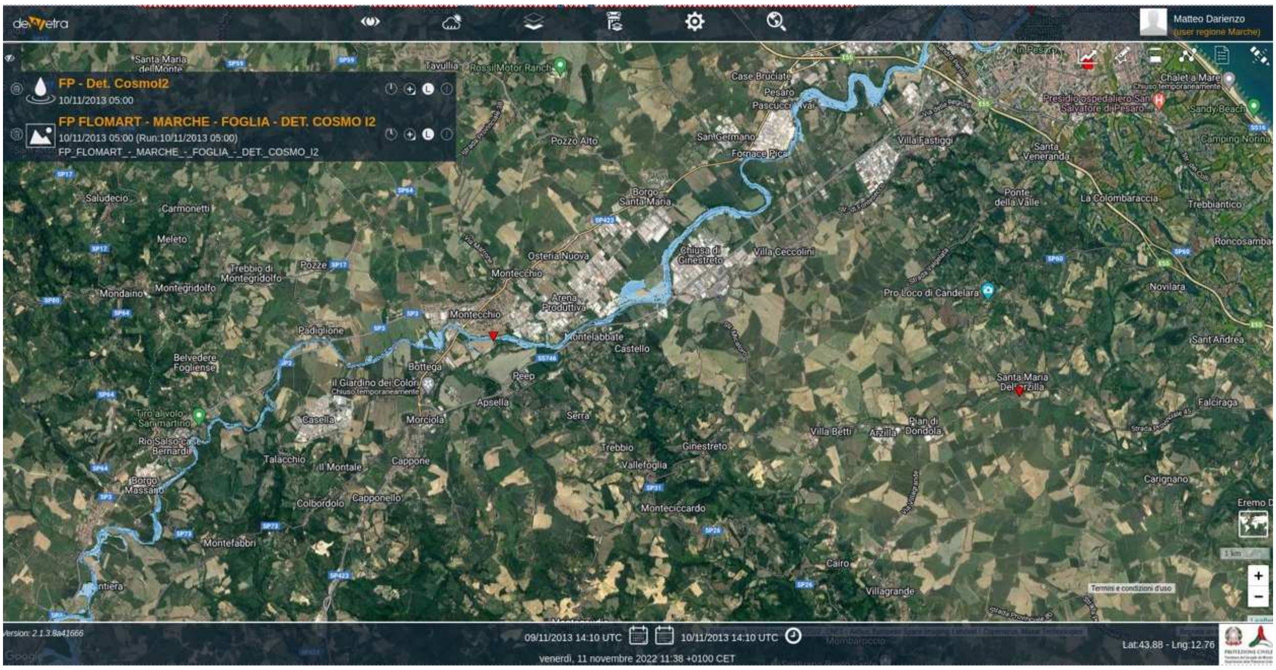


Figure 1.13: Screenshot from MyDewetra platform with the layer Flomart Marche, after zooming in.

1.1.3.5 Legend

By clicking on the button **"L"** located on the layer's panel the legend will appear, with some additional information about the map (Figure 1.14).

Water levels values are represented through a palette of blues, from 0 (transparent) to 8 m (dark blue). Also values between 0 and 0.1 are affected by 10% transparency.

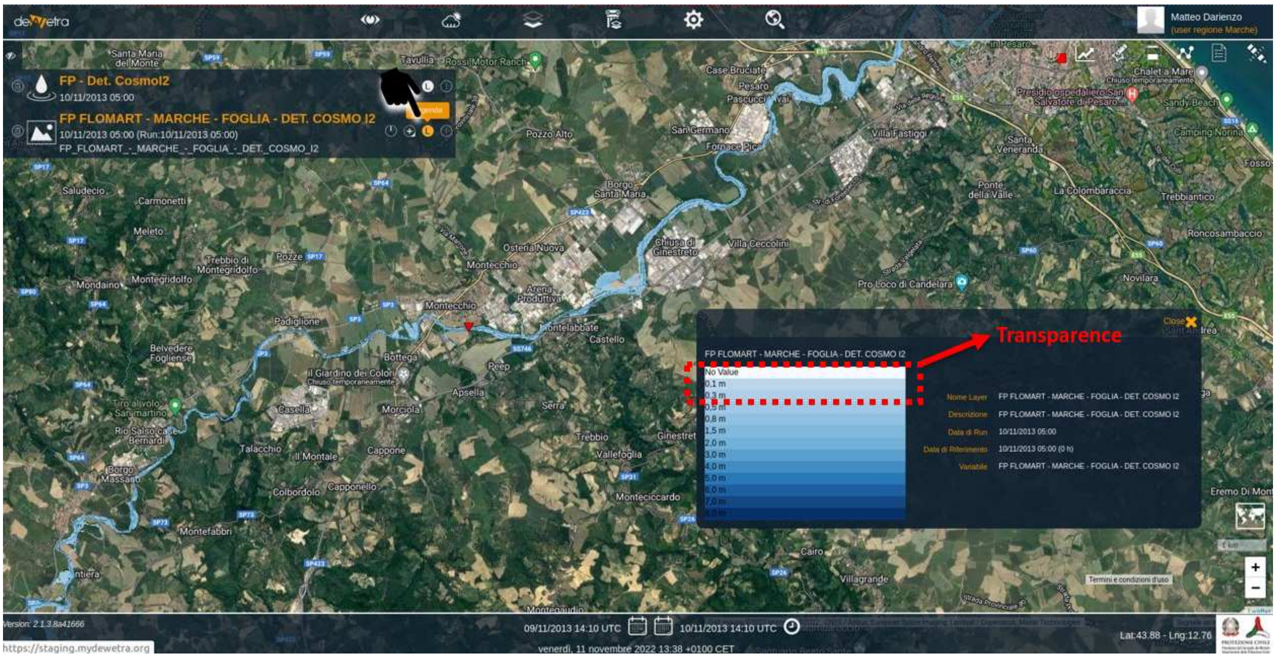


Figure 1.14: Screenshot from MyDewetra platform with the layer Flomart Marche and the legend tool.

1.1.3.6 Read water level values from map

By clicking on the button **"i"** (which stands for "Information") located on the top right of MyDewetra platform you will enable the "get map values" tool (see Figure 1.15, here the button has been highlighted). In particular, by clicking on the map at the point of interest a new panel will appear on the right with the maximum water level forecasted at that section within the forecast window, which usually is +48h, and some additional information on the map.

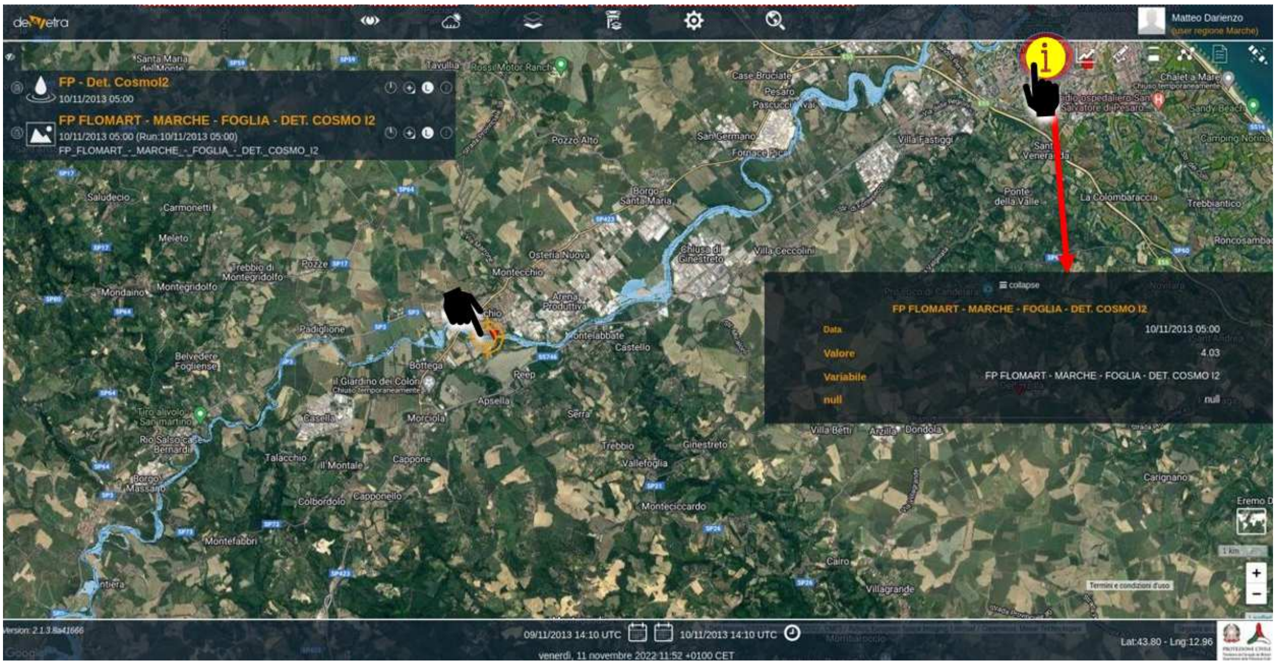


Figure 1.15: Screenshot from MyDewetra platform with the layer Flomart Marche and the "INFO" tool.

1.1.3.7 Change map and unzoom

Then, if we want to choose another layer (another catchment), you will need to come back to the panel with all four layers. Nevertheless, in order to see all layers you will need to unzoom the myDewetra view by turning off the zoom option button (see Figure 1.16 and Figure 1.17):

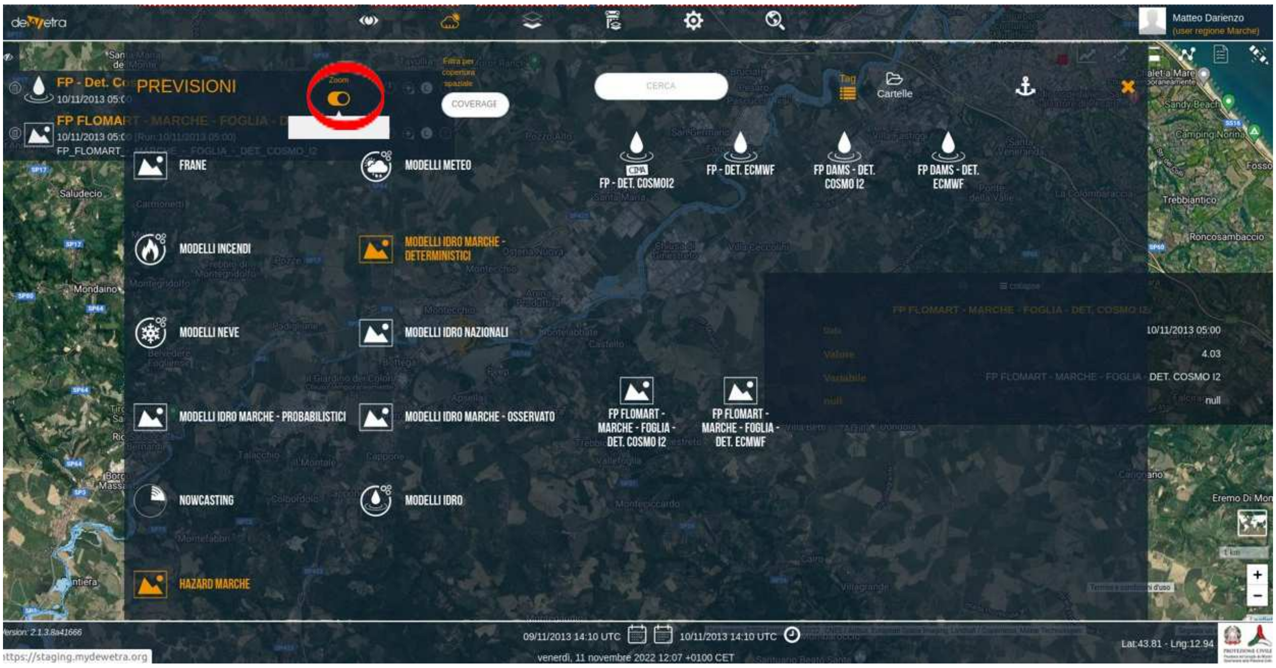


Figure 1.16: Screenshot from MyDewetra platform with the unzoom option. Remember to turn off the zoom in order to see all available layers.

Then select another layer, e.g., FP FLOMART - MARCHE - CHIENTI - DET. COSMO I2 (Figure 1.17).

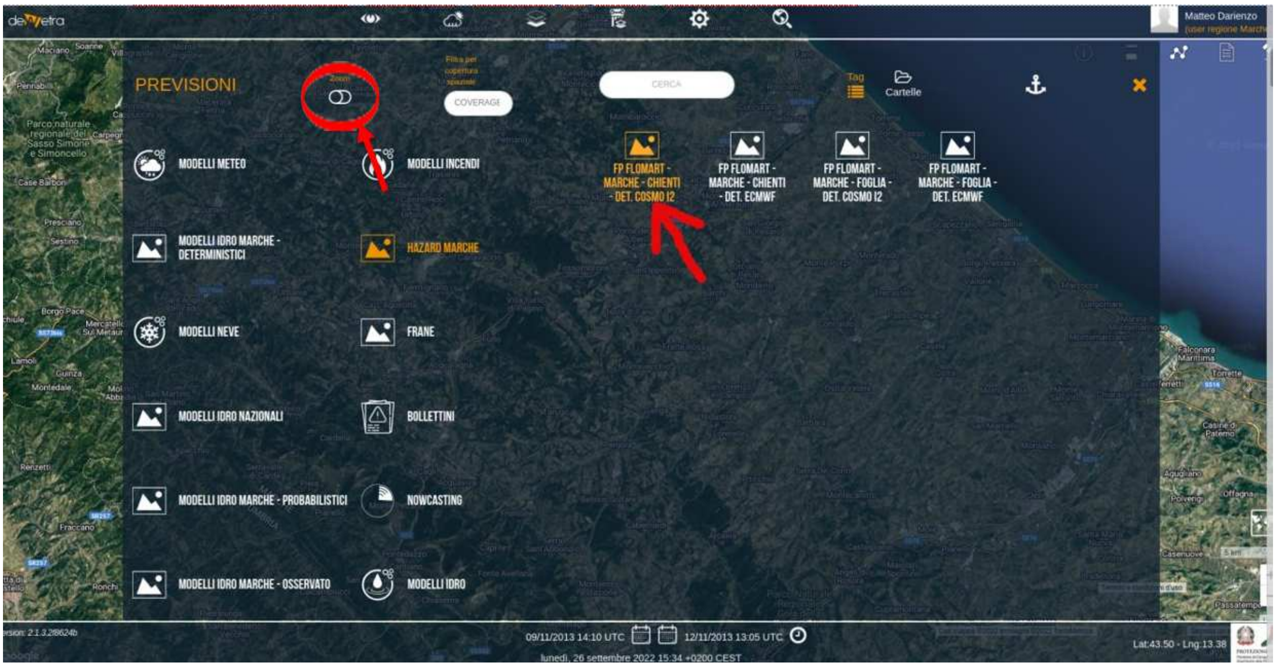


Figure 1.17: Screenshot from MyDewetra platform after having unzoomed.

The new map for Chienti river will appear (Figure 1.18).

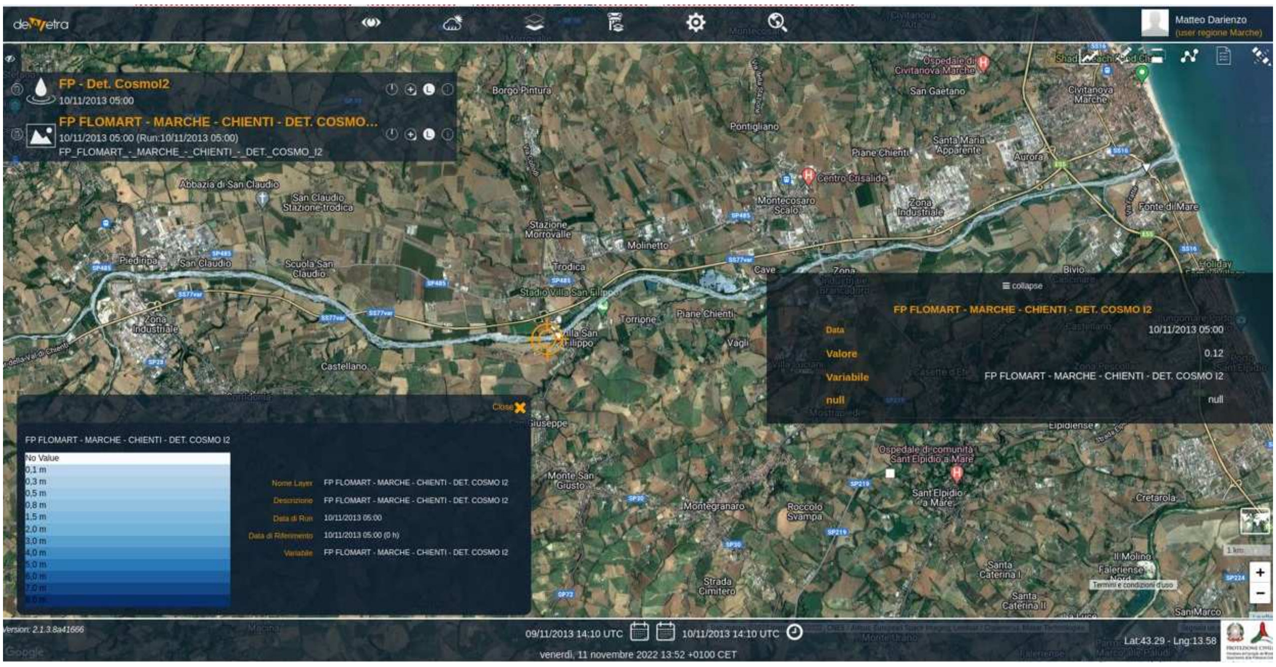


Figure 1.18: Screenshot from MyDewetra platform with the map computed by Flomart for Chienti river, Marche Region, for the simulated flooding event as forecasted on 10 November 2013.

Then if moving the search window to the next day (11/11/2013) we will obtain a new map, Figure 1.19, (new hydrological forecast, new flood map). And we can see here much larger flood extent. We can also zoom further to catch more details (Figure 1.20).

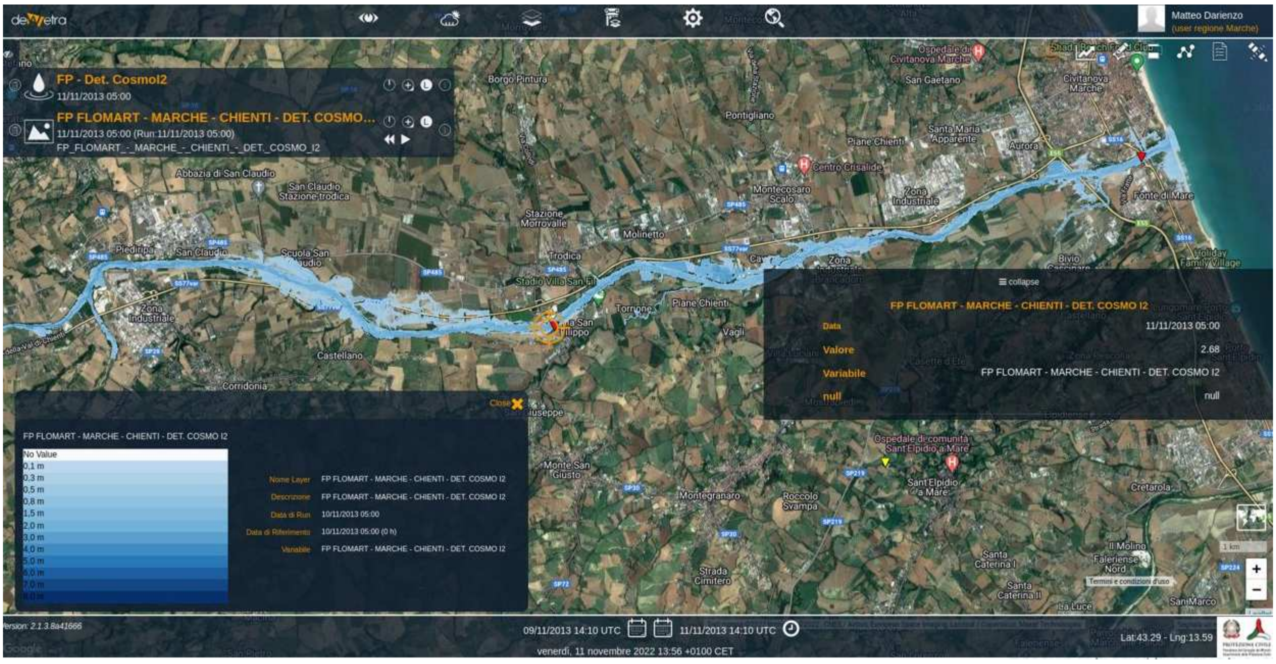


Figure 1.19: Screenshot from MyDewetra platform with the map computed by Flomart for Chienti river, Marche Region, for the simulated flooding event as forecasted on 11 November 2013.

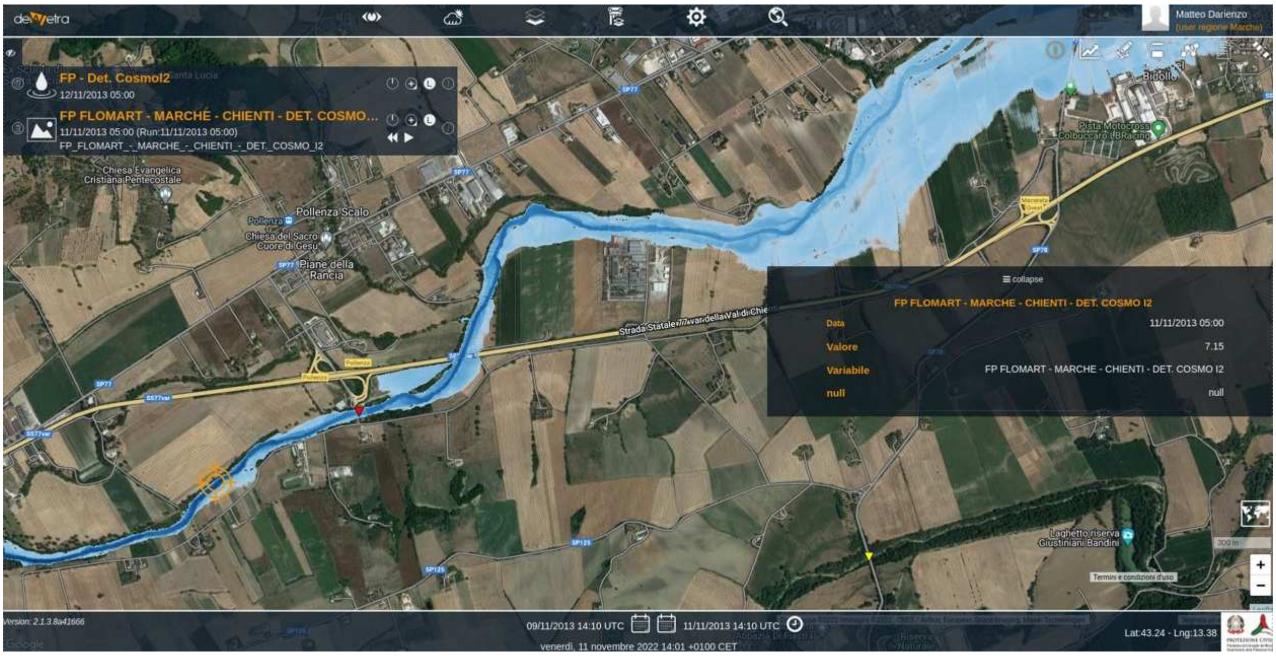


Figure 1.20: Screenshot from MyDewetra platform with the zoom on the map computed by Flomart for Chienti river, Marche Region, for the simulated flooding event as forecasted on 11 November 2013.

1.1.3.8 No-flood scenarios

Being Flomart operationally implemented with the production of daily maps, the majority of these maps should represent the conditions of events with return time $T << 1$ year.

However, since the intermediate scenarios between 0 and 1 year are not here available, a linear interpolation is computed, according to the forecasted discharge, between a map of levels with zeros (corresponding to $T = 0$) and the minimum available scenario (e.g., $T = 1$ year). This will linearly decrease the values of water level of the minimum scenario but not the flood extent.

Most of the maps will have values of the order of 0.001 - 0.1 m, for which a transparency of 10-20% is applied. For this reason, please, check the water level values, not only its extent.

However, it is worth to mention that Flomart has been conceived and designed for generating flood hazard maps for extreme flood events (return times $T > 1 - 5$ years), not for monitoring the water levels during ordinary conditions.

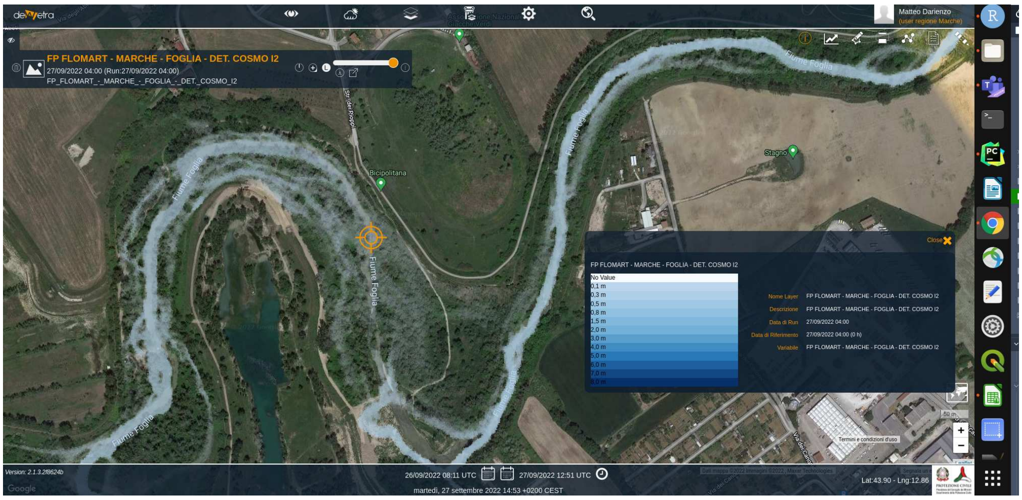


Figure 1.21: Screenshot from MyDewetra platform with the zoom on the map computed by Flomart for Foglia river, Marche Region, for the simulated event with return time $T << 1$. Transparency is automatically applied to the water depth values below 0.1 m.

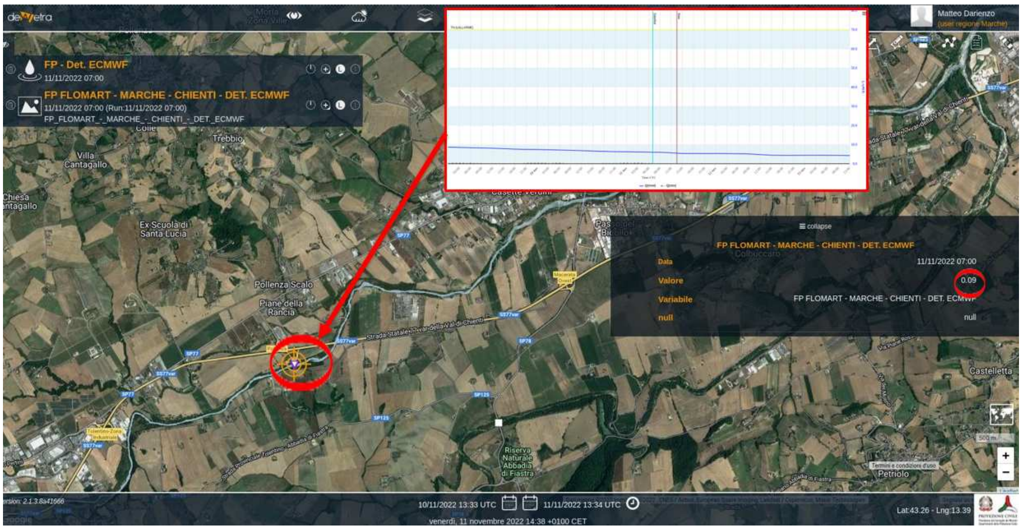


Figure 1.22: Screenshot from MyDewetra platform with the zoom on the map computed by Flomart for Chienti river, Marche Region, for the simulated event with return time $T << 1$. Transparency is applied to the water depth values below 0.1 m.

1.1.3.9 No data

If no map is available for that day, (because the hydrological forecast that day did not run successfully or is still running, or due to other bugs related to the selection of hazard map and the merging, or nan values), a message of ERROR will appear at the top of MyDewetra platform saying: ***"no data found"***. Please, in that case try uploading the map a bit later in the day, or if still not available, do not hesitate to contact us.

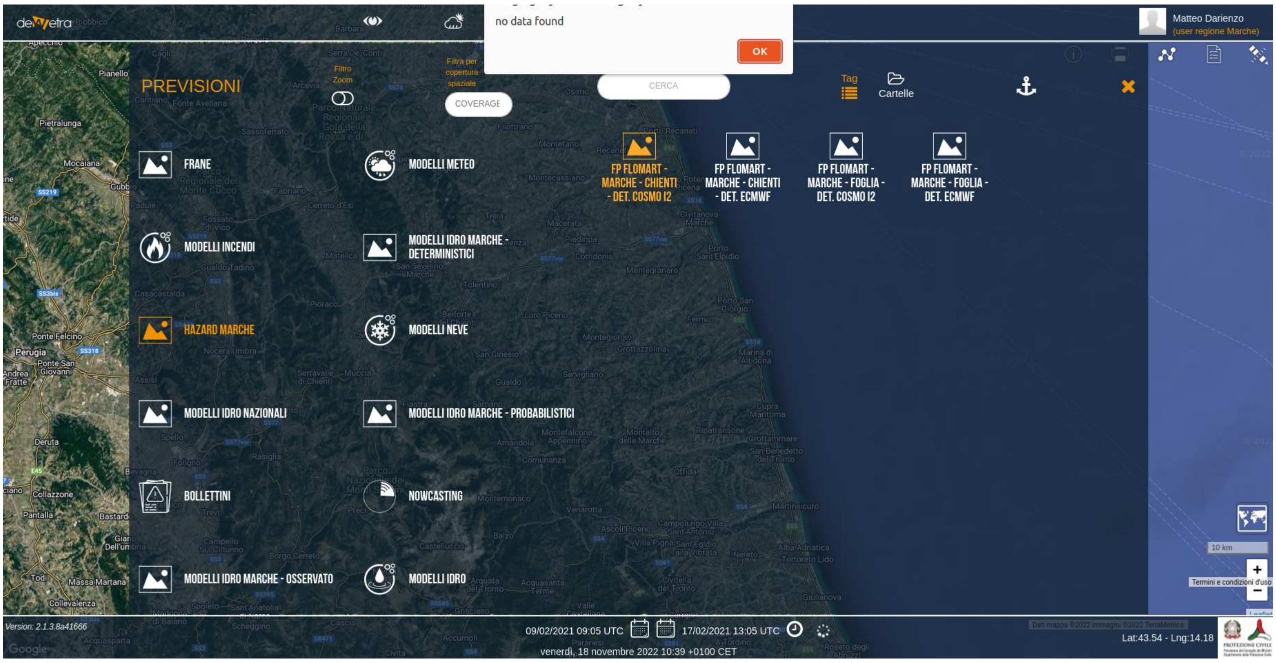


Figure 1.23: Screenshot from MyDewetra platform with the ERROR obtained if tha Flomart map is not available.

1.1.3.10 Extra: superposing Flomart map with static layers of potential exposed assets

Finally, it is possible to upload (superposing to the Flomart map) a few additional layers with static data of potentially exposed assets (such as schools, hospitals, roads, railway, etc.). This feature can certainly further developed if interested.

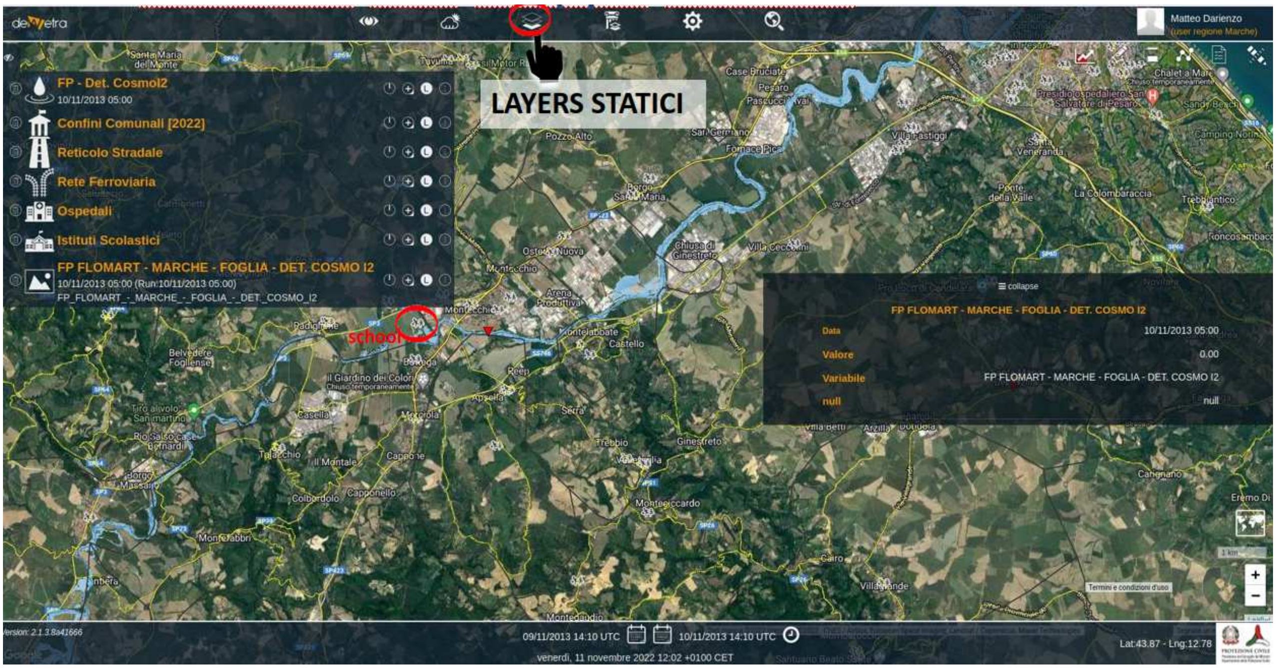


Figure 1.24: Screenshot from MyDewetra platform with Flomart map and additional static layers with potentially exposed assets.

1.2 Structure of Flomart scripts and files needed

The following sections and the next chapters will describe the structure of Flomart source scripts and a brief guideline for the correct preparation and configuration of input files (both static and dynamic). Finally, Flomart output will be illustrated through the case studies of Chienti river and Foglia river (in the Marche Region, Italy).

1.2.1 Structure description

Flomart repository (*flomart-2.0.4*) (available at https://github.com/c-hydro/flomart) contains two folders (***app_flomart_main*** and ***app_flomart_preprocessing***) with the main and preprocessing scripts, and 4 .rst files ('AUTHORS.rst', 'CHANGELOG.rst', 'LICENSE.rst', 'README.rst') with some general information.

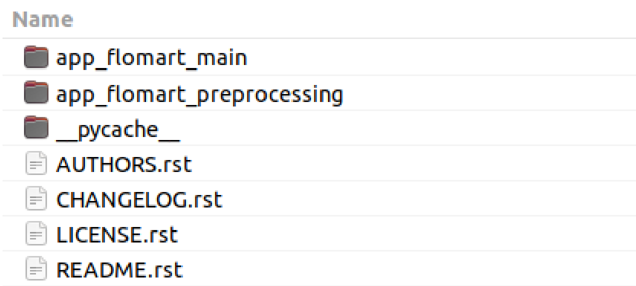


Figure 1.25: Contents of folder 'flomart-2.0.0'.

1.2.2 Main program file and modules (PYTHON): *app_flomart_main*

1.2.2.1 Main

The main program of Flomart and its modules are located within folder *app_flomart_main* and are written in Python language v.3. The main script file is called:

- *app_flomart_main.py*

This script defines the main function ***main()*** which calls other subfunctions located in several other modules (see the full list in the next subsection).

The functions used for Flomart initialization (in order to organize all static geographical and hydrological datasets) are:

- *get_args()* (defined in the same file) to get the script arguments;
- *read_file_json()*, from module *lib_utils_io.py*, to read the Flomart configuration .json file;
- *set_logging_file()*, from module *lib_utils_logging.py*, to set the algorithm logging;
- *set_time()*, from module *lib_utils_time.py*, to organize the timings of the run (time range);
- *organize_geo()*, from Class *DriverGeo()*, defined in module *driver_data_io_geo.py*, to read the static input geographical information and create the geographical datasets;

Then for each instant of the defined time interval, it iteratively calls the following functions:

- *organize_discharge()* from Class *DriverDischarge()* defined in module *driver_data_io_source.py*, to read the dynamic data (the hydrographs at the sections of interest) and create the discharge datasets;
- *organize_scenario_datasets()*, *compute_scenario_map* and *dump_scenario_map* from Class *DriverScenario()* defined in module *driver_data_io_destination.py*, in order to, respectively, create the scenario datasets, compute the scenario map and save results (for each timing it saves the scenario map in .tiff and/or .png and a .json file summarizing the main features of the generated scenario).

1.2.2.2 Modules

Here is the list of all modules that are needed in order to run the main program 'app_flomart_main':

Drivers with classes:

- *driver_data_io_destination.py* (contains Class *DriverScenario*)
- *driver_data_io_geo.py* (contains Class *DriverGeo*)
- *driver_data_io_source.py* (contains Class *DriverDischarge*)

Other libraries:

- *driver_type_io.py* (contains Class *DriverType*)
- *lib_info_args.py*
- *lib_utils_colormap.py*
- *lib_utils_generic.py*
- *lib_utils_geo.py*
- *lib_utils_hazard.py*
- *lib_utils_hydraulic.py*
- *lib_utils_hydrology_ascii.py*
- *lib_utils_hydrology_generic.py*
- *lib_utils_hydrology_json.py*
- *lib_utils_hydrology_mat.py*
- *lib_utils_io.py*
- *lib_utils_logging.py*
- *lib_utils_plot.py*
- *lib_utils_section.py*
- *lib_utils_system.py*
- *lib_utils_time.py*
- *lib_utils_tr.py*
- *lib_utils_ts.py*

1.2.3 Configuration "json" file

Flomart settings (such as the name and directory of all input and output files, and the type of available real-time information) need to be specified in the Flomart configuration *.json* file:

- *app_flomart_configuration_deterministic_example_simulated_FOGLIA.json*

The name of the configuration file is user-defined. A more detailed description of each setting and key included in this file will be presented in Chapter 4.

1.2.4 Input/Output data

Before running Flomart, the user needs to make sure all necessary input files are correctly completed and located. We propose here a possible structure of the data. The folder designed for input/output data includes three subfolders: ***data_dynamic***, ***data_static***, ***log***, whose structure will be described in the following subsections.

Please, notice that all directories of the input and output files need to be defined by the user in the configuration *.json* file. Thus, the user is also free to modify the here presented structure, in particular for operational uses.

Figure 1.26: Contents of the folder (e.g., data_Marche_Foglia') with input data 'data_dynamic', 'data_static', 'log'.

1.2.4.1 Dynamic data: folder 'data_dynamic'

Within folder 'data_dynamic' you will find three other subfolders (see Figure 1.27) called: ***source***, ***outcome*** and ***ancillary*** (see Figure 1.27).

Figure 1.27: Contents of the folder containing the input dynamic data (hydrographs and waterlevel time series.

The user is required to create a folder (arbitrarily named, e.g., 'discharge_run_obs_ws') containing all files of dynamic data (e.g., files with modelled or observed hydrographs and water level time series, if any) and to locate this new folder within subfolder 'source'.

You are then required to fill in the Flomart configuration file all paths, the date formats and structure used to locate the dynamic data files), and type of files (e.g., .json, .mat, ascii).

Figure 1.28 presents an axample of dynamic file (.json format) containing the hydrograph at a section of interest.

Figure 1.28: Example of input dynamic file with the hydrograph at a section of interest (Foglia3, close to Foglia River's delta).

Instead, the other two subfolders (*ancillary* and *outcome*) will be filled once Flomart main program has been executed. They will contain, respectively: the temporary results files (saved while running Flomart) and the final results (with the scenario maps, .tiff or png, and corresponding information .json). For both you are asked, as for the input files, to define the corresponding paths in the configuration file.

1.2.4.2 Static data: folder 'data_static'

The input static data need to be located within folder 'data_static'.

Figure 1.29: Folder with static data.

1.2.4.3 Log file: Folder 'log'

Folder 'log' contains the .txt file with the log, created while executing Flomart.

```
2022-07-13 11:05:17,416 app_flomart_logger INFO     ==========================================================================    app_flomart_main.py:[80    -
main                ()]
2022-07-13 11:05:17,416 app_flomart_logger INFO     ==> FLOMART (Version: 2.0.0 Release_Date: 2022-07-01)                         app_flomart_main.py:[81    -
main                ()]
2022-07-13 11:05:17,416 app_flomart_logger INFO     ==> START ...                                                                 app_flomart_main.py:[82    -
main                ()]
2022-07-13 11:05:17,416 app_flomart_logger INFO                                                                                   app_flomart_main.py:[83    -
main                ()]
2022-07-13 11:05:17,416 app_flomart_logger INFO     ---> Set time run ...                                                         lib_utils_time.py:[29    -
set_time            ()]
2022-07-13 11:05:17,416 app_flomart_logger INFO     ----> Time 2013-11-14 00:00 set by argument                                   lib_utils_time.py:[32    -
set_time            ()]
2022-07-13 11:05:17,417 app_flomart_logger INFO     ---> Set time run ... DONE                                                    lib_utils_time.py:[45    -
set_time            ()]
2022-07-13 11:05:17,816 app_flomart_logger INFO     ---> Organize geographical datasets ...                                       driver_data_io_geo.py:[427    -
organize_geo        ()]
2022-07-13 11:05:17,816 app_flomart_logger INFO     ----> Domain "Foglia" ...                                                     driver_data_io_geo.py:[432    -
organize_geo        ()]
2022-07-13 11:05:17,816 app_flomart_logger INFO     ----> Domain "Foglia" ... SKIPPED. Datasets previously computed.              driver_data_io_geo.py:[457    -
organize_geo        ()]
2022-07-13 11:05:19,838 app_flomart_logger INFO     ---> Organize geographical datasets ... DONE                                  driver_data_io_geo.py:[462    -
organize_geo        ()]
2022-07-13 11:05:19,847 app_flomart_logger INFO     ---> Organize simulated discharge datasets [2013-11-14 00:00] ...             driver_data_io_source.py:[718    -
organize_discharge_sim()]
2022-07-13 11:05:19,847 app_flomart_logger INFO     ----> Domain "Foglia" ...                                                     driver_data_io_source.py:[728    -
organize_discharge_sim()]
2022-07-13 11:05:19,848 app_flomart_logger INFO     ----> Domain "Foglia" ... SKIPPED. Data previously computed                   driver_data_io_source.py:[905    -
```

Figure 1.30: Example of log file while executing Flomart.

1.2.5 Preprocessing scripts (MATLAB): *app_flomart_preprocessing*

Before running Flomart main program, a pre-processing analysis is required. This pre-processing aims at creating a file called ***info_{domain_name}.mat***, which represents an essential static input file, specific to each case study, for the execution of Flomart program.

Indeed, this file contains the geographical information of the domain and of the selected control sections along the river and the corresponding subdivision of the basin which defines the affectation and drainage areas. Scripts for the pre-processing are located within folder ***app_flomart_preprocessing*** and are written in Matlab language.

The main pre-processing script file is called:

- app_flomart_preprocessing.m

which makes use of other small modules/functions:

- aree_drenate.m (function to compute the drainage areas for each selected section of the domain.
- latlon2utm.m (function to convert lat/lon coordinates to utm coordinates).
- Continuum_getMap_NC.m (function to get the map from netcdf output of Continuum modelling).
- example_generation_flood_map.m (function to test at the end of the preprocessing the operational generation of the flood map).
- Get_MatrixCoordinates_FromPoints_NoNaN.m (function to convert lat/lon coordinated to matrix indices from ascii domain).
- grid_Qindex_generation.m (function to compute the gridded variable Qindex from a historical analysis of simulated discharge over the domain).
- ReadAsciiRaster.m (function to read ascii raster file).

This script will be better discussed step by step in Chapter 2. It requires some information (or files) such as: dem, choice, area, Qindex, lat/lon, hazardmap, sections registry, hydrometer's coordinates and the level of the hydrometric zero.

CHAPTER 2

PREPARATION OF STATIC INPUT DATA

2.1 Main steps for input data preparation

Main steps for the configuration of the method and input files are:

1. Creation of the **abacus** of flood hazard maps for increasing return times (through the hydraulic modelling, e.g., using Telemac-2D).
2. Creation of a input .mat file, named **"info_{domain_name}.mat"** (with the geographical and hydraulic information on the studied domain and the selected control sections), generated by means of the preprocessing Flomart matlab scripts.
3. Creation of a input .json file, named **"info_{domain_name}.json"** (with the information on the selected control sections and their connections, in accordance to the information saved in "info_{domain_name}.mat").
4. Creation of files with **input dynamic data** (observed or simulated discharge or water level at the selected control sections: the results of the hydro-meteorological chain - e.g. the json files generated by Continuum model Flood Proofs for Marche Region).
5. Definition of the **relation discharge vs. return time** to be used in real time (operationally, given a discharge value, simulated or observed, this relation provides the corresponding return time T, hence the name of the map from the abacus closer to it).
6. Configuration of Flomart **main Json configuration file** (which defines all paths of input and output files, selects the methods to be applied for the definition of the final map and defines the time window of the analysis).

In this chapter we will see how to generate/configure the input static data required by Flomart app. The main steps will be illustrated through the case study of the Foglia River (Marche Region, Italy).

2.2 Creation of abacus of hazard maps (raster .tif files)

2.2.1 Creation of hazard maps for certain return times

As already mentioned, FLOMART method is retrospectively initialized with the creation of the abacus of flood hazard maps (e.g., of 500 maps for return times from 0 to 500 years). In particular, the method begins with the creation of a limited number of maps (e.g., 10 maps) with an assigned return time T obtained from detailed 2D hydraulic modeling, for the percentiles, for example, from 80 (return time 5 years) to 99.8 (return time 500 years).

For each watercourse, a few hydraulic modeling are carried out using the two-dimensional hydrodynamic model TELEMAC 2D (modelling De Saint-Venant equations).

Figure 2.1 schematizes the main steps of this hydraulic analysis, which requires:

- **cartographic data** (digital terrain model - DEM, hydrography, etc.), urban data (buildings, infrastructures, culvert, etc.) which defines the static mesh.
- dynamic data (**hydrographs**) which defines the boundary/initial condition of the model.

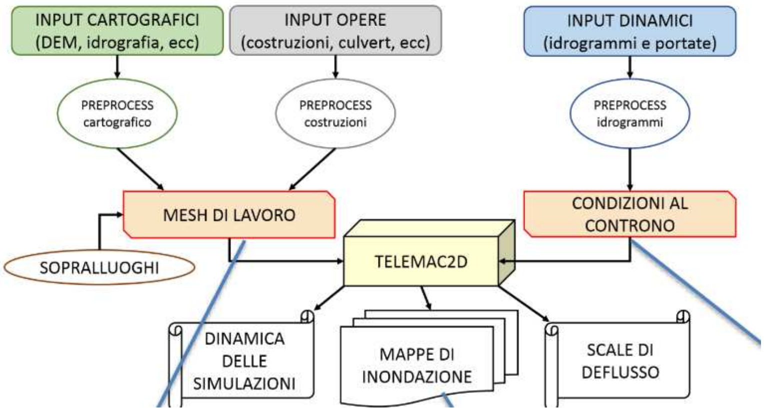


Figure 2.1: Scheme of Telemac2D hydraulic modelling to generate maps of flood hazard with specific return period.

The **final outputs** are the flood maps (rasters .tiff) for the specific hydraulic conditions (return time T). These maps are in meters and georeferenced to the projection system utm of the zone specific to study domain (e.g, Marche region has epsg:32633 - wgs 84 - UTM zone 33N).

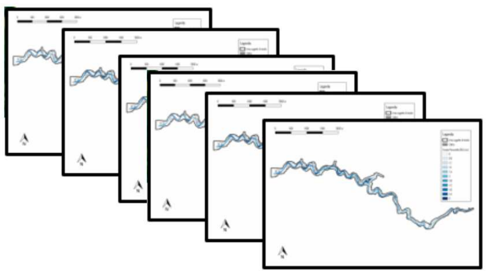


Figure 2.2: Abacus (illustration).

Figure 2.3: Screenshot of the hazard map 500 "Foglia_WD_max_Q500.tif" in Qgis superposed to OpenStreetMap layer (Domain of Foglia River in the Marche Region, Italy).

2.2.2 Creation of all intermediate hazard maps by interpolation

All other maps (for statistical percentiles not calculated with the detailed hydraulic simulations) are then reconstructed with an interpolation procedure with a 1-year step, for each return time T beween, for example, 1 and 500 years.

All maps of the abacus are then located within the Flomart static data folder, "FLOMART/-data_Marche_Foglia/data_static/hazard_data/Foglia/". The path of this folder must then specified in the Flomart json main configuration file (see Chapter4).

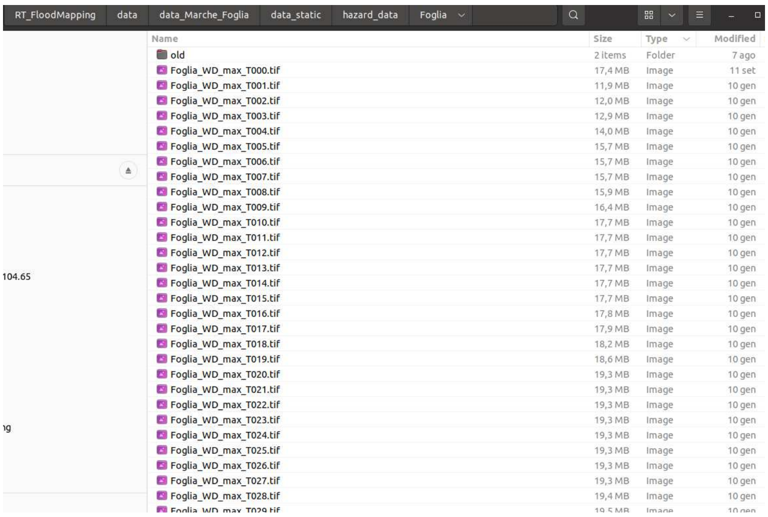


Figure 2.4: Folder with the abacus with all hazard maps for Foglia river, Marche.

2.3 Creation of static file *"info_{Domain_name}.mat"*

Flomart application requires in particular a static file with .mat extension, named *"info_{Domain_name}.mat"*. This file contains the **geographical and hydraulic static data** of the domain and of the sections for which discharge observed or simulated is available. This input file must be located within folder "geo_data" of the static data. It is generated by means of a **MATLAB** (v. 9.10.0.1649659 - R2021a) script called ***"app_flomart_preprocessing.m"*** which is included in the Flomart source codes in homonymous folder *"app_flomart_preprocessing"*. Hereafter the main steps of the script.

2.3.1 Define the domain name

```
clc; clear;
% INPUTS
% please, if a new case study need to be analysed, then insert the name of the new domain
% in the list below. Then execute next command to select the case study from prompt pop-up:
list_domains = {'Foglia', 'Entella', 'Scrivia', 'Chienti'}
domain_name = strjoin(list_domains([listdlg('ListString', ['Choose case study:', ' ', list_domains])]-2))
```

Listing 2.1: Define the domain name (e.g., 'Foglia')

2.3.2 Define input settings and paths

```
if strcmp(domain_name,'Foglia')
    % Path with script matlab (and its modules) for preprocessing in Flomart app:
    path_code = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/flomart-2.0.0_test/app_flomart_preprocessing';
    % path where preparing static files:
    path_preparation_data = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/PREPARATION';
    % Path where the ascii files with gridded geographic information are located:
    %sPathGridGeoData = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/PREPARATION/fp_marche';
    sPathGridGeoData = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/PREPARATION/gridded_marche';
    % Path where to save the file mat output 'info_{domain_name}.mat':
    sPathOutputInfoMat = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/geo_data';
    % Path where hazard maps abacus is located:
    sPathHazardData = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/hazard_data';
    name_hazardmaps = ['Foglia_WD_max_T'];
    type_hazardmaps ='.tif';  % format of the hazard map files
    TR_min=1; % Minimum return time below which the map becomes null
    TR_max=500; % Maximum return time
    modeSelectionSections = 'hardcoded';  %mode of choosing sections ('hardcoded'/'byprompt')
    EPSG_domain = '32633';  % Define the coordinates system EPSG utm.
end
```

Listing 2.2: Define input data and settings.

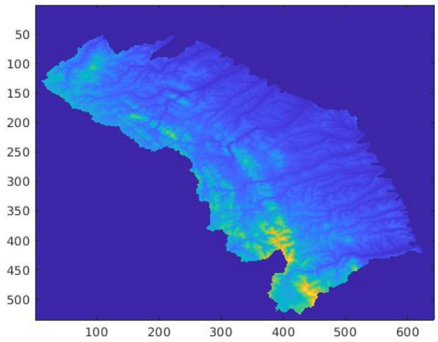
2.3.3 Read geographical gridded information from ascii files:

Then, import the gridded ascii layers to create the required variables *'a2dMap_choice', '2dMap_area', 'a2dMap_cell', 'a2dMap_dem', 'a2dMap_lon', 'a2dMap_lat', 'a2dMap_pnt', 'a2dMapQindice'*.

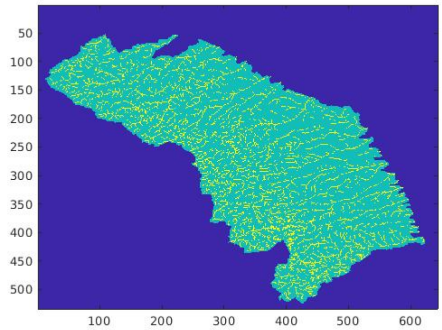
```
if strcmp(domain_name,'Foglia')
    %% names of ascii files with gridded info (choice, dem, ...):
    sFileName_choice = [sPathGridGeoData, '/marche.choice.txt'];
    sFileName_area = [sPathGridGeoData, '/marche.area.txt'];
    sFileName_cell = [sPathGridGeoData, '/marche.areacell.txt'];
    sFileName_dem = [sPathGridGeoData, '/marche.dem.txt'];
    sFileName_lon = [sPathGridGeoData, '/marche.lon.txt'];
    sFileName_lat = [sPathGridGeoData, '/marche.lat.txt'];
    sFileName_pnt = [sPathGridGeoData, '/marche.pnt.txt'];

    %% DEFINING VARIABLES OF FILE .MAT:
    % Choice (where 1 = river, 0 = no river):
    [a2dMap_choice, a2dCoord_choice] = arcgridread(sFileName_choice);
    a2iChoice = a2dMap_choice;
    a2iChoice(isnan(a2iChoice)) = -1;  %replace all NaN with -1
    % Area (drained area):
    [a2dMap_area, a2dCoord_area] = arcgridread(sFileName_area);
    a2dArea = a2dMap_area;
    % Cell (area of cells):
    [a2dMap_cell, a2dCoord_cell] = arcgridread(sFileName_cell);
    a2dCelle = a2dMap_cell;
    a2dCelle(isnan(a2dCelle)) = 1; %replace all NaN with 1
    % Dem (Digital Elevation Model):
    [a2dMap_dem, a2dCoord_dem] = arcgridread(sFileName_dem);
    a2dDem = a2dMap_dem;
    a2dDem(isnan(a2dDem)) = -99;  %replace all NaN with -99
    % Pointers (indicates flow directions):
    [a2dMap_pnt, a2dCoord_pnt] = arcgridread(sFileName_pnt);
    a2iPunt = a2dMap_pnt;
    a2iPunt(isnan(a2iPunt)) = 0;   %replace all NaN with 0
    % Lon (Longitude):
    [a2dMap_lon, a2dCoord_lon] = arcgridread(sFileName_lon);
    Londem = a2dMap_lon;
    Londem(isnan(Londem)) = -99;
    % Lat (Latitude):
    [a2dMap_lat, a2dCoord_lat] = arcgridread(sFileName_lat);
    Latdem = a2dMap_lat;
    Latdem(isnan(Latdem)) = -99;
    % Qindex (reference maximum annual discharge):
    % load a previously computed .mat file with "a2dQindice":
    load([path_preparation_data, '/Qindex_bis_',domain_name,'.mat'])
end
```

Listing 2.3: Upload ascii files with gridded information on the domain.

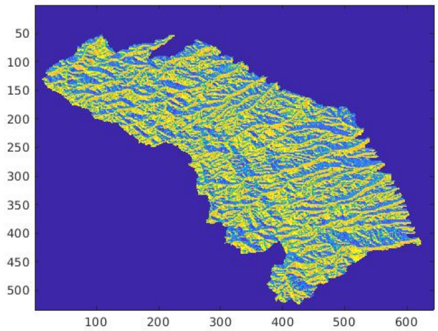
Important: these layers must have the same resolution and no NaN. They usually have larger extent then the one of the hazard maps, thus in the next steps it is described how to clip the domain to the hazard map extent. Moreover, see Annex A for more details on how to generate the gridded variable 'Qindex' (different procedures can be applied, depending on the size of the studied catchment).



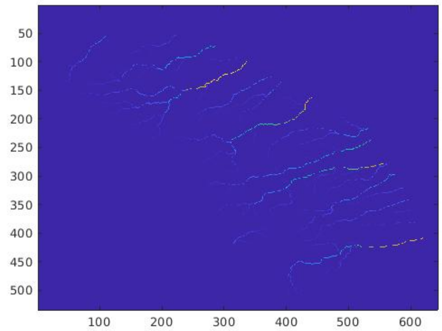
((a)) Dem.



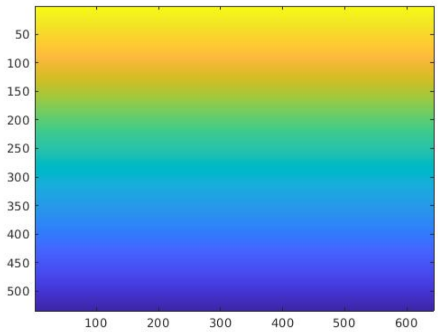
((b)) Choice.



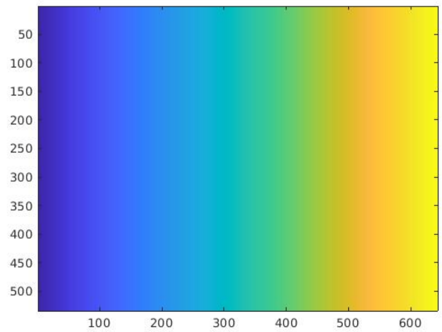
((c)) Pointers.



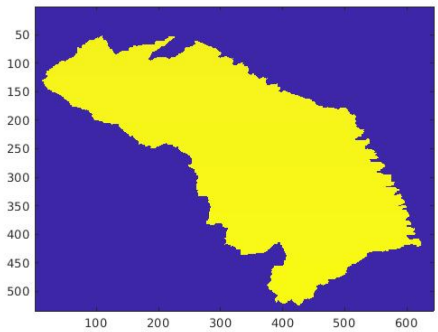
((d)) Area.



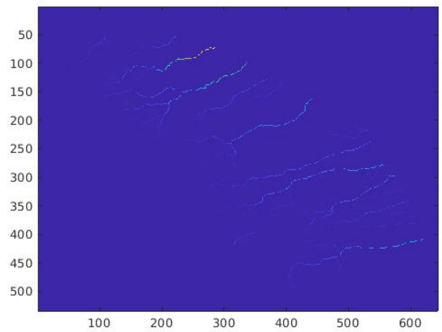
((e)) Latitudes.



((f)) Longitudes.



((g)) Cells area.



((h)) Index discharge.

Figure 2.5: Gridded Layers (from Continuum modelling on Marche dominium) used by Flomart as static data.

2.3.4 Define the new domain grid

Then, define the extent (corners) of the new grid where the studied catchment is located.

```
%% define the new domain grid (corners):
if strcmp(domain_name,'Foglia')
        Lat_min=43.776;
        Lat_max=43.926;
        Lon_min=12.485;
        Lon_max=12.929;
elseif strcmp(domain_name,'Chienti')
        Lat_min=43.0845;
        Lat_max=43.301;
        Lon_min=13.103;
        Lon_max=13.7497;
end

% define corners indexes:
temp=find(((Londem(1,:)-Lon_min)<0)&(Londem(1,:)-Lon_min)>-100);
indice_y_min=temp(end);clear temp
[x,y] =find(((Londem(:,:)-Lon_max)<0)&(Londem(:,:)-Lon_max)>-100);
indice_y_max=y(end);clear temp
temp =find(((Latdem(:,1)-Lat_min)>0)&(Latdem(:,1)-Lat_min)>-100);
indice_x_max=temp(end);clear temp
temp=find(((Latdem(:,1)-Lat_max)>0)&(Latdem(:,1)-Lat_max)>-100);
indice_x_min=temp(end);clear temp
```

Listing 2.4: Save .mat file

2.3.5 Define the new gridded layers according to the new domain grid

The gridded variables ("Drained area, Choice, Pointers, Lat, Lot and Qindex") are then clipped to the new defined grid:

```
% clip maps:
Area_dominio    = a2dArea(indice_x_min:indice_x_max, indice_y_min:indice_y_max);
Choice_dominio  = a2iChoice(indice_x_min:indice_x_max,indice_y_min:indice_y_max);
Punt_dominio    = a2iPunt(indice_x_min:indice_x_max,indice_y_min:indice_y_max);
Lat_dominio     = Latdem(indice_x_min:indice_x_max,indice_y_min:indice_y_max);
Lon_dominio     = Londem(indice_x_min:indice_x_max,indice_y_min:indice_y_max);
Qindice_dominio = a2dQindice(indice_x_min:indice_x_max,indice_y_min:indice_y_max);

% plot to verify:
imagesc(Area_dominio);
caxis([2 30]);
```

Listing 2.5: Save .mat file

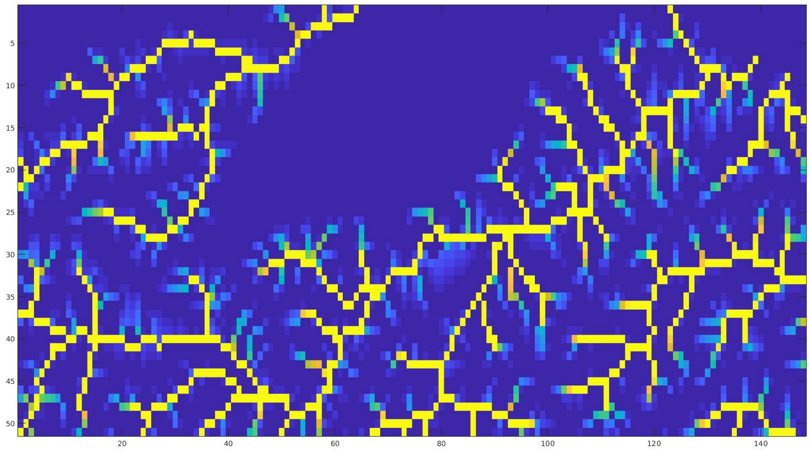


Figure 2.6: Gridded variable 'Area' (Figure 2.5d) for domain Marche clipped to the new domain of Foglia river.

2.3.6 Selection of the control sections

A very important step of the procedure is the definition of the control sections, for which discharge (simulated or observed) or water level is available. You will need to specify, the **number of sections**, their **matrix indexes** (in the original domain), their **names** and corresponding **sub-basins names**. Then, new indexes are computed according to the newly defined grid. For each section the drained area and Qindex are also computed.

Please, notice that the sections names defined at this step will also be used during the configuration of the second input file (*"info_{domain_name}.json"*), see Section 2.4. There must be exact correspondence of the names between the two files! Notice also that basins names should follow the structure '{name of basin}_{name of section}' and section names should follow the structure '{name of section}_{name of basin}'. Moreover, section names must be equal to the section names saved in the output files of the hydrological forecast.

```
if strcmp(domain_name,'Foglia')
        % define the number of control sections:
        quante_sez=6;
        Ymax= length(Latdem(:,1));
        Xmax= length(Latdem(1,:));
        % define indexes of the selected sections [Y X]:
        sezioni_indici_relativi=[106 155;...  % Bronzo
                                 108 190;...  % FlomartTorreCotogna
                                 108 207;...  % FlomartLaBadia
                                 93 228;...   % FlomartCasellaMontecchio
                                 92 241;...   % Montecchio
                                 72 286];     % Foglia3 (Foce)

        % define the names of each selected section:
        nomi_sezioni{1}='Bronzo_Foglia';
        nomi_sezioni{2}='FlomartTorreCotogna_Foglia';
        nomi_sezioni{3}='FlomartLaBadia_Foglia';
        nomi_sezioni{4}='FlomartCasellaMontecchio_Foglia';
        nomi_sezioni{5}='Montecchio_Foglia';
```

```
        nomi_sezioni{6}='Foglia3_Foglia';

        % define name of catchment and section:
        bacino_sezione{1} = 'Foglia_Bronzo';
        bacino_sezione{2} = 'Foglia_FlomartTorreCotogna';
        bacino_sezione{3} = 'Foglia_FlomartLaBadia';
        bacino_sezione{4} = 'Foglia_FlomartCasellaMontecchio';
        bacino_sezione{5} = 'Foglia_Montecchio';
        bacino_sezione{6} = 'Foglia_Foce';

        % verifica aree e calcolo Qindex)
        for i=1:size(sezioni_indici_relativi,1)
            AreaBas(i)=a2dArea(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
            a1dQindex(i)=a2dQindice(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        end

        % Ricavo indici relativi al ritaglio
        sezioni_indici_relativi_corr2(:,1)=sezioni_indici_relativi(:,1)-indice_x_min+1;
        sezioni_indici_relativi_corr2(:,2)=sezioni_indici_relativi(:,2)-indice_y_min+1;
end

% plot to verify sections on clipped grid:
figure
imagesc(Area_dominio)
caxis([2 50])
hold on
for indicew=1:length(sezioni_indici_relativi_corr2)
    plot(sezioni_indici_relativi_corr2(indicew,2), sezioni_indici_relativi_corr2(indicew,1),'or','markersize',13, 'LineWidth',5)
end
```

Listing 2.6: Selection of the control sections: number, indexes and names.

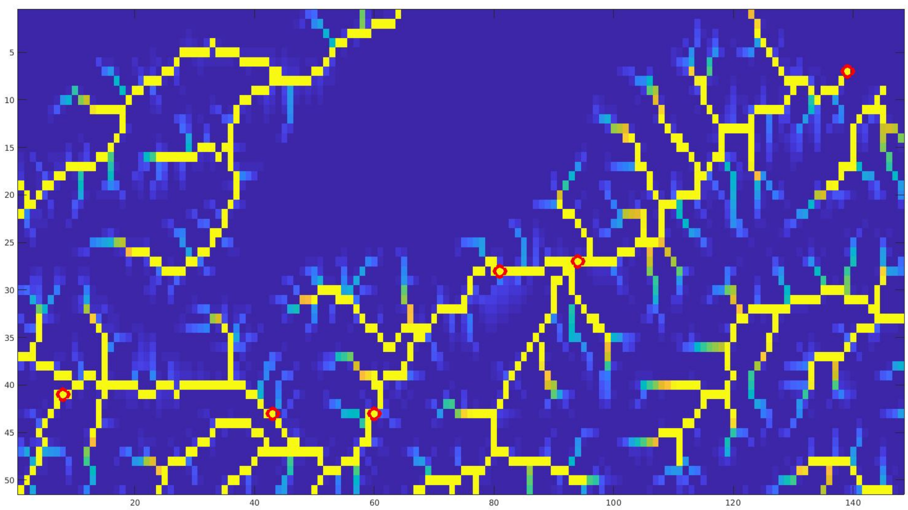


Figure 2.7: The six selected sections (circled in red) for the Foglia river in Marche region.

2.3.6.1 Convert from lat/lon to matrix indexes, if unknown)

If the indexes of the selected sections are unknown, the following script (which makes use of files and functions "Get_MatrixCoordinates_FromPoints_NoNaN.m" and "ReadAsciiRaster.m") computes the transformation of coordinates from Lat/Lon to matrix indexes. A grid layer (e.g., the Choice) is required.

```

% transform lat/lon coordinates into matricial continuum indexes:
tmp = a2dMap_choice;
masknan = nan(size(tmp));
masknan(tmp==1) = 333;
c = readtable([path_preparation_data, '/nuove_sezioni_flomart_Chienti_Foglia.csv']);
x_points = c.Lon;
y_points = c.Lat;
namfascii = sFileName_choice;
[rr, cc] = Get_MatrixCoordinates_FromPoints_NoNaN(x_points, y_points, namfascii, masknan);
% Quindi, salvare le coordinate matrice [rr, cc] nel file 'rigacolonna.txt'.
fid = fopen([path_preparation_data, '/rigacolonna.txt'], 'wt');
for i = 1:length(rr)
    fprintf(fid,'%.0f', rr(i));
    fprintf(fid,'%s', " ");
    fprintf(fid,'%.0f', cc(i));
    fprintf(fid,'\n');
end
fclose(fid);
```

Listing 2.7: Script that converts lat/lon coordinates into matrix indexes.

where function "Get_MatrixCoordinates_FromPoints_NoNaN" is:

```
function [rr_points,cc_points] = Get_MatrixCoordinates_FromPoints_NoNaN(x_points,y_points,
    namfascii,masknan)

[map,nr,nc,xll,yll,dx] = ReadAsciiRaster(namfascii);
xcenter = (xll+dx/2)+(0:nc-1)*dx;
ycenter = (yll+dx/2)+(0:nr-1)*dx;
[xx,yy] = meshgrid(xcenter,ycenter);
yy = flipud(yy);
xx(isnan(masknan)) = NaN;
yy(isnan(masknan)) = NaN;
[kmin,rr_points,cc_points] = deal(nan(length(x_points),1));
for i=1:length(x_points)
    dist2 = (xx-x_points(i)).^2 + (yy-y_points(i)).^2;
    [valmin,kmin(i)] = min(dist2(:));
    [rr_points(i),cc_points(i)] = ind2sub([nr,nc],kmin(i));
end
```

2.3.7 Compute drained area for each selected section

The next step includes the computation of the affected areas (sub-basins) for each section, making use of function "aree_drenate.m". To this aim, the layer with pointers ("Punt_dominio") and the new section indexes ("sezioni_indici_relativi_corr2") are required. Moreover, during this step, the variables of drained area, section names and sub-basin names are sorted according to increasing drained area.

```
% Compute the drainage area for each selected section:
cd(path_code)
aree = aree_drenate(Punt_dominio, sezioni_indici_relativi_corr2);
% Delete superposed areas:
mappa_aree=zeros(size(Punt_dominio));
L=cellfun(@length,aree);
[ordinati,indici_sort]=sort(L);
for i=length(L):-1:1
        valori=unique(mappa_aree(indici_sort(i)));
        mappa_aree(aree{indici_sort(i)})=i;
        nomi_sezioni_sort{i}=nomi_sezioni{indici_sort(i)};
        bacino_sezione_sort{i}=bacino_sezione{indici_sort(i)};
        drainage_area_sort{i}=AreaBas(indici_sort(i));
end
% Verify the areas with a PLOT:
figure
imagesc(mappa_aree)
hold on
[canalix,canaliy]=find(Choice_dominio==1);
for indicew=1:length(canalix)
    plot(canaliy(indicew),canalix(indicew),'.k','markersize',6)
end
for indicew=1:length(sezioni_indici_relativi_corr2)
    plot(sezioni_indici_relativi_corr2(indicew,2),sezioni_indici_relativi_corr2(indicew,1),'or','markersize',6)
end
```

Listing 2.8: Compute drained area for each selected section, defining the sub-basins extents.

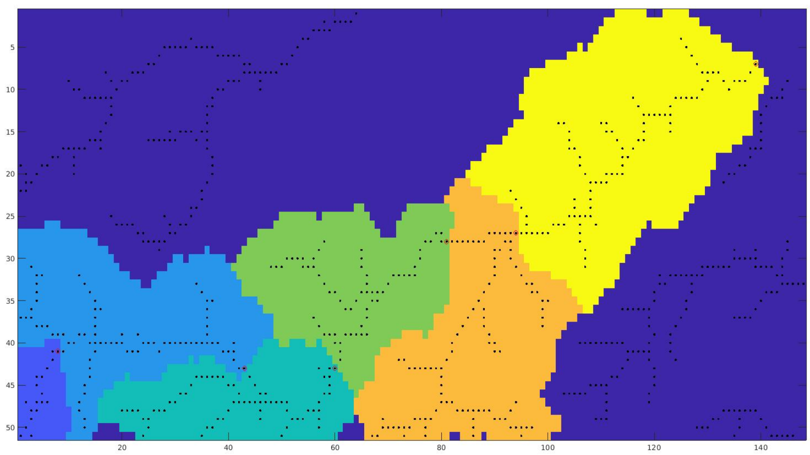


Figure 2.8: Computation of the affected area (sub-basin) of each control section.

2.3.8 Increase or adjust borders of dominium

As shown by Figure 2.8, some borders of these affected areas may need a few adjustments (see for example the orange area at the bottom or the yellow area at the top).

```
[righe,colonne] = size(mappa_aree);
mappa_aree_allargata = mappa_aree;
for indice=1:colonne
        temp=find(mappa_aree(:,indice)>0,1,'first');
        if isempty(temp)
        else
            if temp>1
                mappa_aree_allargata(temp-1,indice)=mappa_aree(temp,indice);
            end
        end
        clear temp
        temp=find(mappa_aree(:,indice)>0,1,'last');
        if isempty(temp)
        else
            if temp<righe
                mappa_aree_allargata(temp+1,indice)=mappa_aree(temp,indice);
            end
        end
        clear temp
end
for indice=1:righe
        temp=find(mappa_aree(indice,:)>0,1,'first');
        if isempty(temp)
        else
            if temp>1
                mappa_aree_allargata(indice,temp-1)=mappa_aree(indice,temp);
            end
        end
        clear temp
        temp=find(mappa_aree(indice,:)>0,1,'last');
        if isempty(temp)
        else
            if temp<colonne
                mappa_aree_allargata(indice,temp+1)=mappa_aree(indice,temp);
            end
        end
        clear temp
end

% verify PLOT:
figure
imagesc(mappa_aree_allargata)
hold on
[canalix,canaliy]=find(Choice_dominio==1);
for indicew=1:length(canalix)
        plot(canaliy(indicew),canalix(indicew),'.k','markersize',6)
end
for indicew=1:length(sezioni_indici_relativi_corr2)
        plot(sezioni_indici_relativi_corr2(indicew,2),sezioni_indici_relativi_corr2(indicew
```

```
            ,1),'or','markersize',6)
end
```

Listing 2.9: Save .mat file

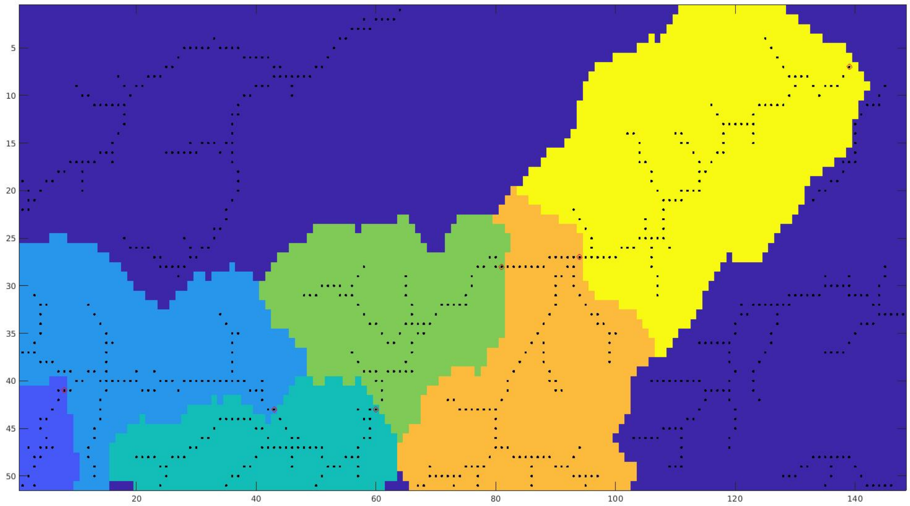


Figure 2.9: Adjustment of borders of the affected areas.

2.3.9 Manually include river delta

Moreover, we may want to slightly extend the affected area of the very downstream section to account for the river mouth. The following lines allow to manually defines a few cells of the grid with the value of the last affected area (yellow area in Figure 5.1).

```
% Manually modify "mappa_aree_allargata" in order to include the river
% delta or other zones known to be flooded that are not included in the domain:
if strcmp(domain_name,'Foglia')
        mappa_aree_allargata(3:10,135:145) = quante_sez;
end

% verify PLOT:
figure
imagesc(mappa_aree_allargata)
hold on
[canalix,canaliy]=find(Choice_dominio==1);
for indicew=1:length(canalix)
        plot(canaliy(indicew),canalix(indicew),'.k','markersize',6)
end
for indicew=1:length(sezioni_indici_relativi_corr2)
        plot(sezioni_indici_relativi_corr2(indicew,2),sezioni_indici_relativi_corr2(indicew
            ,1),'or','markersize',6)
end
```

Listing 2.10: Extend affected areas to account for river mouth.

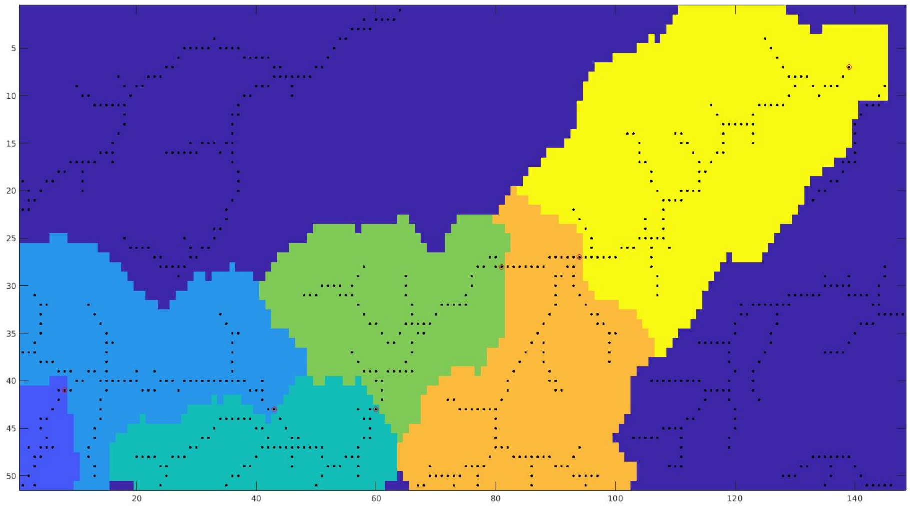


Figure 2.10: Extended affected areas (e.g., to account for river mouth).

2.3.10 Re-grid with the hazard map extent and resolution

Once the sections and the corresponding affected areas are defined, we need to upload one of the flood hazard maps (Figure 2.11) from the abacus, in order to get information on its extent, corners and resolution.

```
% Recover info of metric grid UTM from the hazard maps:
name_file_hazard_read=[sPathHazardData, '/', domain_name, '/', name_hazardmaps, sprintf('
    %03.0f',TR_min), type_hazardmaps];
[A, R]=geotiffread(name_file_hazard_read);
```

Listing 2.11: Save .mat file

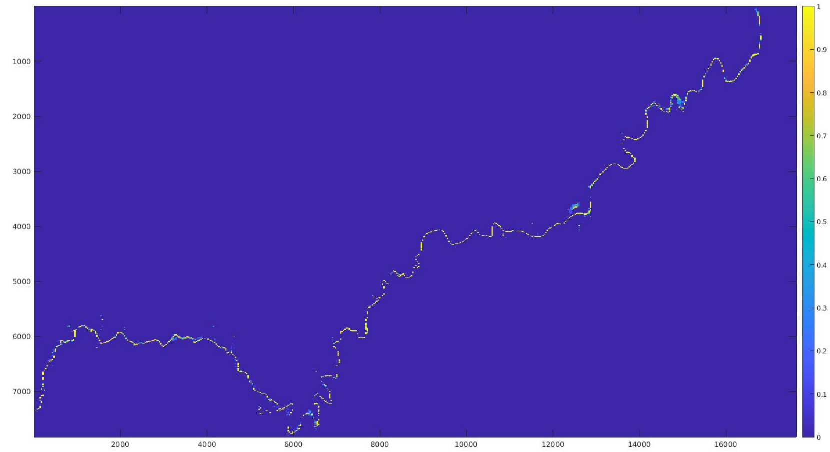


Figure 2.11: Flood hazard map (called A in the script) from the previously produced abacus.

Then, this information is used to re-grid our just-created map of affected areas (variable 'mappa_aree_allargata') to the UTM metric with extent of the hazard maps. The resulted variable is called 'AreeCompetenza'. A check is also computed on the orientation of the axis x and y (using 'flipud()' function in case of reversed axis), to have consistency between the maps.

```

%AGGIUNTO Francesco S.
LonLL=R.XWorldLimits(1);
LonUR=R.XWorldLimits(2);
LatLL=R.YWorldLimits(1);
LatUR=R.YWorldLimits(2);

% convert coord lat/lon to coordinates utm32:
[coord_left, coord_bottom] = latlon2utm(Lat_min, Lon_min);
[coord_right, coord_top]   = latlon2utm(Lat_max, Lon_max);

% get the resolution of the domain.mat in degrees and meters:
% res_Lon = ceil((coord_right - coord_left)/size(mappa_aree_allargata,2));
% res_Lat = ceil((coord_top - coord_bottom)/size(mappa_aree_allargata,1));
res_Lon = ((coord_right - coord_left)/(size(mappa_aree_allargata,2)-1));
res_Lat = ((coord_top - coord_bottom)/(size(mappa_aree_allargata,1)-1));
[Lon_dominio_UTM, Lat_dominio_UTM] = meshgrid(coord_left:res_Lon:coord_right, ...
                                              coord_bottom:res_Lat:coord_top);


% Compute drainage area in km2 for each selected section:
if strcmp(domain_name,'Foglia') | strcmp(domain_name,'Chienti')
    for  i=1:length(drainage_area_sort)
        drainage_area_section_km2{i} = drainage_area_sort{i}/1000000*res_Lon*res_Lat;
    end

elseif strcmp(domain_name,'Entella') | strcmp(domain_name,'EntellaCompleto') | strcmp(domain_
    name,'Scrivia')
    for  i=1:length(drainage_area_sort)
        drainage_area_section_km2{i} = drainage_area_sort{i};   %/1000000*res_Lon*res_Lat;
    end
end


% Interpolate the sample grid data using the griddata() and the 'nearest' method:
% griddata(x, y, v, xq, yq) fits a surface of the form v = f(x,y) to the scattered
% data in the vectors (x,y,v).
% The griddata function interpolates the surface at the query points specified by (xq,yq)
% and returns the interpolated values. The surface always passes through the data points
    defined by x and y.

% check if Lat_dominio_UTM(1,1) < Lat_dominio_UTM(length(Lat_dominio_UTM)), then flip:
if new_y(1,1) > new_y(length(new_y))
    if new_x(1,1) > new_x(length(new_x))
        display('Flipping both y and x is needed! Please, wait ...');
        % flipping both y and x:
        AreeCompetenza = griddata(Lat_dominio_UTM, Lon_dominio_UTM, mappa_aree_allargata, ...
                                  flipud(new_y), flipud(new_x), 'nearest');
    else
```

```
        display('Flipping y is needed! Please, wait ...');
        % flipping y only:
        AreeCompetenza = griddata(Lat_dominio_UTM, Lon_dominio_UTM, mappa_aree_allargata, ...
                                  flipud(new_y), new_x, 'nearest');
    end
else
    if new_x(1,1) > new_x(length(new_x))
        %  flipping x only:
        display('Flipping x is needed! Please, wait ...');
        AreeCompetenza = griddata(Lat_dominio_UTM, Lon_dominio_UTM, mappa_aree_allargata, ...
                                  new_y, flipud(new_x), 'nearest');
    else
        % No flipping needed:
        display('No flipping needed, Please, wait ...');
        AreeCompetenza = griddata(Lon_dominio_UTM, Lat_dominio_UTM, mappa_aree_allargata, ...
                                  new_x, new_y, 'nearest');
    end
end
figure; imagesc(AreeCompetenza);
```

Listing 2.12: Re-grid affected areas variable ('mappa_aree_allargata') to the UTM metric with extent of the hazard maps, into new variable 'AreeCompetenza'. [new$_x$

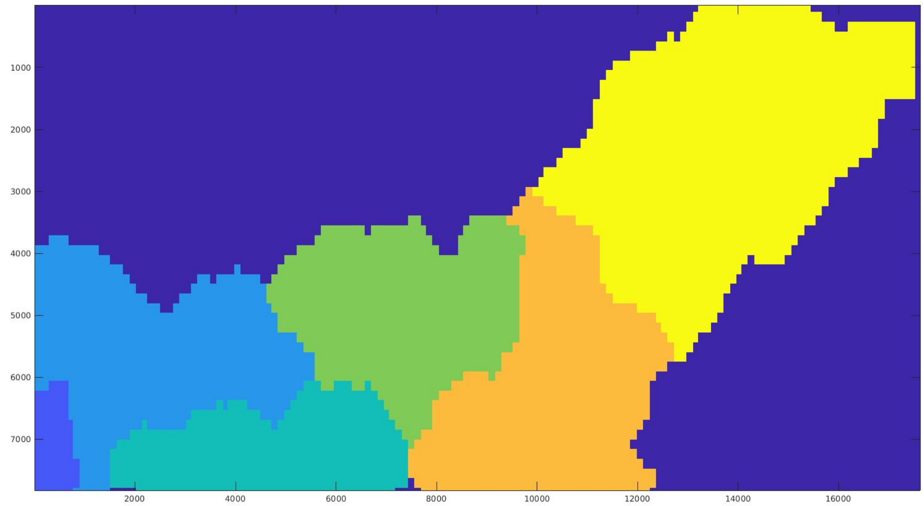


Figure 2.12: Plot of the final variable 'AreeCompetenza' of affected areas of the selected sections, re-gridded to the extent of the hazard maps from abacus.

As an example, Figure 5.2 illustrates the created layer 'AreeCompetenza' for Foglia river (Marche) superposed to a google satellite layer, the Marche 'choice' layer (representing the modelled watercourse), one hazard map from the abacus, and the shapefile witht the selected sections.

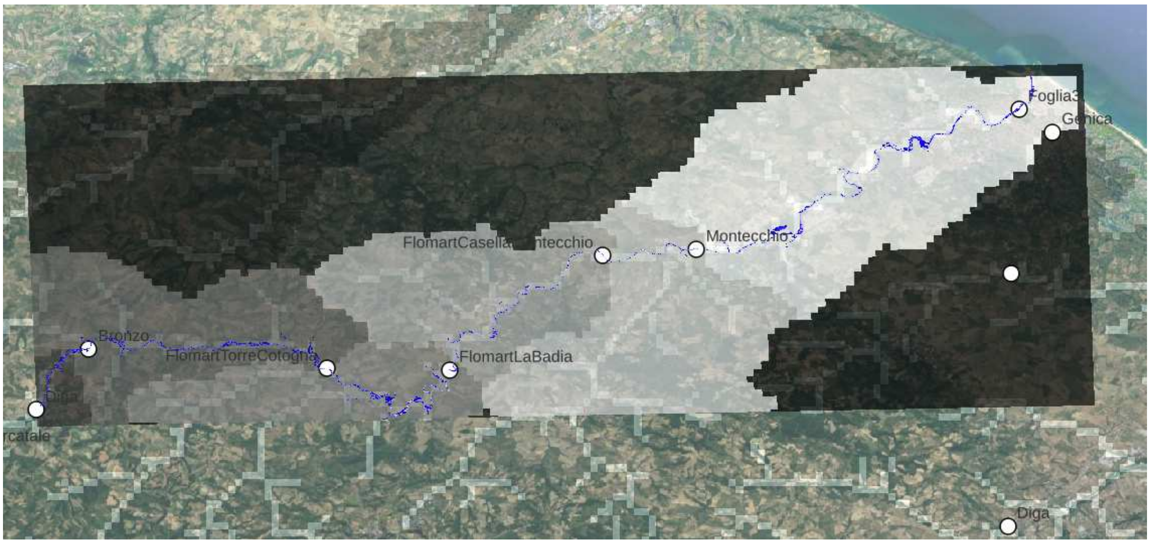


Figure 2.13: Layer 'AreeCompetenza' with the affected areas of the chosen sections for Foglia river (Marche) superposed to a background google satellite layer, the Marche 'choice' layer, one hazard map from the abacus (in blue), and the shapefile with the selected sections (white dots). Screenshot from Qgis.

2.3.11 Save final .mat file with variables required by Flomart

Finally, save all computed variables into a .mat file named 'info_{domain_name}.mat', which is required as static input by Flomart application.

Please, notice that if the size of saved variables exceeds the maximum threshold allowed by matlab, you will need to uncomment the last lines of the following script in order to save the variables into a h5df-structured file using version '-v7.3'.

```
%% please, verify  EPSG_domain!
% save geotiff:
geotiffwrite([path_preparation_data,'/prova_area', domain_name,'.tif'], double(AreeCompetenza),R, 'CoordRefSysCode',['EPSG:',EPSG_domain])


% Save obtained information into mat file (this file is used as static input by Flomart application):
save([sPathOutputInfoMat, '/info_',domain_name,'.mat'], ...
    'LonLL','LonUR','LatLL','LatUR',...
    'AreeCompetenza','mappa_aree_allargata','mappa_aree','Lat_dominio_UTM','Lon_dominio_UTM','a1dQindex','nomi_sezioni_sort', ...
     'indici_sort','bacino_sezione_sort', 'EPSG_domain', 'drainage_area_section_km2');

% % if an error occurs while saving mat file (mainly because of the size of variabile AreeCompetenza then save into h5df format ('-v7.3'):
% save([sPathOutputInfoMat, '/info_',domain_name,'.mat'], ...
%     'LonLL','LonUR','LatLL','LatUR',...
%     'AreeCompetenza','mappa_aree_allargata','mappa_aree','Lat_dominio_UTM','Lon_dominio_UTM','a1dQindex','nomi_sezioni_sort', ...
%      'indici_sort','bacino_sezione_sort', 'EPSG_domain', 'drainage_area_section_km2', '-v7
```

```
    .3');
```

Listing 2.13: Save .mat file

2.4 File *info_{domain_name}.json*

Once the mat file *info_{domain_name}.mat* has been successfully created, another input file *info_{domain_name}.json* is required. This json file is in fact a dictionary containing a few information about the sections (the same sections defined in the mat file), such as their sequence and their connection.

In particular, you will need to specify for each section:

- **"n"**: index of the section, which must correspond to the same value assigned to variable 'AreeCompetenza';
- **"name_point_outlet"**: name of the section (must be equal to the section name defined in the mat file, generally with structure "{section_name}_{domain_name}";
- **"name_point_upstream"**: name of the section located upstream (must be equal to the name of sub-basin defined in the mat file), generally with structure "{domain_name}_{section_name}", if any, otherwise "null";
- **"name_point_downstream"**: name of the section located downstream (must be equal to the name of sub-basin defined in the mat file), generally with structure "{domain_name}_{section_name}", if any, otherwise "null";
- **"name_point_obs"**: name of the hydrometer at the section, if any. For Marche it has the same name as the "name_point_outlet";
- **"description"**: name of the section (must be equal to the name of section affected sub-basin defined in the mat file), generally with structure "{domain_name}_{section_name}".

```
{
{
  "domain_name": "Foglia",
  "domain_sections": {
   "section_1": {
      "n": 1,
      "name_point_outlet": "Bronzo_Foglia",
      "name_point_upstream": null,
      "name_point_downstream": ["Foglia_FlomartTorreCotogna"],
      "name_point_obs": "Bronzo_Foglia",
      "description": "Foglia_Bronzo"
    },
     "section_2": {
      "n": 2,
      "name_point_outlet": "FlomartTorreCotogna_Foglia",
      "name_point_upstream": ["Foglia_Bronzo"],
      "name_point_downstream": ["Foglia_FlomartLaBadia"],
      "name_point_obs": null,
      "description": "Foglia_FlomartTorreCotogna"
    },
     "section_3": {
      "n": 3,
      "name_point_outlet": "FlomartLaBadia_Foglia",
      "name_point_upstream": ["Foglia_FlomartTorreCotogna"],
      "name_point_downstream": ["Foglia_FlomartCasellaMontecchio"],
      "name_point_obs": null,
      "description": "Foglia_FlomartLaBadia"
    },
     "section_4": {
      "n": 4,
      "name_point_outlet": "FlomartCasellaMontecchio_Foglia",
      "name_point_upstream": ["Foglia_FlomartLaBadia"],
      "name_point_downstream": ["Foglia_Montecchio"],
      "name_point_obs": null,
      "description": "Foglia_FlomartCasellaMontecchio"
    },
    "section_5": {
      "n": 5,
      "name_point_outlet": "Montecchio_Foglia",
```

```
      "name_point_upstream": ["Foglia_FlomartCasellaMontecchio"],
      "name_point_downstream": ["Foglia_Foce"],
      "name_point_obs": "Montecchio_Foglia",
      "description": "Foglia_Montecchio"
    },
    "section_6": {
      "n": 6,
      "name_point_outlet": "Foglia3_Foglia",
      "name_point_upstream": ["Foglia_Montecchio"],
      "name_point_downstream": null,
      "name_point_obs": "Foglia3_Foglia",
      "description": "Foglia_Foce"
    }
  }
}
```

2.5 Relation between return time and flood discharge

In real time, the forecasted or observed discharge (current value or the maximum value over the whole forecast window, e.g., 48 h) can be used to select the closest map among all available maps from the abacus of flood scenarios.

As previously mentioned, the abacus maps are characterised and organized by an increasing number of the return time T (e.g., maps *Foglia_WD_max_T001*, *Foglia_WD_max_T002*, ..., *Foglia_WD_max_T500*).

Thus, a relation between the return time (T) and the corresponding discharge (Q_T) is required:

$$T = f(Q_T) \tag{2.1}$$

This relation (which is specific to the case study and eventually to each section) must be implemented within function ***cmp_tr_general****(section_discharge_idx, section_discharge_value, domain_name, section_tr_approx)* in the apposite Flomart module ***lib_utils_tr.py***.

For small basins (<50 km2) this relation may be defined through the regionalization concept, thus using a value of maximum annual discharge of reference (Q_{index}) and a growth factor K_T for different return times T, in common with all sections of the domain. In that case, a unique exponential function for Equation A.1 can be used, except for values close to zero, where a linear relation may be preferred. This was the case of the Liguria domain (e.g., the Entella river). More details are provided in the **Appendix A**.

Instead, for Marche domain, in particular for Foglia river and Chienti river, which are larger catchments, this approach may not be accurate. It is preferred in these cases the use of a specific relation T vs. Q for each section (and affected sub-basin).

In a previous work of CIMA, a hydrological extreme analysis has been performed to Marche region and values of Q_T for increasing values of return times (2,5,10,20,50,100,150,200,500) are available at each cell of the Marche domain. Thus, as shown in the script below, for each analyzed section, a specific data-set (T, Q_T) can be defined (see also Appendix A).
For all intermediate values of discharge a linear interpolation can be performed (function *interp1d()* from *scipy.interpolate*).

Notice that the origin value (0,0) has been added to all data-sets in order to account for all ordinary hydraulic conditions with return time $T \lim 0$.

Finally, function *"cmp_tr_weights()"* is used to get the weights of the computed T with respect to the lower and to the higher available return times.

```

def cmp_tr_general(section_discharge_idx, section_discharge_value, domain_name, section_name,
     section_tr_approx=3):

    # A different procedure is used here for the specific domains.
    if domain_name == 'Foglia':

        # A relation T= f(Q) for each section:
        TT = [0,2,5,10,20,50,100,150,200,500]

        if section_name == 'Foglia_Bronzo':
            QQ = [0,120.1326, 181.3846, 256.1861, 352.8854, 419.2873, 426.1650, 435.9561,
                441.4092, 507.7208]
```

```
        elif section_name == 'Foglia_FlomartTorreCotogna':
            QQ = [0,124.41, 199.022, 277.173, 402.642, 489.577, 530.205, 554.212, 572.15,
                654.73]
        elif section_name == 'Foglia_FlomartLaBadia':
            QQ = [0,148.9, 267.8, 359.6, 497.2, 658.0, 757.2, 784.1, 813.9, 1005.1]
        elif section_name == 'Foglia_FlomartCasellaMontecchio':
            QQ = [0, 156.4, 283.4, 387.6, 532.4, 710.0, 819.4, 841.4, 880.2, 1104.0]
        elif section_name == 'Foglia_Montecchio':
            QQ = [0, 176.4, 328.6, 471.9, 631.7, 834, 1010.6, 1107.2, 1124.2, 1361.1]
        elif section_name == 'Foglia_Foce':
            QQ = [0, 188.5, 361.5, 505.9, 682.7, 904.6, 1080.3, 1231.6, 1270.8, 1516.6]

        log_stream.info(' --------------> Apply linear interpolation to compute tr ...')
        section_scenario_tr_tmp = interp1d(QQ, TT)(section_discharge_value)

        section_scenario_tr_rounded, section_scenario_tr_right, section_scenario_tr_left, \
        section_scenario_weight_right, section_scenario_weight_left = cmp_tr_weights(
            section_scenario_tr_tmp, decimal_tr=section_tr_approx)


    return section_scenario_tr_rounded, section_scenario_tr_right, section_scenario_tr_left,\
           section_scenario_weight_right, section_scenario_weight_left
```

Listing 2.14: Function "cmp_tr_general()" in Flomart module "lib_utils_tr.py", which assigns a return time T to a given value of discharge, according to a previously defined relationship (a different relation for each section).

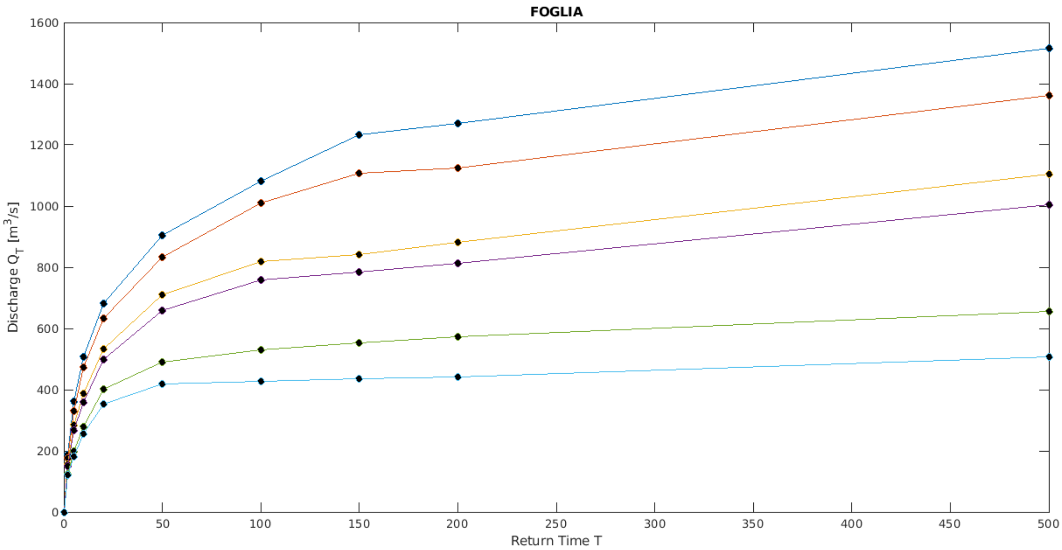


Figure 2.14: Linear interpolation of the relation T vs. Q for Foglia river.

CHAPTER 3

PREPARATION OF DYNAMIC INPUT DATA

3.1 Introduction

Operational use of Flomart requires dynamic inputs such as the **hydrographs** or **water level series** for **each section** of interest. For instance, only the use of hydrographs as input dynamic data have been tested. These hydrographs can be formatted as preferred, however a few modifications may be required for each specific application.

The reader of files is implemented in modules **'driver_type_io.py'** and **'lib_utils_hydrology_generic.py'**. The first module calls other modules which define the specific structure of the input files.

For instance, only readers of asccii, json and mat files (*'lib_utils_hydrology_ascii.py'*, *'lib_utils_hydrology_json.py'* and *'lib_utils_hydrology_mat'*) are implemented. But they could be accordingly modified. In particular in the cases where format of observation files is different from format of simulation files.

However,this guide reports only the type *json* (operationally implemented for Regione marche) for simulated discharges.

3.2 Simulated discharge time series

3.2.1 Files *.json* from Continuum FP hydrological model (©CIMA)

The hydrological model Continuum provides several output files. Among them, the json files with the simulated/observed hydrographs. As an example, the structure of this type of file is illustrated below. This file is basically a dictionary where the first fields describes a few general information on the section (notice that a different file is produced for each section):

- domain name
- the type of the forecast (deterministic or probabilistic)
- the name of the forecast run which created the file
- code, name, location, indexes, drained area, threshold values for alarm and alert of the section.

The second part includes the time period of the run with all instants and the reference time ('time_run') and the time from which the states are initialized ('time_restart').

The last part contains the results of the run (time series of different variables, such as the observed and/or simulated discharge, e.g., "time_series_discharge_simulated").


```
{
    "run_domain": "marche",
    "run_mode": "deterministic",
    "run_name": "weather_stations_history_marche_hazard_map",
    "run_var": "deterministic",
    "section_baseflow": "0.0",
    "section_code": "1126",
    "section_discharge_thr_alarm": "348.75",
    "section_discharge_thr_alert": "197.1",
    "section_domain": "Foglia",
    "section_drained_area": "698.7",
    "section_id": "0",
    "section_idx_i": "286",
    "section_idx_j": "72",
    "section_idx_ji": "72,286",
    "section_name": "Foglia3",
    "section_reference": "Marche",

    "time_period": "2013-11-09␣23:00,2013-11-10␣00:00,2013-11-10␣01:00,2013-11-10␣02:00,␣2013-11-10␣03:00,2013-11-10␣04:00,2013-11-10
          ␣05:00,2013-11-10␣06:00,2013-11-10␣07:00,2013-11-10␣08:00,2013-11-10␣09:00,2013-11-10␣10:00,2013-11-10␣11:00,2013-11-10␣
          12:00,2013-11-10␣13:00,2013-11-10␣14:00,2013-11-10␣15:00,2013-11-10␣16:00,2013-11-10␣17:00,2013-11-10␣18:00,2013-11-10␣
          19:00,2013-11-10␣20:00,2013-11-10␣21:00,2013-11-10␣22:00,2013-11-10␣23:00,2013-11-11␣00:00,2013-11-11␣01:00,2013-11-11␣
          02:00,2013-11-11␣03:00,2013-11-11␣04:00,2013-11-11␣05:00,2013-11-11␣06:00,2013-11-11␣07:00,2013-11-11␣08:00,2013-11-11␣
          09:00,2013-11-11␣10:00,2013-11-11␣11:00,2013-11-11␣12:00,2013-11-11␣13:00,2013-11-11␣14:00,2013-11-11␣15:00,2013-11-11␣
          16:00,2013-11-11␣17:00,2013-11-11␣18:00,2013-11-11␣19:00,2013-11-11␣20:00,2013-11-11␣21:00,2013-11-11␣22:00,2013-11-11␣
          23:00,2013-11-12␣00:00,2013-11-12␣01:00,2013-11-12␣02:00,2013-11-12␣03:00,2013-11-12␣04:00,2013-11-12␣05:00,2013-11-12␣
          06:00,2013-11-12␣07:00,2013-11-12␣08:00,2013-11-12␣09:00,2013-11-12␣10:00,2013-11-12␣11:00,2013-11-12␣12:00,2013-11-12␣
          13:00,2013-11-12␣14:00,2013-11-12␣15:00,2013-11-12␣16:00,2013-11-12␣17:00,2013-11-12␣18:00,2013-11-12␣19:00,2013-11-12␣
          20:00,2013-11-12␣21:00,2013-11-12␣22:00,2013-11-12␣23:00,2013-11-13␣00:00,2013-11-13␣01:00,2013-11-13␣02:00,2013-11-13␣
          03:00,2013-11-13␣04:00,2013-11-13␣05:00,2013-11-13␣06:00,2013-11-13␣07:00,2013-11-13␣08:00,2013-11-13␣09:00,2013-11-13␣
          10:00,2013-11-13␣11:00,2013-11-13␣12:00",

    "time_restart": "201311092300",
    "time_run": "201311130000",

    "time_series_air_temperature_observed":
    "-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0
          ",

    "time_series_discharge_observed":
    "-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
          -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
```

```
        -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
        -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
        -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0,␣
        -9997.0,-9997.0,-9997.0,-9997.0,-9997.0,-9997.0",

    "time_series_discharge_simulated":
    "-9998.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.01,0.04,0.03,0.04,0.46,␣
        1.98,2.71,1.57,0.8,0.68,1.9,6.4,6.96,5.91,23.56,48.32,80.1,127.61,138.74,162.81,178.52,198.25,220.8,266.92,␣
        352.51,516.88,539.37,533.1,512.69,465.17,409.5,357.36,313.92,279.01,248.54,217.78,␣188.98,163.83,140.31,␣
        118.27,99.4,83.85,70.74,60.21,52.11,46.17,41.95,38.8,36.39,34.43,32.45,30.43,28.57,27.05,26.0,25.44,24.97,␣
        24.37,23.95,23.71,23.51,23.32,23.09,22.85,22.62,22.39,22.17",

    "time_series_rain_observed":
    "-9998.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.1,0.0,0.0,0.0,0.0,0.0,2.0,␣1.6,6.8,3.8,3.2,2.7,3.5,3.6,1.6,2.8,3.5,␣
        7.4,9.0,9.5,9.6,8.1,7.6,5.8,5.4,5.7,5.0,4.8,5.9,6.5,5.0,4.2,2.8,3.1,2.0,0.9,1.7,1.5,1.6,1.4,0.3,0.0,0.2,0.5,0.7,0.3,0.1,␣
        0.1,0.2,0.2,0.1,0.1,0.1,0.0,0.0,0.1,0.0,0.0,0.0,0.0,0.0,0.0,0.0,-9998.0,-9998.0,-9998.0,-9998.0,-9998.0,-9998.0,-9998.0,␣
        -9998.0,-9998.0,-9998.0,-9998.0,-9998.0",

    "time_series_soil_moisture_simulated":
    "-9998.0,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,0.3,␣0.3,0.3,0.3,0.4,0.4,0.4,0.4,0.5,0.5,0.5,0.5,␣
        0.6,0.6,0.7,0.8,0.8,0.8,0.9,0.9,0.9,0.9,0.9,0.9,0.9,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,␣
        1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9",

    "time_start": "201311100000"
    }
```

CHAPTER 4

PREPARATION OF MAIN CONFIGURATION JSON FILE

4.1 Configuration Json file for running Flomart

This chapter provides some guidelines for the correct configuration of the run settings for Flomart application, hence the configuration *.json* file, e.g.,

app_flomart_configuration_deterministic_example_simulated_FOGLIA.json

This file includes information on the paths of static, dynamic, ancillary and output (destination) files, as well as the choice of the methods for the creation of the flood maps ("scenario_tiling", "scenario_boundary", "scenario_analysis", "all_period", "scenario_type") and the type of analysis of input dynamic data series.

```

{
  "algorithm": {                                              # In this section insert the options for the algorithm (e.g, method,
                                                              # flag to clean or use existing results, simulation period).

    "general": {                                              # General information on Flomart 2.0.1 application. Nothing to modify.
      "title": "FLOMART",
      "web-site": "",
      "source": "Python_application_developed_by_CIMA_Research_Foundation",
      "history": "2.0.1_[20220701]",
      "project-info": "flomart",
      "algorithm": ""
    },



    "flags": {                                                # Flags to allow cleaning (true) or using (false) previous results.}
      "cleaning_static_data":                  true,          # clean existing static data? e.g., geo info.
      "cleaning_ancillary_data_discharge_obs": true,          # clean existing observed hydrographs stored in ancillary folders?
      "cleaning_ancillary_data_discharge_sim": true,          # clean existing simulated hydrographs stored in ancillary folders?
      "cleaning_ancillary_data_scenario_info": true,          # clean existing info on flood scenario stored in ancillary folders?
      "cleaning_ancillary_data_scenario_file": true,          # clean existing file of flood scenario stored in ancillary folders?
      "cleaning_ancillary_data_scenario_maps": true,          # clean existing flood scenario maps stored in ancillary folders?
      "cleaning_dynamic_plot":                 true,          # clean existing flood scenario plots in results folder?
      "cleaning_dynamic_data":                 true           # clean existing flood scenario data in results folders?
    },




    "ancillary": {
      "__comment__": ...,
      "domain_name" : ["Foglia"],                             # name/s of the studied catchment (list of strings).
      "tr_min" : 0,                                           # minimum return time (years) to be computed (e.g., 0).
```

```
      "tr_max" : 500,                                   # maximum return time to compute (e.g., 500).
      "tr_freq": 1,                                     # return time step to consider between scenarios (e.g., 1).
      "scenario_tiling": "rounded",                     # "rounded" if the return time for the flood scenario must be
                                                        # rounded to the closest available scenario;
                                                        # "weighted" if map is weighted between the two adjacent scenarios.
      "scenario_boundary": "both",                      # "both" if both adjacent maps (upper, lower) are analysed,
                                                        # "max" if only the upper return time map is considered.
      "scenario_analysis": "all_period",                # "max_period" if only the timing and value corresponding to
                                                        # maximum discharge is analyzed (within the define period);
                                                        # "all_period" if you want to analyse (and produce a map) each
                                                        # instant of the run period.
      "scenario_type": "simulated"                      # "observed" if the input dynamic hydrographs are observed,
                                                        # "simulated" if the input are simulated,
                                                        # "mixed" if both observed and simulated have to be used.
    },



    "template": {                                       # In this section you can define the format of the name input/output
                                                         # files (e.g., dates) for the next section.
      "tr": "string_tr",                                # do not change this!
      "section_name": "string_section",                 # do not change this!
      "mask_name": "string_mask",                       # do not change this!
      "domain_name": "string_domain",                   # do not change this!
      "scenario_discharge" : "string_discharge",        # do not change this!
      "source_sub_path_time_discharge_sim": "%Y/%m/%d/%H",   # location of dynamic input files with simulated hydrograph.
      "source_datetime_discharge_sim": "%Y%m%d%H%M",    # date format in the name of dynamic files with simulated hydrograph.
      "source_datetime_to_discharge_sim": "%y%j",       # format of end time in the name of files with simulated hydrograph.
      "source_datetime_from_discharge_sim": "%Y%m%d%H%M",    # format of start time in the name of files with simulated hydrograph.
      "source_sub_path_time_discharge_obs": "%Y%m%d",   # location of dynamic input files with observed hydrograph.
      "source_datetime_discharge_obs": "%Y%m%d%H%M",    # date format in the name of files with observed hydrograph.
      "source_datetime_to_discharge_obs": "%Y%m%d*",    # format of end time in the name of files with observed hydrograph.
      "source_datetime_from_discharge_obs": "%Y%m%d%H%M",    # format of start time in the name of files with observed hydrograph.
      "ancillary_sub_path_time_discharge": "%Y%m%d",    # location of dynamic ancillary files with hydrograph.
      "ancillary_datetime_discharge": "%Y%m%d%H%M",     # date format in the name of dynamic ancillary files with hydrograph.
      "ancillary_sub_path_time_scenario": "%Y/%m/%d/",  # location of dynamic ancillary files with scenario plot.
      "ancillary_datetime_scenario": "%Y%m%d%H%M",      # date format in the name of ancillary file with scenario plot.
      "destination_sub_path_time_scenario_plot": "%Y/%m/%d/", # location of dynamic results files with scenario plot.
      "destination_datetime_scenario_plot": "%Y%m%d%H%M",     # date format in the name of file of dynamic results with scenario plot
          .
      "destination_sub_path_time_scenario_data": "%Y/%m/%d/", # location of dynamic results files with scenario data.
      "destination_datetime_scenario_data": "%Y%m%d%H%M"      # date format in the name of file of dynamic results with scenario data
          .
    }
  },


  "data": {                                             # In this section you are required to insert the path of all static and
                                                         # dynamic files (both input and output).

    "static" : {                                        # Static files:
      "source" : {
        "geo_data": {                                   # Geographic data:
          "generic": {                                  # path of folder with .mat file with geogr. info and drainage areas.
            "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/geo_data/",
            "file_name": "info_{domain_name}.mat"
          },
          "hydraulic": {                                # path of folder with .json file with info on the selected sections.
            "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/geo_data/",
            "file_name": "info_{domain_name}.json"
          }
        },
        "hazard_data": {                                # path of folder with abacus of hazard maps (tiff) for all return times
            .
          "__comment__": "filename:_{domain_name}_hazmap_T{tr}.mat,_{domain_name}_WD_max_T{tr}.mat,_{domain_name}_WD_max_T{tr}.tif",
          "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/hazard_data/{
                domain_name}/",
          "file_name": "{domain_name}_WD_max_T{tr}.tif"
        }
      },
      "destination": {                                  # path of the folder with the static data workspace, created after run.
        "geo_data": {
          "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_static/geo_data/",
          "file_name": "info_{domain_name}.workspace"
        }
      }
    },



    "dynamic": {                                        # Dynamic files:
      "source": {
```

```
      "discharge_data_simulated": {                          # path of files with simulated hydrographs (if "scenario_type": "simulated" or "mixed")
        "_comment" : "file_name_string:_[idro_{source_datetime_from_discharge_sim}_{source_datetime_to_discharge_sim}{mask_discharge}.json]",
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/source/discharge_run_obs_ws/{source_sub_path_time_discharge_sim}/collections/",
        "file_name": "hydrograph_{section_name}_{domain_name}_{source_datetime_from_discharge_sim}.json",
        "type": "json",                                      # define types of file ("json", "matlab", "ascii")
        "variables": {"time":  "time_period",
                      "discharge":  "time_series_discharge_simulated",
                      "water_level": null},                  # define the variable in the input hydrograph files corresponding to
                                                             # time, discharge and water level (if any).
        "method_data_occurrence": "all",                     # "all" if all data in the series must be analyzed.
        "method_data_analysis" : null,                       # "null" if no analysis is performed on input data,
                                                             # "max" if only the maximum values when multiple available series.
        "method_data_filling": null,                         # "null" if no data is filled where missing,
                                                             # "interpolate" if missing data are replaced with interpolated data.
        "method_data_null": null,                            # "null" if no link is performed between sections data
                                                             # "links" if links are computed between the data of adjacent sections.
        "time_period": 48,                                   # time period of the simulation (e.g. 48 instants).
        "time_frequency": "H",                               # time step of simulation (e.g., "H" for hourly step)
        "time_rounding": "D"                                 # rounding precision of the run time (e.g., "D" for daily)
      },



      "discharge_data_observed": {                           # path of files with observed hydrographs (if "scenario_type": "simulated" or "mixed")
        "__comment__": "mat_file:_{section_name}_{source_datetime_from_discharge_obs}_{source_datetime_to_discharge_obs}_1H.mat",
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/source/discharge_observed/{source_sub_path_time_discharge_obs}/",
        "file_name": "{section_name}_{source_datetime_from_discharge_obs}_{source_datetime_to_discharge_obs}_1H.mat",
        "type": "matlab",                                    # define types of file ("json", "matlab", "ascii")
        "variables": {"time":  "a1sDateVet",
                      "discharge":  "a1dQOss",
                      "water_level": "a1dLivOssMean" },      # define the variable in the input hydrograph files corresponding to
                                                             # time, discharge and water level (if any) data.
        "method_data_occurrence": "all",                     # "all" if all data in the series must be analyzed.
        "method_data_analysis" : null,                       # "null" if no analysis is performed on input data,
                                                             # "max" if only the maximum values when multiple available series.
        "method_data_filling": "interpolate",                # "null" if no data is filled where missing,
                                                             # "interpolate" if missing data are replaced with interpolated data.
        "method_data_null": "links",                         # "null" if no link is performed between sections data
                                                             # "links" if links are computed between the data of adjacent sections.
        "time_period": 48,                                   # time period of the simulation (e.g. 48 instants).
        "time_frequency": "H",                               # time step of simulation (e.g., "H" for hourly step)
        "time_rounding": "D"                                 # rounding precision of the run time (e.g., "D" for daily)
      },
    },



    "ancillary": {                                           # In this section define the paths of files/folders that will contain
                                                             # the ancillary (transitory) results:
      "discharge_data_simulated": {
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/ancillary/discharge_run_obs_ws/{ancillary_sub_path_time_discharge}/",
        "file_name": "discharge_{domain_name}_{ancillary_datetime_discharge}_simulated.workspace"
      },
      "discharge_data_observed": {
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/ancillary/discharge_run_obs_ws/{ancillary_sub_path_time_discharge}/",
        "file_name": "discharge_{domain_name}_{ancillary_datetime_discharge}_observed.workspace"
      },
      "scenario_data_info": {
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/ancillary/scenario_run_obs_ws/{ancillary_sub_path_time_scenario}",
        "file_name": "scenario_{domain_name}_{ancillary_datetime_scenario}_info.workspace"
      },
      "scenario_data_file": {
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/ancillary/scenario_run_obs_ws/{ancillary_sub_path_time_scenario}",
        "file_name": "scenario_{domain_name}_{ancillary_datetime_scenario}_file.workspace"
      },
      "scenario_data_map": {
        "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/ancillary/scenario_run_obs_ws/{ancillary_sub_path_time_scenario}",
        "file_name": "scenario_{domain_name}_{ancillary_datetime_scenario}_map.tiff"
      }
    },


```

```
      "destination" : {                                       # In this section define the paths of files/folders that will include
                                                              # all final results (maps, info and data):
        "plot": {
          "scenario_tiff": {                                  # path of final maps as .tiff.
            "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/_data_dynamic/outcome/
                 scenario_run_obs_ws/{destination_sub_path_time_scenario_plot}/plot/",
            "file_name": "scenario_{domain_name}_{destination_datetime_scenario_plot}_graph.tiff",
            "save_status": true                               # activate this flag if you want to save results as .tiff.
          },
          "scenario_png": {                                   # path of final maps as .png.
            "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/_data_dynamic/outcome/
                 scenario_run_obs_ws/{destination_sub_path_time_scenario_plot}/plot/",
            "file_name": "scenario_{domain_name}_{destination_datetime_scenario_plot}_graph.png",
            "save_status": true                               # activate this flag to plot maps as .png (it may require some time!).
          },
          "scenario_info": {                                  # path of final maps information as .json.
            "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/_outcome/
                 scenario_run_obs_ws/{destination_sub_path_time_scenario_plot}/plot/",
            "file_name": "scenario_{domain_name}_{destination_datetime_scenario_plot}_graph.json",
            "save_status": true                               # activate this flag to save info on the final scenarios as json.
          }
        },

        "data": {
          "scenario_time_series": {                           # path of flood scenario timeseries as .json.
            "folder_name":
            "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/data_dynamic/_outcome/scenario_run_obs_ws/{
                 destination_sub_path_time_scenario_data}/time_series/",
            "file_name": "scenario_{domain_name}_{destination_datetime_scenario_data}_ts.json",
            "save_status": true                               # activate this flag to save the flood scenario timeseries as json.
          }
        }
      }
    }
  },



  "log": {                                                    # path and name of the log file.
    "folder_name": "/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Foglia/log/",
    "file_name": "app_flomart_logging_matlab_observed.txt"
  },



  "time": {                                                   # operational time options (starting time, time step and time period).
    "time_now": null,
    "time_period": 1,
    "time_frequency": "H",
    "time_rounding": "H"
  }
}
```

Then, Flomart can be executed with the following command (python 3 libraries are required!):

python3 app_flomart_main.py -settings_algorithm app_flomart_configuration_deterministic_example_simulated_FOGLIA.json -time "2022-09-01 12:00"

CHAPTER 5

RESULTS

5.1 Introduction

This chapter will present a few results yielded by Flomart operationally applied to the Foglia and the Chienti rivers, in Marche region. The event of **November 2013** (known to have effectively caused floodings along both studied rivers) has been chosen to illustrate the functioning of the tool.

5.2 Foglia River in Marche Region, Italy

5.2.1 Input static data

Here are summarized the input static data (reference sections and corresponding affected areas).

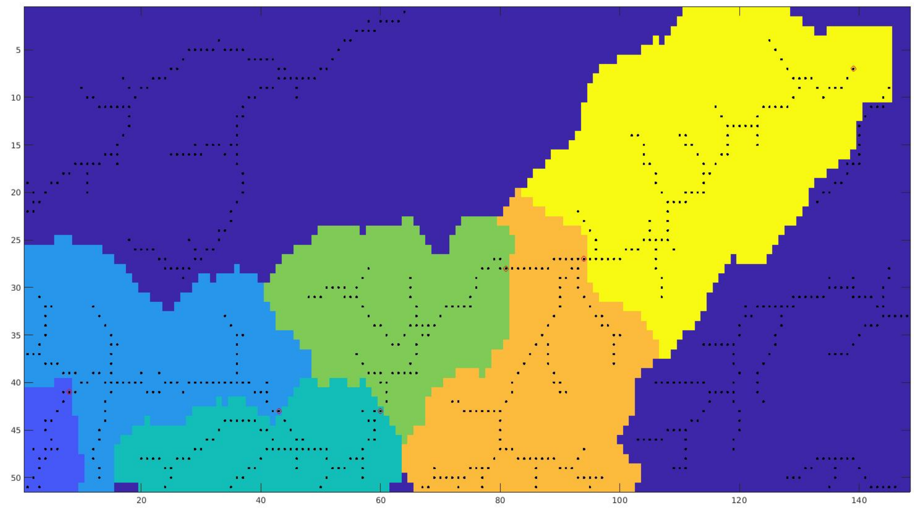


Figure 5.1: Map of the affected areas to each selected section for Foglia river, Marche region. Variable saved in the .mat input file.

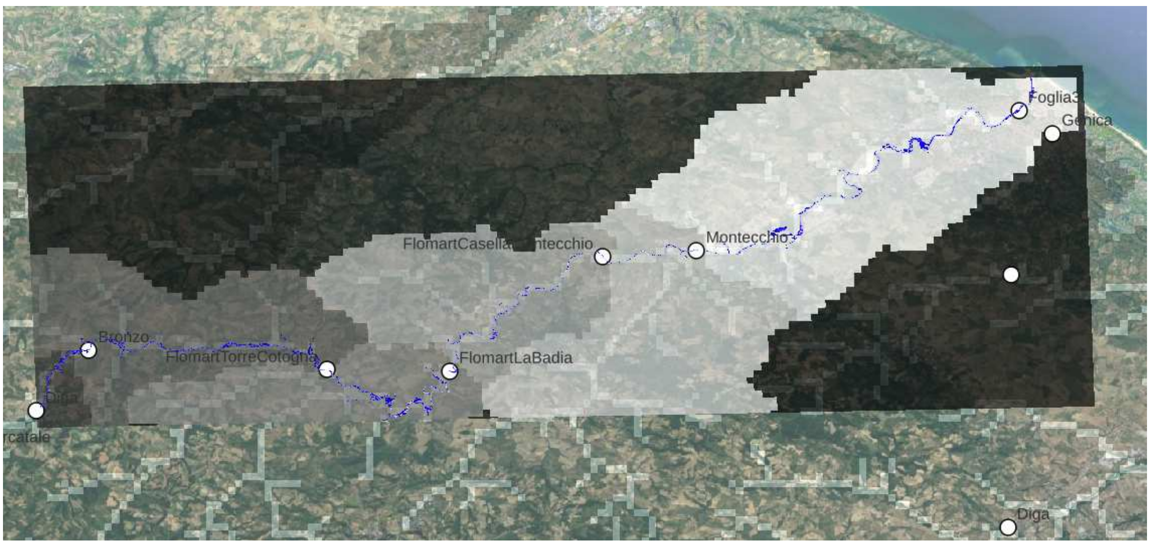


Figure 5.2: Layer 'AreeCompetenza' with the affected areas of the chosen sections for Foglia river (Marche region) superposed to a background google satellite layer, the Marche 'choice' layer, one hazard map from the abacus (in blue), and the shapefile with the selected sections (white dots). Screenshot from Qgis.

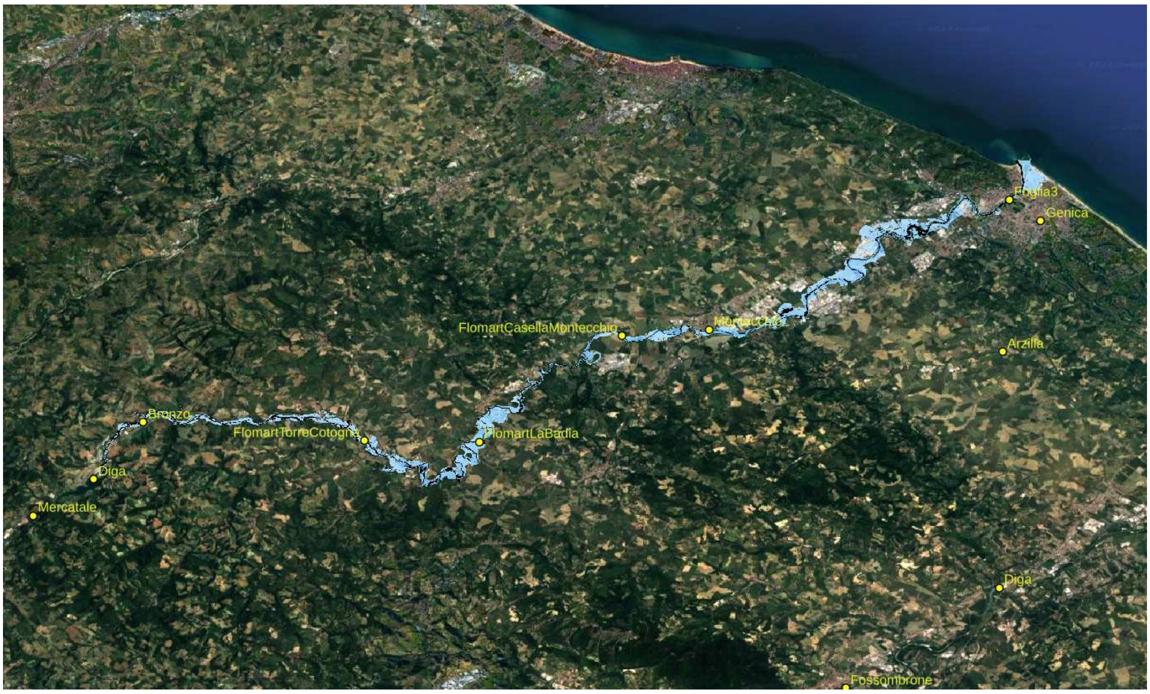
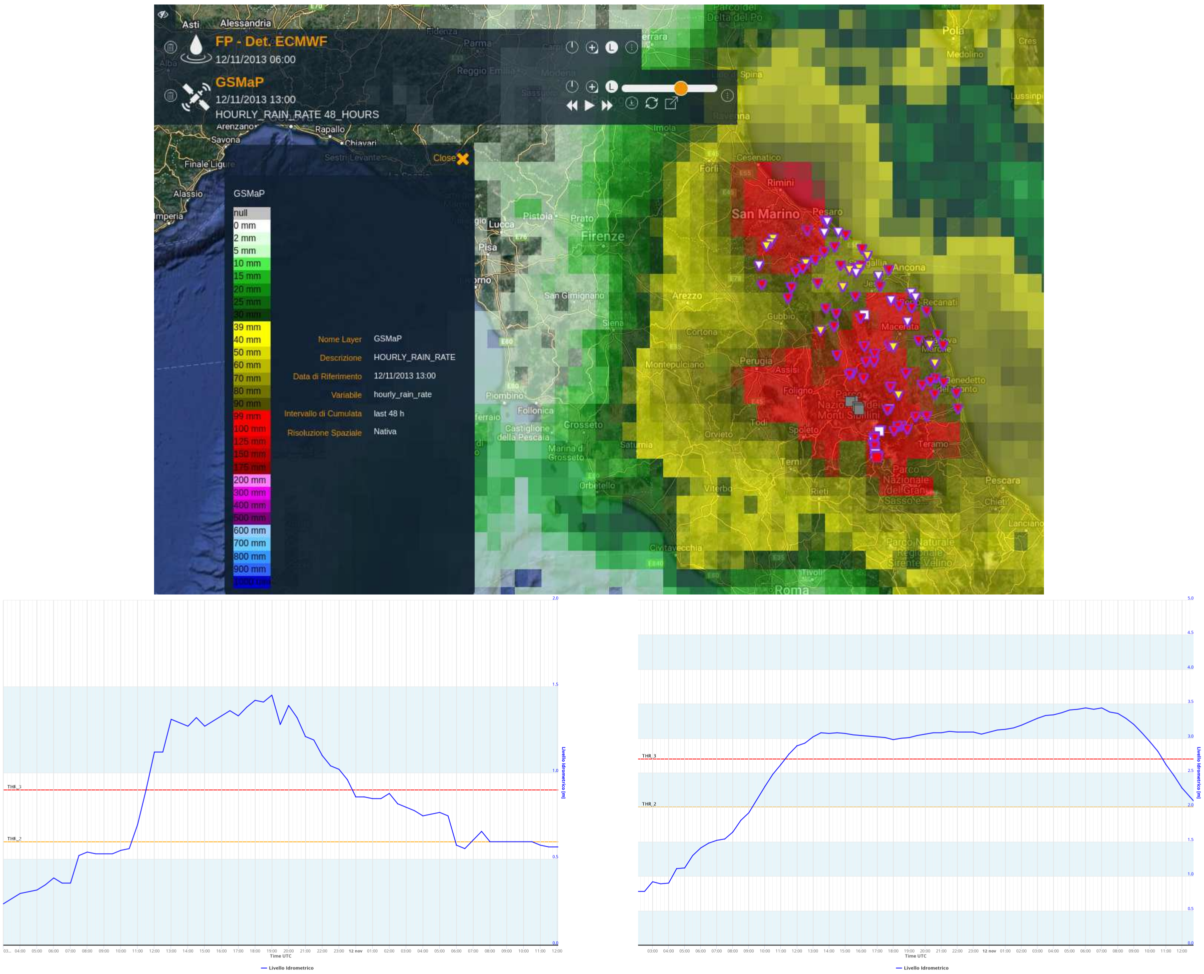


Figure 5.3: Comparison (on Qgis) between the hazard maps from the abacus of Foglia river with the minimum (1 year, dark blue) and the maximum (500 years, light blue) return period, respectively.

5.2.2 Events: input simulated hydrographs

The event of november 11/11/2013 has been chosen. This event is known to have impacted Marche region with several floodings across the region.



((a)) Water level at section Bronzo, Foglia river.

((b)) Water level at section Foglia3 (city of Pesaro), Foglia river.

Figure 5.4: Event of November 2013. Accumulated precipitation and water levels recorded at two stations along Foglia river (taken from MyDewetra platform).

5.2.2.1 Run of 10/11/2013

Figure 5.5 illustrates the simulated hydrographs (taken from MyDewetra platform) at three different sections located from the upstream sector of Foglia river to the river mouth at Pesaro. Hydrographs have been simulated by means of an historical run of FP continuum on the Marche domain with an observed period of three months as initial conditions.

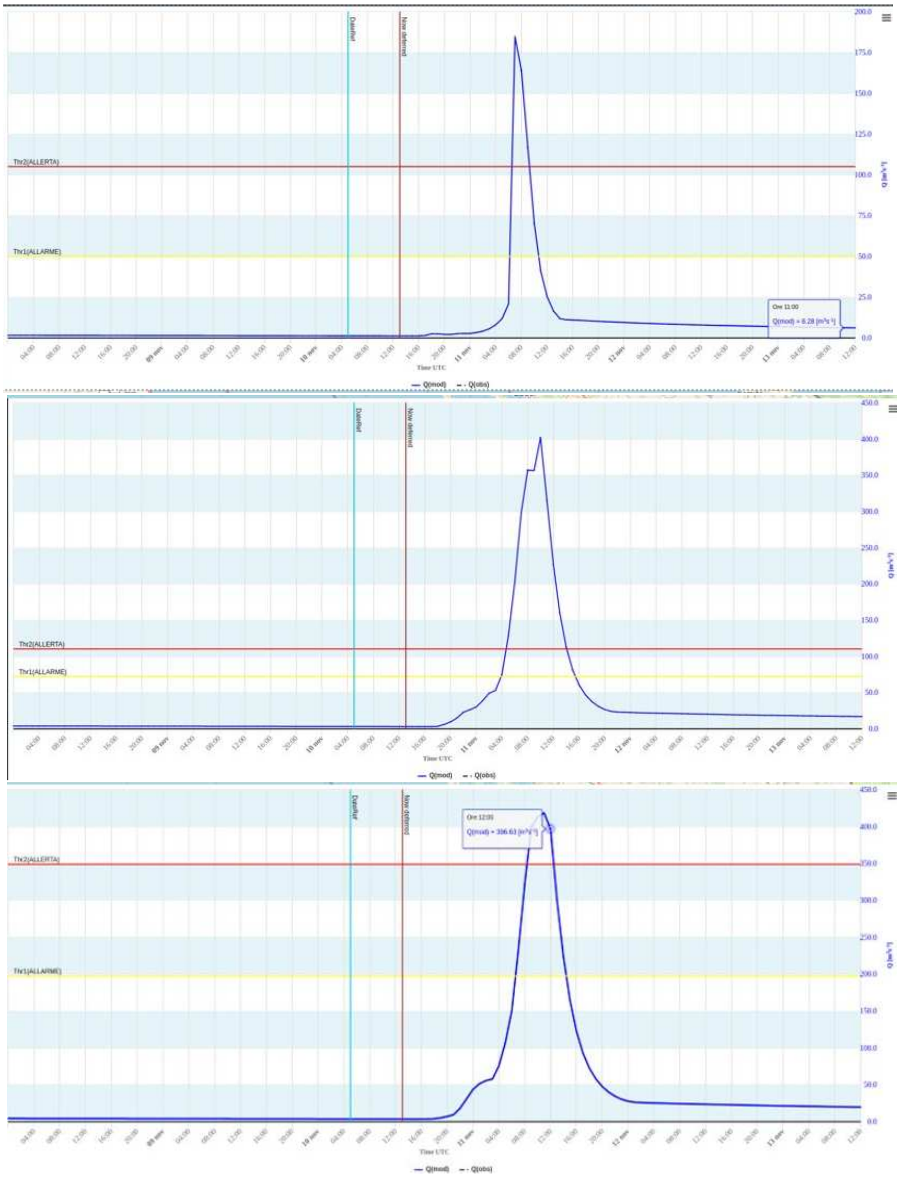
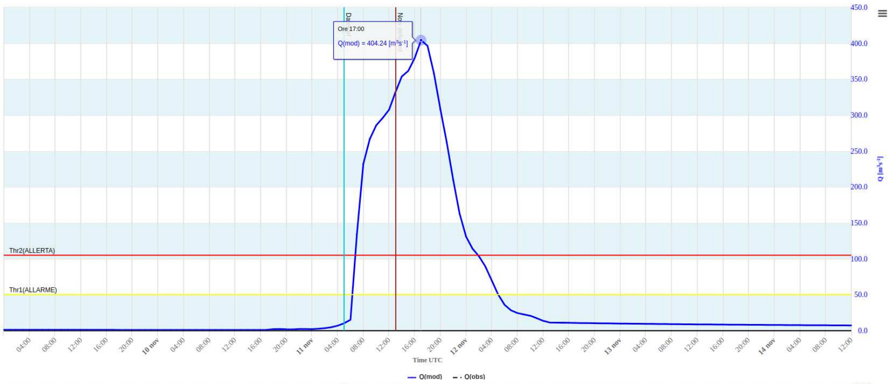
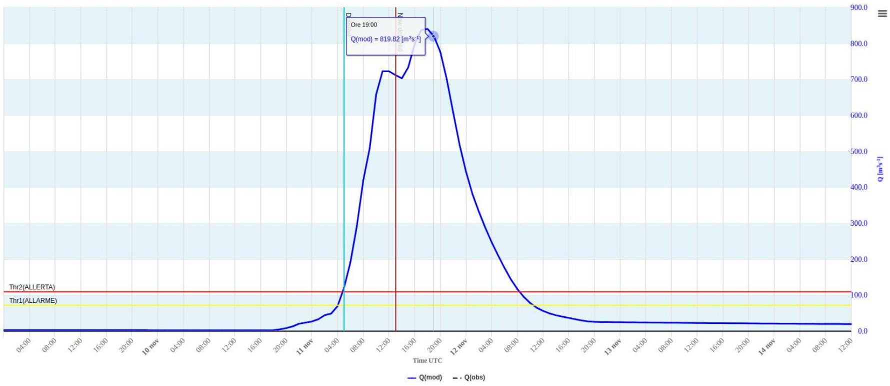


Figure 5.5: Hydrograph for run 2013/11/10 05:00 utc for Foglia river at sections (a) 'bronzo' (located upstream), (b) 'montecchio' (located a bit downstream), (c) 'Foglia3' (located at the river mouth, city of Pesaro).

5.2.2.2 Run of 11/11/2013

Figure 5.6 illustrates the simulated hydrographs (taken from MyDewetra platform) at three different sections located from the upstream sector of Foglia river to the river mouth at Pesaro.





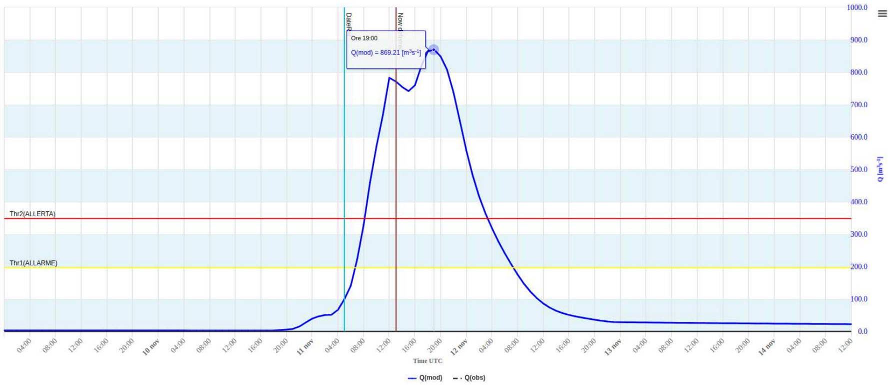


Figure 5.6: Hydrograph for run 2013/11/11 05:00 utc for Foglia river at sections (a) 'bronzo' (located upstream), (b) 'montecchio' (located a bit downstream), (c) 'Foglia3' (located at the river mouth, city of Pesaro).

5.2.3 Output data

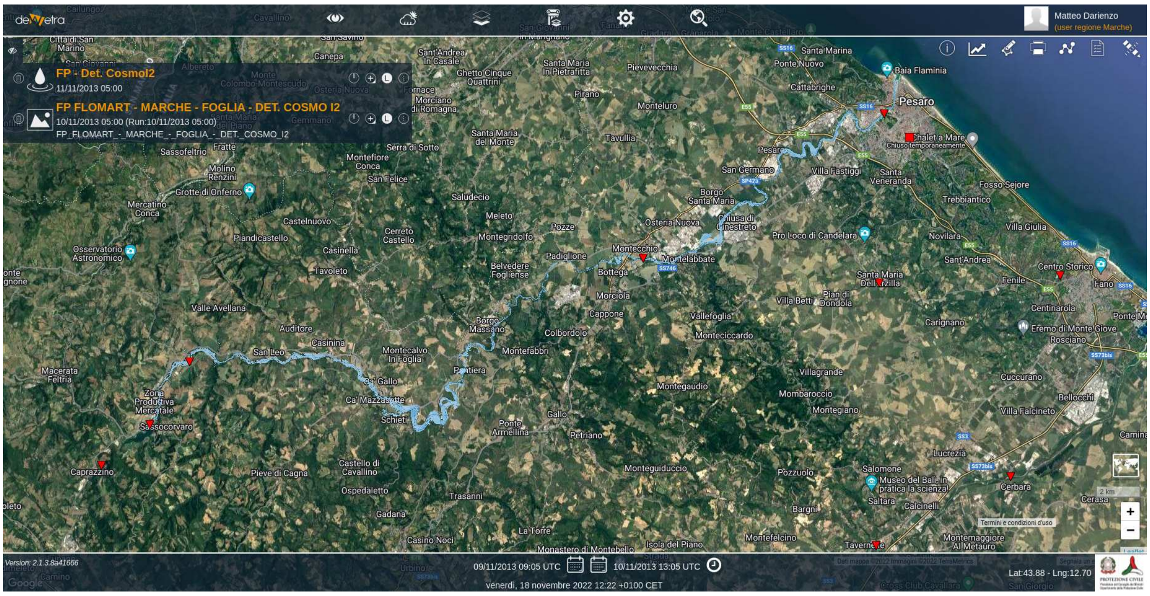


Figure 5.7: Flood map of Foglia river (Marche region) produced by Flomart and visualized in MyDewetra platform for the simulated event of 10/11/2013.


```
{
    "Foglia_Bronzo": {
        "discharge_value": "184.54",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "5",
        "scenario_idx_left": "6",
        "scenario_idx_right": "5",
        "scenario_n": "1",
        "scenario_weight_left": "0.21",
        "scenario_weight_right": "0.79",
        "time": "2013-11-11_07:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_FlomartCasellaMontecchio": {
        "discharge_value": "331.13",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "7",
        "scenario_idx_left": "8",
        "scenario_idx_right": "7",
        "scenario_n": "1",
        "scenario_weight_left": "0.29",
        "scenario_weight_right": "0.71",
        "time": "2013-11-11_10:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_FlomartLaBadia": {
        "discharge_value": "330.56",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
```

```
        "scenario_idx": "8",
        "scenario_idx_left": "9",
        "scenario_idx_right": "8",
        "scenario_n": "1",
        "scenario_weight_left": "0.42",
        "scenario_weight_right": "0.58",
        "time": "2013-11-11_09:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_FlomartTorreCotogna": {
        "discharge_value": "255.01",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "9",
        "scenario_idx_left": "9",
        "scenario_idx_right": "8",
        "scenario_n": "1",
        "scenario_weight_left": "0.58",
        "scenario_weight_right": "0.42",
        "time": "2013-11-11_08:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_Foce": {
        "discharge_value": "418.17",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "7",
        "scenario_idx_left": "7",
        "scenario_idx_right": "6",
        "scenario_n": "1",
        "scenario_weight_left": "0.96",
        "scenario_weight_right": "0.04",
        "time": "2013-11-11_11:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_Montecchio": {
        "discharge_value": "401.78",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "8",
        "scenario_idx_left": "8",
        "scenario_idx_right": "7",
        "scenario_n": "1",
        "scenario_weight_left": "0.55",
        "scenario_weight_right": "0.45",
        "time": "2013-11-11_10:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "scenario_epsg_code": "EPSG:32633",
    "scenario_name": "Foglia",
    "scenario_time_now": "2013-11-10_05:00",
    "scenario_time_step": "2013-11-10_05:00"
}
```

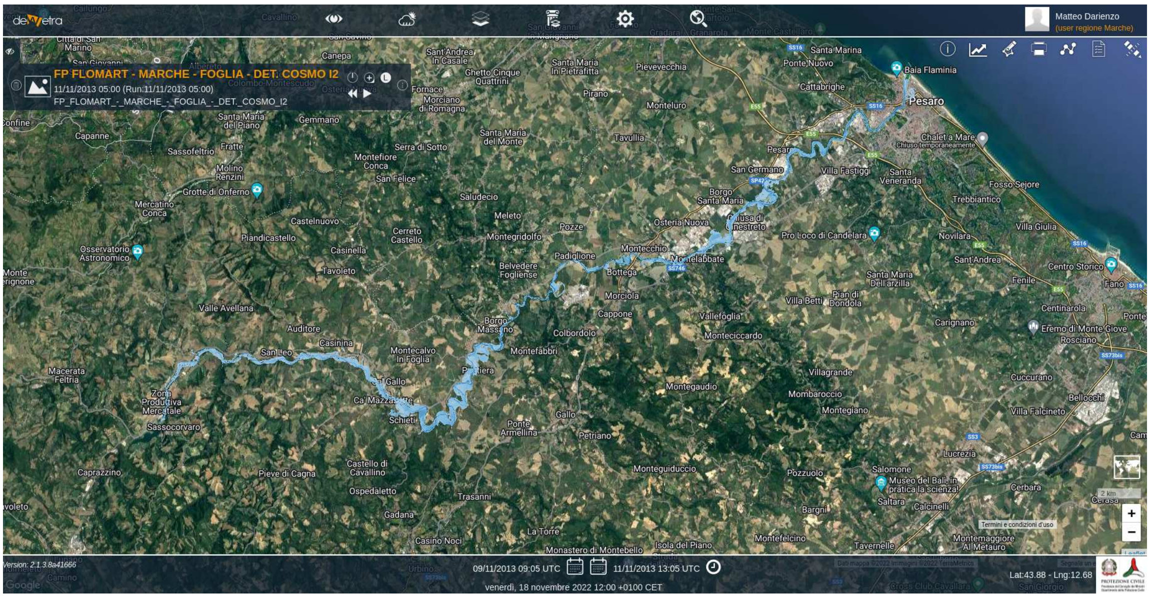


Figure 5.8: Zoom on the Foglia river mouth (city of Pesaro, Marche) for the simulated event of 11/11/2013.


```
{
    "Foglia_Bronzo": {
        "discharge_value": "404.24",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "43",
        "scenario_idx_left": "44",
        "scenario_idx_right": "43",
        "scenario_n": "1",
        "scenario_weight_left": "0.2",
        "scenario_weight_right": "0.8",
        "time": "2013-11-11_17:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_FlomartCasellaMontecchio": {
        "discharge_value": "724.07",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "56",
        "scenario_idx_left": "57",
        "scenario_idx_right": "56",
        "scenario_n": "1",
        "scenario_weight_left": "0.43",
        "scenario_weight_right": "0.57",
        "time": "2013-11-11_18:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_FlomartLaBadia": {
        "discharge_value": "681.07",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "62",
        "scenario_idx_left": "62",
        "scenario_idx_right": "61",
        "scenario_n": "1",
        "scenario_weight_left": "0.63",
        "scenario_weight_right": "0.37",
        "time": "2013-11-11_18:00",
```

```
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_FlomartTorreCotogna": {
        "discharge_value": "494.05",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "56",
        "scenario_idx_left": "56",
        "scenario_idx_right": "55",
        "scenario_n": "1",
        "scenario_weight_left": "0.5",
        "scenario_weight_right": "0.5",
        "time": "2013-11-11_18:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_Foce": {
        "discharge_value": "869.21",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "45",
        "scenario_idx_left": "46",
        "scenario_idx_right": "45",
        "scenario_n": "1",
        "scenario_weight_left": "0.22",
        "scenario_weight_right": "0.78",
        "time": "2013-11-11_19:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Foglia_Montecchio": {
        "discharge_value": "839.99",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "52",
        "scenario_idx_left": "52",
        "scenario_idx_right": "51",
        "scenario_n": "1",
        "scenario_weight_left": "0.7",
        "scenario_weight_right": "0.3",
        "time": "2013-11-11_18:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "scenario_epsg_code": "EPSG:32633",
    "scenario_name": "Foglia",
    "scenario_time_now": "2013-11-11_05:00",
    "scenario_time_step": "2013-11-11_05:00"
}
```

5.3 Chienti River in Marche Region, Italy

5.3.1 Input data

The hydraulic modelling for the Chienti Basin was performed to 3 subdomains, called: Polverina-Caccamo, Caccamo-LeGrazie, LeGrazie-Foce. For each scenario three separated flood maps were available for Chienti river. Then, the maps have been merged and compressed to create one unique abacus.

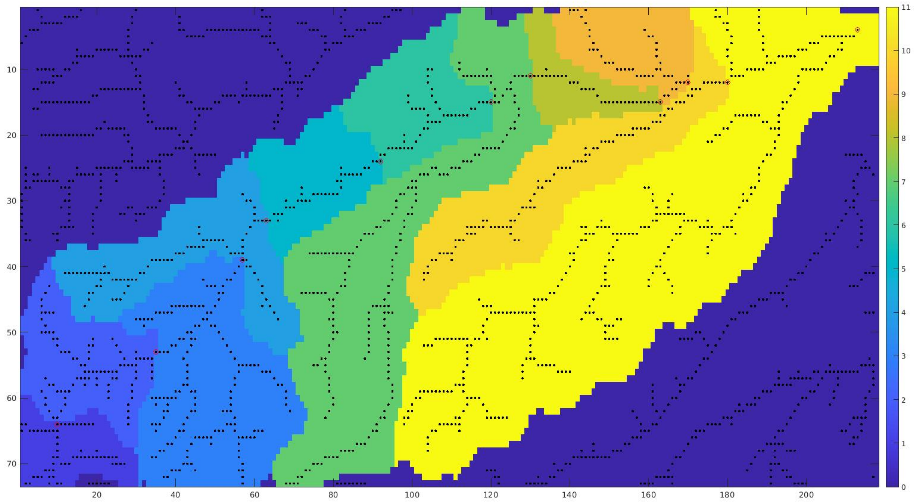


Figure 5.9: Map of the affected areas to each selected section for Chienti river, Marche region. Variable saved in the .mat input file.

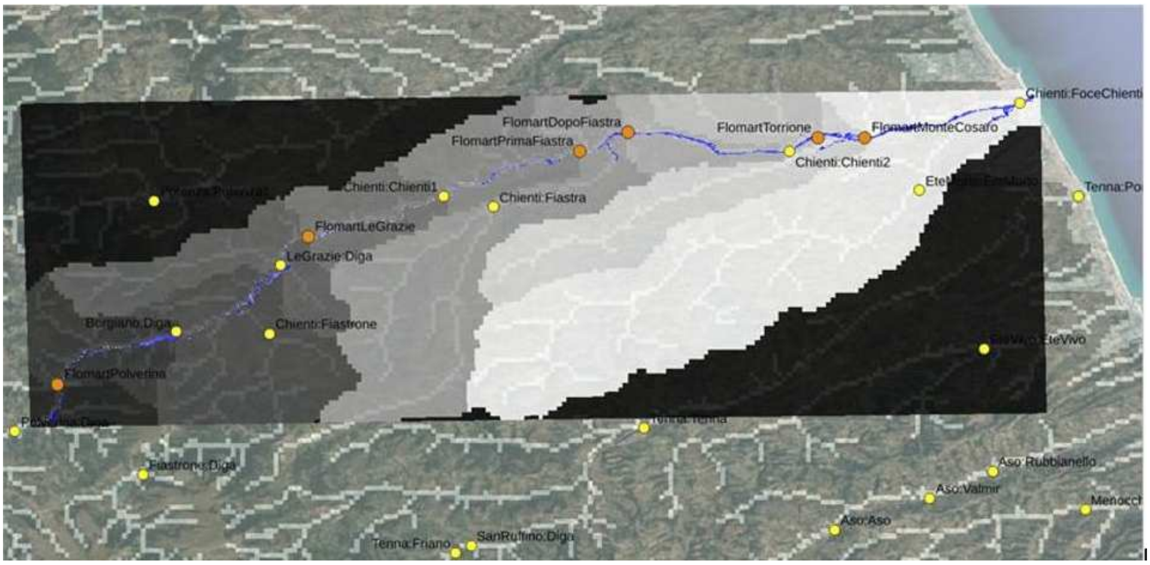


Figure 5.10: Layer 'AreeCompetenza' with the affected areas of the chosen sections for Chienti river (Marche region) superposed to a background google satellite layer, the Marche 'choice' layer, one hazard map from the abacus (in blue), and the shapefile with the selected sections (yellow and orange dots). Screenshot from Qgis.

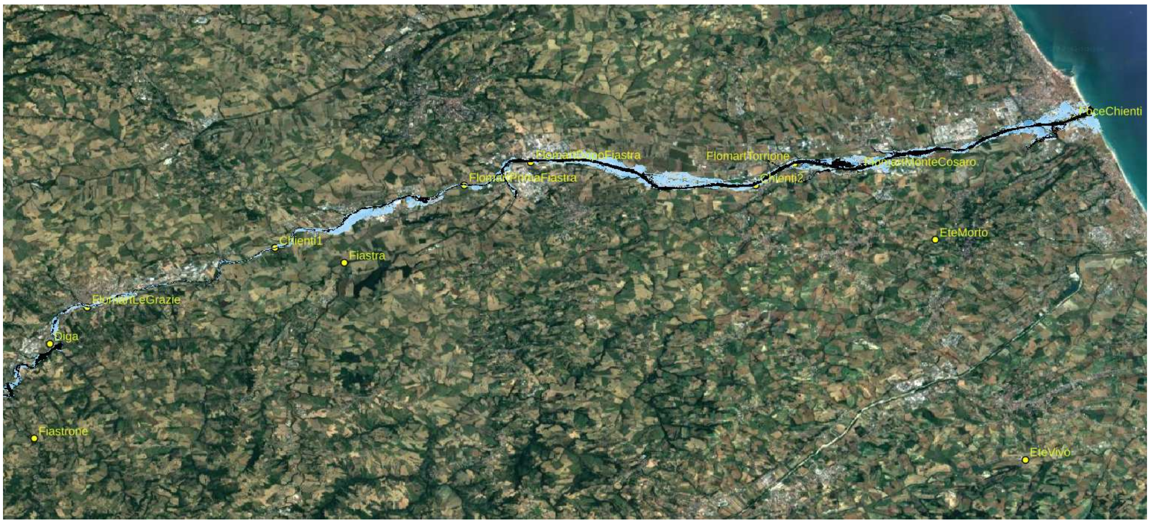


Figure 5.11: Comparison (on Qgis) between the hazard maps from the abacus of Chienti river with the minimum (1 year, dark blue) and the maximum (500 years, light blue) return period, respectively.

5.3.2 Event: simulated hydrographs

As for Foglia river, the event of November 10-12/11/2013 has been chosen. This event is known to have impacted Marche region with several floodings across the region.

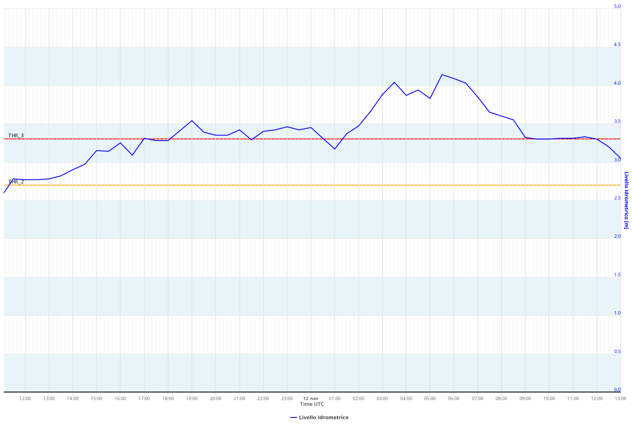
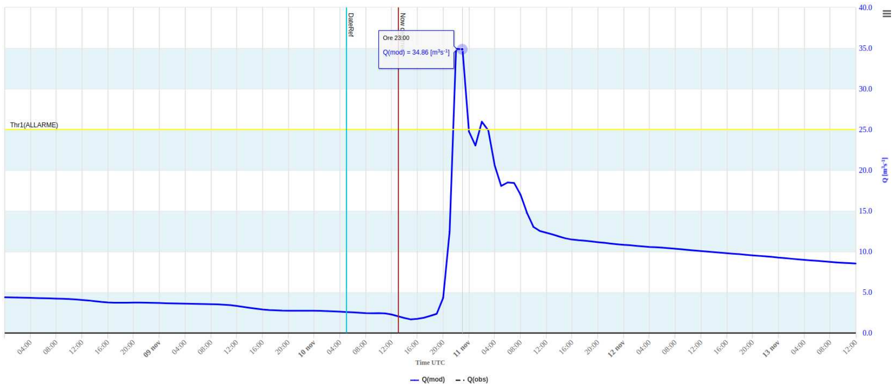


Figure 5.12: Water level recorded at section Chienti1 of Chienti river.

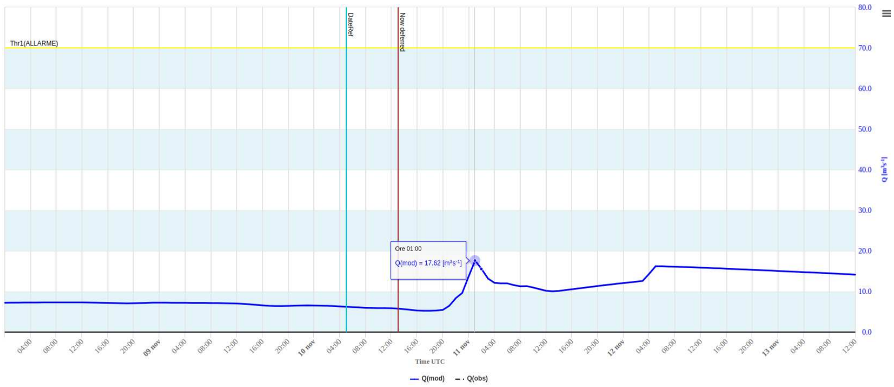
In particular, Figure **??** illustrates the hydrographs (taken from MyDewetra platform) at three different sections located from upstream of Chienti river to the river mouth in Civitanova Marche.

5.3.2.1 Run of 10/11/2013

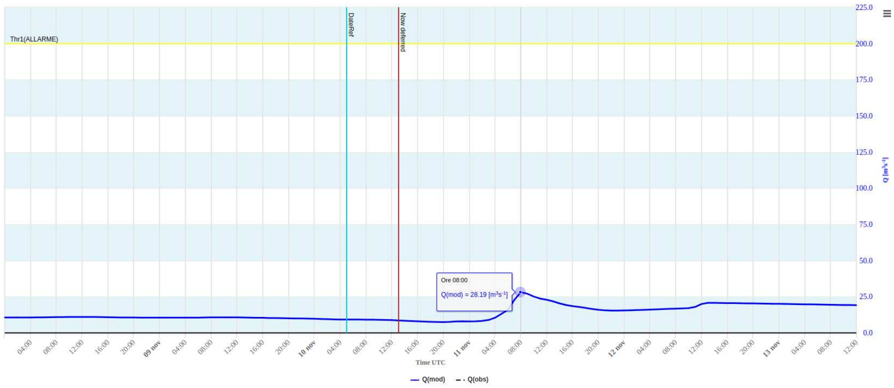
Figure 5.3.2.1 illustrates the simulated hydrographs (taken from MyDewetra platform) at three different sections located from the upstream sector of Chienti river to the river mouth at Civitanova.



((a)) Polverina 10/11/2013



((b)) Chienti1 10/11/2013

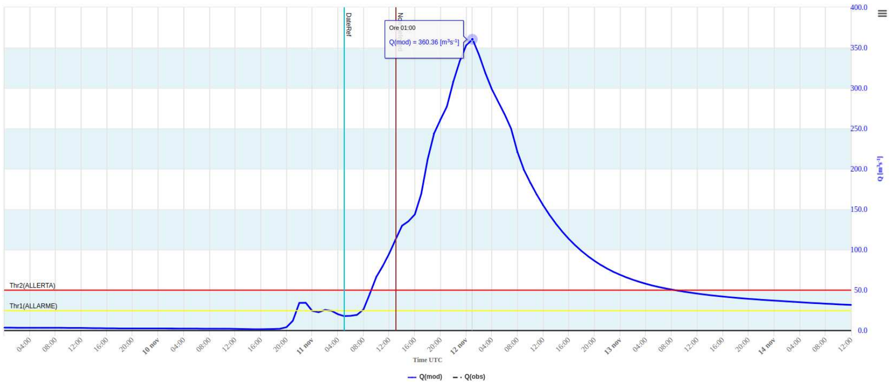


((c)) FoceChienti 10/11/2013

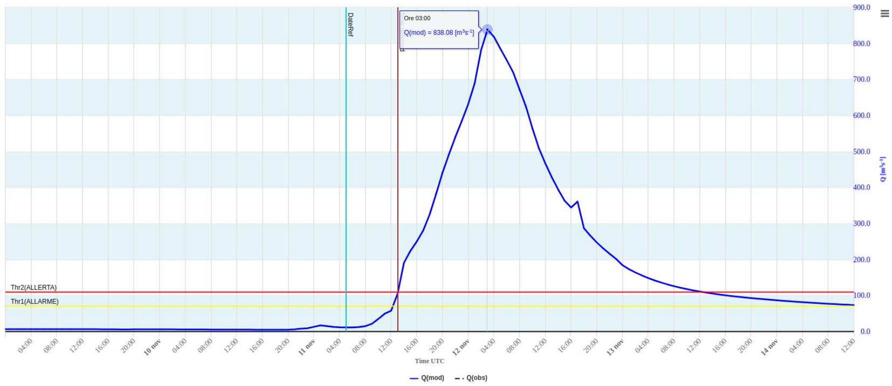
Figure 5.13: Hydrograph for run 2013/11/11-12 05:00 utc for Chienti river at sections (a) 'Polverina' (located upstream), (b) 'Chienti1' (located more downstream), (c) 'FoceChienti' (located at the river mouth, city of Civitanova Marche).

5.3.2.2 Run of 11/11/2013

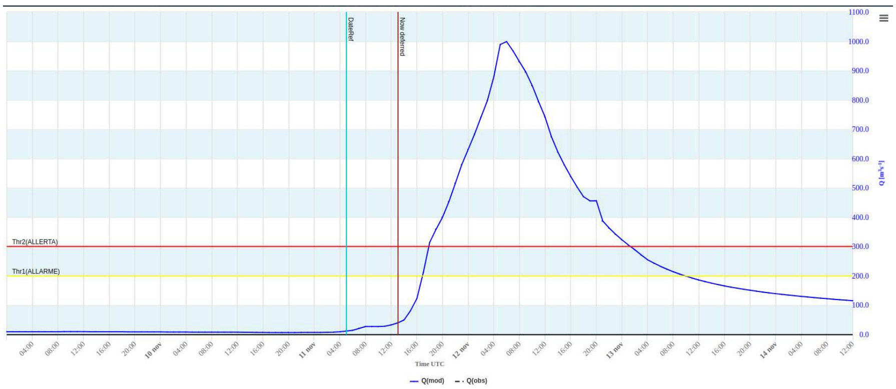
Figure 5.3.2.2 illustrates the simulated hydrographs (taken from MyDewetra platform) at three different sections located from the upstream sector of Chienti river to the river mouth at Civitanova.



((a)) Polverina 11/11/2013



((b)) Chienti1 11/11/2013



((c)) FoceChienti 11/11/2013

Figure 5.14: Hydrograph for run 2013/11/11 05:00 utc for Chienti river at sections (a) 'Polverina' (located upstream), (b) 'Chienti1' (located more downstream), (c) 'FoceChienti' (located at the river mouth, city of Civitanova Marche).

5.3.3 Output data

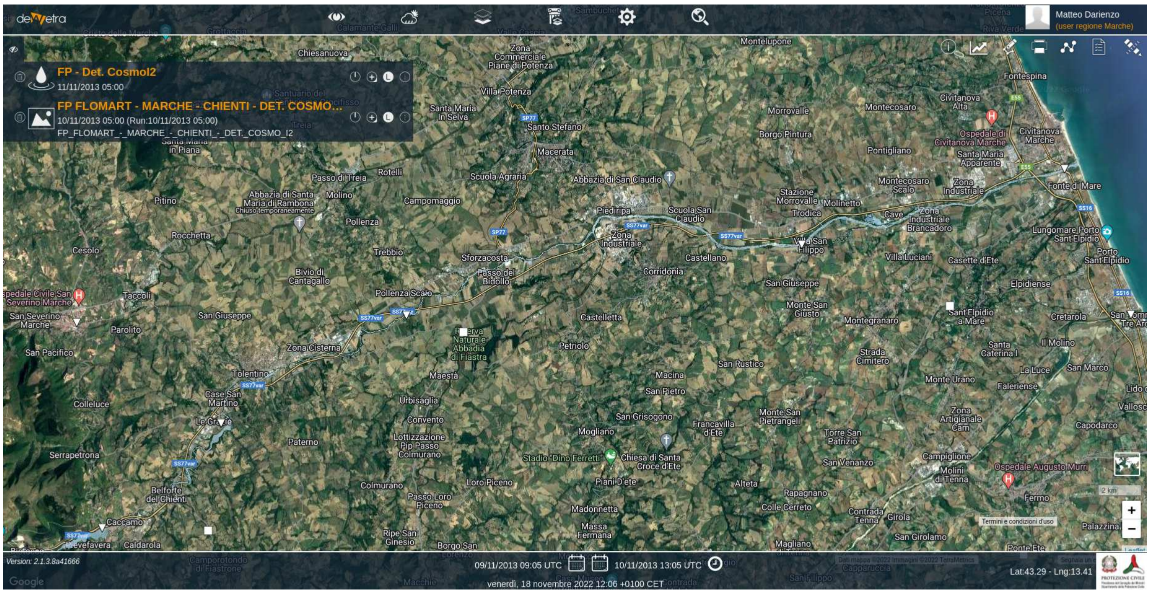


Figure 5.15: Flood map of Chienti river (Marche region) produced by Flomart and visualized in MyDewetra platform for the modelled event of 10/11/2013.


```
{
    "Chienti_Chienti1": {
        "discharge_value": "17.62",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.18",
        "scenario_weight_right": "0.82",
        "time": "2013-11-11_01:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_Chienti2": {
        "discharge_value": "23.27",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.19",
        "scenario_weight_right": "0.81",
        "time": "2013-11-11_05:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_Diga_Borgiano": {
        "discharge_value": "14.29",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
```

```
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.22",
        "scenario_weight_right": "0.78",
        "time": "2013-11-11_07:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_Diga_LeGrazie": {
        "discharge_value": "13.77",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.16",
        "scenario_weight_right": "0.84",
        "time": "2013-11-10_23:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartDopoFiastra": {
        "discharge_value": "23.42",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.2",
        "scenario_weight_right": "0.8",
        "time": "2013-11-11_03:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartLeGrazie": {
        "discharge_value": "16.23",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.18",
        "scenario_weight_right": "0.82",
        "time": "2013-11-10_23:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartMonteCosaro": {
        "discharge_value": "25.8",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "0",
        "scenario_idx_left": "1",
        "scenario_idx_right": "0",
        "scenario_n": "1",
        "scenario_weight_left": "0.2",
        "scenario_weight_right": "0.8",
        "time": "2013-11-11_06:00",
        "time_first_valid": "2013-11-08_00:00",
        "time_last_valid": "2013-11-11_22:00",
        "type_stream": "simulated",
```

```
            "type_value": "simulated"
        },
        "Chienti_FlomartPolverina": {
            "discharge_value": "12.94",
            "filter_stream": "both",
            "idx_first_valid": "0",
            "idx_last_valid": "94",
            "link_stream": "False",
            "scenario_idx": "0",
            "scenario_idx_left": "1",
            "scenario_idx_right": "0",
            "scenario_n": "1",
            "scenario_weight_left": "0.21",
            "scenario_weight_right": "0.79",
            "time": "2013-11-11 16:00",
            "time_first_valid": "2013-11-08 00:00",
            "time_last_valid": "2013-11-11 22:00",
            "type_stream": "simulated",
            "type_value": "simulated"
        },
        "Chienti_FlomartPrimaFiastra": {
            "discharge_value": "17.33",
            "filter_stream": "both",
            "idx_first_valid": "0",
            "idx_last_valid": "94",
            "link_stream": "False",
            "scenario_idx": "0",
            "scenario_idx_left": "1",
            "scenario_idx_right": "0",
            "scenario_n": "1",
            "scenario_weight_left": "0.18",
            "scenario_weight_right": "0.82",
            "time": "2013-11-11 03:00",
            "time_first_valid": "2013-11-08 00:00",
            "time_last_valid": "2013-11-11 22:00",
            "type_stream": "simulated",
            "type_value": "simulated"
        },
        "Chienti_FlomartTorrione": {
            "discharge_value": "23.38",
            "filter_stream": "both",
            "idx_first_valid": "0",
            "idx_last_valid": "94",
            "link_stream": "False",
            "scenario_idx": "0",
            "scenario_idx_left": "1",
            "scenario_idx_right": "0",
            "scenario_n": "1",
            "scenario_weight_left": "0.19",
            "scenario_weight_right": "0.81",
            "time": "2013-11-11 06:00",
            "time_first_valid": "2013-11-08 00:00",
            "time_last_valid": "2013-11-11 22:00",
            "type_stream": "simulated",
            "type_value": "simulated"
        },
        "Chienti_FoceChienti": {
            "discharge_value": "28.19",
            "filter_stream": "both",
            "idx_first_valid": "0",
            "idx_last_valid": "94",
            "link_stream": "False",
            "scenario_idx": "0",
            "scenario_idx_left": "1",
            "scenario_idx_right": "0",
            "scenario_n": "1",
            "scenario_weight_left": "0.18",
            "scenario_weight_right": "0.82",
            "time": "2013-11-11 08:00",
            "time_first_valid": "2013-11-08 00:00",
            "time_last_valid": "2013-11-11 22:00",
            "type_stream": "simulated",
            "type_value": "simulated"
        },
        "scenario_epsg_code": "EPSG:32633",
        "scenario_name": "Chienti",
        "scenario_time_now": "2013-11-10 05:00",
        "scenario_time_step": "2013-11-10 05:00"
}
```

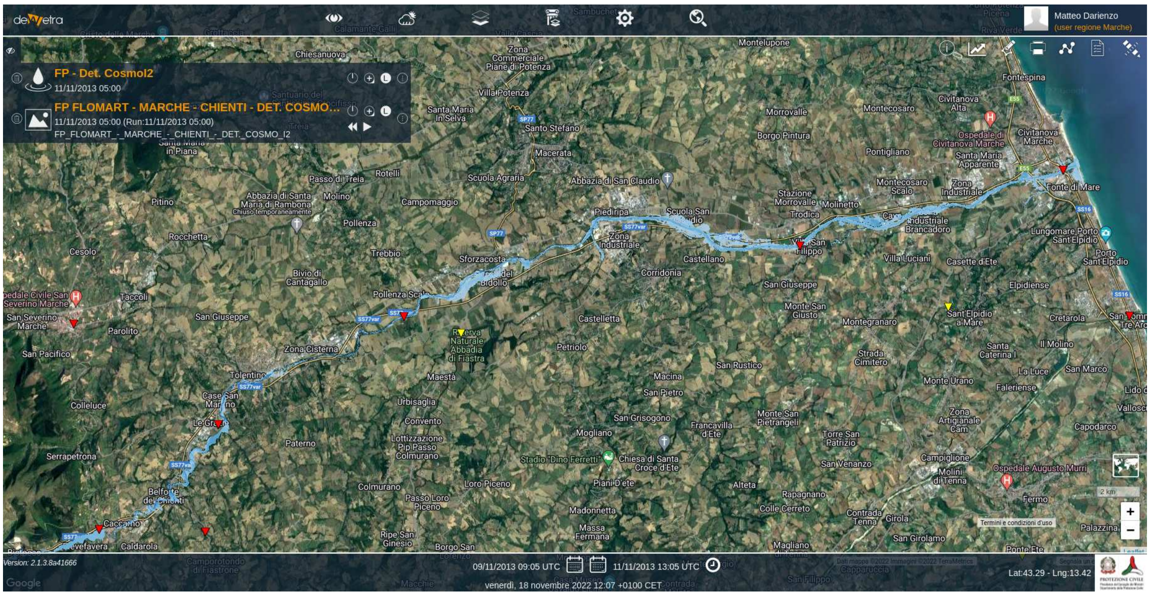


Figure 5.16: Flood map of Chienti river (Marche region) produced by Flomart and visualized in MyDewetra platform for the modelled event of 11/11/2013.


```
{
    "Chienti_Chienti1": {
        "discharge_value": "838.08",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "147",
        "scenario_idx_left": "148",
        "scenario_idx_right": "147",
        "scenario_n": "1",
        "scenario_weight_left": "0.48",
        "scenario_weight_right": "0.52",
        "time": "2013-11-12_03:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_Chienti2": {
        "discharge_value": "923.2",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "76",
        "scenario_idx_left": "76",
        "scenario_idx_right": "75",
        "scenario_n": "1",
        "scenario_weight_left": "0.73",
        "scenario_weight_right": "0.27",
        "time": "2013-11-12_05:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_Diga_Borgiano": {
        "discharge_value": "519.06",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "43",
        "scenario_idx_left": "44",
        "scenario_idx_right": "43",
        "scenario_n": "1",
        "scenario_weight_left": "0.09",
        "scenario_weight_right": "0.91",
```

```
        "time": "2013-11-12_01:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_Diga_LeGrazie": {
        "discharge_value": "788.4",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "88",
        "scenario_idx_left": "89",
        "scenario_idx_right": "88",
        "scenario_n": "1",
        "scenario_weight_left": "0.12",
        "scenario_weight_right": "0.88",
        "time": "2013-11-12_02:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartDopoFiastra": {
        "discharge_value": "926.12",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "89",
        "scenario_idx_left": "89",
        "scenario_idx_right": "88",
        "scenario_n": "1",
        "scenario_weight_left": "0.75",
        "scenario_weight_right": "0.25",
        "time": "2013-11-12_04:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartLeGrazie": {
        "discharge_value": "821.63",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "173",
        "scenario_idx_left": "174",
        "scenario_idx_right": "173",
        "scenario_n": "1",
        "scenario_weight_left": "0.09",
        "scenario_weight_right": "0.91",
        "time": "2013-11-12_03:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartMonteCosaro": {
        "discharge_value": "952.84",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "62",
        "scenario_idx_left": "63",
        "scenario_idx_right": "62",
        "scenario_n": "1",
        "scenario_weight_left": "0.09",
        "scenario_weight_right": "0.91",
        "time": "2013-11-12_05:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartPolverina": {
        "discharge_value": "467.38",
        "filter_stream": "both",
        "idx_first_valid": "0",
```

```
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "40",
        "scenario_idx_left": "41",
        "scenario_idx_right": "40",
        "scenario_n": "1",
        "scenario_weight_left": "0.41",
        "scenario_weight_right": "0.59",
        "time": "2013-11-12_01:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartPrimaFiastra": {
        "discharge_value": "841.38",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "136",
        "scenario_idx_left": "136",
        "scenario_idx_right": "135",
        "scenario_n": "1",
        "scenario_weight_left": "0.56",
        "scenario_weight_right": "0.44",
        "time": "2013-11-12_04:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FlomartTorrione": {
        "discharge_value": "933.32",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "70",
        "scenario_idx_left": "71",
        "scenario_idx_right": "70",
        "scenario_n": "1",
        "scenario_weight_left": "0.18",
        "scenario_weight_right": "0.82",
        "time": "2013-11-12_05:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "Chienti_FoceChienti": {
        "discharge_value": "998.89",
        "filter_stream": "both",
        "idx_first_valid": "0",
        "idx_last_valid": "94",
        "link_stream": "False",
        "scenario_idx": "29",
        "scenario_idx_left": "29",
        "scenario_idx_right": "28",
        "scenario_n": "1",
        "scenario_weight_left": "0.74",
        "scenario_weight_right": "0.26",
        "time": "2013-11-12_06:00",
        "time_first_valid": "2013-11-09_00:00",
        "time_last_valid": "2013-11-12_22:00",
        "type_stream": "simulated",
        "type_value": "simulated"
    },
    "scenario_epsg_code": "EPSG:32633",
    "scenario_name": "Chienti",
    "scenario_time_now": "2013-11-11_05:00",
    "scenario_time_step": "2013-11-11_05:00"
}
```

APPENDIX A

RELATION BETWEEN RETURN TIME AND FLOOD DISCHARGE

A.1 Relation between return time T and flood discharge Q_T

In real time, the forecasted or observed discharge (or the maximum value over the whole forecast window, e.g., 48 h) is used to select among all available maps from the abacus of flood scenarios the map with the closest return time. As previously mentioned, these maps are characterised and organized by an increasing number of the return time with 1-year step (e.g., *Foglia_WD_max_T001*, *Foglia_WD_max_T002*, ... , *Foglia_WD_max_T500*).

A relation between the return time (T) and the corresponding discharge (Q_T) is required:

$$Q_T = f(T) \tag{A.1}$$

For example, the following equation makes use of the concept of reference maximum annual discharge ('portata indice', http://www.pianidibacino.ambienteinliguria.it/IM/nervia/documenti/allegato1.pdf):

$$Q_T = K_T \, Q_{index} \tag{A.2}$$

where:

- Q_T is the discharge for the assigned return time T.
- Q_{index} is the reference maximum annual discharge (which is in general a mean or modal value).
- K_T is the growth factor (which depends on T).

In the next two sections we will briefly see how we estimated this relation for the Liguria and Marche regions, following the regionalization approach.

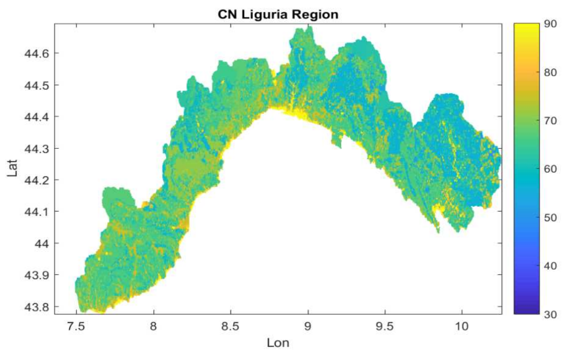
A.2 Liguria regionalization

The value of Q_{index} is considered for the Liguria domain as the discharge with return time equal to 2.9 years, and it is computed as:

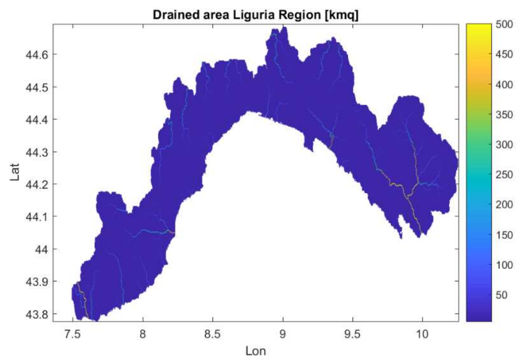
$$Q_{2.9} = 0.3\ A\ C_F\ a_{2.9}^{4/3}\ t_b^{-0.48} \tag{A.3}$$

where:

- $C_F = \frac{3}{4}\left(4\ 25.4 \frac{1000-10\ CN}{CN}\right)^{-1/3}$, method of Soil Conservation Service (USDA).
- $CN =$ runoff curve number (average), see Figure A.1(a)
- $A =$ drained area [km^2] at the considered section, see Figure A.1(b)
- $t_b = 0.25 + 0.27\ A^{1/2} =$ runoff time
- $a_{2.9} = 1.06\ E[H_1]$
- $E[H_1]$ = rainfall index (it varies with longitude), see Table A.1.



((a)) CN Liguria Region.



((b)) Drained area Liguria Region.

Table A.1: Table $E[H_1]$ vs. Longitude for Liguria.

Longitudine		$E[H_1]$	Longitudine		$E[H_1]$	Longitudine		$E[H_1]$
gradi	primi	mm	gradi	primi	mm	gradi	primi	mm
7	30	30.3	8	25	39.2	9	20	39.9
7	32.5	30.7	8	27.5	39.6	9	22.5	39.7
7	35	31.1	8	30	39.9	9	25	39.7
7	37.5	31.5	8	32.5	40.0	9	27.5	39.5
7	40	31.9	8	35	40.2	9	30	39.4
7	42.5	32.4	8	37.5	40.3	9	32.5	39.2
7	45	32.8	8	40	40.4	9	35	39.0
7	47.5	33.2	8	42.5	40.5	9	37.5	38.8
7	50	33.7	8	45	40.6	9	40	38.5
7	52.5	34.1	8	47.5	40.7	9	42.5	38.2
7	55	34.5	8	50	40.8	9	45	37.9
7	57.5	34.9	8	52.5	40.8	9	47.5	37.5
8	0	35.4	8	55	40.8	9	50	37.1
8	2.5	35.8	8	57.5	40.8	9	52.5	36.7
8	5	36.2	9	0	40.8	9	55	36.2
8	7.5	36.6	9	2.5	40.8	9	57.5	35.7

Then, Equation A.4 becomes:

$$Q_T = K_T \; Q_{2.9} \tag{A.4}$$

Table A.2 provides the values of factor K_T regionalized for Liguria for seven increasing return times T (from 5 to 500 years), obtained from a previous study.

Table A.2: Data-set of K_T vs. T (years) for Liguria.

T [anni]	5	10	30	50	100	200	500
K_T	1.29	1.79	2.90	3.47	4.25	5.02	6.04

These values (T, K_T) are also plotted on Figure A.2. For all other intermediate return times T, a regression line (e.g., logarithmic) can be estimated using the available data-set.

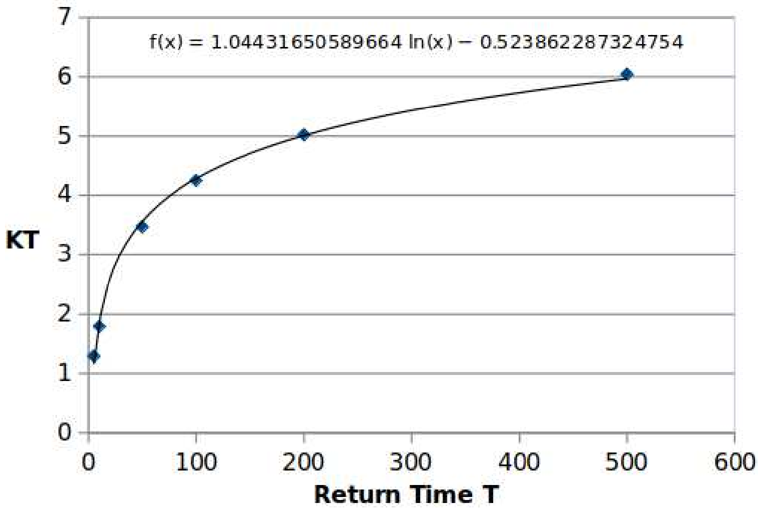


Figure A.2: Logarithmic regression applied to the relation K_T vs. T for Liguria domain.

with equation:

$$K_T = 1.0443 \; \ln T - 0.5239 = Q_T / Q_{2.9} \tag{A.5}$$

from which we can derive T:

$$\ln T = \frac{Q_T/Q_{2.9} + 0.5239}{1.0443} \tag{A.6}$$

hence:

$$T = exp\left(\frac{Q_T + Q_{2.9}\; 0.5239}{1.0443 \; Q_{2.9}}\right) \tag{A.7}$$

For $T < 5$ years (lowest available value) the relation requires extrapolation, thus a simple linear interpolation can be assumed in order to avoid negative values (or nan) for K_T. Thus, the minimum K_T for validity of Equation A.5, is re-estimated as $K_T^* = 1.0443 \ln 5 - 0.5239 = 1.16$. For values of $Q_T/Q_{2.9} < K_T^*$, the following formulation is assumed (considering $T_{min} = 0$):

$$T = \frac{Q_{2.9} * K_T^*}{exp\left(\frac{Q_T + Q_{2.9}\; 0.5239}{(1.0443 \; Q_{2.9}}\right)} \; Q_T \tag{A.8}$$

These two relations must both be implemented in the Flomart python code inside function ***"cmp_tr_general()"*** of module *"lib_utils_tr.py"*, as explained in Section 2.5.

A.3 Marche regionalization

A report released of 2016 on the activity performed by the Commissario Delegato Maltempo Maggio 2014 and CIMA Foundation for the "modelling and definition of the hydrological variables for the structural and non structural restoration of the main hydrography of Marche Region" (Reg Int: 2015/28 – Nr. 670), describes the regionalization of maximum annual discharges performed on Marche region.

Because of the scarsity of historical hydrographs, the regionalization was performed through the following three steps:

- generation of synthetic dataset of extreme rainfall events (with different critical durations) using RAINFARM model.
- computation of hydrological simulations (by means of Continuum ©CIMA calibrated on the Marche Region domain) for each of the generated rainfall events.
- estimation of the probability distribution at each point of the river network.

The high number of simulated events, and therefore the high number of synthetic samples available for each cell, allows a statistical description up to very high return times. Peak flow maps were then produced, with a value for each cell of the grid, corresponding to the return times 2, 5, 10, 20, 50, 100, 150, 200, 500 years.

A.3.1 Basins with drained area < 50 km^2

The reliability of quantiles maps may be lower for small basins; furthermore, the smaller basins are poorly represented due to the low number of cells. A dimensionless sample distribution was considered. The index variable considered is the average of the annual maximum discharge generated synthetically. A regression analysis was performed between the index variable and the upstream drained area (Figure A.3). The drained area has a consistent dependence on the index variable, but with considerable variability. An average power law relationship was then fitted:

$$Q_{index} = 1.6119\ A^{0.9735}\ \ [m^3/s] \tag{A.9}$$

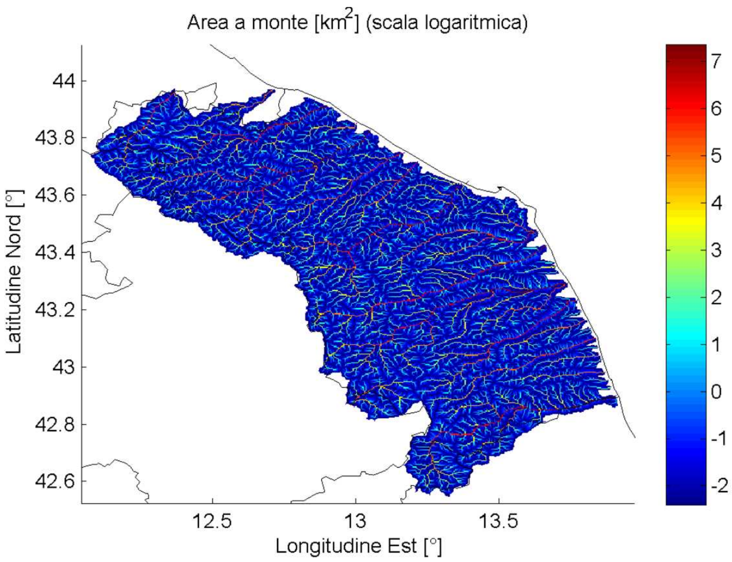


Figure A.3: Gridded layer with drained area values for Marche Region.

Then we can compute the discharge with Equation A.4 ($Q_T = K_T\ Q_{index}$), by using the values of the growth factor K_T reported in Table A.3.

Table A.3: Table of Marche regionalization for values of T vs. K_T for small basins (A<50 Km^2

Tempo di ritorno [anni]	**2**	**5**	**10**	**20**	**50**	**100**	**150**	**200**	**500**	**1000**
Fattore di crescita K_T	0.864	1.375	1.755	2.155	2.730	3.207	3.505	3.725	4.482	5.115

).

A.3.2 Basins with drained area > 50 km^2

Instead, to estimate the discharge value for larger basins (> 50 km^2) only the quantile maps should be used. Table A.4 reports the annual maximum discharge values for a few assigned return times for the main regional basins closed to the river mouth. The two basins studied in this guide and operationally implemented for Regione Marche are Foglia and Chienti basins, and they both drained areas larger than 50 km^2. They are hereafter analyzed.

Table A.4: Table with the regionalization of maximum annual discharges for Marche Region, Italy. Taken from report of CIMA and Commissario delegato maltempo Maggio 2014 "La modellazione a definizione delle grandezze idrologiche per la messa in sicurezza strutturale e non strutturale del reticolo idrografico principale della Regione Marche", 2016.

Tempo di ritorno [anni]	**2**	**5**	**10**	**20**	**50**	**100**	**150**	**200**	**500**
Foglia	188	361	505	682	904	1079	1230	1270	1515
Arzilla	93	147	184	220	259	286	303	307	387
Metauro	306	488	658	838	1018	1310	1407	1481	1711
Cesano	205	342	441	535	659	776	844	870	1112
Misa	216	341	450	587	752	903	991	1038	1085
Esino	329	581	750	1017	1403	1674	1859	2007	2404
Musone	155	286	400	528	676	819	915	954	1157
Potenza	199	341	445	543	664	770	802	842	1147
Chienti	315	547	742	928	1158	1350	1421	1456	1637
Tenna	250	354	451	550	661	789	836	925	1099
Ete Vivo	105	172	218	263	309	344	370	408	467
Aso	130	222	286	396	490	664	697	829	1041
Menocchia	96	145	179	209	241	265	279	289	352
Tesino	118	188	231	284	338	382	400	404	528
Tronto	263	426	546	662	962	1138	1257	1309	1665

A.3.2.1 Foglia river

The procedure consists in extracting and plotting the values of discharge from the maps of quantiles for each section of interest. As an example, Table A.4 shows the values (Q_T, T) for sections close to the river mouth.

By plotting these values against the corresponding return times T we obtain the relation of Figure A.5, and a regression line can be estimated (e.g., logarithmic, spline).

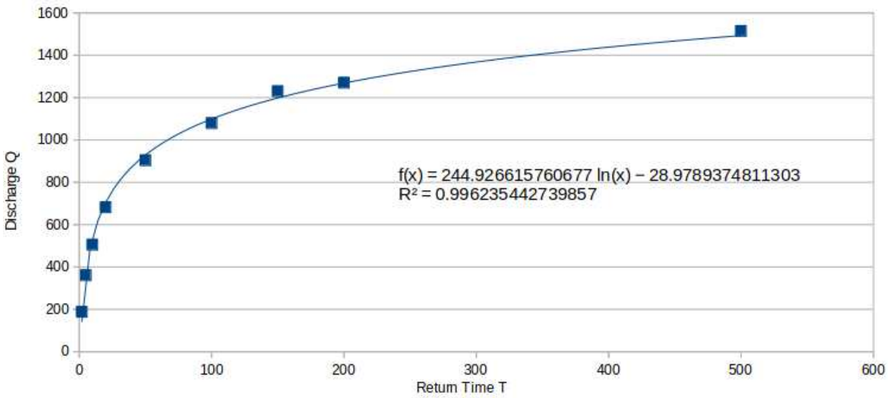


Figure A.4: Graph T vs. Q for Foglia river, Marche Region.

For this section (river mouth of Foglia river) the following logarithmic regression function is used:

$$Q_T = 244.926\ \ln T - 28.98 \tag{A.10}$$

hence:

$$T = exp\left(\frac{Q_T + 28.98}{244.93}\right) \tag{A.11}$$

The last return time for which the relation of Equation A.11 is valid is $T = 2$ years. Below this value, the relation requires extrapolation. The corresponding limit discharge value:

$$Q^* = 244.926\ \ln 2 - 28.98 = 140.79 \tag{A.12}$$

For $Q < Q^*$ a mere linear interpolation is assumed in order to avoid negative values (or nan) (considering the origin to $T_{min} = 0$ and $Q_{min} = 0$):

$$T = \frac{exp\left(\frac{Q^* + 28.98}{244.93}\right)}{2}\ Q \tag{A.13}$$

The same procedure is applied to all other sections. Thus a specific equation is assigned to each section. See Figure A.5.

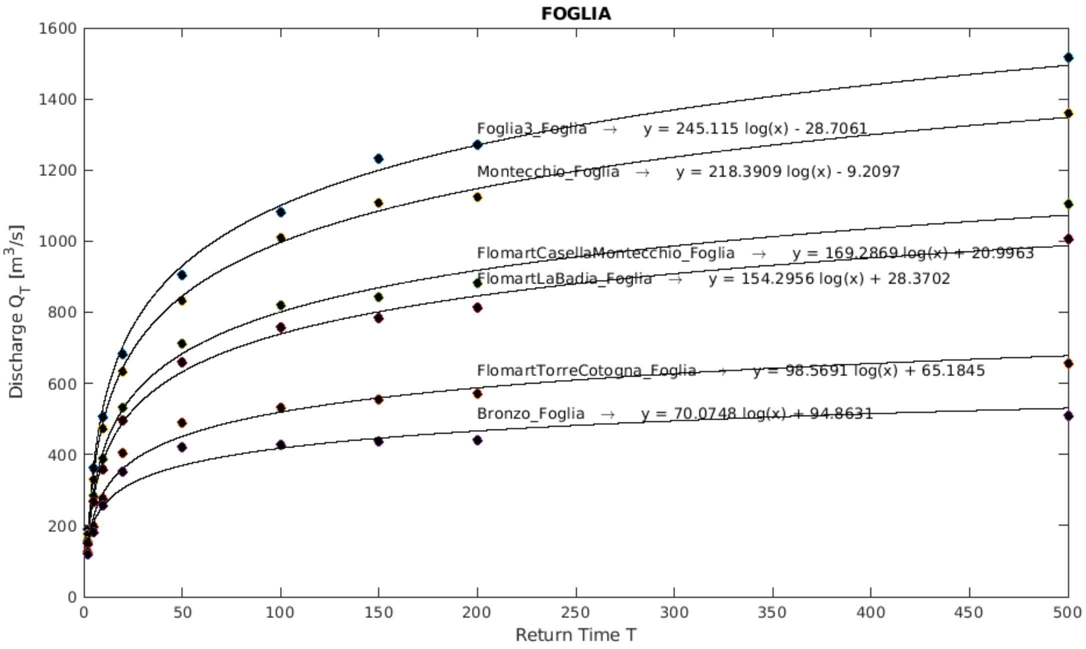


Figure A.5: Graph T vs. Q for Foglia river, Marche Region.

A.3.2.2 Chienti river

Similarly to the case study of Foglia river, extracting and plotting the values for Chienti river from Table A.4 we can estimate a regression line (e.g., logarithmic, spline).

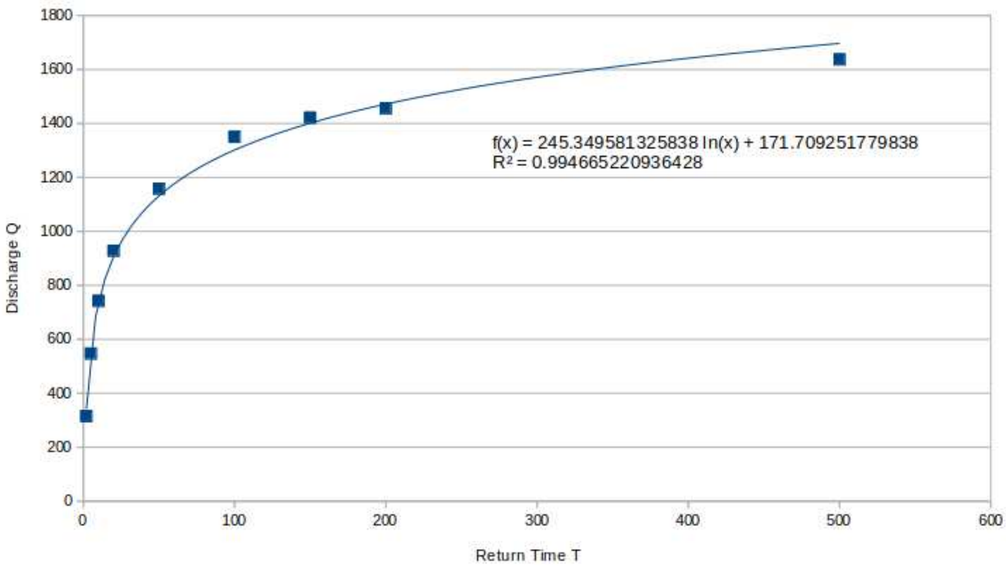


Figure A.6: Relation T vs. Q for Chienti river, Marche Region.

The following logarithmic regression function can be used:

$$Q_T = 245.35 \ \ln T + 171.71 \tag{A.14}$$

hence:

$$T = exp\left(\frac{Q_T + 28.98}{244.93}\right) \tag{A.15}$$

The same procedure is applied to all other sections. Thus, a specific equation is assigned to each section. See Figure A.7.

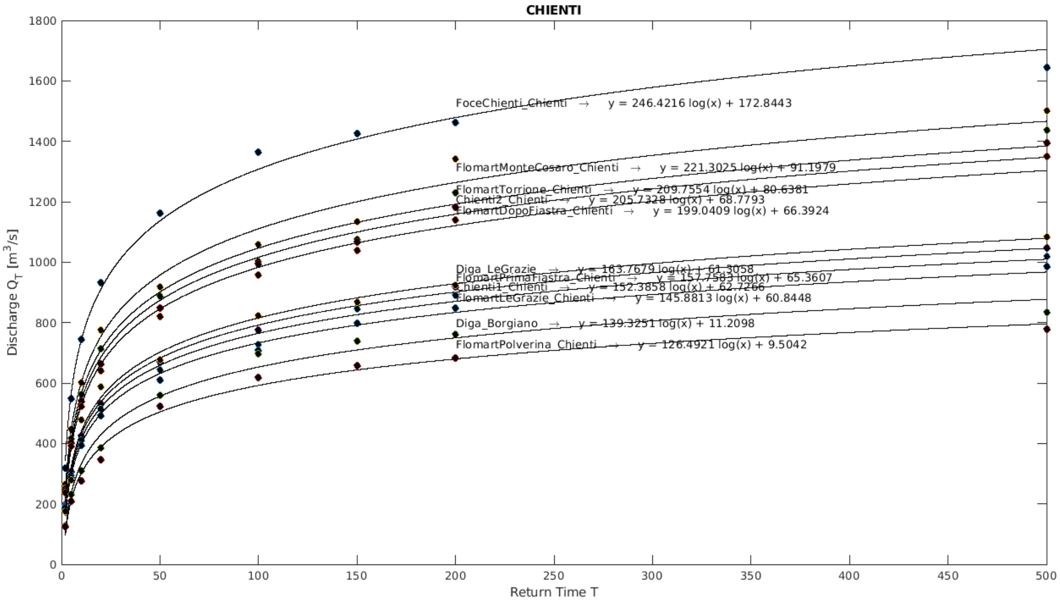


Figure A.7: Relation T vs. Q for Chienti river, Marche Region.

However, observing Figures A.5,A.7 we can notice a poor fit of the relation in particular for the lowest return times, for $T < 2$ years.

Thus, we suggest in this case to implement a mere linear interpolation between the values of the data-set, for each section. This solves also the problem of the validity of the function for $T << 2$ years.

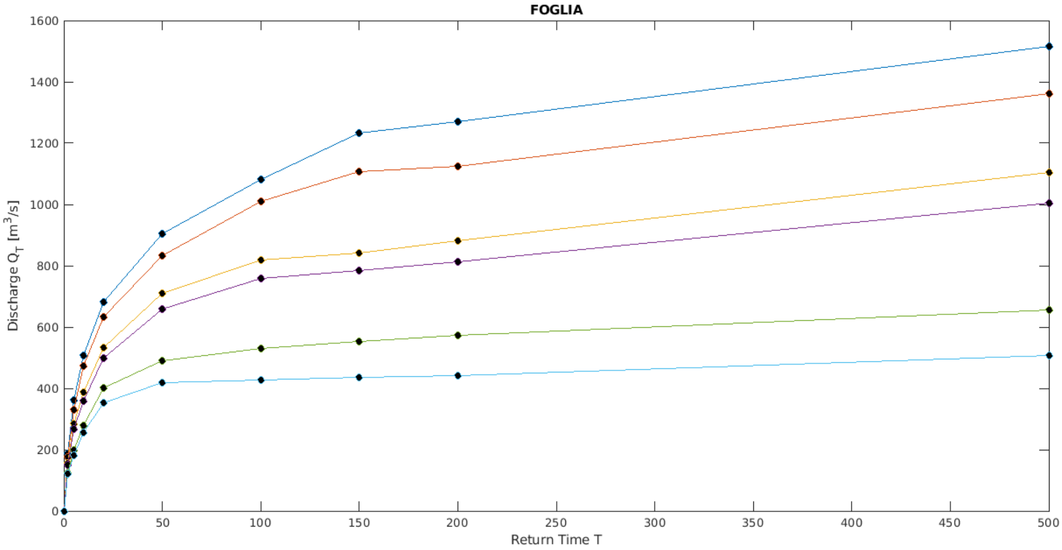


Figure A.8: Linear interpolation of the relation T vs. Q for Foglia river.

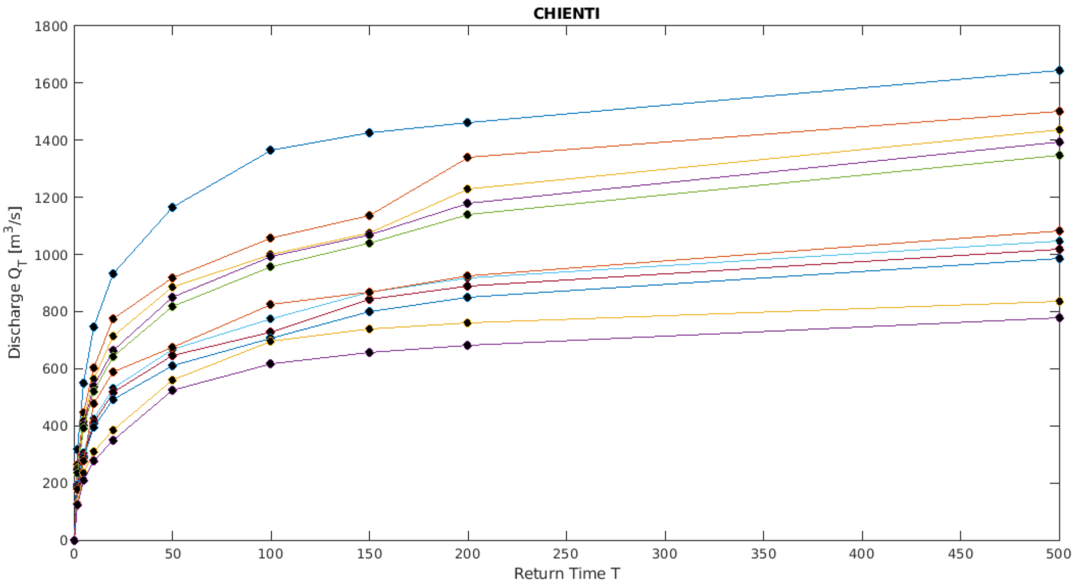


Figure A.9: Linear interpolation of the relation T vs. Q for Chienti river.

The following matlab script allows the upload of the quantiles map with discharge values and the retrieval of data-sets $T - Q$ for each section of interest (defined in the .mat input static file):

```
sFileName_QT2 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T2.asc'];
sFileName_QT5 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T5.asc'];
sFileName_QT10 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T10.asc'];
sFileName_QT20 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T20.asc'];
sFileName_QT50 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T50.asc'];
sFileName_QT100 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T100.asc
    '];
sFileName_QT150 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T150.asc
    '];
sFileName_QT200 = [path_preparation_data, '/Mappe_Regionalizzazione/Mappa_quantili_T200.asc
    '];
sFileName_QT500 = [path_preparation_data, '/Mappe_Regionalizzaziones/Mappa_quantili_T500.asc
    '];

% Load maps of Q for increasing quantiles
[a2dMap_QT2, a2dCoord_QT2] = arcgridread(sFileName_QT2);
a2iQT2 = a2dMap_QT2;
[a2dMap_QT5, a2dCoord_QT5] = arcgridread(sFileName_QT5);
a2iQT5 = a2dMap_QT5;
[a2dMap_QT10, a2dCoord_QT10] = arcgridread(sFileName_QT10);
a2iQT10 = a2dMap_QT10;
[a2dMap_QT20, a2dCoord_QT20] = arcgridread(sFileName_QT20);
a2iQT20 = a2dMap_QT20;
[a2dMap_QT50, a2dCoord_QT50] = arcgridread(sFileName_QT50);
a2iQT50 = a2dMap_QT50;
[a2dMap_QT100, a2dCoord_QT100] = arcgridread(sFileName_QT100);
a2iQT100 = a2dMap_QT100;
[a2dMap_QT150, a2dCoord_QT150] = arcgridread(sFileName_QT150);
a2iQT150 = a2dMap_QT150;
[a2dMap_QT200, a2dCoord_QT200] = arcgridread(sFileName_QT200);
a2iQT200 = a2dMap_QT200;
[a2dMap_QT500, a2dCoord_QT500] = arcgridread(sFileName_QT500);
a2iQT500 = a2dMap_QT500;

% Get the Q for each section and for each available percentile:
for i=1:size(sezioni_indici_relativi,1)
        a1dQT2(i)   =  a2iQT2(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT5(i)   = a2iQT5(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT10(i)  = a2iQT10(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT20(i)  = a2iQT20(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT50(i)  = a2iQT50(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT100(i) = a2iQT100(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT150(i) = a2iQT150(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT200(i) = a2iQT200(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
        a1dQT500(i) = a2iQT500(sezioni_indici_relativi(i,1),sezioni_indici_relativi(i,2));
end

% Create data-set T-Q for each section:
T = [0 2 5 10 20 50 100 150 200 500];
for i=length(L):-1:1
        a1dQT2_sort(i) = a1dQT2(indici_sort(i));
        a1dQT5_sort(i) = a1dQT5(indici_sort(i));
        a1dQT10_sort(i) = a1dQT10(indici_sort(i));
```

```
        a1dQT20_sort(i) = a1dQT20(indici_sort(i));
        a1dQT50_sort(i) = a1dQT50(indici_sort(i));
        a1dQT100_sort(i) = a1dQT100(indici_sort(i));
        a1dQT150_sort(i) = a1dQT150(indici_sort(i));
        a1dQT200_sort(i) = a1dQT200(indici_sort(i));
        a1dQT500_sort(i) = a1dQT500(indici_sort(i));

        Q = [0 a1dQT2_sort(i) a1dQT5_sort(i) a1dQT10_sort(i) a1dQT20_sort(i) a1dQT50_sort(i)
            a1dQT100_sort(i) a1dQT150_sort(i) a1dQT200_sort(i) a1dQT500_sort(i)];
end
```

Listing A.1: Matlab script to get the data-sets T, Q_T and regression equation for each section.

A.4 Creation of gridded variable Qindex from the maps of discharge time series

For small catchments, or in general, if variable Q_{index} is required as input in the relation $T = f(Q_T)$, we can compute this gridded variable Q_{index} as the average maximum annual discharge (considering a discharge time series of a period of at least 10 years).

An example of this approach is proposed in the matlab script *grid_Qindex_generation.m* located in the pre-processing folder of Flomart and reported below.

```
clc; clear;
% COMPUTE Qindex if not available:
% Period with available discharges must be >= 10 years !!

%% INPUTS:
%%%%%%%%
% path with the choice grid layer:
sPathGridGeoData      ='/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Chienti/data_static/PREPARATION/gridded_marche';
% path containing discharge values for each cell of the grid for a time:
sPathQmaps            = [path_preparation_data, '/discharge_2010_2020_NEW'];
sDateFrom             = '201101012300';
sDateTo               = '202012312300';
dt                    =  1;  % time step of files (days)
path_preparation_data = '/home/matteo/Documents/CIMA_projects/RT_FloodMapping/data/data_Marche_Chienti/data_static/PREPARATION';
domain_name           = 'Chienti';


%% START:
%%%%%%%%%
sFileName_choice = [sPathGridGeoData, '/marche.choice.txt'];
[a2dMap_choice, a2dCoord_choice] = arcgridread(sFileName_choice);
a2iChoice = a2dMap_choice;
a2iChoice(isnan(a2iChoice)) = -1;  %replace all NaN with -1
[iNRows,iNCols]= size(a2iChoice);
iNoData        = -9999; % valore per dati mancanti nelle mappe Netcdf
% define startuing and ending time of simulations:
nDateFrom      = datenum(sDateFrom,'yyyymmddHHMM');
nDateTo        = datenum(sDateTo,'yyyymmddHHMM');
%initialise current time:
nNow           = nDateFrom;
%initialise matrixes with discharge:
a3dMapQ        = zeros([iNRows,iNCols,10]);
maxQannual     = zeros([iNRows,iNCols]);
% initialise counter:
iCountDay      = 0;
iCountYear     = 1;

while nNow<=nDateTo
    iCountDay = iCountDay + 1;
    sDate = datestr(nNow,'yyyymmddHHMM');
```

```
    disp(sDate);
    % extract month, day, hour from file name:
    iYear   = str2double(sDate(1:4));
    iMonth  = str2double(sDate(5:6));
    iDay    = str2double(sDate(7:8));
    iHour   = str2double(sDate(9:10));
    sPathNow = [sPathQmaps,'/',sDate(1:4),'/',sDate(5:6),'/',sDate(7:8),'/'];
    try
        sFileNameMap = ['hmc.output-grid.',sDate,'.nc.gz'];
        a2dMap = Continuum_getMap_NC(sPathNow, sFileNameMap,'Discharge');  %instantaneous Discharge (m)
        a2dMap(a2dMap==iNoData) = NaN;
        % compute maximum between actual Q(t) and all previous ones of the
        % year:
        a3dMapQ(:,:,iCountYear) = max(a3dMapQ(:,:,iCountYear), a2dMap);
        %a3dMapQ(:,:,iCountYear) = arrayfun(@(x,y) max(x(:),y(:)), a3dMapQ(:,:,iCountYear), a2dMap);
    catch
        disp('problem with format of netcdf file!! skip!');
    end

    if iMonth==12 & iDay ==31 & iHour == 23
            maxQannual(:,:,iCountYear) =  a3dMapQ(:,:,iCountYear);
            %display(maxQannual);
            display('%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%')
            iCountDay = 0;
            % go to next year:
            iCountYear = iCountYear + 1;
    end
    % pass to next time step:
    nNow = datenum(sDate,'yyyymmddHHMM')+dt;
end


% Calcolo mappa di portata massima annuale media su deici anni:
maxQannual_mean(:,:) = zeros([iNRows,iNCols]);
for iCountYear=1:10
    tmp = maxQannual_mean(:,:);
    %tmp(OutOfCN) = NaN;
    maxQannual_mean(:,:) = tmp + maxQannual(:,:,iCountYear);
end
maxQannual_mean(:,:) = maxQannual_mean(:,:)./10;
a2dQindice = maxQannual_mean;



% Save obtained Qindex layer:
save([path_preparation_data, '/Qindex_bis_',domain_name,'.mat'], 'a2dQindice');
```

Listing A.2: Matlab script to generate gridded variable Qindex

APPENDIX B

LAUNCHER OF FLOMART ON OPERATIONAL SERVER

Flomart has been operationally implemented in the computational server of Marche region ("hydro"). The last version of Flomart program scripts are located at the following path:

/hydro/library/flomart/

Instead, the launcher and the json file with the run settings are located at the following paths (one for Foglia river and one for Chienti river):

/hydro/fp_tools_postprocessing/flomart_app_execution/Foglia/
/hydro/fp_tools_postprocessing/flomart_app_execution/Chienti/

Figure B.1: Folder where the launcher script (.sh) for operationally running Flomart on the server of Marche Region "hydro".

As shown in Figure B.2, both for Chienti and for Foglia, there are two different launchers:

app_flomart_manager_nwp_ecmwf0100_realtime_FOGLIA.sh
app_flomart_manager_nwp_lami-2i_realtime_FOGLIA.sh

One for running Flomart using as input the forecasted hydrographs computed by the daily run ECMWF, and the second one to use the run COSMO-LAMI-2i.

Here below an example of the launcher is reported. The script initially sets the virtual environment for python3 and the required file paths. Then, with variable ***"time_now"*** it is possible to fix the timing of the run.

For example, the current time $(date -u +"%Y-%m-%d %H:00") for operational uses, or by defining a date with format "YYYY-mm-dd HH:MM" for historical runs or debug.

In the last part of the launcher script, the results (the raster of flood map and the .json with a few hydrological info) are transferred to another server of Marche region where MyDewetra platform is synchronized (for the final visualization of the maps as shown in Section 1.1.3).

```
#!/bin/bash -e
#-----------------------------------------------------------------------------------------
# Script information
script_name='APP - RUNNER FLOMART - MODELLED WS DISCHARGE - REALTIME'
script_version="2.0.3"
script_date='2021/11/18'

virtualenv_folder='/hydro/library/fp_libs_python3/'
```

```
virtualenv_name='virtualenv_python3'
script_folder='/hydro/library/fp_package_flomart_application/'

# Execution example:
# python3 app_flomart_main.py -settings_algorithm app_flomart_configuration_nwp_lami-2i_realtime_FOGLIA.json -time "2022-09-01 12:00"
#-----------------------------------------------------------------------------------------

#-----------------------------------------------------------------------------------------
# Get file information
script_file='/hydro/library/fp_package_flomart_application/app_flomart_main/app_flomart_main.py'
settings_file='/hydro/fp_tools_postprocessing/flomart_app_execution/Foglia/app_flomart_configuration_nwp_ecmwf0100_realtime_FOGLIA.json'

script_file_transfer='/hydro/library/fp_package_hyde/tools/tool_processing_datasets_transfer/hyde_tools_transfer_datasets.py'
settings_file_transfer='/hydro/fp_tools_postprocessing/flomart_app_execution/Foglia/app_flomart_transfer_nwp_ecmwf0100_realtime_FOGLIA.json'

# Time period execution
time_period_hour=0 # hour(s)

# Get information (-u to get gmt time)
time_now=$(date -u +"%Y-%m-%d %H:00")
#time_now="2020-10-04 00:45" # DEBUG
#-----------------------------------------------------------------------------------------

#-----------------------------------------------------------------------------------------
# Activate virtualenv
export PATH=$virtualenv_folder/bin:$PATH
source activate $virtualenv_name

# Add path to pythonpath
export PYTHONPATH="${PYTHONPATH}:$script_folder"
#-----------------------------------------------------------------------------------------

# ----------------------------------------------------------------------------------------
# Info script start
echo " ==================================================================================="
echo " ==> "$script_name" (Version: "$script_version" Release_Date: "$script_date")"
echo " ==> START ..."

# Iterate over hours
time_run=$(date -d "$time_now" +'%Y-%m-%d %H:00')
for time_period_step in $(seq 0 $time_period_hour); do

    # Parse time information
    time_step=$(date -d "$time_run ${time_period_step} hour ago" +'%Y-%m-%d %H:00')
  # Run python script (using setting and time)
  echo " ===> COMPUTE FLOOD SCENARIOS [${time_step}] ... "
  echo " ===> COMMAND LINE: " python $script_file -settings_file $settings_file -time $time_step
  python $script_file -settings_file $settings_file -time "$time_step"
  echo " ===> COMPUTE FLOOD SCENARIOS [${time_step}] ... DONE"
```

```
  #-----------------------------------------------------------------------------------
  # Run python model instance (using settings and time)
  echo " ===> RUN FLOMART HAZARD MAPS TRANSFER ... "
  # Run python script (using setting and time)
  python3 $script_file_transfer -settings_file $settings_file_transfer -time "$time_step"
  echo " ===> RUN FLOMART HAZARD MAPS TRANSFER ... DONE"
# ----------------------------------------------------------------------------------------

done
# ----------------------------------------------------------------------------------------

# ----------------------------------------------------------------------------------------
# Info script end
echo " ==> "$script_name" (Version: "$script_version" Release_Date: "$script_date")"
echo " ==> ... END"
echo " ==> Bye, Bye"
echo " ==================================================================================="
# ----------------------------------------------------------------------------------------
```

Listing B.1: Example of the launcher of Flomart application.

The settings for Flomart run, such as the time period, paths of dyanamic and static input files, type of scenario analysis (as described in Chapter 4.1), are also located in this folder as:

app_flomart_configuration_nwp_ecmwf0100_realtime_FOGLIA.sh
app_flomart_configuration_nwp_lami-2i_realtime_FOGLIA.sh

The following options for creation of scenarios have been selected:

- "tr_min" : 0
- "tr_max" : 500
- "tr_freq": 1
- "scenario_tiling": "weighted"
- "scenario_boundary": "both"
- "scenario_analysis": "max_period"
- "scenario_type": "simulated"

And the following options for analysing the input dynamic data:

- "type": "json"
- "variables": "time": "time_period", "discharge": "time_series_discharge_simulated", "water_level": null
- "method_data_occurrence": "all" (all values)
- "method_data_analysis" : "max" (maximum values between different available series)
- "method_data_filling": null (no filling of data where missing)
- "method_data_null": null (no management of null data)
- "time_period": 24
- "time_frequency": "H"
- "time_rounding": "D"

The execution of the launchers is implemented in the crontab of the machine.

```
# POSTPROCESSING TOOL - FLOMART - FLOOD MAP IN REALTIME READING 2 RUN FORECAST ECMWF E LAMI
12,26,49 7,9,11 * * * /hydro/fp_tools_postprocessing/flomart_app_execution/Foglia/app_flomart_manager_nwp_ecmwf0100_realtime_FOGLIA.sh
5,17,40 7,9,11 * * * /hydro/fp_tools_postprocessing/flomart_app_execution/Foglia/app_flomart_manager_nwp_lami-2i_realtime_FOGLIA.sh
14,30,44 7,9,11 * * * /hydro/fp_tools_postprocessing/flomart_app_execution/Chienti/app_flomart_manager_nwp_ecmwf0100_realtime_CHIENTI.sh
9,21,36 7,9,11 * * * /hydro/fp_tools_postprocessing/flomart_app_execution/Chienti/app_flomart_manager_nwp_lami-2i_realtime_CHIENTI.sh
```

Figure B.2: Installation of Flomart app in the crontab of server Marche Region "hydro".