CASCADE

A toolbox
for network-scale
sediment connectivity
assessment

User manual for MATLAB

Marco Tangi
Rafael Schmitt
Simone Bizzi

POLITECNICO
MILANO 1863

Natural Resources
Management

CONTENTS

1 INTRODUCTION

About CASCADE

CASCADE (CAtchment Sediment Connectivity And DElivery) is a modelling framework developed by (Schmitt et al., 2016, 2018) that combines graph theory and sediment transport modelling to provide analysis on the sediment connectivity at the river network scale. It allows the quantification of disaggregated information about provenance and destination of single sediment loads, by simulating how an individual transport process proceeds downstream, depositing part of its load and interacting with the river reaches and other sediment loads.

CASCADE was developed to provide a computationally efficient and highly flexible tool for estimating sediment transport processes on the whole river network, and thus may be employed for the analysis of impacts on sediment connectivity across the whole river basin caused by the introduction of alterations in the river network, such as human infrastructures, land use change and others. By developing proper indicators of sediment connectivity alteration from the model outputs, CASCADE can be an effective tool for multi-objective river system planning and management.

About the toolbox

CASCADE_toolbox was developed to provide to the users the basic CASCADE model. It contains the tools required to understand and apply the model to their case study, and to eventually modify and add features to suit different needs. The toolbox is envisioned for researcher with at least basic knowledge of fluvial geomorphology and river system modeling. To use the toolbox, skills on how to operate within the MATLAB™ environment are also required. All the functions and scripts display comments and suggestions, to ease the comprehension of the model functioning and to encourage the customization of the code. The CASCADE toolbox can be downloaded for the *CASCADE model* website. The folder *cascademodel-master* contains all the functions described in this user guide.

About the manual

This manual is conceived as a support document for user approaching the CASCADE modeling framework. It contains a brief introduction to the toolbox folder structure and the description of the main principles and required inputs behind the scripts and functions of the toolbox, as well as hints on how to customize the functions and add new ones. After brief introduction on the structure of the toolbox (Section 2), the manual proceeds to describe the main functions and script in each of the CASCADE toolbox main steps,i.e. the river network extraction (Section 3), the CASCADE modeling framework (Section 4) and the output visualization (Section 5).

Requirements

For the toolbox to work, it is required MATLAB™ R2017b or higher, equipped with the Text Analytics Toolbox™ and the Bioinformatics Toolbox™ . For the extraction of the river network graph from a DEM, the toolbox requires the MATLAB functions for topographic analysis from *TopoToolbox* (Schwanghart and Scherler, 2013, 2014). *TopoToolbox* is free and open source, and can be downloaded from the *TopoToolbox* website or the GitHub repository. CASCADE has been tested with *TopoToolbox 2.2*, and it is not guaranteed to work with other releases.

2 TOOLBOX STRUCTURE

Figure 2.1 shows how the CASCADE toolbox is organized into folders. The application of the CASCADE model on a case study requires three basic steps:

1. Extraction of the river network from a DEM, identification of the features of each river network reach, and definition of the reach struct *ReachData*, the main CASCADE input;
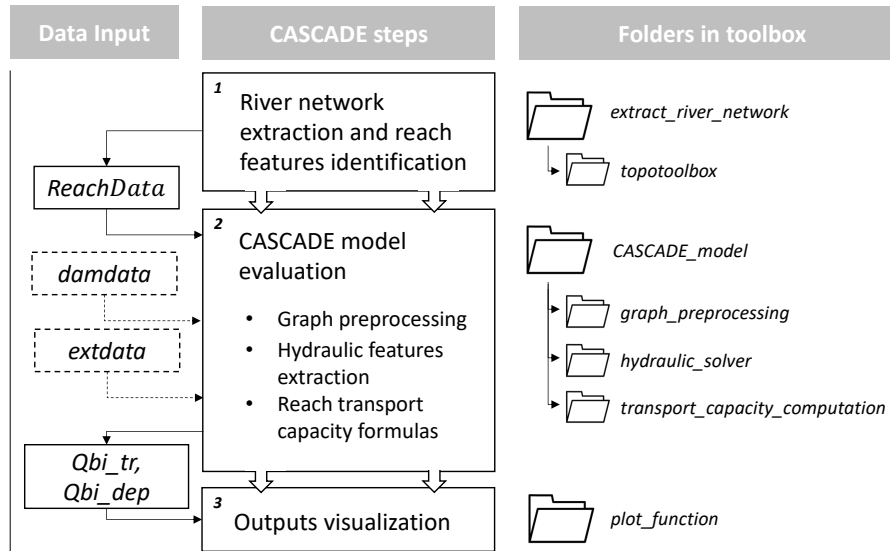
Figure 2.1: Graphical representation of the CASCADE model main steps and the folders containing their relative functions

2. evaluation of the sediment fluxes in the network with the CASCADE model;

3. visualization and interpretation of the model outputs.

The toolbox is structured in a way to easily identify in which folder the functions relative to each step are located. The function in each step are described in different sections of this manual: step 1 in section 3, step 2 in section 4 and step 3 in section 5;

2.1 Example dataset

An example input dataset has been included in the toolbox, to help the user to familiarize with the model and its features and how the model behaves in different case study. The dataset refers to the Vjosa river network, a gravel bed river in southern Albania. The two scripts *script_network_extraction* (Section 3.4) and *main_script_CASCADE* (Section 4.3), included in the toolbox, contain the operations to run the model on the Vjosa river, and are designed for first time user to familiarize with the CASCADE input and functions. Both scripts also contains additional code, left commented, that can be used to perform more complex operations and is designed for user exploring the different options customization options available in the toolbox, to apply it on their own case study.

3 RIVER NETWORK EXTRACTION

The functions in folder *extract_river_network* are designed to extract the river network from an input Digital Elevation Model (DEM). The resulting river network, stored as *ReachData* struct, is the main input to the CASCADE model. The output *ReachData* can be extracted as a river network shapefile, to allow visualization and manipulation in GIS environment.

The main function, *ExtractRiverNetwork.mat*, employs the Topotoolbox functions to extract the network, while the script **script_network_extraction.m** shows how to define the the inputs for the function.

3.1 Preliminary operations

To extract the network, CASCADE toolbox uses both functions from *TopoToolbox* and custom functions derived from *TopoToolbox*, stored in the *network_extraction_function* folder. The *topotoolbox_master* folder, freely available for the download from the *TopoToolbox* website, should be placed in the folder *extract_river_network* before proceeding.

3.2 Network extraction input

Hereby, we define in detail the main inputs for the *ExtractRiverNetwork* function.

The main input of the function is a **DEM** of the basin area, defined as a topotoolbox object *GRIDobj*. Topotoolbox function *GRIDobj('DEM_name.tif')* load the DEM into the workspace. The DEM must be located into the same folder as the script and must be in *.tif* format, as a GRIDobj; *script_network_extraction* showcases how the function is called. In case the DEM includes the drainage basins of more than on river, the function will extract only the largest stream network among the ones present in the area covered by the map. Models with poor spatial or elevation accuracy, or referring to regions with large flat areas may result in river networks which present incorrect stream geometry, non-existing river reaches and poor accuracy in the reach slope estimation.

The minimum drainage area (Amin) parameter defines the minimum drainage area in Km^2 required for a cell to be considered part fo the stream network. Overall, this parameter controls the size of the river network, that is the number of streams that are going to be included in the network and thus the network complexity: larger domains requires more input data but will capture smaller streams. The value must be converted into DEM cell area before the insertion the function; as shown in *script_network_extraction*.

The reach partition strategy defines how the network is split into reaches. The simplest strategy consists in setting the reach length as constant, resulting in an quasi-uniform partition. Since the network is partitioned considering the river branches natural length and the river confluences, the resulting reaches will not measure exactly as the chosen values: the algorithm extracts the reaches considering the parameter as an approximate upper limit to the reach length. Setting a large value to this parameter will results in reach partitioning only in confluences.

An alternative strategy is to perform a manual partition of the network by defining reach breakpoints(lines 34 to 38). The breakpoints are imported as an Bx2 matrix (breaknodes), where *B* is the number of points; each rows contains the X and Y coordinates of the breakpoints. The coordinates must be in the same reference system as the DEM and as close as possible to the river network, it is thus suggested to identify them directly onto the DEM in GIS environment. To avoid automatic partitioning, the *reach_length_km* parameter must be set to a very large value. The network will still be partitioned automatically at confluences. The function *ExtractRiverNetwork* automatically snaps the breakpoints to the nearest point in the extracted network before proceeding with the partitioning. An example breakpoint dataset for the Vjosa case study is given in workspace *Vjosa_breaknodes*. Manual partitioning of the network according to the geomorphological changes in the river, such as, for example, alterations in slope, width and bed material, may benefit the model results by allowing a more distinct characterization of the reach features and thus leading to a more refined identification of the changes in the sediment transport regime.

Both strategies can also be implemented together: the network can be uniformly partitioned with the addition of user-selected breakpoints in specific locations. For example, if dam, barriers or other point of alteration of the network sediment connectivity are present (e.g. locations of sediment delivery to the river), it is suggested to insert its locations as breakpoints, since the CASCADE algorithm requires to attribute a node to each of these features.

Finally, the user has the option to define **the minimum slope of the reaches** (mingradient). This parameter avoids the underestimation of the reach slopes in backwater areas where the slope of the water surface, extracted from the DEM, is less then the slope of the riverbed
The output *ReachData* contains the fundamental data for running CASCADE.

3.3 function *ExtractRiverNetwork*

```
[MS,S] = ExtractRiverNetwork(DEM, Amin, reach_length_km, breaknodes, mingradient)
```

ExtractRiverNetwork.m extracts the river network of a basin or sub-basin based on the elevations (DEM) and computes attribute table of reaches.

Input:

- DEM : GRIDobj (Topotoolbox object), DEM of the basin area, defined from a DEM via Topotoolbox function (see *script_network_extraction.m*) ;
- Amin : minimum drainage area [DEM cells];
- reach_length_km : approximate desired length of stream reaches [km];
- breaknodes : matrix *Bx2* of the (x,y) coordinates of the *B* breakpoints;
- mingradient : set value of minimum gradient value of the reaches [m/m].

Output:

- ReachData : map struct containing the geometry structure and features for the network reaches;
- P : STREAMobj (Topotoolbox object), required to plot the river network.

ExtractRiverNetwork.m extracts the river network and the reaches features using *TopoToolbox*. The function first creates a flow direction object from the DEM, then quantifies the cells flow accumulation and from this information extract the all the river networks in the covered area, according to the minimum drainage area set by the user. Finally, only the largest stream network in the DEM is kept. From the resulting network, the function attributes the reach and nodes ID, the channel length, the node elevation, the channel slope and the drainage area. Since elevation values along stream networks are frequently affected by scatter, often as a result of data artifacts or errors in the DEM, the function uses the value specified in *mingradient* as a lower boundary to the reach slope.

The function returns the **ReachData** struct, which contains the ID, geometrical shape and features of the network reaches, and the *STREAMobj* **P**, that encapsulates the information on geometry and connectivity of a stream network based on the flow direction of a digital elevation model and a logical raster that indicates the position of streams. P is useful as it can be readily plotted using the *plot* function in Matlab.

3.4 Network extraction script

Script **script_network_extraction.m** contains all the operations needed to initialize the inputs to function *ExtractRiverNetwork.m*, and to structure the function outputs for the use in CASCADE. By running the script as it is, the user will obtain as output the *ReachData* struct for the Vjosa river case study. The uncommented code serves the purpose of extracting the Vjosa river network from the DEM, which is reported as a GRIDobj in the workspace *Vjosa_DEM_GRIDobj.mat*, employing the breakpoints in workspace *Vjosa_breaknodes.mat* to perform a manual partitioning (see section 3.3 from more informations). Both dataset are placed in folder *Vjosa_extraction_data*

In the code, all the input variables are explicitly defined in order to be easily identified and changed. The script also contains code left commented, that is designed for user learning the different possible settings of the model. Section *DEM input* contains the operations to extract the GRIDobj from a DEM saved in the same folder as a *.tif* file. Section *reach partition input* contains the code to perform either the manual or uniform partitioning. The function *shapewrite* in section **export map struct** exports ReachData as a shapefile (format *.dbf, .shp, .shx*), that can be imported and manipulated in any GIS environment.

4 CASCADE MODEL EVALUATION

The second and most important step in the toolbox is the evaluation of the sediment transport and connectivity at the network scale via the CASCADE model. The main function *CASCADE_model* contains the operations necessary for the evaluation of the sediment fluxes and sediment connectivity within the river network. The function presents also an user interface with dialog boxes to guide in the selection of the different options of the model.

4.1 CASCADE input

The input of CASCADE are described in the two section below. The mandatory input are the **ReachData** struct and the **Network** struct. Moreover, CASCADE toolbox allow the user to access to different modeling settings and provide the framework with additional input data. The available choices are:

- 4 different **transport capacity formulas**, which provide an estimate of the transport capacity for each considered sediment class in each reach(table 4.2);
- 2 different methods to estimate the **hydraulic features** of the reach : Manning - Strickler formula and via optimization of the flow depth, given the discharge and active channel width (table 4.3);
- position and sediment trapping efficiency of dams and barriers in the network, via matrix **damdata**;
- external sediment flow entering the reach, via matrix **extdata** ;
- changes in the reach supply limitation, reported in the **tr_lim** column in ReachData.

4.1.1 *CASCADE main input*

In this section, the mandatory CASCADE input are described. All these input are necessary for running the model and therefore must be defined before proceeding with the model application:

The **ReachData** [N x 1] struct is the mathematical structure used by CASCADE to represent the river network as an interconnected tree graph composed by reaches delimit by nodes, and the main output of the network extraction operations described in section 3. For each of the N reaches, ReachData defines specific features that are used in the CASCADE framework to derive the sediment transport capacity. Figure 4.1 illustrates how a fully defined ReachData matrix appears.

The ReachData struct derived by the DEM (section 3) already contains some of the input features of CASCADE, while others must be defined by the user. Table 4.1 describes the field in *ReachData* as well as how to obtain the relevant data. The user must fill the field relative to the user-defined features before proceeding with the CASCADE implementation.

The scalar struct **Network** reports information on the network connectivity. In particular, it describes upstream and downstream paths and distances between nodes and defines the reach hierarchy, i.e. the maximum number of connected nodes located upstream the reach, that is used to determine the order in which each reach is processes in CASCADE. Function *graph_preprocessing*, located in the folder of the same name, extracts struct *Network* from *ReachData* (see section 4.4.1).

4.1.2 *CASCADE optional input*

In this section, the optional CASCADE input are described. These input can help the user to better characterize the river network by adding features new features like dams, sediment inputs and reach supply limitation, and personalize the model by defining the transport capacity formulas to be used.

The CASCADE framework contains 4 different functions for the computation of the reach transport capacity **sediment transport formulas**, listed in section 4.3. Employing different sediment transport function may lead to different transport capacity estimation and model outputs. With the exception of *Wilcock and Crowe*, all other equations return a single value of transport capacity given the geomophological and hydrological features of the reach. As the model requires instead separated values of transport capacity for each of the sediment classes, the value is then partitioned via the Wu - Molinas fractional transport rate

		ReachData struct fields
Provenance	Name	Description
Extracted from DEM via river network extraction functions	`Geometry`	Type of geometry of the reaches, set to *'Line'*.
	`X and Y`	X and Y coordinates of the point composing the reach line.
	`reach_id`	Identification number of the reach.
	`FromN`	Identification number of the upstream node of the reach (FromNode). [a]
	`ToN`	Identification number of the downstream node of the reach (ToNode). [b]
	`Length [m]`	Length of the reach.
	`el_FN and el_TN [m]`	Elevation reported in the DEM for the fromNode and ToNode, respectively.
	`Slope [m/m]`	Slope of the channel, derived from the formula $((el\_FN)-(el\_TN))/Length$.
	`x_FN and y_FN`	X and Y coordinates of the FromNode.
	`x_TN and y_TN`	X and Y coordinates of the ToNode.
	`Ad [Km`^{2}`]`	Area of the basin drained by each of the reaches ToNode.
	`directAd [Km`^{2}`]`	Incremental area drained directly by the reach.
	`StrO`	Strahler number of the reach in the network.
User defined	`Q[m`^{3}`/s]`	Discharge passing through the reach in the chosen discharge scenario.
	`Wac[m]`	Active channel width of the network, i.e. the width of the channel section where sediment entraining occurs give the considered water flow.
	`D16[m]`	D16 of the reach surface bed material.
	`D50[m]`	D50 of the reach surface bed material.
	`D84[m]`	D84 of the reach surface bed material.
	`n`	Manning's roughness coefficient for the bed material in the channel.
	`tr_limit`	transport limitation parameter (see section 4.1.2)

Table 4.1: Description and provenance of the attributes reported in the colum fields of the *ReachData* struct

a the function automatically attributes to FromNode the same ID of the reach
b the outlet reach presents the same ID for the upstream and downstream node

Figure 4.1: example of an ReachData struct.

computation (Molinas and Wu (2000); Wu et al. (2003)). The sediment transport functions are stored in folder *transport_capacity_computation* (see section 4.4.4).

The estimation of the **hydraulic parameters** necessary for the transport capacity computation, that is the average water velocity and flow depth, is performed using Manning-Strickler formula, incorporated directly into the main function *CASCADE_model*, or via an optimization of the water height given the reach features and discharge scenario. In the framework folder, the optimization functions are inside folder *hydraulic_solver*.

The presence of dams and barriers in the network is defined by matrix **damdata**. In the CASCADE framework presented in the toolbox, the effect of dams on sediment delivery is described only by the trapping efficiency of the structure for each of the grainsize classes defined in the model (see section 4.2 and table 4.5 for more informations). Reservoir sediment trapping is not yet explicitly modelled within the framework. Reservoir trap efficiencies can be derived from empirical models, such as the Brune or Brown curves, or more detailed models of hydrodynamics and sediment transport in a reservoir impoundment. *damdata* is a (DxC + 1) matrix, where each rows contains as first element the ID of the node where the barrier is placed. The other C elements on the row report the trapping efficiency (between 0 and 1) of the dam for each of the C sediment classes defined in the main function *CASCADE_model*. If the dam node is a junction node, the dam is considered sited after the confluence, and thus both the upstream reaches are affected by the barrier. Since the presence of dam reservoirs on river reaches is not explicitly accounted in the model, the deposition of sediment due to dam trapping is concentrated only in the reach directly upstream the dam.

If the DEM from which the network is extracted already accounts for the dam reservoir, the value of the slope of the upstream reach may be affected; in this case, the effects of the reservoir trapping may appear in the model output without explicitly including the barrier in the *damdata* matrix. In fact, the sediment transport capacity in the reach with the reservoir may be very low due to the reduced slope, causing an extensive sediment deposition and consequential sediment disconnectivity.

Matrix **extdata** defines the additional external sediment flows that are brought into the reach from areas outside the river network, e.g. hillslopes and river banks. These fluxes may represents the sediment contribution from the uplands, as well as sediment inputs from bluffs, ravines, debris flows, landslides or anthropic

Transport capacity formula	
Number ID	Name
1	Wilcock and Crowe (Wilcock and Crowe, 2003)
2	Engelung and Hansen (Engelund and Hansen, 1967)
3	Yang (Yang, 1984)
4	Wong and Parker (Wong and Parker, 2006)

Table 4.2: Formulas for transport capacity included in CASCADE and relative ID

Hydraulic features estimation	
Number ID	Name
1	Manning - Strickler (Manning et al., 1890)
2	Hydraulic solver

Table 4.3: Methods for estimating the hydraulic features of the reach included in CASCADE and relative ID

activities. These sediment fluxes are defined by a sediment load and a grain size distribution; and each reach can receive multiple external sediment fluxes.

extdata is a (Mx1) struct with 5 fields. Each rows refers to a single sediment load, and the fields refers to 1) the ID of the input reach, 2) the instantaneous sediment flow, in Kg/s, and 3-5) the D16, D50 and D84 of the sediment load, in meter. The grain size distribution of the sediment flow is then converted to a grain size distribution via function *GSDcurvefit*, described in section 4.4.2.

Field **tr_limit** in ReachData allows the user to limit for each reach the potential for sediment supply to the network from the river bed. For each reach, it contains a value between 0 to 1, ranging from a reach where no sediment can be entrained by local bed erosion, to a reach in which local erosion can be equal to the total transport capacity.

Due to the simplicity of this method, it is suggested to set these parameters to either 1 (default case), or 0. In the latter case, since no sediment can be entrained from any reaches, in order to guarantee some degree of sediment supply to the network, it is recommended to either provide the network with external sediment fluxes or to leave the *tr_limit* parameter of the source reaches to 1, assuming it represents the sediment contribution from the area drained by the sources.

4.2 function *CASCADE_model*

```
[ Qbi_tr , Qbi_dep, QB_tr, QB_dep , Fi_r] = CASCADE_model( ReachData , Network )
```

CASCADE_model.m runs the CASCADE model for the input river network.

Input:

- ReachData: *Nx1* struct defining the features of the *N* network reaches;
- Network: Scalar struct containing information on the network connectivity for each reach;

Optional input:

- 'default': sets transport capacity and hydraulic calculator to default values, does not open the dialog windows;
- damdata: *D x (C+1)* matrix containing, for the *D* dams in the network, the node ID and the trapping efficiency for each of the *C* sediment classes;
- extdata : *Sx3* vector containing., for the *S* external sediment fluxes, the 1) input reach ID; 2) sediment load [Kg/s], 3) D50 of the load;

Output:

- Qbi_tr : Cubic matrix *NxNxC* composed by *C* elements (the sediment classes) for the *N* reaches. In each cell *(Nx, Ny, Cz)*, it contains the sediment flux brought to node *Ny* by the sub-cascade originated in node *Nx*, of sediment class *Cz*;
- Qbi_dep: structured as Qbi_tr, reports information on deposited sediment per sediment class. In each cell *(Nx, Ny, Cz)*, it contains the sediment flux deposited in node *Ny* by the sub-cascade originated in node *Nx*, of sediment class *Cz*;
- QB_tr: *NxN* matrix. It reports the total fluxes transported by each CASCADE to each node. Obtained by summing up the sub-cascades in Qbi_tr;
- QB_tr: *NxN* matrix. It reports the total fluxes deposited by each CASCADE in each node. Obtained by summing up the sub-cascades in Qbi_dep;
- Fi_r : *NxC* matrix reporting the grain size frequency of the *N* reaches for the *C* sediment classes, given as output of function *Fi_extraction* (see section 4.4.2).

CASCADE_model is the main function of the CASCADE framework. It contains the code to run the model on the river network and save the model outputs. The function allows the user to customize the CASCADE run by choosing among different type of modeling equations and by loading the additional inputs described in section 4.1.2 into the framework. It also presents different ways to set the model parameters and import the optional inputs: with an interactive framework for first time users, by providing the additional data as arguments to the function or by running the model with the default settings:

1. `[ Qbi_tr , Qbi_dep, QB_tr, QB_dep , Fi_r] = CASCADE_model( ReachData , Network )` activates the interactive interface, that guides the user to the choice of the model equations and additional input insertion via dialog boxes. If dams or external sediment fluxes are present, the interface will ask to select the *.mat* file containing the desired input; if the workspace contains more variables, if will select the first one.

2. `[ _] = CASCADE_model( _,'default' )` runs CASCADE with the default settings, defined in section *Default settings* in the function;

3. `[ _] = CASCADE_model( _, Name, Value)` allows the user to insert specific optional input using one or more name-value pair argument, listed in table 4.4.

Name	Input Value
'tr_cap_equation'	Number ID of the transport capacity equation (table 4.2)
'hydr_estimation'	Number ID of the hydraulic feature estimation method (table 4.3)
'dams'	*damdata* matrix
'external_sed _flow'	*extdata* matrix
'Fi_r'	Fi_r matrix, from previous CASCADE run. Avoid the re-extraction of the grain size frequency in each re-run.

Table 4.4: List of valid name-pair values arguments in CASCADE_model.

D (φ)	-9.5 , -8.5 , -7.5 , -6.5	-5.5 , -4.5 , -3.5 , -2.5 , -1.5	-0.5 , 0.5 , 1.5 , 2.5 , 3.5	4.5 , 5.5 , 6.5 , 7.5
Sediment type	Boulder	Gravel	Sand	Silt

Table 4.5: Diameter of the 18 sediment classes defined in *CASCADE_model*, and the sediment type they belong, in Krumbein (φ) scale.

The function automatically defines the number and amplitude of the *C sediment classes considered* in section *define sediment classes* of the function. Table 4.5 shows the default sediment classification, with 18 classes defined in Krumbein (φ) logarithmic scale (Krumbein and Sloss, 1963), with an amplitude of *1 φ*.

The outputs of CASCADE can be precessed to obtain numerical results describing the major sediment fluxes in the network. Table 4.6 shows the major processed outputs as well as the MATLAB code to obtain them.

4.3 CASCADE main script

The script *main_script_CASCADE* prepares the necessary input for running the CASCADE framework, either by directly defining them or by loading them into the MATLAB environment, and launches the main CASCADE function. By running the script as it is, the user will launch CASCADE on the *ReachData* struct contained in workspace *Vjosa_ReachData*, relative to the Vjosa river network. The example dataset data on the Vjosa river are placed in folder *Vjosa_model_data*. Note that the *ReachData* struct is different to the *ReachData* obtained as output to the *script_network_extraction* script, as it has been extracted from a different DEM and modified in GIS to better represent the river network.

The function *CASCADE_model* is set by default to be run in interactive mode (see section 4.2), allowing the user to choose among the different customization options and eventually upload workspaces *Vjosa_damdata* and *Vjosa_extdata*, found in folder *Vjosa_model_data* and containing respectively the *extdata* and *damdata* matrices relative the Vjosa river. These matrices' purpose is just to help the user familiarize with the structure of these input, and are by no means designed to represent the reality of a case study or to be used in actual sediment alteration analysis. Once the CASCADE outputs have been produced, the script will automatically open *interactive_plot* (see section 5.4) to visualize the results.

In this script are also included sections with commented code to run CASCADE with different customization options and settings. In particular, since ReachData can be exported as a shapefile by *script_network_extraction.m* , *main_script_CASCADE* uses the function *shaperead* to load ReachData back into the MATLAB workspace.

Moreover, section *Run Cascade for a different discharge scenario* includes the code for running the CASCADE model for a different water discharge scenario. To do so, the code employs the *Vjosa_water_discharge_scenarios* workspace in the folder *Vjosa_model_data*, reporting for each reach in ReachData different values of water discharge and active channel width (in matrix *Vjosa_Q_scenario* and *Vjosa_Wac_scenario* respectively). Matrix *Vjosa_scenario_frequency*, also included in the workspace, reports the annual frequency of each scenario.

Name	MATLAB code	Description
Total sediment transported	`sum(QB_tr)`	N vector of the total sediment transported in reach N by all the cascades passing through.
Total sediment deposited	`sum(QB_dep)`	N vector of the total sediment deposited in reach N by all the cascades passing through.
Total sediment entrained	`diag(QB_tr)`	N vector of the total sediment entrained in reach N by the newly formed cascade.
Sediment transported per class	`nansum(Qbi_tr(:,:,C))`	N vector of sediment transported from all cascades in reach N, for sediment class C (according to the order seen in table 4.5).
Sediment deposited per class	`nansum( Qbi_tr(:,:,C))`	N vector of sediment deposited from all cascades in reach N, for sediment class C.
Sediment transported per multiple classes	`sum(nansum(Qbi_tr(:,:, C1:C2)),3)`	N vector of sediment transported from all cascades in reach N, for sediment classes from C_1 to C_2.
Sediment deposited per multiple classes	`sum(nansum(Qbi_tr(:,:, C1:C2)),3)`	N vector of sediment deposited from all cascades in reach N, for sediment classes from C_1 to C_2.

Table 4.6: List of vector containing sediment connectivity informations that can be obtained from the CASCADE outputs, with their relative description and MATLAB code.

4.4 Auxiliary functions

4.4.1 *graph_preprocessing*

```
[ Network ] = graph_preprocessing( ReachData )
```

graph_preprocessing.m uses the node ID of the reaches to return the struct Network, that describes network connectivity.

Input:

- `ReachData`: *Nx1* struct defining the features of the *N* network reaches.

Output:

- `Network`: Scalar struct containing information on the network connectivity for each reach;

Function *graph_preprocessing.m*, located in the folder of the same name, uses reach FromNode, ToNode and Length to define network connectivity, distance and pathways between each two reaches in the network, and extracts the node hierarchy, which is used in *CASCADE_model* to establish which reach is processed first based on how far upstream in the network it is located. Function *write_adj_matrix*, in the same folder, is used to create a sparse adjacency matrix of the network nodes, a pre-requisite to derive the *Network* struct.

4.4.2 *GSDcurvefit*

```
[ Fi_r ] = GSDcurvefit( D16, D50, D84 )
```

GSDcurvefit.m returns the grain size frequency for the input sediment classes based on the D50, using the Rosin distribution (Shih and Komar, 1990).

Input:

- D16: *1xN* vector defining the D16 of the N input reaches [m];
- D16: *1xN* vector defining the D50 of the N input reaches [m];
- D50: *1xN* vector defining the D84 of the N input reaches [m].

Output:

- Fi_r: *NxC* vector of grain size frequency of the *N* reaches for the *C* sediment classes defined;

Function *GSDcurvefit* converts the river bed D16, D50, D84 of the reaches into a grain size frequency distribution for the *C* classes defined in the main function, used in the computation of the sediment transport capacity for each class. In order to do so, it fits the input data to a Rosin distribution curve (Shih and Komar, 1990), defined as:

$$F(\leqslant D) = 1 - \exp[-(D/k)^s] \tag{1}$$

The Rosin distribution is a 2-parameters cumulative distribution function, where *k* is the mode of the GSD and *s* is an inverse measure of the curve spread around the D50.

4.4.3 *hydraulic_solver*

```
[ hydraulicData ] = hydraulic_solver( Slope, Q, Wac, plot )
```

hydraulic_solver.m finds the hydraulic features of the reach via optimization.

Input:

- Slope: *Nx1* vector of the slope of the *N* network reaches;
- Q: *Nx1* vector of the water flow of the *N* network reaches, in m^3/s;
- Wac: *Nx1* vector of the active channel width of the *N* network reaches, in m.

Optional input:

- plot: if plot $= 1$, plots details on the hydraulic features of the reaches and optimization results.

Output:

- hydraulicData: vector of hydraulic variables of the reach.

Function *hydraulic_solver*, located in the folder of the same name, is used to derive the hydraulic parameters of the network reaches via optimization. For each reach, it minimizes the objective function, i.e. the difference between observed and estimated discharge, by changing the flow depth *h* value. The objective function is computed by function *hydraulic_solver_objective.m*, that solves the open channel flow equations and calculates grain size given a bankfull discharge for a given water height to estimate the discharge and compare it to the observed value.

4.4.4 *transport_capacity_computation*

Folder *transport_capacity_computation* contains the functions to calculate sediment transport capacities using the different formulas described in 4.2, given the reach hydraulic and geomorphologic features. For each of the formulas in table 4.2, the framework contains a different function which receives as input the features of the reach, the sediment distribution obtained by function *Fi_extraction* and the sediment class vector *psi*, and returns the instantaneous mass transport rate for each class *Qtr_cap*.

```
[ Qtr_cap ] = formula_name_tr_cap( Fi_r_reach, psi, reach_features )
```

The functions of type *equation_name_tr_cap* find the instantaneous mass fractional transport rate in the reach given the reach hydrological and geomorphological features. Each function may require different reach features as input.

Input:

- Fi_r_reach: *1xC* vector of grain size frequency values for the *C* sediment classes fot he considered reach;
- *reach_features*: different for each formula; geomorphologic and hydrological features of the reach, defined in *CASCADE_model* and extracted from *ReachData*.

Output:

- Qtr_cap: *1xC* vector, reporting the instantaneous mass transport rate for each of the *C* sediment classes, in Kg/s.

As stated before, *Engelund & Hansen*, *Yang* and *Wong & Parker* formulas both returns a single value of transport capacity for the given considered reach; to obtain a fractional transport rate for each sediment class, the total transport capacity is split between classes using the Molinas formulas. This is accomplished via the function *Molinas_rates*, which returns the coefficient of fractional transport rates for each class *Pci*.

The basic structure of the transport capacity functions can be replicated with different formulas if the user wish to implement a different transport capacity equation. However, for the model to work the output of said function must still be the matrix *Qtr_cap*.

5 OUTPUTS VISUALIZATION FUNCTIONS

The CASCADE toolbox provides the user with several functions which produce plots to visualize the outputs and aid in the understanding of the model results. These functions are contained in folder *plot_function*.

5.1 *plot_long_profile*

```
plot_long_profile ( ReachData , Network, n_branches)
```

Function *plot_long_profile* plots the river long profile, i.e. the changes in altitude of the river network from the sources to the outlet.

Input:

- ReachData: *Nx1* struct defining the features of the *N* network reaches;
- Network: Scalar struct containing information on the network connectivity for each reach;

Optional input:

- n_branches : scalar number defining the number of river branches to be highlighted in the figure. The figure will display the longest *n* river branches in different color and increased line width (n*n* is the value of *n_branches*); the remaining branches will appear as thin, light gray lines. If not specified, all branches will be colored.

Function *plot_long_profile* can be used to analyze the extracted network to gain insight on the geomorphological processes occurring in the network like knickpoints (Pederson and Tressler, 2012) or to identify outliers in the *ReachData* matrix before running CASCADE. The resulting plot distinguish with different colors the river branches, extracted from ReachData by function *river_branches_finder*. A river branch is defined as a set of connected reaches sited between a source and a confluence or a source and the outlet. As a consequence of this definition, a river network have the same number of sources reaches and branches.

Name	Input Value
'ShowID'	if 'on', displays the reach ID on the reach midpoint
'cMap'	name of color map used in the figure
'legend_type'	if 'colorbar', displays data with continuous colorbar, instead of percentiles color classes
'title'	figure title
'Linewidth'	width of the reach lines, either a single value or a Nx1 vector of reach-specific width
'ClassNumber'	number of percentile color classes used for displaying plotvariable in the plot.

Table 5.1: List of valid name-pair values arguments in *plot_network*.

5.2 *plot_network*

```
plot_network ( ReachData, plotvariable)
```

Function *plot_network* plots the river network and visualized continuous data on the network.

Input:

- ReachData: *Nx1* struct defining the features of the *N* network reaches;
- plotvariable: *Nx1* vector containing, for each reach, the continuous data to be displayed on the network.

plot_network plots the river network using the spatial coordinates contained in attribute *ReachData.X* and *ReachData.Y*. If *plotvariable* is given as input, the reaches are allocated in different percentile color classes according to the values in the input vector and to each class is attributed a distinct color of a defined colormap. The function can also be provided with additional inputs that allow for further customization of the visual output:

1. plot_network (ReachData) plots an 'empty' network, with the same color for all reaches;

2. plot_network (ReachData, attribute) plots the network with the attribute specified in *attribute* vector;

3. plot_network (_, Name, Value) modifies properties of the figure using one or more name-value pair arguments, described in table 5.1;

4. f = plot_network (_) : returns a *Figure* object. Use f to query or modify properties of the figure after it is created.

Since CASCADE provides for each reach in the network outputs on the different sediment fluxes, *plot_network* can be used to visually represents these on the network and thus to help identify areas of interest for sediment connectivity in the network. Section *default setting* in the function contains the default settings employed if the relative name-pair values arguments are not given.

5.3 *reach_subplot*

```
reach_subplot (reach_ID, QB_tr, Qbi_tr, Qbi_dep , Fi_r, ReachData ,Network )
```

Function *reach_subplot* plots specific information on the features and sediment connectivity fluxes on a user-defined reach

Input:

- `reach_ID`: scalar number corresponding to the reach ID to be visualized;
- `Qbi_tr` , `Qbi_dep`, `Fi_r` : CASCADE outputs;
- `ReachData`: *Nx1* struct defining the features of the *N* network reaches.
- `Network`: Scalar struct containing information on the network connectivity for each reach;

Function *reach_subplot* plots a figure with four different subplots, displaying: 1) the reach features, 2) the provenance ad magnitude of the total sediment flux passing through the reach, 3) the grain size distribution and 4) the deposited and entrained flows for each sediment class.

5.4 *interactive_connectivity_evaluation*

```
interactive_connectivity_evaluation (plotvariable, Qbi_tr, Qbi_dep, Fi_r, ReachData, Network,
graphtitle, damdata, extdata )
```

Function *interactive_connectivity_evaluation* plots the river network with a continuous data and allows the user to select a reach to display the reach features and sediment connectivity outputs of CASCADE.

Input:

- `plotvariable`: *Nx1* vector containing, for each reach, the continuous data to be displayed on the network;
- `Qbi_tr` , `Qbi_dep`, `Fi_r` : CASCADE outputs;
- `ReachData`: *Nx1* struct defining the features of the *N* network reaches.
- `Network`: Scalar struct containing information on the network connectivity for each reach

Optional input:

- `graphtitle`: title of the plot;
- `damdata`: CASCADE input. If given, plots position of dams on the network;
- `extdata`: CASCADE input. If given, plots input location of the external sediment fluxes on the network.

Function *interactive_connectivity_evaluation* supplies the user with a tool to visually explore the river network and read the CASCADE outputs for each reach, aiding in the identifications of sediment connectivity hotspots and the tracing of sediment fluxes provenance and type in each reach. The user can also decide to display the reach ID and the nodes in the network, and, if given as input, the dam siting and external sediment loads input points.

Furthermore, the visualization tool also allows for in-detail analysis of sediment connectivity for a specific reach. By activating the *manual reach identification* function, the user can manually select a reach on the network. Alternatively, the user can insert the reach ID in the dialog box by pressing the *reach selection via ID*

relative button. After selecting the reach, the visualization tool calls function *reach_subplot* for the reach and displays the subplots. By clicking again on the new figure, it closes and returns to the main figure, allowing the user to select a new reach or perform another operation.

5.5 *sediment_management_analysis*

```
sediment_management_analysis(ReachData ,Network, graphtitle, damdata, extdata )
```

Function *sediment_management_analysis* plots the river network with a continuous data and allows the user to activate or de-activate dams and external sediment contributions in the network and visualize the changes in sediment transport processes. All the features present in *interactive_plot* are also available.

Input:

- ReachData: *Nx1* struct defining the features of the *N* network reaches.
- Network: Scalar struct containing information on the network connectivity for each reach

Optional input:

- damdata: CASCADE input. If given, plots position of dams on the network and aa;
- extdata: CASCADE input. If given, plots position of input of the external sediment fluxes on the network.

Function *sediment_management_analysis* supplies the user with a tool to visually explore the alterations in sediment connectivity caused by the introduction or removal of external sediment fluxes and dams. To run the function, at least one input between *damdata* and *extdata* should be provided, otherwise the function will behave like *interactive_plot*.

The function allows the user to select which dams or external fluxes are present in the network via a list selection dialog box. Then, it re-runs CASCADE with the transport capacity and hydraulic parameters formulas specified in section *CASCADE settings* in the function, and including the active dams and external sources specified in the dialog box. All the features present in *interactive_plot* are kept, including the visualizations of the patterns of deposition and entraining and the possibility to select a reach and visualize the relative details.

REFERENCES

Engelund, F. and Hansen, E. (1967). A monograph on sediment transport in alluvial streams. *Technical University of Denmark ostervoldgade 10, Copenhagen K.*

Krumbein, W. C. and Sloss, L. L. (1963). Stratigraphy and sedimentation. Technical report.

Manning, R., Griffith, J. P., Pigot, T., and Vernon-Harcourt, L. F. (1890). *On the flow of water in open channels and pipes*.

Molinas, A. and Wu, B. (2000). Comparison of fractional bed-material load computation methods in sand-bed channels. *Earth Surface Processes and Landforms: The Journal of the British Geomorphological Research Group*, 25(10):1045–1068.

Pederson, J. L. and Tressler, C. (2012). Colorado river long-profile metrics, knickzones and their meaning. *Earth and Planetary Science Letters*, 345:171–179.

Schmitt, R., Bizzi, S., Castelletti, A., and Kondolf, G. (2018). Improved trade-offs of hydropower and sand connectivity by strategic dam planning in the mekong. *Nature Sustainability*, 1(2):96.

Schmitt, R. J., Bizzi, S., and Castelletti, A. (2016). Tracking multiple sediment cascades at the river network scale identifies controls and emerging patterns of sediment connectivity. *Water Resources Research*, 52(5):3941–3965.

Schwanghart, W. and Scherler, D. (2013). Topotoolbox 2–an efficient and user-friendly tool for earth surface sciences. *Earth Surface Dynamics Discussions*, 1:261–275.

Schwanghart, W. and Scherler, D. (2014). Topotoolbox 2–matlab-based software for topographic analysis and modeling in earth surface sciences. *Earth Surface Dynamics*, 2(1):1–7.

Shih, S.-M. and Komar, P. D. (1990). Differential bedload transport rates in a gravel-bed stream: A grain-size distribution approach. *Earth Surface Processes and Landforms*, 15(6):539–552.

Wilcock, P. R. and Crowe, J. C. (2003). Surface-based transport model for mixed-size sediment. *Journal of Hydraulic Engineering*, 129(2):120–128.

Wong, M. and Parker, G. (2006). Reanalysis and correction of bed-load relation of meyer-peter and müller using their own database. *Journal of Hydraulic Engineering*, 132(11):1159–1168.

Wu, B., Molinas, A., and Shu, A. (2003). Fractional transport of sediment mixtures. *International Journal of Sediment Research*, 18(3):232–247.

Yang, C. T. (1984). Unit stream power equation for gravel. *Journal of Hydraulic Engineering*, 110(12):1783–1797.