Code and data: mean length

This file provides some insights on how to deal with the code and the data to reproduce the results of the paper *"On the relationship between mean hillslope length and drainage density"*. The DTM data (for the computation of the empirical mean hillslope length) is contained in the folder named *Catchments*. Different MATLAB scripts are available at the same repository (together with the DTM input data and this readMe file), respectively named as *"p1_empiricalHillslopeLengthDistributions.m"*, *"p3_computeMeansAndPlot.m"* and *"p4_computeMeansGlobalError.m"*. The scripts must be placed in the same location of the folder *Catchments* to work properly ad they must be run in the order specified by their numbers (only exception: the second script *"p2_calibrateHyperbolicParam_AllBasins.m"* is not provided as it is related to the computation of hyperbolic hillslope length distributions which are not necessary for the mean hillslope length). The first script *"p1_empiricalHillslopeLengthDistributions.m"* performs the computation of the empirical hillslope length distributions from the DTM (for a single specified catchment); the computation of these empirical distributions represents the first step for the computation of the empirical mean length, thus the user is asked to run the script *"p1_empiricalHillslopeLengthDistributions.m"* first, as many time as the possible catchments are, by selecting the proper input catchment and specifying the input variables (see below). Then it'll be possible to run the other scripts: *"p3_computeMeansAndPlot.m"* computes the empirical mean length of the various catchments in a single run (starting from the previously computed distributions) and plot them; *"p4_computeMeansGlobalError.m"* provides the results related to the global error MRE related to each analytical model fo the mean length. Running the codes *"p3_computeMeansAndPlot.m"* and *"p4_computeMeansGlobalError.m"* does not require any specification by the user of the value of variables used by the code, so they will not be discussed in the detail in this file. Some details about their functioning are specified at the beginning of each script as a comment. Instead the user is asked to interact with the first section of the first script *"p1_empiricalHillslopeLengthDistributions.m"* (at least to specify which catchment to analyze during the run). For this reason, the following part of this file aims to give some hints on the functioning of the first script.

The code structure of *"p1_empiricalHillslopeLengthDistributions.m"* follows these three sections:

- Input specifications;
- Import data, morphology indexes and initialization;
- Network analysis and hillslope distributions.

0.1 Code inputs

Information about the morphology of the catchments is contained in the folder named Catchments. The code automatically imports the information about the DTM of the requested catchment. The user is asked to interact with the first section (named as Input specification), by properly setting the values of the following variables (at least to select the desired catchment to be analyzed):

- *report_iterations*, is a logical variable that should be set equal to *true* to receive a message in the command window about every network expansion performed by the code, *false* otherwise;
- *save_on_file*, is a logical variable that specifies whether the workspace with the variables should be saved at the end of the code and as a backup after a certain amount of iterations (*true*) or not (*false*). It is recommended to set it equal to *true*;
- *save_after_nIters*, is the integer variable specifying the amount of iterations after which the code will save the workspace variables as backup. The variables are saved at the end of the code regardless of the value of this variable;
- *catchment_id*, is the integer variable that must be set equal to a value in between 1 and 17 (according to the list provided in the code) to select the desired catchment;
- *restart*, is a logical value that specify if you want to continue an old analysis that you had already performed (*true*) or to start a completely new analysis (*false*). It is recommended to set it equal to *true* unless the analysis to perform is really time consuming. To continue an old analysis will automatically skip the computation of the morphometric indexes;
- *computeAllInfo*, should be set equal to *true* if you want to compute the morphometric indexes (as for instance the Gravelius coefficient), *false* otherwise;
- *preserve_eccentricity*, is a logical variable determining whether to continue to compute the elliptic eccentricity from a previously started (and not completed) attempt

(*true*), or not (*false*). It is recommended to set it equal to *false* since the computation is generally not heavy for headwater catchments.

The second and third sections of the code (*"Import data, morphology indexes and initialization"* and *"Network analysis and hillslope distributions"* respectively) don't require any variable specification by the user.

0.2 Code outputs

The results provided by the code are given in MATLAB '-mat' format and they are contained in a folder named as *Results_nameCatchment* which is automatically created by the code. Two different relevant '-mat' files will be created: the first one is named *totpixel.mat* and contains the value of the amount of pixels of the DTM which belong to the drainage area of the selected catchment outlet (pixel with the highest value of the contributing area in the DTM). The second one is named *Workspace_Final.mat* and it contains all the interesting output variables:

- *dx*, is the size of the pixel side in meters;
- *network_Lengths*, is an array such the i^{th} component corresponds to the length L of the channel network at the i^{th} iteration in units of pixels;
- *branching_Index*, is an array containing the value of a branching index along the iterations. The number of branches of all the spatial configurations of the channel network can be found as *num_branches = 1./(ones(size(branching_Index)) - branching_Index)*.
- *distributions_support*, is a matrix, the i^{th} column of which corresponds to the hillslope length ℓ values at which the hillslope distribution has been sampled (the hillslope distribution related to the i^{th} spatial configuration of the channel network);
- *distributions_w0*, is a matrix, the i^{th} column of which corresponds to the values of the hillslope length distribution conditioned to the fact that the river network has assumed its i^{th} spatial configuration (the hillslope distribution related to the i^{th} spatial configuration of the channel network, which means $\omega_e(\ell, L = L_i)$). The ℓ values at which $\omega_e(\ell, L = L_i)$ is evaluated are provided by the i^{th} column of the matrix distributions_support. Please, note that, despite its name, this variable is not the known initial condition ω_0 for the hyperbolic model, but rather the empirically observed function $\omega_e(\ell, L)$;

- *distributions_w*, is a matrix analogous to distributions_w0 and it is used by the code for the computation of the impact coefficient $\alpha(\ell, L)$. Please note that, despite its name, this variable does not consist in the hyperbolic model of the hillslope length distribution and it is not the empirically observed distribution $\omega_e(\ell, L)$;
- *distributions_alpha*, is a matrix, the i^{th} column of which corresponds to the values of the empirically observed impact coefficient conditioned to the fact that the river network has assumed its i^{th} spatial configuration (the i^{th} column consists in the function $\alpha(\ell, L = L_i)$). The ℓ values at which $\alpha(\ell, L = L_i)$ is evaluated are provided by the i^{th} column of the matrix distributions_support.

Please note that all the values given in the matrix *distributions_support* are in units of pixels, so the user should multiply them by $dx/1000$ to get the corresponding values in kilometers (and, in that case, the *distributions_w0* and *distributions_w* matrices should be multiplied by $1000/dx$ to preserve the unitary integral of hillslope length distributions). Partial results of the analysis obtained from intermediate backups of the code should not be considered. In particular, the *distributions_w0* matrix from intermediate backups should be multiplied by 2 to get properly normalized distributions.