User guide for *FluvialCorridor* toolbox

Spatial aggregation and longitudinal discontinuities location

Toolset name : *STATISTICS*

Tool’s name : *Hubert test*

How to cite : Roux, C., Vaudor, L., Alber, A., Piégay, H., 2013. Hubert test guideline for the *FluvialCorridor* toolbox, a new ArcGIS toolbox package for exploring multiscale riverscape at a network scale. Sedalp (Sediment Management in Alpin Basins) and CNRS (UMR5600).

FluvialCorridor package for ArcGIS

Version V01 - 2014

CNRS - UMR5600 Environnement Ville Société
Alpine Space Program - Sedalp

For each use of the *FluvialCorridor* GIS package leading to a publication, a report, a talk presentation or any other document, please refer to the following paper :

Roux, C., Alber, A., Bertrand, M., Vaudor, L., Piégay, H., submitted. "FluvialCorridor" : A new ArcGIS package for multiscale riverscape exploration. Geomorphology.

I. Concept and methods

One of the key step of the fluvial characterization defined by Alber and Piégay (2011) and involved within the *FluvialCorridor* toolbox is the spatial aggregation of DGO into homogeneous reaches along the fluvial continuum. These elementary DGO of a constant length must first be attributed according one or several metric in order to process the spatial aggregation.

The process involved into the *Hubert test* tool to implement the spatial aggregation is the statistical test of Hubert (Hubert, 2000 ; Kehagias and al., 2005) . It is an univariate statistical test based on the standard deviation, enabling to identify locations of breaks into a set of numerical data (Fig. 1A). The output set of data is modeled as succession of straight and constant lines (or "segments") ended by breaks (or discontinuities). These breaks must be located so that the difference between two consecutive segments is consistent.

The *Hubert test* tool can be run against every kind of table (i.e. table, shapefile of points, polylines or polygons). The only one required condition is that the input table must be sequenced and so populated with the ordination fields "*Order_ID*", "*Rank_UGO*" et "*Distance*". The output shape will be same than the input shape.

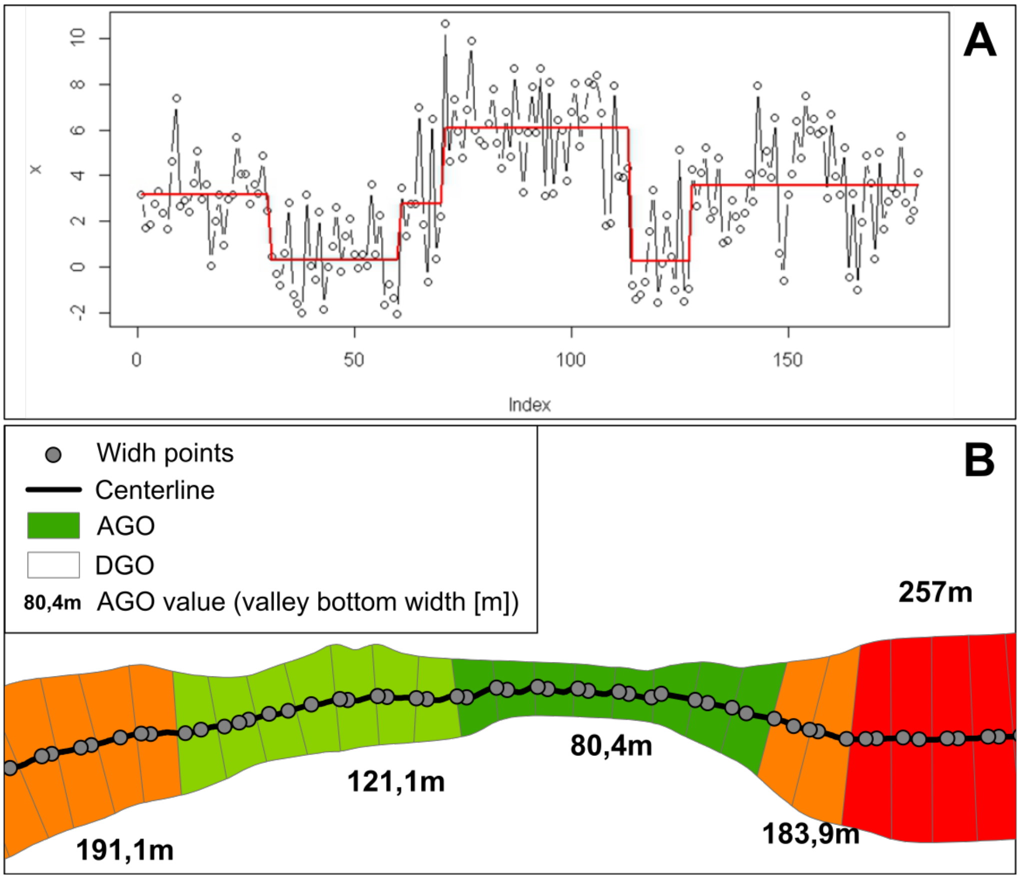


Figure 1 (A) : example of a numerical signal on which the Hubert test can be processed. (B) Result of an application of the *Hubert test* tool against a disaggregated valley bottom continuum attributed with the related valley bottom width.

Implementation of the Hubert test has been done with a GIS software (ArcGIS 10.0). It uses a Python script ("*def__Pack_Hubert*") including a main function and some annex function used to run the Hubert test. This script has been translated from an original R script developed by Lise Vaudor.

General algorithmic framework

The algorithmic scheme developed for the *Hubert test* tool is presented in the Fig. 2.

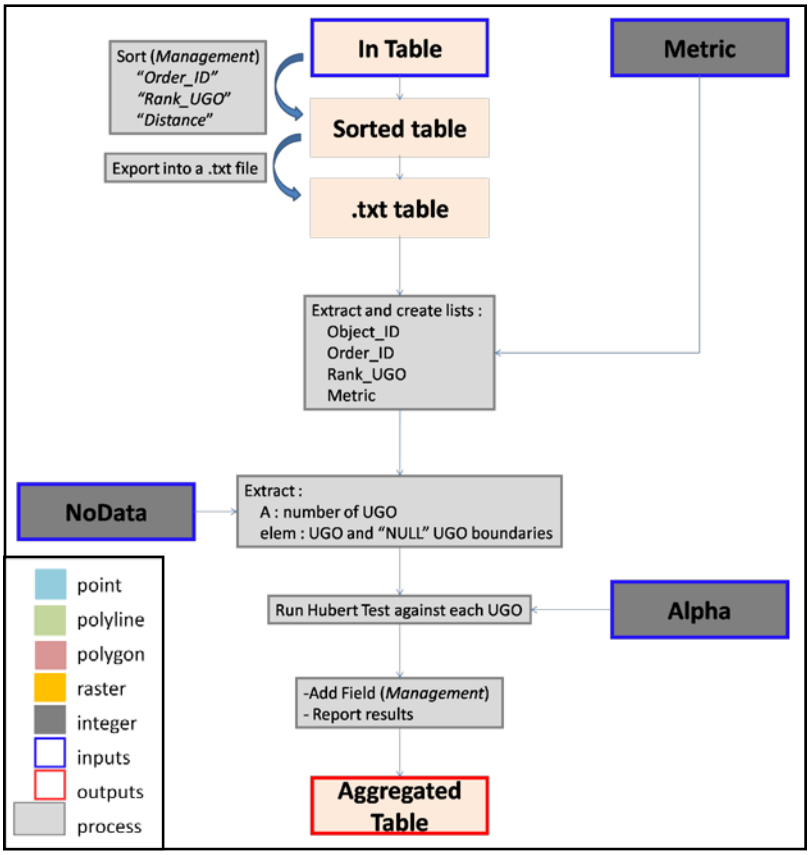


Figure 2 General algorithmic framework used into the *Hubert test* tool

In order to make the handling easier (formatting and running the Hubert test), the set of numerical data stored into the attribute table of the input entity is exported into a .txt file. Most of the treatments are done with Python, without the ArcPy library.

In a first hand, input table is sorted according the ordination fields "*Order_ID*", "*Rank_UGO*" and "*Distance*". Then, the table is exported into a .txt file. This file is saved within the folder which include the default geodatabase. If the user does not modify this geodatabase in the general environment proprieties, its path must looks like C:\Documents and Settings\<user>\My Documents\ArcGIS\Default.gdb.

Five variable are then created :

- a list named *ObjectID* : "*OBJECTID*" column of the sorted table ;
- a list named *Order*_ID : "*Order_ID*" column of the sorted table ;
- one list named *Rank_UGO* : "*Rank_UGO*" column of the sorted table ;
- one list named *Metric* : column related to the user given metric and against which the Hubert test will be run ;
- an integer name *NbUGO* : number of distinct UGO ;
- a list name *elem* : boundaries (in terms of "*OBJECTID*") of the different elements of the table. One element can be a set of values contained into a same UGO, a set of "*Null*" values (i.e. DGO with no metric value) or a set of NoData values (user-defined).

The Python script with the functions enabling to run the Hubert test is then ready to be launch. Results are stored into two lists :

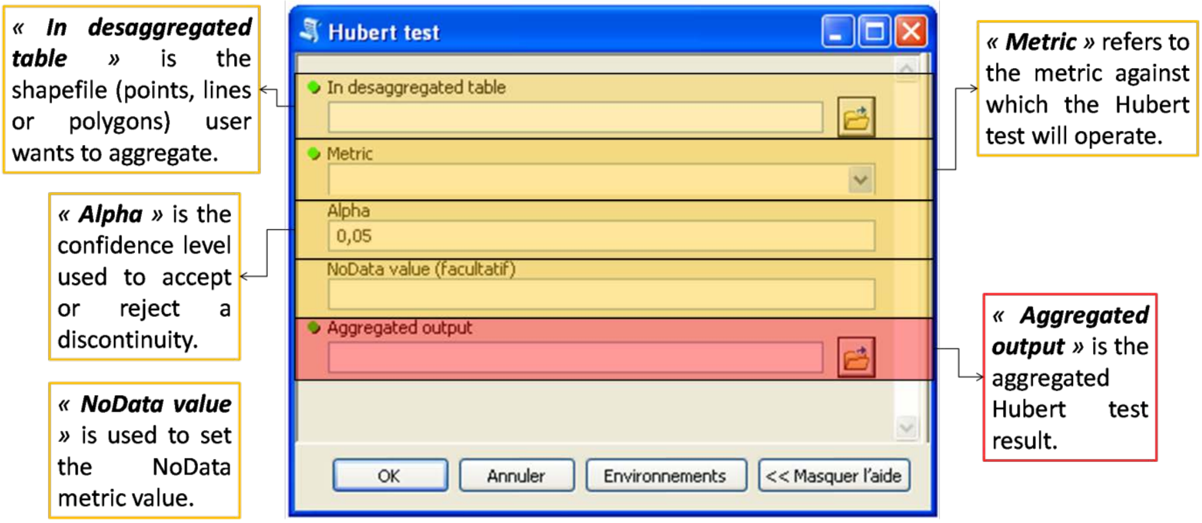
- *breaks* : for each element listed into *elem*, this list includes the locations of identified breaks and a information of the segmentation accuracy.
- *HReach* : each row of that list relates to a DGO of the input table. It includes the initial metric value of that DGO, an id related to the AGO it belong to and the AGO value estimated by the model.

The *Alpha* parameter is used during the Hubert test run. It can be viewed as a confidence level enabling to accept or reject a tested break.

II. Screen user interface

II.1. Startup screen

Into the screen user, several fields have to be filled (Fig. 3). Be careful that a green mark in front of a field is not a guaranty that this field is not optional. Into *Hubert test*, if a field is available, that means that it **<u>must be filled</u>**.



Hubert test

In desaggregated table

Metric

Alpha

0,05

NoData value (facultatif)

Aggregated output

« ***In desaggregated table*** » is the shapefile (points, lines or polygons) user wants to aggregate.

« ***Alpha*** » is the confidence level used to accept or reject a discontinuity.

« ***NoData value*** » is used to set the NoData metric value.

« ***Metric*** » refers to the metric against which the Hubert test will operate.

« ***Aggregated output*** » is the aggregated Hubert test result.

Figure 3 Screen user interface of the *Hubert test* tool

Use of the *Hubert test* tool is quite simple. User must give the disaggregated entity (i.e. set of DGO attributed with the ordination fields and with at least one or several metrics) he wants to aggregate, the metric he wants to use to assess the spatial aggregation and the confidence level in order to accept or reject a tested segmentation. During the process, and from upstream to downstream, a DGO or a set of DGOs with a NoData value will be considered as a unique AGO. The "*NoData value*" field is optional so that if it is not filled, only "*Null*" values will be considered as NoData.

Finally, user must filled the destination path of the output aggregated shapefile.

Note : *Alpha* default value

For the confidence level, a default value of 0,05 is proposed. It has been noted that such a value provided rather good results. However, user is free to modify this value :

- increase this value enable to decrease the segmentation significance so that more breaks and homogeneous reaches are identified ;
- conversely, decrease *Alpha* ensure a strong significance but decrease the number of resulting breaks and homogeneous reaches.

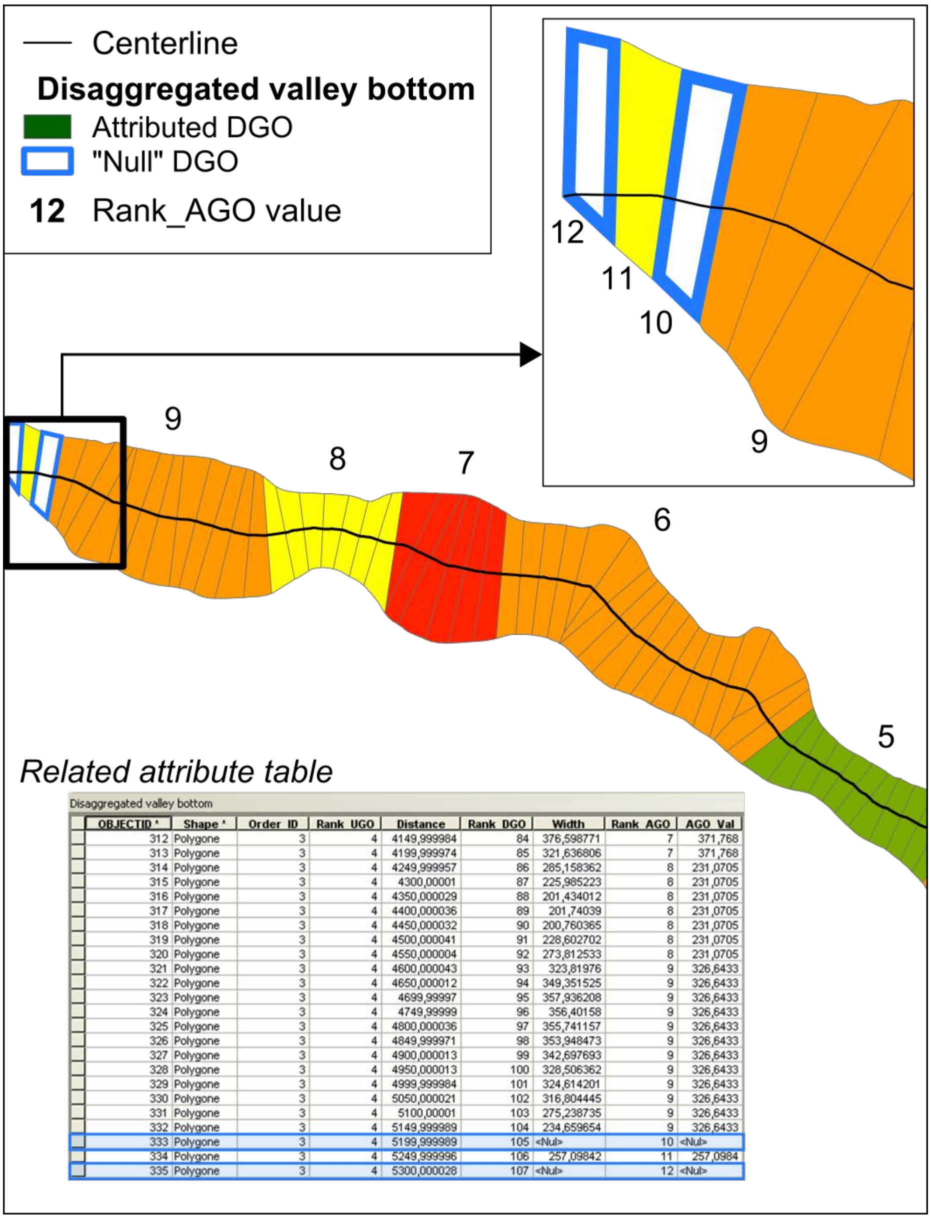
Note : Input data

Input entity must be disaggregated (*Disaggregation* tool). It must also include ordination fields "*Order_ID*", "*Rank_UGO*" and "*Distance*". These conditions are essential to a complete run of the *Hubert test* tool.

Note : Output attribute table

Two new fields populates the output attribute table : "*Rank_AGO*" and "*AGO_val*". The first one is a unique id used to identify each AGO separately. The second one is the metric value of an AGO. It is the mean metric value of DGO metric values over a same AGO. Into an AGO, these fields are the same for all DGO.

If one DGO or several successive DGO do not have a metric value (i.e. "*Null*"), this will be considered as a homogeneous reach (Fig. 4).


Centerline
Disaggregated valley bottom
Attributed DGO
"Null" DGO
12 Rank_AGO value
12
11
10
9
9
8
7
6
5
Related attribute table
Disaggregated valley bottom

OBJECTID *	Shape *	Order_ID	Rank_UGO	Distance	Rank_DGO	Width	Rank_AGO	AGO_Val
312	Polygone	3	4	4149,999984	84	376,598771	7	371,768
313	Polygone	3	4	4199,999974	85	321,636806	7	371,768
314	Polygone	3	4	4249,999957	86	285,158362	8	231,0705
315	Polygone	3	4	4300,00001	87	225,985223	8	231,0705
316	Polygone	3	4	4350,000029	88	201,434012	8	231,0705
317	Polygone	3	4	4400,000036	89	201,74039	8	231,0705
318	Polygone	3	4	4450,000032	90	200,760365	8	231,0705
319	Polygone	3	4	4500,000041	91	228,602702	8	231,0705
320	Polygone	3	4	4550,000004	92	273,812533	8	231,0705
321	Polygone	3	4	4600,000043	93	323,81976	9	326,6433
322	Polygone	3	4	4650,000012	94	349,351525	9	326,6433
323	Polygone	3	4	4699,99997	95	357,936208	9	326,6433
324	Polygone	3	4	4749,99999	96	356,40158	9	326,6433
325	Polygone	3	4	4800,000036	97	355,741157	9	326,6433
326	Polygone	3	4	4849,999971	98	353,948473	9	326,6433
327	Polygone	3	4	4900,000013	99	342,697693	9	326,6433
328	Polygone	3	4	4950,000013	100	328,506362	9	326,6433
329	Polygone	3	4	4999,999984	101	324,614201	9	326,6433
330	Polygone	3	4	5050,000021	102	316,804445	9	326,6433
331	Polygone	3	4	5100,00001	103	275,238735	9	326,6433
332	Polygone	3	4	5149,999989	104	234,659654	9	326,6433
333	Polygone	3	4	5199,999989	105	<Nul>	10	<Nul>
334	Polygone	3	4	5249,999996	106	257,09842	11	257,0984
335	Polygone	3	4	5300,000028	107	<Nul>	12	<Nul>

Figure 4 Result of the *Hubert test* tool over a 50m disaggregated valley bottom attributed with valley bottom width. Each color represent one resulting AGO. DGO with "*Null*" value are considered as an AGO.

II.2. Management of temporary files

Temporary files created during the compilation are managed thanks to the ArcGIS default geodatabase (*%ScratchWorkspace%*). If the user does not modify this geodatabase in the general environment proprieties, its path must looks like C:\Documents and Settings\<user>\My Documents\ArcGIS\Default.gdb. With the box "*Delete Temporary Files*", the user has the choice to keep or erase temporary files.

The *Hubert test* tool creates only one shapefile corresponding to the sorted input table. This entity is deleted from the default geodatabase at the end of the process.
Conversely, the .txt file created during a run is not deleted. It is stored into the folder which contains the default geodatabase.

III. References

Alber, A., and Piégay, H., 2011. Spatial disaggregation and aggregation procedures for characterizing fluvial features at the network-scale: Application to the Rhône basin (France). Geomorphology, Vol.125, No.3, pp.343-360

Hubert, P., 2000. The segmentation procedure as a tool for discrete modeling of hydrometeorological regimes. Stochastic Environmental Research and Risk Assessment, 14 (4-5), 297-304. 13th Jacques Cartier Conference on Climate Variability – Seasonal Forecast for Optimal Management of Water Resources, MONTREAL, CANADA, OCT 03-05,2000.

Kehagias, A., Nidelkou, E., and Petridis, V., 2005. A dynamic programming segmentation procedure for hydrological and environmental time series. Stochastic Environmental Research and Risk Assessment, 20.