Geometric Attributes Toolbox

Version 0.1

by Björn Nyberg[1*]

Department of Earth Sciences, University of Bergen, P.O. Box 7803, 5020 Bergen, Norway.[1]

Contact: bjorn.nyberg@uib.no

Contents

1.0 Background 3
1.1 License 3
1.2 Citation 3
2.0 Installation 3
2.1 Toolbox Installation 3
3.0 Algorithms 5
3.1 Centerlines 5
3.2 Directional Centerline 6
3.3 Geometric Attributes 7
3.4 Shape Analysis 8
3.5 Topologically Consistent Polygons 8
3.6 Adjacency 9
4.0 Example Dataset 10
4.1 Background 10
4.2 Project Setup 11
4.3 Centerlines 11
4.4 Directional Centerlines 14
4.5 Geometric Attributes 15

4.6 Shape Analysis 19

4.7 Adjacency 20

5.0 Help 21

5.1 Bugs and Issues 21

5.2 Feedback and Comments 21

1.0 Background

The Geometric Attributes Toolbox is a set of tools designed for the quantitative geometric attribute characterization of digitized features in QGIS 3.x >. This readme file is under construction showing the installation procedure, algorithms and a quick user guide showing the workflow.

1.1 License

The scripts used in this program are written in the Python programming language under a GNU General Public License V3 which states:

> "This program is free software: you can redistribute it and/or modify
> it under the terms of the GNU General Public License as published by
> the Free Software Foundation, either version 3 of the License, or
> (at your option) any later version.
>
> This program is distributed in the hope that it will be useful,
> but WITHOUT ANY WARRANTY; without even the implied warranty of
> MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
> GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License
> along with this program. If not, see <http://www.gnu.org/licenses/>."

1.2 Citation

These scripts can be used directly in your research or modified for a specific application but please cite the original work as:

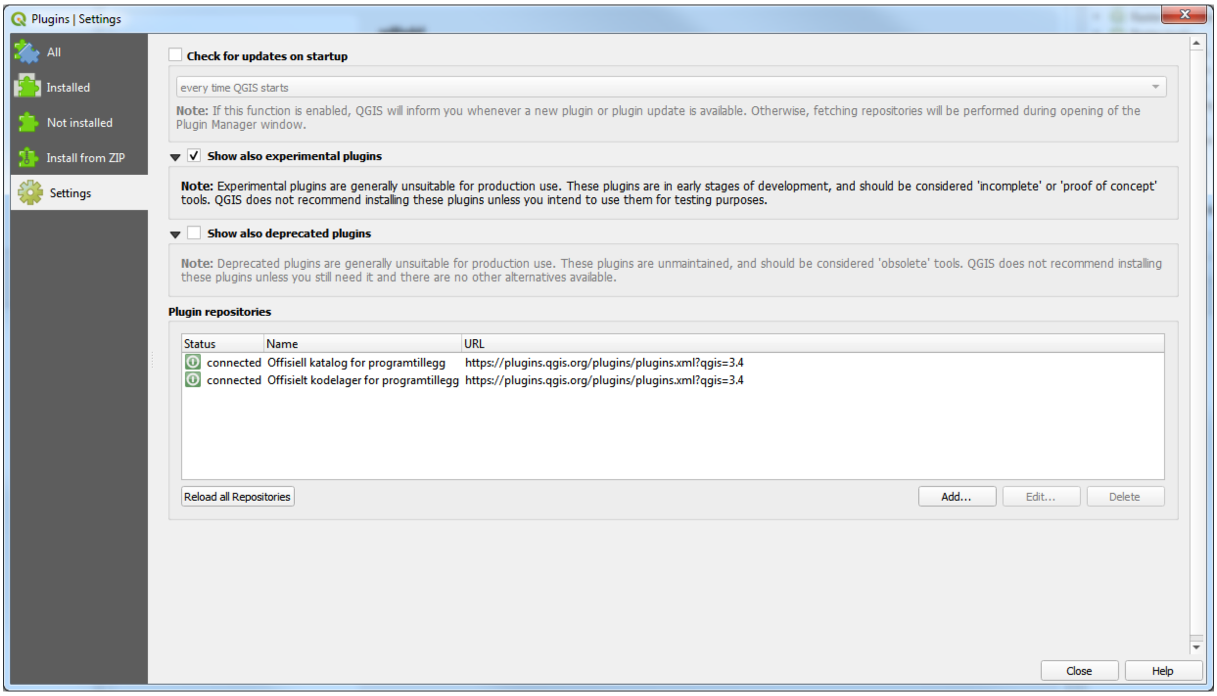
Nyberg, B., Buckley, S.J., Howell, J.A. and Nanson, R.A., 2015. Geometric Attribute and Shape Characterization of Modern Depositional Elements: A Quantitative GIS Method for Empirical Analysis. Computers & Geosciences, 82:191-204. doi:10.1016/j.cageo.2015.06.003.

2.0 Installation

2.1 Toolbox Installation

Install the geometric attributes using the QGIS plugin manager following the steps:

1. **Check the "Show also experimental plugins" button** in the plugin manager found at Plugins → Manage and Install Plugins… → Settings → Show also experimental plugins



□ Check for updates on startup

every time QGIS starts

Note: If this function is enabled, QGIS will inform you whenever a new plugin or plugin update is available. Otherwise, fetching repositories will be performed during opening of the Plugin Manager window.

☑ Show also experimental plugins

Note: Experimental plugins are generally unsuitable for production use. These plugins are in early stages of development, and should be considered 'incomplete' or 'proof of concept' tools. QGIS does not recommend installing these plugins unless you intend to use them for testing purposes.

□ Show also deprecated plugins

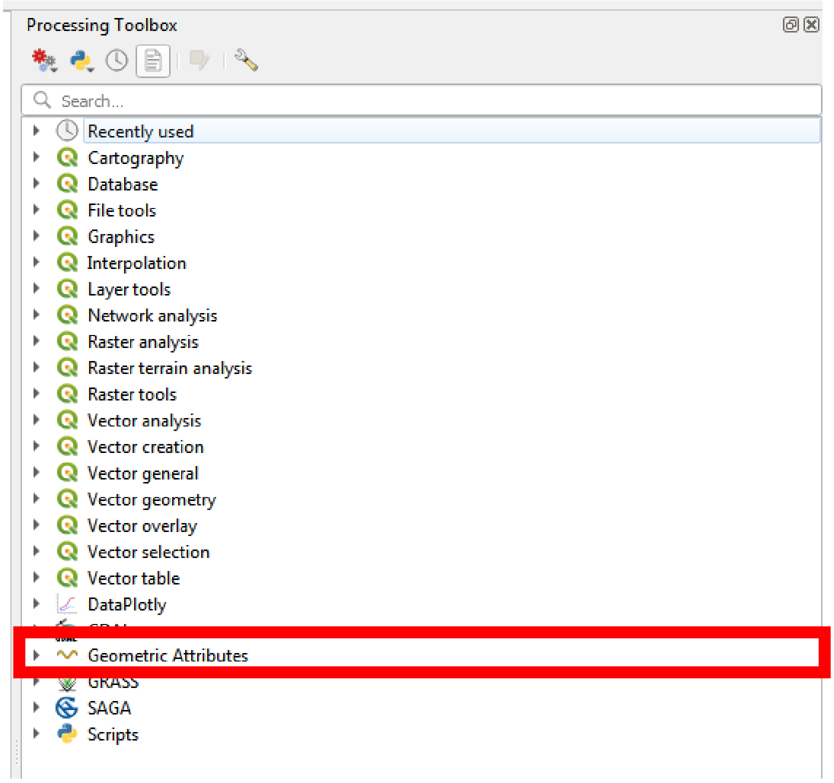
Note: Deprecated plugins are generally unsuitable for production use. These plugins are unmaintained, and should be considered 'obsolete' tools. QGIS does not recommend installing these plugins unless you still need it and there are no other alternatives available.

Plugin repositories

Status	Name	URL
connected	Offisiell katalog for programtillegg	https://plugins.qgis.org/plugins/plugins.xml?qgis=3.4
connected	Offisielt kodelager for programtillegg	https://plugins.qgis.org/plugins/plugins.xml?qgis=3.4

2. **Search and install the Geometric Attributes tool.** Plugins → Manage and Install Plugins… → Search "Geometric Attributes" → Install Plugin

3. The plugin is now available for use under the processing toolbox. Processing → Toolbox (Ctrl + Alt+ T)



3.0 Algorithms

The **Geometric Attribute Toolbox** algorithms require a vectorized dataset where polygons are represented as individually interpreted features. All features **must** be in a projected coordinate system. For more information, please refer to the publication:

Nyberg, B., Buckley, S.J., Howell, J.A. and Nanson, R.A., 2015. Geometric Attribute and Shape Characterization of Modern Depositional Elements: A Quantitative GIS Method for Empirical Analysis. Computers & Geosciences, 82:191-204. doi:10.1016/j.cageo.2015.06.003.

3.1 Centerlines

The Centerlines script will calculate the centerlines of polygons based on the voronoi lines of a feature.

Note – Overlapping and/or touching features and multipart features are not supported! It is **recommended** to use to the buffer tool and **buffer by an insignificant negative** amount to **create non-overlapping or touching features.**

Parameters

Polygons – Polygons to calculate the centerlines

Method –

1. **'Centerlines'** will calculate the shortest centerline path from start to furthest available endpoint
2. **'All'** will calculate all centerlines within a polygon from start to furthest endpoint
3. **'Circles'** will calculate all loops or circles present within a polygon
4. A number e.g. **'2'** will create all centerlines from start to multiple endpoints where the branch of each pathway is preserved when trimming the voronoi lines within each polygon segment.

In figure 1, the blue centerline is the longest centerline by distance in the network; the green centerline is created if a number in method #4 is equal or less than 2 (i.e. removing 2 segments will still preserve the start of the branch), the branch of the red centerline is only preserved if 1 segment is removed. Default value is set to 0 which will only create the longest centerline (blue).

Line Spacing – The density of vertices along the polygon features to calculate the voronoi line mesh. Smaller number yields a more detailed centerline but requires additional processing. Default of 0.0 will use the original vertices of each polygon feature.

Centerlines – Output location for the centerlines feature class.

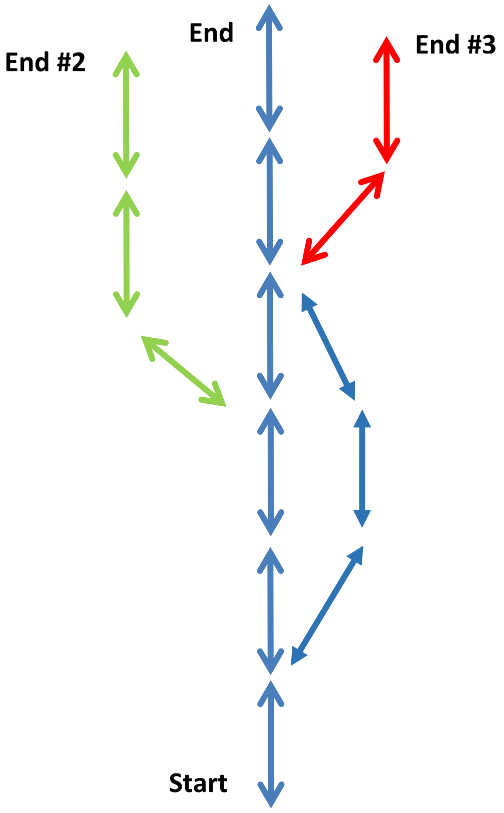


Figure 1 – Centerline creation

3.2 Directional Centerline

The directional centerline script will reverse the distance field based on the proximity to another feature class line to indicate the start point of each centerline. Alternatively, the algorithm will reverse and calculate the distance in relation to another centerline by assigning the start point of each centerline as the shortest 'Distance' attribute to the *Direction* feature class.

Parameters

Centerlines – Centerlines of each polygon feature with an unique ID field.

Direction – A feature layer representing the start direction to orient all centerlines. Alternatively, another centerline with a 'Distance' field will reverse and orient all centerlines by assigning the startpoint based on the lowest observable 'Distance' value.

Directional Centerlines - Output location for the directional centerlines feature class.

Directional Centerlines Fields

Field Name	Field Type	Description
AlongDist	Float	Distance along another centerline feature

> **Note** – AlongDist or along distance field will only be created if another centerline feature is used in the Direction parameter

3.3 Geometric Attributes

Calculate geometric attributes of width and centerline deviation along a centerline of a polygon.

Parameters

Polygons – Polygons to calculate the centerlines with a unique ID field.

Centerlines – Centerlines of each polygon feature with an unique ID field.

> **Note** – Polygons require an ID field that match the ID attribute of its centerline!

Samples – The number of samples in each polygon to calculate geometric attributes.

Sample by distance – Sample by distance rather than number using the 'Samples' parameter

Fast Compute – Calculate width of a feature using the polygon boundary vertices rather than defining a perpendicular line to each centerline. Recommended for large datasets.

Geometric Attributes – Output location for the geometric attributes feature class.

Geometric Attributes Output

Field Name	Field Type	Description
Distance	Float	Distance along centerline in map units
SP_Dist	Float	Shortest path to the centerline segment in map units
Width	Float	Width of the feature in map units
Deviation	Float	Deviation to shortest path line in map units
DWidthL	Float	Deviated width left of the centerline
DWidthR	Float	Deviated width right of the centerline
Diff	Float	Difference in width left and right of the centerline

3.4 Shape Analysis

The Shape Analysis tool will define the shape of a feature class based on the measured geometric attributes. The defined shapes are Ellipsoidal (E), Crescentic(C) or Sinuous (L) with an Asymmetrical (A), Symmetrical (S) or Linear (L) profile.

Parameters

Polygons – Polygons to calculate the centerlines with a unique ID field.

Geometric Attributes – Geometric attributes of each polygon created in step 3.2.

Note – Polygons require an ID field that match the ID attribute of its geometric attributes.

Directional (optional) – Will define an additional parameter to the shape classification whereby 0 indicates that the asymmetry is in the direction along increasing distance of the centerline and 1 is vice versa. Recommended if using a directional centerline (step 3.3).

Output - Output location for the shape analysis feature class.

Shape Analysis Attribute Fields

Field Name	Field Type	Description
Length	Float	Length of the polygon
Sinuosity	Float	Sinuosity of the polygon
Width	Float	Maximum width of the polygon
Deviation	Float	Maximum centerline deviation of the polygon
Shape	Float	Shape of the polygon

Additional Information – For more information refer to the publication:

Nyberg, B., Buckley, S.J., Howell, J.A. and Nanson, R.A., 2015. Geometric Attribute and Shape Characterization of Modern Depositional Elements: A Quantitative GIS Method for Empirical Analysis. Computers & Geosciences, 82:191-204. doi:10.1016/j.cageo.2015.06.003.

3.5 Topologically Consistent Polygons

The topologically consistent polygon algorithm will examine all selected feature within a feature layer and remove any overlap with all other features in the layer to create a topologically consistent feature class.

Parameters

Polygons – Polygons to analyze. If features are selected, only the selected features will be modified.

Tolerance – Number indicating the distance (buffer) in map units to create a topologically consistent polygon dataset.

Output - Output location for the topologically consistent polygons feature class.

3.6 Adjacency

The Connected Elements script will examine each individual feature and provide information which group that feature is connected to and the direct neighbors that feature shares a border with.

Parameters

Polygons – Polygons to analyze. If features are selected, only the selected features will be analysed for clusters.

Field – Category field in the Polygons attribute table to summarize the total perimeter of a polygon shared with other features of that same category.

Tolerance – Number indicating the distance (buffer) in map units to analyze the approximate shared border length between features. This is calculated as the overlapped area between the two polygons divided by two. The tolerance parameter also controls the distance by which to cluster all features.

Output - Output location for the adjacency feature class.

Adjacency Attribute Fields

The number of new fields depends on the number of unique values in the 'Field' parameter. Each new field will contain a float representing an approximate perimeter length shared to that feature class type.

> **Note** – The shared border perimeter is an approximation calculated as the overlapped region divided by two. Errors will occur if there is significant overlap (or tolerance) in input Polygons feature layer.

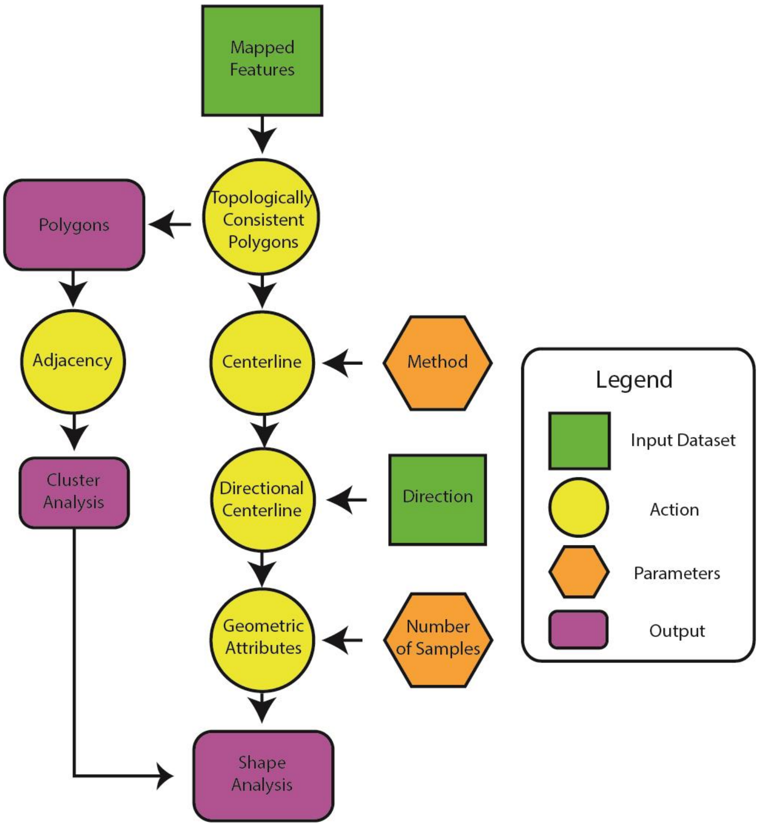
Field Name	Field Type	Description
Adjacent	Text	A list of ids connected to the polygon
Connection	Integer	Unique cluster ID for the connected group of polygons

4.0 Example Dataset

This tutorial assumes that you have a basic understanding of the QGIS environment and vector layer creation. A brief introduction can be found on the QGIS documentation manual available at https://docs.qgis.org/3.4/en/docs/training_manual/index.html

4.1 Background

The purpose of this exercise is to demonstrate the versatility and efficiency of the algorithms to analyze geometry of large vector datasets. We will use a sample dataset from the Democratic Republic of the Congo to show the geometry along the **Congo River**, which is available for **download** on the GitHub page https://github.com/BjornNyberg/Geometric-Attributes-Toolbox/Datasets



Geometric Attributes Workflow

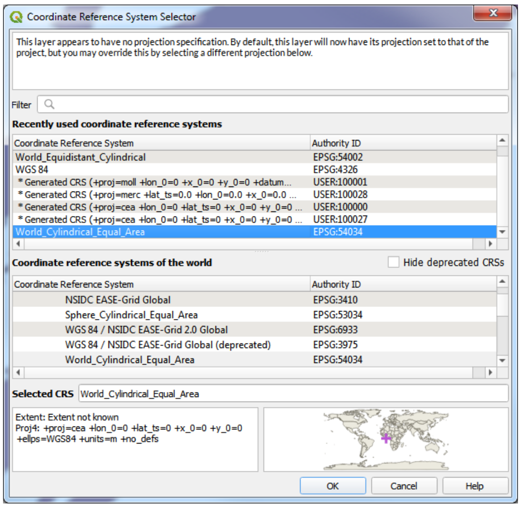
For more information regarding the dataset and pre-processing, please refer to the original paper at:

Nyberg, B., Buckley, S.J., Howell, J.A. and Nanson, R.A., 2015. Geometric Attribute and Shape Characterization of Modern Depositional Elements: A Quantitative GIS Method for Empirical Analysis. Computers & Geosciences, 82:191-204. doi:10.1016/j.cageo.2015.06.003.

4.2 Project Setup

Open a new QGIS project and **add the vector** data **CongoRiver.shp** and **CongoBars.shp** downloaded from step 4.1. If the layer is added without a spatial reference then set the reference to the World_Cylindrical_Equal_Area (EPSG:54034) by right clicking the layer under the layers tab and

Set CRS → Set Layer CRS… → World_Cylindrical_Equal_Area → OK



Coordinate Reference System Selector

This layer appears to have no projection specification. By default, this layer will now have its projection set to that of the project, but you may override this by selecting a different projection below.

Filter

Recently used coordinate reference systems

Coordinate Reference System	Authority ID
World_Equidistant_Cylindrical	EPSG:54002
WGS 84	EPSG:4326
* Generated CRS (+proj=moll +lon_0=0 +x_0=0 +y_0=0 +datum...	USER:100001
* Generated CRS (+proj=merc +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 ...	USER:100028
* Generated CRS (+proj=cea +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 ...	USER:100000
* Generated CRS (+proj=cea +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 ...	USER:100027
World_Cylindrical_Equal_Area	EPSG:54034

Coordinate reference systems of the world ☐ Hide deprecated CRSs

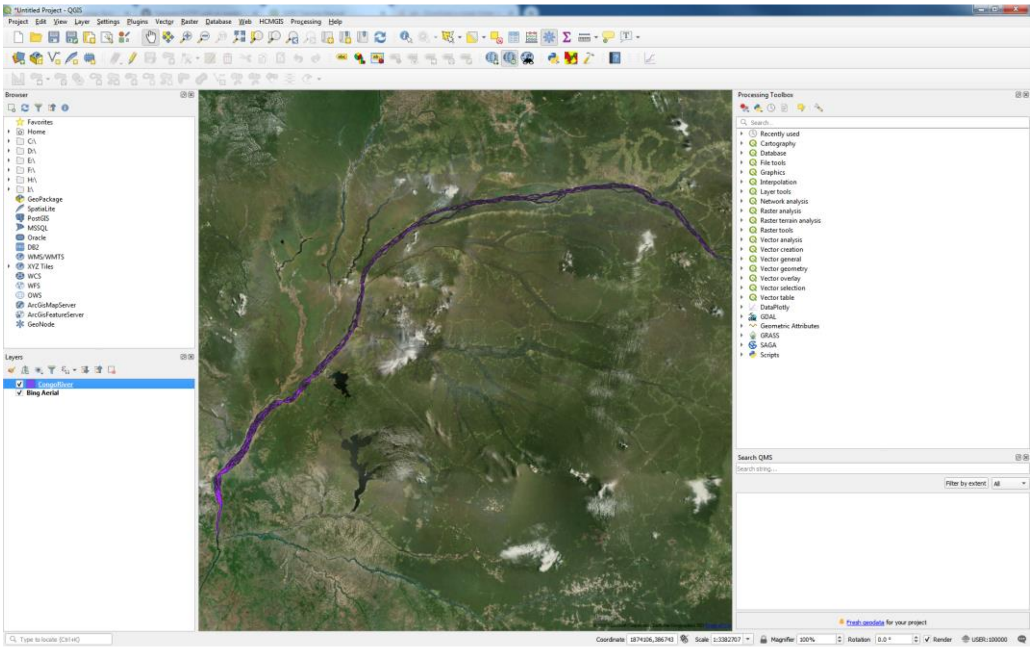
Coordinate Reference System	Authority ID
NSIDC EASE-Grid Global	EPSG:3410
Sphere_Cylindrical_Equal_Area	EPSG:53034
WGS 84 / NSIDC EASE-Grid 2.0 Global	EPSG:6933
WGS 84 / NSIDC EASE-Grid Global (deprecated)	EPSG:3975
World_Cylindrical_Equal_Area	EPSG:54034

Selected CRS World_Cylindrical_Equal_Area

Extent: Extent not known
Proj4: +proj=cea +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs

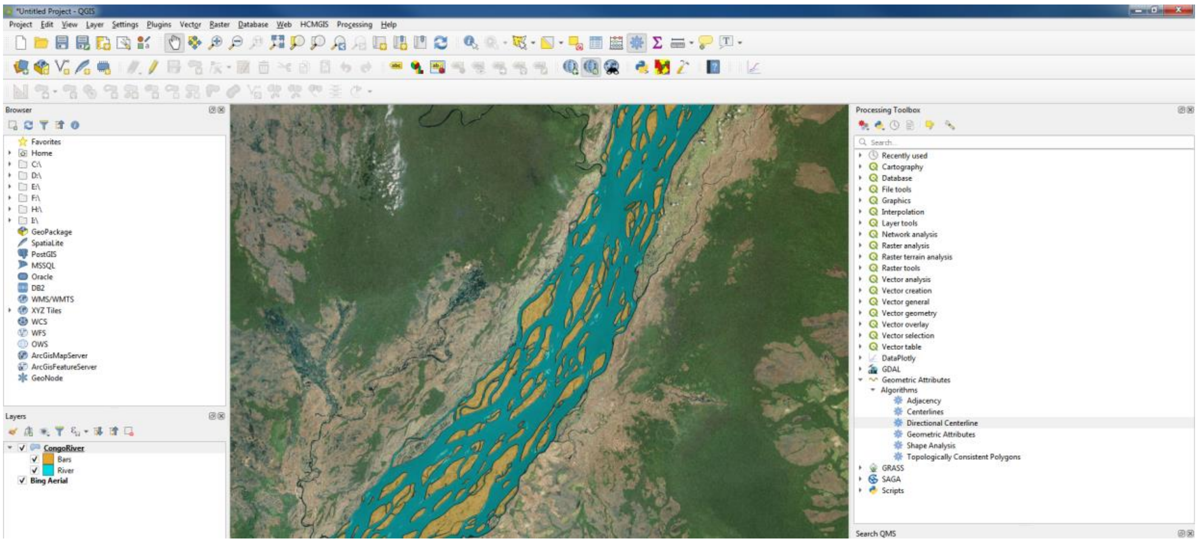
It may be useful to visualize the location of the dataset in relation to satellite imagery. If you have the OpenLayers Plugin (experimental) installed, add Bing Aerial imagery to your map:

Web → OpenLayers Plugin → Bing Maps → Bing Aerial

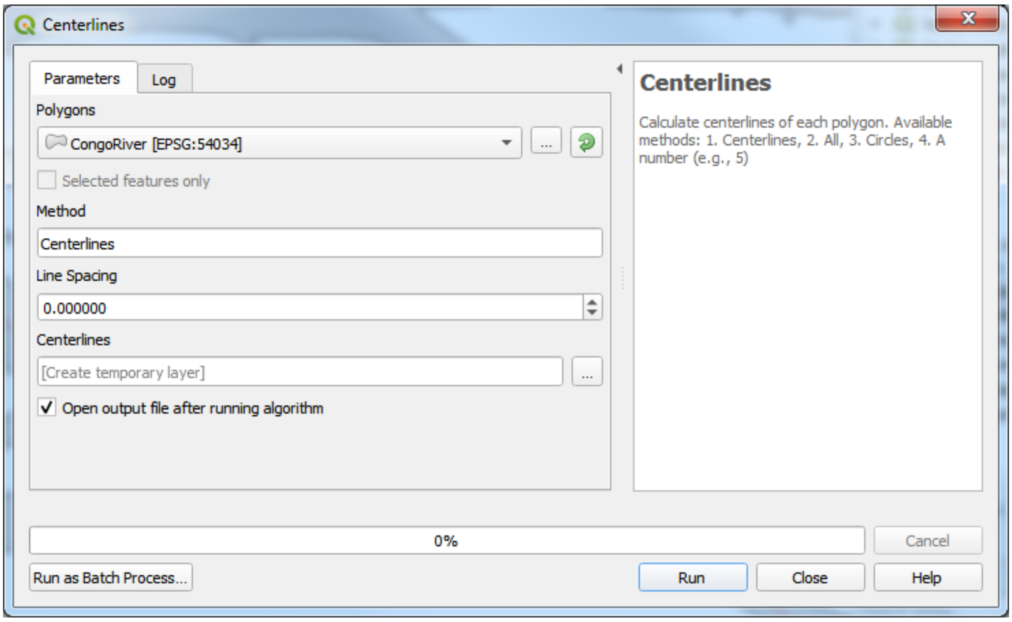


4.3 Centerlines

The centerline analysis of the Congo River can be split into two parts: 1) the river itself and 2) the mid-channel bars of the river system (see figure below).



1. First let's analyze the Congo River using the **default values in the "Centerlines" tool**. Use the **CongoRiver feature layer** as the input into the **Polygons parameter** and press **'Run'**



Centerlines

Parameters | Log

Polygons

CongoRiver [EPSG:54034]

□ Selected features only

Method

Centerlines

Line Spacing

0.000000

Centerlines

[Create temporary layer]

☑ Open output file after running algorithm

Centerlines

Calculate centerlines of each polygon. Available methods: 1. Centerlines, 2. All, 3. Circles, 4. A number (e.g., 5)

0%

Important - notice the centerline of the resulting feature class is not particularly smooth. This is because the algorithm used the original vertices of the input features. Using the Line Spacing parameter will increase detail to the centerline but also increase processing time and memory.

In the attributes table notice that there are four fields of 'Distance', 'RDistance', 'SP_Dist' (SP_Dist = Shortest Path Distance') and 'RSP_Dist'. Considering the centerline is created spatially unaware of its start point, the fields of 'RDistance' and 'RSP_Dist' simple represent the reverse values of 'Distance' and 'SP_Dist'.

2. Run the Centerlines tool again and type **All** in the '**Methods**' parameter and use a **Line Spacing** of **250**. Notice that the output creates multiple centerlines from start to finish. Keep this centerline for the next steps.

Tip – Before moving on, test the other centerline methods on the CongoRiver dataset. Type **Circles** in the '**Methods**' parameter and use a **Line Spacing** of **250** and press **Run.** Lastly, type **100** in the '**Methods**' parameter and use a **Line Spacing** of **250**. Check the result – notice any difference?

3. Visualize the results of the centerline in step 2 by using Graduated Symbology on the 'Distance' field.

Right click → Properties → Symbology → Graduated → Distance → Classify → OK

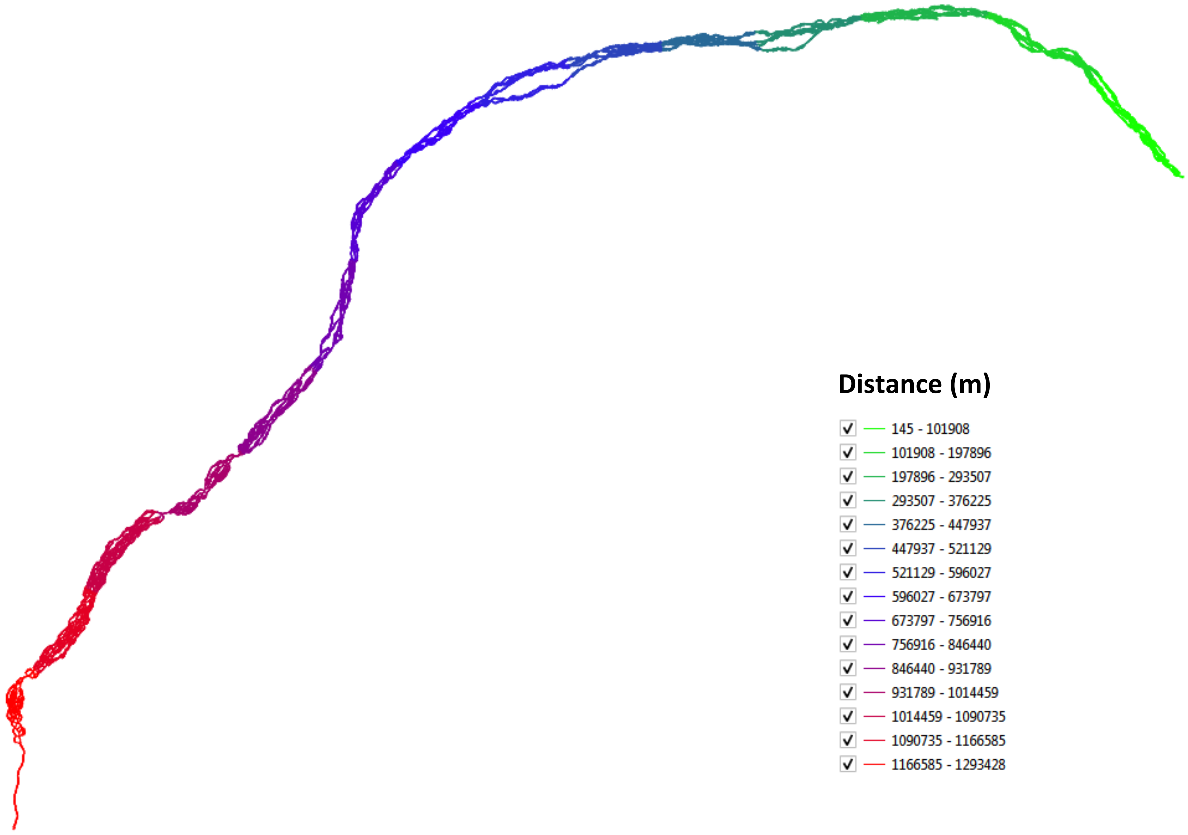
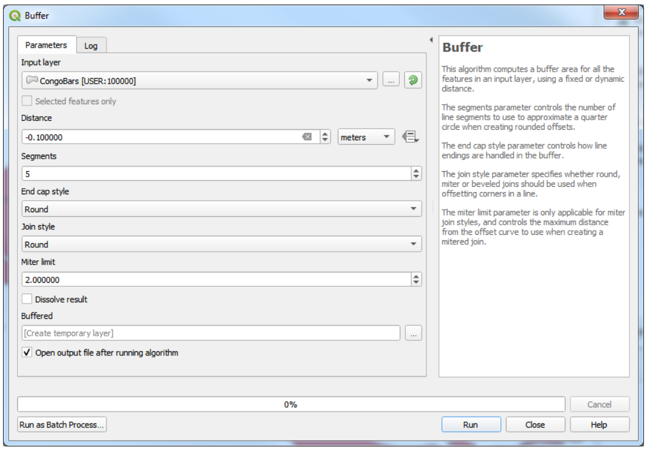


Figure – Green colors indicate proximal values and red colors indicate distal values.

4. Next we will analyse the CongoBars dataset, however the dataset contains bars that touch one another and hence needs pre-processing. To accomplish this we will apply the **Buffer tool** to ensure that there are no features that touch one another in the datasets. To do this we simply create a small insignificant negative '**Distance'** of **-0.1**.


Buffer

Input layer: CongoBars [USER:100000]

☐ Selected features only

Distance: -0.100000 meters

Segments: 5

End cap style: Round

Join style: Round

Miter limit: 2.000000

☐ Dissolve result

Buffered: [Create temporary layer]

☑ Open output file after running algorithm

Buffer

This algorithm computes a buffer area for all the features in an input layer, using a fixed or dynamic distance.

The segments parameter controls the number of line segments to use to approximate a quarter circle when creating rounded offsets.

The end cap style parameter controls how line endings are handled in the buffer.

The join style parameter specifies whether round, miter or beveled joins should be used when offsetting corners in a line.

The miter limit parameter is only applicable for miter join styles, and controls the maximum distance from the offset curve to use when creating a mitered join.

5. Run the Centerlines tool again this time with the negative **Buffered** feature class as the ‘**Polygons’ parameter** input, **Centerlines** in the ‘**Methods**’ parameter and **50** for the line spacing. Visualize the ‘Distance’ field of the Congo Bar centerlines as shown previously in step 3. Notice that the centerlines are spatially unaware with no sense of start or endpoint.

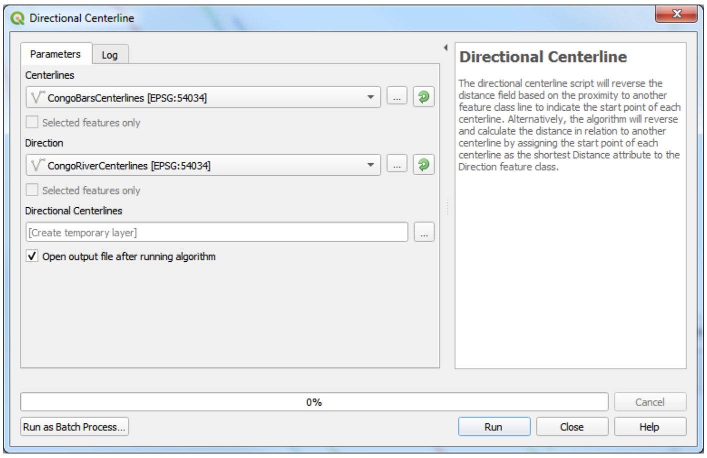
Tip – This may take some time depending on the speed of your computer. If it takes too long – consider using a higher line spacing value (e.g., 100) or subset the dataset!

Tip #2 – The amount to buffer your dataset will depend on the scale of investigation. A -10 meter buffer may be appropriate for a 10km feature, while a -10milimeter may be suitable for a 1meter long feature.

4.4 Directional Centerlines

6. The spatially unaware centerlines can be corrected using the **‘Directional Centerlines’ tool.** For the ‘**Centerlines’ parameter** use the Congo Bar Centerlines from step 5, and for the **Direction parameter** use the Congo River Centerlines from step 2. Press Run.

   Visualize the output compared to the original ‘Distance’ field created in step 5. Furthermore, take a note of the new AlongDist parameter that shows the Distance value from the Direction parameter feature class and applied that to the Congo Bars dataset.


Directional Centerline

Centerlines: CongoBarsCenterlines [EPSG:54034]

☐ Selected features only

Direction: CongoRiverCenterlines [EPSG:54034]

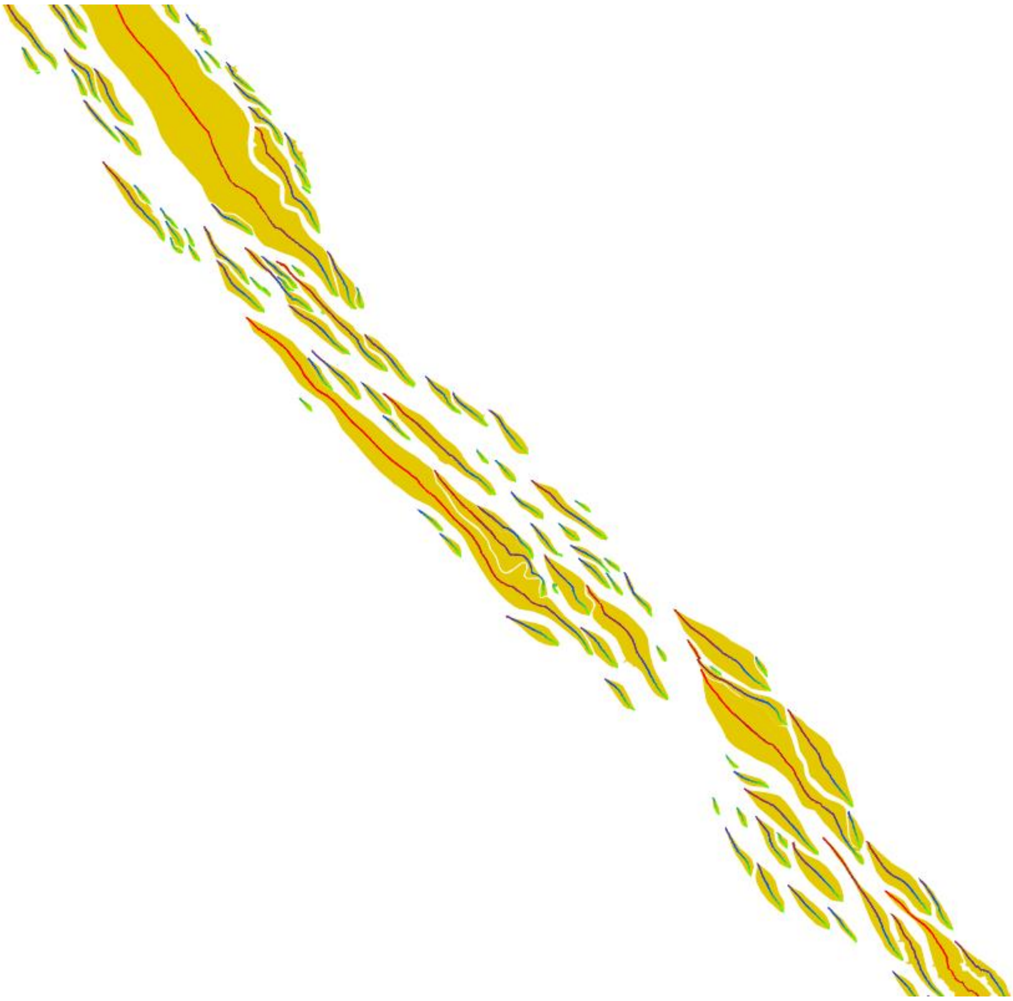
☐ Selected features only

Directional Centerlines: [Create temporary layer]

☑ Open output file after running algorithm

Directional Centerline

The directional centerline script will reverse the distance field based on the proximity to another feature class line to indicate the start point of each centerline. Alternatively, the algorithm will reverse and calculate the distance in relation to another centerline by assigning the start point of each centerline as the shortest Distance attribute to the Direction feature class.



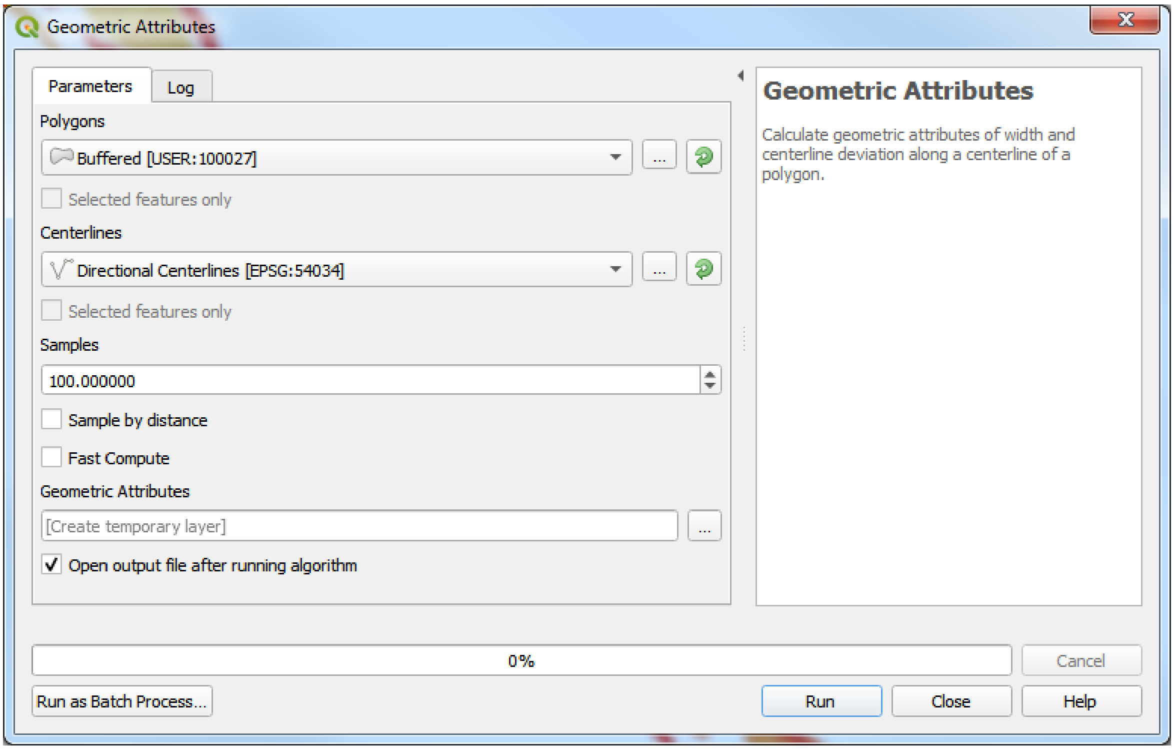
Screenshot showing the distance attribute of the 'Directional Centerlines' tool feature class output. Notice the increase in distance is in the same orientation for all mid-channel bars.

4.5 Geometric Attributes

7. To calculate width and centerline deviation along the centerline of each river bar we will use the '**Geometric Attributes' tool.** Use the **Buffered** feature class as the '**Polygons' parameter** input from step 4 and **CongoBars Directional Centerlines** feature class from step 6 as the '**Centerlines' parameter** input. Keep the default values.

Tip – If you want to measure width samples along all segments of the centerline – change the 'Samples' parameter to 0.0.

Tip #2 – Try step 7 again and use the 'Sample by distance' check box. Notice that the output defines a new width and centerline deviation at the closest distance to the given 'Samples' parameter interval.



Polygons

Buffered [USER:100027]

□ Selected features only

Centerlines

Directional Centerlines [EPSG:54034]

□ Selected features only

Samples

100.000000

□ Sample by distance

□ Fast Compute

Geometric Attributes

[Create temporary layer]

☑ Open output file after running algorithm

Geometric Attributes

Calculate geometric attributes of width and centerline deviation along a centerline of a polygon.

8. Notice in the previous step that a number of lines were created perpendicular to each centerline. This provides an opportunity to check and quality control the results prior to any statistical analysis. The 'Diff' attribute field shows the difference in length between the width measurements made left and right of each centerline whereby 100% is the same (width1/width2). This field may be used to delete erroneous measurements. Use the **Select by Expressions** function found in the **attributes table** of the **Geometric Attributes output.**

Right click → Open Attributes Table → Select by Expression

Type **"Diff" < 95** and press Select features to select all features that differ by more than 5%.

Delete those selected features in the attributes table by **'Toggle Editing'** -> **'Delete selected features'** and press the **'Toggle Editing'** **.** Remember to save all your edits!

9. Lets visualize the results using DataPlotly - https://github.com/ghtmtt/DataPlotly (download the plugin using the QGIS plugin manager – see step 2.1).

First, let us select one of the bars measurements to graph. Use the **Select by Expressions** function found in the **attributes table** of the **Geometric Attributes output.**

Right click → Open Attributes Table → Select by Expression

Type **"ID" = 1440** and press Select features to select all geometric attributes belonging to feature id 1440.

Open DataPlotly and select the following options:

Plot Type: **Scatter Plot**

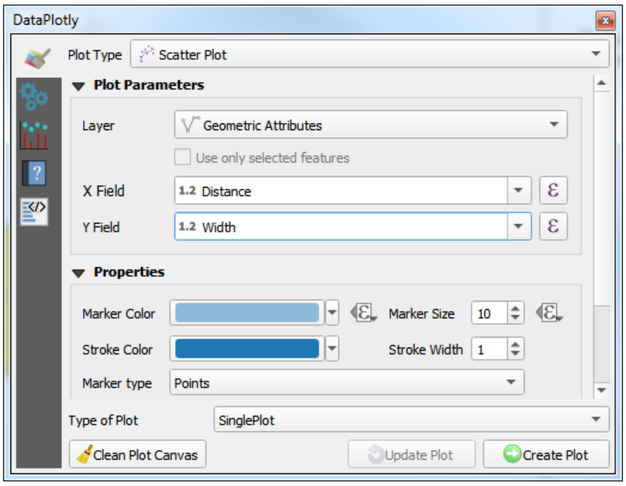
Layer: **Geometric Attributes**

Use only selected features – check

X Field: **Distance**

Y Field: **Width**

Press **Create Plot** Create Plot



DataPlotly

Plot Type: Scatter Plot

Plot Parameters

Layer: Geometric Attributes

□ Use only selected features

X Field: 1.2 Distance

Y Field: 1.2 Width

Properties

Marker Color

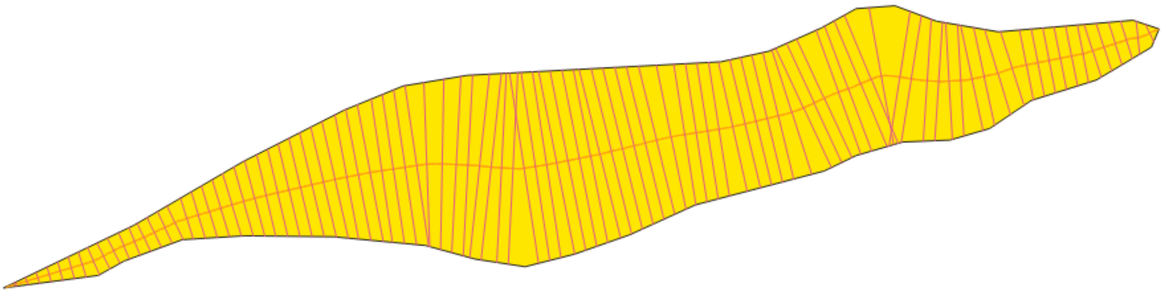
Marker Size: 10

Stroke Color

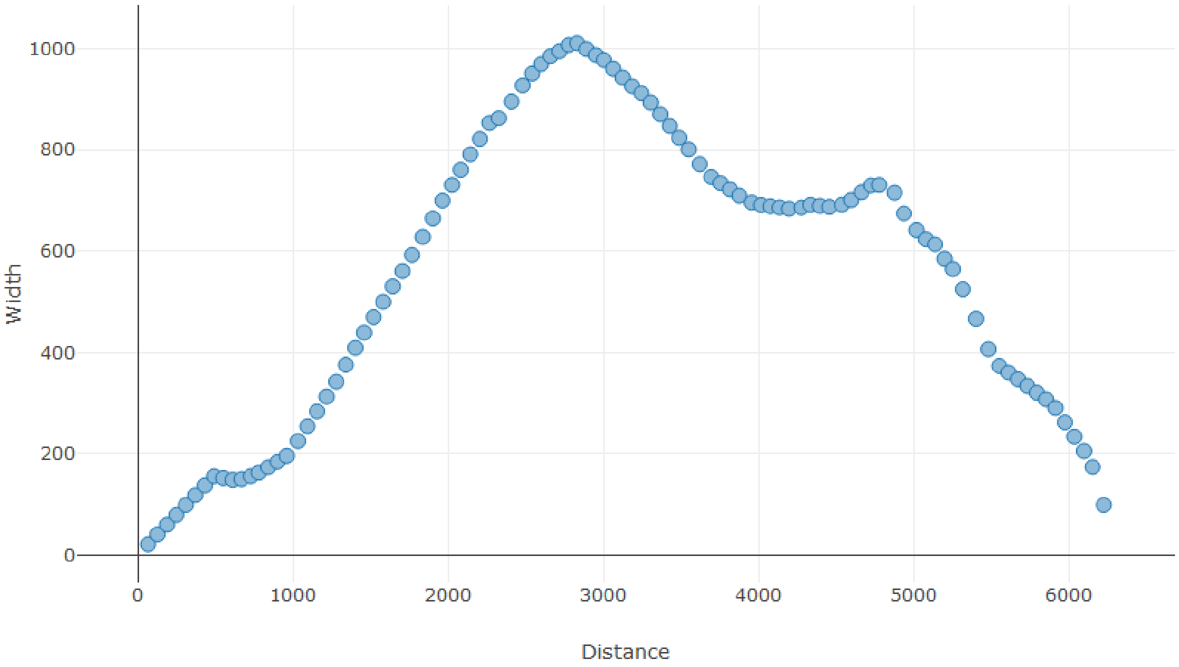
Stroke Width: 1

Marker type: Points

Type of Plot: SinglePlot



Original bar example

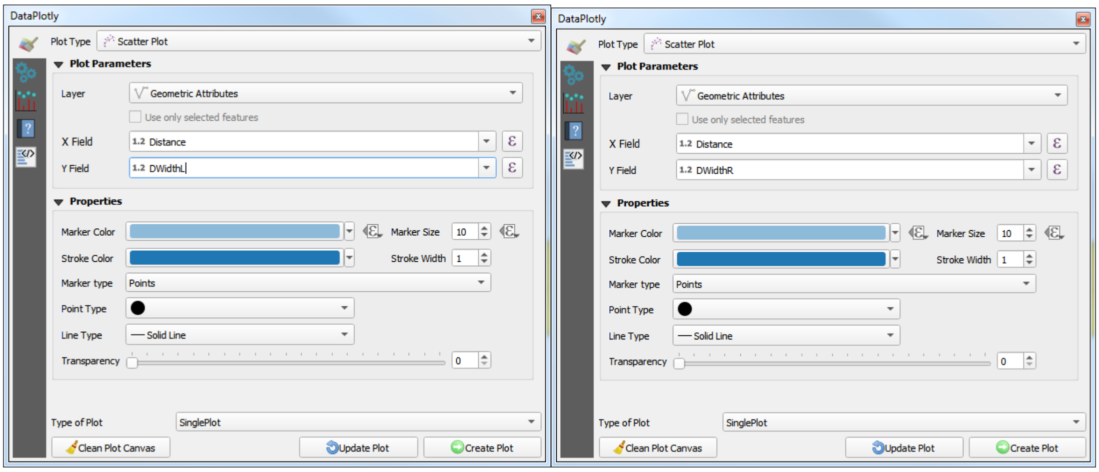


Screenshot above shows - distance versus width along the centerline of the bar

10. Clear the plot canvas in DataPlotly using the Clean Plot Canvas function.

    Now, **repeat** the procedure in step 9 and create **two superimposed scatter plots.** Use the **same parameter as above but modify the Y Field to DWidthR and DWidthL.**

Press the **Create Plot** Create Plot twice once for DWidthR and once for DWidthL.



DataPlotly

Plot Type: Scatter Plot

Plot Parameters

Layer: Geometric Attributes

□ Use only selected features

X Field: 1.2 Distance

Y Field: 1.2 DWidthL

Properties

Marker Color | Marker Size: 10

Stroke Color | Stroke Width: 1

Marker type: Points

Point Type: ●

Line Type: Solid Line

Transparency: 0

Type of Plot: SinglePlot

Clean Plot Canvas | Update Plot | Create Plot

DataPlotly

Plot Type: Scatter Plot

Plot Parameters

Layer: Geometric Attributes

□ Use only selected features

X Field: 1.2 Distance

Y Field: 1.2 DWidthR

Properties

Marker Color | Marker Size: 10

Stroke Color | Stroke Width: 1

Marker type: Points

Point Type: ●

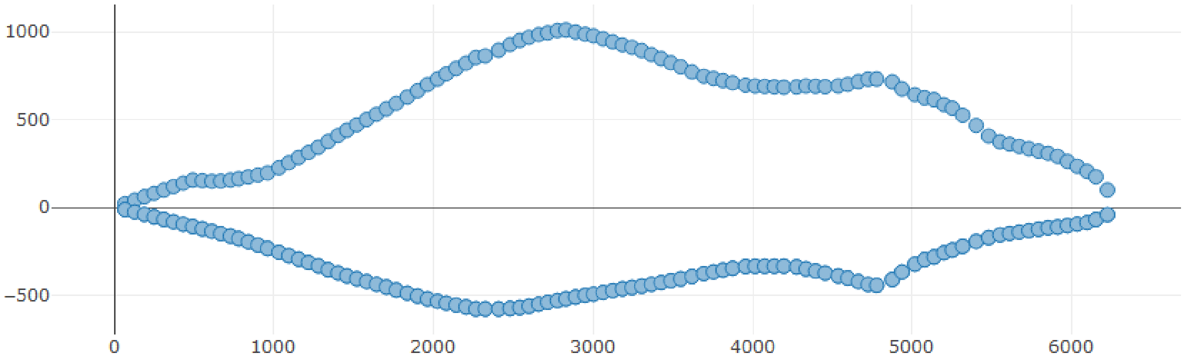
Line Type: Solid Line

Transparency: 0

Type of Plot: SinglePlot

Clean Plot Canvas | Update Plot | Create Plot

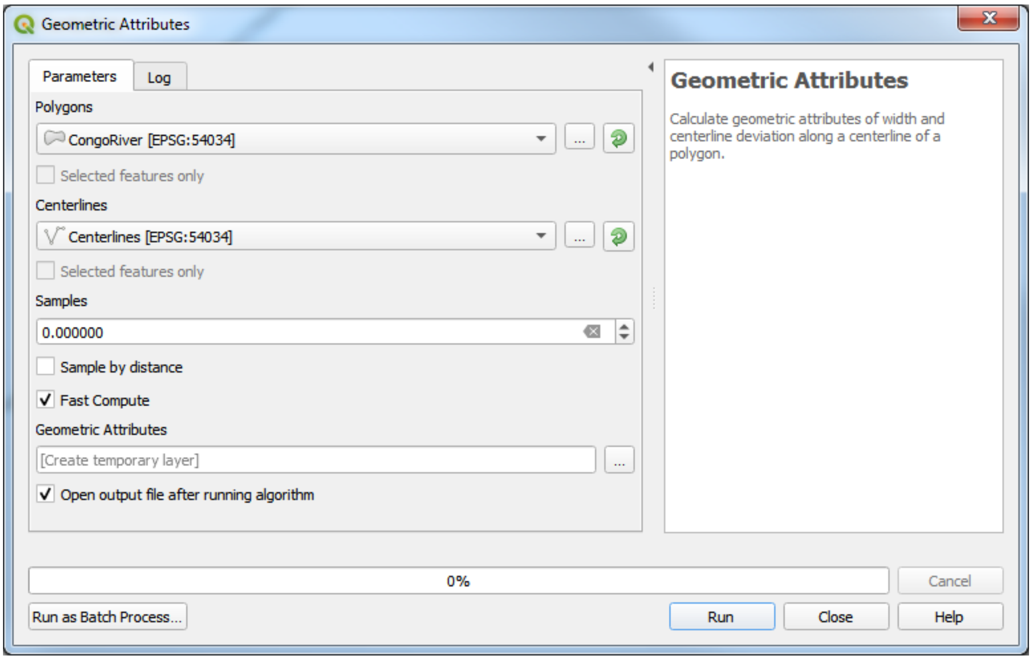
The figure should be similar to the one shown below indicating the width of the feature both left and right of its deviated centerline!



11. To calculate width and centerline deviation along the centerline of a large polygon feature that is complex, consider using the fast compute option in the 'Geometric Attributes

    Open the **Geomtric Attributes tool.**

    Use the **CongoRiver** feature class as the '**Polygons' parameter** input and **CongoRivers Centerlines** feature class from step 2 as the '**Centerlines' parameter** input. Use a '**Samples' parameter of 0.0** and check the '**Fast Compute'** option.



Geometric Attributes

Parameters | Log

Polygons: CongoRiver [EPSG:54034]

□ Selected features only

Centerlines: Centerlines [EPSG:54034]

□ Selected features only

Samples: 0.000000

□ Sample by distance

☑ Fast Compute

Geometric Attributes: [Create temporary layer]

☑ Open output file after running algorithm

Geometric Attributes

Calculate geometric attributes of width and centerline deviation along a centerline of a polygon.

0%

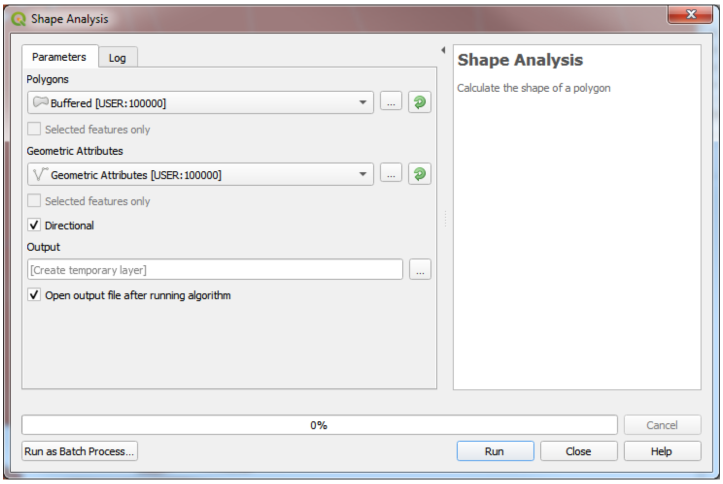
The results will now show an approximation of width by calculting the shortest distance from each centerline segment to the edge of the polygon for the Congo River!

4.6 Shape Analysis

12. Using the geometric attributes data, we can now objectively and automatically categorize these objects into groups depending on their shape using the ‘Shape Analysis’ tool.

Open the **‘Shape Analysis’ tool.**

For the ‘**Polygons’ parameter** use the **Buffered** feature class from step 4 and for the **‘Geometric Attributes’ parameter** use the geometric attributes feature class from step 7. Considering the centerline is directional from the step 6, **check** the **Directional box**. **Press Run.**



Shape Analysis

Parameters | Log

Polygons: Buffered [USER:100000]

□ Selected features only

Geometric Attributes: Geometric Attributes [USER:100000]

□ Selected features only

☑ Directional

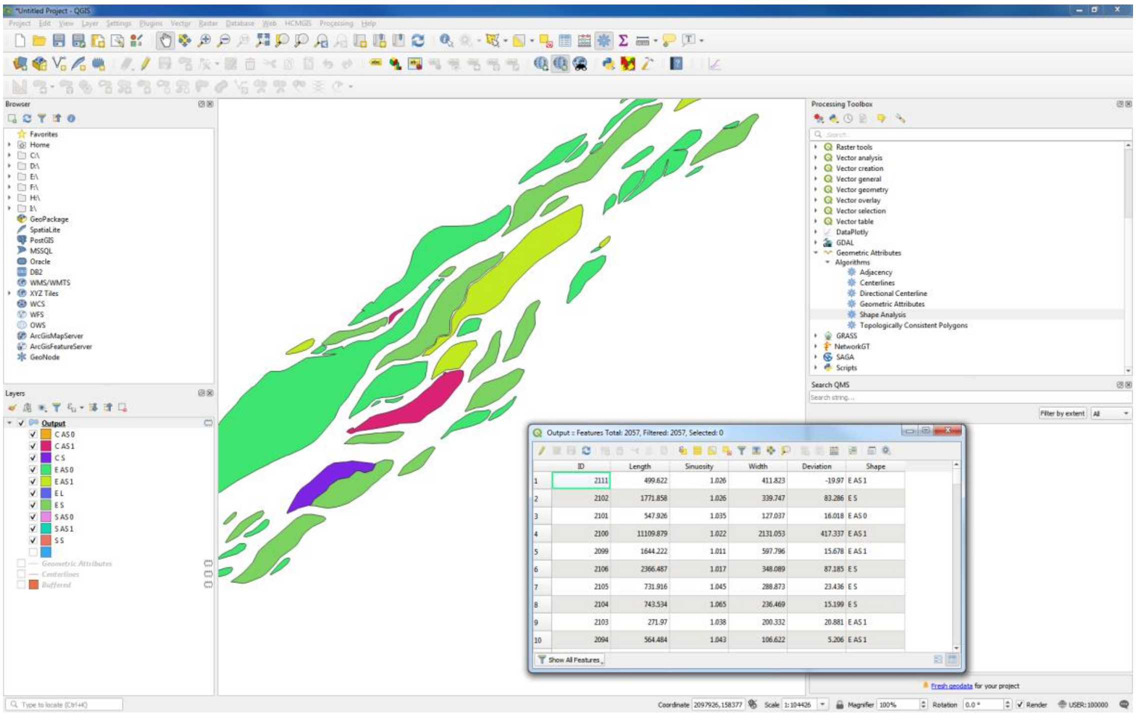
Output: [Create temporary layer]

☑ Open output file after running algorithm

Shape Analysis

Calculate the shape of a polygon

0%



Output :: Features Total: 2057, Filtered: 2057, Selected: 0

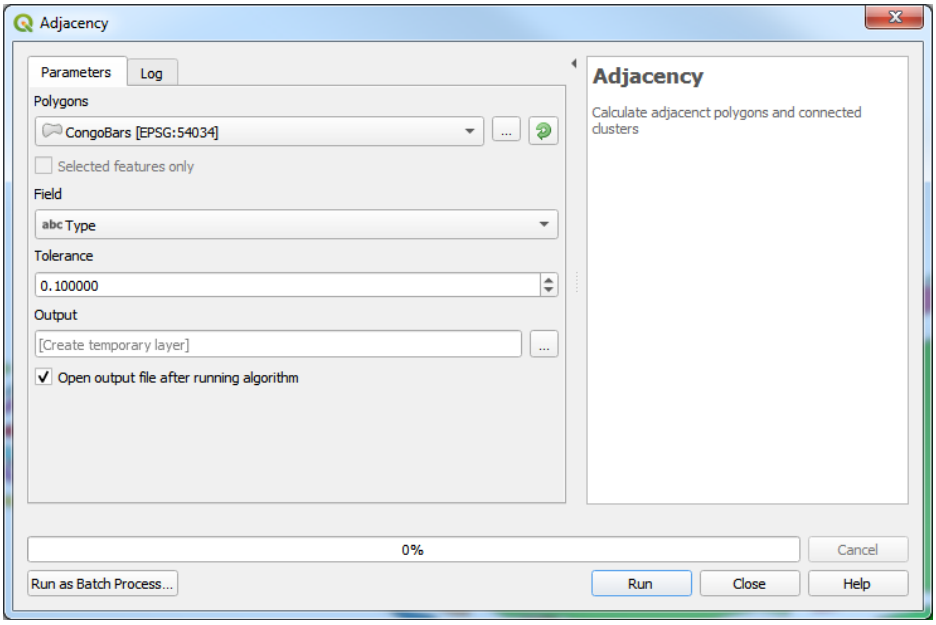
	ID	Length	Sinuosity	Width	Deviation	Shape
1	2111	499.622	1.026	411.823	-19.97	E AS 1
2	2102	1771.858	1.026	339.747	83.286	E S
3	2101	547.926	1.035	127.037	16.018	E AS 0
4	2100	11109.879	1.022	2131.053	417.337	E AS 1
5	2099	1644.222	1.011	597.796	15.678	E AS 1
6	2106	2366.487	1.017	348.089	87.185	E S
7	2105	731.916	1.045	288.873	23.436	E S
8	2104	743.534	1.065	236.469	15.199	E S
9	2103	271.97	1.038	200.332	20.881	E AS 1
10	2094	564.484	1.043	106.622	5.206	E AS 1

The results will not only classify the shape of the features **Ellipsoidal (E), Crescentic(C)** or **Sinuous (L)** with an **Asymmetrical (A), Symmetrical (S)** or **Linear (L)** profile but also summarize the maximum **length**, **sinuosity**, **width** and **deviation** of each feature.

4.7 Adjacency

13. **Open** the **'Adjacency' tool.**

For the '**Polygons' parameter** use the original **CongoBars** feature class and for the **'Field' parameter** use **Type** attribute. **Press Run.**



Adjacency

Calculate adjacent polygons and connected clusters

Polygons: CongoBars [EPSG:54034]

□ Selected features only

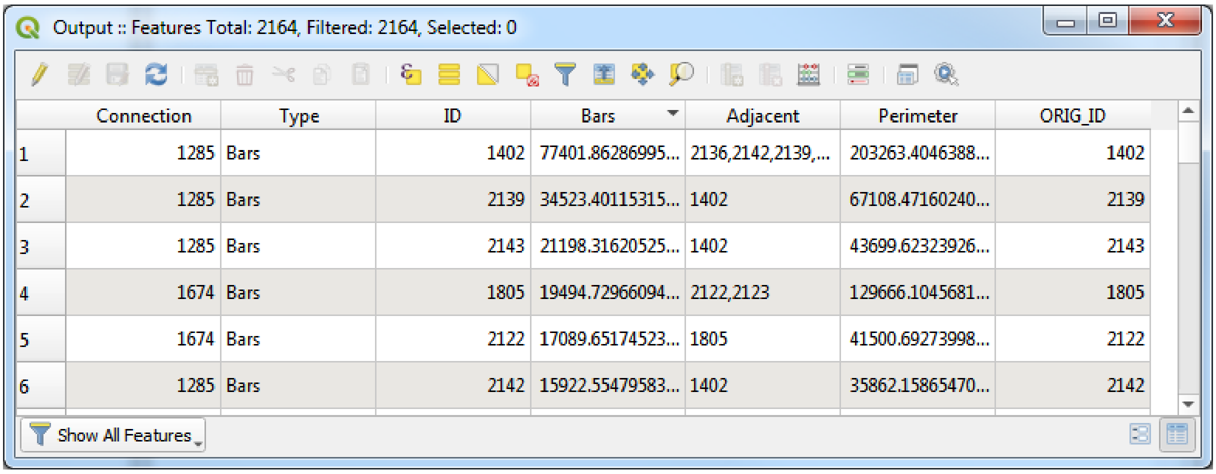
Field: abc Type

Tolerance: 0.100000

Output: [Create temporary layer]

☑ Open output file after running algorithm

The resulting feature class will have an attribute table showing the unique feature ids that are connected to each feature (**'Adjacent'**) and the cluster that feature belongs to (**'Connection'**). In addition, notice that the output contains a new **'Bars'** attribute field. This attribute shows the perimeter that feature is shared with all other features with a category of **'Bars'.**



Output :: Features Total: 2164, Filtered: 2164, Selected: 0

	Connection	Type	ID	Bars	Adjacent	Perimeter	ORIG_ID
1	1285	Bars	1402	77401.86286995...	2136,2142,2139,...	203263.4046388...	1402
2	1285	Bars	2139	34523.40115315...	1402	67108.47160240...	2139
3	1285	Bars	2143	21198.31620525...	1402	43699.62323926...	2143
4	1674	Bars	1805	19494.72966094...	2122,2123	129666.1045681...	1805
5	1674	Bars	2122	17089.65174523...	1805	41500.69273998...	2122
6	1285	Bars	2142	15922.55479583...	1402	35862.15865470...	2142

5.0 Help

5.1 Bugs and Issues

Please note that this toolbox is in development and may include unforeseen bugs and issues. Bugs and issues related to the toolbox or feature requests and enhancement should be reported on the GitHub page at https://github.com/BjornNyberg/Geometric-Attributes-Toolbox/issues

5.2 Feedback and Comments

For any additional assistance or feedback please do not hesitate to contact the author at bjorn.nyberg@uib.no.