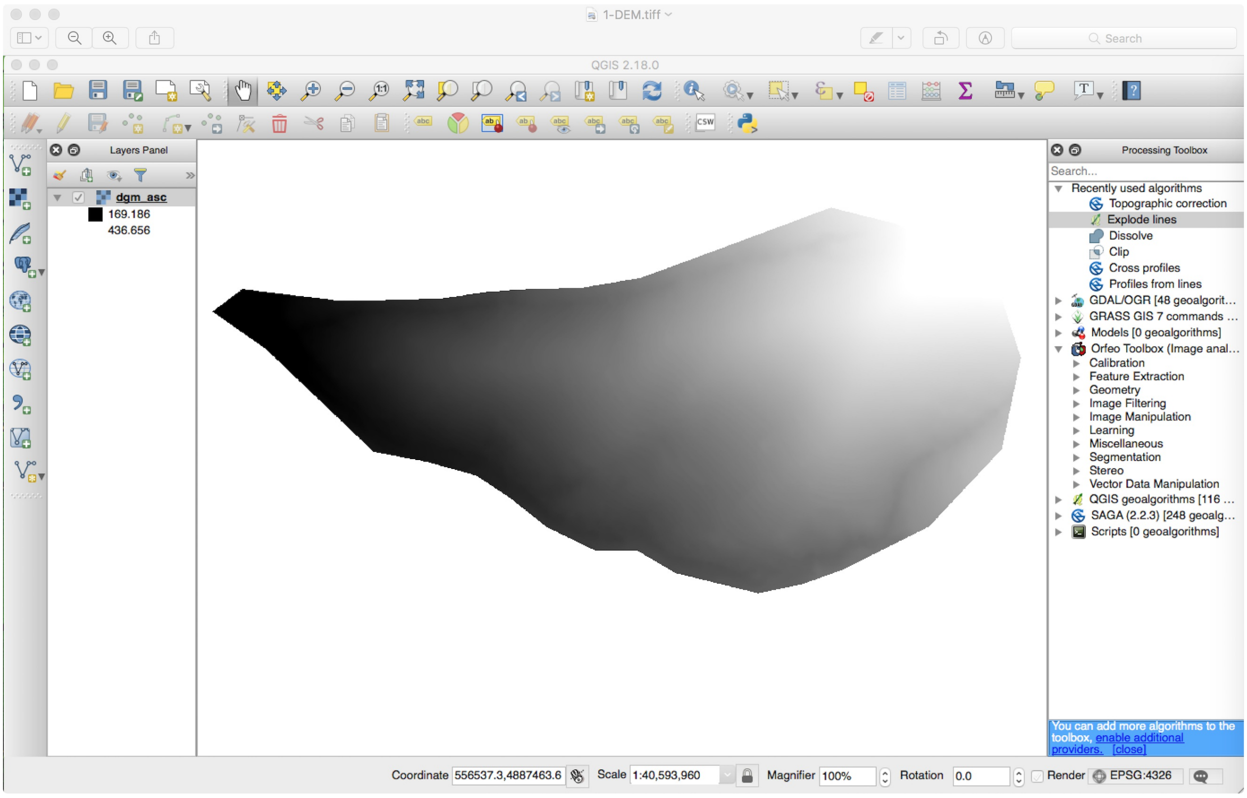
The purpose of this tutorial is to extract terrace riser walls from a Digital Elevation Model using only open-source software. This tutorial is based on steps for extracting terrace walls from a DEM using ArcGIS, which you can find here . These instructions use QGIS 2.18.0 and a command line tool called Centerline. QGIS can be downloaded from download.qgis.org. Centerline can be downloaded from github or downloaded using PyPI (see github page for details).

Here are the basic steps:

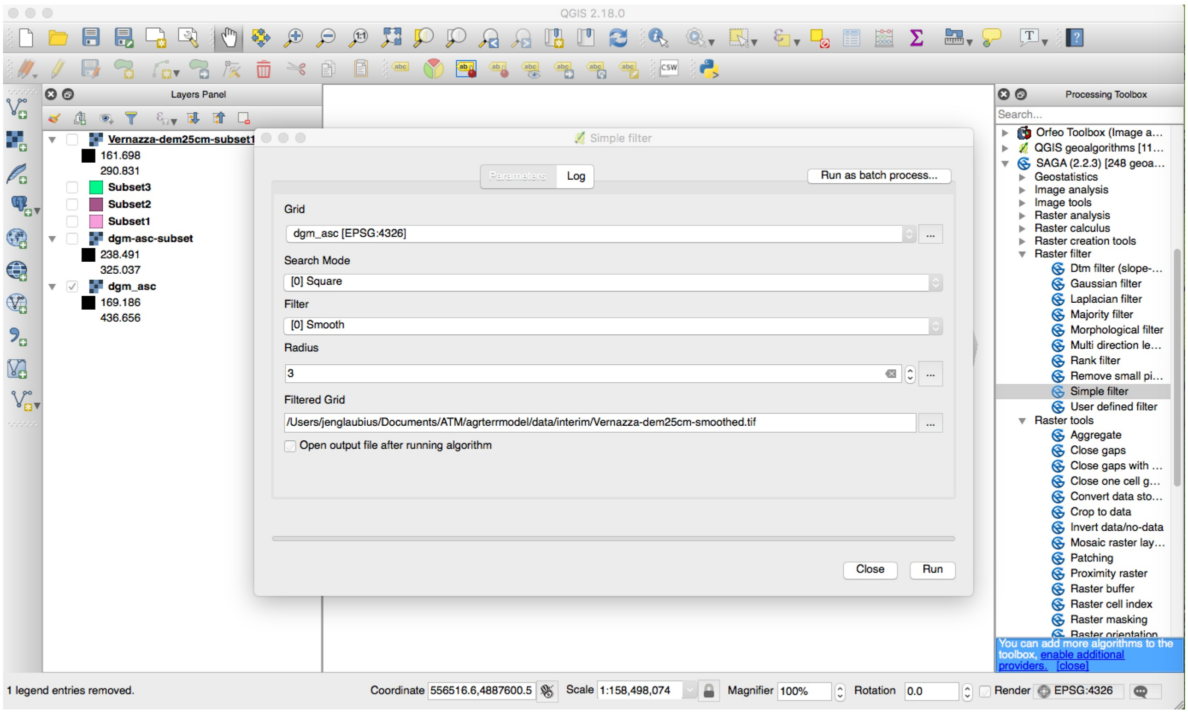
1. Calculate profile curvature to find areas of high curvature (i.e. the riser walls).
2. Convert the areas of high curvature (polygons) into lines.
3. Manually clean the resulting lines.

Ready to get started?

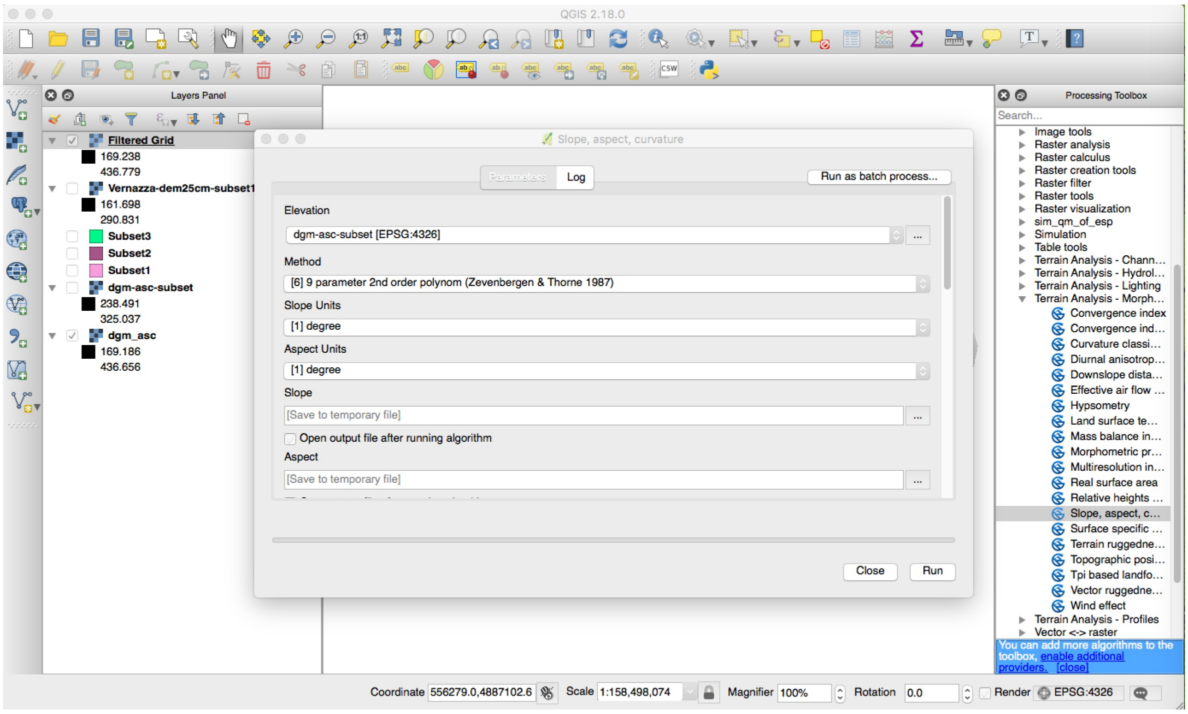
Open QGIS and open your DEM file



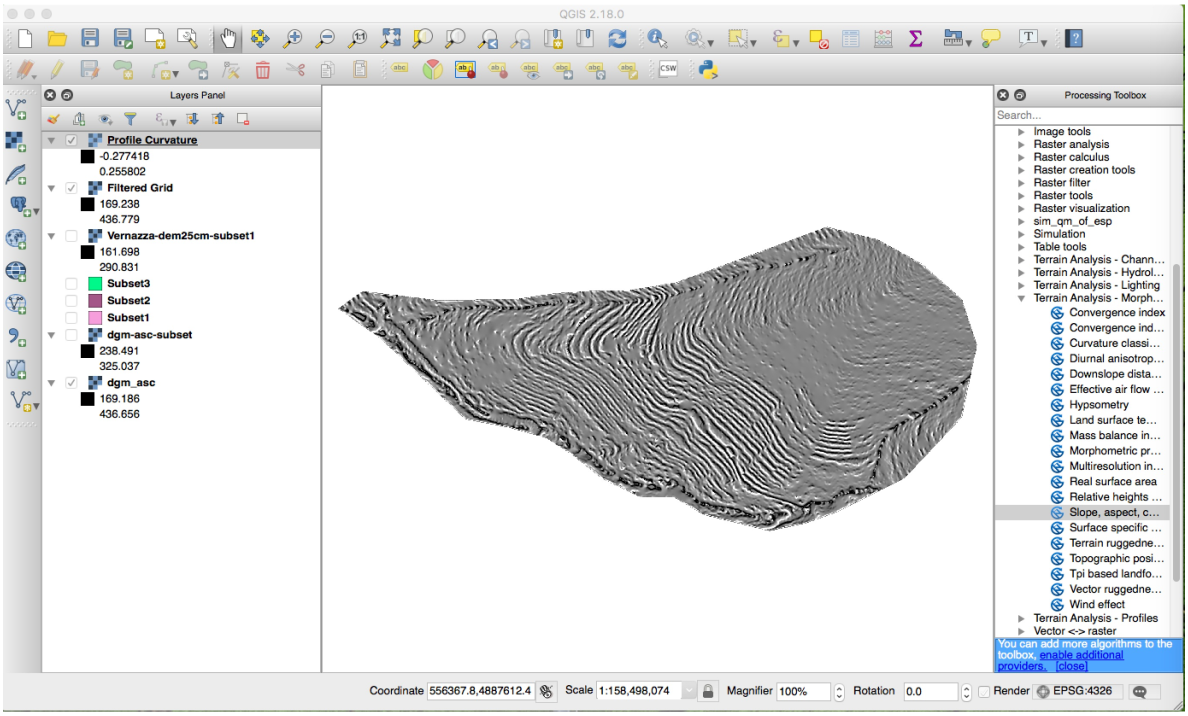
1. Smooth the DEM: SAGA -> Raster filter -> Simple filter



2. Compute profile curvature of smoothed DEM: SAGA -> Terrain Analysis - Morphology -> Slope, aspect, curvature

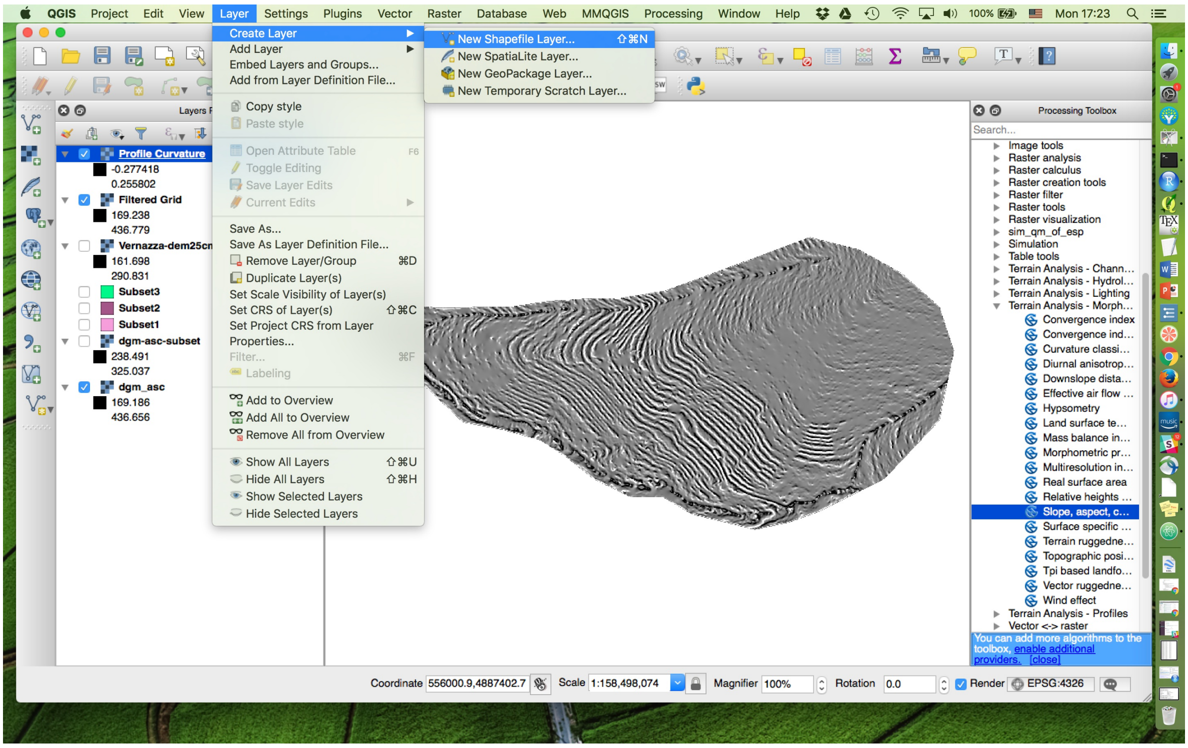


Result of profile curvature

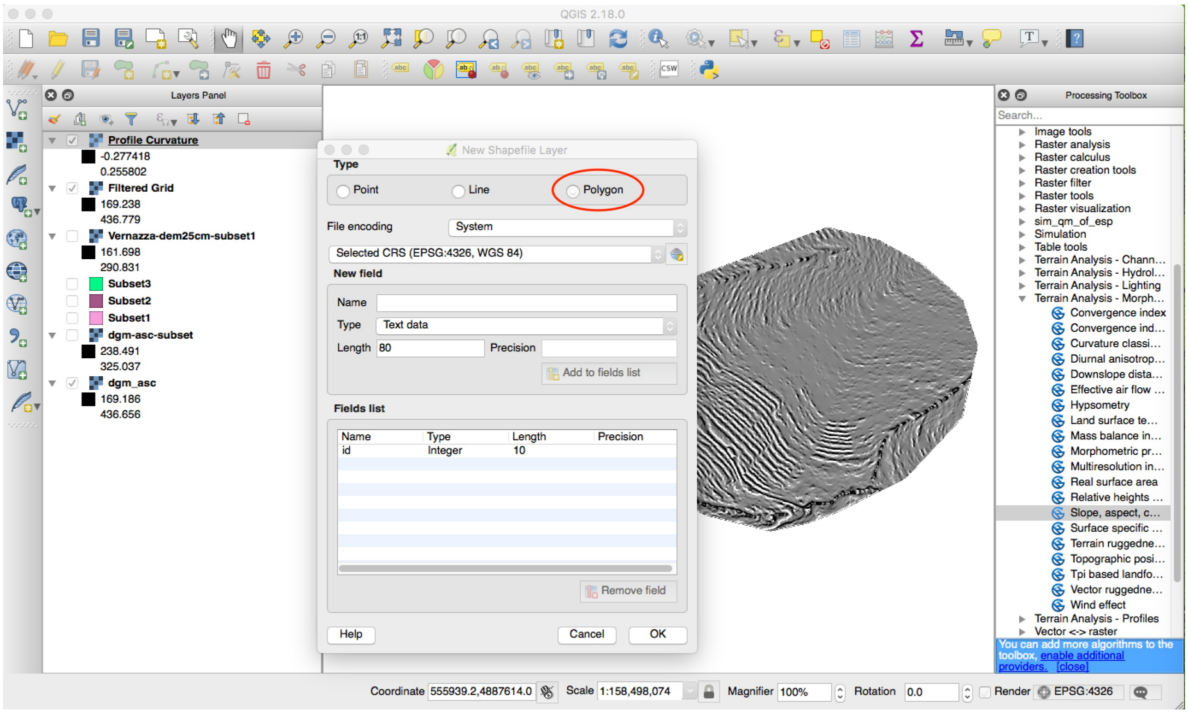


3. Crop profile curvature layer to exclude outside edges:

   a) Create a mask layer for the clipping: Layer -> Create Layer -> New Shapefile Layer

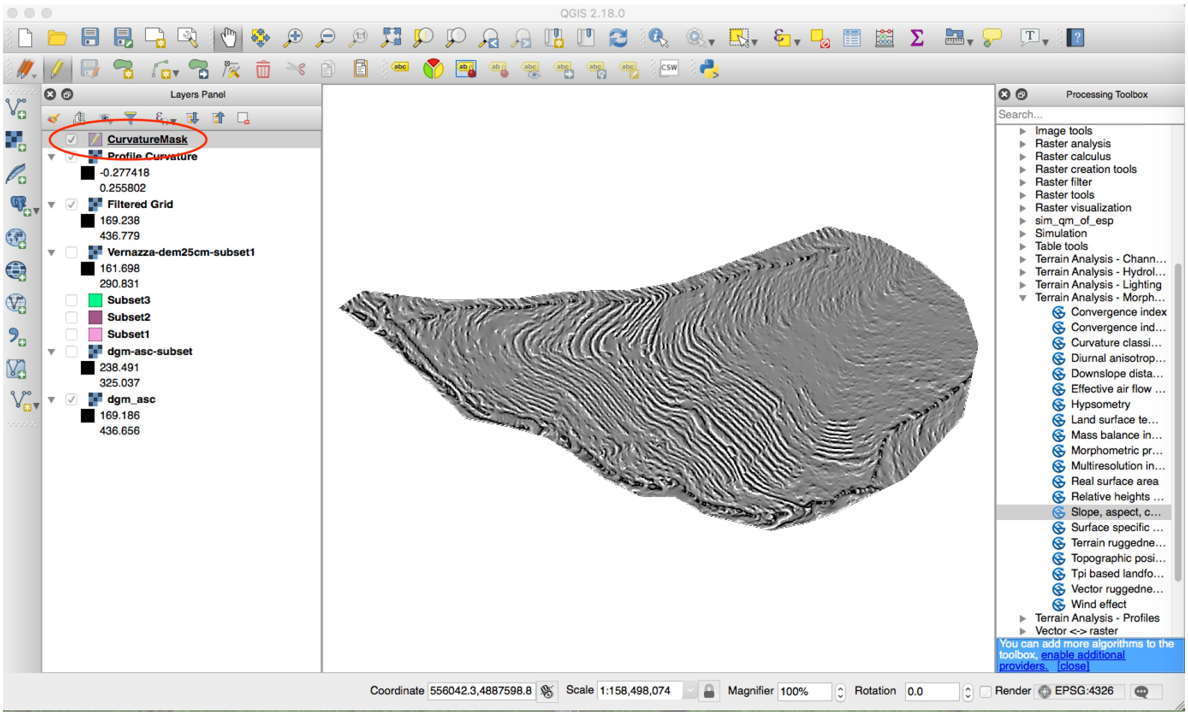


. Be sure to create a polygon and not a point layer

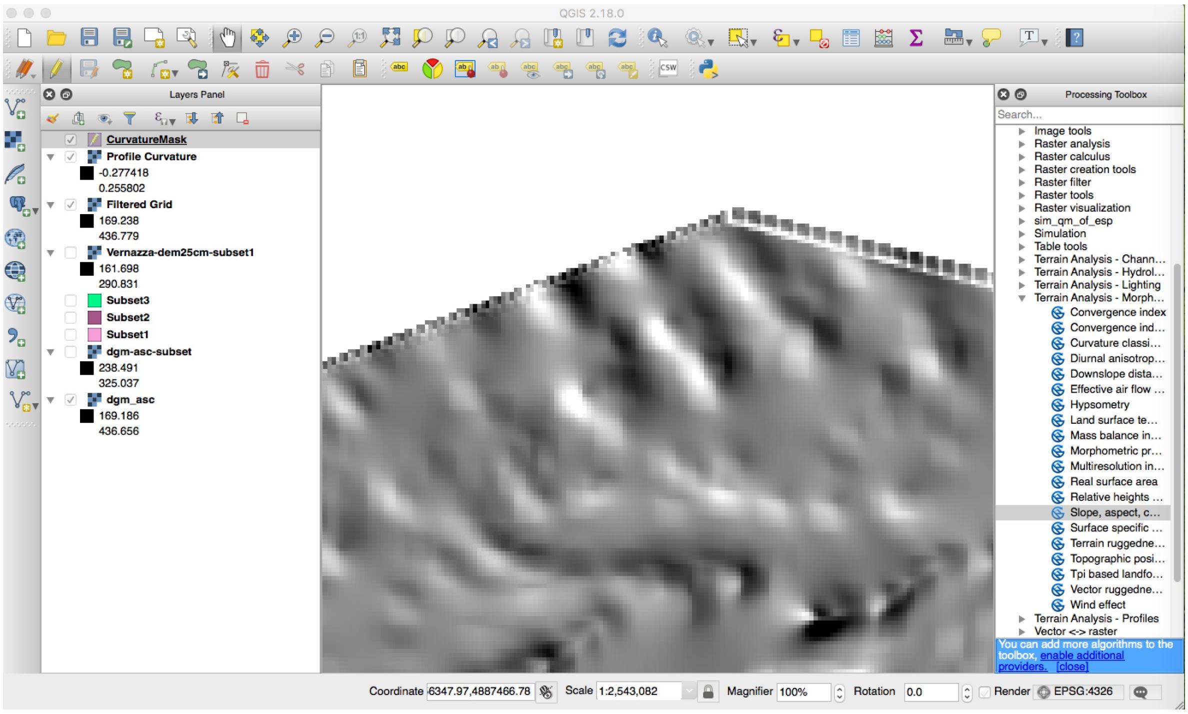


.

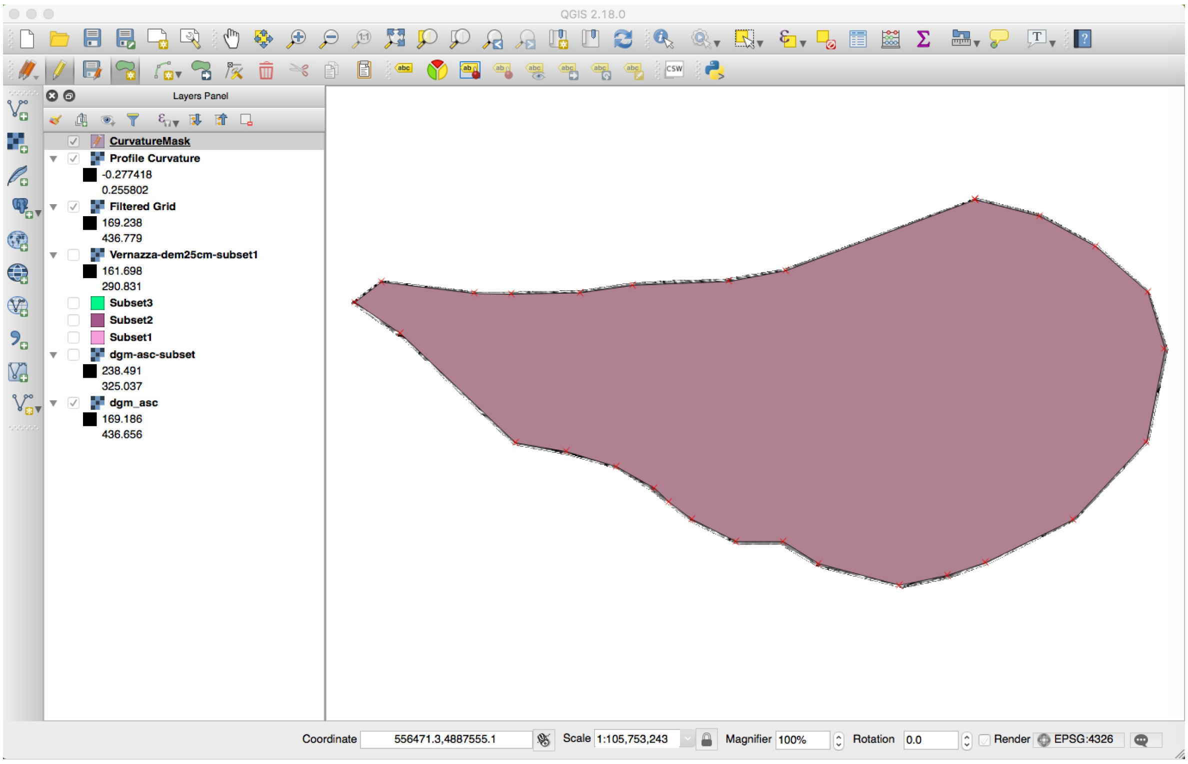
b) Set the new layer to editing



. Zoom in and you will see why you want to clip the edge

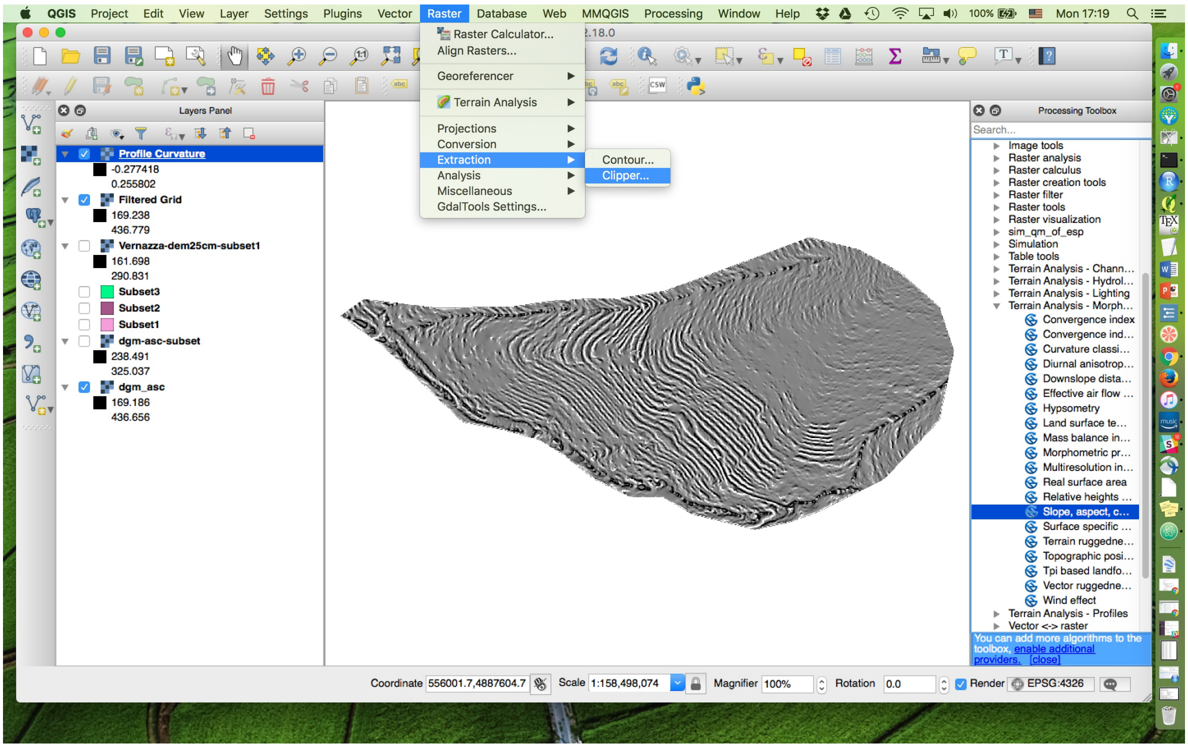


and then zoom out to start creating a new feature

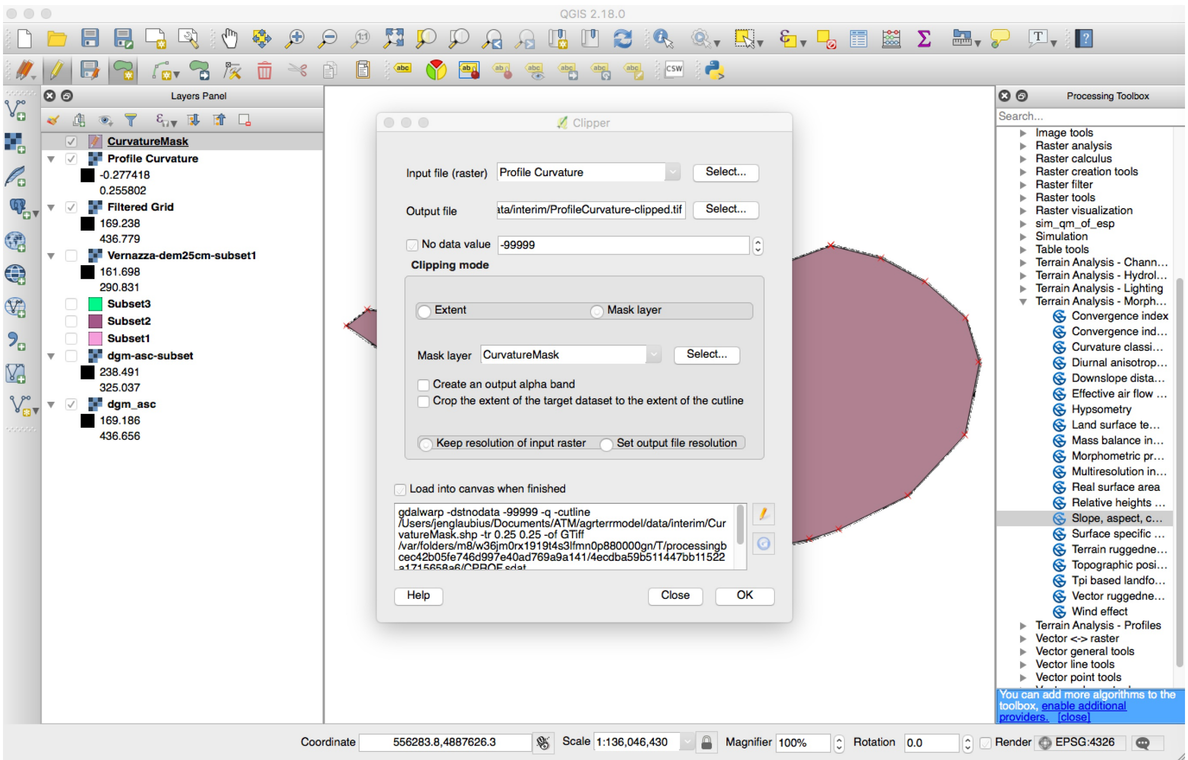


. Be sure to save the new polygon shapefile. Then turn off editing.

c) Raster -> Extraction -> Clipper to crop the Profile Curvature layer to the mask layer



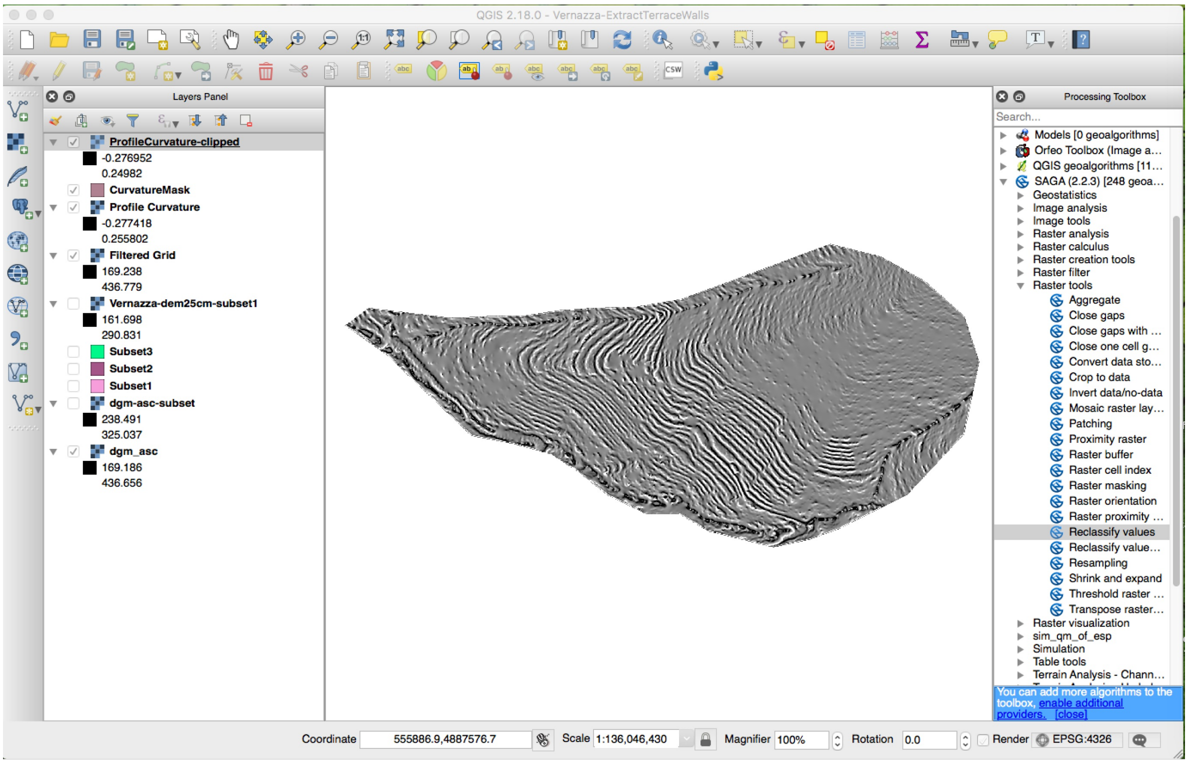
. Clipper window



.

4. Reclassify the clipped profile curvature layer into 2 classes: 1) wall (high curvature) and 2) non-wall (low curvature). You may need to experiment with the threshold between the 2 classes.

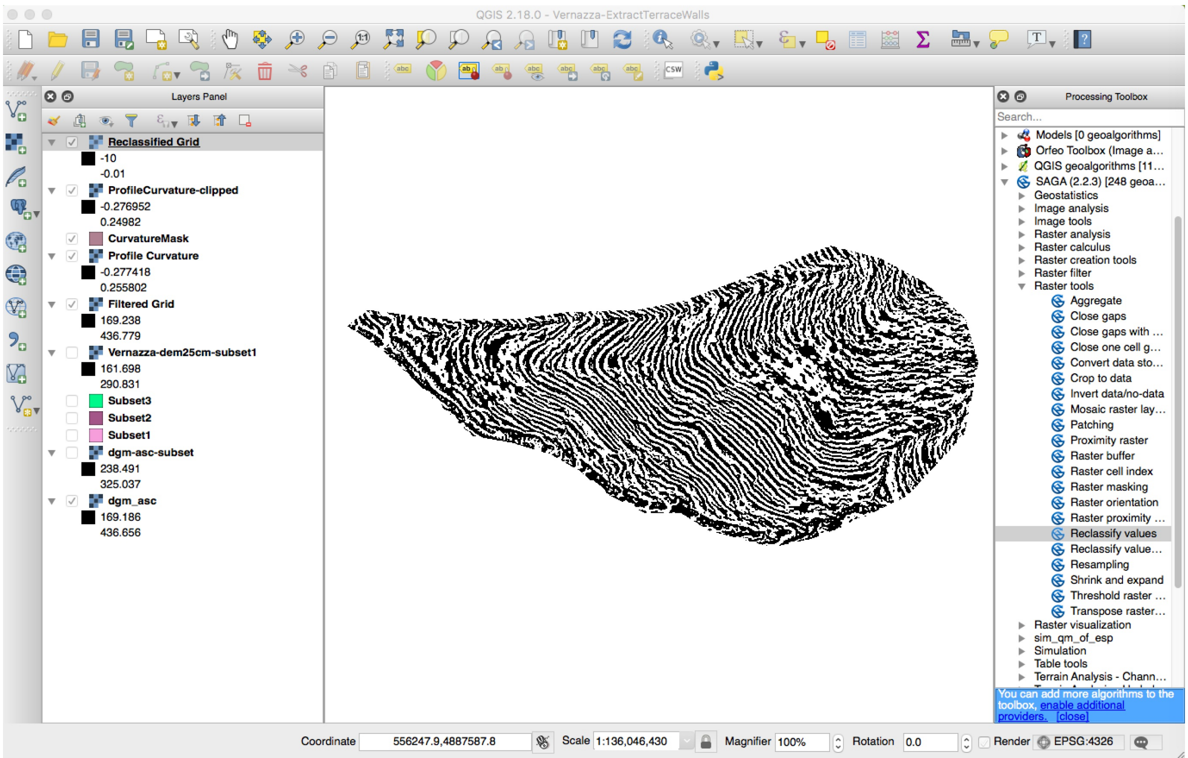
   a) Open the Reclassify values tool: SAGA -> Raster tools -> Reclassify values



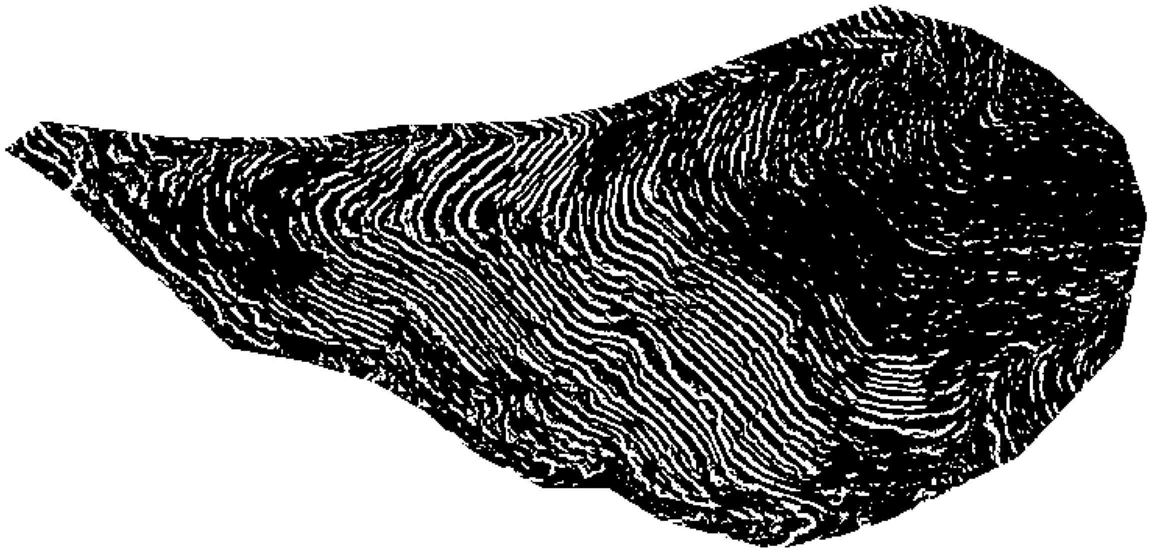
.

b) In the Reclassify values window: Grid = clipped Profile Curvature layer; Method = [1] range; minimum value (for range) = less than minimum for curvature layer; maximum value (for range) = threshold value (0.0 to start; 0.05, 0.1, 0.075, 0.025); new value (for range) = -10; operator (for range): [0] <= Reclassify values window. Click Run.

c) Evaluate results

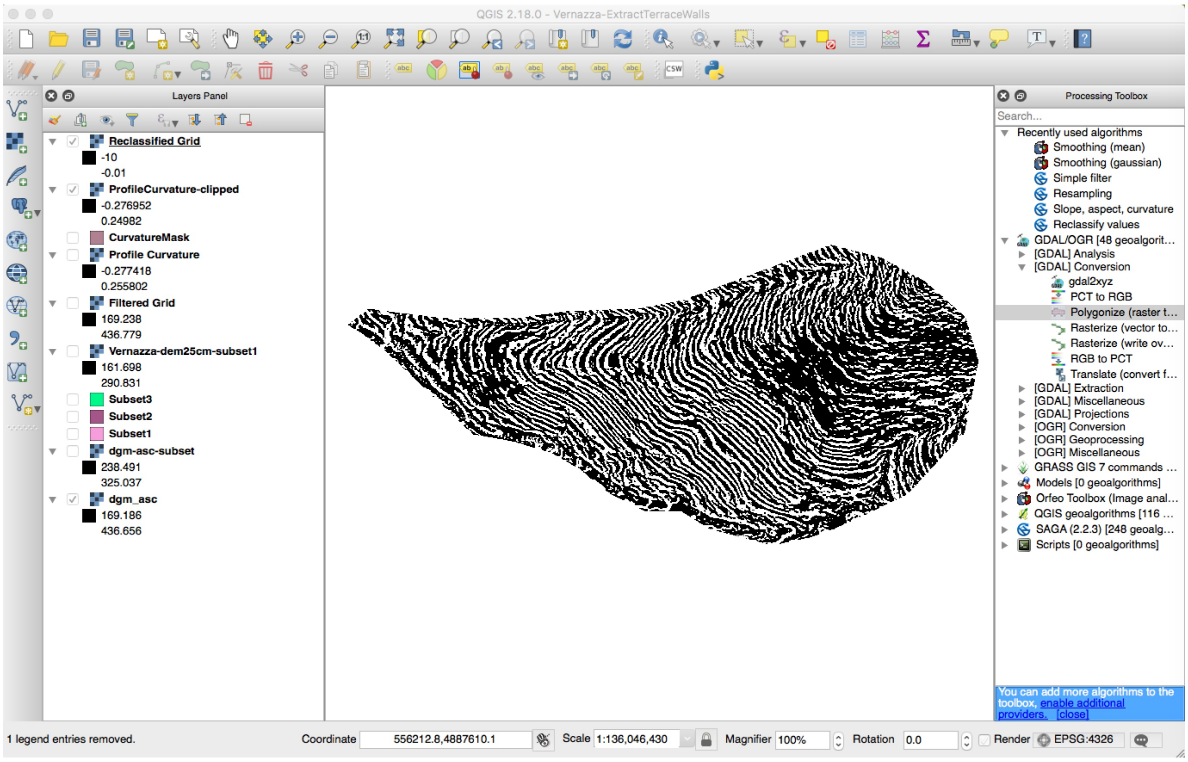


and try other threshold values

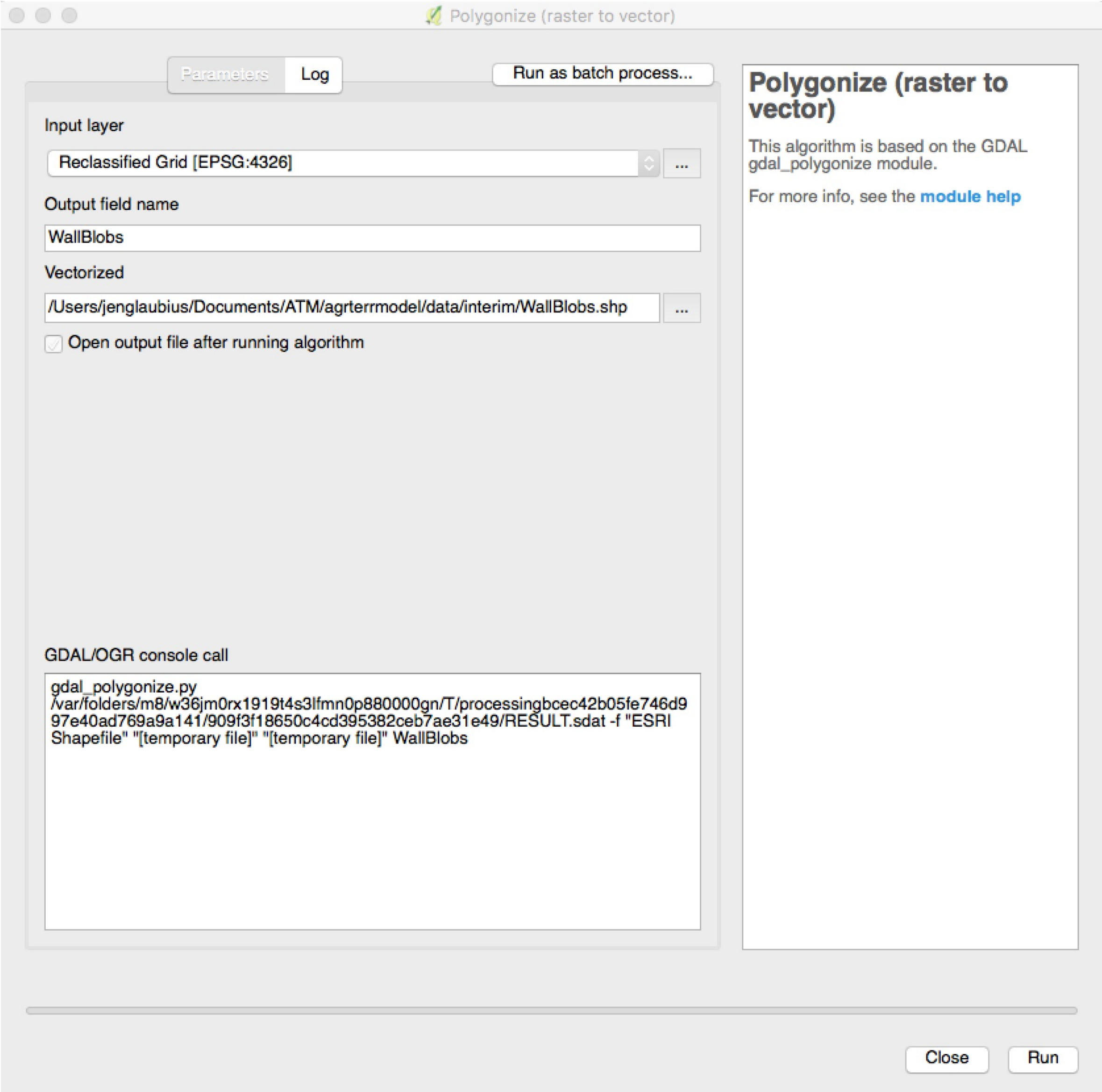


. Choose a reclassified grid that captures as many walls as possible. False walls will be removed in future step.

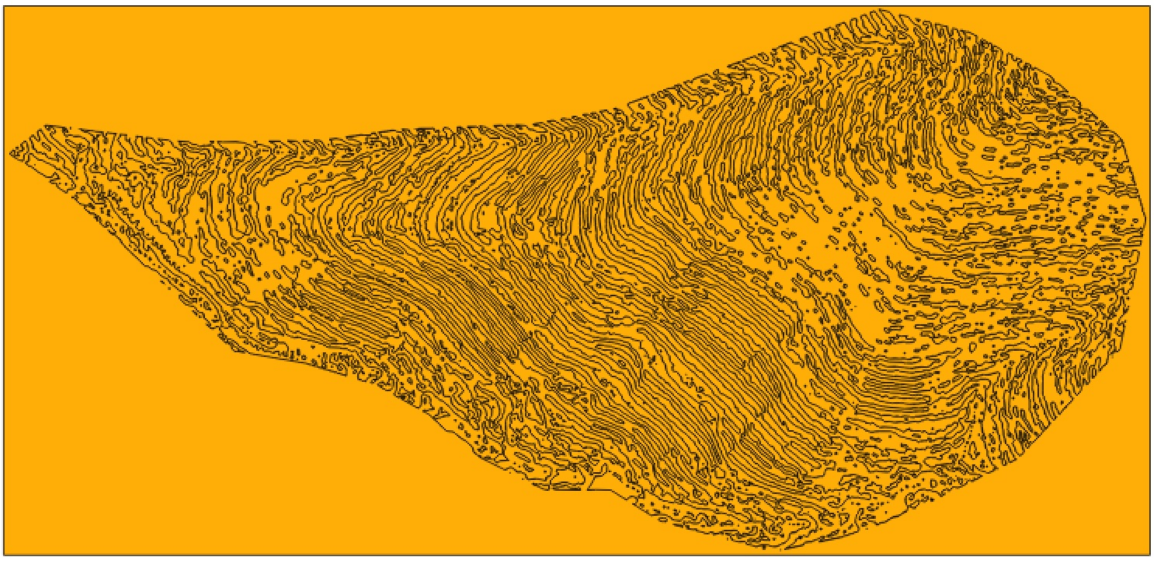
5. Convert the chosen reclassified layer (I used threshold 0.01) from a raster to polygons: GDAL/OGR -> [GDAL] Conversion -> Polygonize (raster to vector)



. Set reclassified grid as Input layer, change Output field name ("WallBlobs"), set file to be saved, and Run



. Results look like this

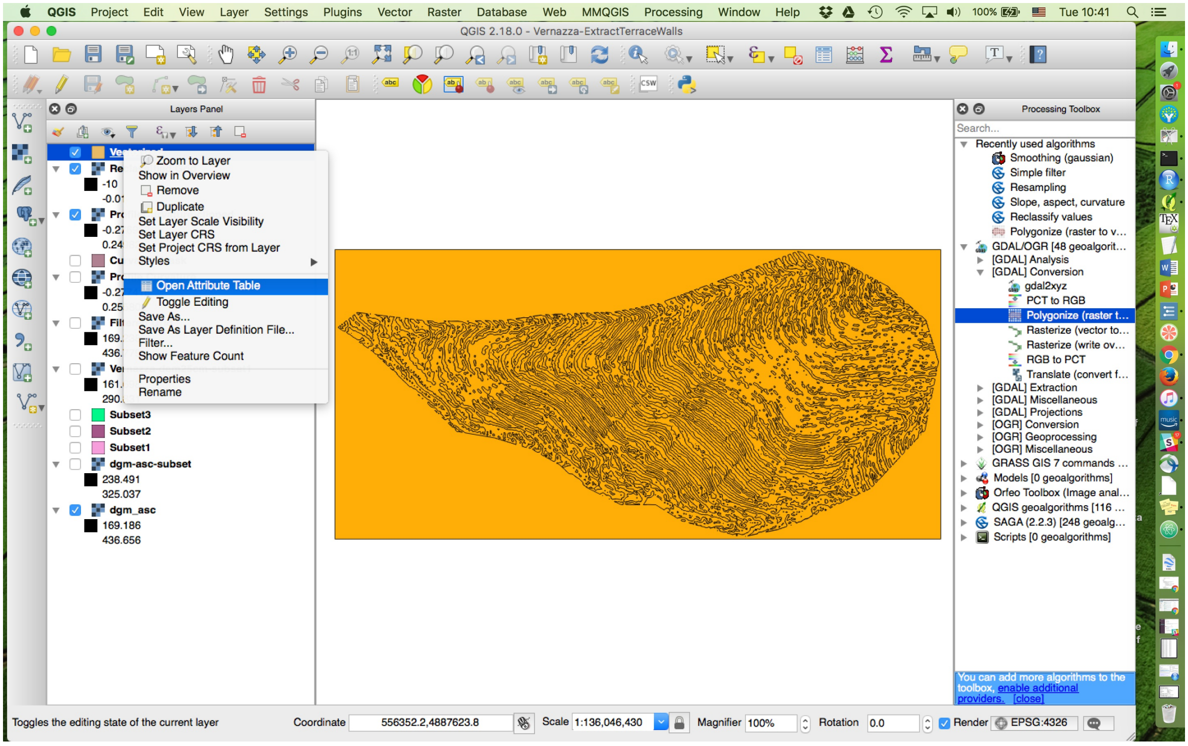


.

6. Clean up the polygons before conversion to lines.

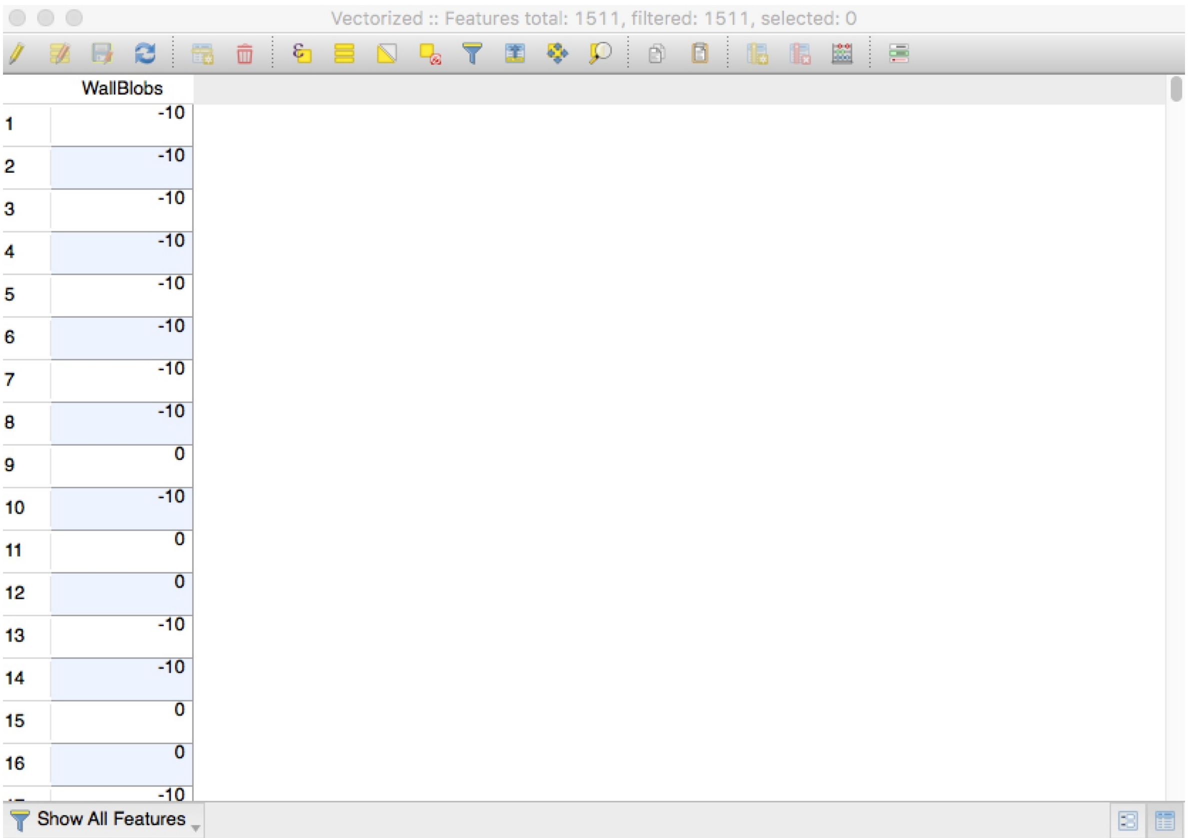
a) Delete non-wall blobs.

i) Open vectorized attribute table by right-clicking on layer in Layers Panel

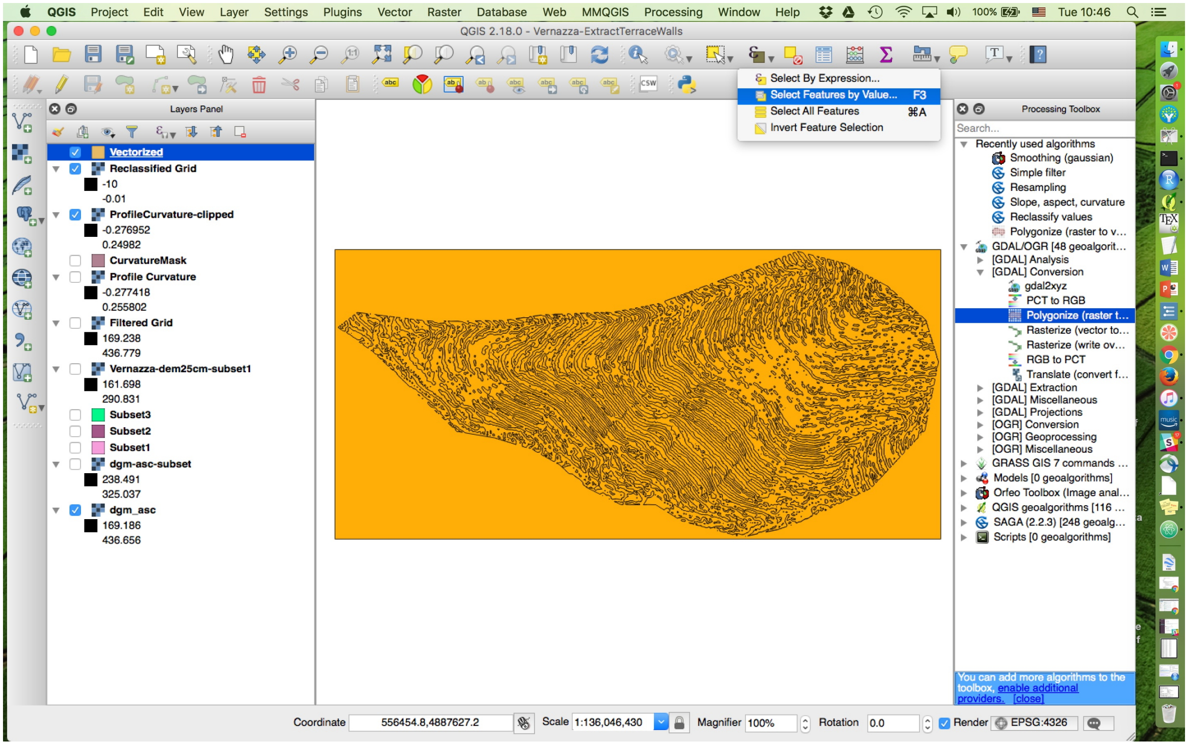


.

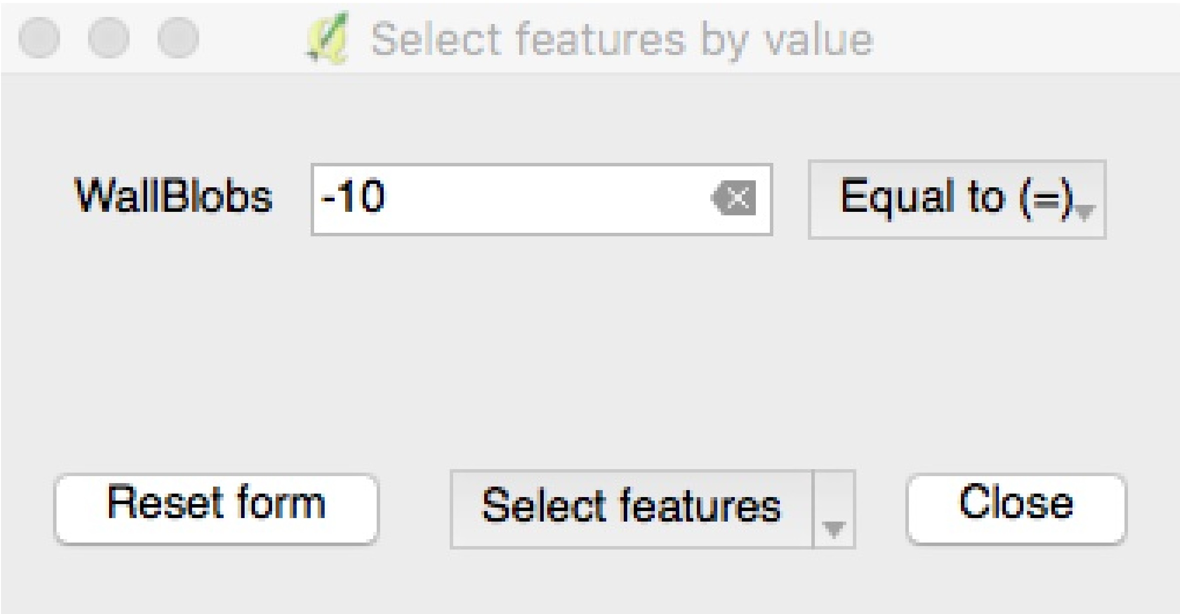
ii) Delete non-wall polygons from the layer. Notice that there are 2 types of WallBlobs: [-10] for non-wall polygons and [0] for wall polygons



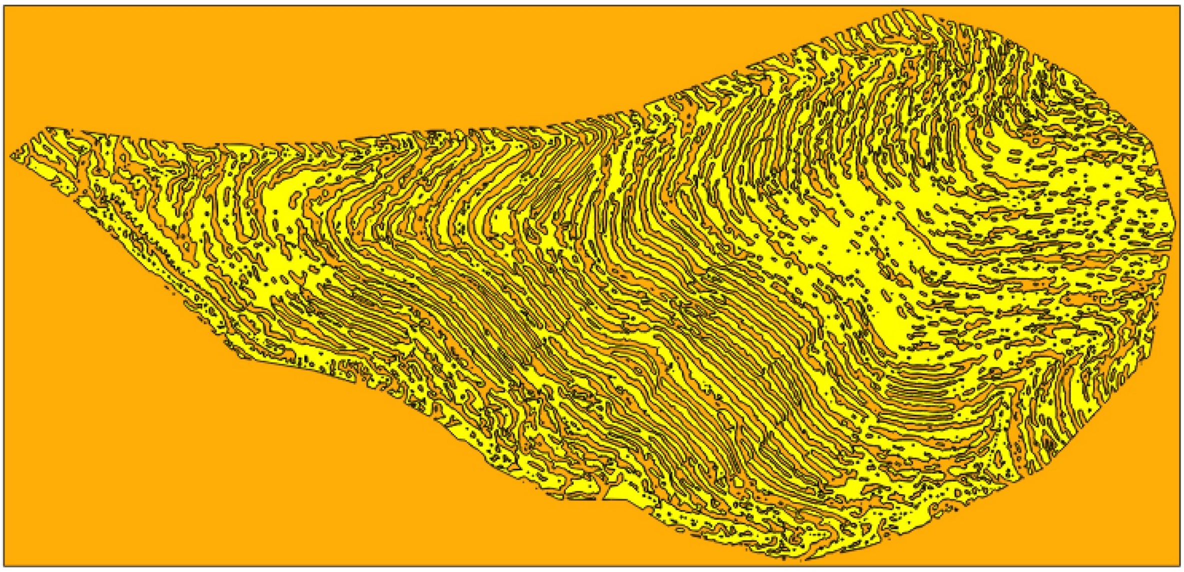
. Select all non-wall polygons using Select features by value



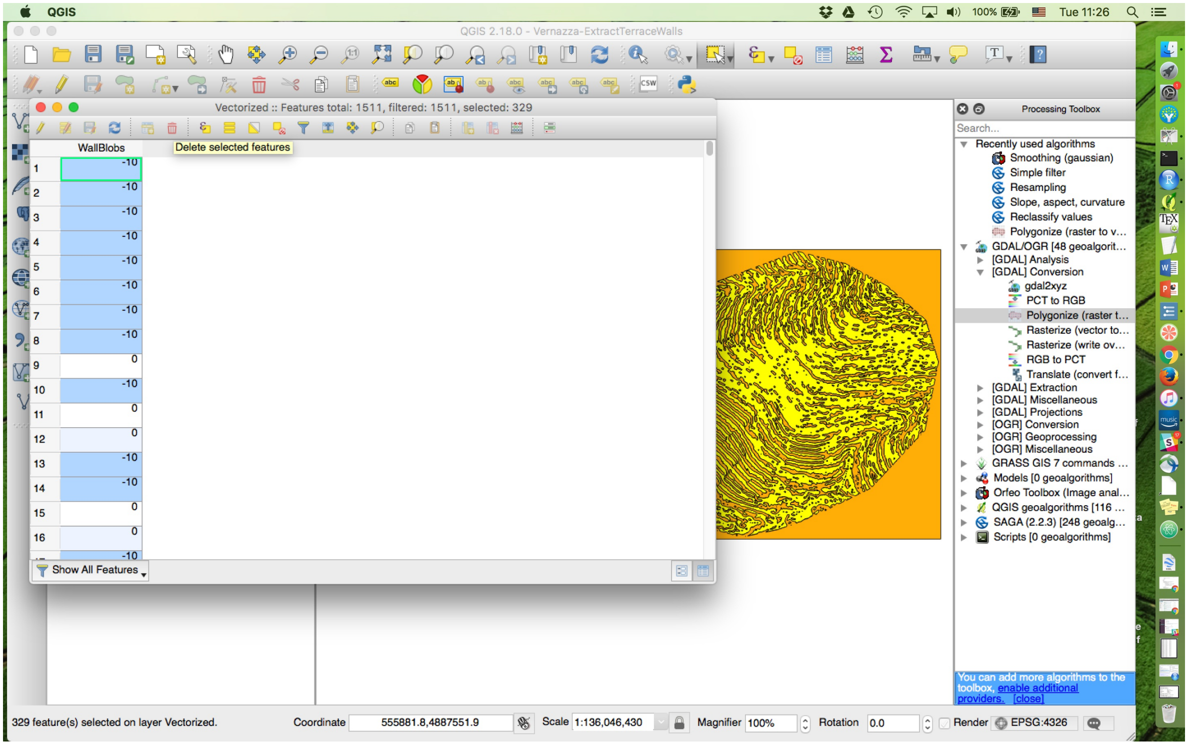
. Select all non-wall polygons (value = -10)



by clicking Select Features, then close the window. Selected non-wall polygons should be yellow

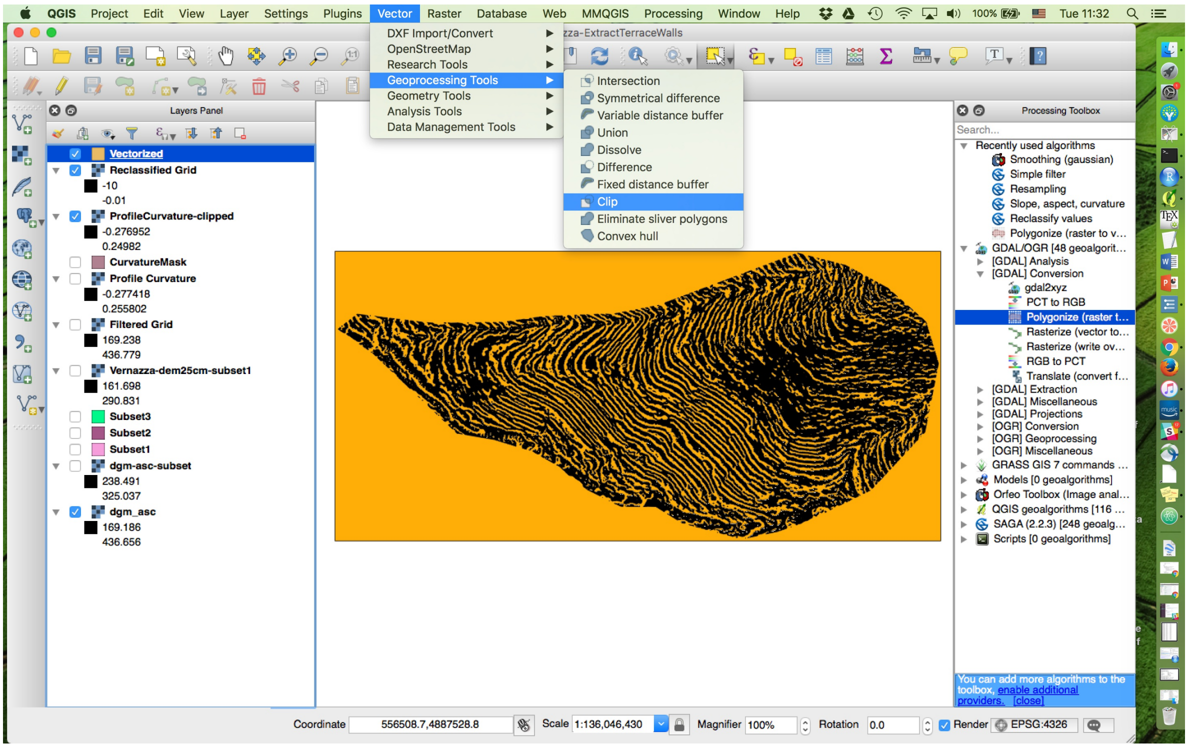


. Reopen the attribute table, if you closed it, turn on editing (Pencil on far left), then click Delete selected features (red trash can)

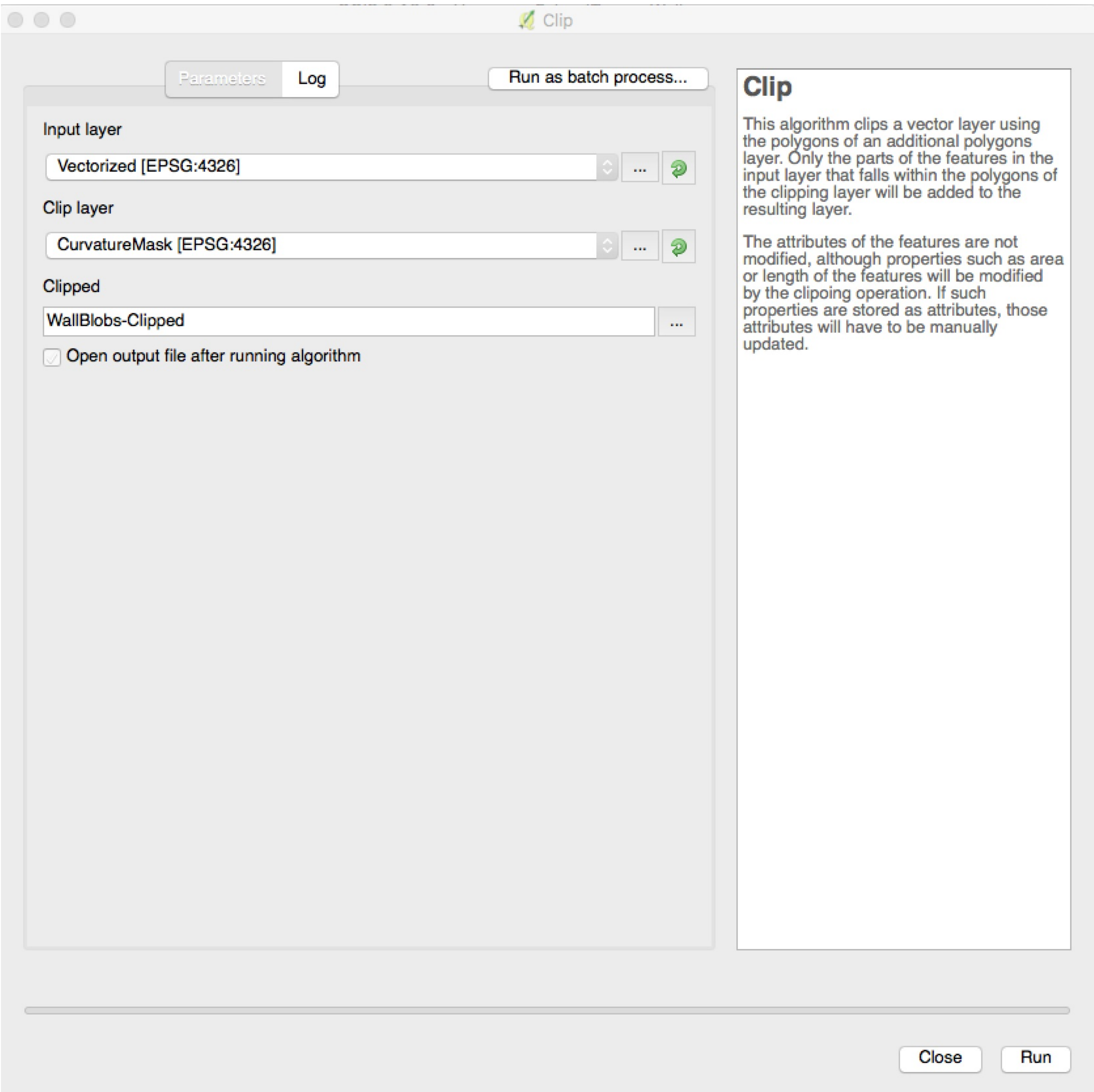


. Save edits.

iii) Clip Vectorized layer to earlier Curvature Mask layer. Vector -> Geoprocessing Tools -> Clip

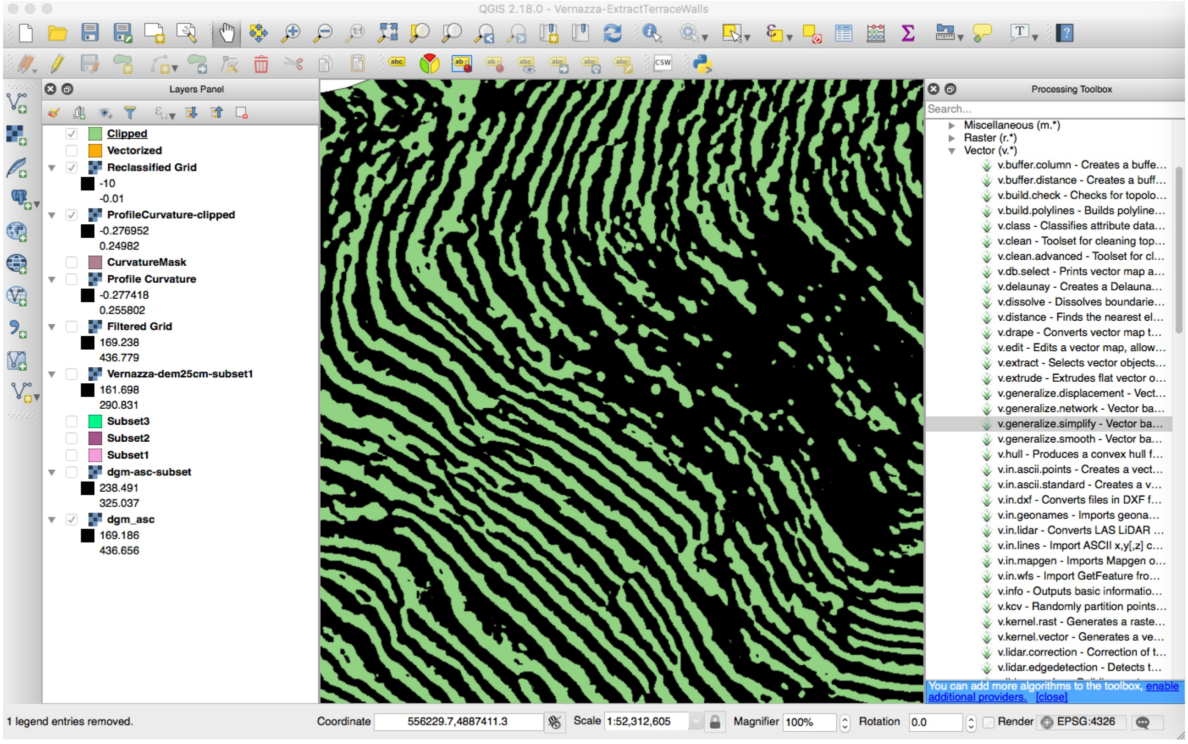


. Input layer = Vectorized; Clip layer = CurvatureMask, and set name for Clipped layer, then click "Run"

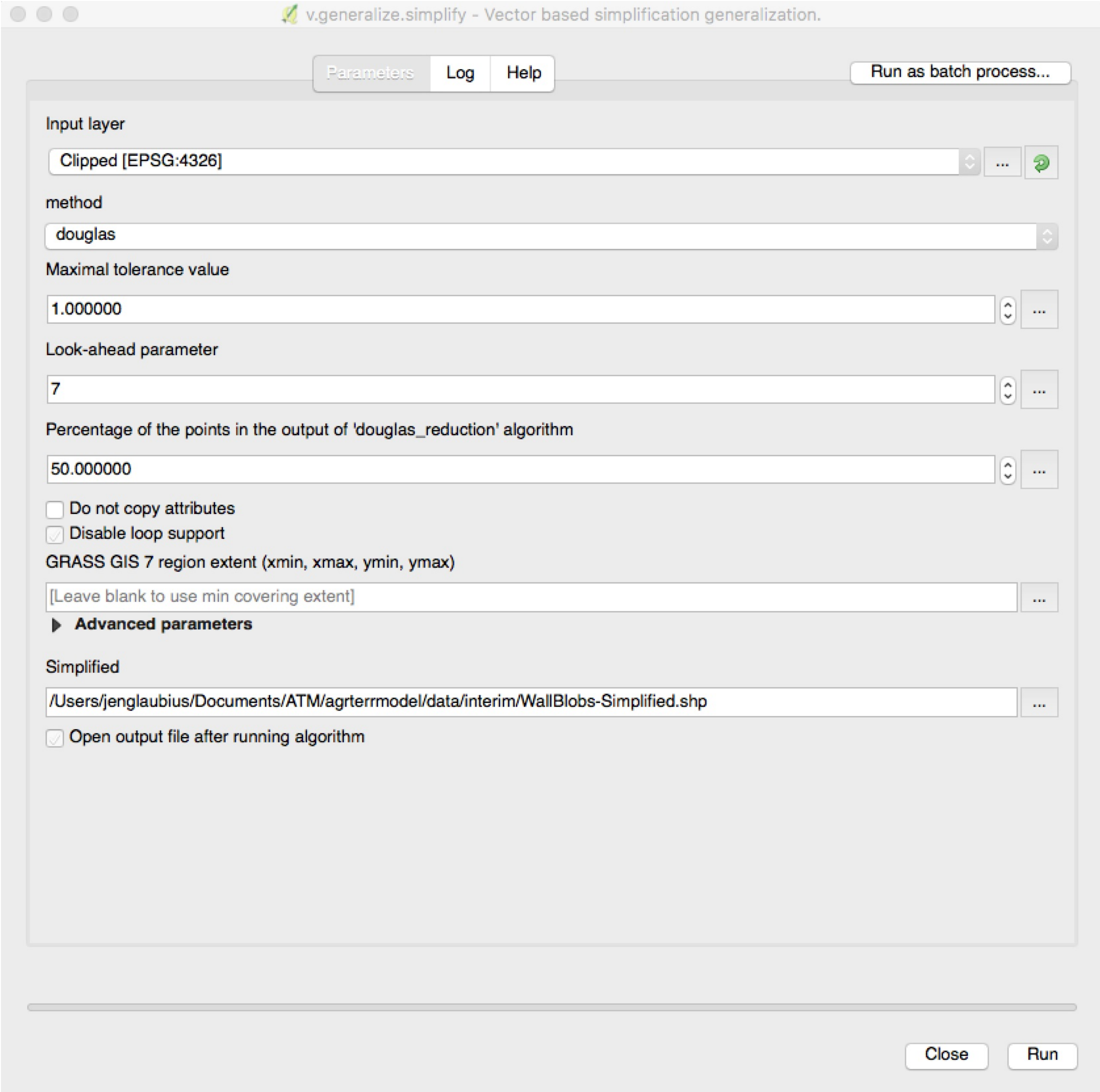


.

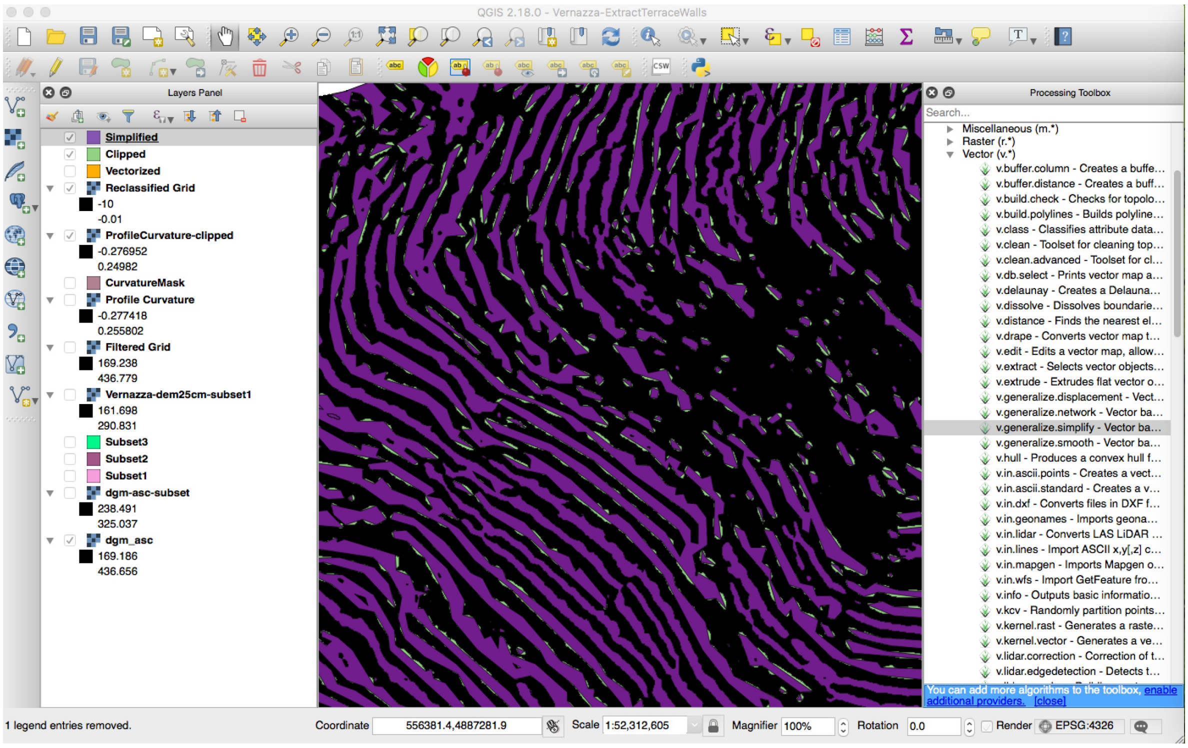
b) Simplify wall blob polygons: GRASS -> Vector -> v.generalize.simplify



. In the v.generalize.simplify window, choose the method (I used the default "douglas" because polygon boundaries were simplified to straight lines), save a file, and Run



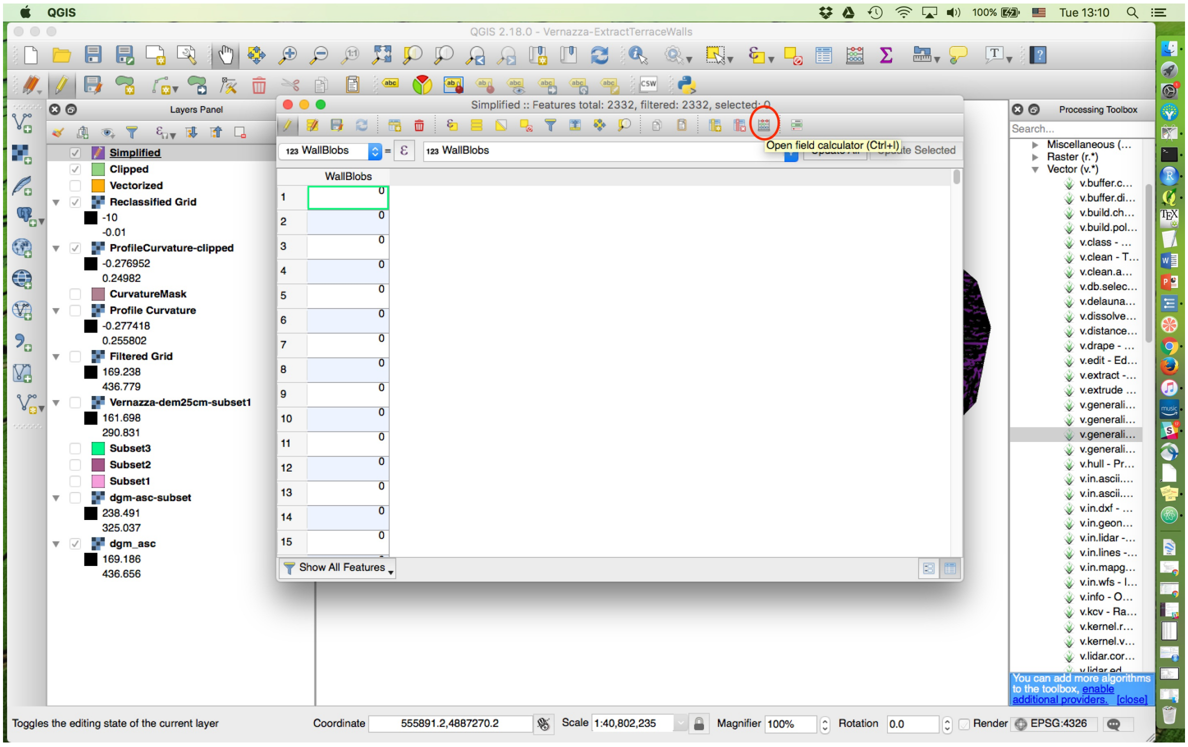
. The resulting simplified layer has straight lines



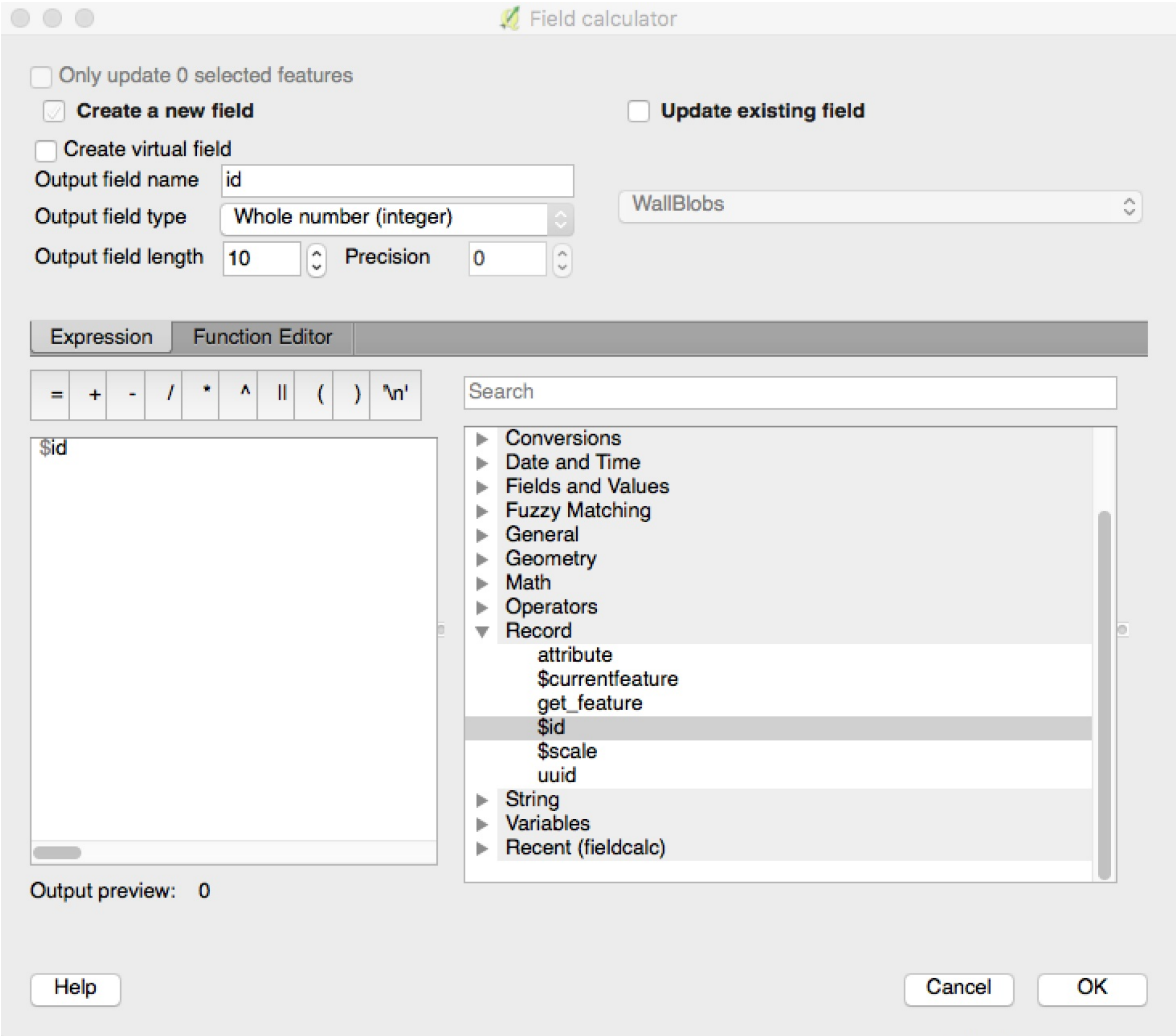
.

c) Delete very small polygons.

i) Add id and area fields to polygons. Open the attribute table for the Simplified layer. You will need unique id numbers for lines created from the polygons later on. So add an id field using the field calculator

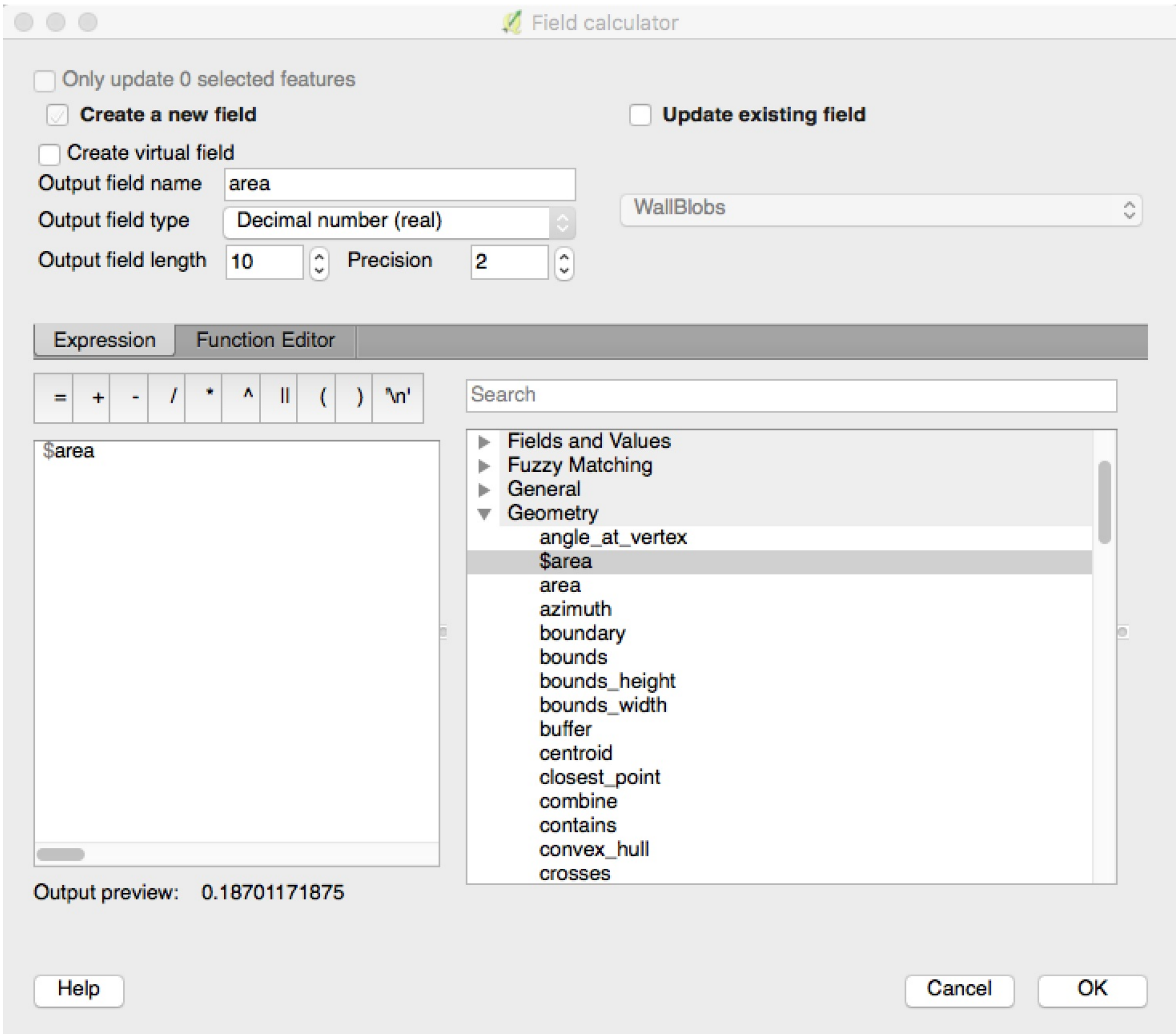


. In the field calculator, make sure that the "Create a new field" box is checked. Add an "Output field name" (e.g. "id"), double click on $id in Record to add it to the expression, then click "OK"

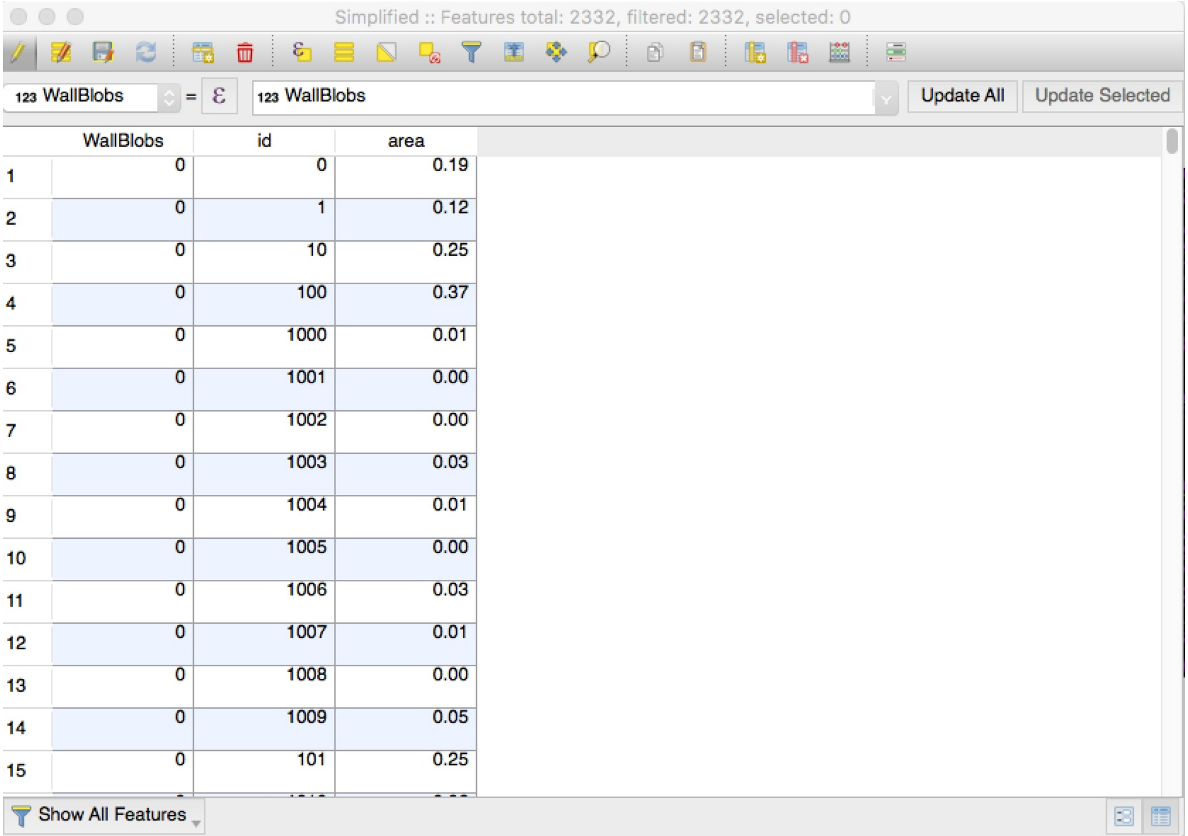


.

ii) Repeat to create a field for area, except use Geometry -> $area in the expression



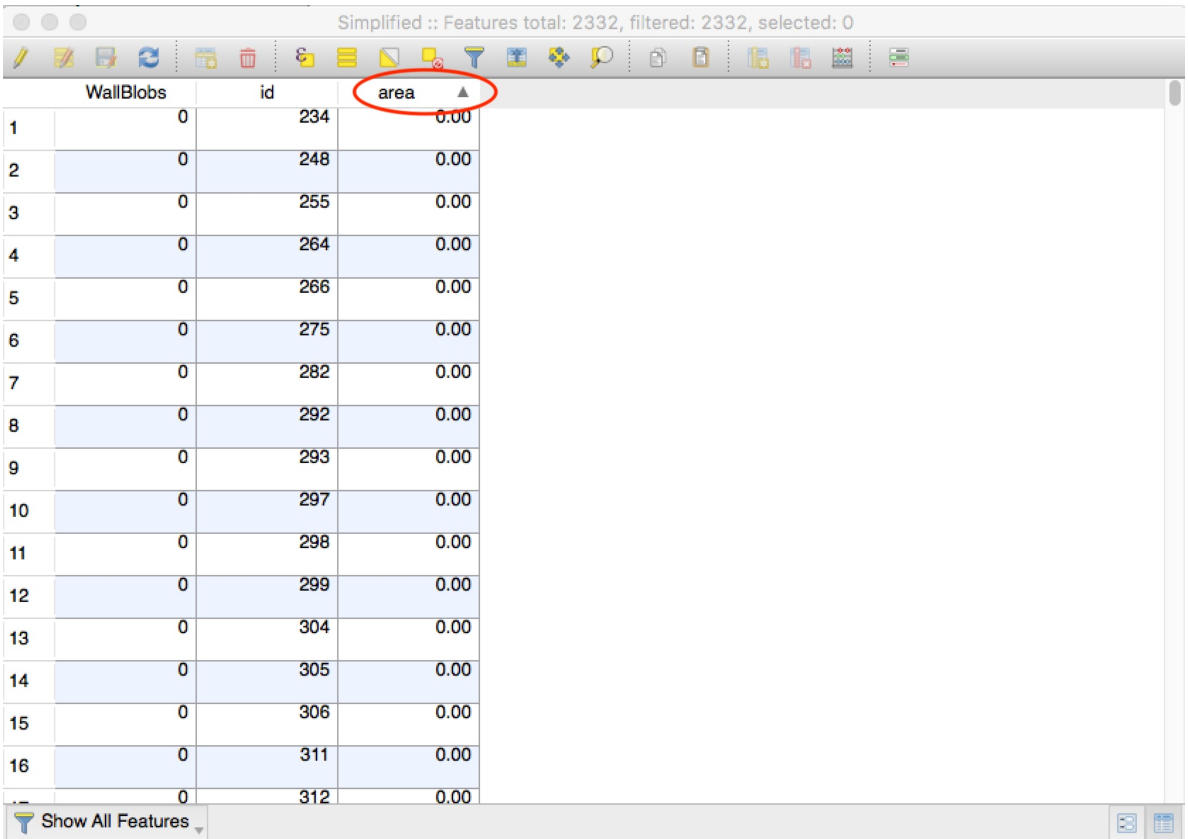
. Attribute table should now have 3 fields



	WallBlobs	id	area
1	0	0	0.19
2	0	1	0.12
3	0	10	0.25
4	0	100	0.37
5	0	1000	0.01
6	0	1001	0.00
7	0	1002	0.00
8	0	1003	0.03
9	0	1004	0.01
10	0	1005	0.00
11	0	1006	0.03
12	0	1007	0.01
13	0	1008	0.00
14	0	1009	0.05
15	0	101	0.25

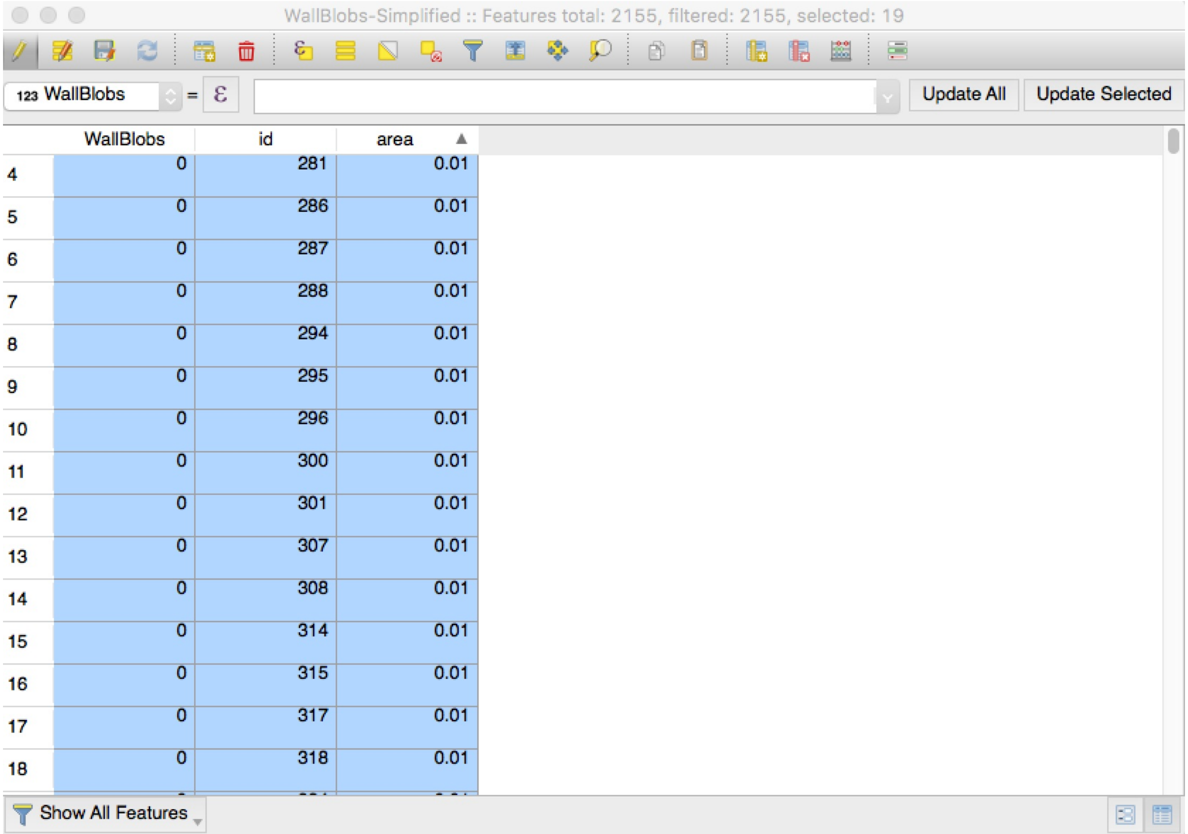
. Save edits.

iii) Delete small polygons. In the attribute table, click on the area field to sort polygons by size from small to large



	WallBlobs	id	area
1	0	234	0.00
2	0	248	0.00
3	0	255	0.00
4	0	264	0.00
5	0	266	0.00
6	0	275	0.00
7	0	282	0.00
8	0	292	0.00
9	0	293	0.00
10	0	297	0.00
11	0	298	0.00
12	0	299	0.00
13	0	304	0.00
14	0	305	0.00
15	0	306	0.00
16	0	311	0.00
	0	312	0.00

. Highlight rows that have small polygons (< 5.0 m2 for my example) and click the "Delete selected features" button to delete



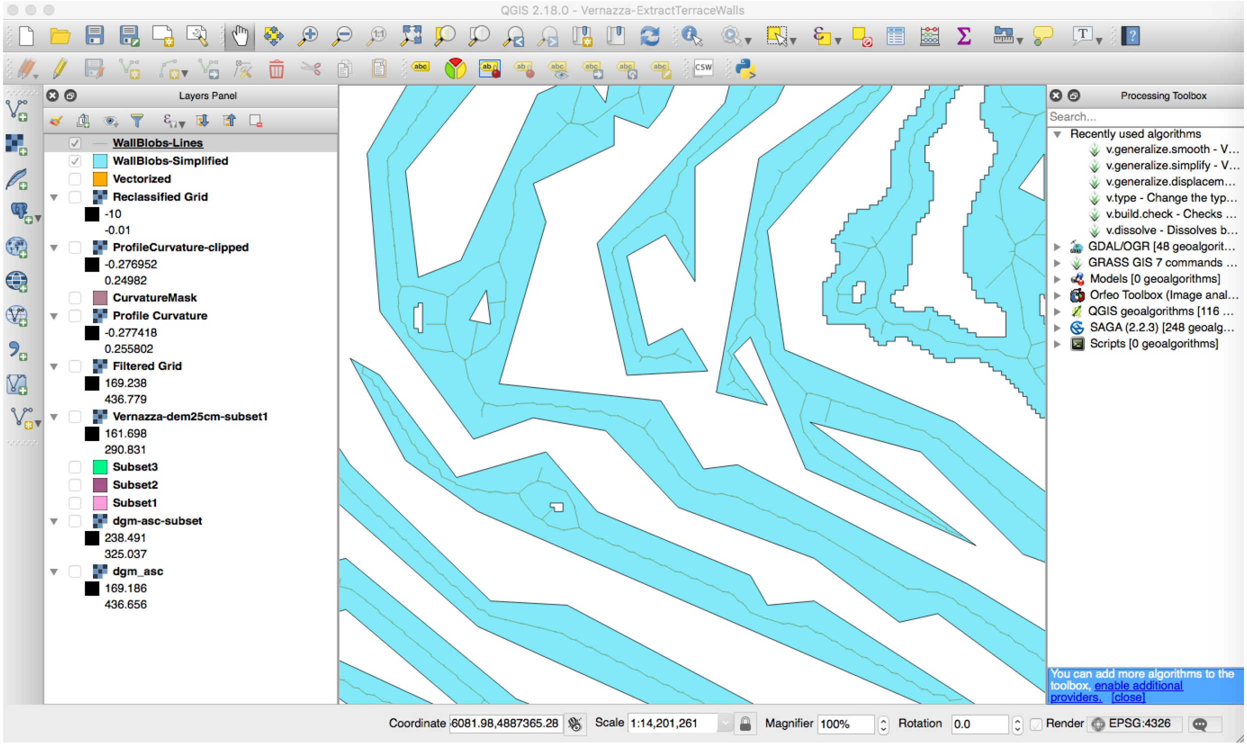
. I suggest deleting a small number of polygons at a time and saving edits often.

7. Skeletonize the polygons to create lines using Centerline command line tool. Before installing, make sure you have a Python2 environment. If you do not already have Python, I recommend using miniconda (https://conda.io/miniconda.html). Here is some information about managing miniconda (https://conda.io/docs/using/envs.html). Create an environment using Python 2x, not Python 3x. Then install numpy, scipy, and GDAl/OGR (use "conda install numpy"). Once all of the dependencies have been installed, install Centerline "pip install centerline".

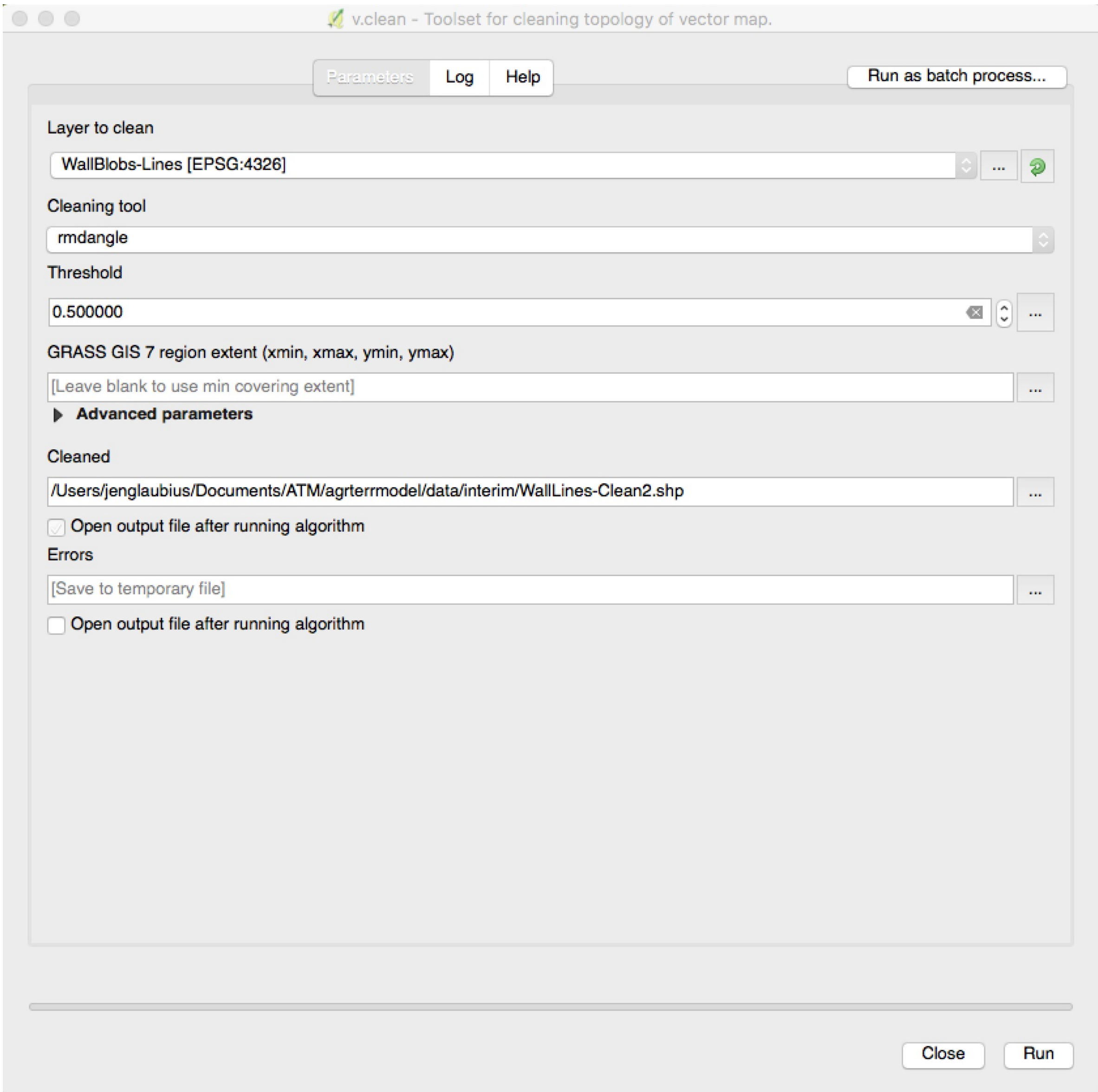
Open a Terminal window (if using Mac) or Command Line (Windows). Navigate to location of Simplified Layer (cd path/to/file/location). Use the command "shp2centerline polygon.shp lines.shp 1.0"

```
(python2) Eastern-Box:interim jenglaubius$ shp2centerline WallBlobs-Simplified.shp WallBlobs-Lines.shp 1.0
Importing polygons from: WallBlobs-Simplified.shp
Calculating centerlines.
Exporting centerlines to: WallBlobs-Lines.shp
Calculation complete.
```

. The command will let you know when it is finished. Open the new shapefile in QGIS and zoom in to look at the results.



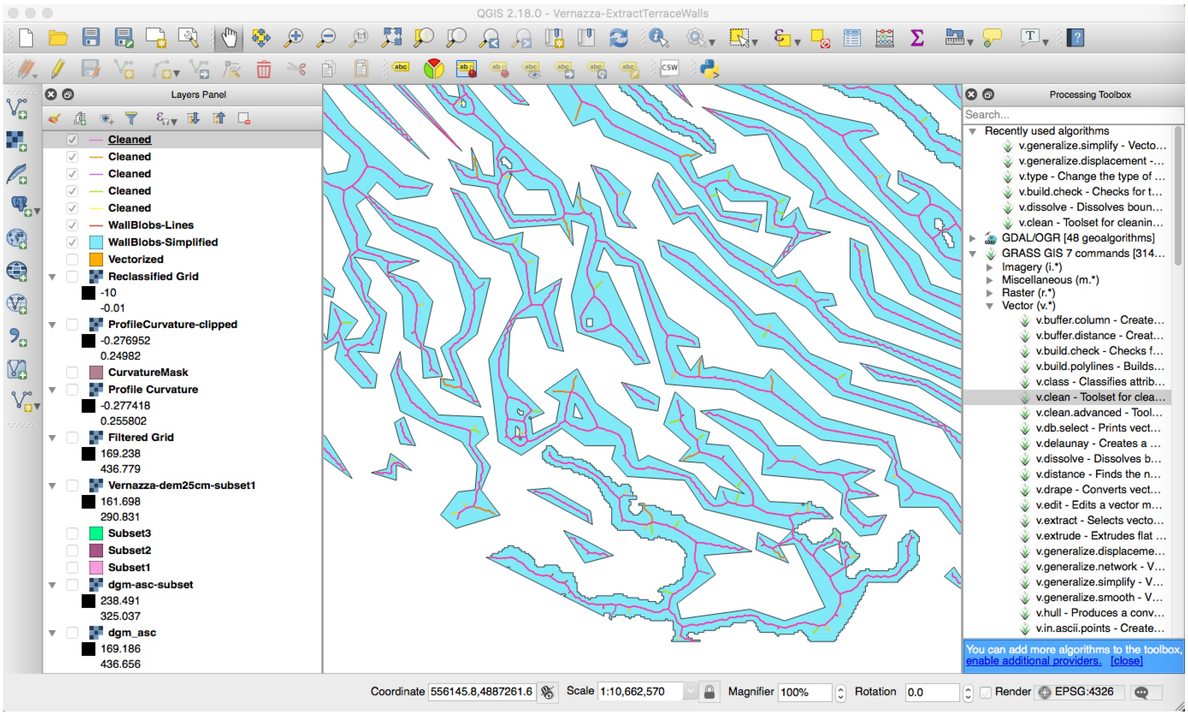
8. Remove small dangles (extraneous lines): Grass -> Vector -> v.clean. Set Layer to clean to your line layer, Cleaning tool = rmdangle, set a threshold for length of lines that may be removed (I started with 0.1, which did virtually nothing), and save the cleaned file



. With each iteration, I used the cleaned file created during the previous iteration.

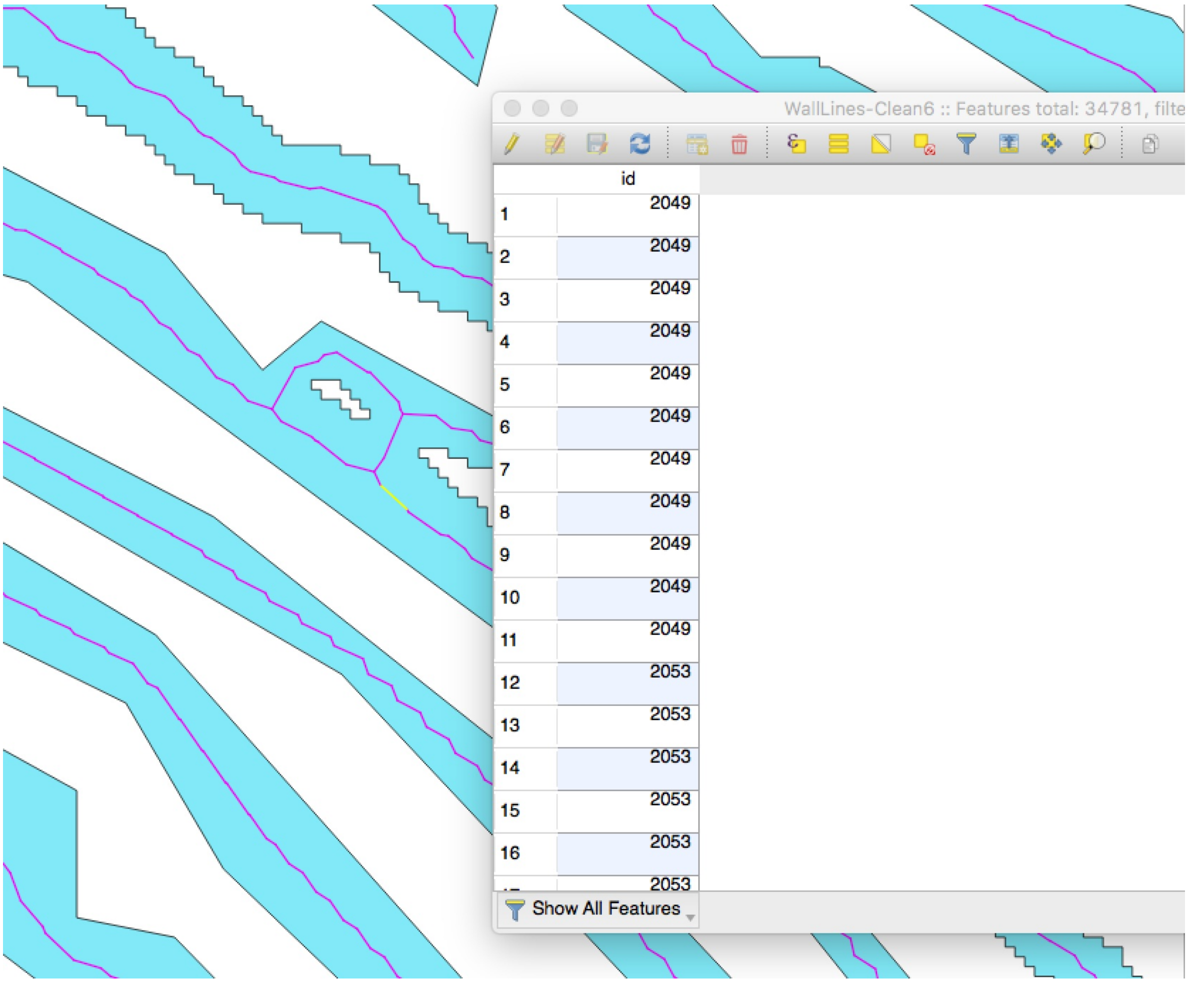
- Clean1: 0.1 threshold
- Clean2: 0.5 threshold
- Clean3: 1.0 threshold
- Clean4: 1.5 threshold
- Clean5: 2.0 threshold
- Clean6: 3.0 threshold

I stopped after the 3.0 threshold when most dangles were removed



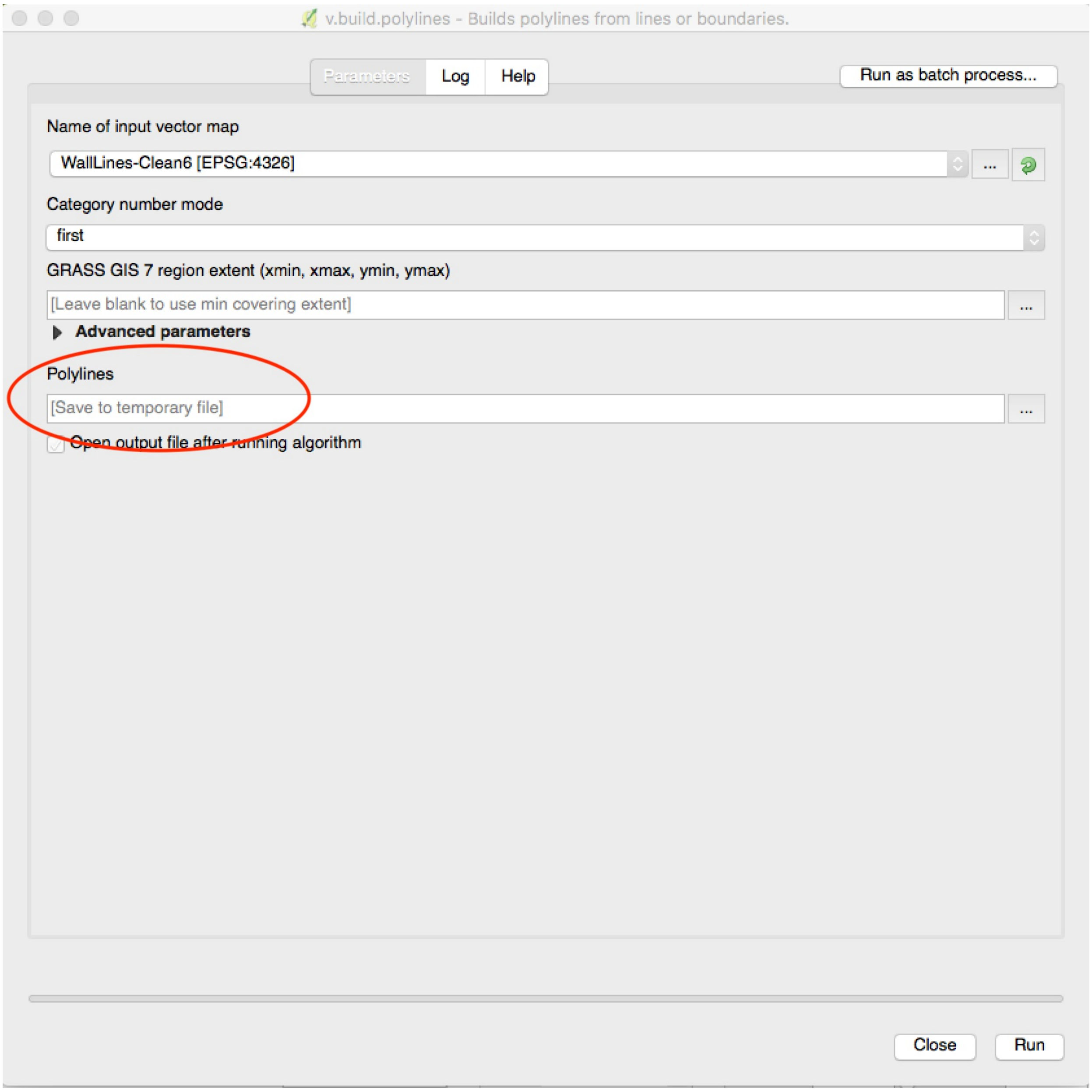
. Notice that each different color shows the iteration in which it was removed.

8. Merge individual segments into polylines. The cleaned layer contains thousands of individual segments that make up each wall. For example, after cleaning, my cleaned layer has 34,781 individual features (segments)



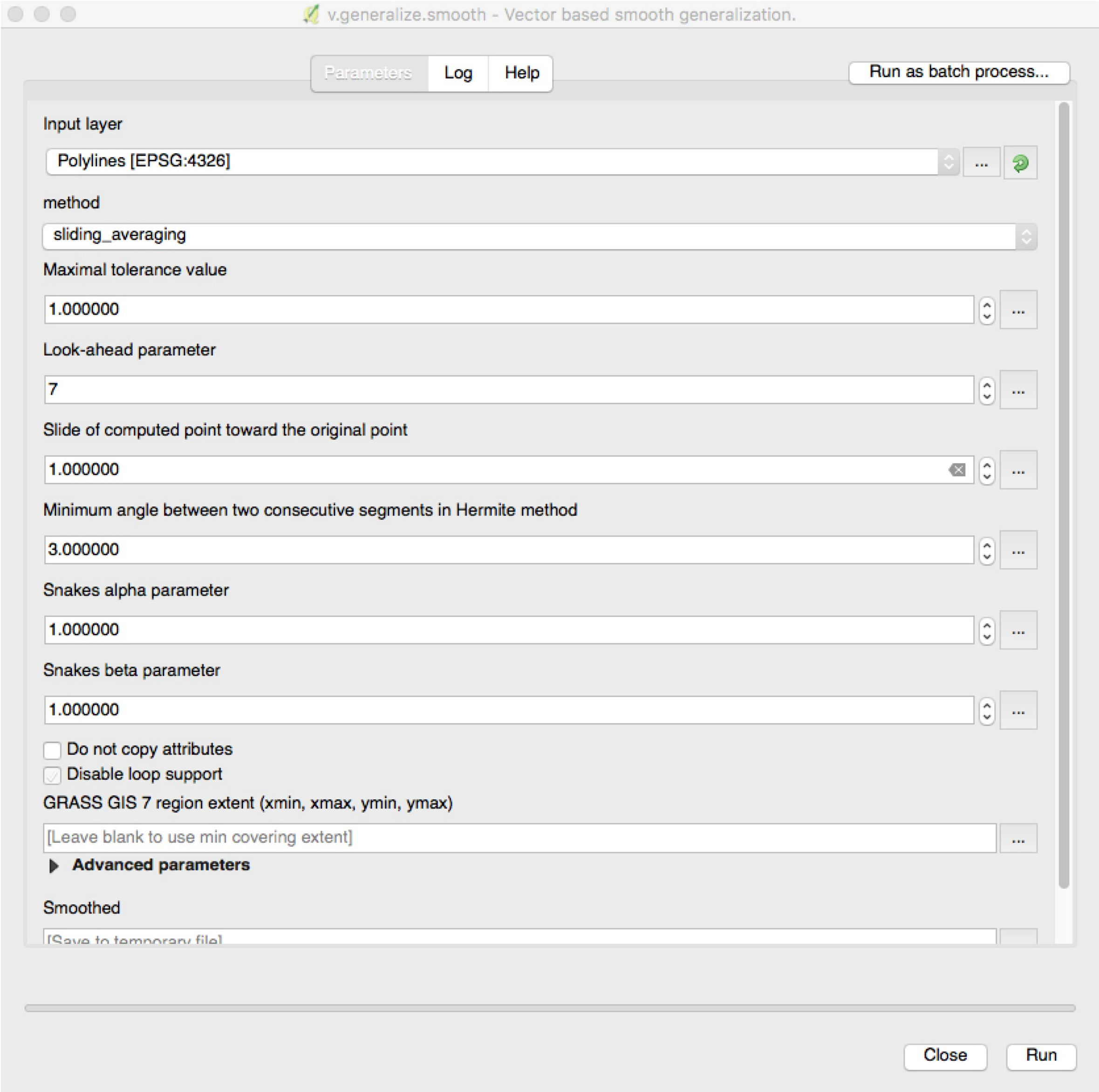
. Before we can use tools in QGIS to simplify the lines, the individual segments need to be merged into polylines.

GRASS -> Vector -> v.build.polylines: Input layer (cleaned layer); Category number mode = "first" (adds id number to attribute table for new polyline layer); DO NOT save to file. On my system, this caused the tool to fail

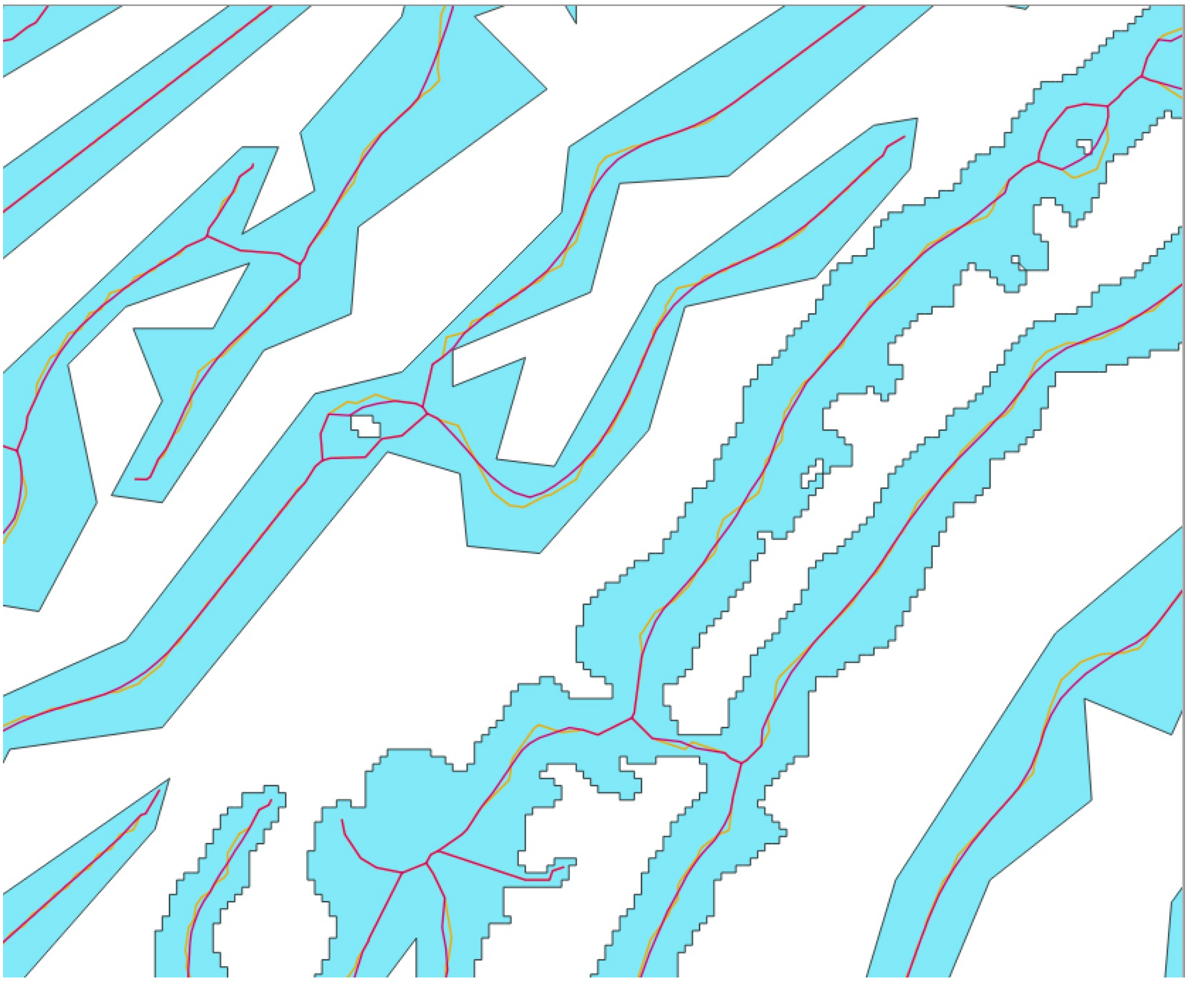


.

10. Straighten the new polylines using GRASS -> Vector -> v.generalize.smooth. Input Polylines layer, method = sliding_averaging to create smooth lines, I changed Slide of computed point toward the original point to 1.00, but this parameter can be modified

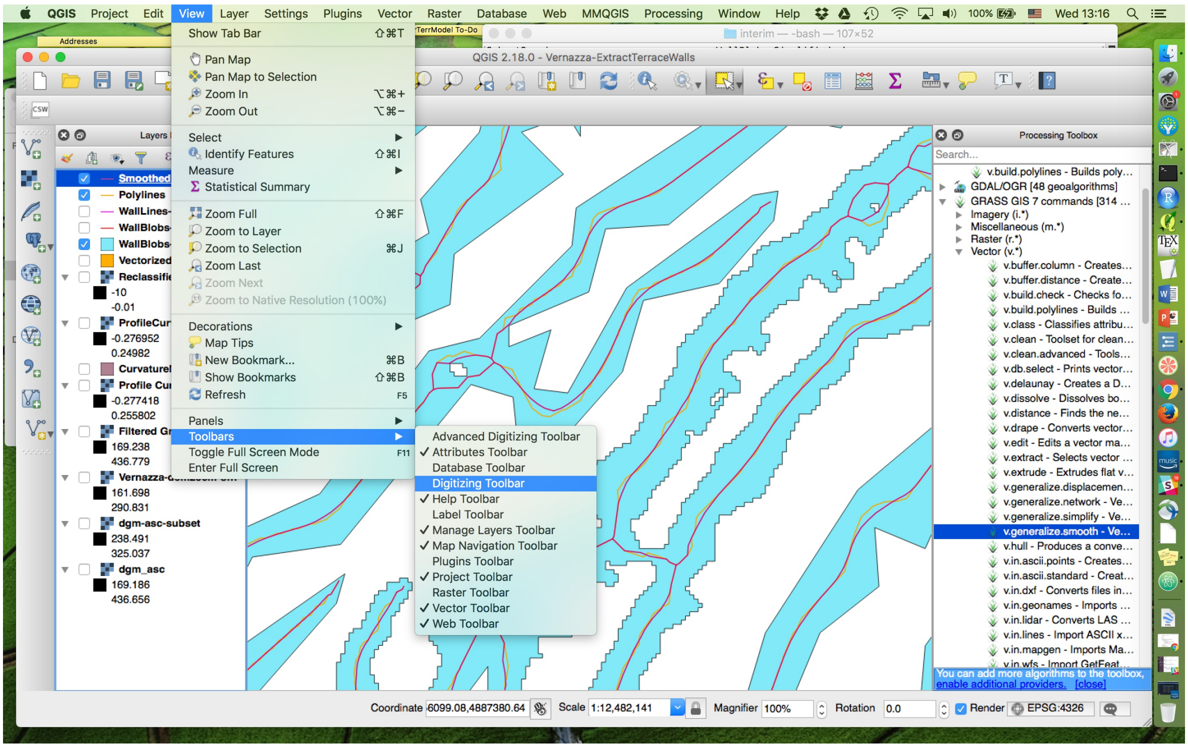


. The resulting polylines have fewer bends in them as you can see from

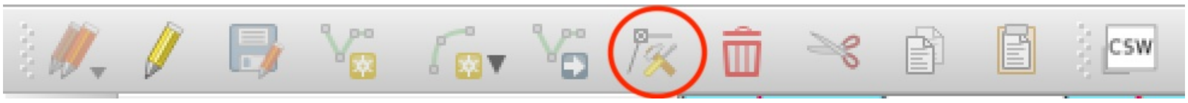


where the orange lines are the pre-smoothed polylines, while the red lines are the smoothed polylines.

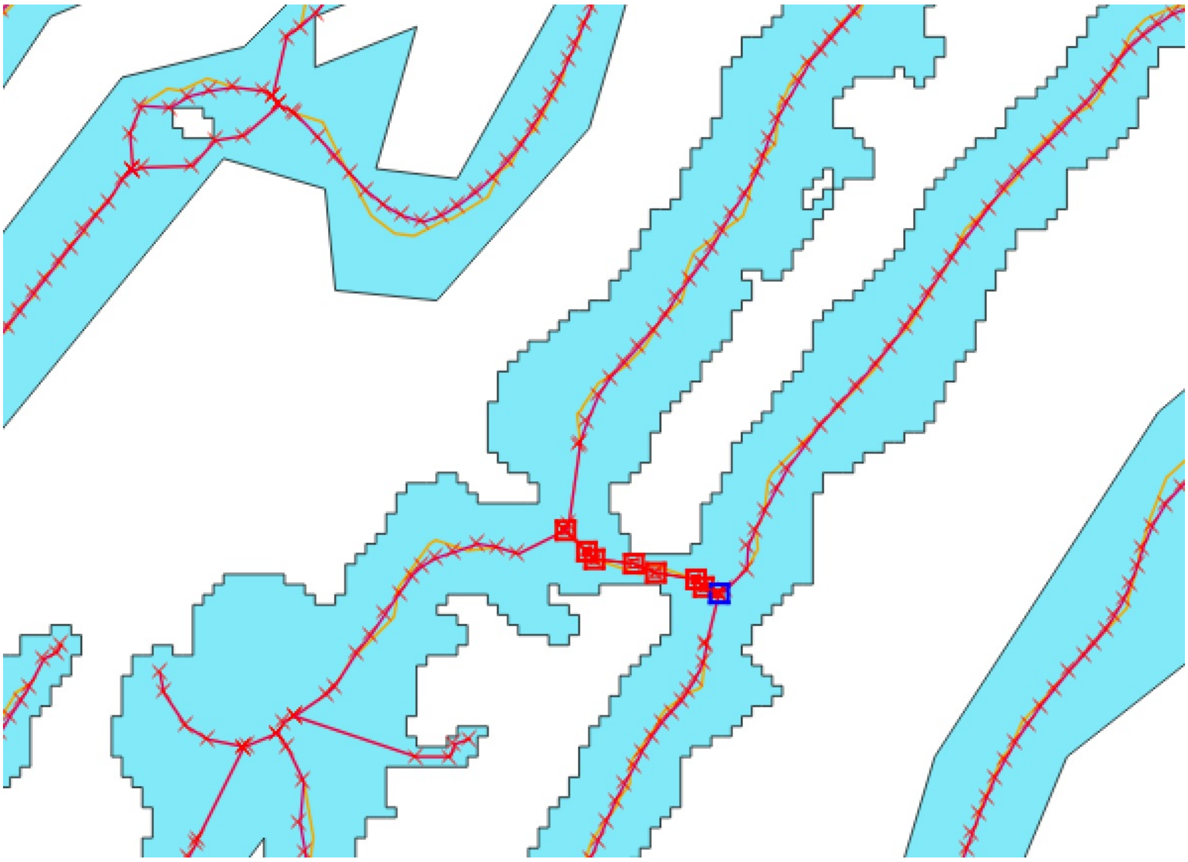
11. At this point, there may be some extraneous lines that have not been removed by the previous steps. Manual editing can fix those issues. Make sure that the Digitizing Toolbar is turned on



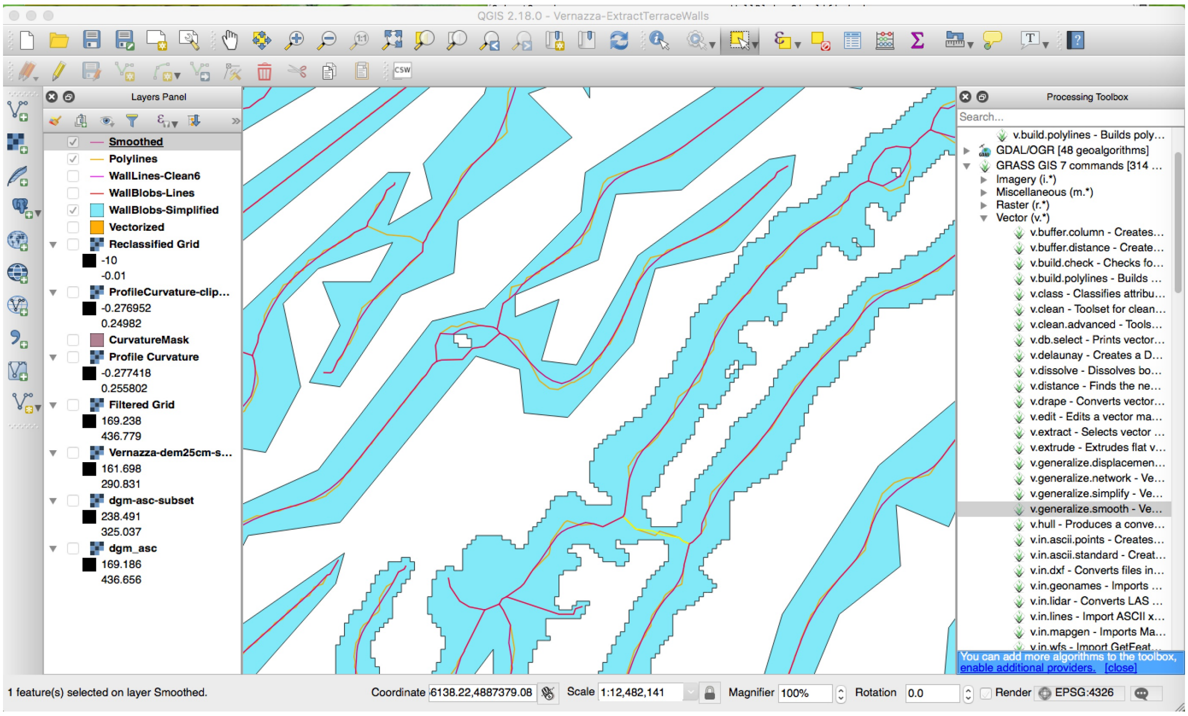
. Click the yellow pencil button to Toggle Editing on. Then click the Node Tool



to start pruning the polylines. Double check that the layer being edited is the Smoothed layer (should have pencil beside it in Layers Panel). Click on a segment you would like to delete or adjust, then click on one of the red boxes to adjust that node (note: when a node is selected it turns blue,

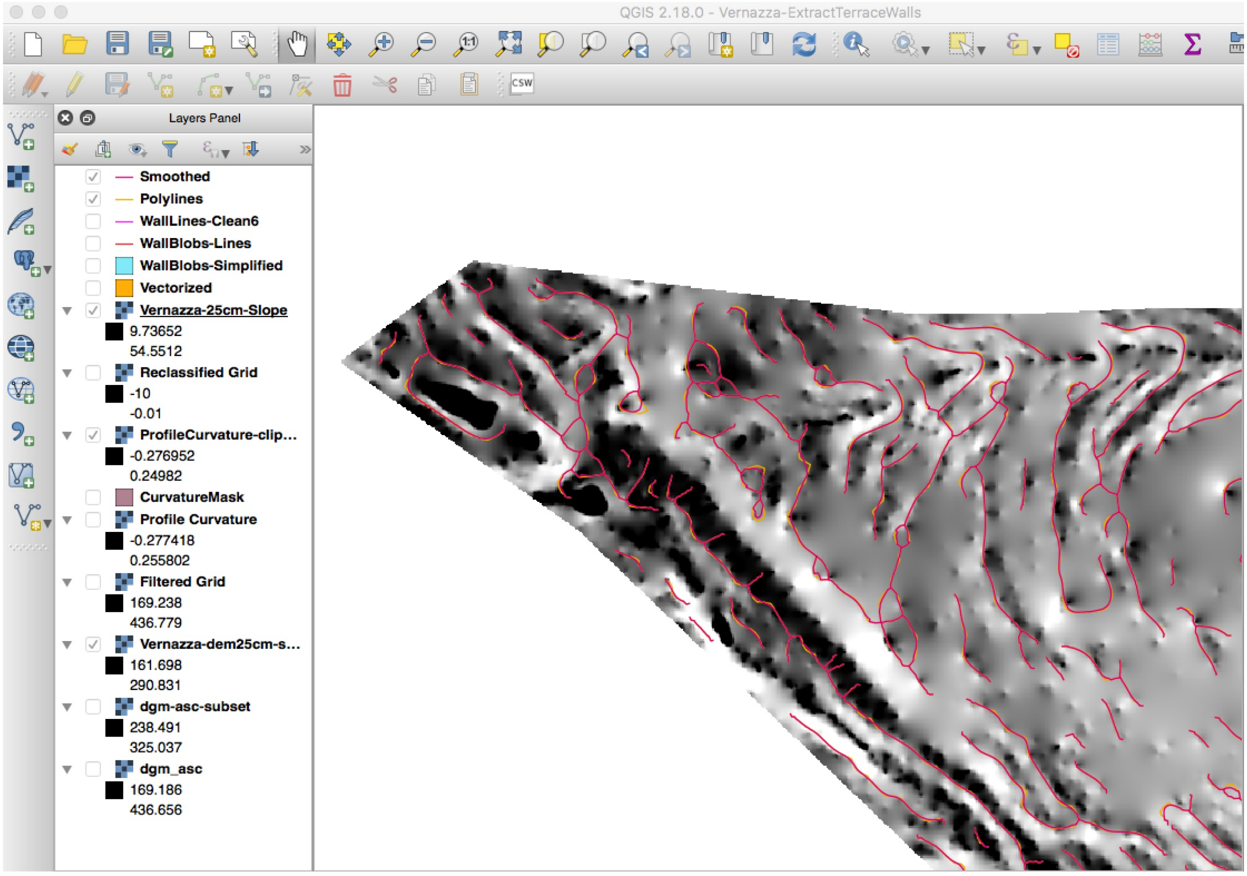


). Individual polylines or segments can also be deleted using the Select features by area or single click tool to select 1 or more segments (they turn yellow,



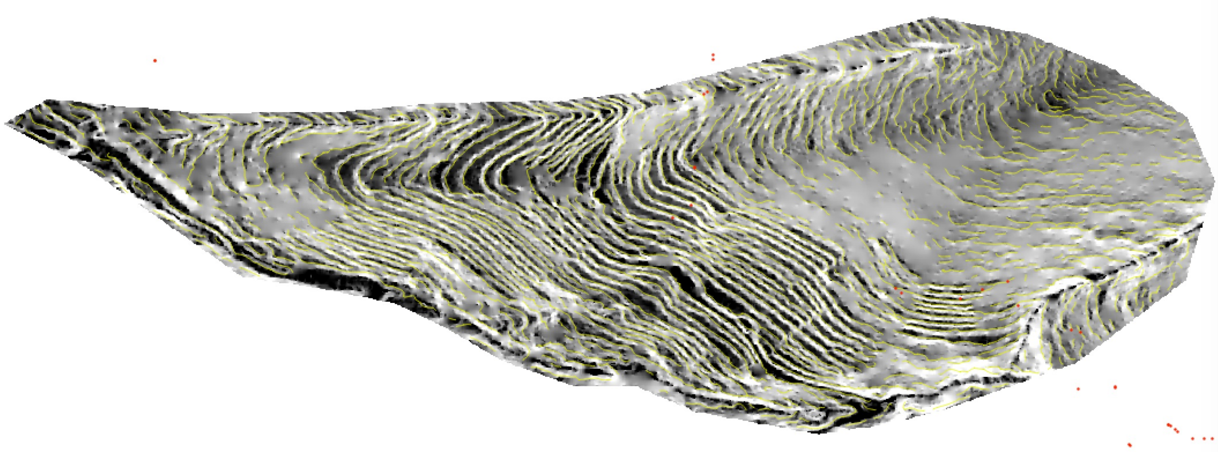
) and then using Delete button on keyboard.

I use a slope layer derived from the DEM as background to help with deciding which segments are extraneous or need to be separated



.

Here is an example of the final, simplified terrace features



.