Review of Clustering methods & results for the GEP

# Overview

This brief provides an overview of the findings regarding the methods used to generate population clusters (or settlements) for the GEP. The research activity focused on existing and/or newly developed (explorative) methods and their performance in terms of a) visual representation and b) quantitative accuracy.

The district of Kisoro (Uganda) was used as reference for this comparison. There were six methods that were studied; they are presented in brief below.

1. **Method 1 (GEP)**. This method has been developed by KTH as part of the GEP analysis in 2019. It is based on processing (resampling, polygonization, buffering, dissolving) of HRSL data. Detailed description of the method is available in [this publication](https://www.mdpi.com/1996-1073/12/7/1395).
2. **Method 2 (KTH MEN)**. This is an updated version of the above method. It is still based on processing of HRSL data but with minor differences in step order. In addition, this method is available in the form of a [ipynb pipeline](https://github.com/babakkhavari/Clustering), which makes it easier to replicate and/or customize. The output result also includes urbanization (urban or rural) and electrification status (electrified or not-electrified) characterization of settlements (clusters).
3. **Method 3 (KTH MEN Buff)**. An [updated version](https://github.com/akorkovelos/Clustering/tree/master/DBSCAN_Clustering) of the above ipynb that allows additional custom buffer assumptions when it comes to dissolving polygons (or clusters). Allowing higher buffer can lead to lower number of clusters to be generated.
4. **Method 4 (CA Qgis)**. This method is developed by Christopher Arderne in Qgis. It seems like a combination of the above. Similarly, it is based on HRSL. It is a Qgis command-based pipeline (similar to Method 1) that includes several steps (polygonize, fix, drop with area<12500, buffer 0.005deg, fix, dissolve, multipart to single) when processing (similar to methods 2 & 3). This was developed by Chris in response to the investigation over the issue of extremely high densities in GEP cluster results.
5. **Method 5 (CA cluster)**. This is a method developed by Christopher Arderne; it used clustering [algorithm(s) in python](https://github.com/carderne/clusterize) to classify HRSL population into groups based in distance; it then calculates a convex hull around them.
6. **Method 6 (DBSCAN)**. This method was developed as a [ipynb pipeline](https://github.com/akorkovelos/Clustering/tree/master/DBSCAN_Clustering) by Alexandros Korkovelos; it uses DBSCAN clustering algorithm to assign buildings (from building footprint dataset) into clusters based in two criteria a) minimum number of building per cluster and b) max distance between buildings. The code calculates then concave polygons surrounding buildings that fall into the same cluster. Building that are not fall into a cluster are extracted in a separate point layer.

Methods 1-4 are quite similar in principle. Differences between them occur due to minor modifications of processing order, cleaning and or different assumption of input parameters.

Methods 5-6 differ from the previous as they deploy un-supervised learning to deduct clusters. Nevertheless, they are quite similar between them in terms of functionality but employ different input data and/or polygon processing algorithms (convex/concave), which might affect total area of the clusters.

Evaluating how “visually representative” the resulted clusters are in comparison to real settlements is a rather challenging task. The most updated - and perhaps - accurate dataset in terms of capturing build-up are is the [building footprint dataset](https://github.com/microsoft/Uganda-Tanzania-Building-Footprints). For the case of Kisoro, this consists of 62483 building features as shown in Figure 1. Therefore, this was used as a reference for visual inspection of the generated clusters.

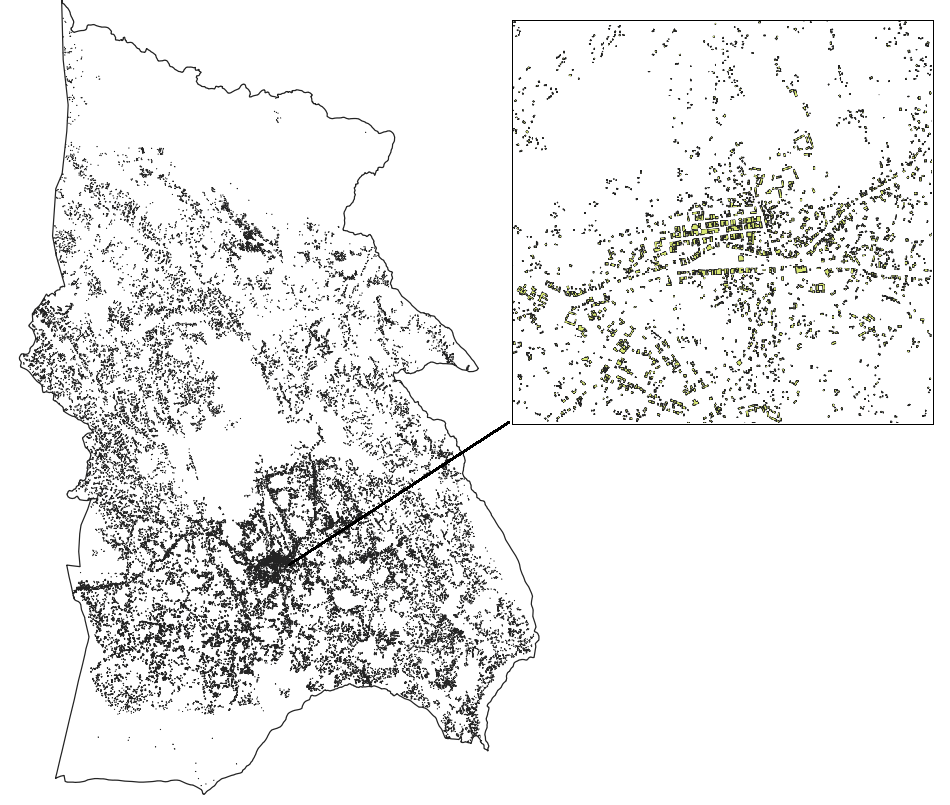


Figure . Building footprint in Kisoro district, Uganda. Data retrieved from Microsoft’s [AI for Humanitarian Action program](https://github.com/microsoft/Uganda-Tanzania-Building-Footprints).

# Findings

## Qualitative evaluation - Visual inspection

### Method 1

In terms of visual inspection, method 1 seems to be performing relatively well. As seen in Figure 2, the main built-up areas are captured, although there are some areas (especially in urban centres – see encircled area in Figure 2) that get enclosed in the cluster even if there are no buildings in that location. This might lead to increased area for the cluster, which in turn might have an effect on the results of the electrification analysis (area is used to calculate MV extension costs).

**Note!** It should be highlighted here that the resulted clusters in this method do indicate some discrepancy in terms of the administrative boundary around the district (no clusters in the mid-southwest part of the province). This might be explained from the source of administrative layer, which in this assignment was derived from GADM, while method 1 (GEP results) were based on World Bank administrative boundaries.

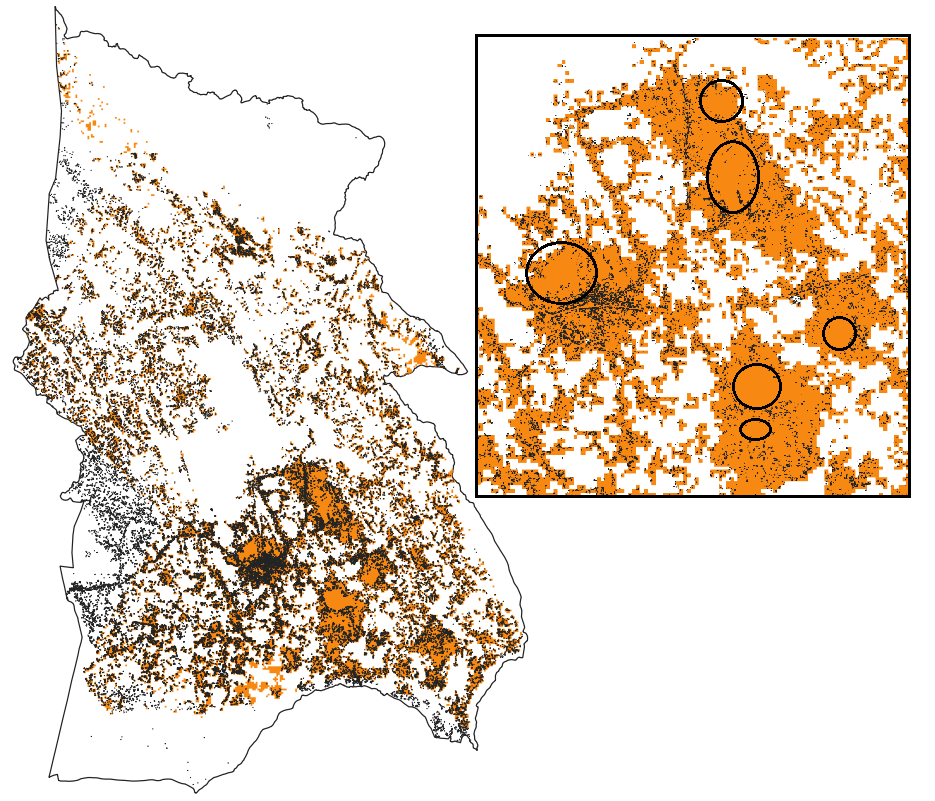


Figure . Generated cluster using method 1 (GEP). These clusters are currently used in (and thus retrieved from) the [GEP Explorer](https://electrifynow.energydata.info/).

### Method 2

In terms of visual representation, clusters deriving from method 2 seems to yield very similar results to method 1 (see Figure 3). That is, overall performance looks relatively good, however, it seems to overestimate clusters especially in dense, urban areas. This might lead to similar area/density issues described earlier.

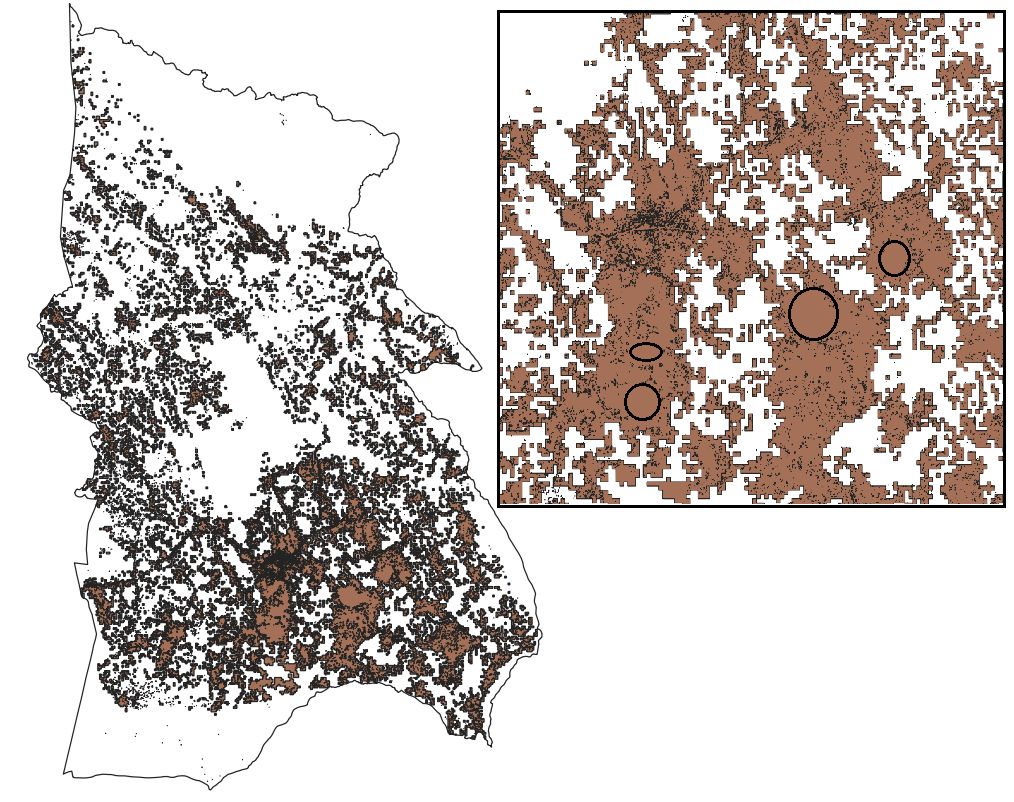


Figure . Generated cluster using the update “clustering” method developed by KTH (KTH MEN). The clusters were retrieved from the [Mendeley open database](https://data.mendeley.com/datasets/z9zfhzk8cr/4).

### Method 3

Method 3 seems to be outperforming the previous two methods in terms of “visual accuracy”. As seen in Figure 4, the resulted clusters seem to cover more accurately the built-up area in the province and do a better job in cutting out non-built area in urban centres (in contrast to the previous two methods).

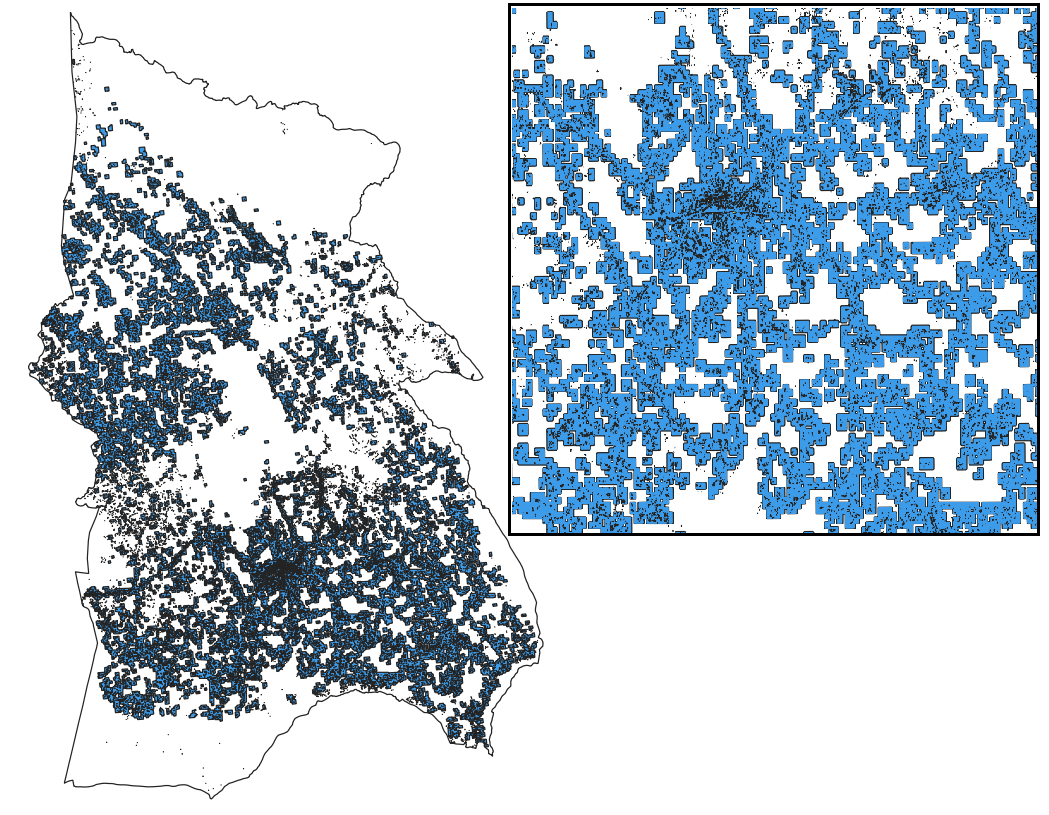


Figure . Generated cluster using the update “clustering” method developed by KTH, with an additional added buffer/dissolve step (KTH MEN Buff). The clusters were generated from scratch.

### Method 4

In terms of visual representation, this method indicates poor performance in comparison to the previous ones. Big portion of the built-up area (thus population) is not captured in the clusters. That is due to the cut-out limit in the HRSL set in the Qgis pipeline, which excludes many areas in the clustering process. In addition, this method leads to a few big clusters (see Figure 5), which will act as one entity in the following OnSSET analysis. Finally, the process of buffering in this method leads to enclosing non-built up area in the clusters.

All the above will have an impact on population density/area and effectively on the results of the electrification analysis. It should be highlighted however, that poor results shall be primarily be linked to rough assumptions (or non-proper calibration of those) in the method and not to the pipeline itself. That is, with proper calibration this method might lead to better results. Nevertheless, excluding pixels from the HRSL (which is the defining characteristic of this method) seems to be causing a lot of visual distortion, which is something that we need to consider carefully before deploying the method.

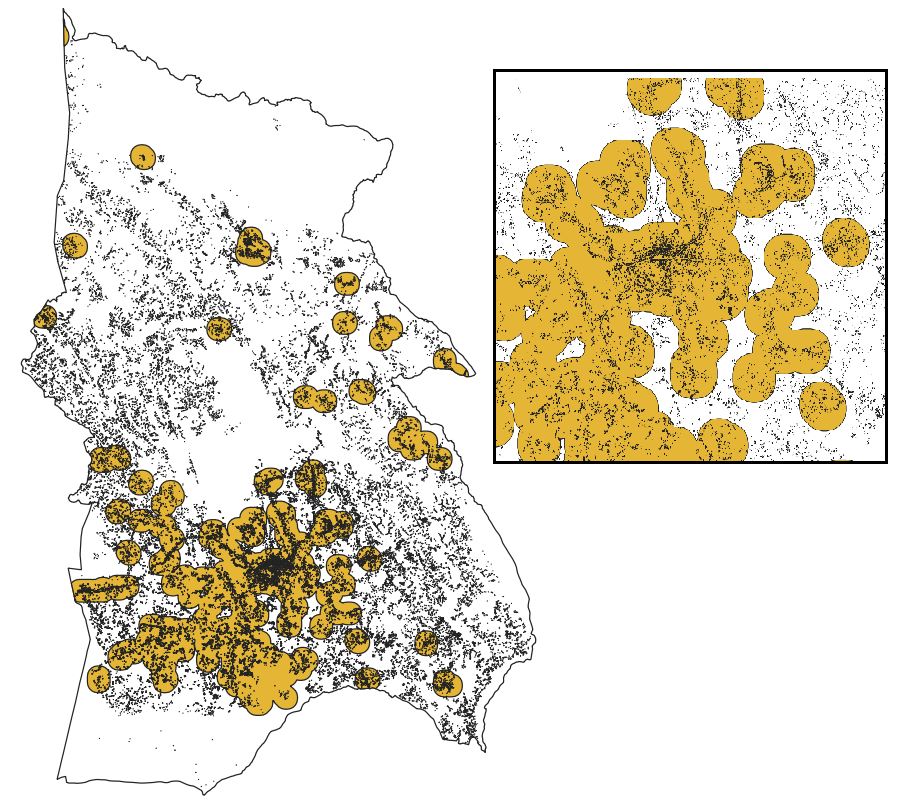


Figure . Population clusters in Kisoro as resulted from the Qgis pipeline proposed by Christopher Arderne (CA Qgis). Clusters shared by Chris.

### Method 5

Similar to the previous method, in this case the resulted clusters indicate very poor performance in terms of visually capturing the built-up area in Kisoro. First, the actual number of clusters is very low, which might be due to the assumptions set in the clustering algorithm. Second, the generated clusters seem to be arbitrarily generated. As explained before, this might not be a problem solely attributed to the method itself but to its calibration in regards to the input parameters and design criteria.

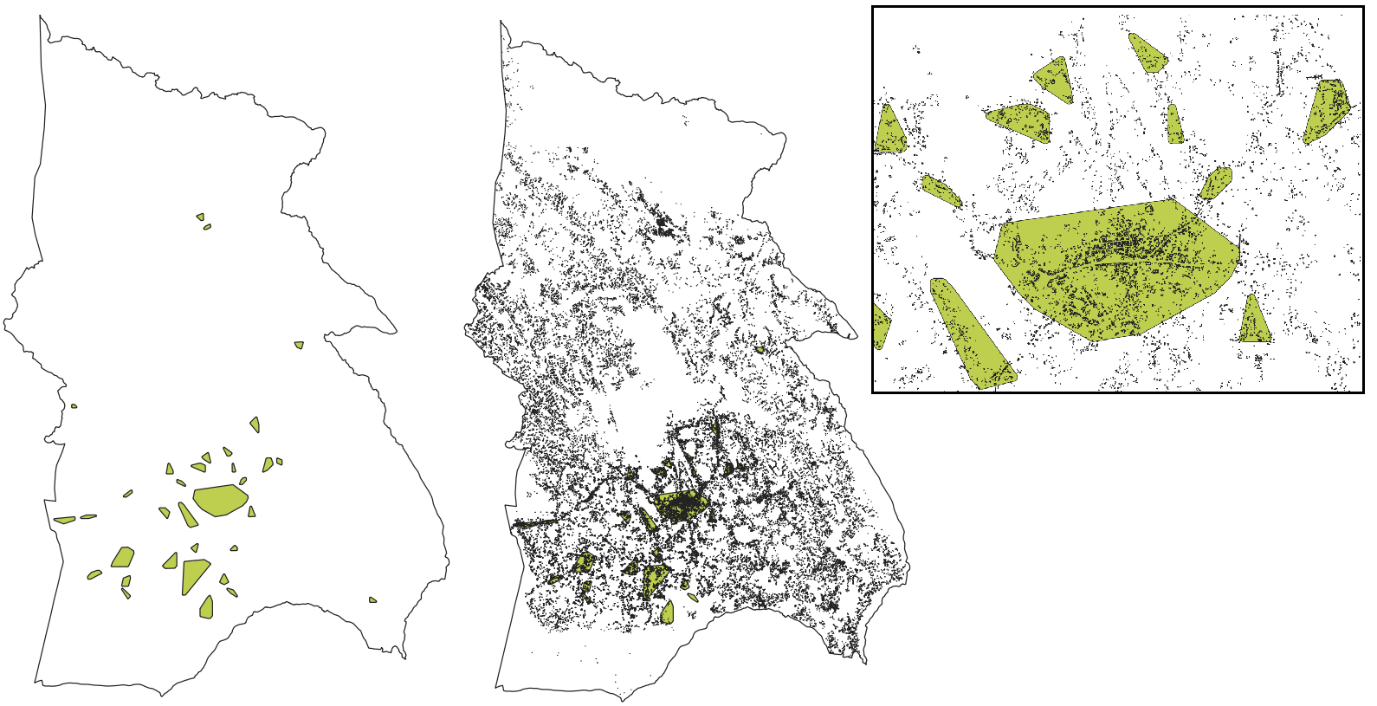


Figure . Population clusters developed using the un-supervised clustering algorithm (--method=radius --radius=3 --buffer=50) and convex polygonization (CA cluster). Clusters were shared by Christopher Arderne.

### Method 6

Finally, method 6 (DBSCAN) seem to be the most “visually representative”. This is expected as the method is based on the building footprint dataset itself. The method led to the creation of 1075 clusters that enclose 25429 buildings. The rest (37054) are represented as single features (points). Results illustrated in Figure 7.

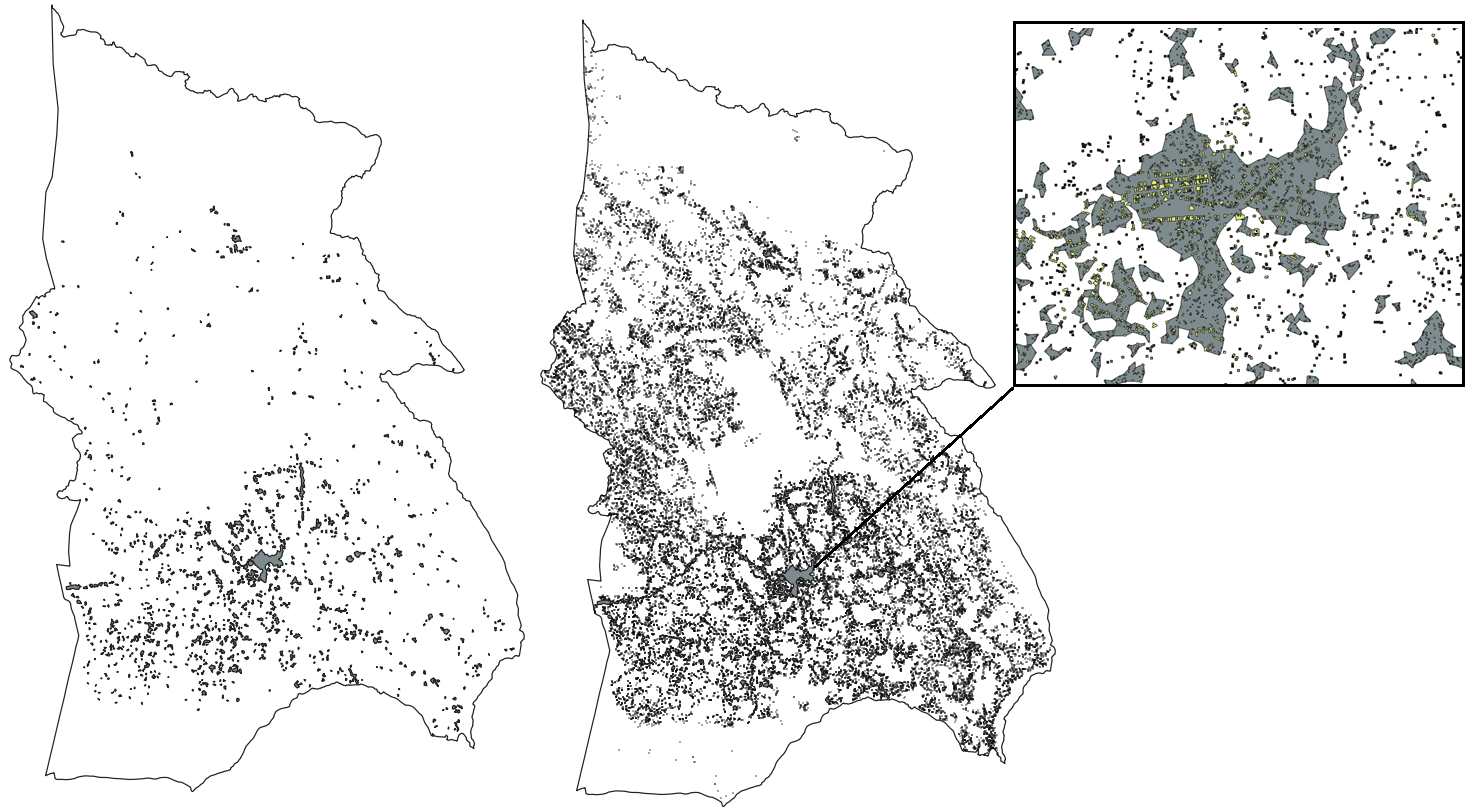


Figure . Population clusters developed using the un-supervised clustering algorithm (DBSCAN, max distance: 50m, min cluster size: 15) and concave polygonization (DBSCAN). Clusters were generated from scratch.

Note that the resulted clusters were based on a very specific selection of parameters for the DBSCAN algorithm. Although the results look representative, there are still questions around the clustering process. For example, why nearby clusters are not merged into one; what happens to the buildings that are not clustered? How we extract and calibrate population in those?

These lead to the second order of evaluation that follows.

## Quantitative evaluation

Visual representation is an important first level assessment and can – as seen above – provide a rough guide as to which method might performs better than another. Nevertheless, it is important that the generated clusters do convey proper quantitative information as well.

### Total population

In this regard, we may want to look at the total population and how well do the methods described earlier capture the number of people that are statistically living in the area of focus. Using population values from HRSL, the total population in Kisoro is estimated at ~271386 people.

This value seems to be captured relatively well by methods 1-3 although in all cases the total population is higher than what we get with HRSL. This is due to different sources and calibration methods, which were not consistent between methods (this is a rough exploratory assignment, in the future it is suggested that same sources and calibration references are used). Methods 4-6 capture only part of the population. That is due to the exclusion of built-up areas in the process. See example graph for the methods studied [here](Total%20population%20in%20clusters%20per%20method.html).

### Number of clusters

These results make sense when one considers the number of clusters generated in each method. Methods 4-5 only resulted in very few clusters, 26 and 33 respectively. On the contrary, method 6 resulted in 1075 clusters; more than method 3 (971) but less than methods 1-2 with 1354, 1568 respectively. Note however that non clustered buildings in method 6 were not included in the population summations. That is, assuming a number of people per building one could estimate the total population including non-clustered buildings. For example, if one assumes 5 people/building the total population in method 6 is estimated at 254390 people (>6.5% difference). See example graph for the methods studied [here](No%20of%20clusters%20per%20method.html).

### Population density

As indicated above, the method used can lead to different number of clusters, different area included in the clusters and different total population in those clusters. That is, population density might change accordingly. In general, the “tighter” a cluster is around the built-up are it represents the higher the population densities observed in the clusters. In contrast, “looser” (e.g. additional buffer) cluster area may lead to lower population densities. See example graph for the methods studied [here](Cluster%20pop%20density%20comparison.html). **Note!** It is important to differentiate the arithmetic population density with what we calculate here, which is more in line with residential/urban/built-up density.

# Conclusion – Verdict

Estimating population density from GEP results (area/population) is possible, however it is unclear to what extent this value can be used to deduct useful insight for policy. The retrieved density is more akin to the residential/urban population density rather than the commonly used arithmetic population density.

Nevertheless, even those values in some countries areas (based on GEP results) seem unreasonably high. The problem has two sources. First, there are very dense, urban areas for which residential population densities are very high. Second, there are very small areas (single pixels from HRSL) which do have considerable amount of people living in them. That is, when those areas are converted to polygons and become clusters, they continue to convey the high population value, which in turn leads to very high densities. Both issues are difficult to solve as this is perhaps related to erroneous data at the source. One might refer to [this publication](https://www.unsdsn.org/leaving-no-one-off-the-map-a-guide-for-gridded-population-data-for-sustainable-development) for more info on the types of gridded population and their uncertainty.

When it comes to the clusters results, current GEP clusters do seem to perform well for the scope they serve. The updated clustering algorithm – with some updated in terms of buffers (e.g. as in method 3) - seem to be a reasonable way forward for the next iteration of the GEP; thus, highly suggested.

On the other hand, applying un-supervised learning techniques to deduct a more scientifically-proof result is promising. However, is computationally expensive (input parameter calibration might require many iterations) and as such I am not confident to what degree it will offer a massive upgrade (and/or added value) to the GEP (considering its scope). Let alone, it is based on building footprint data that are not available globally. One might also consider how non-clustered buildings are assigned population and modelled in the electrification model. Finally (and most importantly) is it very difficult to cross-validate the output of the clustering algorithm; the results might be “statistically correct”, but they might fail to capture the realities on the ground.

Perhaps, if time and effort can be spent towards this direction a combination of method 3 & 6 might yield a very satisfying result.?