

PHIA Geospatial Data Use Manual

Reference Guide for Using Geospatial Data from the Population-based HIV Impact Assessments



The mark "CDC" is owned by the US Dept. of Health and Human Services and is used with permission. Use of this logo is not an endorsement by HHS or CDC of any particular product, service, or enterprise.

This project is supported by the U.S. President's Emergency Plan for AIDS Relief (PEPFAR) through CDC under the terms of cooperative agreement #U2GGH001226.
The contents of this document do not necessarily represent the official position of the funding agencies.

PHIA Collaborating Institutions

ICAP at Columbia University
Centers for Disease Control and Prevention (CDC)
Westat

Donor Support

This project has been supported by the President's Emergency Plan for AIDS Relief (PEPFAR) through the Centers for Disease Control and Prevention (CDC) under the terms of cooperative agreement #U2GGH001226. The findings and conclusions are those of the authors and do not necessarily represent the official position of the funding agencies.

Suggested Citation

Population-based HIV Impact Assessment (PHIA) Geospatial Data Use Manual. New York, NY. April 2021.

Access this Manual Online

The PHIA Project: <http://phia.icap.columbia.edu>

Contact Information

ICAP at Columbia University
722 West 168th Street
New York, NY 10032
Website: icap.columbia.edu
Email: icap-communications@columbia.edu

Table of Contents

List of Abbreviations.....	4
List of Tables	5
List of Figures	5
Executive Summary	6
1. Introduction	7
1.1 Background.....	7
1.2 Geoprivacy.....	7
1.3 Geomasking.....	8
2. Methods	9
2.1 Types of Geospatial Data Collected in PHIA Project Surveys.....	9
2.1.1 Cluster boundaries and centroids	9
2.1.2 Household GPS coordinates collected during household mapping and listing	9
2.1.3 Household GPS coordinates collected during data collection	9
2.2 Creation of Final Unmasked Cluster Centroid Datasets.....	11
2.2.1 Assessing data availability, data quality, and spatial coincidence of cluster datasets.....	11
2.2.2 Creating final cluster boundary and unmasked cluster centroid datasets.....	14
2.3 Creation of Final Geomasked Cluster Centroid Datasets.....	14
2.3.1 Determining the minimum and maximum displacement distance for each cluster.....	14
2.3.2 Defining additional displacement constraints	15
2.3.3 Displacing cluster centroids	16
2.3.4 Implementing final quality control.....	16
2.3.5 Producing final geomasked cluster centroid datasets	16
3. PHIA Project Public-Release Geospatial Dataset Policy and Procedures	17
3.1 Policy for Public Access to and Use of Geospatial Datasets.....	17
3.2 Procedures for Requesting, Accessing, and Utilizing Geospatial Datasets.....	17
3.2.1 Requesting Geospatial Datasets	17
3.2.2 Accessing and Using Geospatial Datasets	17
References	19
Appendix A Sample SAS code for joining geospatial datasets with household and individual datasets	20
Appendix B Cluster centroid geomasking Python script	21

List of Abbreviations

General	
CDC	Centers for Disease Control and Prevention
CSV	Comma Separated Values
DHS	Demographic and Health Surveys
FTP	File Transfer Protocol
HBTC	Household-Based Testing and Counseling
ODK	Open Data Kit
PEPFAR	U.S. President's Emergency Plan for AIDS Relief
PHIA	Population-based HIV Impact Assessment
PII	Personally Identifying Information
Geographical and Geospatial	
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
CRS	Coordinate Reference System
DEM	Digital Elevation Model
EA	Enumeration Area
EPSG	European Petroleum Survey Group
GADM	Global Administrative Areas
GCS	Geographic Coordinate System
GIS	Geographic Information System
GPS	Global Positioning System
KML	Keyhole Markup Language
GPS	Global Positioning System
OSM	OpenStreetMap
PCS	Projected Coordinate System
UTM	Universal Transverse Mercator
SNU	Subnational Unit
SRTM	Space Shuttle Radar Topography Mission
USGS	United States Geological Service
WGS	World Geodetic System

List of Tables

Table	Title	Page
1	Types of geospatial data collected in PHIA Project surveys	11
2	Cluster boundary and household GPS data availability and spatial coincidence analysis: Approach to creating final cluster boundaries	14

List of Figures

Figure	Title	Page
1	Illustration of possible outcomes for geospatial data availability and spatial coincidence analysis	13
2	Illustration of the donut masking method	16

Executive Summary

Geospatial data play an important role in the efficient planning, preparation, and implementation of PHIA Project surveys. These data include geolocation information about survey clusters and households, as well as laboratory facilities utilized for processing and testing blood samples collected in survey participants' households.

While used primarily for operational purposes, survey geospatial data also have critical analytical value. However, the locations of survey clusters and household are confidential information, which must be managed with similar protections as any other personally identifying information (PII). The PHIA Project has carefully balanced the analytic potential of geolocation data with the imperative to protect the privacy of survey participants by creating a geographic dataset in which the locations of survey households and of clusters are “geomasked.” This involves two important elements. First, data pertaining to each individual and household in each survey cluster are assigned the location of the centroid of that cluster, which is a form of spatial aggregation. Second, the location of each survey cluster centroid is displaced by a random direction and a random distance, which is a form of random perturbation. This method preserves spatial relationships well enough for most spatial analyses while protecting the anonymity of participants.

The *PHIA Project Geospatial Data Use Manual* consists of three chapters and several appendices. Chapter 1 offers a brief introduction to geoprivacy and geomasking. Chapter 2 provides an in-depth description the types of geospatial data collected during PHIA Project surveys, as well as methods for creating both unmasked and geomasked cluster centroid datasets. Chapter 3 highlights policies and procedures associated with requesting, obtaining, and utilizing geomasked cluster centroid datasets. Finally, appendices provide dataset-specific metadata; SAS code for joining a geomasked cluster centroid dataset with its corresponding household questionnaire, individual questionnaire, and biomarker datasets; and Python code for conducting the “donut masking” method utilized to displace the locations of cluster centroids

1. Introduction

1.1 Background

The purpose of the PHIA Project is to collect nationally representative data regarding the status of the HIV epidemic in PEPFAR priority countries, including HIV prevalence, HIV incidence, and viral load suppression among persons living with HIV, alongside social, geographic, and economic data. Used together, these data enable analyses of the correlates of HIV risk, risk behaviors, and access to care and treatment. The inclusion of geospatial data alongside PHIA biomarker and interview data is a powerful tool for researchers, policy makers, and program planners and managers to identify spatial correlates of HIV risk. For example, it can help program planners to evaluate geographic hotspots for new infections, or to evaluate the impact of the location of health centers, roads, and other services on the relative uptake of care and treatment in a region. It also allows researchers to combine PHIA data with other sources of geographic data such as elevation or rainfall data to investigate the dynamics of climate variation and population health. But the analytical potential of geographic data must also be balanced against the imperative to protect the confidentiality of those who participated in PHIA surveys. “Geolocation” data identify locations across a broad range of geographic scales: nations, sub-national areas, census enumeration areas (EAs), and households. For most household-based surveys (e.g., PHIA, DHS), the locations of selected EAs and households are strictly confidential information, as these location data can be used to identify survey participants and their characteristics (e.g., HIV status, sexual behavior). That is, fine-scale location data (e.g., EAs, households) linked to survey data (e.g., questionnaire responses, biomarker results) can be or are personally identifying information (PII). Geo-location data therefore should be provided the protections required of any PII.

To that end, critical privacy protection measures have been implemented for PHIA Project data in order to allow meaningful geospatial analyses while protecting the privacy of survey participants. All individual and household identifiers have been stripped from publicly available datasets, including any household-level geographic information. The publicly available geographic datasets are comprised of cluster-level geographic coordinates, where data pertaining to each individual and household in each survey cluster are assigned the location of the centroid of that cluster, which is a form of spatial aggregation. Second, the location of each survey cluster centroid is displaced by a random direction and a random distance, which is a form of random perturbation. The remainder of this document describes, in detail, the geographic data collected, the procedures employed to produce the geomasked centroid cluster datasets (Chapter 2); the policies and procedures for use of the geographic data (Chapter 3); and finally, how to access and use the geomasked data (Chapter 3).

1.2 Geoprivacy

The increasing sophistication of GIS software and methods present an opportunity to analyze geospatial data at ever-refined levels, but also potentially permit researchers to link personal and health information to individuals via their geographic data. Several researchers have shown that geographic data can be used in conjunction with publicly available data to re-identify individual addresses using reverse geocoding (Brownstein et al 2006; Curtis, Mills, and Leitner 2006; Zandbergen 2014). Both Microsoft Bing Maps and Google Maps provide structure level

geocoding as part of their online mapping services, making it relatively easy to combine GIS data and commonly available address and telephone directories to identify individuals in a dataset. (Zandbergen 2014). For this reason, it is imperative that the geographic data released with health data be manipulated or “masked” before being released to users in order to protect the privacy of the respondents. However, masking must occur in such a way that the spatial relationships are preserved well enough to be analytically informative. As there is no universally accepted standard for how to balance the competing needs, researchers are left to determine the correct balance based on a combination of technical parameters, such as population density and spatial resolution, and the unique characteristics of the health information they are evaluating. For example, HIV or drug overdose data likely requires a higher emphasis on individual privacy protection than influenza data.

Given the relative sensitivity of PHIA data, the project has attempted to give maximum priority to the privacy of the individuals by stripping interview and biomarker datasets of any information more geographically specific than an anonymous identifier for each survey cluster, and then, within separate geographic datasets, employing a “donut masking” technique to randomly displace the geographic coordinates of each survey cluster within a prescribed area around the genuine coordinates. The PHIA Project also restricts access to unmasked cluster centroid and household-level location data and constrains use of data in terms of allowable uses, analyses, and acceptable presentation of results. In order to protect privacy, only geomasked cluster centroid datasets will be made publicly available. Unmasked cluster centroid and household-level location data will not be provided to individuals outside of investigating institutions.

1.3 Geomasking

Geomasking is a class of techniques for changing a geographic location in a dataset in an unpredictable way to protect confidentiality, while still trying to retain the relationship between location and disease occurrence (Sherman and Fetters 2007). The predominant techniques in geomasking are aggregation (i.e., assigning everyone in an area to a single coordinate, normally a central point in the area) and displacement (i.e., altering the coordinates randomly within set parameters). The PHIA Project employs both aggregation as well as displacement using a technique known as “donut masking.” All households and individuals within a dataset are assigned the location of the centroid of their respective survey cluster, and then the centroid is randomly displaced within an area around the actual center of the cluster. The centroid is placed in a random direction and a random distance within a defined torus (i.e., “donut”). This ensures the centroid is moved at least a minimum distance from its original location, and preserves the spatial pattern of the original data well enough to allow for spatial analysis (Hampton et al 2010; Allshouse et al 2010; Zandbergen 2014). The inner and outer radii of the torus are spatially adapted for each cluster, based on the area of the cluster and the population density in and around the cluster. Details regarding method and its application to PHIA data are described in Chapter 2.

2. Methods

2.1 Types of Geospatial Data Collected in PHIA Project Surveys

Several types of geospatial data are collected as part of each PHIA Project survey (Table 1). While these data are used primarily for operational purposes, they also form the foundation of the public-access cluster-level geospatial dataset that can be linked to household- and individual-level datasets for analytical purposes.

Production of PHIA Project public-access geospatial datasets involves various geoprocessing routines and spatial analyses, which are conducted utilizing geographic information systems (GIS) software, including ArcGIS Desktop (<https://desktop.arcgis.com/en/>) and QGIS (<https://www.qgis.org/en/site/>). An appropriate coordinate reference system (CRS) is utilized for evaluation and geoprocessing of each survey's geospatial datasets. Datasets for countries located entirely within a single Universal Transverse Mercator (UTM) zone are projected into the appropriate World Geodetic System (WGS) 84 UTM projected coordinate system (PCS), while datasets for countries situated in multiple UTM zones are projected into the WGS 84 Africa Albers Equal Area Conic PCS. Final geomasked cluster centroid datasets are re-projected into the WGS 84 geographic coordinate system (GCS) prior to public release.

2.1.1 Cluster boundaries and centroids

Enumeration area (EA) boundary data for all survey clusters are typically provided by each country's national statistical office. These data are usually provided in shapefile format. In some cases, data for cluster centroids may be provided instead of, or in addition to, data for cluster boundaries. In other cases, no cluster geolocation data may be available. When available, cluster geolocation data are loaded onto encrypted and passcode-protected tablet computers in keyhole markup language (KML) format using the MAPS.ME mobile app (<https://maps.me>), which includes OpenStreetMap (OSM, <https://www.openstreetmap.org>) data. Together, the cluster geolocation data and OSM data assist household listing and mapping personnel in navigating to and within survey clusters.

2.1.2 Household GPS coordinates collected during household mapping and listing

During the household listing and mapping phase of each survey, Global Positioning System (GPS) coordinates are collected for all structures, thereby identifying all potential dwelling units within each survey cluster. Geolocation data are collected using GPS-enabled tablet computers and Open Data Kit (ODK, <https://opendatakit.org>) software. After selection of survey households, GPS coordinates for selected households in each cluster are loaded onto tablet computers together with cluster location and OSM data as described in the previous paragraph. These data assist data collection personnel in navigating to and within survey clusters, and in locating survey households.

2.1.3 Household GPS coordinates collected during data collection

During the data collection phase of each survey, GPS coordinates are collected for each household in which one or more individual participates in HIV household-based testing and counseling (HBTC). No GPS coordinates are collected for households that refuse to participate in the survey, or for households with no individuals who participate in HIV HBTC. In contrast, households with more than one individual who participates in HIV HBTC may have multiple GPS coordinates associated with that household. These data assist survey personnel in relocating individuals who may require HIV HBTC-related follow-up.

Table 1 Types of geospatial data collected in PHIA Project surveys

DATA TYPE	FORMAT	GEOMETRY	SOURCE	NOTES
cluster boundaries	shapefile, kml	line and/or polygon	national statistical office	For some surveys, cluster centroids rather than cluster boundaries were provided. For these surveys, cluster boundaries were created as circular polygons centered on the cluster centroid, with a radius of 200 m for urban clusters and 1,000 m for rural clusters. In some cases, neither cluster boundaries nor cluster centroids were provided.
cluster centroids	shapefile, kml	point	national statistical office	For surveys with cluster boundaries, cluster centroids were also created.
household GPS coordinates	csv, kml	point	household listing	Includes all listed households with valid GPS data.
household GPS coordinates	csv, kml	point	data collection	Includes only households with at least one HIV testing participant. Households with more than one HIV testing participant may have multiple GPS coordinates. GPS data collection is not mandatory, such that households with HIV testing participants may have missing GPS data.

2.2 Creation of Final Unmasked Cluster Centroid Datasets

Creation of unmasked cluster centroids is a multi-step process, which involves: (1) assessing data availability, data quality, and spatial coincidence of data sources for each cluster; (2) creating a final cluster boundary dataset; and (3) creating a final unmasked cluster centroid dataset.

2.2.1 Assessing data availability, data quality, and spatial coincidence of cluster datasets

As a first step in the process of creating unmasked cluster centroids for subsequent geomasking, an evaluation is done to determine (1) the types of geospatial data available for each cluster; (2) the quality of each available dataset; and (3) the spatial coincidence of available datasets for each cluster.

First, for each survey cluster, a simple evaluation is done to determine the availability of (1) cluster boundary and/or centroid data, (2) household GPS coordinate data collected during household listing and mapping, and (3) household GPS coordinate data collected during data collection.

Second, several preliminary quality control measures are implemented. Geospatial datasets are evaluated for consistency in the unique identifier variable for each cluster. Household GPS coordinate datasets are evaluated for spatial outliers, which are removed. For data collection household GPS coordinate datasets, households with multiple GPS coordinates are identified, and a single geolocation for such households is generated by calculating the spatial average of the multiple GPS coordinates associated with that household.

Third, an analysis of the spatial coincidence (i.e., intersection) of each cluster polygon associated with each of the three spatial datasets is conducted: (1) cluster boundary polygons; (2) polygons defined by the minimum bounding circle encompassing all household GPS coordinates from household listing and mapping; and (3) polygons defined by the minimum bounding circle encompassing all household GPS coordinates from data collection. For any cluster with only a centroid, a circular boundary polygon is created, with a radius of 200 meters for urban clusters and 1,000 meters for rural clusters. Minimum bounding circle polygons for household GPS coordinates are constrained to have a minimum radius of 200 meters for urban clusters and 1,000 meters for rural clusters. There are 18 distinct possible outcomes for this spatial coincidence analysis (Figure 1, Table 2).

Figure 1 Illustration of possible outcomes for geospatial data availability and spatial coincidence analysis, with spatial coincidence analysis result codes 1 through 18

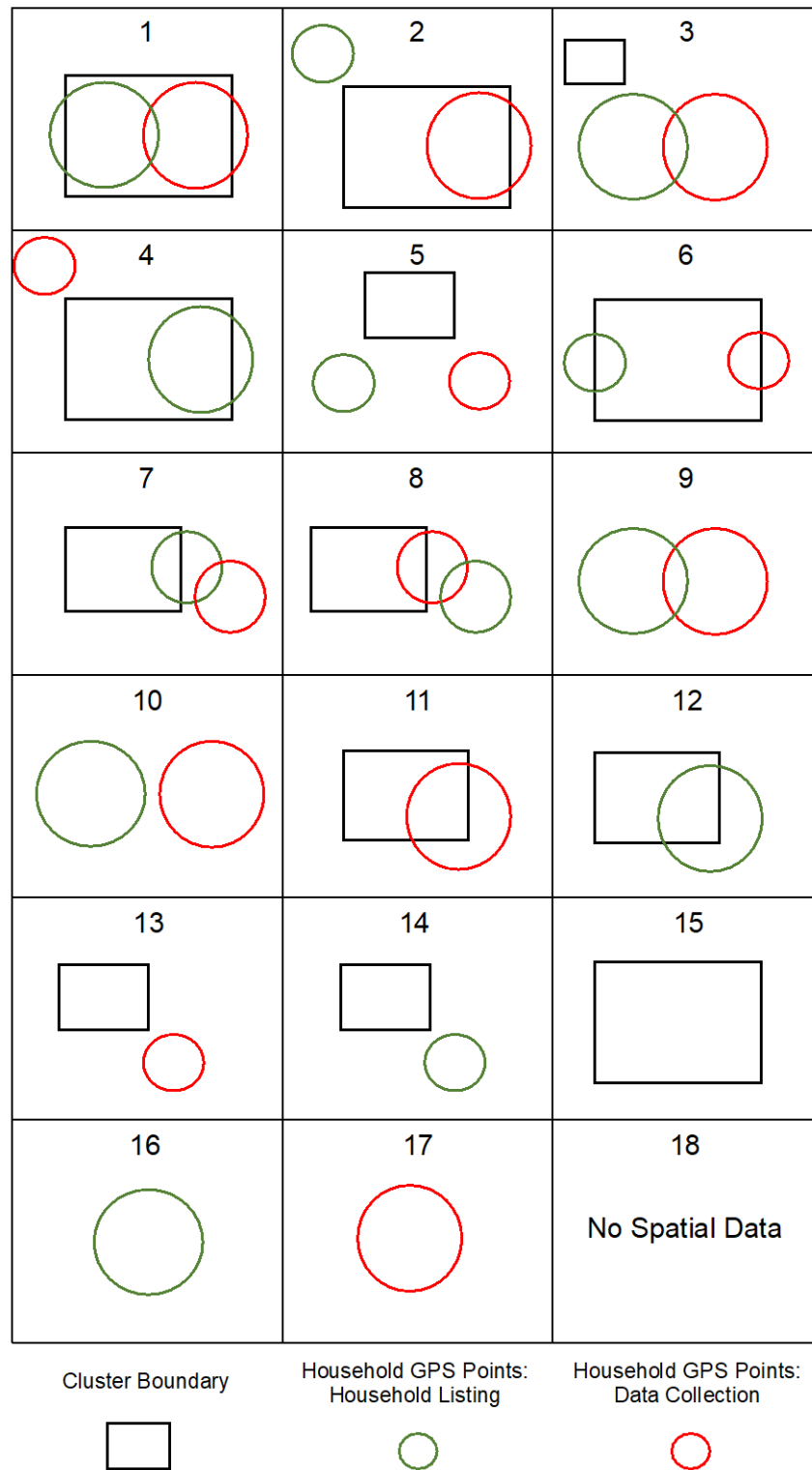


Table 2 Cluster boundary and household GPS data availability and spatial coincidence analysis: Approach to creating final cluster boundaries

SPATIAL COINCIDENCE ANALYSIS RESULT CODE	DATA SOURCE AVAILABILITY			SPATIAL INTERSECTION			APPROACH TO CREATING FINAL CLUSTER BOUNDARIES
	CLUSTER BOUNDARIES (CB)	HOUSEHOLD GPS POINTS: HOUSEHOLD LISTING (GPS-HL)	HOUSEHOLD GPS POINTS: DATA COLLECTION (GPS-DC)	CB / GPS-DC ^{1,3}	GPS-DC ^{1,3} / GPS-HL ^{2,3}	CB / GPS-HL ^{2,3}	
1	YES	YES	YES	YES	YES	YES	Maintain original CB.
2	YES	YES	YES	YES	NO	NO	Maintain original CB.
3	YES	YES	YES	NO	YES	NO	Create an artificial CB defined by the minimum bounding circle encompassing all GPS-DC and GPS-HL points. ³
4	YES	YES	YES	NO	NO	YES	Flag for evaluation.
5	YES	YES	YES	NO	NO	NO	Flag for evaluation.
6	YES	YES	YES	YES	NO	YES	Maintain original CB.
7	YES	YES	YES	NO	YES	YES	Maintain original CB.
8	YES	YES	YES	YES	YES	NO	Maintain original CB.
9	NO	YES	YES	N/A	YES	N/A	Create an artificial CB defined by the minimum bounding circle encompassing all GPS-DC and GPS-HL points. ³
10	NO	YES	YES	N/A	NO	N/A	Flag for evaluation.
11	YES	NO	YES	YES	N/A	N/A	Maintain original CB.
12	YES	YES	NO	N/A	N/A	YES	Maintain original CB.
13	YES	NO	YES	NO	N/A	N/A	Flag for evaluation.
14	YES	YES	NO	N/A	N/A	NO	Flag for evaluation.
15	YES	NO	NO	N/A	N/A	N/A	Maintain original CB.
16	NO	YES	NO	N/A	N/A	N/A	Create an artificial CB defined by the minimum bounding circle encompassing all GPS-HL points. ³
17	NO	NO	YES	N/A	N/A	N/A	Create an artificial CB defined by the minimum bounding circle encompassing all GPS-DC points. ³
18	NO	NO	NO	N/A	N/A	N/A	Flag for evaluation.

¹The polygon defined by the minimum bounding circle encompassing all GPS-DC points.

²The polygon defined by the minimum bounding circle encompassing all GPS-HL points.

³Minimum bounding circles encompassing GPS-DC and/or GPS-HL points are constrained to have a minimum radius of 200 meters for urban clusters and 1,000 meters for rural clusters.

2.2.2 Creating final cluster boundary and unmasked cluster centroid datasets

Based on the outcome of the spatial coincidence analysis, original cluster boundaries are either maintained, or artificial cluster boundaries are created (Table 2). For most clusters, all three geospatial data sources are available and are spatially coincident (i.e., spatial coincidence analysis result code = 1), such that the original cluster boundary is typically maintained for most clusters for most surveys. Less common are other outcomes that maintain the original cluster boundary, and still less common are outcomes that require creation of an artificial circular boundary. Outcomes that require visual evaluation and subjective determination are exceptionally rare.

Once a cluster boundary dataset has been finalized, an unmasked cluster centroid dataset is created by converting polygons to points, where each point is the geometric center of its respective polygon boundary, with the constraint that a point must be located inside its polygon boundary.

2.3 Creation of Final Geomasked Cluster Centroid Datasets

The geolocation of each cluster centroid is geomasked utilizing a “donut masking” method, whereby each cluster centroid is displaced a random direction and a random distance within a defined torus (i.e., “donut”) (Figure [#]). The inner and outer radii of the torus (D_{\min} and D_{\max} , respectively) are spatially adapted for each cluster, based on (1) the area of the cluster and (2) the population in and around the cluster, as described below.

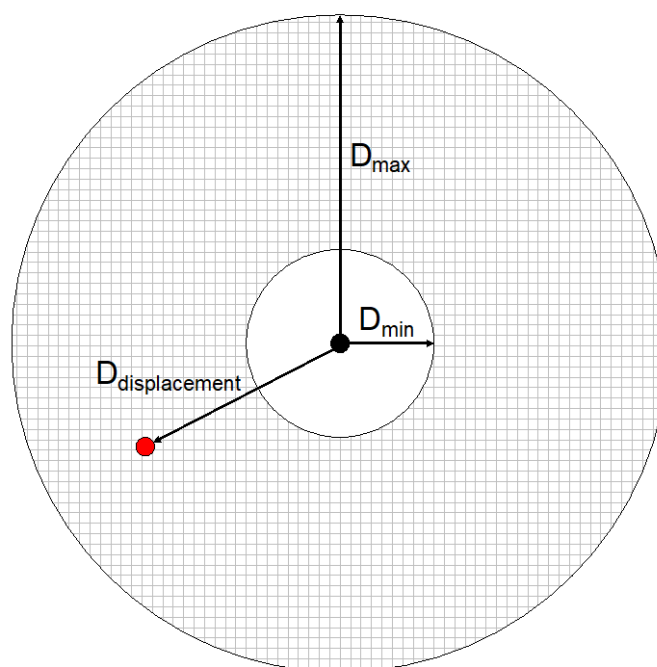
2.3.1 Determining the minimum and maximum displacement distance for each cluster

Each cluster centroid is displaced by at least a minimum distance (D_{\min}), where D_{\min} is defined such that the area of the circle defined by radius D_{\min} is equal to the area encompassed by a cluster’s boundary (Figure 2).

Each cluster centroid is displaced by at most a maximum distance (D_{\max}), where D_{\max} is defined such that the area of the torus defined by radii D_{\min} and D_{\max} encompasses a population at least five times the population encompassed by the circle defined by radius D_{\min} .

For each cluster, population estimates are determined by overlaying the circle with a radius equal to D_{\min} on a country-specific approximately 100-meter resolution gridded population count dataset from WorldPop (<http://www.worldpop.org.uk>). The radius of the circle is then increased in increments of $0.1 \times D_{\min}$ until the area of the torus (“donut”) defined by D_{\min} and D_{\max} encompasses at least five times the population encompassed by the circle (“donut hole”) defined by radius D_{\min} (Figure 2). D_{\max} is, however, constrained to a maximum value of 15 kilometers, irrespective of the urban-rural status of a cluster.

Figure 2 Illustration of the donut masking method



Each unmasked cluster centroid (black dot) is displaced by a random direction between 0 and 359 degrees and a random distance, $D_{\text{displacement}}$, between D_{min} and D_{max} to its geomasked cluster centroid location (red dot). D_{min} and D_{max} are spatially adapted for each cluster. D_{min} is adapted such that the area of the circle defined by radius D_{min} (white “donut hole”) is equal to the area encompassed by a cluster’s boundary. D_{max} is adapted such that the area of the torus defined by radii D_{min} and D_{max} (cross-hatched “donut”) encompasses a population at least five times the population encompassed by the circle defined by radius D_{min} .

2.3.2 Defining additional displacement constraints

While each cluster centroid is constrained to be randomly displaced a distance between D_{min} and D_{max} , two additional constraints are also imposed.

First, cluster centroids may not be displaced outside of the stratum-level administrative area in which the cluster is located. Stratum-level administrative areas typically correspond to level-one subnational units (SNU), such as provinces or regions. In such cases, the appropriate GADM (<https://gadm.org/index.html>) dataset is typically utilized to enforce this constraint. In a few cases, GADM does not contain current data for the requisite SNU boundaries. In such cases, an up-to-date SNU boundary dataset is typically provided by a country’s national statistical office. And, in other cases, stratum-level administrative areas correspond to specialized aggregations of SNUs, such as health zones. In such cases, the requisite stratum-level administrative area boundary dataset is typically provided by a country’s national statistical office; is obtained from The DHS Program Spatial Data Repository (<https://spatialdata.dhsprogram.com/home/>); or is constructed by PHIA Project personnel. Metadata provided in Appendices C through P specify the stratum-level administrative area boundary dataset utilized to enforce this constraint for each survey.

Second, cluster centroids may not be displaced inside a major water body, such as a large lake or an ocean. The World Water Bodies 2016-11-30 Data and Maps for ArcGIS is utilized to enforce this constraint for each survey.

2.3.3 Displacing cluster centroids

Each cluster is displaced by (1) randomly selecting a distance, $D_{\text{displacement}}$, from a uniform distribution of values between D_{min} and D_{max} ; (2) randomly selecting an angle from a uniform distribution of values between 0 and 359 degrees, where 0 is north and 90 is east; and (3) displacing the cluster centroid to the location specified by the random distance and random direction. If a cluster centroid is displaced outside of the stratum-level administrative area in which the cluster is located or inside a major water body, then the process is reiterated until the displacement adheres to all constraints. Summary statistics (minimum, maximum, mean, median, P25, P75) for $D_{\text{displacement}}$ for each survey are provided in the metadata in Appendices C through P.

2.3.4 Implementing final quality control

Finally, several quality control measures are implemented.

First, occasionally an unmasked cluster centroid is located just outside the boundary of the stratum-level administrative area in which it should be located — or an unmasked cluster centroid is located just inside a major water body. Such cases are typically due to inaccuracies in the geospatial data. In such cases, geomasking involves manual displacement of the unmasked cluster centroid into the appropriate stratum-level administrative area.

Second, any geomasked cluster centroid whose displacement requires more than one iteration of the random distance, random direction selection is visually inspected to ensure no geoprocessing errors have occurred. Such cases are typically associated with clusters located close to a stratum-level administrative area boundary or a major water body. Very rarely, a cluster centroid cannot be geomasked, or a cluster lacks geospatial data. Such clusters are omitted from the geospatial dataset and are documented as missing data in the metadata provided in Appendices C through P.

Third, each geospatial dataset is joined with its corresponding household and individual dataset to ensure consistency in key variables, including cluster unique identifier, stratum-level administrative area, and urban-rural status. On rare occasions, minor discrepancies occur, and these discrepancies are documented in the metadata provided in Appendices C through P.

2.3.5 Producing final geomasked cluster centroid datasets

After final quality control has been implemented for a geomasked cluster centroid dataset, the file is converted from shapefile to comma separated value (csv) format. The csv file contains only three variables: a cluster unique identifier as well as longitude and latitude in the WGS 84 GCS. No elevation data are provided in PHIA Project geomasked cluster centroid datasets. However, users interested in adding elevation information to a dataset may do so utilizing the appropriate GIS tools and a digital elevation model (DEM), such as the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) DEM or the Space Shuttle Radar Topography Mission (SRTM) DEM, both of which are available from the United States Geological Service (USGS) EarthExplorer (<https://earthexplorer.usgs.gov>).

3. PHIA Project Public-Release Geospatial Dataset Policy and Procedures

The Public-Release Geospatial Dataset for each PHIA Project survey contains three variables: cluster unique identifier, geomasked cluster centroid longitude, and geomasked cluster centroid latitude. The cluster unique identifier, *centroidid*, enables users to link the location of each geomasked cluster centroid to a corresponding record in the PHIA Public-Release Household Questionnaire, Individual Questionnaire, and Biomarker Datasets. The geographic coordinates for each cluster centroid have been be geomasked using a spatially adaptive random perturbation method described in Chapter 2. Appendices C through P provide metadata regarding the Public-Release Geospatial Dataset for each survey.

3.1 Policy for Public Access to and Use of Geospatial Datasets

Each survey's Public-Release Geospatial Dataset will become available to the public separately from and generally within six months after corresponding PHIA Public-Release Household Questionnaire, Individual Questionnaire, and Biomarker Datasets become available. To access a Public-Release Geospatial Dataset, interested individuals must complete a *PHIA Survey Public-Release Geospatial Dataset Access and Use Agreement*, which is available on the PHIA Project website (<https://phia-data.icap.columbia.edu/>).

3.2 Procedures for Requesting, Accessing, and Utilizing Geospatial Datasets

3.2.1 Requesting Geospatial Datasets

Interested individuals must utilize the online *PHIA Survey Public-Release Geospatial Dataset Access and Use Agreement*, which is found on the PHIA Project website (<https://phia-data.icap.columbia.edu/>). Each survey's Public-Release Geospatial Dataset must be requested separately. Requests will be reviewed, and requesters will be notified of a decision within 10 business days. If a request is approved, requesters will be provided with instructions on accessing the dataset; if a request is declined, requesters will be provided with a rationale for the refusal. All communications will be sent to a requester-specified email address.

3.2.2 Accessing and Using Geospatial Datasets

Researchers who are approved to use a Public-Release Geospatial Dataset will be provided with the dataset as csv, dta, and sas7bdat files via a secure file transfer protocol (FTP) link where they can download approved dataset(s). It is expected that the dataset(s) will be used by the researcher for the described analysis and not disseminated beyond the individuals identified in the *PHIA Survey Public-Release Geospatial Dataset Access and Use Agreement*.

Once downloaded, a Public-Release Geospatial Dataset can be merged with its corresponding household questionnaire, individual questionnaire, and biomarker datasets using a randomly generated cluster identifier variable, *centroidid*. This variable is specified as a two-letter country abbreviation and a six-digit sequential number, where the six-digit number starts at 000001 and runs up to the total number of cluster centroids in the country. The order of the clusters in the for *centroidid* is random, so closeness in location has no relation to closeness in the *centroidid* number.

The purpose of the *centroidid* variable is to facilitate joining geomasked cluster centroid coordinates to the Public-Release household questionnaire, individual questionnaire, and biomarker datasets. Each record in these datasets, representing either a household or an individual, includes a *centroidid* to facilitate this join. Sample SAS code for joining cluster centroid coordinates with Public-Release household questionnaire, individual questionnaire, and biomarker datasets may be found in Appendix A.

References

Allshouse W. B., Fitch M. K., Hampton K. H., et al. Geomasking sensitive health data and privacy protection: an evaluation using an E911 database. *Geocarto International*. 2010;25(6):443–452.

Brownstein J. S., Cassa C. A., Mandl K. D. No place to hide—reverse identification of patients from published maps. *New England Journal of Medicine*. 2006;355(16):1741–1742.

Curtis A. J., Mills J. W., Leitner M. Spatial confidentiality and GIS: re-engineering mortality locations from published maps about Hurricane Katrina. *International Journal of Health Geographics*. 2006;5, article 44.

Hampton K. H., Fitch M. K., Allshouse W. B., et al. Mapping health data: improved privacy protection with donut method geomasking. *American Journal of Epidemiology*. 2010;172(9):1062–1069.

Sherman, J. E., and T. L. Fetters. (2007). “Confidentiality Concerns with Mapping Survey Data in Reproductive Health Research.” *Studies in Family Planning* 38(4), 309–21.

Zandbergen PA. Ensuring Confidentiality of Geocoded Health Data: Assessing Geographic Masking Strategies for Individual-Level Data. *Adv Med*. 2014;2014:567049.

Appendix A Sample SAS code for joining geospatial datasets with household and individual datasets

```
* Change 'this2016' to the prefix on the datasets for the appropriate country,  
* for example, for Uganda all files have the prefix 'uphia2016' and for Malawi  
* they all have an 'mphias2015' prefix.;
```

```
%let survey = this2016;
```

```
libname geo "[path to coordinate dataset]";  
libname &survey. "[path to public release datasets]";
```

```
proc sql;  
    create table &survey._geojoin as  
    select b.*, g.Latitude, g.Longitude  
    from &survey..&survey.adultbio b  
    left join geo.&survey.centroids g  
    on b.CentroidID = g.CentroidID;  
quit;  
run;
```

Appendix B Cluster centroid geomasking Python script

Geomasking Script

```
1 import arcpy
2 import math
3 import logging
4 import numpy
5 from collections import OrderedDict
6
7
8 def my_msg(s, suppress_log=False):
9     if not suppress_log:
10         logger.info(s)
11     print s
12
13
14 def centroid_setup(in_layer, new_fields, drop_first=False):
15     try:
16         if drop_first:
17             arcpy.DeleteField_management(in_layer, new_fields.keys())
18             arcpy.DeleteField_management(in_layer, ["Centroid_Pop"])
19
20         for field in new_fields:
21             if len(arcpy.ListFields(in_layer, field)) != 1: # Only add field if it doesn't exist
22                 arcpy.AddField_management(in_layer, field, field_type=new_fields[field],
23                                           field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED")
24     except arcpy.ExecuteError:
25         my_msg(arcpy.GetMessages())
26
27
28 def get_centroid_population(masking_centroid, population_lyr):
29     arcpy.gp.ExtractMultiValuesToPoints_sa(masking_centroid, population_lyr + " Centroid_Pop", "NONE")
30
31
32 def calc_equal_area(in_centroid, in_polygon, min_radius):
33     centroid_fields = ['EA', 'EqualAreaRadius', 'Phia_Min']
34     polygon_fields = ['EA', 'shape@area']
35
36     with arcpy.da.UpdateCursor(in_centroid, centroid_fields) as cursor:
37         for row in cursor:
38             where_clause = "EA = '{0}'".format(row[0])
39             with arcpy.da.SearchCursor(in_polygon, polygon_fields, where_clause) as cursor2:
```

Page 1 of 10

Geomasking Script

```

40         for row2 in cursor2:
41             poly_area = row2[1]
42             temp_radius = math.sqrt(poly_area / math.pi)
43             if temp_radius < min_radius:
44                 row[1] = min_radius
45                 row[2] = min_radius
46             else:
47                 row[1] = temp_radius
48                 row[2] = temp_radius
49             cursor.updateRow(row)
50
51
52 def create_table(out_path, out_name, fields):
53     arcpy.CreateTable_management(out_path, out_name)
54
55     for field in fields:
56         arcpy.AddField_management(in_table=out_name, field_name=field, field_type=fields[field],
57                                   field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED")
58
59
60 def create_geomasked_fc(outpath, out_name, in_reference_feature_class):
61     out_path = outpath
62     out_name = out_name
63     geometry_type = "POINT"
64     has_m = "DISABLED"
65     has_z = "DISABLED"
66
67     spatial_reference = arcpy.Describe(in_reference_feature_class).spatialReference
68     arcpy.CreateFeatureclass_management(out_path, out_name, geometry_type, "", has_m, has_z, spatial_reference)
69
70     new_fields = {'EA': 'TEXT',
71                  'RndDistance': 'DOUBLE',
72                  'RndAngle': 'DOUBLE',
73                  'Count': 'SHORT'}
74
75     for field in new_fields:
76         arcpy.AddField_management(in_table=out_name, field_name=field, field_type=new_fields[field],
77                                   field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED")
78

```

Geomasking Script

```

79
80 def sum_population(in_centroid, raster_lyr, radius, where_clause):
81     population = 0
82     table_fields = ['EA', 'SUM']
83     try:
84         arcpy.SelectLayerByAttribute_management(in_centroid, "NEW_SELECTION", where_clause)
85         temp_buffer = arcpy.CreateFeatureclass_management("in_memory", "temp_buffer")
86         table = arcpy.CreateTable_management("in_memory", "temp_table")
87         arcpy.Buffer_analysis(in_centroid, temp_buffer, radius, "FULL", "ROUND", "NONE", "", "PLANAR")
88         arcpy.gp.ZonalStatisticsAsTable_sa(temp_buffer, "EA", raster_lyr, table, "DATA", "SUM")
89         with arcpy.da.SearchCursor(table, table_fields, where_clause) as cursor2:
90             for row2 in cursor2:
91                 population = row2[1]
92
93     except arcpy.ExecuteError:
94         my_msg("sum_population Error: {} and Distance {}".format(where_clause, radius))
95         population = -1
96     finally:
97         arcpy.Delete_management("in_memory")
98         return population
99
100
101 def calc_base_population(in_centroid, raster_lyr):
102     counter = 0
103     fields = ['EA', 'EqualAreaRadius', 'MinBuffer_pop', 'Flag', 'Phia_Min', 'Centroid_Pop']
104     ea_value = ""
105
106     with arcpy.da.UpdateCursor(in_centroid, fields) as cursor:
107         for row in cursor:
108             try:
109                 ea_value = row[0] # EA_Value
110                 radius = row[4] # Phia_Min
111                 radius_multiplier = 1.0
112                 where_clause = "EA = '{0}'".format(ea_value)
113                 population = sum_population(in_centroid, raster_lyr, radius, where_clause)
114                 if population == -1:
115                     population = row[5]
116                     my_msg('Sum Pop error trapped - Used Centroid Population')
117                 while population == 0:

```

Geomasking Script

```

118         radius_multiplier += 0.5
119         new_radius = radius * radius_multiplier
120         population = sum_population(in_centroid, raster_lyr, new_radius, where_clause)
121         row[4] = new_radius # Update Phia Min because of loop
122     if radius_multiplier > 1:
123         logger.info('Non Standard Base Population for EA {}. Radius Multiplier is {}'.format(ea_value, radius_multiplier))
124         row[3] = row[3] + 1 # Set Flag = 1 to indicate non standard min population / Radius.
125         # See Notes in Main method
126     row[2] = population
127     cursor.updateRow(row)
128     counter = counter + 1
129     my_msg("Base Zonal Counter: {}. EA - {}".format(str(counter), ea_value), True)
130 except Exception as e:
131     logger.error(str(e))
132     logger.error("Error in calc_base_population. EA = {}".format(ea_value))
133
134
135
136 def calc_max_radius(in_centroid, raster_lyr, target_k, max_distance):
137     fields = ['EA', 'Phia_Min', 'Phia_Max', 'MinBuffer_pop', 'MaxBuffer_pop', 'Flag']
138     counter = 1
139     with arcpy.da.UpdateCursor(in_centroid, fields) as cursor:
140         for row in cursor:
141             ea_value = row[0]
142             where_clause = "EA = '{}'".format(ea_value)
143             my_msg("Calc Max Radius: {}. EA - {}".format(str(counter), ea_value), True)
144             radius = row[1]
145             population = row[3]
146             radius_multi = numpy.arange(1.1, 20.5, 0.1)
147             for x in radius_multi:
148                 temp_pop = sum_population(in_centroid, raster_lyr, radius * x, where_clause)
149                 if (temp_pop - population) / population >= target_k:
150                     if radius * x > max_distance:
151                         row[2] = max_distance
152                         row[4] = sum_population(in_centroid, raster_lyr, max_distance, where_clause)
153                         my_msg('Maximum Radius Exceeded for EA {}.'.format(ea_value))
154                         row[5] = row[5] + 2 # Set the Flag variable - See Notes in Main method
155                         if row[1] > max_distance:
156                             row[1] = max_distance * .5

```


Geomasking Script

```

157         else:
158             row[2] = radius * x
159             row[4] = temp_pop
160             cursor.updateRow(row)
161             counter = counter + 1
162             break
163
164
165 def populate_admin(in_centroid, in_admin, admin1_column_name):
166     admin_fields = [admin1_column_name]
167     with arcpy.da.SearchCursor(in_admin, admin_fields) as cursor:
168         for row in cursor:
169             area_name = row[0]
170             where_clause = "{} = {}".format(arcpy.AddFieldDelimiters(in_admin, admin1_column_name), area_name)
171             arcpy.SelectLayerByAttribute_management(in_admin, "NEW_SELECTION", where_clause)
172             arcpy.SelectLayerByLocation_management(in_centroid, "INTERSECT", in_admin)
173             arcpy.CalculateField_management(in_table=in_centroid, field="Admin1", expression="{}".format(area_name),
174                                             expression_type="VB", code_block="")
175
176
177 def shift_xy(shape, angle, distance):
178     point = shape.getPart(0)
179     point.Y += distance * math.cos(math.radians(angle))
180     point.X += distance * math.sin(math.radians(angle))
181     return point
182
183
184 def is_not_within_lake(shape, layer_name):
185     try:
186         arcpy.MakeFeatureLayer_management(layer_name, "water")
187         with arcpy.da.SearchCursor("water", ['OBJECTID', 'SHAPE@']) as cursor:
188             for row in cursor:
189                 if row[1].contains(shape):
190                     return False
191             return True
192     except Exception as e:
193         my_msg(str(e))
194
195

```

Geomasking Script

```

196 def is_inside_admin(shape, layer_name, ea, column_name):
197     try:
198         expression = "{} = {}".format(column_name, ea)
199         arcpy.MakeFeatureLayer_management(layer_name, "admin1", expression)
200         with arcpy.da.SearchCursor(in_table="admin1", field_names=['SHAPE@', column_name]) as cursor:
201             for row in cursor:
202                 return row[0].contains(shape) # Only 1 zone should be selected. Returns TRUE if new point is contained
203     except Exception as e:
204         my_msg(str(e))
205
206
207 def generate_geomask(in_centroid, out_fc, fields, admin1_name, water_name, method, admin1_column_name):
208     new_fields = ['EA', 'RndDistance', 'RndAngle', 'SHAPE@XY', 'Count']
209     try:
210         with arcpy.da.SearchCursor(in_centroid, fields) as cursor:
211             for row in cursor:
212                 ea_value = row[0]
213                 where_clause = "EA = {}".format(ea_value)
214                 arcpy.SelectLayerByAttribute_management(in_centroid, "NEW_SELECTION", where_clause)
215                 temp_min = row[2]
216                 temp_max = row[3]
217                 counter = 0
218                 my_msg("Masking EA: {}".format(ea_value), True)
219                 while True:
220                     counter += 1
221                     random_distance = numpy.random.uniform(low=temp_min, high=temp_max)
222                     random_angle = numpy.random.uniform(low=0, high=360)
223                     new_point = shift_xy(row[1], random_angle, random_distance)
224                     not_inlake = is_not_within_lake(new_point, water_name)
225                     same_admin = is_inside_admin(new_point, admin1_name, row[4], admin1_column_name)
226                     if not_inlake and same_admin:
227                         insert_cursor = arcpy.da.InsertCursor(out_fc, new_fields)
228                         insert_cursor.insertRow((ea_value, random_distance, random_angle, new_point, counter))
229                         del insert_cursor
230                         break
231                     if counter > 100:
232                         my_msg("{} method. No geomask found for EA: {}".format(method, ea_value))
233                         break
234     except Exception as e:

```

Geomasking Script

```

235     logger.error("Error in generate_geomask. EA = {}".format(ea_value))
236     logger.error(str(e))
237
238
239 def main():
240
241     # NOTES:
242     # Country Specific Values - The country_values dictionary created below is country specific. Replace the values
243     # depending for each country.
244     # Ensure that all the keys are the same.
245
246     # ArcMap / ArcCatalog must be closed while running this script to avoid schema locks.
247
248     # To make this sample code generic, the values in the dictionary have been replace by generic values.
249     # Replace items surrounded by brackets with actuals values.
250     country_values = {
251         'country_name': '[CountryName]',
252         'geodatabase_path': '[Path to File GeoDatabase]',
253         'population_raster': '[Population Raster]',
254         'ea_centroid': 'Masking_Centroids',
255         'ea_polygon': 'Masking_EA',
256         'admin_constraint': 'Final_Admin',
257         'admin_column': '[column name]',
258         'water_features': '[water feature layer]',
259         'target_k': 5,
260         'max_k_radius': 15000,
261         'urban_rural_column': 'UrbanRural',
262         'random_iterations': 20,
263         'personal_geodatabase': '[Path to Personal Geodatabase]',
264         'log_file': '[Path to Log file]'
265     }
266
267     # NOTES:
268     # Flag Variable in the layer defined by: country_values['ea_centroid']
269     # 0 = Default
270     # 1 = Min Radius and Min Population are non-standard.
271     # This is caused by NULL population values under the buffers.
272     # I expand the buffer by half radius increments until a population is found.
273     # The Equal Area Radius is NOT equal to the Area of the EA.

```

Geomasking Script

```
274
275 #      2 = Maximum Buffer size set to 15000m. Desired k-value not attained.
276 #      If the minimum buffer size is also greater than the max value, the minimum size is set to 7500m.
277
278 #      4 = No Geomask solution found - Phia Method - Manual Placement of masked points required. May not be placed.
279 #      8 = No Geomask solution found - DHS Method - Manual Placement of masked points required. May not be placed.
280
281 arcpy.CheckOutExtension("Spatial")
282
283 # Internal Table & Feature Class Names Created by Script:
284 fc_phia_masked = "PhiaGeoMasked"
285
286 # Start
287 my_msg('Start geomasking run for {0}'.format(country_values['country_name']))
288
289 # Set up layers - Making Feature and Raster Layers
290 arcpy.env.workspace = country_values['geodatabase_path']
291 arcpy.env.overwriteOutput = True
292 arcpy.MakeRasterLayer_management(country_values['population_raster'], "raster", "#", "#", "#")
293 arcpy.MakeFeatureLayer_management(country_values['ea_centroid'], "centroids")
294 arcpy.MakeFeatureLayer_management(country_values['ea_polygon'], "polygons")
295 arcpy.MakeFeatureLayer_management(country_values['admin_constraint'], "admin1")
296
297 # Setting the output strings of MakeLayer to variables.
298 raster_lyr = "raster"
299 centroids_lyr = "centroids"
300 polygons_lyr = "polygons"
301 admin1_lyr = "admin1"
302
303 # Set up Raster Layer
304 arcpy.env.cellSize = raster_lyr
305 min_radius = math.ceil(float(arcpy.env.cellSize) / 2.0)
306 arcpy.env.pyramid = "NONE"
307
308 target_k = country_values['target_k']
309 max_distance = country_values['max_k_radius'] # maximum radius for finding the K-value.
310
311 # Brief description of the steps involved with this script.
312 # Step 1: Add Fields to EA centroid file
```

Geomasking Script

```

313     # Calculate population at centroid
314     # Step 2: Calculate the equal area radius for each polygon / centroid
315     # Step 3: For each equal area buffer, generate sum of population
316     # Step 4: Determine maximum radius. First radius that statisfies the target_k
317     # Step 5: Populate the Constraint field with spatial value
318     # Step 6: Generate a geomasked point
319
320     # Controlling which steps will run
321     b_step1 = True # centroid_setup()
322     b_step2 = True # calc_equal_area()
323     b_step3 = True # calc_base_population()
324     b_step4 = True # calc_max_radius()
325     b_step5 = True # Populate the Admin1 column in EA_Centroids
326     b_step6 = True # Generate a geomasked point
327
328     if b_step1:
329         my_msg('Started Step 1')
330         new_fields = OrderedDict()
331         new_fields['EqualAreaRadius'] = 'DOUBLE'
332         new_fields['MinBuffer_pop'] = 'DOUBLE'
333         new_fields['MaxBuffer_pop'] = 'DOUBLE'
334         new_fields['Phia_Min'] = 'DOUBLE'
335         new_fields['Phia_Max'] = 'DOUBLE'
336         new_fields['Admin1'] = 'TEXT'
337         new_fields['Flag'] = 'SHORT'
338         new_fields['MaskedFlag'] = 'SHORT'
339
340         centroid_setup(centroids_lyr, new_fields, True)
341         arcpy.CalculateField_management(in_table=centroids_lyr, field="Flag", expression="0",
342                                         expression_type="VB", code_block="")
343         get_centroid_population(masking_centroid=centroids_lyr, population_lyr=raster_lyr)
344         my_msg('Completed Step 1')
345
346     if b_step2:
347         my_msg('Started Step 2')
348         calc_equal_area(centroids_lyr, polygons_lyr, min_radius)
349         my_msg('Completed Step 2')
350
351     if b_step3:

```

Geomasking Script

```

352     my_msg('Started Step 3')
353     arcpy.SelectLayerByAttribute_management(centroids_lyr, "CLEAR_SELECTION")
354     calc_base_population(centroids_lyr, raster_lyr)
355     my_msg('Completed Step 3')
356
357     if b_step4:
358         my_msg('Started Step 4')
359         arcpy.SelectLayerByAttribute_management(centroids_lyr, "CLEAR_SELECTION")
360         calc_max_radius(centroids_lyr, raster_lyr, target_k, max_distance)
361         my_msg('Completed Step 4')
362
363     if b_step5:
364         my_msg('Started Step 5 - Populate Admin Constraint')
365         populate_admin(centroids_lyr, admin1_lyr, country_values['admin_column'])
366         my_msg('Completed Step 5 - Populate Admin Constraint')
367
368     if b_step6:
369         arcpy.SelectLayerByAttribute_management(centroids_lyr, "CLEAR_SELECTION")
370         my_msg('Started Step 6 - Generate Geomasked Point.')
371         if arcpy.Exists(fc_phia_masked):
372             arcpy.DeleteRows_management(fc_phia_masked)
373         else:
374             create_geomasked_fc(arcpy.env.workspace, fc_phia_masked, country_values['ea_centroid'])
375             fields = ['EA', 'SHAPE@', 'Phia_Min', 'Phia_Max', 'Admin1']
376             generate_geomask(centroids_lyr, fc_phia_masked, fields, country_values['admin_constraint'],
377                             country_values['water_features'], "Phia", country_values['admin_column'])
378             my_msg('Finished Step 6 - Generate Geomasked Point.')
379
380
381     if __name__ == '__main__':
382         logger = logging.getLogger(__name__)
383         logger.setLevel(logging.INFO)
384         handler = logging.FileHandler(['path to logfile'])
385         formatter = logging.Formatter(fmt='%(asctime)s.%(msecs)03d %(levelname)s %(module)s - %(funcName)s: %(message)s',
386                                     datefmt="%Y-%m-%d %H:%M:%S")
387         handler.setFormatter(formatter)
388         logger.addHandler(handler)
389         main()
390

```