Detailed Overview and Description of Components in U.S. Pipeline 2019

Authors:

Henning S. Mortveit (*)
Samarth Swarup
Dawen Xie
Hannah Baek
Mandy Wilson
Srini Venkatramanan

NSSAC Technical Report: #2019-025

Year:

2019

Last Updated:

13 Feb 2023

Summary: This document gives an overview of the core modules in the U.S. synthetic population pipeline, also referred to as the detailed population pipeline. Helper modules and their purposes are also described. Modeling will be described in a separate document. The current version and data formats described apply to US-SP 1.9 which is currently under construction.

1. Population Sharing

To be determined:

- Sharing of preliminary data? E.g.: not fully calibrated
- Sharing to what depth? (E.g., everything, people and locations etc)
- Shared formats taking off certain attributes (e.g. residence location details)
- Terms of sharing; sharing older populations (now: 2017)
- Carving out sub-regions
- Samarth: Texas shared

2. Automation, data storage directory structure and module signatures

Populations generated from the detailed pipeline will use the storage structure and construction parameters detailed in the following. In the automated pipeline, nodules will be called with arguments reflecting the construction and the region being considered. For now, the pipeline is only guaranteed to work with U.S. state. Extensions may be added in the future, but additional tools for projections and combinations may also be considered. The construction arguments are as follows with examples below:

- **base:** the base directory where output will be generated
- **instance:** the particular population instance being generated
- **subregion:** the sub-region of the larger region being constructed note (USPS lower-case state code)
- **activitylength:** the duration of activity sequences (either weekly or daily)
- **networkmodelchoice:** the string representing the contact network model (for now, only relevant for the population network module)

In addition, the following run parameters are broadcast to each module:

- **randomseed:** an integer to use as seed for random number generation
- **ntasks:** the maximal number of processes to request during any point of the module execution
- **walltime:** length of longest path in execution DAG for Slurm.

The following examples illustrates possible values for an invocation of the pipeline using the Rivanna cluster:

- **base**=/project/biocomplexity/nssac/SyntheticPopulations/DetailedDataPipeline/
- **instance**=usa_840/2017/ver_1_0
- **subregion**=va
- **activitylength**=weekly
- **networkmodelchoice=**m0
- **randomseed=**217845
- **ntasks=**4500
- **walltime=**48

Additionally, a module may optionally register one path string to point to pre-computed data, e.g., a collection of standard activity templates that is used by the module in question across invocation instances. This variable will be:

- **modulepathstring**

The module path string may include multiple paths in which case it is strongly advised that follow the UNIX convention by concatenating individual paths using ":".

Preprocessing: It is expected that some modules will require some preprocessing prior to starting a major pipeline execution generating an entire new U.S. population. As examples, the locations module may require an update to the NSSAC building database; the activity sequence module may require preparation of new standard input templates.

Documentation: Instance includes the region, the year deemed most representative as per data, and a version number. An instance will incorporate specific choices of data sources (e.g., ATUS, NHTS). Such information can either be documented in $root/instance.txt or inside module specific output directories following the conventions below.

Note: we will likely operate the pipeline in two stages using a construction instance and a production instance. Once a population is deemed adequate, a population may be moved/copied from construction- to production locations. This copy/move procedure is the responsibility of the pipeline, not the modules. For this – and many other reasons – a module must only use the supplied directory names to locate and generate data.

Structure

- The current set of standardized module names are:
  - base_population
  - home_location_assignment
  - activity_assignment
  - locations
  - location_assignment
  - population_network
- Standardized module names are used in the directory structure of the output, full details provided below.
- Each module provides a single binary or script that can be used to execute that module for a single instance (e.g., Virginia with subregion code va).
- Error and success code reporting: each module will return a single integer value upon termination; zero (0) encodes success, and any other value encodes an error.
- The pipeline will terminate at any point where a module returns a non-zero exit status.
- Each module may assume that all data generated by prior modules in the pipeline is present and error free at the point where the module is being invoked.
- Additional, per-module "module load" dependencies and required environment variables (e.g. R_LIBS) can be specified. This must be documented in the module's deployment instructions.
- Each module has a set of output files that it is required to generated following the naming convention described below. Upon module completion, the automation script will check for the presence of these files. Only upon successful termination (return value 0) and verification that the module's file(s) is(are) present, will the pipeline continue to its next stage.

Common module arguments

- Each module binary or script will be invoked with the following arguments/parameters:
  - `-m|-modulepathstring     <modulepathstring>`
  - `-b|-base                 <base>`
  - `-i|-instance             <instance>`
  - `-s|-subregion            <subregion>`
  - `-a|-activitylength       <(daily|weekly)`

```
-d|-moduleoutputdirectory     <moduleoutputdirectory>
-M                           {m0}
-n                           <ntasks>
-r                           <randomseed>
-t                           <walltime>
```

Module specific arguments
Currently, the population network module will have a special argument. This is to be deprecated with instructions to other modules to ignore arguments that are not recognized.

```
-M                           {m0}
```

Environment variables common to all modules

- Prior to executing each module, the following environment variables will be set and equal the command line arguments above:
  - `modulepathstring`
  - `base`
  - `instance`
  - `subregion`
  - `activitylength`
  - `moduleoutputdirectory`
  - `randomseed`
  - `jobmaxnumber`
  - `networkmodelchoice`

Structure of generated data

The following describes the structure of directories and names used for any invocation of the pipeline. The description uses the auxiliary variable:

- **`root:`**`=$base/$instance`

Note that instance specific data will be place in `$root` (e.g., location assignment's flow matrix). This is data common to all population constructions for the particular instance (e.g. U.S. 2017 with NHTS).

Base population data:

```
$root/${subregion}/base_population/:
     ${subregion}_person.csv
     ${subregion}_household.csv  (Note: must be sorted)
```

Household to residence assignment data:

```
$root/${subregion}/home_location_assignment:
     ${subregion}_household_residence_assignment.csv
```

Commented [W(1]: This entity is called "Household-to-Residence Mapping" later in this document. Should they be consistently named?

Commented [MHS(2R1]: Thank you, Mandy. I am reluctant to change this now. It could be part of a final cleanup after the pipeline has been completed and we have a ver. 1.0 of the U.S. states.

Location data:

```
$root/${subregion}/locations:
     ${subregion}_residence_locations.csv
     ${subregion}_activity_locations.csv
```

Activity assignment data:

```
$root/${subregion}/activity_assignment/{weekly|daily}
     ${subregion}_adult_activity_assignment_{week|day}.csv
     ${subregion}_child_activity_assignment_{week|day}.csv
```

Location assignment data:

```
$root/${subregion}/location_assignment/{weekly|daily}:
     ${subregion}_adult_activity_location_assignment_{week|day}.csv
     ${subregion}_child_activity_location_assignment_{week|day}.csv
```

Population network data:

```
$root/${subregion}/population_network/{weekly|daily}:
     ${subregion}_population_network_{networkmodelchoice}_{week|day}.csv
```

NOTE: the naming of the network file now includes construction parameters.

Definition: A synthetic population for a (sub-)region is a collection data files, precisely one of each from the above list, organized in one common directory, that were constructed in a compatible manner from the detailed pipeline. Specifically, it would contain the following files:

- ${subregion}_person.csv
- ${subregion}_household.csv
- ${subregion}_household_residence_assignment.csv
- ${subregion}_residence_locations.csv
- ${subregion}_activity_locations.csv
- ${subregion}_adult_activity_assignment_{week|day}.csv
- ${subregion}_child_activity_assignment_{week|day}.csv
- ${subregion}_adult_activity_location_assignment_{week|day}.csv
- ${subregion}_child_activity_location_assignment_{week|day}.csv
- ${subregion}_population_network_{networkmodelchoice}_{week|day}.csv

For SciDuct purposes, a population is a collection of references (or IDs) to the same collection of files constructed in a compatible manner from the population pipeline. Note that a population may have multiple population networks (as an example), and thus that there is a need to have a notion of record that can register such intances, and that is potentially different than a container containing output from an invocation.

Note: can consider this section after Mandy's completion of export tool.

3. Automated pipeline

The automated pipeline source is located here:
git@github.com:NSSAC/PopulationPipeline
Documention will be included when complete.

4. Construction parameters and notes

Activity encoding:

```
Home:     1
Work:     2
Shopping: 3
Other:    4
School:   5
College:  6
Religion: 7
(Transit: 0 – not an official NSSAC activity, but appears in
activity assignment files (only))
```

Note regarding IDs: note that IDs are generally only unique within a population. Populations from different regions, or different version of populations from the same region, will generally have incompatible IDs. In particular, be aware of the following:

- pid, hid, alid, rlid: not unique across instances;
- alid, rlid: these become lid (location IDs) but *they overlap* causing ambiguity after location assignment and in the network construction. They can be pulled apart by (a) the modeling assumption that only Home activities happen at residences (and the lid of a such a located activity is thus an rlid), and (b) in the population network where an offset of 1,000,000,000 has been added to rlids to form lids.

Both points above are being resolved in the current development of the NSSAC BDB (building database).

5. Base population construction [Samarth]

Person file (CSV):

```
hid,pid,serialno,person_number,record_type,age,relationship,sex,school_enroll
ment,grade_level_attending,employment_status,occupation_socp,race,hispanic,de
signation
0,0,2014000633042,1,P,55,0,1,1,,1,533020,1,1,education
1,1,2016001009388,1,P,77,0,1,1,,6,,1,1,none
1,2,2016001009388,2,P,77,1,2,1,,6,119013,1,1,none
2,3,2016001169185,1,P,39,0,1,1,,1,512090,1,1,none
```

- **hid:** Household ID

- **pid:** Person ID
- **serialno:** Housing unit serial number from PUMS
- **person_number:** numbering of persons within household (SPORDER in PUMS)
- **record_type:** {P: Person; H: Household} (RT in PUMS)
- **age:** age of person (AGEP in PUMS)
- **relationship:** relationship to reference person (RELP in PUMS)
- **sex:** {1: MALE; 2: FEMALE} (SEX in PUMS)
- **school_enrollment:** whether this person is in school, and type of school (SCH in PUMS)
- **grade_level_attending:** if in school, what grade (SCHG in PUMS)
- **employment_status:** employed or not, civilian, or armed forces (ESR in PUMS)
- **occupation_socp:** Occupation recode based on 2010 codes (SOCP in PUMS)
- **race:** race of householder (see Section 13 under modeling; RAC1P in PUMS)
- **hispanic:** a Boolean (see Section 13 under modeling; HISP in PUMS)
- **designation:** NAICS-derived string (separate document to be pulled in)

NOTE: possibly make comment about data elements that from census and possibly inconsistent in the synthetic populations.

Household file (CSV):

```
admin1,admin2,admin3,admin4,hid,serialno,puma,record_type,hh_unit_wt,hh_size,
vehicles,hh_income,units_in_structure,business,heating_fuel,household_languag
e,family_type_and_employment_status,workers_in_family
51,1,90100,1,0,2015001351876,51125,H,17, 4,2,55000,1,2,2,1,7,2
51,1,90100,1,100,2015001501486,51125,H,13,1,2,42000,2,2,3,1,,
51,1,90100,1,101,2017000226403,51125,H,21,2,2,52000,2,9,3,1,2,1
51,1,90100,1,10,2017000289124,51125,H,35,9,4,41900,2,9,3,2,5,2
51,1,90100,1,102,2016000240524,51125,H,9,2,2,128600,2,9,3,1,4,0
```

Commented [W(3]: Typo that made it into the output file sample.

Commented [MHS(4R3]: Fixed. Samarth said he has updated. Should be clean at this point.

- **admin1:** FIPS code for state (ST);
- **admin2:** FIPS code for county;
- **admin3:** FIPS code for tract;
- **admin4:** FIPS code for blockgroup (note: blockgroups do not have a distinct FIPS code; they are just numbered from 1 within each tract)
- **hid:** Household ID; same as in the person file;
- **serialno:** Housing unit serial number from PUMS
- **puma:** Public Use Microdata Area code based on 2010 Census definitions
- **record_type:** {P: Person; H: Household} (RT)
- **hh_unit_wt:** Housing unit weight (WGTP)
- **hh_size:** Number of persons in family (NPF)
- **vehicles:** Vehicles available (VEH)
- **hh_income:** Household income in the past 12 months (HINCP) in local currency
- **units_in_structure:** Units in structure (type of residence) (BLD)
- **business:** Business or medical office on property (BUS)
- **heating_fuel:** House heating fuel (HFL)
- **household_language:** Household language (HHL)

- **family_type_and_employment_status:** Married couple or not, in labor force or not (FES)
- **workers_in_family:** Workers in the family in the past 12 months (WIF)

GitHub repository:
https://github.com/NSSAC/basepop-ipf

6. Household grouping [Samarth]

Census information typically comes in two basic varieties: individual- and household-level PUMS. For the U.S., PUMS are given for household permitting on to construct households directly through sampling. For (most) other regions throughout the world, census information is given at the level of individuals. For such countries, one will need to form households in other ways, e.g., by obtaining household composition statistics from other sources such as the European Social Survey.

For now, this section is a placeholder for a module handling household grouping for non-U.S. regions. This is an internal module in the sense that a complete household file will must have this modeling step included. More generally, this step would be a population partition encompassing traditional household as well as grouping relevant for group quarters.

7. Location construction [Dawen]

Output format specification(s)

Residence locations file (CSV):

```
rlid,longitude,latitude,altitude,admin1,admin2,admin3,admin4,area_sqm,associa
te_link_func_class,pub_pk,pub_kg,pub_01,pub_02,pub_03,pub_04,pub_05,pub_06,pu
b_07,pub_08,pub_09,pub_10,pub_11,pub_12
1052998,-82.3277705,29.6538605,-1,12,001,000200,1,179,5,185323,185323,\
185323,185323,185323,185323,185323,185318,185318,185318,185318,185318,\
185318,185318
1114821,-82.332464,29.6548805,-1,12,001,000200,1,125,5,185323,185323,185323,\
185323,185323,185323,185323,185266,185266,185266,185266,185266,185266,185266
1114827,-82.3280665,29.654216,-1,12,001,000200,1,72,5,185323,185323,185323,\
185323,185323,185323,185323,185266,185266,185266,185266,185266,185266,185266
```

- **rlid**: residence location ID (unique within subregion only)
- **longitude**: the longitude of the location
- **latitude**: the latitude of the location
- **altitude**: the altitude of the location, set to -1 for now
- **admin1**: 2-digit FIPS code for the state
- **admin2**: 3-digit FIPS code for the county
- **admin3**: 6-digit FIPS code for the census track
- **admin4**: 1-digit FIPS code for the block group

- **area_sqm**: building's area in square meter
- **associate_link_func_class**: functional class for the associate HERE link
- **pub_pk**: NCES school ID for closest public school that offers preschool
- **pub_kg**: NCES school ID for closest public school that offers kindergarten
- **pub_01**: NCES school ID for closest public school that offers 1st grade
- **pub_02**: NCES school ID for closest public school that offers 2nd grade
- **pub_03**: NCES school ID for closest public school that offers 3rd grade
- **pub_04**: NCES school ID for closest public school that offers 4th grade
- **pub_05**: NCES school ID for closest public school that offers 5th grade
- **pub_06**: NCES school ID for closest public school that offers 6th grade
- **pub_07**: NCES school ID for closest public school that offers 7th grade
- **pub_08**: NCES school ID for closest public school that offers 8th grade
- **pub_09**: NCES school ID for closest public school that offers 9th grade
- **pub_10**: NCES school ID for closest public school that offers 10th grade
- **pub_11**: NCES school ID for closest public school that offers 11th grade
- **pub_12**: NCES school ID for closest public school that offers 12th grade

Activity locations file:

Note: the activity locations file will in general have a varying number of columns reflecting the activities being used in the construction for the synthetic population. There is one column for each type appearing in the activity encoding listed above, with the exception that activity `Home` is omitted.

Output format specification(s)

Activity locations file (CSV):

```
alid,longitude,latitude,altitude,admin1,admin2,admin3,admin4,wor
k,shopping,school,other,college,religion,designation
1,-99.45144,27.53944,-1,48,479,001714,2,1,0,0,1,0,0,none
2,-98.75611,33.3707,-1,48,503,950200,2,1,218,0,1,0,0,none:retail
3,-98.18166,26.26202,-1,48,215,023902,2,1,0,0,1,0,0,none
4,-95.40831,29.94019,-
1,48,201,222600,1,1,471,0,1,0,0,none:retail
```

- **alid**: the activity location ID (unique with subregion only)
- **longitude**: the longitude of the location
- **latitude**: the latitude of the location
- **altitude**: the altitude of the location, set to -1 for now
- **admin1**: 2-digit FIPS code for the state
- **admin2**: 3-digit FIPS code for the county
- **admin3**: 6-digit FIPS code for the census track
- **admin4**: 1-digit FIPS code for the block group
- **work**: attractor weight for work activity

- **shopping**: attractor weight for shopping activity
- **school**: attractor weight for school activity
- **other**: attractor weight for other activity
- **college**: attractor weight for college activity
- **religion**: attractor weight for religion activity
- **designation:** [string] colon-separated designation strings of person designations supported for activity “Work” at the location

Repository information

Codes used to augment data and generate residence/activities are in following repository: `https://github.com/NSSAC/GeoDatabase/tree/master/USLocations`

Codes used to assign closest public school for MS building data are in following repository: `https://github.com/NSSAC/GeoDatabase/tree/master/Augmentation/building_to_closest_school`

8. Household-to-residence mapping [Samarth]

Output format specification(s)

Residence assignment file (CSV):

```
hid,rlid,longitude,latitude,altitude,admin1,admin2,admin3,admin4,area_sqm,associate_link_func_cla
ss,pub_pk,pub_kg,pub_01,pub_02,pub_03,pub_04,pub_05,pub_06,pub_07,pub_08,pub_09,pub_10,pub_11,pub
_12
0,29652,-75.3518219000019,37.9524336745262,-1,51,001,090100,1,266,5,77651,77651,77651,\
77651,77651,77651,77651,77652,77652,77652,77652,77652,77652,77652
100,2733792,-75.3601628525653,37.9434599722783,-1,51,001,090100,1,160,5,77651,77651,77651,\
77651,77651,77651,77651,77652,77652,77652,77652,77652,77652,77652
101,1792446,-75.3388035,37.9624605,-1,51,001,090100,1,203,5,77651,77651,77651,77651,\
77651,77651,77651,77652,77652,77652,77652,77652,77652,77652
10,1557218,-75.3363510889829,37.9575134076075,-1,51,001,090100,1,225,5,77651,77651,\
77651,77651,77651,77651,77651,77652,77652,77652,77652,77652,77652,77652
102,793574,-75.3514908904536,37.9518685812242,-1,51,001,090100,1,244,5,77651,77651,\
77651,77651,77651,77651,77651,77652,77652,77652,77652,77652,77652,77652
103,1469235,-75.352363,37.9507715,-1,51,001,090100,1,110,5,77651,77651,\
77651,77651,77651,77651,77651,77652,77652,77652,77652,77652,77652,77652
```

Dictionary notes: this file combines the `hid` from the household file with the fields in the residence locations file; see the respective sections for the elements of the data dictionary.

GitHub repository:

`https://github.com/NSSAC/assign_home_locations`

9. Activity sequence template construction for weekly sequences [Hannah]

The activity sequence template framework first collects and unifies activity templates from a broad set of sources as well as versions in the form of *standard input templates*. Its output is a precursor to the activity assignment module described in Section 4.

10. Activity sequence assignment to population [Hannah]

Output format specification(s)

Adult and child activity files (CSV):

```
hid,pid,activity_number,activity_type,detailed_activity,start_time,duration,mode,driver_flag,passenger_flag,month,day,survey_id
0,0,0,1,1,0,27840,-1,-1,-1,3,1,14070
0,0,1,0,0,27840,120,4,1,2,10,2,6020
0,0,2,4,6,27960,300,-1,-1,-1,10,2,6020
0,0,3,0,0,28260,1140,4,1,2,10,2,6020
0,0,4,2,3,29400,25500,-1,-1,-1,10,2,6020
0,0,5,0,0,54900,300,4,1,2,10,2,6020
0,0,6,3,11,55200,1200,-1,-1,-1,10,2,6020
```

Commented [MHS(5]: How about activity_modifier and defining this further resolution of activity_type? E.g. home + cooking. activity_modifier definitions would be given separately for each activity type.

- **hid:** the household ID of the person;
- **pid:** the person ID;
- **activity_number:** the activity number in the sequence;
- **activity_type:** the activity type (note: includes 0 – transit);
- **detailed_activity:** [HSM: need detailed definition];
- **start_time:** the start time of the activity measured in seconds since T0 (midnight between Sunday and Monday);
- **duration:** the duration of the activity measured in seconds;
- **mode:** the travel mode used to get to the activity [HSM: need encoding]
- **driver_flag:** a Boolean value; true only if person is the driver; [HSM: need encoding];
- **passenger_flag:** [HSM: need encoding];
- **month:** an integer giving the month of the activity [HSM: how was this determined?]
- **day:** the day of the activity [HSM: 1 – Monday – verify]
- **survey_id:** the ID of the survey

GitHub repository:

```
git@github.com:NSSAC/ActivityAssignment.git
```

11. Location assignment [Henning;Mandy]

Output format specification(s)

Location assignment file (CSV):

```
hid,pid,activity_number,activity_type,start_time,duration,lid,longitude,latitude,travel_mode
71,173,2,3,19800,900,70735,-75.6536,37.782,-1
71,173,4,4,21300,600,11786,-75.6799,37.7237,-1
71,173,6,2,23400,39600,55530,-75.3532,37.9235,-1
72,175,2,2,24300,33300,77511,-75.7508,37.6405,-1
72,176,2,2,25200,32400,58793,-75.6297,37.7488,-1
73,177,2,4,40320,1080,55535,-75.3537,37.9246,-1
```

- **hid:** the household ID of the person
- **pid:** the person ID of the person
- **activity_number:** the number of the activity in the activity sequence to which it belongs
- **activity_type:** the activity type (HSM: see code reference)
- **start_time:** the start time of activity in seconds since midnight Sunday/Monday
- **duration:** the duration of the activity in seconds
- **lid:** the location ID of the location where the activity takes place. Note that that for a `Home` activity type, the location ID is a residence location ID; otherwise, it is an activity location ID
- **longitude:** the longitude of the location (HSM: note to check on precision in output)
- **latitude:** the latitude of the location (HSM: note to check on precision in output)
- **travel_mode:** an enumeration value specifying the travel mode used immediately prior to the activity specified. This field is currently unused and set to its default value of -1.

Note: location IDs and residence IDs use separate indexing. To have easy access to the coordinates of the location, the (longitude, latitude) pair is included in the output, thus saving conditional lookup into files or tables for residences and activity locations.

GitHub repository:
`https://github.com/NSSAC/LocationAssignment`

12. Network construction [Henning;Samarth;Abhijin;Mandy]

This module generates the *population network* and performs *sub-location modeling.* Whereas these steps were done separately in the past, performance issues resulting from network size makes the combined version more scalable.

Output format specification(s)

The module operates under the assumption that times are specified in seconds with *t* = 0 corresponding to midnight Sunday/Monday. This is referenced at T_0. No time-of-year (date) is specified. Times are assumed to satisfy *t* \ge 0, and although there are no upper bound, a value of *t* greater than one week (in seconds) is likely a mistake.

Population network file (CSV):

```
pid1,pid2,daynum,lid,slid,starttime,duration,activity1,activity2
1510010902002007501,1510010902002007500,0,1,-1,0,25200,1,1
1510010902002014101,1510010902002014100,0,1,-1,0,27000,1,1
1510010902002049501,1510010902002049500,0,1,-1,0,34200,1,1
1510010902002073601,1510010902002073600,0,1,-1,0,30000,1,1
```

- **pid1:** the pid of the source node;
- **pid2:** the pid of the target node;
- **daynum:** the day at which the contact started;
- **lid:** location ID where contact took place; (see note in location assignment module)
- **slid:** sub-location ID (-1 if undefined)
- **starttime:** the start time of the activity measured in seconds since T_0;
- **duration:** the duration of the contact measured in seconds;
- **activity1:** the activity type of the source node during the contact;
- **activity2:** the activity type of the target node during the contact;

Days are enumerated as

- Monday: 0; time interval: [0x24x3600, 1x24x3600-1],
- Tuesday: 1; time interval: [1x24x3600, 2x24x3600-1],
- …
- Sunday: 6; time interval: [6x24x3600, 7x24x3600-1],

and `daynum` can be computed as floor(`starttime`/86,400).

GitHub repository:

https://github.com/NSSAC/ConstructSocialContactNetwork-ER

Additional Supporting Tools

Composite regions

Method for composite regions (e.g. 2 states)
Status:
Implemented and tested on WMATA (Contact: Henning)

Projection methods

Methods for projection onto a particular sub-region covered by a population.
Status:
Implemented and tested (Contact: Mandy)

Population export/sharing

Note:

- Involves selection of columns from each file.
- Shared under CC-BY-4.0
  Contact: Henning

Network Export [Henning]

The following is not part of the pipeline per se, but form the glue to other applications and tools with NSSAC.

Please note the following for constructions of export formats of a population network.

- An edge may span multiple days. In principle, it can cover an entire week (the max duration at the moment) or longer. Depending on the application, such edges may have to be split into the respective days with adjustments to daynum as appropriate.
- Depending on the application, accumulation of contacts/edges within a day (or accumulation window) may have to be done. The precise details related to accumulation will depend on the application (e.g., will accumulation be agnostic to the activity types and combinations thereof)

Input format specification.

- A population network file (with sub-location modeling), see Section 10;

Some export tools may require additional information (e.g., specific location data to assess whether a location is indoors/outdoors)

Social contact network
The social contact network corresponding to a population network is a collection of one or more undirected networks obtained by accumulating all contacts taking place inside 24 hour windows. The network does only consider durations of contact with each day.

Output format specification(s)

```
#sourcePID,sourceActivity,targetPID,targetActivity,duration,dayNum
```

- **sourcePID:** the PID of the source node;
- **sourceActivity:** the activity type of the source node during the contact;
- **targetPID:** the PID of the target node;
- **targetActivity:** the activity type of the target node during the contact;

- **duration:** the total time of contact between source and target taking place within the time interval corresponding to dayNum;
- **dayNum:** the day at which the activity started;

Questions: options in accumulation. Regard or disregard activities? Isolated vertices?

EpiHiper network export

The ASCII EpiHiper network format is described in the EpiHiper technical report Section 2.4.

Modeling

13. Base population construction

IPF was done using five variables:

a. Household income:
   i. HINCP in the PUMS,
   ii. Table B19037 in the ACS summary data (AGE OF HOUSEHOLDER BY HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2017 INFLATION-ADJUSTED DOLLARS))
   iii. hh_income in the synthetic population

b. Age of householder:
   i. AGEP in PUMS
   ii. Table B19037 in the ACS summary data (AGE OF HOUSEHOLDER BY HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2017 INFLATION-ADJUSTED DOLLARS))
   iii. age in the synthetic population

c. Household size:
   i. NP in PUMS
   ii. Table B11016 in the ACS summary data (HOUSEHOLD TYPE BY HOUSEHOLD SIZE)
   iii. hh_size in the synthetic population

d. Householder is Hispanic: this marginal for IPF is just 'not Hispanic/Hispanic'.
   i. HISP in PUMS
   ii. Summary distribution constructed from the tables B25003 (TENURE) and B25003I (TENURE (HISPANIC OR LATINO HOUSEHOLDER)) in the ACS summary data
   iii. hispanic in the synthetic population

e. Race of householder: this marginal for IPF is '1 .White alone 2 .Black or African American alone 3 .American Indian alone 4 .Alaska Native alone 5 .American Indian and Alaska Native tribes specified; or .American Indian or Alaska Native, not specified
and no other .races 6 .Asian alone 7 .Native Hawaiian and Other Pacific Islander alone 8 .Some Other Race alone 9 .Two or More Races'
   i. RAC1P in PUMS
   ii. Table B25006 in the ACS summary data (RACE OF HOUSEHOLDER)
   iii. race in the synthetic population

Designation: This column is based on the NAICSP variable in PUMS, as we had generated post hoc in the previous round. Code for this is now integrated into the base population generation step in a fairly general way. This is done by adding a column called MODULE to the fieldname_mapping_person.csv and fieldname_mapping_household.csv files. When a variable has to be pre-processed/transformed before adding to the synthetic population, we can write a short python script to do it and include its name in the MODULE column for that variable in the appropriate fieldname_mapping file. You can see the generate_designation.py script in the git repository for an example.

14. Household grouping [Samarth]

Census information comes in two basic varieties: individual- and household-level PUMS. For the U.S., PUMS are given for household permitting on to construct households directly by sampling. For (most) other regions, census information is given at the level of individuals. For such countries, one will need to form households in other ways, e.g., by obtaining household composition statistics from other sources such as the ESS.

For now, this section is a placeholder for future non-U.S. regions. This is an internal module in the sense that a complete household file will have had to modeling step included, see the previous section.

15. Activity sequence template construction for weekly sequences [Hannah]

Need description of method and the organization into directories. E.g., the Mon;Tue-Wed-Thu;Fri;Sat;Sun breakdown, and the methodology to combine these for a particular individual.

Note: Please describe issue present in the Netherlands. Here there are complete week-long surveys, which can possibly be used for validation, but there are consistency issues for the associated demographic information.

16. Activity sequence assignment to population [Hannah]

Need description of assignment methodology.

17. Location construction [Dawen]

6.1 Modeling

In this module, the goal is to generate residence locations and activity locations, one State at a time.

The data sources for this module are:

- Microsoft U.S. Building Footprints (version 1.1)
- HERE Streets Prime US 2019 Q1 with extended POIs
- Schools related data from National Center for Education Statistics (NCES)
- Shapefile for U.S. block groups (2017) from U.S. Census Bureau

We use MS building data as the primary source to generate residence locations. On high level, we run augmentation processes on MS building data to compute/assign following information for each building:

- Building centroid
- Building footprint in square meter
- 12-digit U.S. block group code
- NSSAC augmentation flag, i.e., **aug_flag** (details can be found: https://github.com/NSSAC/GeoDatabase/blob/master/USLocations/README.res.locations.md#augmentation
- Closest public school at each grade level, i.e., pk, kg, 01, 02, …, 12

Once all buildings have been augmented, we generate initial set of residence locations. Currently residence locations are defined as **aug_flag = 4 OR aug_flag = 5**. We merge data from MS building table and closest school tables and store it in a residence locations table.

Since it is possible that some block groups do not have any MS building, or there is no MS building in the block group was classified as residence location in above step, for each such block group, we use a random point within block group boundary with area sets to 500 square meters as the residence location. We also compute closest public schools for newly generated locations.

We use POIs tables in HERE Premium US 2019 Q1 with extended POIs data as the primary source to generate activity locations. In particular, we use the following 12 tables:

- Auto maintenance, service
- Business facilities
- Community server center
- Entertainment
- Financial Institutions
- Hospitals
- Parks and Recreation
- Restaurants
- Shopping
- Transportation hub
- Travel destinations, e.g., hotel
- Church locations from extended listing

For each POI in above list, we first assign 12-digit block group code. (More details needed. Do we need to explain each activity type? I need some help explain this part. DX)

6.2 Input format specification(s)

All input for this module are tables in database. In particular, we are using a database called **geodb** on server **postgis1** and below are datasets and their schema mappings.

Dataset	Schema
Census geography data (2017)	CENSUS_2017
HERE Premium Streets with Extended POI Listing	HERE_US_2019_Q1

MS US Building Footprints	MS_US_BUILDINGS_V1_1
NCES school data	NCES

For census geography data, the source shapefiles were downloaded from U.S. Census Bureau and loaded to CENSUS_2017 schema. There's no augmentation done on table level except that we created an additional table to store block groups for all U.S. States.

For HERE premiums streets with extended POI listing data, the purchased data were loaded to HERE_US_2019_Q1 schema. HERE data is organized using Database Coverage Area (DCA) and sub DCA. For each sub DCA, there are a fixed number of tables associated and sub DCA is used as part of the table name. For example, the corresponding sub DCA for State of Virginia is 83, the table is named as **dca_83_DataClass** in which DataClass corresponds to different data. A State could have one or more sub DCAs, for example, State of California has 5 sub DCAs and they are 11, 12, 13, 14, and 15. A sub DCA could cover more than one State, for example, sub DCA 83 covers Virginia and North Carolina.

For State that has more than one sub DCAs, we introduce a concept called **Pseudo sub DCA** to streamline augmentation process. We use **State's USPS code plus '_slim'** as the pseudo sub DCA code and create same set of database tables; then we populate limit fields from each sub DCA table to corresponding pseudo sub DCA table. Once this pre-processing is done, we can use pseudo sub DCA code to access all tables that are used in our location generation.

For MS US building footprints data, the data (version 1.1) were downloaded from https://github.com/microsoft/USBuildingFootprints and loaded to MS_US_BUILDINGS_V1_1 schema.

School related data were generated and curated from NCES. This includes geo-locations data for public schools, private schools and postsecondary schools. It also includes school enrollment, FTE counts and characteristic like whether a given grade is offered. Details on curation process can be found in: https://github.com/NSSAC/GeoDatabase/blob/master/USLocations/README.sch.locations.md

18. Location assignment [Henning]

NOTE: NEED UPDATE REGARDING DESIGNATION CONSTRAINED ASSIGNMENT (separate document to be pulled in)

NOTE: there may be cases in the code where non-anchor activities do not get assigned a location: Notes from Stephen:

> **From:** "Eubank, Stephen Gardner (sge3qp)" <eubank@virginia.edu>
> **Date:** Thursday, 24, October, 2019 at 21:31
> **To:** "Wilson, Mandy L (alw4ey)" <alw4ey@virginia.edu>, "Mortveit, Henning S (hsm2v)"

<Henning.Mortveit@virginia.edu>
Subject: Re: Question about LocationAssignment output

Here's my best guess, after looking for a path through the code that avoids error messages but doesn't assign a valid location:

For a non-anchor activity (currently, that's shopping and other), we randomly choose a location from the work zone. If there are no locations in the work zone with an attractor for the activity type, we choose the home zone. If the home zone also has no attractors for the activity type, we choose a random zone biased by the flow from the home zone. (This happens at line 631 of LocationAssignment.C, if you're reading along at home) I never check to see whether that randomly chosen zone has any attractors for the activity type, so it's possible that the location assigned is invalid.

That would produce the symptoms you see, and at a fairly low rate: it's unlikely that neither the home nor work nor randomly chosen zones have the right kind of activity, and the number of shop and other activities is kind of small compared to the number of anchor activities. So without any actual evidence, I'll go with that. To fix it, you could just put the random zone selection in a loop until it finds one with the needed activity. Inefficient, but seems like a rare case. Otherwise, I'd construct a vector with the total attraction for the activity type by zone and somehow weight it by the zone-zone flow, and choose randomly from that.

Stephen

For a specific region (e.g. state)

19. Network construction [Henning;Samarth;Abhijin]

This module generates the *population network* and performs *sub-location modeling.* Whereas these steps were done separately in the past, performance issues resulting from network size makes the combined version more scalable.

The population network module first generates all possible contacts for all locations. In reality, co-occupying a location does not necessarily imply an interaction or contact. Within-location modeling represent an additional modeling step selecting a subset of the edges from the initial population network to better capture contacts as they occur inside a location. Clearly, the nature of contacts may differ by location type as well as activity and other factors. Currently, all models approach this through edge thinning.

Possible models for locations include:

- An Erdos-Renyi random graph instance *G* in $G_{n,p}$. Challenge: what is an appropriate value for *p*? Advantage: simple subset selection of edges.
- Independent cliques of size at most *k.* This is the earlier model used for detailed population modeling.
- Subgraph models, e.g., as for Montgomery VA high schools.

Input format specification(s)

- A location assignment output file, see Section 8;

Output format specification(s)

The module operates under the assumption that times are specified in seconds with $t = 0$ corresponding to midnight Sunday/Monday. This is referenced at T_0. No time-of-year (date) is specified. Times are assumed to satisfy *t* \ge 0, and although there are no upper bound, a value of *t* greater than one week (in seconds) is likely a mistake.

QUESTION: is there any compelling reason to generate a directed network? The export tools can easily construct that whenever needed. No

NOTE: as a temporary solution to avoid overlapping IDs between residences and activity locations, an offset is added to all residence IDs causing them to all become larger than the largest activity location ID. This step will be removed as soon as this issue is resolved in residence instantiation.

20. Network and Population Export [Henning]

Please note the following for constructions of export formats of a population network.

- An edge may span multiple days. In principle, it can cover an entire week (the max duration at the moment) or longer. Depending on the application, such edges may have to be split into the respective days with adjustments to dayNum as appropriate.
- Depending on the application, accumulation of contacts/edges within a day (or accumulation window) may have to be done. The precise details related to accumulation will depend on the application (e.g., will accumulation be agnostic to the activity types and combinations thereof)

Input format specification.

- A population network file (with sub-location modeling), see Section 10;

Some export tools may require additional information (e.g., specific location data to assess whether a location is indoors/outdoors)

Social contact network
The social contact network corresponding to a population network is a collection of one or more undirected networks obtained by accumulating all contacts taking place inside 24 hour windows. The network does only consider durations of contact with each day.