Version date: Feb 2018
Authors: Carolyn Orbann and Lisa Sattenspiel,
SAMort model provided the stimulus for many of the structures and methods used in this model.

INTRODUCTION

The SanDiego model is an epidemiological model designed to test hypotheses related to the spread of the 1805 measles epidemic among indigenous residents of Mission San Diego during the early mission period in Alta California. The model community is based on the population of the Mission San Diego community, as listed in the parish documents (baptismal, marriage, and death records). Model agents are placed on a map-like grid that consists of houses, the mission church, a women's dormitory (monjeria) adjacent to the church, a communal kitchen, priest's quarters, and agricultural fields. They engage in daily activities that reflect known ethnographic patterns of behavior at the mission. A pathogen is introduced into the community and then it spreads throughout the population as a consequence of individual agent movements and interactions.

MODEL INITIALIZATION

During the set-up procedure, the agent and map variable files are read in, the community layout and visualization are established, data output files are created, and variables for recording epidemic data are initialized. The first case (or cases, if desired) are selected. Users may decide whether to select a first case at random or according to certain criteria (e.g. an agent of a particular sex, age or occupation type).

EXTERNAL INPUT FILES

The model requires the use of two external input files, one to read in essential agent characteristics, and one to read in building characteristics. To facilitate explorations of the impact of population size on epidemic outcomes, we have made a number of different agent files and associated building files corresponding to target population sizes. In making different sized agent populations, households are kept together. Each larger population includes all agents from the smaller population files, with newly added households chosen randomly from among remaining households included in the full population until the target population size is reached. Because agents can repeatedly attempt to visit empty houses in small population runs, we eliminated from the building files all houses without any assigned agents. We always assume that one priest lives in the community. He ministers to the entire population. This priest is also assumed to be permanently and totally immune to measles, as a consequence of childhood exposure.

AGENT CHARACTERISTICS

Agent definition files (SDFullpop.txt and other similar files for different population sizes) include the following variables:

1. The first column of this file is the user-determined agent ID.

2. The second column of this file is the Record Identification Number (RIN) corresponding to the real person's entry in the geneology program file (ancestral quest). This RIN allows connection to the baptismal, marriage, and death record from the San Diego.

3. Column 3 is the id number of the mission. San Diego is designated as 1, and is the only mission in the data at this time.

4. Column 4 is a user generated father's patriline id (theoretical only, may be connected to a known clan names in the future).

5. Column 5 is a user generated father's matriline id (theoretical only, may be connected to a known clan names in the future).

6. Column 6 is a user generated mother's patriline id (theoretical only, may be connected to a known clan names in the future).

7. Column 7 is a user generated mother's matriline id (theoretical only, may be connected to a known clan names in the future).

8. Column 8 is an agent's disease status. Currently this variable is initialized at zero for all agents; the program picks an initial infected or exposed agent (or agents) and resets its disease status whenever appropriate, given the agent's infection state. (0=susceptible, 1=exposed, 2=infected, 3=recovered, 4=dead).

9. Column 9 is the dwelling of an agent. This is the agent's home base. The assigned numbers correspond to particular buildings on the model space. For example, individuals assigned to dwelling #35 will use building #35 as their home unless no spaces are available, in which case the program reassigns a new permanent dwelling. Dwellings may change depending on epidemic-related events (deaths in the household, for example)

10. Column 10 designates nuclear family membership. The variable allows connections between closely related individuals, such as siblings. Currently not being used, all set at 0.

11. Column 11 indicates sex of an agent (0 = male, 1 = female). The assigned sex corresponds to information inferred from baptismal and marriage data.

12. Column 12 designates an agent's age in years.

13. Column 13 corresponds to an agent's relative health status. This variable is designed to take into account different possible influences that may impact an agent's outcome when faced with a potential disease-transmitting contact. At present, this variable can range from -1 to 1, with -1 corresponding to a maximum negative impact (i.e., 100% reduction), 0 corresponding to no impact on health, and 1 corresponding to a maximum positive impact. In the current input data files, health-history is set at 0 for all agents because this is something that will be incorporated into the model later.

14. Column 14 designates an agent's occupation. This variable is user-defined and influences the activity patterns of agents. All agents have been assigned a 3-digit occupation code. The assignment rubric is described in the section on occupation categories.

15. Column 15 designates the ID of the agent's spouse. Spouse information information comes from the San Diego parish marriage and baptismal records. All unmarried agents are given a spouse ID of 0.

BUILDING CHARACTERISTICS

The present model has seven building types that we have designed to reflect important places at Mission San Diego, our study community: houses, a monjeria (women's dorm), priest quarters, a kitchen, a church, agricultural fields, and an unused storage area. The number of buildings of each type is calculated by the program as the community map is initialized. Building definition files (SDbldgs100.txt and other similar files for different population sizes) include the following variables:

1. Columns 1 and 2 give the coordinates of the lower left hand cell of a building (x-coordinate, then y-coordinate).

2. Columns 3 and 4 give the dimensions of the building (width, then length).

3. Column 5 is the building ID, assigned by the user.

4. Column 6 designates the building type (house=1, monjeria=2, priest quarter=3, kitchen=4, church=5, agricultural fields= 6, storage area = 7).

ESSENTIAL PARAMETERS

The model consists of a number of sliders that can be used to adjust the values of essential parameters. At the present time all parameters are set at constant values. Eventually some parameters other than run length and population size may be modeled using a probability distribution rather than constant values. The slider variables include the following:

1. Length of latent period: This is the number of time ticks that an agent remains in the exposed category. The slider is set up to range from 0 to 480 ticks (0 to 20 days in the present model). A reasonable baseline value of 240 ticks (10 days) was derived from an assessment of various values published in the measles literature (e.g. Kim-Farley 1993).

2. Length of infectious period: This is the number of time ticks that an agent remains infectious. This slider is also set up to range from 0 to 288 ticks (0 to 12 days in the present model). A baseline value of 144 ticks (6 days) was derived from an assessment of various values published in the measles literature (e.g.Kim-Farley 1993).

3. Transmission probability: This slider is set up to range between 0 and 1 and corresponds to the probability of transmission when a contact occurs between susceptible and infectious agents. A baseline value of 0.025 per contact was chosen to achieve an estimated transmission probability of at least 90%,

as is suggested by published accounts of measles transmission in unvaccinated populations (Aaby et al 1983; Shanks et al 2011; and Wolfe 1982).

4. Probability of death: This slider also ranges from 0 to 1 and corresponds to the probability of death per tick. Death can only occur in the model during the infectious period. The baseline estimate of the death probability was derived by setting the case fatality rate (cfr) equal to $(1-(1-d)^{i})$, where d is equal to the probability of death per tick and i is the length of the infectious period in ticks. The quantity $(1-d)^{i}$ gives the overall probability of NOT dying throughout the period of risk. When this value is subtracted from 1 it gives the cfr (overall probability that an infected individual dies at some time during the infectious period). The equation is then solved for d to give the desired probability of dying PER TICK. Multiple studies indicate extreme mortality during the 1805/06 epidemic in California (Cook 1976 and Milliken 1995), as high as 25% of the entire mission population at some northern missions. Since this epidemic is assumed to be a virgin soil epidemic, it is reasonable to assume that everyone was or could be infected, which means generates a cfr of 0.25. For an infectious period of 144 ticks (6 days), $1-(1-d)^{144} = 0.25$. The solution to this equation gives a per tick probability of death of 0.001996. NOTE: the model at present assumes that an agent is at risk of dying only when it is infectious and that the risk is equal for each tick it is infectious.

5. Run length: This is the number of time ticks the simulation will be run.

6. Start tick: This slider allows one to start the epidemic on a specified day. The simulation starts at midnight (12am) on Monday morning.

OCCUPATION CATEGORIES

All occupations have been assigned a 3-digit number that is a multiple of 100. The first digit corresponds to the general type of occupation. The model at present does not separate agents within specific occupational classes, but we retain the 3-digit code to allow this to happen later. The overall classification is as follows:

100 -- married men age 13 and over. These agents work in the fields all day Monday through Saturday. The agents are distinguished from single men because when they die, there are special considerations for their wives, all female children, and preschool aged male children.

200 -- married women age 11 and over and widows age 45 and older. These agents spend their occupational time Mon-Sat daytime weaving and doing other household activities in the vicinity of their house. While weaving, they can be either inside or outside, but will only range a maximum of 2 cells from the boundary of their house.

300 -- single men age 13 and over. These agents work in the fields all day Monday through Saturday. They are not parents and generally live with their own parents or other patrilineal relatives rather than in their own households.

400 -- single women ages 11 to 45. All, including widows under age 45, are required to live in the monjeria. They spend all their time there except for church and optional Sunday afternoon visits to relatives' houses.

500 -- male children (age 4-12). The primary responsibility of these agents is to go to the kitchen and bring back the food for each household meal. They can also hang out with other boys at the kitchen (which acts as a proxy social center) at other times of the day. They go to the church one hour each day for catechism, and they may go out to the fields in the afternoon to help older males, with the probability of that activity increasing as they get older.

600 -- female children (age 4-10). Girls also go to catechism one hour a day. Outside of that, they stay with their mothers and do what they do.

700 -- infants (age 0-3). Stay with their mothers and do what they do. During the day Mon-Sat, when mothers and infants are at home, all agents move within the house ± 2 cells. The infants are allowed to be outside when the mothers are inside and vice versa (under the assumption that they may be napping on a bed or under a tree and that other children or neighbors are also around to help keep an eye on them).

800 -- priests. Priests do all church services and conduct the catechism. Outside of these times during the day, they visit random locations in the community, including the fields and the storeroom. The assumption is that because they are responsible for the entire community, they may need to check up on places even if other agents aren't there. During the evening times that the other agents are visiting each other, the priests are in their quarters alone. This is their time to prepare lessons, paperwork, sermons, letters, etc. and to pursue their own religious devotions.

STEP-DEPENDENT PROCEDURES

The general structure of the go procedure follows. The information in quotes at the end of each component gives the section in which more information can be found.

1. Variables that indicate that infection or death occurred during the previous time step are reset so that these events happening during the present time step are tallied properly.
2. A randomly chosen turtle determines the daily time block corresponding to the current tick of the model ("schedule");
3. SDagents that are exposed or infectious update their disease status ("disease-related procedures");
4. all living SAagents engage in appropriate activities determined by their occ-type and the specific time block corresponding to the current tick of the model ("schedule" and "movement-related procedures");
5. the SAagents with appropriate disease statuses determine to or from whom they might transmit the disease ("disease-related procedures");
6. newly dead ghosts reassign any dependents to new caretakers or the monjeria as appropriate ("death-related procedures");
7. data output files and interface plots are updated ("display and output procedures").

SCHEDULE

The program is set up for 24 time ticks per day with each time tick 1 hour long. A series of statements using the variable 'timekeeper' sets up a schedule within which agent activities will occur, with the week starting on Monday at 1am. Values of timekeeper between 1 and 24 correspond to successive hours on Mondays through Saturdays; values of timekeeper between 25 and 48 correspond to Sunday time slots.

The findDaysActivities method sends an agent to the proper activities for the specific time tick indicated by the timekeeper variable (and designated by the specific do…acts method for that time tick). Although there are 24 time slots per day, there are no do...acts methods for timekeeper = 1-6 (1am-6am Monday through Saturday), 23-24 (11pm-12am Monday through Saturday), 25-30 (1am-6am Sunday), and 47-48 (11pm Sunday -12am Monnday) because these are times when agents are assumed to be sleeping.

MOVEMENT-RELATED PROCEDURES

The model contains a specific "do…acts" method for each of the time slots other than 11pm-6am daily (when agents are assumed to sleep). These methods specify particular behaviors for agents based on their occupation. All of which, involve some type of move method. The model contains 4 basic move methods: move-home, move-weave, move-visit, and move-to-bldg. **Move-home** is called whenever an agent either stays at home from one time slot to the next (in which case, they move to another available location within their dwelling) or returns home after spending the previous time slot elsewhere. **Move-weave** is called by any females that do not live the in monjeria and the infants and toddlers being cared for by those females. The method has agents find spots inside or nearby their dwellings to simulate the domestic activities of women Mission San Diego. **Move-visit** governs the movement of families when they visit other households during designated visiting times. The method calls two sub-methods: _find-visit-dwelling_, which identifies a location to visit that is the home of other members of the family's lineage, currently occupied, and has enough space for the entire family, and _move-group_, which ensures that all family members do move to the chosen visit location. The fourth move method, **Move-to-bld** is a general method that requires the input of a specific building's id and ensures that the calling agent moves to that building. This is used to send agents to the kitchen, church, and to the fields and it is also used by the priest, who can move to any location on the space.

DISEASE RELATED PROCEDURES

The underlying disease transmission model in this simulation is an SEIR epidemic process. All agents begin the simulation in the susceptible state, and then the status of one randomly chosen agent is set to "exposed". The specification of multiple initial infected individuals or specific types of initial cases can be made through simple adjustments of the "infect-first-case" method. Consistent with the SEIR epidemic process, exposed agents convert to the infectious state after a user-specified latent period, which they remain in until recovery (after a user-specified infectious period) or death (which occurs with a user-specified probability during each tick of the infectious period). Immunity is permanent upon recovery. The "update-disease-status" method governs this series of transitions.

Disease transmission can occur between susceptible-infectious pairs of agents that are adjacent to each other on the grid. Agents are considered to be adjacent if they are von Neumann neighbors, i.e., those to the north, south, east, or west. When an agent moves it checks its own disease status as well as that of its neighbors. If a moving agent is susceptible, it calls the method "transmit-from" for all infectious neighbors; if it is infectious, it calls "transmit-to" for all susceptibles. In both cases, transmission itself is determined by comparing the user-specified transmission probability to a randomly chosen number between 0 and 1. If transmission occurs, the status of the susceptible agent(s) is set to "exposed" and the clock for the disease process begins.

DEATH DEPENDENT PROCEDURES

Death of an agent -- agents have a set probability of dying at each tick of the infectious period. Upon death, a ghost (a different turtle breed) with the same agent characteristics as the dying agent is made and data about where and when the agent died are collected. The agents move to a "cemetery" (lower left-hand corner of the grid); the ghosts remain at the location of death. Users can control whether the ghosts are visible or hidden. Each dying agent also takes the shape of a ghost, with the option to make the size proportional to the number of agents that have died. In this case, the ghost that appears in the cemetery gets larger as the epidemic progresses.

Consequences for survivors after an agent's death -- what happens to surviving family members after an agent's death depends on the characteristics of the dying agent (eg sex, age, marital status), as well as the age and sex of the survivors.

1. The dying agent is a married male -- if his widow is of reproductive age, she, their small children of either sex, and all of their older daughters move to the monjeria. Older sons stay in the house if any other adults are present (eg a women over age 45, or adult brother). If no adults are present, they try to find a kin house. If that is not available, they just stay in the house for the duration of the epidemic.
2. The dying agent is an unmarried male -- If he is not a caretaker, no changes are made upon his death. If he is a caretake, the only possible children in the house are males who then try to find a kin house to move to. If that is not available, they stay alone in the house.
3. The dying agent is an unmarried female -- If she over age 45, she has to make the same decisions as an unmarried male (ie. what happens depends on whether she is sharing her house with children). If she is under age 45, she should be living in the monjeria and her death does not require any changes in residence of others.
4. The dying agent is a married woman -- If she is already widowed (possible because we don't change the occupation number of a widowed female), she is treated like an unmarried female. If the dying agent is not widowed, then the consequences depend on whether she has children and their age and sex. With the procedures set up to guarantee that children end up in a kin house if possible and if not possible, small children and all daughters go to the monjeria.

5. The dying agent is a child of any occupation -- nothing needs to be done. No other agents are reassigned when children die
6. The dying agent is a priests -- ERROR! Priests are assumed to be immune to measles and don't have a chance of death in this simulation.

DISPLAY AND OUTPUT PROCEDURES

As the simulation proceeds, a graph showing the numbers of susceptible, exposed, infectious, recovered, and dead agents is created and updated each tick. In addition, three csv (comma-delimited) output files are produced (SanDiego-cases.txt, SanDiego-daily.txt, and SanDiego-Final.txt). In each file, run numbers, global parameters (e.g. transmission probability), and attributes of the first case are always recorded. The “Cases” file also records, for all individuals in the model population, the place and time the individual was infected (the default value of -1 is recorded if the individual escapes the simulated epidemic), and if applicable, the place and time it died as well as characteristics of the agent that infected this individual. It also includes parents matriline and patrline information for both the infector and infected agents. The “Daily” file records the number of individuals in each disease status at each tick of the simulation and keeps track of the number of agents that are newly infected and newly dead. The “Final” file records the total number of individuals in each disease status at the end of the simulation, the total number of individuals ever infected [RD], and a count to verify all members of the model community were either susceptible or removed (recovered or dead) [SRD]. This last count provides an easy way to determine that the simulated epidemic finished in the allotted time; in that case the count will equal the total population size.

ACKNOWLEDGEMENTS

The code for this model was adapted from a Repast model developed by Lisa Sattenspiel, and Carolyn Orbann at the University of Missouri. The code for reading in the external tab-delimited files that contain building and agent characteristics was modified from code written by Uri Wilensky and submitted to Netlogo's model library. He has waived all copyright and related or neighboring rights to the sample code.