ECNA Model Overview

|  |  |  |
| --- | --- | --- |
|  | **Description** | **Procedure Name** |
| **0** | **Initialize Simulation Parameters:**   * Load Neural Network Matrices * Initialize global variables and outputs * Initialize Path model stopping condition ‘maxTicks’ * Initialize transmission rate * Initialize partnership exposure distribution | **setupECNA:**   * **get-mats** * **Initialize-output, setup-intermediate-globals, setup-static-globals** * **initialize-exposures** * **setup-ECNA-globals** * **setup-ECNA-people** * **write-stage-proportion-header** |
| **1** | **Initialize ECNA global parameters:**   * Determine the number of total nodes in the simulation. * Determine the number of infected nodes as stopping condition**.** * Determine the proportion of initial infected. * Determine the exposures per month * Determine the scale-free degree distribution to be used | **setup-ECNA-globals:**   * **normalizeDegreeDist** * **scaleDegreeDistToPop-matrix-version** |
| **2** | **Initialize degree distribution:**   * Normalize the scale-free distribution across the total number of bins. | **normalizeDegreeDist** |
| **3** | **Initialize compartmental degree distribution by age:**   * Generate matrix of age group x logarithmic degree bin, using initial number of people in each age group and distributing each row across the scale free degree distribution. | **scaleDegreeDistToPop-matrix-version** |
| **4** | **Initialize the initial infected people and their attributes:**   * Creates non-agent susceptibles * Create initial infected and set their age, desired degree, partners per month., time of infection, stage of disease, diagnosis time * Initialize contacts [5] * Kill links attached to agents with no infected contacts * Calculates the actual degree of each person * Count number of susceptibles exposed to an infected person. * Initialize max degree | **setup-ECNA-people:**   * **initializeTrans**   + **setDiagTime** * **initialize contacts** * **layout-ECNA Ttree-links** * **kill-not-needed-links** * **check-degree** |
| **5** | **Initialize number of contacts activated at each time for new infected node:**   * Using lifetime-degree distribution of partners, initialize a vector (num-of-contacts) of the number of contacts activated at each age group of the node. * Convert the num-of-contacts vector to a binary vector representation, where a non-zero value is 1 and zero value is 0. This vector (age-link-active) is used for optimization later on. | **initialize contacts** |
| **6** | **Main ECNA Simulation:**   * Every year,   + Age the susceptible non-agent population[7]   + Reset partnerships   + Assign sexual behavior based on degree   + Calculate the exposure per year, based on the exposure probability * Determines if a susceptible agent will become infected[8] * Calculates the proportion of people infected * Determine nodes eligible to be newly added susceptible contacts[9] * Update the actual degree for each person * Calculate the average degree of the newly infected person * Set time-infected of newly infected nodes. * Update PATH Disease Progression   + Match continuum care proportions based on timestep (last 10 years to match 2006-2015, everything before that dry run of 2006. First 10 months to generate network matching 2006.) * Update simulation and all attributes that change every timestep. | **runECNA:**  **goECNA:**   * **age-population-yearly** * **reset-partnerships** * **assign-sexual-behavior** * **calc-exposure-prob** * **infect-population-modified** * **calc-prop-infected** * **determine-non-eligible-modified** * **check-degree** * **calc-avg-inf-degree** * **reset** * **(PATH disease progression part)** |
| **7** | **Move non-agent susceptibles between age groups every year:**   * People move in to the population (the first age group) at a constant birth rate. * People move from age group i to j at a rate of 1/(age-interval of i). * Count the number of people moving out of the last age group | **age-population-yearly** |
| **8** | **Determine if a susceptible contact will become infected or not:**   * Loop through infected alive people with susceptible contacts aged less than 85   + Only let receivers be those whose link is active within that age group.   + Determine probability of infection (transmission rate, based on stage of infected, infectiousness, exposures per partner, the proportion of acts condom protected )   + If infected, initialize time of infection, stage of disease, diagnosis time | **infect-population-modified:**   * **initializeTrans**   + **setDiagTime** |
| **9** | **Determine people eligible to be newly added susceptible contact:**   * For every newly infected person,-   + Determine the number of contacts to add, i.e. if a newly infected person has actual degree and the desired degree , add links for this person   + Determine the degree of the newly added contacts by drawing from the degree correlation distribution , where is the desired degree of the newly added contacts   + Calculate the age-link-active vector for each new contact, and formulate a matrix (N) of the vectors.   + Calculate the probability matrix (V) of the infected node in age i, having a partner of age j, by multiplying (row-wise) the num-of-contacts vector into the age-mixing-matrix.   Match the age-link-active of the infected node and the newly added contacts using the optimization module[10]   * + Using optimization result, determine time each link should be activated and deactivated, and determine current age of the neighbor.   + Determine persons who are eligible to be the new added contacts, i.e. their desired degree should be equal to actual degree should be less than , they should have an active link at the same time the infected node has an active link (as determined by the optimization module), and they should be susceptible   + Randomly choose the new contacts to add from among all those who are eligible. Add the contacts to the simulation.   + Update degree distribution list by decrementing the degree bins corresponding to the desired degree of the newly added person (If the randomly chosen persons are not agents in the simulation, then update the susceptible age x degree bin list by decrementing the degree bins corresponding to the degree and age group of new contacts and initialize-contacts; otherwise, no need to update) * Update vectors (age link active and num-of-contacts) for each newly added susceptible contact. * Once link requirements of infected have been fulfilled set their vectors (num-of-contacts and age-link-active) to 0 * Create the links. | **determine-non-eligible-modified:**   * **calculate-dist-NN** * **desired-neighbor-degree-calc** * **initialize-contacts-neighbors** * **return-Vmatrix** * **optimization-mystic** * **link-activation-details** |
| **10** | **Optimization in Python**   * Min * Subject to:   + Xij is binary | **optimization-mystic** |