This document describes the functions available in the github repository github.com/tsleng93/SocialBubble, which are used in the manuscript *The effectiveness of social bubbles as part of a COVID-19 lockdown exit strategy, a modelling study* by Leng et al. While in that study, we consider households and the effect of introducing bubbles to an epidemic, and consider three age classes (children, adults, and older adults), our simulation methods are general, and could be used for an arbitrary probability matrix with an arbitrary number of risk classes.

Functions

1. **HouseholdMakerAge** - This function outputs an adjacency matrix of household contacts, `M`, an adjacency matrix of bubble contacts, `B`, as well as several other vectors that can be used to produce specific bubble scenarios.

   `[M, B, C, Age, BH, SizeHouse, TypeHouse, Position] = HouseholdMakerAge(NumHouse, House_List ProbHouse, House_Sizes, SizeBubble,SizeBubble2))`

   Inputs

   - `NumHouse` is the number of houses we want in our synthetic population.
   - `House_List` is a list of house types that we select from to form house types. Each 'house type' is a $1 \times a$ array, where a is the size of the largest household from the distribution we are drawing from. Different age groups are denoted by different positive integers, and if a house type has less members than the size of the largest households, all other positions have a value of 0.

     For example, a household type including a couple of age groups 3 and 4 (which in our model represent the age bands 20 to 29 and 30 to 39), and the largest household has size 6, then this household type is represented as [3 4 0 0 0 0].

     Hence `House_List` is a $b \times a$ array, where b is the number of household types in the distribution we are drawing from.
   - `ProbHouse` is a $b \times 1$ array of the probabilities of selecting a specific house type from `House_List`
   - `House_Sizes` is a list of the sizes of houses from `House_List` and is a $b \times 1$ array.
   - `SizeBubble` is the number of houses within each bubble that are store in the adjacency matrix `B`. In this instance, all households enter into bubbles.
   - `SizeBubble2` is the number of houses within each bubble that are stored in the adjacency matrix `BH`. In this instance, only the second half of households enter into bubbles. .

   Outputs

   - `M` is an adjacency matrix of household contacts drawn from the household distribution given by `House_List` and `ProbHouse`
   - `B` is an adjacency matrix of bubble contacts, where each household has entered into a bubble with `SizeBubble`-1 other households. In this intsance, all households have entered into bubbles.
   - `C` is a $1 \times |M|$ array, containing the size of each individual's household.
   - `Age` is a $1 \times |M|$ array containing the age group of each individual.
   - `BH` is an adjacency matrix of bubble contacts, where each household has entered into a bubble with `SizeBubble2`-1 other households. In this intsance, only the second half of households have entered into bubbles.
   - `SizeHouse` is a $1\times$ `NumHouse` array containing the size of each household.
   - `TypeHouse` is a $1\times$ `NumHouse` array containing whether or not a house contains a child of < 10 (denoted by 1), a child < 20 (denoted by 2) or contains no children (denoted by 0).
   - `Position` is a $1\times$ `NumHouse` array containing the position in the population of the first member of each household.

2. **InfectionProcessIndividual** - This function simulates the first 10 generations of 1000 epidemics on a population with infectious contacts given by `M`, which is a pruned adjacency matrix, i.e. an adjacency matrix containing only the connections that will result in an infection.

   `[EpiSize, RSize, Rgen, Igen, Deaths] = InfectionProcessIndividual(M, eps, C, Infect_0, A, RelTrans, RelInf, Death_Prop, randnum)`

   Inputs

- `M` is a pruned adjacency matrix (see PruneMatrixFull)
- `eps` is the baseline meanfield transmission rate to an individual i (i.e. it is ϵ from the manuscript).
- `C` is a $1 \times |M|$ array, containing the size of each individual's household. To set an individual's baseline mean-field transmission to be independent of household size, set `C` to be a an array of 1s.
- `Infect_0` is the initial number of infected individuals.
- `A` is a $1 \times |M|$ array containing the age group of each individual.
- `RelTrans` is a 1-D array that contains the relative transmissibility (scaling the rate an individual transmits infection) of each age group (i.e. it is T from the manuscript).
- `RelInf` is a 1-D array that contains the relative susceptibility to infection (scaling the rate of tranmsmission to an individual), of each each group (i.e. it is C from the manuscript).
- `Deaths_Prop` is the expected proportion of deaths upon infection for each age group, i.e. it is the case fatality ratio for each age group.
- `randnum` is an integer random seed, to replicate results if desirable.

Outputs

- `EpiSize` is the mean proportion of individuals infected.
- `RSize` is the obtained estimate for R, given by the fourth entry in Rgen.
- `Rgen` is a 1×9 array containing the mean reproduction number estimate for each generation, as defined by the equation for R in the manuscript
- `Igen` is an array containing the mean number of infected individuals in each generation, for the first nine generations.
- `Deaths` is the mean number of deaths.

3. **InfectionProcessIndividual2** - This function is equivalent to `InfectionProcessIndividual`, but it stores an extra output which slows down computation.

   `[EpiSize, RSize, Rgen, Igen, Deaths, Indiv_Infected] = InfectionProcessIndividual2(M, eps, C, Infect_0, A, RelTrans, RelInf, Death_Prop, randnum)`

   Extra Outputs

   - `Indiv_Infected` is a $1 \times |M|$ array containing the mean number of infections for each individual, i.e. the proportion of simulations in which each individual is infected.

4. **NonAdherenceBubble** - this function creates an adjacency matrix for bubble connections, where a proportion d of eligible households enter into social bubbles. This function receives its name because this is the function used to create the additional bubble contacts B_2 in the nonadherence counterfactual scenarios.

   `NonAdherenceBubble(H, TypeorSize, Position, d)`

   Inputs

   - `H` is an adjacency matrix of household contacts.
   - `TypeorSize` is a logical $1\times$ `NumHouse` array. It contains information around whether a household has a particular attribute, such as whether it is a certain 'type' (e.g. a household with children) or whether it is a certain size (e.g. a single occupancy household).
   - `Position` is a $1\times$ `NumHouse` array containing the position in the population of the first member of each household.
   - `d` is the proportion of eligible households that we want to form into a bubble.

   Outputs

   - `B` is an adjacency matrix of bubble contacts.

5. **PruneMatrixFull** - this function samples an underlying probability matrix, defined by a combination of the inputs. Doing so, we end up with a 'pruned' adjacency matrix, where we retain only those contacts where infection will be passed on (conditional on the transmitting individual being infected).

   `[NewM, SAR] = PruneMatrixFull(M,tau, Type, A, RelTrans, RelInf)`

   Inputs

   - `M` is an adjacency matrix of either household or bubble contacts (these must be pruned separately, as they have different transmission rates).
   - `tau` is the baseline transmission rate (household or bubble), depending on the type of adjacency matrix we wish to prune.
   - `Type` is a character that tells the function the type of adjacency matrix (either household or bubble) we wish to prune.
   - `A` is a $1 \times |M|$ array containing the age group of each individual.
   - `RelTrans` is a 1-D array that contains the relative transmissibility (scaling the rate an individual transmits infection) of each age group (i.e. it is T from the manuscript).
   - `RelInf` is a 1-D array that contains the relative susceptibility to infection (scaling the rate of tranmsmission to an individual), of each each group (i.e. it is C from the manuscript).

   Outputs

   - `NewM` is a pruned adjacency matrix.
   - `SAR` is the ratio of connections present after and before pruning. We use this to tune τ_H so that we have a 20 percent secondary attack rate, i.e. 20% of contacts remain after pruning a household matrix.

6. **RewirePrunedMatrix** - the function takes a pruned adjacency matrix `M`, and rewires a proportion of edges `p`. Doing so, we preserve the number of infectious contacts each individual makes, but have changed some of the population's network properties by removing clustering.

   `[Mnew] = RewirePrunedMatrix(M, p, Cnum)`

   Inputs

   - `M` is a pruned adjacency matrix (see PruneMatrixFull).
   - `p` is the proportion of edges that we wish to rewire.
   - `Cnum` is a character that tells the function the scenario we want to rewire for. With input 'C2', directed and undirected links are rewired separately, so that the number of undirected links remains constant after rewiring. With input 'C3' all links are treated as directed, and rewired independently.

   Outputs

   - `Mnew` us the new pruned adjacency matrix obtained after rewiring.

7. **ScenarioTypeHouseBubble**- this function creates an adjacency matrix of bubble connections for all households that have a particular attribute. Eligible households are matched up with one another.

   `B = ScenarioTypeHouseBubble(H, TypeorSize, Position)`

   Inputs

   - `H` is an adjacency matrix of household contacts.
   - `TypeorSize` is a logical $1\times$ `NumHouse` array. It contains information around whether a household has a particular attribute, such as whether it is a certain 'type' (e.g. a household with children) or whether it is a certain size (e.g. a single occupancy household).
   - `Position` is a $1\times$ `NumHouse` array containing the position in the population of the first member of each household.

   Outputs

   - `B` is an adjacency matrix of bubble contacts.

8. **ScenarioTypeHouseBubble2** - this function creates an adjacency matrix for households that have a particular attribute. Using this function, a proportion `d` of eligible households are matched up with any other household.

   `[B,Inbubble] = ScenarioTypeHouseBubble2(H, TypeorSize, Position, Inbubble, d)`

   Inputs

   - `H` is an adjacency matrix of household contacts.
   - `TypeorSize` is a logical $1\times$ `NumHouse` array. It contains information around whether a household has a particular attribute, such as whether it is a certain 'type' (e.g. a household with children) or whether it is a certain size (e.g. a single occupancy household).
   - `Position` is a $1\times$ `NumHouse` array containing the position in the population of the first member of each household.
   - `Inbubble` is a $1 \times$ `NumHouse` array containing whether that household is already part of a bubble (denoted by 1) or not (denoted by 0).
   - `d` is the proportion of eligible households that we want to enter into a bubble.

   Outputs

   - `B` is an adjacency matrix of bubble contacts.
   - `Inbubble` is an updated $1 \times$ `NumHouse` array containing whether that household is part of a bubble or not.