

Vignette

This document contains a brief vignette explaining how to use the functions in github.com/tsleng93/SocialBubble, 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. to obtain your own results surrounding bubble strategies.

Roughly, results can be obtained in five steps:

1. Create or load synthetic population of households.

This can be done by either loading a workspace containing a synthetic population (use ‘FullCensusHouseholdWorkspace.mat’ for the synthetic population from the manuscript, or use ‘CT1088workspace.mat’ for a synthetic population derived from the CT1088 census data set), or can be built using the function `HouseholdMakerAge`.

Relevant functions: `HouseholdMakerAge`

2. Create social bubbles for target households.

This can be done using a variety of available functions, which can be used to form bubbles consisting of two eligible households. Using these functions, a user could create custom bubbling scenarios (e.g. looking at the impact of allowing households of size 2 to form bubbles). The adjacency matrix of all households entering into a bubble consisting of N households is already an output of the function `HouseholdMakerAge`.

Relevant functions: `HouseholdMakerAge`, `ScenarioTypeHouseBubble`, `ScenarioTypeHouseBubble2`, `NonAdherenceBubble`

3. Sample probability matrix

Before simulating the epidemic, we must sample the probability matrix to obtain only the links between individuals that will lead to infection.

Relevant functions: `PruneMatrixFull`

4. Obtain counterfactual scenarios

To compare the effectiveness of social bubbles, results can be compared to counterfactual scenarios *C2* and *C3*, where individuals make the same number of infectious contacts, only these are now unclustered.

Relevant functions: `RewirePrunedMatrix`

5. Simulate Epidemics

Finally, using the sample probability matrix from step 3, and if desired the counterfactual sampled probability matrix from step 4, we can simulate the epidemic.

Relevant functions: `InfectionProcessIndividual`, `InfectionProcessIndividual2`

Example

Suppose we are interested in the impact allowing all households with 3 or less individuals to form bubbles. In this example, we consider the virus to be like SARS-CoV-2, and we use the parameters from the manuscript.

1. We begin by loading the synthetic population from the workspace:

```
load('FullCensusHouseholdWorkspace.mat');
```

Here, the sparse matrix *H* contains the adjacency matrix of the synthetic household population. We will be using the vectors *C*, *Age*, *SizeHouse*, *Position*, which are exemplified in the functions document.

2. To create our custom bubble scenario, we have to create a logical vector telling us whether a household is eligible to form a bubble - in our example a household is eligible if it contains 3 or less individuals. The size of each house is contained in the vector *SizeHouse*, and so we can create an eligibility vector, *Eligible*, using the following:

```
Eligible = SizeHouse <= 3;
```

From this, we can use the function `ScenarioTypeHouseBubble`, to create the social bubbles.

```
Bcustom = ScenarioTypeHouseBubble(H, Eligible, Position);
```

- Household and bubble matrices must be sampled separately as there are different baseline household and bubble transmission rates. We use the default transmission parameters from the manuscript, i.e. τ_H satisfies that the household secondary attack rate is 20% and that $\tau_B = 0.5\tau_H$. Here we must also specify that children are half as susceptible as adults, and that transmissibility is the same for all age groups. In the model, age groups are delineated into 9 ten-year age bands (1 signifying 0 to 9, 2 signifying 10 to 19 and so on). So, we specify:

```
tauH = 0.345; tauB = 0.5*tauH; RelInf = [0.5 0.5 1 1 1 1 1 1 1]; RelTrans = ones(1,9);
```

With these parameters set, we can now sample the underlying probability matrices for household and bubble contacts:

```
[NewH, SAR] = PruneMatrixFull(H, tauH, 'H', Age, RelTrans, RelInf);
```

Doing so we find that $SAR \approx 0.2$, i.e. the household secondary attack rate is 20%

```
NewB = PruneMatrixFull(Bcustom, tauB, 'B', Age, RelTrans, RelInf);
```

- To compare the effectiveness of social bubbles in this instance, we will compare our results to individuals in eligible households making the same number of infectious contacts, but rather than contacts being in bubbles, contacts are random and change each generation, i.e. counterfactual scenario *C3*:

```
NewBc3 = RewirePrunedMatrix(NewB, 1, 'C3');
```

- Now we are almost in a position to simulate epidemics. First, we have to specify the baseline mean-field rate of transmission, the number of initially infected individuals, and case fatality ratio for each age-group, and a random seed:

```
eps = 1.13; Death_Prop = [0.00161,0.00695,0.0309,0.0844,0.161,0.595,1.93,4.28,7.8]./100;
Infect_0 = 50; randnum = 1;
```

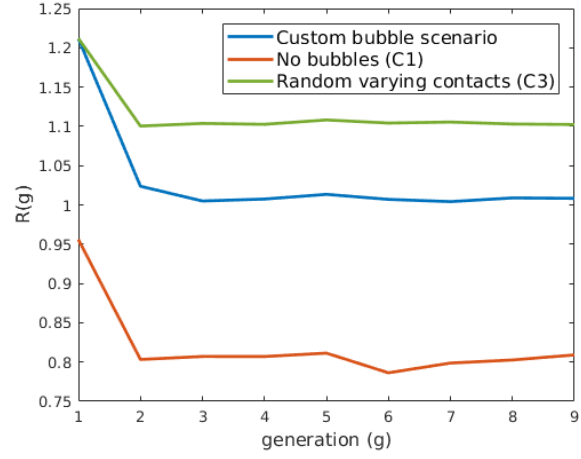
In the paper we adjust initial infecteds to satisfy 1% of the population having been infected by generation 4, but for simplicity here we have set it as 50 for all scenarios. Suppose we are mainly interested in the reproduction number, then the second and third output arguments of `InfectionProcessIndividual` are the relevant outputs. We will also compare these to a situation where no households enter into bubbles, i.e. Scenario *C1*.

```
[~, Rc1, Rgenc1] = InfectionProcessIndividual(NewH, eps, C, Infect_0, Age, RelTrans,
RelInf, Death_Prop, randnum);
```

```
[~, R, Rgen] = InfectionProcessIndividual(NewH + NewB, eps, C, Infect_0, Age, RelTrans,
RelInf, Death_Prop, randnum);
```

```
[~, Rc3, Rgenc3] = InfectionProcessIndividual(NewH + NewBc3, eps, C, Infect_0, Age,
RelTrans, RelInf, Death_Prop, randnum);
```

Here *R* is our *R* estimates for the bubbling scenario, and *Rc1* and *Rc3* are the *R* estimates for counterfactual scenarios *C1* and *C3*. We find that $R \approx 1$, $Rc1 \approx 0.8$, and $Rc3 \approx 1.1$. We can see this visually by plotting the respective *Rgens*



To obtain a more constant estimate over generations, `InfectionProcessIndividual` can be run several times, and results averaged. In the manuscript, results are taken from 1000 epidemics simulated from 10 sampled probability matrices, i.e. `PruneMatrixFull` and `InfectionProcessIndividual` are each run 10 times, and then averaged over.