

OpenABM-Covid19 - an agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing

Supplementary tables:

Supplementary Table 1: Percentage of age-stratified population of the UK and percentage of households containing up to six people, provided by the ONS. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Description	Value, %
Population 0 - 9 years old	12.1
Population 10 - 19 years old	11.3
Population 20 - 29 years old	13.1
Population 30 - 39 years old	13.3
Population 40 - 49 years old	12.8
Population 50 - 59 years old	13.5
Population 60 - 69 years old	10.6
Population 70 - 79 years old	8.3
Population 80+ years old	4.9
Household with one person	27
Household with two people	36
Household with three people	16
Household with four people	15
Household with five people	4
Household with six people	2

Supplementary Table 2: Average number of non-household interactions (random and occupational interactions) for an individual in each age group per day from empirical estimates [1]. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Age group	Number of daily non-household interactions
0 - 19	12
20 - 69	11
70+	6

Supplementary Tables 3: Mean numbers of daily occupational connections for members of each age group, fraction of adults involved in occupational networks for children and for elderly people, and rewiring parameters for randomisation of daily interactions. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Age group	Number of daily occupational interactions
0 - 19	10
20 - 69	7
70+	3

Age group	Ratio of adults to corresponding age groups in their occupational networks
0 - 19	0.2
70+	0.2

Fraction of people in occupational network that an individual interacts with each day	0.5
Rewire parameter on the Watts-Strogatz work networks	0.1

Supplementary Table 4: Parameters for numbers of random connections that members of each age group have per day. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Age group	Mean (SD) of daily random interactions
0 - 19	2 (2)
20 - 69	4 (4)
70+	3 (3)

Supplementary Table 5: Susceptibility by age

Age group	Relative susceptibility
0 - 9	0.35
10 - 19	0.69
20 - 29	1.03
30 - 39	1.03
40 - 49	1.03
50 - 59	1.03
60 - 69	1.27
70 - 79	1.52
80+	1.52

Supplementary Table 6: Description of infection parameters and their values. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Description	Value	Source
Number of infections seeded at simulation start	5	-
Mean of the generation time distribution (days)	5.5	[2–6]
Standard deviation (days) of infectious period	2.14	[2–6]
Mean number of individuals infected by each infectious individual with moderate to severe symptoms	5.18	Derived from calibration to doubling time
Relative infectivity of asymptomatic individuals relative to symptomatic individuals	0.33	Personal communication, [7]
Relative infectivity of mildly symptomatic individuals relative to symptomatic individuals	0.72	Personal communication, [7]
Relative susceptibility to infection, aged 0-9	0.35	[8]
Relative susceptibility to infection, aged 10-19	0.69	[8]
Relative susceptibility to infection, aged 20-29	1.03	[8]
Relative susceptibility to infection, aged 30-39	1.03	[8]
Relative susceptibility to infection, aged 40-49	1.03	[8]
Relative susceptibility to infection, aged 50-59	1.03	[8]
Relative susceptibility to infection, aged 60-69	1.27	[8]
Relative susceptibility to infection, aged 70-79	1.52	[8]
Relative susceptibility to infection, aged 80+	1.52	[8]
Relative infectious rate of household interaction	2	-
Relative infectious rate of occupation interaction	1	-
Relative infectious rate of random interaction	1	-

Supplementary Table 7: Proportion of people in each stage of illness whose disease progresses further [Calibration of Riccardo et al. 2020[9] & Spanish Serology Survey for fraction of asymptomatic and mild symptoms; Calibration of Riccardo et al. 2020[9] & Spanish Serology Survey & Ferguson et al. 2020[10] for fraction hospitalised; Ferguson et al, 2020[10] for fraction of hospitalised that require critical care; Lu et al. 2020[11], Dong et al. 2020[12], Ferguson et al, 2020[10], Yang et al 2020[13] for fatality fraction];[14,15]. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Age group	Fraction of infected people who are asymptomatic	Fraction of infected people with mild symptoms	Fraction hospitalised among people with severely ill	Fraction of hospitalised people that require critical care	Fraction of fatalities among people in critical care	Proportion with critical symptoms that reach critical care
0 - 9	0.456	0.533	0.001	0.05	0.33	1
10 - 19	0.412	0.569	0.006	0.05	0.25	1
20 - 29	0.37	0.597	0.015	0.05	0.5	0.9
30 - 39	0.332	0.614	0.069	0.05	0.5	0.9
40 - 49	0.296	0.616	0.219	0.063	0.5	0.8
50 - 59	0.265	0.602	0.279	0.122	0.69	0.8
60 - 69	0.238	0.571	0.37	0.274	0.65	0.4
70 - 79	0.214	0.523	0.391	0.432	0.88	0.4
80+	0.192	0.461	0.379	0.709	1	0.05

Supplementary Table 8: Mean and standard deviation for density functions of the times that each transition – disease progression or recovery – takes. For the shape of the functions see Supplementary Figure 2. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

Description	Mean (SD) value, days	Source
Time from infection to onset of symptoms	5.42 (2.7)	[16]
Time from symptom onset to hospitalisation	5.14	[17]
Time from hospitalisation to critical care admission	2.27 (2.27)	[*]
Time to recovery if hospitalisation is not required	12 (5)	[13]
Time to death after acquiring critical care	11.74 (8.79)	[*]
Time from infection to recovery for an asymptomatic individual	15 (5)	[13]
Time to recover after hospitalisation	8.75 (8.75)	[*]
Time to recover after requiring critical care	18.8 (12.21)	[*]

* Personal communication with SPI-M; data soon to be published

Supplementary Table 9: Smartphone usage by age in the UK; based on the Technology Tracker (fieldwork 9 Jan – 7 Mar 2020) and the Children’s Media Literacy tracker (fieldwork 25 April – 11 July 2019), data provided by the Office of Communications

Description	0 - 9	10-15	16-24	25-34	35-44	45-54	55-64	65-74	75+	Total
Population by demographic, 000s	8007	4691	7073	9011	8415	9063	8161	6687	5688	66797
Share of demographic with a smartphone, %	9	67	97	97	94	94	78	61	33	72

Supplementary Table 10: Parameters for hospitalisation and self-quarantine upon symptoms. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

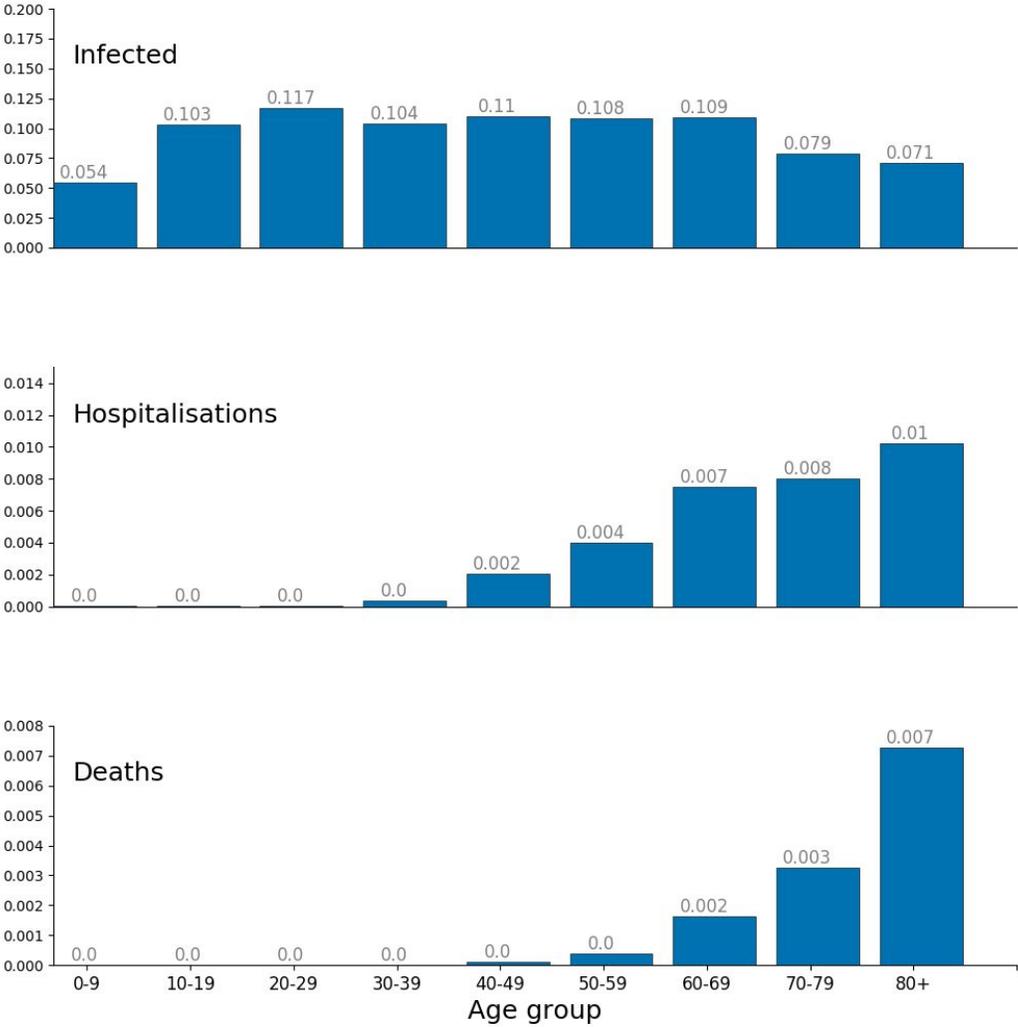
Description	Value
Daily probability of reporting similar symptoms which are not covid-19, including seasonal flu (UK flu survey)	0.002
Maximum number of days quarantine for individuals self-reporting symptoms	7
Daily probability of drop out for an individual quarantining after self-reporting symptoms	0.02
Daily random interactions of a quarantined individual	0
Daily random interactions of a hospitalised individual	0
Proportion of people who self-quarantine upon symptoms	0
Relative change in number of occupation network contacts on lockdown for primary age	0.29
Relative change in number of occupation network contacts on lockdown for secondary age	0.29
Relative change in number of occupation network contacts on lockdown for working age (Based on an estimate of the number of key workers)	0.29
Relative change in number of occupation network contacts on lockdown for retired age	0.29
Relative change in number of occupation network contacts on lockdown for elderly	0.29
Relative change in random network contacts on lockdown	0.29
Relative change in household network contacts on lockdown	1.5

Supplementary Table 11: Parameters corresponding to testing and tracing. Parameter values match the OpenABM-Covid19 baseline parameters, September 15, 2020

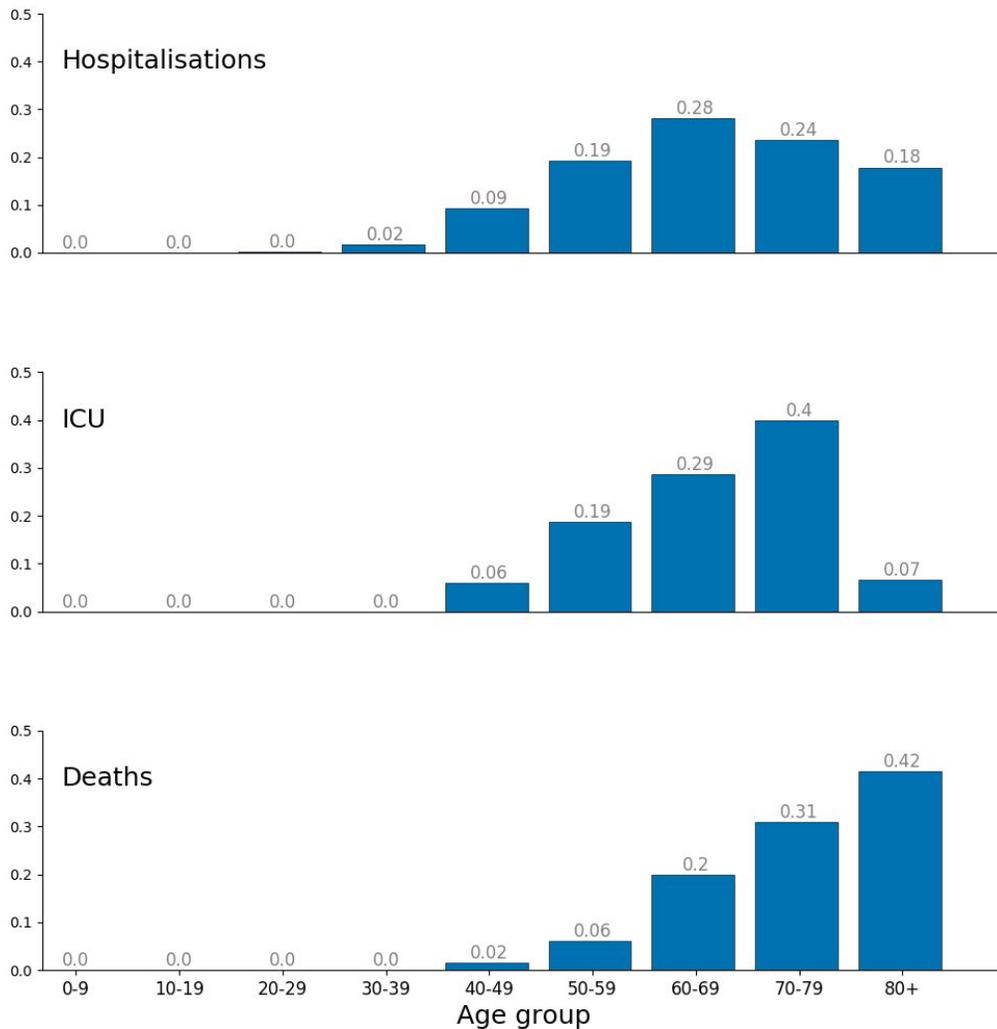
Description	Value
Length of historic interactions traced (days)	10
Maximum number of days quarantine for individuals who are traced after a contact reported symptoms	14
Maximum number of days quarantine for individuals who are traced after a contact tested positive	14
Maximum number of days quarantine for individuals with a positive test result	14
Daily probability of drop out for an individual quarantining after being traced following contact with an individual self-reporting symptoms	0.04
Daily probability of drop out for an individual quarantining after being traced following contact with an individual who tested positive	0.03
Daily probability of drop out for an individual quarantining after a positive test result	0.01
Fraction of individuals who initially comply with a quarantine notification after their contact reported symptoms	0.5
Fraction of individuals who initially comply with a quarantine notification after their contact tested positive	0.9
Test individuals who show symptoms (0=no, 1=yes)	0
Test individuals who have been contact traced (0=no, 1=yes)	0
Release individuals following a negative test (0=no, 1=yes)	1
Trace contacts of individuals who show symptoms (0=no, 1=yes)	0
Trace contacts of an individual who tests positive (0=no, 1=yes)	0
Repeat contact tracing be carried out on a positive test if already traced on symptoms (0=no, 1=yes)	0
Quarantine individuals who are traced (0=no, 1=yes)	0
Fraction of interactions that are captured if both users have the app	0.8
Depth of interaction network to contact	0
Commence contact tracing on a hospital clinical diagnosis	1
Quarantine household members of a person with a positive test (0=no, 1=yes)	0
Quarantine household members of a person with symptoms (0=no, 1=yes)	0
Quarantine household members of a person who has been traced following contact with an individual who has tested positive (0=no, 1=yes)	0
Quarantine household members of a person who has been traced following contact with an individual who has self-reported symptoms (0=no, 1=yes)	0
Quarantine the contacts of each household member of a person who tests positive (0=no, 1=yes)	0
Quarantine the contacts of other household members when someone gets symptoms	0
The number of previous days' contacts to be traced and contacted	7
Release a chain of quarantined people if after this number of days nobody has shown symptoms on the chain	0
Number of days following infection the test is insensitive	3

Number of days following infection to end of period where test is sensitive	14
Sensitivity of test in the time where it is sensitive	0.8
Specificity of test (at any time)	0.999
Minimum number of days to wait to take a test	1
Number of days to wait for a test result	1
Maximum fraction of the population with smartphones aged 0-9	0.09
Maximum fraction of the population with smartphones aged 10-19	0.8
Maximum fraction of the population with smartphones aged 20-29	0.97
Maximum fraction of the population with smartphones aged 30-39	0.96
Maximum fraction of the population with smartphones aged 40-49	0.94
Maximum fraction of the population with smartphones aged 50-59	0.86
Maximum fraction of the population with smartphones aged 60-69	0.7
Maximum fraction of the population with smartphones aged 70-79	0.48
Maximum fraction of the population with smartphones aged 80+	0.32

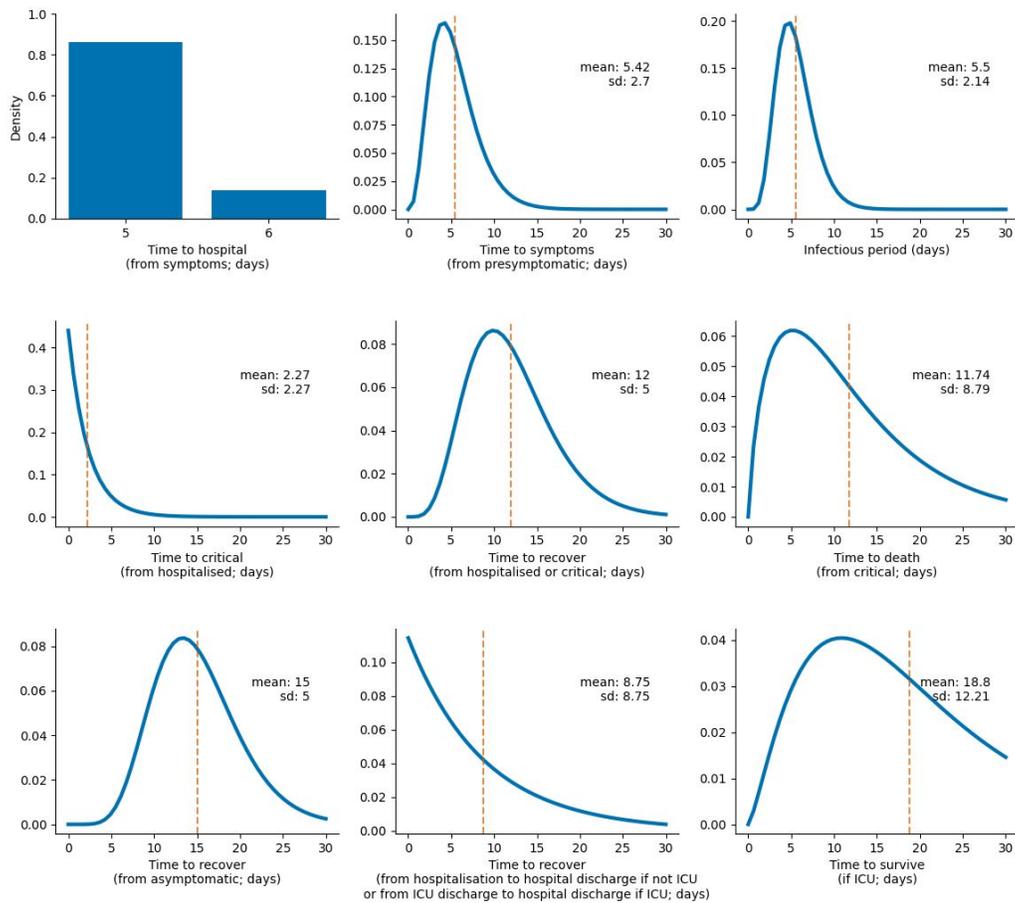
Supplementary figures



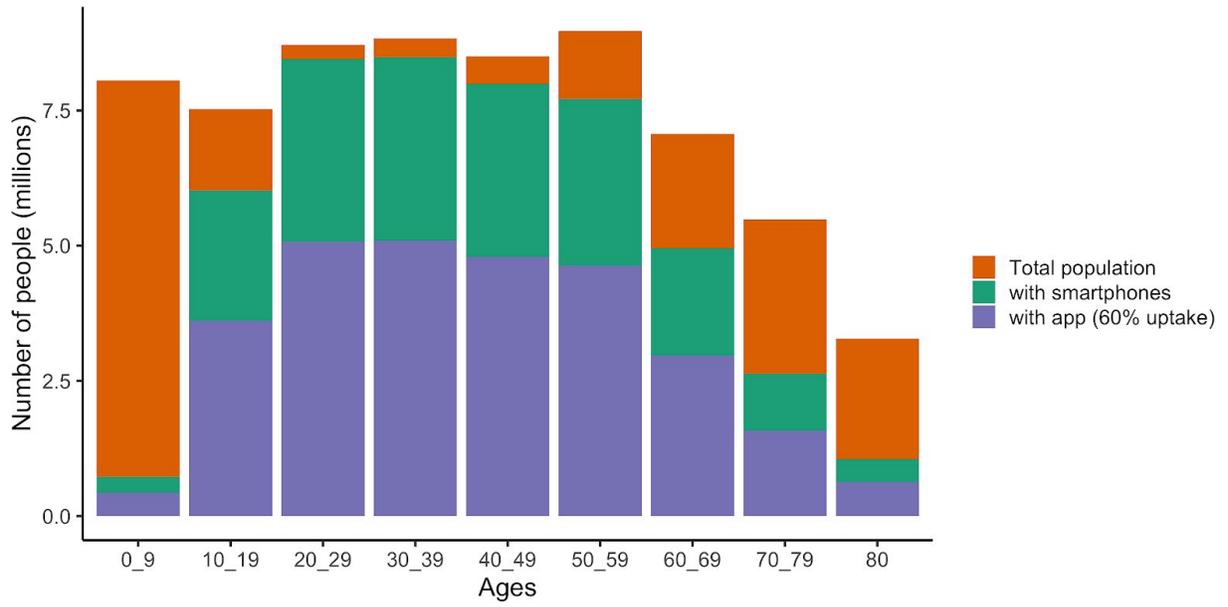
Supplementary Figure 1a: Proportion of each age group ever infected, hospitalised, or died at the end of a single simulation in a population of 1 million individuals with UK-like demographics and control interventions. The simulation was run for 70 days after lockdown started. The denominator in each calculation is the number of individuals in each age group in the total population.



Supplementary Figure 1b: Age distribution of hospital admissions, ICU admissions, and deaths at the end of a single simulation of 1 million individuals with UK-like demographics and control interventions. The simulation was run for 70 days after lockdown started. The denominator is the number of individuals ever having been in the state in question.



Supplementary Figure 2: Waiting time distributions for transitions between infection and disease states. All distributions are gamma except time to hospital which is a shifted Bernoulli distribution. Mean of each gamma distribution is shown with a vertical dashed line.



Supplementary Figure 3: Smartphone usage by age in the UK

OpenABM-Covid19 - Example Self-Isolation on Symptoms

This examples show how to run a simulation on 50k people and demonstrates the effect in the reduction in incidence which occurs when people and their household self-isolate on symptoms.

First we set the get the baseline parameters, change the population to 50k and then initialise the simulation.

```
In [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )

sim = utils.get_simulation( params )
model = sim.env.model
```

Now we run for 40 days.

```
In [2]: sim.steps( 40 )
timeseries = pd.DataFrame( sim.results )
```

We now run a new simulation from scratch for 20 days (same seed an initial parameters).

```
In [3]: del( model ); del( sim );
params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
sim = utils.get_simulation( params )
model = sim.env.model

sim.steps( 20 )
```

At day 20 we introduce self-quarantine of 80% those who are symptomatic and quarantine all members of their household.

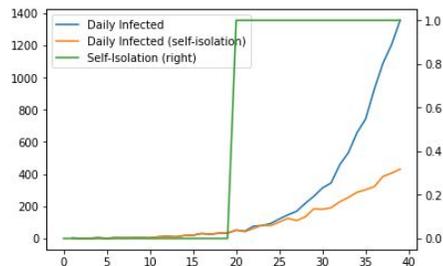
```
In [4]: model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("quarantine_household_on_symptoms", 1)
sim.steps( 20 )
```

Plot the number of daily new infections over the course of the simulation and compare between with and without self-isolation.

```
In [5]: timeseries2 = pd.DataFrame( sim.results )
timeseries["daily_infected"] = timeseries["total_infected"].diff(1)
timeseries2["daily_infected_sd"] = timeseries2["total_infected"].diff(1)
df = pd.merge( timeseries, timeseries2, on = "time" )
df["self_isolate"] = timeseries.apply(lambda X: 1 if ( X.time >20 ) & ( X.time <= 40 ) else 0, axis=1)

plt.figure()
df.daily_infected.plot( x = "time", legend = True, label = "Daily Infected" )
df.daily_infected_sd.plot( x = "time", legend = True, label = "Daily Infected (self-isolation)" )
df.self_isolate.plot( x = "time", legend = True, secondary_y = True, label = "Self-Isolation" )
```

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1289b7b10>



Supplementary Figure 4a: Example notebook of self-isolation on symptoms

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_self_isolation.ipynb)

OpenABM-Covid19 - Example Lockdown

This examples show how to run a simulation on 50k people and turn on a lockdown after 20 days.

First we set the get the baseline parameters, change the population to 50k and then initialise the simulation.

```
In [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )

sim = utils.get_simulation( params )
model = sim.env.model
```

Now we run for 30 days.

```
In [2]: sim.steps( 30 )
```

Now we turn on a lockdown where the number of interactions on the occupation networks and random network are cut to 10% of their pre-lockdown level and the strength of household transmissions is 1.5 times the pre-lockdown level. And run for another 10 days.

```
In [3]: model.update_running_params("lockdown_occupation_multiplier_primary_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_secondary_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_working_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_retired_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_elderly_network", 0.1)
model.update_running_params("lockdown_random_network_multiplier", 0.1)
model.update_running_params("lockdown_house_interaction_multiplier", 1.5)
model.update_running_params("lockdown_on", 1)
sim.steps( 10 )
```

Turn the lockdown off and run for another 10 days.

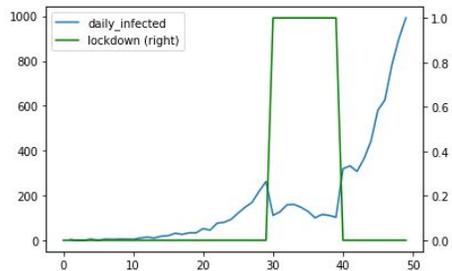
```
In [4]: model.update_running_params("lockdown_on", 0)
sim.steps( 10 )
```

Plot the number of daily new infections over the course of the simulation.

```
In [6]: timeseries = pd.DataFrame( sim.results )
timeseries["daily_infected"] = timeseries["total_infected"].diff(1)

plt.figure()
timeseries.daily_infected.plot( x = "time", legend = True )
timeseries.lockdown.plot( x = "time", legend = True, secondary_y = True, style = 'g' )
```

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x11f3aead0>



Supplementary Figure 4b: Example notebook of a lockdown.

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_lockdown.ipynb)

OpenABM-Covid19 - Example Lockdown then Shield

This examples show how to run a simulation on 50k people and turn on a lockdown after 20 days.

First we set the get the baseline parameters, change the population to 50k and then initialise the simulation.

```
In [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )

sim = utils.get_simulation( params )
model = sim.env.model
```

Now we run for 30 days.

```
In [2]: sim.steps( 30 )
```

Now we turn on a lockdown where the number of interactions on the occupation networks and random network are cut to 10% of their pre-lockdown level and the strength of household transmissions is 1.5 times the pre-lockdown level. And run for another 10 days.

```
In [3]: model.update_running_params("lockdown_occupation_multiplier_primary_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_secondary_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_working_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_retired_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_elderly_network", 0.1)
model.update_running_params("lockdown_random_network_multiplier", 0.1)
model.update_running_params("lockdown_house_interaction_multiplier", 1.5)
model.update_running_params("lockdown_on", 1)
sim.steps( 10 )
```

Children and adults occupational network interactions return to 80% of their prior interactions level, random interactions go to 40% of their prior levels and the elderly remain shielded.

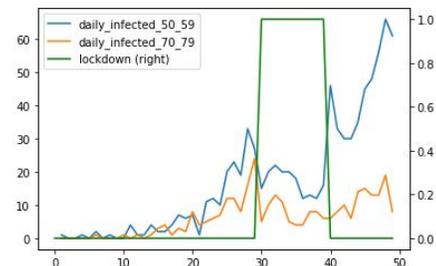
```
In [4]: model.update_running_params("lockdown_occupation_multiplier_primary_network", 0.8)
model.update_running_params("lockdown_occupation_multiplier_secondary_network", 0.8)
model.update_running_params("lockdown_occupation_multiplier_working_network", 0.8)
model.update_running_params("lockdown_random_network_multiplier", 0.4)
model.update_running_params("lockdown_house_interaction_multiplier", 1)
sim.steps( 10 )
```

Plot the number of daily new infections over the course of the simulation for the 50y-59y age group and the 70y-79y age group. Note infections rise very slowly in the 70y-79y age group who are being shielded.

```
In [5]: timeseries = pd.DataFrame( sim.results )
timeseries["daily_infected_50_59"] = timeseries["total_infected_50_59"].diff(1)
timeseries["daily_infected_70_79"] = timeseries["total_infected_70_79"].diff(1)
timeseries["lockdown"] = timeseries.apply(lambda X: 1 if ( X.time >30 ) & ( X.time <= 40 ) else 0, axis=1)

plt.figure()
timeseries.daily_infected_50_59.plot( x = "time", legend = True )
timeseries.daily_infected_70_79.plot( x = "time", legend = True )
timeseries.lockdown.plot( x = "time", legend = True, secondary_y = True, style = 'g' )
```

```
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x117634c90>
```



Supplementary Figure 4c: Example of lockdown followed by shielding.

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_lockdown_shield.ipynb)

OpenABM-Covid19 - Example Lockdown then Social Distance

This examples show how to run a simulation on 50k people and turn on a lockdown after 20 days.

First we set the get the baseline parameters, change the population to 50k and then initialise the simulation.

```
In [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )

sim = utils.get_simulation( params )
model = sim.env.model
```

Now we run for 30 days.

```
In [2]: sim.steps( 30 )
```

Now we turn on a lockdown where the number of interactions on the occupation networks and random network are cut to 10% of their pre-lockdown level and the strength of household transmissions is 1.5 times the pre-lockdown level. And run for another 10 days.

```
In [3]: model.update_running_params("lockdown_occupation_multiplier_primary_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_secondary_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_working_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_retired_network", 0.1)
model.update_running_params("lockdown_occupation_multiplier_elderly_network", 0.1)
model.update_running_params("lockdown_random_network_multiplier", 0.1)
model.update_running_params("lockdown_house_interaction_multiplier", 1.5)
model.update_running_params("lockdown_on", 1)
sim.steps( 10 )
```

Lockdown is ended, and replaced by social distances measures (e.g. such as wearing masks). These are modelled by a 70% reduction in the rate of transmission in occupation and random interactions (but no change to household transmission rates).

```
In [4]: model.update_running_params("lockdown_on", 0)
model.update_running_params("relative_transmission_occupation", 0.3 )
model.update_running_params("relative_transmission_random", 0.3)
sim.steps( 10 )
```

The social distancing measures are then removed and transmission rates return their pre-lockdown levels.

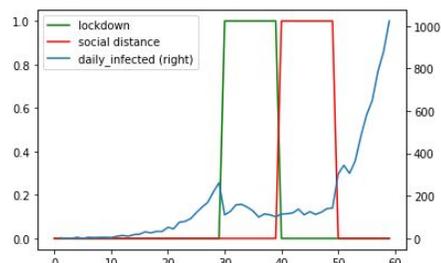
```
In [5]: model.update_running_params("lockdown_on", 0)
model.update_running_params("relative_transmission_occupation", 1 )
model.update_running_params("relative_transmission_random", 1)
sim.steps( 10 )
```

Plot the number of daily new infections over the course of the simulation. Note the increase in infections post-lockdown is much less than that without social distancing measures.

```
In [6]: timeseries = pd.DataFrame( sim.results )
timeseries["daily_infected"] = timeseries["total_infected"].diff(1)
timeseries['social_distance'] = timeseries.apply(lambda X: 1 if ( X.time >40 ) & ( X.time <= 50 ) else 0, axis=1)

plt.figure
timeseries.lockdown.plot( x = "time", kind = "line", label = "lockdown", legend = True, style = 'g' )
timeseries.social_distance.plot( x = "time", kind = "line", label = "social distance", legend = True , style = 'r' )
timeseries.daily_infected.plot( x = "time", legend = True, secondary_y = True )
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x11c32b190>
```



Supplementary Figure 4d: Example lockdown followed by social distancing.

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_lockdown_social_distance.ipynb)

OpenABM-Covid19 - Example Self-Isolation After Testing

This examples demonstrates the importance of quick testing if self-isolation only occurs following a positive test.

First we set the baseline parameters: change the population to 50k (`n_total=50000`); turn off self-quarantining on symptoms (`quarantine_length_self=0`); set the length and daily drop-out rate of quarantining following a positive test (`quarantine_length_positive=14` `quarantine_dropout_positive=0.02`); and set the test sensitivity to 90% (`test_sensitivity=0.9`). At the start of the simulation there is no self isolation and we run the simulation for 15 days.

```
n [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
params.set_param( "quarantine_length_self", 0 )
params.set_param( "quarantine_length_positive", 14 )
params.set_param( "quarantine_dropout_positive", 0.02 )
params.set_param( "test_sensitivity", 0.9 )
sim = utils.get_simulation( params )
model = sim.env.model
sim.steps( 15 )
```

After 15 days we turn on interventions for those with symptoms. Setting `self_quarantine_fraction=0.8` means that 80% of the those with symptom enter the intervention cascade (by default this would mean self-isolation, but setting `quarantine_length_self=0` means there is no initial self-isolation test is ordered (`test_on_symptoms=1`) immediately (`test_order_wait=0`) and a result comes back after 24hrs (`test_result_wait=1`). If the res is positive then the individual and their household quarantines and the simulation is run for 25 days.

```
n [2]: model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("test_on_symptoms", 1)
model.update_running_params( "test_order_wait", 0 )
model.update_running_params( "test_result_wait", 1 )
model.update_running_params("quarantine_household_on_positive", 1)
sim.steps( 25 )
timeseries = pd.DataFrame( sim.results )
```

We now re-run the simulation but with slower testing, including a 1 day wait for the test to be done and a 3 day wait for the test result.

```
n [3]: del( model ); del( sim );
params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
params.set_param( "quarantine_length_self", 0 )
params.set_param( "quarantine_length_positive", 14 )
params.set_param( "quarantine_dropout_positive", 0.02 )
params.set_param( "test_sensitivity", 0.9 )
sim = utils.get_simulation( params )
model = sim.env.model
sim.steps( 15 )

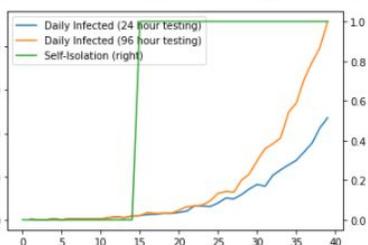
model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("test_on_symptoms", 1)
model.update_running_params( "test_order_wait", 1 )
model.update_running_params( "test_result_wait", 3 )
model.update_running_params("quarantine_household_on_positive", 1)
sim.steps( 25 )
```

Plot the number of daily new infections over the course of the simulation and compare between quick and slow testing.

```
n [6]: timeseries2 = pd.DataFrame( sim.results )
timeseries["daily_infected"] = timeseries["total_infected"].diff(1)
timeseries2["daily_infected_slow"] = timeseries2["total_infected"].diff(1)
df = pd.merge( timeseries, timeseries2, on = "time" )
df["self_isolate"] = timeseries.apply(lambda X: 1 if ( X.time >15 ) & ( X.time <= 40 ) else 0, axis=1)

plt.figure()
df.daily_infected.plot( x = "time", legend = True, label = "Daily Infected (24 hour testing)" )
df.daily_infected_slow.plot( x = "time", legend = True, label = "Daily Infected (96 hour testing)" )
df.self_isolate.plot( x = "time", legend = "True", secondary_y = True, label = "Self-Isolation" )
```

```
wt[6]: <matplotlib.axes._subplots.AxesSubplot at 0x119d42fd0>
```



Supplementary Figure 4e: Example of self-isolation after testing.

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_testing.ipynb)

OpenABM-Covid19 - Example Digital Contact-Tracing

This examples demonstrates the effect of digital contact-tracing following testing. Digital contract tracing is turned on following the end of a lockdown period and moderate social-distancing measures are added (20% reduction in transmission for occupation and random interactions).

First we set the baseline parameters: change the population to 50k (`n_total=50000`); the test sensitivity to 90% (`test_sensitivity=0.9`); and the fraction of app users to be 60% in the 10y-69y age groups (`app_users_fraction_10_19=0.6` etc.) and 30% in the over 70s. At the start of the simulation there is no self-isolation and we run the simulation for 15 days.

```
In [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
params.set_param( "test_sensitivity", 0.9 )
params.set_param( "app_users_fraction_0_9", 0 )
params.set_param( "app_users_fraction_10_19", 0.6 )
params.set_param( "app_users_fraction_20_29", 0.6 )
params.set_param( "app_users_fraction_30_39", 0.6 )
params.set_param( "app_users_fraction_40_49", 0.6 )
params.set_param( "app_users_fraction_50_59", 0.6 )
params.set_param( "app_users_fraction_60_69", 0.6 )
params.set_param( "app_users_fraction_70_79", 0.3 )
params.set_param( "app_users_fraction_80", 0.3 )
sim = utils.get_simulation( params )
model = sim.env.model
sim.steps(30 )
```

After 30 days we implement a general lockdown (default parameters) and also introducing self-isolation of (80%) of individuals showing symptoms and their households.

```
In [2]: model.update_running_params("lockdown_on", 1)
model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("quarantine_household_on_symptoms", 1)
sim.steps( 30 )
```

Following the lockdown we introduce digital contact tracing (`app_turned_on=1`). We use the set up that symptomatic individuals who are self-isolating get tests (`test_on_symptoms=1`), tracing occurs following a positive test (`trace_on_positive=1`) with those traced being asked to quarantine (`quarantine_on_traced=1`) and their household members also being asked to quarantined (`quarantine_household_on_traced_positive=1`). Social-distancing measures reduce the transmission rate by 20% in occupation and random interactions (`relative_transmission_occupation=0.8` etc.). We then the the simulation for another 50 days.

```
In [3]: model.update_running_params("lockdown_on", 0)
model.update_running_params("app_turned_on", 1 )
model.update_running_params("test_on_symptoms", 1)
model.update_running_params("test_order_wait", 0 )
model.update_running_params("test_result_wait", 1 )
model.update_running_params("trace_on_positive", 1)
model.update_running_params("quarantine_on_traced", 1)
model.update_running_params("quarantine_household_on_traced_positive", 1 )
model.update_running_params("relative_transmission_occupation", 0.8 )
model.update_running_params("relative_transmission_random", 0.8)
sim.steps( 50 )
timeseries = pd.DataFrame( sim.results )
del( model ); del( sim );
```

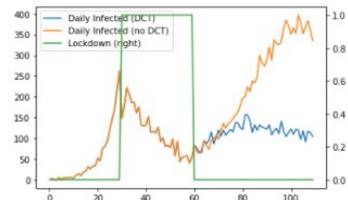
We now re-run the simulation with all the same settings except without turning on the app for comparison.

```
In [4]: params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
params.set_param( "test_sensitivity", 0.9 )
sim = utils.get_simulation( params )
model = sim.env.model
sim.steps(30 )
model.update_running_params("lockdown_on", 1)
model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("quarantine_household_on_symptoms", 1)
sim.steps( 30 )
model.update_running_params("lockdown_on", 0)
model.update_running_params("relative_transmission_occupation", 0.8 )
model.update_running_params("relative_transmission_random", 0.8)
sim.steps( 50 )
timeseries2 = pd.DataFrame( sim.results )
del( model ); del( sim );
```

Plot the number of daily new infections over the course of the simulation and compare the effect of adding digital contract tracing post-lockdown.

```
In [5]: timeseries["daily_infected"] = timeseries["total_infected"].diff(1)
timeseries["daily_infected_base"] = timeseries2["total_infected"].diff(1)
plt.figure()
timeseries.daily_infected.plot( x = "time", legend = True, label = "Daily Infected (DCT)" )
timeseries.daily_infected_base.plot( x = "time", legend = True, label = "Daily Infected (no DCT)" )
timeseries.lockdown.plot( x = "time", legend = "True", secondary_y = True, label = "Lockdown" )
```

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x10b430490>



Supplementary Figure 4f: Example digital contact tracing.

https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_digital_contact_tracing.ipynb

OpenABM-Covid19 - Example Manual Contact-Tracing

This examples demonstrates the effect of manual contact-tracing following testing. A well-staffed manual contract tracing service is turned on following the end of a lockdown period and moderate social-distancing measures are added (20% reduction in transmission for occupation and random interactions).

First we set the baseline parameters: change the population to 50K (`n_total=50000`); the test sensitivity to 90% (`test_sensitivity=0.9`). At the start of the simulation there is no self-isolation and we run the simulation for 15 days.

```
In [21]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt

params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
params.set_param( "test_sensitivity", 0.9 )
sim = utils.get_simulation( params )
model = sim.env.model
sim.steps(30 )
```

After 30 days we implement a general lockdown (default parameters) and also introducing self-isolation of (80%) of individuals showing symptoms and their households.

```
In [22]: model.update_running_params("lockdown_on", 1)
model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("quarantine_household_on_symptoms", 1)
sim.steps( 50 )
```

Following the lockdown, we introduce manual contact-tracing (`manual_trace_on=1`) which is staffed by 50 workers (`manual_trace_n_workers=1`) and we assume when tracing 80% of occupational (`manual_traceable_fraction_occupation=0.8`) and 5% of random contacts (`manual_traceable_random_occupation=0.05`) are traced a day after receipt of a positive test (`manual_trace_delay=1`). We use the set up that symptomatic individuals who are self-isolating get tests (`test_on_symptoms=1`), tracing occurs following a positive test (`manual_trace_on_positive=1`) with those traced being asked to quarantine (`quarantine_on_traced=1`) and their household members also being asked to quarantined (`quarantine_household_on_traced_positive=1`). Social-distancing measures reduce the transmission rate by 20% in occupation and random interactions (`relative_transmission_occupation=0.8` etc). We then the the simulation for another 50 days.

```
In [23]: model.update_running_params("lockdown_on", 0)
model.update_running_params("manual_trace_on", 1 )
model.update_running_params("manual_trace_n_workers", 50 )
model.update_running_params("manual_trace_on_positive", 1 )
model.update_running_params("manual_trace_delay", 1 )
model.update_running_params("manual_traceable_fraction_occupation", 0.8 )
model.update_running_params("manual_traceable_fraction_random", 0.05 )
model.update_running_params("test_on_symptoms", 1)
model.update_running_params("test_order_wait", 0 )
model.update_running_params("test_result_wait", 1 )
model.update_running_params("quarantine_on_traced", 1)
model.update_running_params("quarantine_household_on_traced_positive", 1 )
model.update_running_params("relative_transmission_occupation", 0.8 )
model.update_running_params("relative_transmission_random", 0.8 )
sim.steps( 50 )
timeseries = pd.DataFrame( sim.results )
del( model ); del( sim );
```

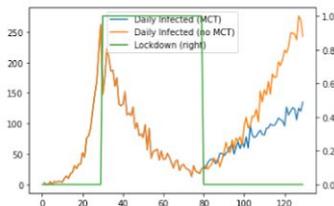
We now re-run the simulation with all the same settings except without turning on the app for comparison.

```
In [24]: params = utils.get_baseline_parameters()
params.set_param( "n_total", 50000 )
params.set_param( "test_sensitivity", 0.9 )
sim = utils.get_simulation( params )
model = sim.env.model
sim.steps(30 )
model.update_running_params("lockdown_on", 1)
model.update_running_params("self_quarantine_fraction", 0.8 )
model.update_running_params("quarantine_household_on_symptoms", 1)
sim.steps( 50 )
model.update_running_params("lockdown_on", 0)
model.update_running_params("relative_transmission_occupation", 0.8 )
model.update_running_params("relative_transmission_random", 0.8 )
sim.steps( 50 )
timeseries2 = pd.DataFrame( sim.results )
del( model ); del( sim );
```

Plot the number of daily new infections over the course of the simulation and compare the effect of adding manual contract-tracing post-lockdown.

```
In [26]: timeseries["daily_infected"] = timeseries["total_infected"].diff(1)
timeseries["daily_infected_base"] = timeseries2["total_infected"].diff(1)
plt.figure()
timeseries.daily_infected.plot( x = "time", legend = True, label = "Daily Infected (MCT)" )
timeseries.daily_infected_base.plot( x = "time", legend = True, label = "Daily Infected (no MCT)" )
timeseries.lockdown.plot( x = "time", legend = "True", secondary_y = True, label = "Lockdown" )
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x123842f90>



Supplementary Figure 4g: Example manual contact tracing.

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_manual_contact_tracing.ipynb)

OpenABM-Covid19 - Example Digital Contact-Tracing Protection Calculation

This examples demonstrates the effect of digital contact-tracing following testing (same as Example Digital Contact-Tracing) and shows the protection gained by users of the system. Digital contact tracing is turned on following the end of a lockdown period and moderate social-distancing measures are added (20% reduction in transmission for occupation and random interactions).

First we set the baseline parameters: change the population to 100K (`n_total=100000`); the test sensitivity to 90% (`test_sensitivity=0.9`); and the fraction of app users to be 60% in the 10y-69y age groups (`app_users_fraction_10_19=0.6` etc.) and 30% in the over 70s. At the start of the simulation there is no self-isolation and we run the simulation for 15 days.

```
In [1]: import example_utils as utils
import pandas as pd
import matplotlib.pyplot as plt
import sys
```

```
params = utils.get_baseline_parameters()
params.set_param("n_total", 100000)
params.set_param("test_sensitivity", 0.9)
params.set_param("app_users_fraction_0_9", 0)
params.set_param("app_users_fraction_10_19", 0.6)
params.set_param("app_users_fraction_20_29", 0.6)
params.set_param("app_users_fraction_30_39", 0.6)
params.set_param("app_users_fraction_40_49", 0.6)
params.set_param("app_users_fraction_50_59", 0.6)
params.set_param("app_users_fraction_60_69", 0.6)
params.set_param("app_users_fraction_70_79", 0.3)
params.set_param("app_users_fraction_80+", 0.3)
sim = utils.get_simulation(params)
model = sim.env.model
sim.steps(30)
```

After 30 days we implement a general lockdown (default parameters) and also introducing self-isolation of (80%) of individuals showing symptoms and their households.

```
In [2]: model.update_running_params("lockdown_on", 1)
model.update_running_params("self_quarantine_fraction", 0.8)
model.update_running_params("quarantine_household_on_symptoms", 1)
sim.steps(30)
```

Following the lockdown we introduce digital contact tracing (`app_turned_on=1`). We use the set up that symptomatic individuals who are self-isolating get tests (`test_on_symptoms=1`), tracing occurs following a positive test (`trace_on_positive=1`) with those traced being asked to quarantine (`quarantine_on_traced=1`) and their household members also being asked to quarantine (`quarantine_household_on_traced_positive=1`). Social-distancing measures reduce the transmission rate by 20% in occupation and random interactions (`relative_transmission_occupation=0.8` etc.). We then the the simulation for another 50 days.

```
In [3]: model.update_running_params("lockdown_on", 0)
model.update_running_params("app_turned_on", 1)
model.update_running_params("test_on_symptoms", 1)
model.update_running_params("test_order_wait", 0)
model.update_running_params("test_result_wait", 1)
model.update_running_params("trace_on_positive", 1)
model.update_running_params("quarantine_on_traced", 1)
model.update_running_params("quarantine_household_on_traced_positive", 1)
model.update_running_params("relative_transmission_occupation", 0.8)
model.update_running_params("relative_transmission_random", 0.8)
sim.steps(50)
```

Now we write all the transmissions from the model and all the individuals characteristic's to file and load to dataframes.

```
In [4]: model.write_transmissions()
model.write_individual_file()
df_transmission = pd.read_csv("transmission_Run1.csv", comment="#", sep=",", skipinitialspace=True)
df_indiv = pd.read_csv("individual_file_Run1.csv", comment="#", sep=",", skipinitialspace=True)
```

Finally we look at the number of infections (post DCT starting) by age group and for app users versus non app users and calculate the relative risk of infection. The simulation shows that app users are about 11% less likely to be infected than non-app users.

```
In [6]: df_trans_post = df_transmission[df_transmission["time_infected"] > 60].loc[:, ["ID_recipient"]]
df_trans_post["infected"] = 1
df = pd.merge(df_indiv, df_trans_post, left_on="ID", right_on="ID_recipient", how="left")
df.fillna(0)
df["total"] = 1
df = df.groupby(["age_group", "app_user"]).agg({"infected": "sum", "total": "count"}).reset_index()
df["infected_fraction"] = df["infected"] / df["total"]
df = df.pivot_table(index=["age_group"], columns="app_user", values="infected_fraction").reset_index()
df["protection"] = 1 - df[1] / df[0]
sys.stdout.write("App user protection = %.1f%% % (100 * df["age_group"] > 0)["protection"].mean())
df.loc[:, ["age_group", "protection"]]
```

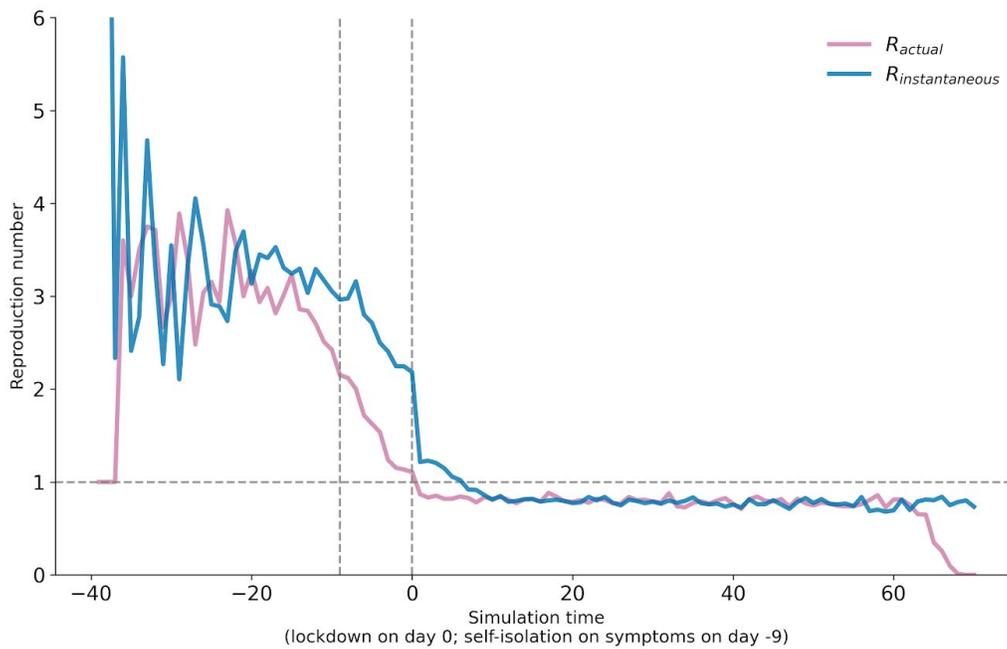
App user protection = 10.8%

Out[6]:

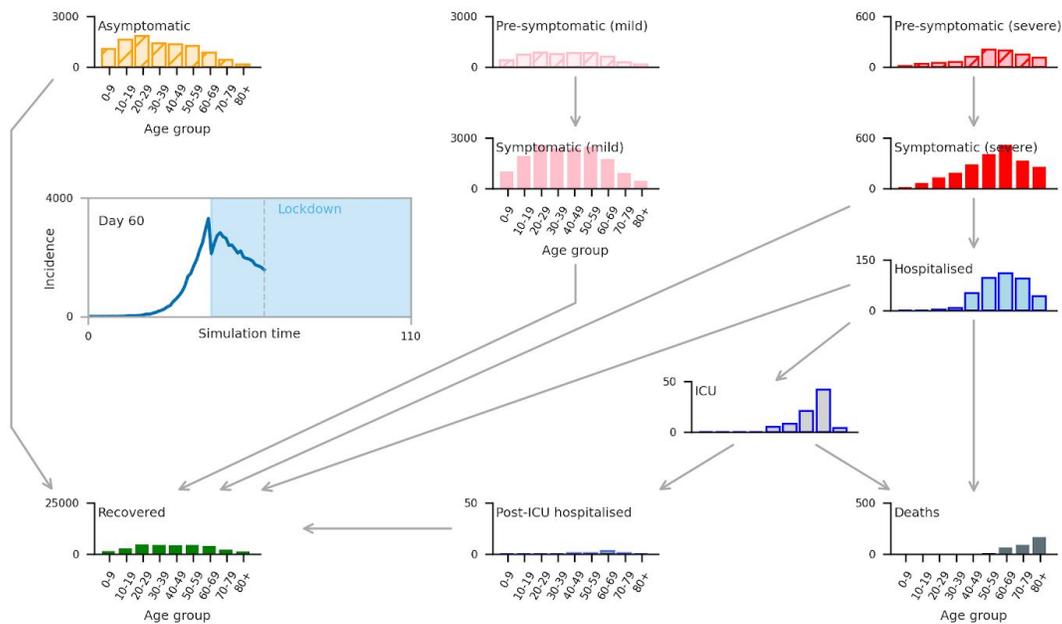
app_user	protection	age_group
0	NaN	0
1	0.124714	1
2	0.141762	2
3	0.086521	3
4	0.165012	4
5	0.046772	5
6	0.088578	6
7	0.029167	7
8	0.184938	8

Supplementary Figure 4h: Example app user protection calculation

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/notebooks/example_digital_contact_tracing_protect.ipynb)



Supplementary Figure 5: Reproduction number in a single simulated outbreak calculated using the complete simulated transmission tree (actual) or using the time series (instantaneous). Simulation data are for a single simulation in a population of 1 million individuals with UK-like demographics. The vertical dashed mark where interventions were introduced (self-isolation on symptoms followed by lockdown), note that R_{actual} is reduced prior to the introduction of each intervention.



Supplementary Figure 6: Animation of a simulated outbreak in a population of 1 million individuals with UK-like demographics and control interventions showing age-stratified histograms of individuals in each compartment.

(https://github.com/BDI-pathogens/OpenABM-Covid19-model-paper/blob/master/figures/fig_outbreak_animation.mp4) Arrangement of sub-panels are in the same arrangement as the model compartments (Figure 5).

References

1. Mossong J, Hens N, Jit M, Beutels P, Auranen K, Mikolajczyk R, et al. Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLoS Med.* 2008;5: e74.
2. He X, Lau EHY, Wu P, Deng X, Wang J, Hao X, et al. Temporal dynamics in viral shedding and transmissibility of COVID-19. *Nat Med.* 2020;26: 672–675.
3. Xia W, Liao J, Li C, Li Y, Qian X, Sun X, et al. Transmission of coronavirus disease 2019 during the incubation period may lead to a quarantine loophole. *medRxiv.* 2020. doi:10.1101/2020.03.06.20031955
4. Cheng H-Y, Jian S-W, Liu D-P, Ng T-C, Huang W-T, Lin H-H. Contact Tracing Assessment of COVID-19 Transmission Dynamics in Taiwan and Risk at Different Exposure Periods Before and After Symptom Onset. *JAMA Intern Med.* 2020. doi:10.1001/jamainternmed.2020.2020
5. Ferretti L, Wymant C, Kendall M, Zhao L, Nurtay A, Abeler-Dörner L, et al. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science.* 2020;368. doi:10.1126/science.abb6936
6. Ferretti L, Ledda A, Wymant C, Zhao L, Ledda V, Dorner LA-, et al. The timing of COVID-19 transmission. *medRxiv.* 2020; 2020.09.04.20188516.
7. Sun et al. (2020) Personal communication.
8. Zhang J, Litvinova M, Liang Y, Wang Y, Wang W, Zhao S, et al. Changes in contact patterns shape the dynamics of the COVID-19 outbreak in China. *Science.* 2020;368: 1481–1486.
9. Riccardo F, Ajelli M, Andrianou X, Bella A, Del Manso M, Fabiani M, et al. Epidemiological characteristics of COVID-19 cases in Italy and estimates of the reproductive numbers one month into the epidemic. *medRxiv.* 2020. Available: <https://www.medrxiv.org/content/10.1101/2020.04.08.20056861v1.abstract>
10. Ferguson NM. Impact of Non-pharmaceutical Interventions (NPIs) to Reduce COVID-19 Mortality and Healthcare Demand. 2020.
11. Lu X, Zhang L, Du H, Zhang J, Li YY, Qu J, et al. SARS-CoV-2 Infection in Children. *N Engl J Med.* 2020;382: 1663–1665.
12. Dong Y, Mo X, Hu Y, Qi X, Jiang F, Jiang Z, et al. Epidemiology of COVID-19 Among Children in China. *Pediatrics.* 2020;145. doi:10.1542/peds.2020-0702
13. Yang X, Yu Y, Xu J, Shu H, Xia J 'an, Liu H, et al. Clinical course and outcomes of critically ill patients with SARS-CoV-2 pneumonia in Wuhan, China: a single-centered, retrospective, observational study. *The Lancet Respiratory Medicine.* 2020;8: 475–481.
14. Mizumoto K, Kagaya K, Zarebski A, Chowell G. Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan, 2020. *Euro Surveill.* 2020;25. doi:10.2807/1560-7917.ES.2020.25.10.2000180

15. de Souza TH, Nadal JA, Nogueira RJN, Pereira RM, Brandão MB. Clinical manifestations of children with COVID-19: A systematic review. *Pediatr Pulmonol.* 2020;55: 1892–1899.
16. McAloon C, Collins Á, Hunt K, Barber A, Byrne AW, Butler F, et al. Incubation period of COVID-19: a rapid systematic review and meta-analysis of observational research. *BMJ Open.* 2020;10: e039652.
17. Pellis L, Scarabel F, Stage HB, Overton CE, Chappell LHK, Lythgoe KA, et al. Challenges in control of Covid-19: short doubling time and long delay to effect of interventions. *medRxiv.* 2020. doi:10.1101/2020.04.12.20059972