COVIDAGENT: INDIVIDUAL BASED MODELLING OF THE COVID-19 PANDEMIC

MURAD BANAJI*, SEPTEMBER 24 2020

Abstract. This document describes the COVIDAGENT modelling environment (formerly known as COVIDSIM) available on github at https://github.com/muradbanaji/COVIDAGENT. The version described here is version 0.2 (abbreviated as v0.2).

1. Basic description of the modelling environment. The environment, termed `COVIDAGENT`, is stochastic and agent based. The agents are infected individuals and it is assumed that there is a pool of individuals "available" to be infected (this may or may not be infinite). An infection event corresponds to an individual's "creation" since, from the modelling point of view, the individual only exists after this moment. Recovery or death can be seen as the individual's deletion.

Time steps are days, and each day an infected individual can either (i) infect new individuals; (ii) do nothing; (iii) recover, or (iv) die. The decision on whether a given individual infects others or not at a given time step is a probabilistic one, determined by overlaying all the different effects in play at that moment, including any mitigation measures in force. Roughly speaking, predetermined infection events are associated with individuals but can be "cancelled" by mitigation.

2. More detail on the modelling environment. Below are the key features of the modelling environment, along with the names of the parameters controlling features of a simulation.

1. **Time course.** Each individual has an associated clock. Time steps are days. Individuals are infected on day 0. Thus day n refers to the time point n days after infection. An individual remains infected until they either recover or die.
2. **Number of simulations.** Since the simulations are stochastic, we may wish to run several simualations with a given set of parameters. The number of simulations to run is set by the parameter `number_of_runs`.
3. **Duration of a simulation.** The total duration of a simulation in days is set by the parameter `totdays`.
4. **Initiating an outbreak.** An outbreak may be initiated by creating a certain number of infected individuals – this number is set via a parameter termed `initial_infections`. Because of the stochastic nature of the model, it is more or less likely, given the values of other model parameters, that the introduction of infected individuals fails to cause an outbreak.
5. **Recovery or death.** At the time of the individual's creation, it is decided, probabilistically, whether they will recover or die. The probability that they will die is the **infection fatality rate**. It is set, as a percentage, via the parameter `death_rate`.
6. **Time to recovery.** For individuals who recover, the time from creation to recovery is set at the time of their creation. An individual's time to recovery is chosen from a distribution with mean value `time_to_recovery`. In v0.2, the default distribution is normal with standard deviation set to be *minus* the parameter `dist_on_recovery`. If this parameter is zero, then the time to recovery is constant. If the parameter is positive, deviations from mean are chosen according to a binomial distribution (this has been kept for legacy purposes).
7. **Time to death.** For individuals who die, the time from creation to death is set at the time of their creation. An individual's time to death is chosen from a distribution with mean value `time_to_death`. The distribution is normal with standard deviation set to be *minus* the parameter `dist_on_death`. If

*Middlesex University, London, Department of Design Engineering and Mathematics. `m.banaji@mdx.ac.uk`.

this parameter is zero, then the time to death is constant. If the parameter is positive, deviations from mean are chosen according to a binomial distribution (this has been kept for legacy purposes).

8. R_0. The population R_0 value (parameter name `R0`) is the average number of individuals that an individual would, in the absence of any other effect, infect. The value of R_0 can in practice deviate from the set value because of truncation errors, and also because of the effects of immunity and mitigation. Actual infection events may not occur because of quarantining, immunity, or mitigation.

9. **Infectivity.** Each individual has an "infectivity": the number of individuals that they would, in the absence of any other effects, infect. An individual's infectivity is chosen, at the time of its creation, from a gamma distribution (mean `R0`, and shape parameter `infshp`), a Poisson distribution (mean `R0`), or a geometric distrbution (mean `R0`). The choice of distribution is set by the parameter `geometric`, with a value of 1 for a geometric distribution, 0 for a Poisson distribution, and -1 for a gamma distribution.

10. **Timing of infection events.** Once an individual's infectivity is set, the times at which they infect others are also chosen. Although the timings are predetermined at the time of creation, the actual infection events may or may not occur as a consequence of quarantining, encounter with immune individuals, or mitigation. The timing of infection events can be chosen from either a gamma distribution (mean `inf_mid` and shape parameter `inf_tm_shp`) modified with rounding and truncation; or a uniform integer distribution on the closed interval [`inf_start`, `inf_end`]. The choice of distribution is set by the parameter `inf_gam` which is set to 1 for a gamma distribution and to 0 for a uniform distribution.

11. **Quarantining.** Infected individuals may be quarantined. Once an individual is quarantined, then they do not infect any more individuals regardless of whether infection events have been scheduled. Whether an individual is quarantined or not is decided, probabilistically, at the time of their creation. The probability (as a percentage) that they will be quarantined is given by the parameter `percentage_quarantined`. If an individual is to be quarantined, then the time at which this occurs is also set at the time of their creation. The mean time from creation to quarantining is given by the parameter `quardate`. If an individual is to be quarantined, then the time after infection at which this occurs may be chosen from a normal distribution with mean `quardate` and standard deviation equal to *minus* the parameter `dist_on_quardate`. If this parameter is zero, then the time to quarantining is constant. If the parameter is positive, deviations from mean are chosen according to a binomial distribution (this has been kept for legacy purposes).

12. **Testing.** Quarantined individuals may be tested. This is how we obtain simulated testing data to compare with official numbers of infections. Whether an individual will be tested or not is decided, probabilistically, at the time of their creation. The probability (as a percentage) that a quarantined individual will be tested is `percentage_tested`. Testing always occurs after quarantining with some delay which which is chosen from a gamma distribution (mean `testdelay`, shape parameter `testdelay_shp`). If `testdelay` is set to 0, then testing occurs immediately on quarantining.

13. **Seroconversion.** In order to reproduce serology data, we can track the development of antibodies to COVID-19. The parameter `time_to_sero` sets the mean time between creation of the individual and detectable levels of antibodies for an individual. Each individual's time to seroconversion may be chosen from a normal distribution with standard deviation set to be *minus* the parameter `dist_on_sero`. If this parameter is zero, then the time to seroconversion is constant. If the parameter is positive, deviations from mean are chosen according to a binomial distribution (this has been kept for legacy purposes).

14. **Infectible population and herd immunity.** The total "infectible" population is a notional quantity representing the number of individuals available to be infected. This allows modelling of **herd immunity** without explicitly introducing non-infected individuals into the model. If the parameter `herd` is set to 0, then the infectible population is infinite, and all parameters related to the population are ignored. Otherwise the population is set initially via the parameter `population` and may be modified by lockdowns (described below).
    The number of infectible individuals at any moment is the current infectible population minus the

(cumulative) total of infected individuals. The current number of infectible individuals affects the probability with which an infection event takes place. Suppose on a given day a given individual is scheduled to infect n individuals. If the individual is quarantined, then these infection events are "cancelled". Even if the individual is not quarantined, then with a certain probability these infection events may be cancelled. This may be because of physical distancing reducing the likelihood of the infection event (see below). If this does not occur, then the infection events may still be cancelled because of immunity. For example, if the current number of infected individuals are 75% of the current infectible population, then there is a 75% probability that infection events scheduled for a given individual on a given day, and not cancelled on account of quarantining or mitigation, fails to take place.

15. **Physical distancing** is used as a catch-all term for all measures which decrease contact. Suppose on a given day a non-quarantined individual is scheduled to infect n individuals. Suppose, that at that moment there is 30% physical distancing. Then, with probability 0.3, these events are all cancelled. This decision is made before the decision on whether to cancel infection events as a consequence of immunity.

16. **Mitigation**. Three mitigation "events" are currently allowed: the introduction of physical distancing alone; and two lockdowns. Events are triggered at the start of a time-step according to some test.
    (a) **A physical distancing event.** This is set to occur if the parameter `physical_distancing` is nonzero. It is triggered either by either infections, recorded infections, or deaths reaching a certain value given by parameters `pd_at_inf`, `pd_at_test` or `pd_at_dth` respectively. (Only one of these parameters should be given a positive value.) Once triggered, the level of physical distancing is set according to the parameter `pdeff1` which gives the probability (as a percentage) that physical distancing occurs at a given time step. (See above.)
    (b) **Lockdowns.** Upto two lockdowns are allowed. A lockdown corresponds to two events: a change in the level of physical distancing; and (if the parameter `herd` is nonzero) a change in the infectible population. The parameter `haslockdown` can take values 0 (no lockdown), 1 (a single lockdown) or 2 (two lockdowns).
    A **first lockdown** can be triggered by either infections, recorded infections, or deaths reaching or exceeding a certain value given by parameters `lockdown_at_inf`, `lockdown_at_test` or `lockdown_at_dth` respectively. (Only one of these parameters should be given a positive value.) Once lockdown is triggered, physical distancing is set by the parameter `pdeff_lockdown`. This overrides any previous value of physical distancing. At the start of lockdown the infectible population is reset via the parameter `infectible_proportion`; the new infectible population is set to be this parameter multiplied by the original infectible population (parameter `population`). However, during lockdown, the infectible population can also change. On day `popleak_start_day` after lockdown, and until and including day `popleak_end_day` after lockdown, the infectible population is increased by `popleak` individuals each day. The length of the lockdown is set by the parameter `lockdownlen`. When this day is reached, physical distancing reverts to its prior value, and the infectible population reverts to the value of the parameter `population` (i.e., disease localisation effectively ends). Lockdown may end earlier than this, as a consequence either of a second lockdown, or the simulation ending.
    A **second lockdown** may be triggered a certain number of days after the first lockdown if the parameter `haslockdown` is set to the value 2. The second lockdown starts `lockdown2startday` days after the start of the first lockdown. The first lockdown may have ended, or may be ongoing at this point. Physical distancing is set by a second value of the parameter `pdeff_lockdown`. The infectible population is reset via a second value of the parameter `infectible_proportion`; the new infectible population is set to be this parameter multiplied by the original infectible population (parameter `population`). On day `popleak_start_day` (second value) after lockdown, and until and including day `popleak_end_day` (second value) after lockdown, the infectible population is increased by `popleak` (second value) individuals each day. The length of the second lockdown is set by the second value of the parameter `lockdownlen`. When this number of days after the start of the second lockdown is reached, physical distancing reverts to its prior value, and the infectible population reverts to the value of the parameter `population`.

17. **Data.** A data file (parameter name `datafile`) may be supplied. This is assumed to contain (at least) two columns of data. The first entry on each row is assumed to be the number of recorded cases upto a given day, while the second entry is the number of recorded deaths upto a given day. This is (at the moment) solely for convenience: outputting this data into a single file along with model outputs allows easier comparisons with data.

18. **Synchronisation.** A synchronisation trigger is useful for aligning several runs of the model, or comparing model outputs to data. Synchronisation is triggered when either infections, recorded infections, or deaths reach or exceed a certain value given by parameters `sync_at_inf`, `sync_at_test` or `sync_at_death`. (Only one of these parameters should be given a positive value.) Synchronisation only affects output to various files, and not the simulation itself. A positive integer parameter `sync_at_time` is set to define a nominal time for the moment that synchronisation takes place. For example, if the number of recorded cases first exceeds the value 1000 at the 10th time step in the data file, then we might set `sync_at_test` to be 1000 and `sync_at_time` to be 9 (the first line is treated as line 0). In the output files `<output_file_sync>`, `<output_file_sync1>` and `<output_file_av>` (see below) outputting of data begins at nominal time 0.

19. **Output files.** If the program is run with, say, the command

    ./a.out `<parameter_file>` `<output_file>`

    then actually several output files are created. Firstly, there is the raw file `<output_file>` with the results of each model run in the following order: time-step, cumulative infections to date, new infections at this time step, current infections (not dead or recovered), cumulative deaths, new deaths at this time step, recorded infections to date, new recorded infections at this time step, number currently infections (approximately - an individual is considered to be infectious during the interval [`inf_start`, `inf_end`] even when timings of actual infection events are chosen from a gamma distribution), and total seroconversion to date. Then there are also a synchronised data file `<output_file_sync>` with the results of each model run but beginning at nominal time 0; a synchronised data file `<output_file_sync1>` with the results of each model run beginning at nominal time 0 and outputs from each run at a given nominal time blocked together; and an average data file `<output_file_av>` with the average of the results from all model runs beginning at nominal time 0.

3. To do list. Here are some possible improvements, as yet not implemented in this version. These can be regarded as a to-do list.

1. There is minimal error-checking on user-entered values in parameter files. If a parameter is given a value outside the range of allowed values, this could lead to silent failure or unexpected behaviour.
2. Allow initiation of an outbreak by a more complex process than creation of individuals at a single time point - e.g., creation of individuals in a stochastic way over a period of time.
3. Allow failure to seroconvert/fading of antibodies in the estimates of seroconversion.
4. Allow fading of immunity. E.g., at each time-step, there is a finite probability that an individual who who has been infected and recovered becomes susceptible again. (Since susceptibles are not explicitly modelled, this event would correspond to an increase in size of the susceptible population.)