

AMTraC-19 (v9.0) user guide

Quang Dang Nguyen,¹ Sheryl L. Chang,¹ Cameron Zachreson,¹ Oliver M. Cliff,¹ and Mikhail Prokopenko¹

¹*Centre for Complex Systems, The University of Sydney, Sydney, New South Wales, Australia, 2006.*

I. SUMMARY

This document describes how to simulate COVID-19 pandemics in Australia using the AMTraC-19 simulation platform. This version (AmTraC-19 v9.0) has an integrated opinion dynamics modelling component within the pandemic simulator. Part (A) describes the steps involved in using the population generation component of the platform to convert census data from the Australian Bureau of Statistics (ABS) into surrogate populations for use with the epidemic simulator. Part (B) describes how to set up and run the epidemic simulator, including how to implement vaccinations, waning immunity, and various types of non-pharmaceutical interventions, with an option to model opinion dynamics on social distancing adoption.

II. PART A: POPULATION GENERATION

A. Input data

The following datasets are required as inputs into `population_generation_2021.cpp` (note that the data files listed here are for 2021, the code will need to be modified slightly if different input file names are used) :

- `AGE.dat` - the age group distribution for each census district (Statistical Local Area Level 1). The columns correspond to the following:
 - Column 1: SA1 identifier (7-digit code)
 - column 2: TOTAL, number of individuals residing in the corresponding SA1
 - Column 3: M0-4, number of males between 0 and 4 years of age
 - Column 4: F0-4, number of females between 0 and 4 years of age
 - Column 5: M5-18, number of males between 5 and 18 years of age
 - Column 6: F5-18, number of females between 5 and 18 years of age
 - Column 7: M19-29, number of males between 19 and 29 years of age
 - Column 8: F19-29, number of females between 19 and 29 years of age
 - Column 9: M30-64, number of males between 30 and 64 years of age
 - Column 10: F30-64, number of females between 30 and 64 years of age
 - Column 11: M64+, number of males over the age of 64
 - Column 12: F64+, number of females over the age of 64
- `CDinSLA.dat` - a list of correspondences between Statistical Area Level 1 (SA1) and Statistical Area Level 2 (SA2). [Note: prior to 2011, areas on the scale of SA1 were referred to as “Census Districts”, and those on the scale of SA2 were referred to as “Statistical Local Areas”.]
- `DZinSLA.dat` - a list of correspondences between Destination Zones (DZN) and SA2s (DZN is the partition used by the ABS for partitioning workplaces).
- `HOUSEHOLD_COMPOSITION.dat` - the distribution of household composition types for each SA1. These are listed in the following order:
 - Column 1: the SA1 identifier (7-digit code)
 - Column 2: S, single adults
 - Column 3: CWOC, couples without children (two adults)
 - Column 4-7: CWC1signl-4, couples with from one to four children (two adults, 1-4 children)
 - Column 8-12: SPF1-5, single parent family with one adult and from one to five children (one adult, 1-5 children)

– Column 13-17: G2-6, non-family groups from with 2 to 6 adults.

- `TravelToWork.dat` - the list of commuter numbers between SA1 and DZN partitions. This is formatted as follows:
 - Column 1: SA1 identifier (7-digit code)
 - Column 2: DZN identifier (11-digit code)
 - Column 3: number of commuters
- `DistanceSLA.txt` - the matrix of geographic distances between the centroids of each SA2 pair (this was created using the ArcMAP GIS software platform with ABS-provided shapefiles) - the top row lists the SA2 identifier for each column and row of the distance matrix.
- `postcode_enrolment_2021.dat` - a list of postcodes and number of enrolled students for reach school (primary and secondary school) in Australia for the year 2016. These are documented by the Australian Curriculum Assessment and Reporting Agency (ACARA), not by the ABS.
- `Postcode_SA2_2021.dat` - a correspondence between postcodes and SA2 partitions (these are not exact correspondences - see Zachreson *et al.* 2018 for a description of how they were generated using ArcMAP to analyse partition overlap).

B. Compiling and running `generate_population_2021.cpp`

This script must be located in the same directory as `debug.h`, `utils.hpp`, `enums.h`, and `generate_population.h`. The header file `generate_population.h` must be edited with appropriate locations for input and output directories (specified with the parameters “database” and “output_directory”). If new input files are used (i.e., for different census years) the `generate_population_2021.cpp` script must be edited to read the data in from the correct files. Note that compilation errors will occur if input files are not edited properly.

The code can be compiled from the Linux command line using `gcc`, specifying C++11 libraries and optimisation:

```
g++ generate_population_2021.cpp -std=c++11 -O2
```

The resulting program can then be executed without input arguments, and will produce command line output related to the population generation process.

C. population generation output files

A successful run of the program will produce the following outputs:

- `population.dat*` : A $5 \times N$ matrix where each row represents an individual Agent generated by the software and the columns correspond to the following:
 1. Age group - an index between zero and 4, corresponding to the age groups listed above, (index increasing with age).
 2. Household - an index corresponding to the household group to which the Agent belongs
 3. Household Cluster - an index corresponding to the Agent’s household cluster
 4. Working Group - an index corresponding to the Agent’s work group (zeros indicate that the Agent does not belong to a working group, and negative indices indicate school mixing groups).

5. SA1 - the region at Statistical Area Level 1 in which the Agent lives.

- WG.dat* : a $3 \times N_{DZN}$ matrix in which each row represents a destination zone partition (DZN) and the columns correspond to the following:
 1. The 11-digit DZN code corresponding to a unique employment region.
 2. The number of working groups in all previously listed DZNs (i.e., the value for the first item in the list is zero).
 3. The number of working groups located in the corresponding DZN.

Since the working groups are indexed as they are generated, this list allows specification of the working group indices located in each DZN region.

- CD_DZ_Nstudents.dat : a list of student commuter numbers between each SA1 \rightarrow DZN pair. The columns correspond to:
 1. The 7-digit SA1 code.
 2. The 11-digit DZN code.
 3. The number of students residing in the corresponding SA1 and going to school in the corresponding DZN.

This dataset can be reproduced by cross-referencing population.dat and WG.dat. In combination with the TravelToWork.dat input file, it gives an estimate of commuting numbers between each SA1 and DZN location.

- SLA_Age_output.dat : a matrix where each row corresponds to a unique SA2 partition and the columns correspond to the following:
 1. The 5-digit SA2 identifier.
 2. Columns 2 - 6 correspond to the number of individuals from each of the five age groups, residing in the corresponding SA2.
- SCHOOLinDZ.dat* : a matrix where each row corresponds to a different school and the columns correspond to the following:
 1. The school's index.
 2. The DZN in which the school is located.
 3. The number of students attending the school.

This dataset allows reconstruction of the spatial distribution of schools, conditional on their student populations.

Data files marked with asterisk (*) are used by the epidemic simulator in the final implementation of the population data structure. Other files (i.e., not marked with asterisk) are produced for validation and visualisation purposes. For calibration of R_0 , refer to AMTraC-19 v7.9 for detailed instructions.

III. PART B: EPIDEMIC SIMULATOR

This section is divided into three parts. The first describes how to run simulations of COVID-19 pandemics with no interventions. The second describes how to apply non-pharmaceutical interventions with an option to model social distancing adoption driven by opinion dynamics. The third describes the implementation of vaccinations and waning immunity from both vaccination and past infections.

A. Basic epidemic simulator

The AMTraC-19 simulator as described in Chang *et al.* (2020) is implemented as follows.

1. compilation

To compile the AMTraC-19 program, navigate to the directory containing the source files and execute the commands in the bash script called ‘re_compile’. This will use GNU Autotools to configure and make the program. Makefile.am contains the compiler flags and lists headers and .cpp source files while configure.ac contains generic configuration commands. This should produce an executable called “AmTraC.exe”.

Alternatively, the code can be compiled from the Linux command line using gcc, specifying C++11 libraries:

```
g++ -std=c++11 virus.cpp agent.cpp create_population.cpp epidemic.cpp statistical_areas.cpp main.cpp -fopenmp -o AMTRAC
```

2. input arguments and files

Table I lists the basic input arguments, their flags, and default values.

argument	flag	default value	description
trans_scaler_mean trans_scaler_std	-t -y	3.33 0	a Normal distribution to sample trans_scaler for each agent (modulating transmission probability - 3.33 corresponds to $R_0 \approx 2$ for 2006, 2011, and 2016)
n_runs	-r	1	number of instances
n_days	-d	196	number of days for the simulation
database_input	-f	./AmTraC_inputs/ 2021_population/	directory from which AMTraC-19 will read input
database_output	-b	./AmTraC_outputs/ 2021_population/	directory into which AmTraC will write
non_pharm_intervention_input_fname	-q	non_pharm_intervention_input/ non_pharm_intervention_input.dat	a file within database_input from which AMTraC-19 will read parameters for non-pharmaceutical interventions
daily_SD_level_input_fname	-l	non_pharm_intervention_input/ daily_SD_level.dat	a file within database_input from which AMTraC-19 will read SD profile for daily basis
daily_fatigue_rate_input_fname	-g	non_pharm_intervention_input/ daily_fatigue_rate.dat	a file within database_input from which AMTraC-19 will read parameters for agents’ risk perception fatigue
pharm_intervention_input_fname	-h	pharm_intervention_input/ pharm_intervention_input.dat	a file within database_input from which AMTraC-19 will read parameters for pharmaceutical interventions (e.g., vaccination)
infection_parameter_input_fname	-v	infection_input /infection_input.dat	a file within database_input from which AMTraC-19 will read parameters related to viral infection.

TABLE I. Input arguments, their command line flags, and a brief description of the parameters.

Additionally, input parameters can be set in the file **infection_input.dat**. If an input parameter is set both as an input argument (i.e., in the comment line) and in input file (i.e., **infection_input.dat**), the value specified in input file takes priority. Parameters that can be configured using the file **infection_input.dat** are as follows:

- `trans_scaler_mean`: mean value of the Normal distribution sampling transmission scaler in population
- `trans_scaler_std`: standard deviation of the Normal distribution sampling transmission scaler in population
- `incubation_period_mu`: mean value (μ) of log normal distribution sampling incubation period in population
- `incubation_period_sig`: standard deviation (σ) of log normal distribution sampling incubation period in population
- `latent_period_mean`: mean value of the Normal distribution sampling latent period in population
- `latent_period_std`: standard deviation of the Normal distribution sampling latent period in population
- `recovery_period_hi`: upper bound of uniform distribution sampling recovery period in population
- `recovery_period_low`: lower bound of uniform distribution sampling recovery period in population
- `after_recovery_period_mean`: mean value of post-recovery period before reinfection
- `after_recovery_period_std`: standard deviation of Post-recovery period before reinfection
- `adult_symptomatic_fraction`: fraction of symptomatic cases in adult population aged over 19
- `child_symptomatic_fraction`: fraction of symptomatic cases in children population aged under 18
- `asymptomatic_infectivity`: relative infectivity of asymptomatic cases compared to symptomatic cases, range [0,1]
- `Detection_base`: baseline detection rate of symptomatic cases
- `Detection_asymp`: detection rate of asymptomatic cases

The value `trans_scalar` independently determines R_0 (as described in Cliff *et al.* “Investigating Spatiotemporal Dynamics and Synchrony of Influenza Epidemics in Australia: An Agent-Based Modelling Approach” (2018)) and this relationship was found consistently using populations constructed from the 2006, 2011, 2016, and 2021 Census. However, the relationship may depend on population structure/demographics and vary in other situations. Calibration can be carried out by initialising many independent index cases and locating the mean number of secondary cases produced for each value of `trans_scalar` tested. An updated description of R_0 calibration is included in the manuscript by Zachreson *et al.* “Interfering with influenza: nonlinear coupling of reactive and static mitigation strategies” (2020), which includes Age-stratified biases for a more accurate calibration of R_0 . Refer to AMTraC-19 v7.9 for detailed instructions and scripts to calibrate R_0 .

The following input files are necessary for the program to run:

- `population.dat*`
- `WG.dat*`
- `CDinSLA.dat*`
- `DZinSLA.dat*`

- SCHOOLinDZ.dat*
- SLA.dat: a two-column data file where the first column contains the list of 5-digit SA2 codes, and the second column contains a dummy index list of increasing integers.
- SeedSLAs.dat: a data file containing information relevant to continuous seeding of index cases: a matrix with rows equal to the number of international airports into which index cases can be introduced. The first column is the daily incoming international air traffic, and the subsequent columns list the 5-digit SA2 identifiers for SA2s within infection range of the corresponding airport (i.e., the airport corresponding to the row). The probability of a new index case being introduced is then a function of the incoming traffic and a scaler value ‘expected_number_infected_’ which is defined in the file ‘epidemic.cpp’ (the default is 0.00004, the proportion of infections per incoming traveller). This format can be used for a broad range of seeding conditions that do not have to correspond to airport traffic. The first entry in a row simply corresponds to the relative incoming infection strength, while the subsequent entries are the SA2 regions affected by that infection strength. The probability of infection for each individual residing in the SA2 regions listed in each row is equal to

$$p_{index} = [\text{incoming passengers (row)}] \times [4 \times 10^{-5}] \times [\text{n affected individuals}]^{-1}, \quad (1)$$

where the last term is the total number of individuals living in all SA2s listed in the corresponding row. On average, each incoming ‘passenger’ has a 0.004 % chance of producing an index case.

Files marked with asterisk (*) are produced by the generate_population_2021.cpp script.

3. command line output

The simulation will produce command line output to monitor the progression of each run. This provides useful information on the number of index cases, incidence, prevalence etc.. [Note: if a segmentation fault is produced just after the program initialises, double-check that all required input files are located in the input directory.]

4. output files

The simulation will create two layers of output directories, one for the entire set of runs, and a subdirectory for each individual run. The names of these directories contain important information about the parameters used (parameter information will also appear in the command line output):

The upper-level directory follows the naming convention as: D[date]_[time]_I[random_number]_AP

The initials “_AP” stands for ‘airport’ and indicate the type of seeding algorithm used for introducing index cases, airport seeding is the default for the variable seed_type which controls how seeding is implemented. This can be altered, but at this point, only two other options have been implemented in previous versions of AMTraC-19. These are (a) NumberOfSLAs and (b) PopulationRandom, which implement random seeding protocols and require the additional command line flags with specified values. These will not be discussed here as the seeding protocols were specific to a particular investigation by Cliff *et al.*(2018).

The subdirectory names for each run are formatted as: R[run index]_D[date]_[time]

Each simulation will produce the following output files:

- I.SLA.dat, a matrix of timeseries incidence data, with one column for each SA2 region (the row indices correspond to simulation each day of the simulation).
- I.SLA-[0-4].dat, each file corresponds to the location-specific incidence timeseries for each of the five age groups.

- I_detection_Ag.dat, a matrix of timeseries incidence (detection) data, with one column for each age group (the row indices correspond to each day of the simulation).
- I_illness_Ag.dat, a matrix of timeseries incidence (illness) data, with one column for each age group (the row indices correspond to each day of the simulation).
- I_infection_Ag.dat, a matrix of timeseries incidence (infection) data, with one column for each age group (the row indices correspond to each day of the simulation).
- P_SLA.dat, as with I_SLA.dat, but tabulating prevalence rather than incidence.
- P_SLA-[0-4].dat, location-specific prevalence timeseries for each age group.
- P_detection_Ag.dat, as with I_detection_Ag, but tabulating prevalence (detection) rather than incidence (detection).
- P_illness_Ag.dat, as with I_detection_Ag, but tabulating prevalence (illness) rather than incidence (illness).
- P_infection_Ag.dat, as with I_detection_Ag, but tabulating prevalence (infection) rather than incidence (infection).
- IP.dat, the incidence timeseries (first column) and prevalence (second column) for the whole population.
- Infected_Pinfect_CDI_CDP.dat - a four-column data set. The first column is the incidence of infected individuals (including asymptomatic cases), the second column is the prevalence of infected individuals (including asymptomatic cases), the third column is the incidence of SA1 regions transitioning from 0 infected residents to > 0 infected residents, and the fourth column is the number of SA1 regions with > 0 infected residents, the row numbers correspond to time in days.
- N_health_states.dat - a four-column data set specifying the number of agents in each of four selected health states over time: Susceptible, Infected - Asymptomatic, Infected - Symptomatic, Recovered. The row numbers correspond to time in days.
- N_recovered_susceptibles.dat - a three-column data set specifying the number of agents in each of three conditions over time: recovered, recovered from an infection, recovered from two or more infections. The row numbers correspond to time in days.

B. Non-pharmaceutical interventions

Non-pharmaceutical interventions (NPIs) are included in the program, as described in Chang *et al.* (2020). These interventions are applied to affected agents following either random assignment (for case isolation, household quarantine, and school closure), or opinion dynamics (for social distancing, see manuscript Chang *et al.* (2024), “Impact of opinion dynamics on recurrent pandemic waves: balancing risk aversion and peer pressure” for detailed descriptions).

1. input arguments and files

Implementation of NPIs requires one or more of the following files depending on the selection of NPIs. These input files must be located in the input directory specified by the command line argument with flag -f.

- non_pharm_intervention_input.dat
- daily_SD_level.dat
- daily_fatigue_rate.dat

The first file “non_pharm_intervention_input.dat” specifies most of the intervention-specific parameters and must be included for implementation of NPIs. The second file “daily_SD_level.dat” specifies varying social distancing adoption levels during the simulation time frame. The third file “daily_fatigue_rate.dat” specifies the fatigue rate of perceived risk of infection.

Note that the compliance to case isolation (CI), household quarantine (HQ), and school closure (SC) follows a top-down assignment rule where agents are sampled randomly until the compliance reaches the specified level defined in the input file. The assignment of SD-adopting agents, however, could either follow a top-down approach similar to other NPIs, or a bottom-up approach where compliance for each agent is driven by opinion dynamics. The switch between the two assignment rules are governed by boolean parameter “opinion_based_compliance”. Note that SD compliance levels defined in the file “daily_SD_level.dat” are only read when SD assignment follows the top-down approach (i.e., input-driven). When SD assignment follows the bottom-up approach (i.e., opinion-driven), SD compliance is dynamically determined in simulation based on the current pandemic severity. These possible scenarios are summarised below.

- All NPIs follow top-down assignment with a static SD adoption level :
non_pharm_intervention_input.dat; opinion_based_compliance=0
- All NPIs follow top-down assignment with dynamic SD adoption levels:
non_pharm_intervention_input.dat and **daily_SD_level.dat**; opinion_based_compliance=0
- SD follows opinion-driven, bottom-up assignment and other NPIs follow top-down assignment:
non_pharm_intervention_input.dat and **daily_fatigue_rate.dat**; opinion_based_compliance=1

non_pharm_intervention_input.dat follows the format contained in the existing parameter file, the order of parameter specification is not important. This file contains parameters used in NPIs following top-down approach (Section III B 1 a) and opinion dynamics (Section III B 1 b).

a. Input-driven NPIs following top-down approach

- SD_intervention: boolean flag for turning on social distancing $\{0, 1\}$
- n_ill_SD_trigger: cumulative number of illnesses triggering social distancing
- compliance_rate_SD: proportion of the population who comply with social distancing, range $[0, 1]$. This value will be overlooked if the opinion dynamics flag is turned on.
- duration_SD: number of days social distancing measures apply after they are implemented. This value will be overlooked if the opinion dynamics flag is turned on.
- SD_assignment_period: number of days during which a random assignment of agents compliant with social distancing stays unchanged, unless interrupted by a different daily SD level from the file “daily_SD_level.dat”
- FoI_SD_community: multiplier for community interactions under social distancing, range $[0, +\infty]$
- FoI_SD_work: multiplier for workplace interactions under social distancing, range $[0, +\infty]$
- FoI_SD_home: multiplier for household interactions under social distancing, range $[0, +\infty]$

- CI_intervention: boolean flag for turning on case isolation 0, 1
- compliance_rate_CI: proportion of population who comply with case isolation, range $[0, 1]$
- FoI_CI_community: multiplier for community interactions under case isolation, range $[0, +\infty]$
- FoI_CI_work: multiplier for workplace interactions under case isolation, range $[0, +\infty]$
- FoI_CI_home: multiplier for household interactions under case isolation, range $[0, +\infty]$

- HQ_intervention: boolean flag for turning on home quarantine 0, 1

- `compliance_rate_HQ`: proportion of population who comply with home quarantine, range $[0, 1]$ (NOTE: home quarantine compliance is conditional on case isolation compliance.)
- `FoI_HQ_community`: multiplier for community interactions under home quarantine, range $[0, +\infty]$
- `FoI_HQ_work`: multiplier for workplace interactions under home quarantine, range $[0, +\infty]$
- `FoI_HQ_home`: multiplier for household interactions under home quarantine, range $[0, +\infty]$
- `School_closure`: boolean flag for turning on school closure 0, 1
- `n_ill_SC_trigger`: cumulative number of illnesses triggering school closure]
- `duration_SC`: number of days school closure measures apply after they are implemented
- `FoI_SC_community`: multiplier for community interactions under school closure, range $[0, +\infty]$
- `FoI_SC_work`: multiplier for workplace interactions under school closure, range $[0, +\infty]$
- `FoI_SC_home`: multiplier for household interactions under school closure, range $[0, +\infty]$
- `p_parent_stays_home`: probability that a parent will stay home under school closure, range $[0, 1]$.

b. Opinion-driven NPI following bottom-up approach

- `opinion_based_compliance`: boolean flag for turning on opinion dynamics on social distancing adoption $\{0, 1\}$
- `beta_perc_initial`: initial value of the perceived infection probability β_0
- `beta_perc_fatigue_rate`: a constant fatigue rate of the perceived infection probability β_0 . Varying fatigue rate can also be implemented if “`daily_fatigue_rate.dat`” is non-empty.
- `opinion_threshold`: risk threshold k
- `opinion_horizon_mean_sigmoid_midpoint`: parameter u in the manuscript as the midpoint value of memory horizon following the sigmoid function.
- `opinion_horizon_mean_sigmoid_v`: parameter v in the manuscript that governs steepness in sigmoid function
- `opinion_horizon_mean_sigmoid_delay`: the simulation day D when the memory horizon reaches the midpoint
- `opinion_horizon_base`: baseline value of memory horizon B in population
- `opinion_horizon_std`: Standard deviation of a Normal distribution for memory horizon with the mean as the sum of `opinion_horizon_mean_sigmoid_midpoint` and `opinion_horizon_base` ($u + B$)
- `social_pressure_weight`: weight of peer pressure, $\lambda \in [0, 1]$
- `social_pressure_household`: weight of household in selected social contexts, $\psi_{HH} \in [0, 1]$.
- `social_pressure_workinggroup`: weight of working groups in selected social contexts $\psi_{WG} \in [0, 1]$
- `fraction_always_compliant`: fraction of complying agents, $[0, 1]$
- `fraction_never_compliant`: fraction of non-complying agents, $[0, 1]$

In order for AMTraC-19 to read the parameter inputs listed in “non_pharm_intervention_input.dat”, the command line argument:

`-q <name of parameter input file>`

must be entered at runtime. The default argument is “non_pharm_intervention_input/non_pharm_intervention_input.dat”, and this file must be located in the database input directory specified by the command line argument with flag `-f`.

daily_SD_level.dat is required when SD assignment follows the input-driven top-down approach. This file specifies the varying social distancing adoption levels during the simulation time frame. It is a 2-column matrix where the second column specifies the social distancing adoption level (range [0, 1]) effective from the simulation day specified in the first column. The input file can have multiple rows to vary the compliance level in different simulation periods. For example, if the file contains entries [1 0], [22 0.5], and [43 0.3], then, following the initial period with zero social distancing, the social distancing level is set to 0.5 from day 22, and then to 0.3 from day 43 onwards. Note that:

- The specified simulation days must not exceed the number of simulation days, `n_days`, specified with flag `-d` in input parameters (see Section A.2, TABLE I);
- Social distancing will only be triggered if the cumulative number of illnesses exceeds the threshold (`n_ill_SD_trigger`) specified in “SD_intervention_input.dat”;
- The overall duration of social distancing is limited by `duration_SD` specified in “non_pharm_intervention_input.dat”;
- If the file “daily_SD_level.dat” is not found in the directory, or a simulated day is outside the range specified by the file, then the simulator applies the default `compliance_rate_SD` specified in “non_pharm_intervention_input.dat”.
- If opinion dynamics flag is turned on, “daily_SD_level.dat” will be overlooked as SD compliance levels are determined dynamically by opinion dynamics.

In order for AMTraC-19 to read the parameter inputs listed in “daily_SD_level.dat”, the command line argument:

`-l <name of parameter input file>`

must be entered at runtime. As mentioned above, this file must be located in the database input directory specified by the command line argument with flag `-f`.

daily_fatigue_rate.dat is an optional file to enable varying fatigue rates during the simulation time frame. It contains a 2-column matrix where the second column specifies the fatigue rate effective from the simulation day specified in the first column. The input file can have multiple rows to vary fatigue rates in different simulation periods. For example, if the file contains entries [0 0.001], [10 0.05], and [20 0.1], then, following the initial period with a fatigue rate of 0.001, the fatigue rate changes to 0.05 from day 10, and then to 0.1 from day 20 onwards. Note that:

- the specified simulation days must not exceed the number of simulation days, `n_days`, specified with flag `-d` in input parameters (see Section A.2, TABLE I);
- if the file “daily_fatigue_rate.dat” is not found in the directory, or a simulated day is outside the range specified by the file, then the simulator applies the default fatigue rate specified in “non_pharm_intervention_input.dat”.

In order for AMTraC-19 to read the parameter inputs listed in “daily_fatigue_rate.dat”, the command line argument:

`-g <name of parameter input file>`

must be entered at runtime. As mentioned above, this file must be located in the database input directory specified by the command line argument with flag `-f`.

2. command line output

The command line output will display the numbers of individuals who are affected by each of the social distancing measures. It will also print the input parameters at the beginning of each run. [Note: if a segmentation fault is produced just after the program initialises, double-check that all required input files are located in the input directory.]

3. output files

In addition to the output files listed for the basic implementation of the program, the following additional output file will be created:

- `CIHQ.dat` - a two-column data set. The first column is the number of individuals affected by case isolation, and the second column is the number of individuals affected by home quarantine.
- `LSD_compliance.dat` - a two-column dataset where the first column is the daily incidence and the second column is the corresponding social distancing adoption level on the day.

C. Vaccination intervention and waning immunity

Vaccinations have been included by allocating two types of vaccines (priority and general), using predefined age-dependent vaccine allocation ratios, as described in Zachreson *et al.* (2021). Two types of vaccination campaigns are implemented: pre-pandemic coverage and progressive vaccination rollout. In addition, waning immunity (from both vaccination and previous infections) is integrated into the model by tracking the vaccination and infection record of each agent.

1. input arguments and files

Simulation of vaccinations and waning immunity requires the input file “`pharm_intervention_input.dat`”, which must be located in the input directory specified by the command line argument with flag `-f`. This file will specify the vaccination-related (Section III C 1 a) and immunity-related parameters (Section III C 1 b).

a. Vaccination-related parameters

- `Pharm_intervention`: boolean flag for turning on vaccination [0, 1].
- `n_vac_pre_epidemic_PV`: the number of individuals receiving the priority vaccine pre-emptively (pre-pandemic mode).
- `fraction_vac_pre_epidemic_PV`: fraction of population receiving the priority vaccine pre-emptively (pre-pandemic mode). Note that `n_vac_pre_epidemic_PV` is prioritised over `fraction_vac_pre_epidemic_PV` if disagreement in input occurs.
- `n_vac_per_week_PV`: the number of individuals receiving the priority vaccine per week in a progressive rollout mode.
- `n_vac_pre_epidemic_GV`: the number of individuals receiving the general vaccine pre-emptively (pre-pandemic mode).
- `fraction_vac_pre_epidemic_GV`: fraction of population receiving the general vaccine pre-emptively (pre-pandemic mode). Note that `n_vac_pre_epidemic_GV` is prioritised over `fraction_vac_pre_epidemic_GV` if disagreement in input occurs.
- `n_vac_per_week_GV`: the number of individuals receiving the general vaccine per week in a progressive rollout mode.
- `p_vac_65_and_over`: the number of vaccines allocated to older adults aged over 65, in each iteration of the age-stratified allocation cycle

- `p_vac_18_to_64`: the number of vaccines allocated to adults aged between 18 and 64, in each iteration of the age-stratified allocation cycle
- `p_vac_children`: the number of vaccines allocated to children aged under 18, in each iteration of the age-stratified allocation cycle
- `VEc_PV`: maximum efficacy against symptomatic infection for the priority vaccine
- `VEt_PV`: maximum efficacy against infection transmission for the priority vaccine
- `VEc_GV`: maximum efficacy against symptomatic infection for the general vaccine
- `VEt_GV`: maximum efficacy against infection transmission for the general vaccine

b. Immunity-related parameters

- `VEc_NR`: maximum compound natural immunity against symptomatic infection
- `VEt_NR`: maximum natural immunity against infection transmission from infection
- `daily_waning_VEc_NR`: waning rate of compound natural immunity against symptomatic infection per cycle
- `per_cycle_waning_VEt_NR`: waning rate of natural immunity against symptomatic infection per simulation cycle
- `per_cycle_waning_VEc_PV`: immunity waning rate of `VEc_PV` per simulation cycle
- `per_cycle_waning_VEt_PV`: waning rate of `VEt_PV` per simulation cycle
- `per_cycle_waning_VEc_GV`: waning rate of `VEc_GV` per simulation cycle
- `per_cycle_waning_VEt_GV`: waning rate of `VEt_GV` per simulation cycle

In order for AMTraC-19 to read the parameter inputs listed in 'pharm_intervention_input.dat', the following command line argument must be entered at runtime:

`-h <name of parameter input file>`

The default argument is "pharm_intervention_input/pharm_intervention_input.dat".

2. command line output

The command line output will display the discrete-step (i.e., for each simulation day) and the cumulative (i.e., total) number of vaccinated individuals by age groups (i.e., in accordance with the vaccine allocation ratio) and the type of vaccine received (i.e., priority or general). It will also print out the input parameters at the beginning of each run.

3. output files

In addition to the output files listed in section III.A.4, the following additional output files are created:

- `I_detection_VACge_Ag.dat`, a matrix of time series incidence (detection) data for the individuals who have received the general vaccine, with one column for each age group (the row indices correspond to each day of the simulation).
- `I_detection_VACpr_Ag.dat`, a matrix of time series incidence (detection) data for the individuals who have received the priority vaccine, with one column for each age group (the row indices correspond to each day of the simulation).
- `I_infection_VACge_Ag.dat`, a matrix of time series incidence (infection) data for the individuals who have received the general vaccine, with one column for each age group (the row indices correspond to each day of the simulation).

- `I_infection_VACpr_Ag.dat`, a matrix of time series incidence (infection) data for the individuals who have received the priority vaccine, with one column for each age group (the row indices correspond to each day of the simulation).
- `IinfGE_IinfPR_IdetGE_IdetPR.dat` - a four-column data set. The first column is the incidence of *infected* individuals (including asymptomatic cases) for those who have received the *general* vaccine, the second column is the incidence of *infected* individuals (including asymptomatic cases) for those who have received the *priority* vaccine, the third column is the incidence of *detected* individuals (including asymptomatic cases) for those who have received the *general* vaccine, and the fourth column is the incidence of *detected* individuals (including asymptomatic cases) for those who have received the *priority* vaccine. The row numbers correspond to time in days.
- `Vaccinations_general_Ag.dat` - a four-column data set recording the number of individuals that received the general vaccine by age group. The first column is the total number of individuals, the second column is the number of children (aged < 18), the third column is the number of adults aged $18 - 64$, and the fourth column is the number of older adults (aged > 65). The row numbers correspond to time in days.
- `Vaccinations_priority_Ag.dat` - a four-column data set recording the number of individuals that received the priority vaccine by age group. The first column is the total number of individuals, the second column is the number of children (aged < 18), the third column is the number of adults aged $18 - 64$, and the fourth column is the number of older adults (aged > 65). The row numbers correspond to time in days.
- `N_immunity_states.dat` - a nine-column data set where each row corresponds to simulation days and each column specifies the number of agents in the nine immunisation conditions as follows:
 - Column 1: No immunity,
 - Column 2: Immunity from a single vaccination,
 - Column 3: Immunity from two or more vaccinations,
 - Column 4: Immunity from a natural recovery,
 - Column 5: Immunity from two or more natural recoveries,
 - Column 6: Immunity from both a single vaccination and a natural recovery,
 - Column 7: Immunity from a single vaccination and two or more natural recoveries,
 - Column 8: Immunity from two or more vaccinations and a natural recovery,
 - Column 9: Immunity from two or more vaccinations and two or more natural recoveries.