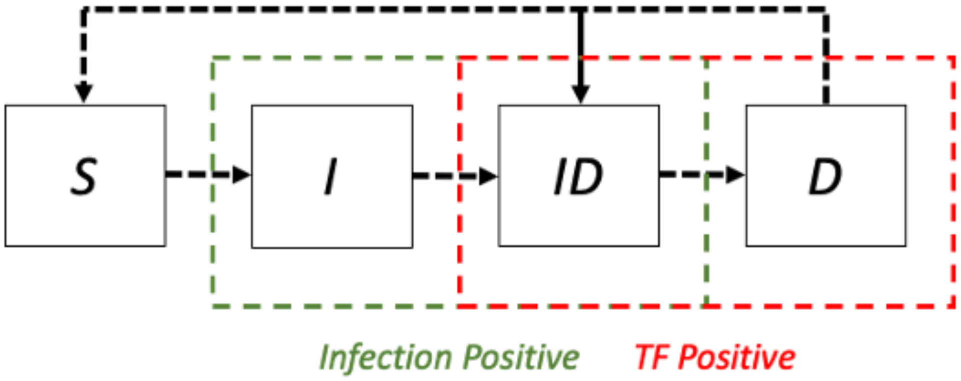
Notes on TrachomaTrack: Agent-based trachoma model updated October 2023.

1. Overall model Structure:



The model is a stochastic individual-based model of *C. trachomatis* transmission, with people transitioning through four sequential states: Susceptible (S), infected but not yet diseased (I), infected and diseased (ID) or diseased but no longer infected (D). Here disease refers to clinical trachoma, specifically TF. Information about every individual in the population is tracked at each timestep.

All model functions are stored in file "Trachoma_Track_functions.R" and code to run simulations is stored in "Simple_runs_Trachoma_Track"

2. "vals" individual-level information stored at each timestep.

The timestep is 1 week. For each individual in the model population, there are 11 values given in "vals" which are updated at each timestep.
The individual's infection and disease status, which are tracked as positive or negative (1,0, Boolean) and correspond to the compartment of the model each individual is in:

	IndI	**IndD**
S	0	0
I	1	0
ID	1	1
D	0	1

IndI=Individual infection status
IndD=Individual disease status

The other values given in vals are:
No_Inf: The number of infections each individual has had in their lifetime, increases by one every time they have an infection.
T_latent: This relates to the latent period, the amount of time a person spends in the I category. When a person becomes infected (newInf in the timestep function), this value

gets set to 2, with one subtracted at each timestep. When their time in the I category has expired, they acquire disease (newDis) and transition into the ID category.
T_ID: This relates to the amount of time a person spends in the ID category. When a person becomes diseased (newDis in the timestep function) the amount of time they will spend in the ID category is calculated and given here using the function ID_period_function. The T_ID value will then count down with every timestep, and when it has expired, they will have cleared infection (newClearInf) and transition into the D category. ID_period_function has been updated from previous versions so is now a function of both age and number of infections.

T_D: This relates to the amount of time a person spends in the D category. When a person clears infection (newClearInf) this time is calculated using the function D_period_function. This then counts down with each timestep. People can either move out of this category into the S category when T_D has expired, or they can acquire a new infection and move into the ID category. D_period_function has been updated so is now a function of both age and number of infections.

Ind_latent: Is each individual's latent period. This is currently fixed at 2 weeks for all individuals
Ind_ID_period_base: This is each individuals "baseline" maximum ID period, currently taken from a Poisson distribution with average (lambda) being the point estimate from the data This is used to calculate their T_ID each time they acquire a new infection.
Ind_D_period_base: This is each individuals "baseline" maximum D period, currently taken from a Poisson distribution with average (lambda) being the point estimate from the data.

bact_load: Relates to each person's bacterial load, which is 1 for their first infection and reduces with infection history (No_Inf) following a negative exponential.

Age: Individual's age in weeks. Increases by 1 with each timestep until the age of 70 (maximum age in the model, set in "demog", can be varied).

3. Functions.

Can give further information on each function if requested, this is just a brief summary on what main functions do

stepF_fixed: This is the main step function which identifies transitions to be made at each "normal" timestep (timesteps in which MDA is carried out are given in MDA_timestep). Individuals to make transitions are given as NewInf (new infections), newDis (new disease) newClearInf (clear infection and move into D category), newClearDis (clear disease and move into S category), NewReInf (people getting infected from the D state) or reset_indivs (people who die at this timestep).
Which individuals are susceptible to infection are identified, either Ss or Ds, as people can infected in either the S category or the D category (previously susceptible at a scaled rate, now equally susceptible).
The force of infection (lambda) on each individual at each timestep is calculated as lambda_step, using the function getlambdaStep.

From this, which individuals become infection (newInf) is given.
Individuals then transition through the categories after fixed periods of time which are given as Ind_latent for duration in I, and calculated using ID_period_function for ID, D_period_function for D. When their time in each category has expired, they are identified as newInf, newDis, or newClearInf.

People also age 1 week at each timestep, and when they reach the age of 70 they are reset (reset_indivs), effectively recreating them as a new person with age 0 and no infection history, to simulate new births into the population. They are also reset if they die due to natural background mortality rate.

getlambdaStep: this calculates the force of infection on each person in the population, which is a function of the total number of infectious people at each timestep and their bacterial load, the social mixing matrix, and the age group of each person in the population. Age groups are given as young children (y_children, ages 0-9y) older children (o_children, ages 9-15y) and adults (ages 15+).

Reset: This determines who dies in each timestep. This is either due to reaching the maximum age of 70, or due to random background mortality rate. These individuals are "reset" in the next timestep.

set_t_MDA: This sets timesteps at which MDA is to take place dependent on the number of rounds of MDA intended and the frequency of MDA, and returns these timesteps as a vector. Use this for generic scenarios where don't know actual dates of MDA

MDA_dates_as_timestep: If have actual dates of MDA use this to return timesteps for MDA.

Tx_matrix: This creates a matrix for whether an individual will receive MDA at each timestep. Matrix rows correspond to each person in the population and matrix columns relate to rounds of MDA. The first round of MDA (column 1) is randomly assigned for each individual as a function of MDA coverage. For each subsequent round the probability a person receives MDA is a function of the number of rounds they have previously received, given a correlation parameter rho. Rho is in parameters and should be varied in sensitivity analysis if no data to inform it.

MDA_timestep: This is the step function which identifies transitions to be made at a timestep in which MDA is carried out. Those individuals to be treated and cured are identified by the function doMDA, and these individuals will have their infection status and bacterial load set to 0.

doMDA: This determines who is cleared of infection during a round of MDA. For each person who is indicated to be treated by the treatment matrix, the probability they clear infection is determined randomly by the treatment efficacy rate, given as MDA_Eff. Young babies (<6m old) are given eye ointment rather than tablets and the efficacy is assumed to be 50% less that the tablets. This function returns those individuals who are treated and cured at and MDA timestep.

ID_period_function: This determines the amount of time a person spends in the ID category as a function of their baseline ID period (Ind_ID_period_base) the number of infections they have had ad their age.

D_period_function: This determines the amount of time a person spends in the D category as a function of their baseline D period (Ind_D_period_base), age and the number of infections they have had. New parameters were estimated for this function, documented in: Age_vs_acquired_immunity_update.pdf

bacterialLoad: This function scales a person's bacterial load in their current infection according to how many infections they have previously had.

Set_inits: This function is carried out at the start of any simulation to create a population and assign their ages, number of infections, and baseline I, ID and D periods.

Seed_infection: This function seeds an initial number of infected based on target TF prevalence, can just use a fixed % e.g 5%, but may need longer burn-in

Init_ages: This is called by the Set_inits function to assign each individual in the population an age at the start of the simulation, use truncated geometric distribution.

4. Initialising simulations

Parameters relating to the population are stored in "parameters".
Main parameters you may vary here are:
MDA coverage (given here as **MDA_Cov).**
Compliance correlation (given here as **rho).**

Other parameters relating to the simulation are given in "sim_params".
This includes burn-in period (**burnin**), total duration you want the simulation to run for in weeks (**timesim**), number of MDA rounds (**N_MDA**) and the number of simulations you want to carry out (**n_sim**).

Some additional parameters relating to the demography of the population are given in "demog". This is maximum age, mean age and death rate (**tau**).

To initialise a simulation, use **initialise_pop_and_seed.** You need to give a "Target_TF_1_9" here which is the TF prevalence in ages 1-9 that you want at equilibrium, and it will seed infection and assign number of infections at the start according to this value. However, the TF you get to is actually determined by beta, putting it here just means equilibrium is reached faster, and if using a long burn-in doesn't matter

Then create **MDA_times**, generally have MDA dates when fitting to actual data, so use **MDA_dates_as_timestep.**

Alternative if want to just generic MDA programme use **set_t_MDA** to create a vector for when MDA is to be carried out.

Use **Tx_mat** to determine who gets treated at each MDA round.

sim_Ind_MDA is a function to run a single simulation with MDA at the time points given as MDA_times. The output is:
t= Time
True_Prev_Infection_children_1_9= Prevalence of infection in children aged 1-9 at each timestep. Prevalence of infection is the total number of infected individuals of an age group (those for whom IndI=1) over the total population for that age group.
True_Prev_Disease_children_1_9= Prevalence of disease in children aged 1-9 at each timestep (total number of children for whom IndD=1 over number of children of that age group). This corresponds to TF in ages 1-9 in the data. This is generally what we are fitting to.
True_Prev_Disease= Prevalence of disease in the whole population
True_Prev_Infection= Prevalence of infection in the whole population.

5. Transmission parameter beta "bet"

The parameter "bet" is the main parameter that can be varied to represent different levels of transmission and create different levels of prevalence. Very rough estimates for a given TF prevalence are in "inits" file, however this is just a starting point, beta needs to be explicitly fitted.