ModelMapping code instructions

This files describes how to run the correct files to reproduce the results in the main paper (Pellis, L. et al, 2019, "Systematic selection between age and household structure for models aimed at emerging epidemic predictions", Nature Communications).

General comments

To **run a MatLab file**:

1. Open the file directly in MatLab and run it;
2. When MatLab prompts you, click on "Change Folder", to change the current directory to the directory where the file lives. The rest of the folder structure should be set up to handle everything else.

The **folder structure** is organised as follows:

1. "code-…" contains MatLab codes, the input files and the executable files for the individual-based stochastic simulation;
2. "saved-…" contains the complete set of workspaces and figures used in the paper;
3. "output-…" contains by default the results of the codes run by the user (workspaces, figures, etc.), so that files are not overwritten by mistake.

Generally, a code to run the model mapping (in its many variants) generates a workspace in "output-workspaces", while a code to create a figure, reads the relevant workspaces from "saved-workspaces" and generates a figure in "output-figures" (an option allows trying to generated the figure from "output-workspaces", if desired).
In addition:

4. "tools" contains general purpose codes for tasks not directly related to the model mapping, either custom made or downloaded from the internet;
5. "stochastic-simulation" contains the C++ code to generate the executable file to run the individual-based stochastic simulation, both as an Xcode project for Mac and a Visual Studio solution and project for Windows. The subfolder "saved-files" contains a copy of all files related to the stochastic simulation used by the authors.
6. "data" contains code and original data to generate the data files used directly in the Matlab and C++ code.

Important notes

- Due to change of notation, in the code "pAA" is used instead of "p_{aa}" and "Rh" instead of "\beta_h".
- Due to the way the stochastic simulation code was originally set up, for anything to do with the simulation, "W" refers to "households" (i.e. *W_GB_5.dat* refers to the household size distribution for the simulations, output file names contain "Rw", etc.) and "H" for the schools (which are irrelevant for the current work, but their size distribution is still required by the simulation: the *H_GB_5.dat* has only groups of size 100 by default)
- Similarly, the names of the parameters in the stochastic simulation are different from the notation in the paper: SIGMA is used for the relative susceptibility of children (psiG in the paper, which coincide with psiH); RHO for the relative infectivity of children (instead of phiG = phiH); PHI for the assortativity (instead of thetaG). This is reflected in the names of the temporary files generated by the simulation, where these three values are called: "sigma", "rho" and "ass".
- In most figures, I have experienced issues in writing the x-axis label that occur only on my version of Mac (MatLab R2016a), for no apparent reason (they seem not to occur on Windows, though other problems might arise): specifically, when using the subscript in "(p_{aa})", spacing goes all over the place. The issue does not appear in the figure created

in MatLab, but only becomes visible in the generated pdf. I offer a decent compromise in the figure codes with a slightly modified text string for each axis. However, the final figures are then generated in Windows MatLab R2019a, where this problem does not occur.

- I also noted that grid-based figures (e.g. left part of Figure 2 or all of Figure 3 in the main text) generated on MatLab 2016a for Mac, when opened in a pdf reader, sometimes show colours filling the cells of the grid exactly (each cell has 1 colour) and sometimes show blurred edges between coloured regions. This appears not to be a problem of the figure itself, but it depends on the piece of software used to view the pdf. For example, Mac Viewer shows the edges blurred while Adobe Acrobat Reader DC for Mac shows them neat (when opening exactly the same pdf file). However, when generated on MatLab 2019a on Windows (as in the final version of the manuscript), this does not seem to be the case, and all figures appear neat on any pdf viewer.

Running specific codes

Main model mapping

Run code *Model_mapping_code.m* from folder "code-model-mapping".

<u>Structural options</u> ("switches"):

- Activate_checks =
    - 0: This computes the relevant variables only once and in the most efficient way, and should only use code files in the same folder as the main file.
    - 1: This should run other sub-codes to cross check results of the main code by computing them in multiple different ways. It uses code files in the subfolder "check-codes".
- Activate_C_codes =
    - 0: Individual-based stochastic simulations are not run, so only the final size is computed, and not the peak incidence and time to the peak.
    - 1: The stochastic simulations are called by the MatLab code and all three outputs are computed. The executable (at least for Mac) and all supporting files should be in the folder "code-model-mapping". Temporary files containing the output of the simulation are created and destroyed in each loop.
- Activate_continue =
    - 0: The code performs the full for-loops (runs for a long time).
    - 1: This option first checks if you have attempted a previous run with the same parameters, which might have crashed half-way through, and continues from the last saved loop. Very useful if your have discovered bugs hours into a simulation, or you had to interrupt previous simulations for other reasons.
- Activate_workspace_saving =
    - 0: The final workspace is not saved, and only an emergency file with a standard name is saved (NOT RECOMMENDED)
    - 1: Save workspace at the end of the calculations (default)
- Activate_delete_files =
    - 0: The files created by the individual-based stochastic simulation are created in each loop but not removed, so they accumulate (NOT RECOMMENDED)
    - 1: Files are deleted once used, to avoid cluttering (default).

<u>Parameters to vary</u>:

- Country: This gives the population structure (fraction of adults and children and household composition used). Values:
    - GB: Great Britain (default)
    - SL: Sierra Leone
    - SA: South Africa

- pop = Population: This gives the contact pattern (ratio between the number of contacts per day children have, versus adults) and mixing (random or assortative). Relevant values for the parameters are fixed when commenting/uncommenting the relevant line.
  Main values:
    - 2ran: random (default). Note: the "2" comes from the fact that F_c = 22.73% for GB. Then in was easy to keep files with this option appearing first in alphabetical order.
    - UK: based on UK contact patterns (POLYMOD), so highly assortative.

  Other values (rarely used):
    - m4r, m4UK, m5r, m5UK: see Supplementary Discussion (Section 2.3.1), for details on these intermediate contact patterns
    - ass: extremely assortative values, just as an extra test.
- R0: Basic reproduction number:
    - Basic values: 1.5, 2 and 4
    - Other values: 1.1, 1.3, 1.7, 2.3, 2.7 and 3.2
- phiG: value for the relative infectivity of children versus adults (G = global, i.e. in the community, but then the code sets phiH = phiG)
    - Values: 1, 1.5 and 2

<u>Output</u>:
- The main output is a workspace with all variables saved for values of "p_{aa}" and "\phi" on a grid.
- The workspace is saved in the relevant subfolder of the folder "output-workspaces".
- For emergency, the workspace is also saved (in "code-model-mapping") with name *workspace_emergency_save*.
- Temporary workspaces (after each loop) are saved in "code-model-mapping/temp".

<u>Warnings</u>:
- When Activate_delete_files = 1, at the end of each loop MatLab attempts to cancel the temporary .dat files created by the individual-based stochastic simulation. Sometimes (e.g. when there is no transmission in households) the three files created for the three models have the same parameters and therefore are overwritten on each other. When this happens and MatLab is requested to delete each of them, only one can be found, so MatLab issues a warning message for the second and third deletion attempt. This is nothing to worry about.
- Sometimes, when running multiple instances of MatLab to run multiple versions of the model comparison with different parameters on the same machine, one instance might delete one of the temporary files that has just been created by another. At this point the code crashes. The simplest solution is to run the code again with option Activate_continue = 1, to continue the simulation from the last reached point. This was a very rare occurrence, but I attempted to made it even rarer by adding more significant figures to the assortativity in the names of the intermediate .dat files (before it was rounded to 1 decimal place). I am expecting to have solved this issue, but the reader should still be aware of the potential issue.

Generating data for the model mapping:

Everything related to data is self-contained in folder "data". Once files to be used for the model mapping code or the stochastic simulation are generated, they need to be manually copied in the right folder where they can be used. The subfolder "saved_data" contains the data files used in generating the results for the paper. They were created without specifying the seed for the random number generator, so generating new files will not produce exactly the same results as in the "saved_data" folder.

To generate data for <u>Great Britain</u>:

1. Open *MyGBdata.xlsx*. This contains a copy of Table *C0844.xls* (which comes exactly as given by the 2001 UK census data, and is available in "data/saved-data") in the first few sheets. Go to the last sheet and save it as *GB_H_structure_ModelMapping.txt.*

2. Run *Generate_pop_for_sim.m* to create:
   a. *GB_H_structure_ModelMapping_sim_e5.txt*
   b. *H_GB_5.dat*
   c. *W_GB_5.dat*

   Note that these files come from a Monte Carlo simulation, so will show some stochastic variability compared to the ones used for the original results.
3. Copy all 4 files (.txt and .dat, not the .xlsx ones) in “code-model-mapping”, but mind you might override the ones already present (however, the original ones can always be recovered from “data/saved-data”). The two .dat files will also be needed if you want to reproduce the results for the simulation dynamics of Figure 2B, so are also needed in folder “code-figures/simulation-dynamics”.

To generate data for Sierra Leone:
1. Ensure you obtain the file *SLPR51FL.SAV* from the DHS website (free registration – FILE NOT AVAILABLE HERE: it is in the “Household Member Recode” dataset SLPR51SV.ZIP at https://www.dhsprogram.com/data/dataset/Sierra-Leone_Standard-DHS_2008.cfm)
2. Transform *SLPR51FL.SAV* in a .csv file using the R template file *DHSconvert.R*.
3. Open the .csv file in Excel and save it as an .xlsx file.
4. Run *ReadDHShhdata.m* with option: country = 'SL'. This generates *SL_H_structure_ModelMapping.txt.*
5. Run *Generate_pop_for_sim.m* with option: country = 'SL' to create:
   a. *SL_H_structure_ModelMapping_sim_e5.txt*
   b. *H_SL_5.dat*
   c. *W_SL_5.dat*

   Note that these files come from a Monte Carlo simulation, so will show some stochastic variability compared to the ones used for the original results.
6. Copy all 4 files (.txt and .dat, not the .xlsx ones) in “code-model-mapping”, but mind you might override the ones already present (however, the original ones can always be recovered from “data/saved-data”).

To generate data for South Africa:
1. Ensure you obtain the file *ZAPR31FL.SAV* from the DHS website (free registration – FILE NOT AVAILABLE HERE: it is in the “Household Member Recode” dataset ZAPR31SV.ZIP at https://www.dhsprogram.com/data/dataset/South-Africa_Standard-DHS_1998.cfm)
2. Transform *ZAPR31FL.SAV* in a .csv file using the R template file *DHSconvert.R*.
3. Open the csv file in Excel and save it as an .xlsx file.
4. Run *ReadDHShhdata.m* with option: country = 'SA'. This generates *SA_H_structure_ModelMapping.txt.*
5. Run *Generate_pop_for_sim.m* with option: country = 'SA' to create:
   d. *SA_H_structure_ModelMapping_sim_e5.txt*
   e. *H_SA_5.dat*
   f. *W_SA_5.dat*

   Note that these files come from a Monte Carlo simulation, so will show some stochastic variability compared to the ones used for the original results.
6. Copy all 4 files (.txt and .dat, not the .xlsx ones) in “code-model-mapping”, but mind you might override the ones already present (however, the original ones can always be recovered from “data/saved-data”).

Generate figures

General structural options (“switches”):

- Activate_save_fig:
  - 0: Figures are generated by MatLab, but not saved in a computer file.
  - 1: Figures are saved by MatLab in the correct subfolder of “output-figures”.
- Activate_plot_from_new_workspaces:

- 0: Default: figures are created by using the workspaces saved in “saved-workspaces” (i.e. the ones generated by the authors and used for the figures in the paper).
- 1: Figures are created using the workspaces in “output-workspaces” (which need to be filled in by the user).

The analysis of the rule of thumb (Figure 4) uses “saved-rule-of-thumb” and “output-rule-of-thumb”, instead of “saved-figures” and “output-figures”.

The only exception to the location of saved and newly computed intermediate files is for Figure 2 (right-hand side) of the main text, where newly run simulations appear in folder “code-figures/simulation-dynamics” and the saved simulations in “code-figures/simulation-dynamics/saved-simulation-dynamics”.

Figure 1 (main text)

1. Run code *MainFig1_outputs.m* from folder “code-figures”.

Optional:

2. To generate again the required workspaces (in this case, only the one for the baseline scenario), run the model mapping code with: country = GB; pop = 2ran; R0 = 1; \phi = 1. Use option Activate_plot_from_new_workspaces = 1.

Figure 2 (main text)

1. Run code *MainFig2left_2x2_intersection.m* from folder “code-figures”;
2. Run code *MainFig2right_2x2_FullDyn.m* from folder “code-figures”, with options:
    a. Run_simulation_anyway = 0;
    b. Activate_plot_from_new_simulations = 0.
3. Use the *CreateFig2.pptx* file in “saved-figures/main/PowerPoint-join-figures” to piece together:
    a. *Main_Figure2left.pdf* (created in point 1 above)
    b. *Main_Figure2right.pdf* (created in point 2 above)
    c. *Main_Figure2legend.pdf*, created by hand in LegendFig2.pptx (available in “saved-figures/main/PowerPoint-join-figures”)

Note: Pasting pdf files in PowerPoint and save figures as pdf files seems possible only in MS PowerPoint 2011 for Mac, but not in PowerPoint with MS Office Professional Plus 2010 on Windows.

Optional:

4. To generate again the required simulations, run code *MainFig2right_2x2_FullDyn.m* from folder “code-figures”, with options:
    a. Run_simulation_anyway = 0 (or 1, if you want to override user-made previous simulations)
    b. Activate_plot_from_new_simulations = 1.
5. To generate again the executable file to run the stochastic simulations,
    a. In Mac, load the Xcode project in “stochastic-simulation/ModelMapping_Mac”, comment out line 19 (no need to invoke function *print_final_size()*) and uncomment line 20 (invoke *print_real_time()*) in file *print_output.cpp*. Then build the executable file again and copy it in folder “code-figures\simulation-dynamics”, by changing its name to *ModelMapping_Mac_dyn*.
    b. In Windows, load the Xcode project in “stochastic-simulation/ModelMapping_Win”, comment out line 19 (no need to invoke function *print_final_size()*) and uncomment line 20 (invoke *print_real_time()*) in file *print_output.cpp*. Then build the executable file again and copy it in folder “code-figures\simulation-dynamics”, by changing its name to *ModelMapping_Win_dyn*.

Figure 3 (main text)

1. Run code *MainFig3left_3x3_OAR_grid.m* from folder “code-figures”;
2. Run code *MainFig3right_3x3_OAR.m* from folder “code-figures”;
3. Use the *CreateFig2.pptx* file in “saved-figures/main/PowerPoint-join-figures” to piece together:
    a. *Main_Figure3left.pdf* (created in point 1 above)
    b. *Main_Figure3right.pdf* (created in point 2 above)

Optional:

4. To generate again the required workspaces run the model mapping code with all required parameter combinations and then run again the figure files with option Activate_plot_from_new_workspaces = 1.

Figure 4 (main text)

1. Run code *MainFig4_1x2_ROT_line_2pop.m* from folder “code-figures”.

Optional:

2. To generate again the required workspaces for the rule of thumb (to appear in folder “output-rule-of-thumb” but already available in folder “saved-rule-of-thumb”), run code *Analyse_data_for_RuleOfThumb.m* from folder “code-figures” with all the required parameter combinations (note that running them for country = SL, is computationally expensive). Then run again *MainFig4_1x2_ROT_line_2pop.m* with Activate_plot_from_new_ROT_analysis = 1.
3. To generate again the workspaces used by the code *Analyse_data_for_RuleOfThumb.m*, run first the model mapping code with all required parameter combinations and then code *Analyse_data_for_RuleOfThumb.m* with option Activate_plot_from_new_workspaces = 1.

Supplementary Tables 1-9

These tables are essentially constructed by hand, or using Excel and then converted from Excel to latex using “excel2latex” (package available online).

Supplementary Tables 10-13

1. Run code *Model_Mapping_code_for_SAR.m* from folder “code-other-analyses”, for \phi = 1, 1.5 and 2, and pop = ‘2ran’ and ‘UK’.
2. This creates .csv files in folder “output-workspaces/GB/SAR”
3. Each of them should be copied on the correct line in the tables already available in files *Tables_for_SAR_and_Fh_2ran.xls* and *Tables_for_SAR_and_Fh_UK.xls.* Both are available in “saved-workspaces/GB/SAR”.
4. Tables are then converted from Excel to latex using “excel2latex” (package available online). It is convenient to transform the table line by line (excel2latex seems very computationally intensive).

Supplementary Figure 1

Run *SuppFig_inf_profile.m* from folder “code-figures”.

Supplementary Figure 2

Run *SuppFig_pAA_VS_Rh.m* from folder “code-figures”.

Supplementary Figure 3

1. Run *SuppFig_find_v.m* from folder "code-figures", with option Activate_plot_from_new_workspaces = 0.

Optional:

2. To generate again the required workspaces, run *Model_Mapping_code_for_v_plots.m* from "code-other-analyses" with options:
    a. pAA_min = 0.5;
    b. pAA_max = 0.5;
    c. dpAA = 0.5;
    d. psiG_vec = [0.2, 0.5, 0.8, 0.99, 1, 1.01, 1.1, 1.5]; psiG_vec_lab = '_psiGcustom';
3. Then run again *SuppFig_find_v.m* from folder "code-figures", with option Activate_plot_from_new_workspaces = 1.

Supplementary Figure 4

Run *SuppFig_3x3_heatmap.m* from folder "code-figures", with which_output = 'vc', 'vhc', 'SAR' and 'Fh', in sequence (and corresponding letters). Other options are: country = 'GB'; popfig = '2ran'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 5

Run *SuppFig_2x2_mix.m* from folder "code-figures", first with output = 'z' and then with output = 'pi'. In both cases, run in sequence through the value of R0 = 1.5, 2 and 4, choosing manually the correct first_subletter (a, e and I for the first output, m, q and u for the second one). In all cases, popfig = '2ran'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 6

Like Supplementary Figure 5, but first with output = 't' and then output = 'sz'.

Supplementary Figure 7

Run *SuppFig_2x2_intersection.m* from folder "code-figures", in sequence through the value of R0 = 1.5, 2 and 4, and for \phi = 1 and 2. It is sufficient to comment/uncomment the lines of code in sequence (which takes care of the panel letter) in the "5% threshold - variable phi" block of code. Other options are: country = 'GB'; popfig = '2ran'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 8

Run *SuppFig_2x2_intersection.m* from folder "code-figures", in sequence through the value of R0 = 1.5, 2 and 4, and for tolval = 0.01 and 0.1. It is sufficient to comment/uncomment the lines of code in sequence (which takes care of the panel letter) in the "phi = 1 - variable threshold" block of code. Other options are: country = 'GB'; popfig = '2ran'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 9

Run *SuppFig_3x3_OAR_ROT.m* from folder "code-figures", with:

- use_match_r = 0;
- use_intermediate = 0;

- use_log2psi = 0.

In sequence, try different tolerances by uncommenting lines (66-68):

- tolval = 0.01; figletter = 'A';
- tolval = 0.05; figletter = 'B';
- tolval = 0.1; figletter = 'C';

(no need to change figletter, here). Other options are: country = 'GB'; popfig = '2ran'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 10

Run *SuppFig_3x3_heatmap.m* from folder "code-figures", with options, in sequence:

- popfig = '2ran' + (which_output = 'ass'; figletter = 'A') for the left-hand side;
- popfig = 'UK' + (which_output = 'ass'; figletter = 'B') for the right-hand side;

In both cases, use country = 'GB'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 11-16

Like Supplementary Figures 4-9, but with popfig = 'UK'.

Supplementary Figure 17

Run *SuppFig_3x3_OAR_ROT.m* from folder "code-figures", with:

- use_match_r = 0;
- use_intermediate = 1;
- use_log2psi = 0.

In sequence, uncomment lines 39-42 for popfig = 'm4r', 'm4UK', 'm5r' and 'm5UK'.

Optional: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 18

Run *SuppFig_3x3_OAR_ROT.m* from folder "code-figures", with options: use_match_r = 1.
Manually change the figletter depending on whether popfig = '2ran' or 'UK'.

Optional: generate required workspaces with *Model_Mapping_code_match_r.m* from "code-other-analyses".

Supplementary Figure 19

Run *SuppFig_3x3_OAR_ROT.m* from folder "code-figures", with options: use_log2psi = 1.
Manually change the figletter depending on whether popfig = '2ran' or 'UK'.

Optional: generate required workspaces with *Model_Mapping_code_log2psi.m* from "code-other-analyses".

Supplementary Figure 20

Like Supplementary Figure 10, but with country = 'SL'.

Supplementary Figure 21

Run *SuppFig_3x3_OAR_ROT.m* from folder "code-figures", with:

- use_match_r = 0;
- use_intermediate = 0;

- use_log2psi = 0.

Use country = ‘SL’ and, in sequence:

- popfig = ‘2ran;
- popfig = ‘UK’.

<u>Optional</u>: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 22

Like Supplementary Figures 10 and 20, but with country = ‘SA’.

Supplementary Figure 23

Like Supplementary Figure 21, but with country = ‘SA’.

Supplementary Figure 24

Run *SuppFig_3x3_ROT_table.m* from folder “code-figures”. Use country = ‘GB’ and, in sequence:

- popfig = ‘2ran + figletter = ‘A’
- popfig = ‘UK’ + figletter = ‘B’.

<u>Optional</u>: generate required workspaces with *Model_Mapping_code.m*.

Supplementary Figure 25

1. Run *SuppFig_2x3_ROT_line.m* from folder “code-figures” for tolval = 0.01, 0.05 and 0.1, to generate subfigures A, B and C.
2. Run SuppFig_1x1_metaROT.m from folder “code-figures” to generate subfigure D.