Global Model Structure

This is a primer to learn the basic operation of the model. There are basically three things that you will want to do with the model: (1) calibrate, (2) run projection scenarios, and (3) plot results.

For calibration, the highest-level file is **Run\_ABC.m**. It loads model parameters and calibration data for a specified country (given by its unique ISO code) and initiates the ABC-SMC calibration routine. To load model parameters, **Load\_Priors.m** is called. This function creates prior distributions for model parameters and saves them as a country-specific .mat file. To load Calibration data, **Load\_Cal\_Data.m** is called. This function reads calibration data from data sheets and organizes it into a country-specific file.

**ABC1.m** is the brains of the ABC-SMC calibration algorithm. It works by iteratively sampling from prior distributions, perturbing parameters, running simulations, and filtering out poorly performing sample sets. To measure performance of a sample set, an ODE system is simulated. This happens a few steps downstream of here, but first **Distance.m** is called. It also calls **Get\_Parameters.m** to turn the sampled parameters into formats the model will use. Returns Diff, a measurement of goodness of fit.

**Distance.m** runs an ODE simulation using ode23(), organizes outputs, and calculates a goodness of fit comparing model trajectories to calibration data. The system of differential equations used by ode23() are written in **Ukraine\_model.m**.

Once calibration is complete (up to the user: max time is exceeded, tolerance is sufficiently low, max number of iterations exceeded, etc), it is time to run projection scenarios. A projection scenario would be, for example, “scale up OST enrollment until coverage hits 40% in 2030.” The top-level file for this is **ABC\_Combine\_Results\_iterative.m**, which saves a few files containing model outputs for each scenario and for each sample set k from 1 through N. The individual “k” files are saved in **Results\_Disability\_concatenated.m**, and are then collated in **ABC\_Combine\_Results\_iterative.m**. For the example scenario described above, which we’ll call “scale\_OST”, we generate an OST scale-up factor that results in 40% coverage by 2030 for each sample set. This is handled by **get\_scale\_OST.m**, which runs a nonlinear least-squares optimization. Once the optimal scale-up factors are known, a final simulation is performed by **model\_proj.m**, where output trajectories are also generated.

Once the projections code has been run, you can plot results. For this, simply use **ABC\_plot\_john\_projections\_duration.m**. Additionally, to generate projection statistics like % HIV reduction, use **ABC\_Get\_HIV\_reduction\_combined\_CI.m**.

INDEX

**ABC1.m**

% ABC1.m is the brains of the ABC-SMC calibration algorithm. It works by

% iteratively sampling from prior distributions, perturbing parameters,

% running simulations, and filtering out poorly performing sample sets.

%

% Arguments:

% priors | prior distributions

% N | number of sample sets

% stopping\_criteria | stops the calibration when a condition is met

% alpha | determines how well a sample set must perform to pass on

% sigma\_scale | perturbation parameter

% seed | seed for random number generator

% Tol\_length | used for running independent calibrations for different

% calibration parameters

% ISO | unique country ID

**ABC\_plot\_john\_projections\_duration.m**

% ABC1.m is the brains of the ABC-SMC calibration algorithm. It works by

% iteratively sampling from prior distributions, perturbing parameters,

% running simulations, and filtering out poorly performing sample sets.

%

% Arguments:

% priors | prior distributions

% N | number of sample sets

% stopping\_criteria | stops the calibration when a condition is met

% alpha | determines how well a sample set must perform to pass on

% sigma\_scale | perturbation parameter

% seed | seed for random number generator

% Tol\_length | used for running independent calibrations for different

% calibration parameters

% ISO | unique country ID

% duration | total duration of the ABC run to use for outputs (e.g. 12 to output results after 12hrs)

**ABC\_plot.m**

% top level function for plotting results in matlab

**Age\_calculator.m**

%%% This function takes data for average age of onset of injecting from

%%% Hines (2020) and outputs average age of young PWID (<25) at onset of

%%% injecting, and proportion of new injectors younger than 25. To do this,

%%% we assume a triangular distribution constructed of the central estimate

%%% and upper/lower bounds.

**ciplot.m** (not ours)

% Plots a shaded region on a graph between specified lower and upper confidence intervals (L and U).

% l and u must be vectors of the same length.

% Uses the 'fill' function, not 'area'. Therefore multiple shaded plots

% can be overlayed without a problem. Make them transparent for total visibility.

% x data can be specified, otherwise plots against index values.

% colour can be specified (eg 'k'). Defaults to blue.

**ABC\_Combine\_Results\_iterative.m**

% Top-level file for running projections. For each scenario, for each sample set k from 1 to

% N, Combine\_Results calls Results, which saves a .mat file containing model outputs and

% intervention scale-up factors. These are stored in the Results folder.

% Combine\_Results combines all of these files into consolidated files.

%

% Arguments

% ABC\_filename | ABC calibration filename

% ISO | unique country code

% numrun: number of ABC runs to combine (1 if only one run)

% pp | parallel pool option (1=on, 0=off)

**Death\_calculator2.m**

% This function calculates the Estimates and Lower/Upper intervals from the Calibration

% parameter file ‘death\_params.csv’

% Arguments

% ISO | unique country code

**discrete\_sample.m**

% independently draws n samples (with replacement) from the

% distribution specified by p, where p is a probability array

% whose elements sum to 1.

**Distance.m**

% This is the wrapper function for Distance3()\*. It also calls Get\_Parameters()

% to turn the sampled parameters into formats the model will use. Returns

% Diff, a measurement of goodness of fit

%

% arguments

% samp\_params | posterior parameter samples

% ISO | unique country code

% t | generation

% it | number of "tries" for ABC1 to produce N good sample sets

% r | sample set number during the "it"th try

% tol | ABC tolerance

% \* the Distance3()function calculates the "difference", or goodness of fit, between

% the model trajectories and the calibration data, for a single parameter sample

% set. Distance3() simulates the ODE model by calling ode23.

%

% arguments

% params | model parameters

% Data | calibration data

%

% returns

% model | a struct containing model outputs

% Diff | scalar goodness of fit score

% Diff\_vec | categorized goodness of fit scores

**Get\_ART\_HR\_prison**

% This function returns ART, OST, and NSP availability in prison for the

% country given by ISO.

**Get\_end\_dates**

% This function returns the ART end (AKA interruption) dates, if they

% exist, for the country given by ISO.

**ABC\_Get\_HIV\_reduction\_combined\_CI.m**

%%% run this code to calculate % reduction in HIV incidence from 2021 to

%%% 2030 under OST scale-up and ART scale-up scenarios (each scenario

%%% scales OST or ART to 50% coverage by 2030)

%

% args

% ISO | unique country code

% dir | directory where projection results are saved. This will look like

% "Results/..."

% numrun: number of ABC runs to combine (1 if only one run)

**Get\_Initial\_Conditions.m**

% This function generates initial conditions to start a model simulation.

%

% arguments

% params | model parameters

% model | either "Calibration" or "Projections"

**Get\_Parameters.m**

% This function takes a set of parameter samples (x) and organizes and/or

% operates on them to return parameters (params2) used in the model.

%

% args

% x | sampled parameters

% ISO | unique country code

% model | either 'Projections' or 'Calibration'

%

% returns

% params2 | parameters formatted for use in the model

**Get\_scale\_ART.m**

% This function solves for the optimal ART scale-up factor to achieve 81%

% coverage in 2030

%

% args

% params | model parameters

% SS | steady state initial conditions

**Get\_scale\_OST.m**

% This function solves for the optimal OST scale-up factor to achieve 40%

% coverage in 2030

%

% args

% params | model parameters

% SS | steady state initial conditions

**Get\_seed\_date.m**

% This function returns HIV seed date (AKA HIV epidemic start year) for a

% country given by ISO

**Get\_start\_dates.m**

% This function returns HIV seed date (AKA HIV epidemic start year) for a

% country given by ISO

**Load\_Cal\_Data.m**

% This function reads calibration data from data sheets and organizes it

% into a country-specific file

**Load\_Priors.m**

% This function creates prior distributions for model parameters and saves

% them as a country-specific .mat file.

**make\_lognormal\_dist.m**

% This function creates a lognormal distribution from mean and lower/upper

% confidence intervals. If given an ISO and variable name, function will

% load distribution parameters from data sheet. If given a numerical array

% containing distribution parameters, function uses these.

**make\_normal\_dist.m**

% This function creates a normal distribution from mean and lower/upper

% confidence intervals. If given an ISO and variable name, function will

% load distribution parameters from data sheet. If given a numerical array

% containing distribution parameters, function uses these.

**make\_triangular\_dist.m**

% This function creates a triangular distribution from support limits and peak location.

% If given an ISO and variable name, function will

% load distribution parameters from data sheet. If given a numerical array

% containing distribution parameters, function uses these.

**make\_uniform\_dist.m**

% This function creates a uniform distribution from lower/upper bounds.

% If given an ISO and variable name, function will

% load distribution parameters from data sheet. If given a numerical array

% containing distribution parameters, function uses these.

**model\_proj.m**

% this function generates model outputs including prevalence, incidence,

% coverage, and burden of disease metrics. It runs an ode simulation, then

% calculates output quantities from model trajectories.

%

% args

% params | model parameters

% Initial | initial conditions

% Scenario | 0 for status quo, 1 for ost scale-up, 2 for art scale-up

% Disability\_weights | for burden of disease calculations

% Disability\_weights\_PWID | for burden of disease calculations

% Disability\_weights\_HIV

%

% returns

% model | struct containing outputs

**prior\_post\_histogram\_combined.m**

% This function generates histograms of prior and posterior distributions

% of model parameters

%

% args

% N | number of sample sets

% posteriors | (discrete) posterior distributions

% priors | (continuous) prior distributions

% ISO | unique country code

% scenario | 0=status quo, 1=ost scale-up, 2=art scale-up

**Prop\_sexual.m**

% This function uses generates Estimates and upper/lower bounds from the calibration file

% PWID\_prop\_HIV\_sexual\_data.csv

**Results\_Disability\_concatenated.m**

% This function is called by ABC\_Combine\_Results. For each scenario, Results

% saves a .mat file containing model outputs and intervention scale-up

% factors.

%

% args

% k | sample set number

% ABC\_filename | ABC calibration file

% ISO | unique country ID

% dir | directory for saving results

% Note: editied to load ABC concatenated filenames from the ./ABC\_concatenated/{counttrycode}/ folder. If only one run, just run ABC\_concatenate.m above with numrun=1.

% also load Disability file renamed with {countrycode} in its name to allow multiple countries to run in parallel on the server. The Disability file remains the same for all countries.

**Run\_ABC.m**

% This is the top-level file for running calibrations. It loads model

% parameters and calibration data and initates the ABC-SMC routine (ABC1).

%

% args

% ISO | unique country code

% pp | parallel pool option (1=on, 0=off)

**Ukraine\_model.m**

% This function contains the system of differential equations for disease

% and population dynamics. It is called from an ode solver. Note the model

% features that are used/unused in each "model" option: Initialization,

% Calibration, and Projections.

%

% args

% t | current timepoint

% y\_tmp | state vector

% params | model parameters

% model | 'Initialization', 'Calibration', or 'Projections'

% scenario | 0=status quo, 1=ost scale-up, 2=art scale-up

%

% returns

% ydot2 | time derivative of state vector, appended by some extra outputs