LTBIscreening: Running the full model

Nathan Green

2019-01-18

Introduction

This document explains how to run the cost-effectiveness model in the `LTBIscreeningproject` R package. There are a large number of files in this package but many of them are not essential to running the model. They are for model checking or plotting of outputs. Scripts are in `./scripts/` and functions are in `./R/`.

Top-level

The model running consists of scripts (which can be `sourced`) and functions (which are available when the package is `loaded`). The highest level script is `000-programme-level-params-scenario-runner.R` This simply loads required packages, input values and then runs the model using `run_model()`.

Before this can be done though, we need to prepare the input values using `create_input_workspace.R`.

Create input workspace

There are 2 terms used to define a model run. They are ordered in terms of if and how they are modified between model runs.

- `intervention`These are the higher-level simulation settings. Some of these are alway kept fixed and others can be varied. Any given set of intervention values we say define a `policy`. These are discrete options.
  - `N.mc`: Number of Monte Carlo samples; Default 1.
  - `no_students`: TRUE/FALSE; default: FALSE.
  - `force_everyone_stays`: TRUE/FALSE; default: FALSE.
  - `screen_with_delay`: Rather than screen *everyone* on entry screen at random 0-5 years from entry.
  - `MAX_SCREEN_DELAY`: 5 years after entry to country.
  - `FUP_MAX_YEAR`: Time horizon for active TB progression; default 100 years.
  - `screen_age_range`: e.g. 18-35 years old.
  - `year_cohort`: 2012 is most recent complete year; largest cohort, corresponds with Pareek () LTBI risk.
  - `min_screen_length_of_stay`: Default 0 i.e. everyone included.
  - `discount_rate`: 3.5%
  - `incidence_list`: which incidence by country of origin groups to target for LTBI screening. WHO categories per 100,000 are $[0, 50), [50, 150), [150, 250), [250, 350), > 350$
  - `endpoint`: calculate QALYs and costs including those that exit EWNI or not i.e. time horizon
  - `LTBI_test`: type of test QFT, QFT-plus, TSPOT.TB
  - `treatment`: 6 months or 3 month LTBI treatment
- `scenario`These are varied within `policies`. These can be discrete, deterministic values or defined distibutionally. We don't vary any health state utilities.
  - `p`: screening pathway branch probabilities
  - `cost`: screening pathway per individual costs

Therefore, the script `create_input_workspace.R`

- Load raw cohort data `051206 - IMPUTED_sample.RData`. This is from Aldridge() in Lancet.

- Create and save `policy` data using `data-prep_policies.R`. This is a list of different model run inputs created using `create_and_save_policies()`.
- Create list of `intervention` parameter values using `interv_constructor()`.
- Create cost and QALY input lists `unit_costs.RData`, `cost_effectiveness_params.RData`, `synthetic_cohort_params.RData` using `01b-data-prep_cost-effectiveness.R`.
- Create list of dataframes each representing a scenario using `create_and_save_scenarios()` in `01-data-prep_scenario.R`. This reads from an Excel workbook.
- Clean the raw individual level data using `01c-data-prep_modelling.R` and save as `sample_cleaned.RData`. This basically remove individuals with inconsistent event times or missing data.
- This is the main working script in the data prep stage. A TB progression curve is estimated in `active-TB-extrapolation.R`. This is used in `04a_3-include-new-tb-events.R` to extrapolate the times to TB progression for the total sample, under other event contraints. We can also now estimate the QALYs gained for disease-free, case fatality and cured for each patient.

`run_model()` and `run_policy()`

`run_model()` is a wrapper for `run_policy()`, iterating over all policies. `run_policy()` has these main steps:

1. Set-up

1. Define output folders, with `setup_folders()`
2. The specific policy and intervention values are loaded in to the workspace and the cohort modified accordingly in `data-prep_constants-policy.R`, using `policy_interv()` and `policy_cohort()`
3. The particular intervention and policy probability, cost and health state value are substitutes in to the screening decision tree using `prep-decisiontree.R`.

2. Modelling

4. `parallel_decision_tree()` is the parallelised wrapper around the cost-effectiveness calculations for a decision tree `decision_tree_cluster()`.
5. The output of `decision_tree_cluster()` includes the probability of an LTBI individual being successfully treated to cured. This is used in `activetb_qaly_cost()` which calculates the population QALYs and costs due to active TB.
6. The costs and QALYs from the decision tree model and the population model are combined to give a total cost-effectiveness using `combine_popmod_dectree_res()`.

3. Post-processing

7. The combined output data are plotted using `plots_and_tables_scenarios()`.