

Quick start: using the invitation algorithm

Adam Brentnall

August 22, 2022

Program version 2.0

1 Introduction

This document describes the basic steps needed to run the algorithm to generate invitation schedules using the demonstration site.

2 Software file structure

The software distributed has the following files / directories in the root directory.

- `docs` documentation (including this file)
- `inputgen` code to generate input files for the invitation algorithm (R)
- `prog` code to run the invitation schedule algorithm (python3)
- `README.md` Readme file about the software

3 Generating lists, overview of process

We next describe how to use the software to generate invitation lists.

1. Create one or two csv files to generate input files for the algorithm. Examples are provided in `inputgen/updatefiles`. They are:
 - GP list data (csv file)
 - Bookings update data (csv file; only needed if wave 2+)

Demonstration files are included with the program.

2. Then generate the algorithm input files by running the R script in `inputgen`
3. Before running the algorithm to generate the list you should consider what uptake you expect. Also you need to consider how many slots are available and what round of invitations are being planned. Once these are decided you need to let the program know, either by editing the configuration file directly in `./prog/input/runpars.ini` using a text editor and saving, or through running the program using `./prog/trial-invitation-frontend.py` rather than the command-line `./prog/trial-invitation-nogui.py`
4. Run the algorithm
5. The stratified sample list (accounting for national opt-outs) is saved in `./prog/output/nhsd-invite-inflated.csv`

4 Detailed descriptions

4.1 Generating input files for the algorithm (inputgen)

The algorithm needs a number of inputs in order to prioritise selection of GP/age groups, inflate the request to take into account national data opt outs, and tell the algorithm capacity and number of bookings so far in a format it can understand. The R script in `./inputgen` does this. To run the script one first needs to include your one or two CSV files in `./inputgen/updatefiles`

- GP list data (csv file)
- Bookings update data (csv file)

Then, how to call the R script depends on the wave of invitations, because the second wave onwards will use data on bookings to date; the first wave will not.

To call the R script for a first round

```
Rscript trial-invitation-geninput.r <path-to-file/gplist-file.csv>
```

To call the script for a second round one needs to append the bookings file.

```
Rscript trial-invitation-geninput.r <path-to-file/gplist-file.csv> <path-to-file/update-file.csv>
```

4.2 Updating capacity, uptake assumption and round

The algorithm uses a configuration file in `./prog/input/runpars.ini` where parameters for number of slots available, assumed overall uptake and the current wave of invitations are kept. You may edit this file directly using a text editor, or use the user interface if calling the program using `trial-invitation-frontend.py` rather than `trial-invitation-nogui.py`. If you choose to edit directly then number of slots is at the top of `./prog/input/runpars.ini`:

```
[number_slots]  
n = 1000
```

Set this to be the number of available slots **minus the number of slots already booked but not linked to GPs already invited** (wave 2+ only).

Uptake correction:

```
[uptake]  
upt = 1.0
```

This adjusts the assumption for uptake overall. The assumption for uptake overall is based on a lookup table that allows it to vary by age/sex/GP. If there is evidence (eg. from the first round of invites) that uptake is not well calibrated for the current site, then this parameter lets the user change it by a constant, the same for all age/sex/GPs. For example, if uptake in the first wave was double expected, set this to be 2.0. If it was half, set it to be 0.5. Proportion of slots left to fill:

```
[invitation.target] btarget = 0.5
```

We used 50% for the first round, but this might be changed with this parameter. This parameter is only active if `[invitation.round]` `r` = 0. If `r` = 1 then default percentages based on the wave are used, as defined in `config.ini` below.

4.3 Running the algorithm

To run the algorithm you can use the program

- `trial-invitation-nogui.py` (no graphical interface - assumes you edited `runpars.ini` above) or

- `trial-invitation-frontend.py` (graphical interface - could edit or not edit `runpars.ini` in advance. If you haven't edited it then enter the correct values as reviewed above).

These are run simply by calling from python without additional arguments. eg.

```
> python3 trial-invitation-nogui.py.
```

As it runs you will see a lot of output on the screen. If you see

```
!!! Not feasible !!! -> check constraints by varying input
```

then there has been a problem running the code and you may need to update the program constraints (next).

4.4 Program constraints

The program makes sure certain constraints are met, which are described in the technical documentation. If the program cannot meet these constraints you may relax them to get a schedule. The file which controls the constraints is in `./prog/config.ini`.

invbound

If `r =` in `runpars.ini` then the program will use a pre-set for the round to determine the target number of bookings (ordered by wave here). ie. It is 0.5 for rounds 1 and 2, and 1.1 for rounds 3 or 4:

```
[invbound]
b = 0.5,0.5,1.1,1.1
```

maxgp

This controls how many patients from a GP may be invited in any round, currently set to 50%.

```
[maxgp]
g = 0.5
```

If the program fails in later rounds relaxing this constraint may help. You could try changing to eg. 0.6, 0.7, 0.8, until the program runs.

propmen

This is the targeted proportion of men who book (50%). This is unlikely to ever be changed, but might be needed if uptake overall to the trial is starting to be imbalanced by sex, due to poor calibration of the uptake model. If this is the case the uptake model should be re-calibrated, but also in order to achieve balance again it may be necessary to over invite the group that is under-represented. This parameter allows this possibility.

```
[propmen]
e = 0.50
```

uptakeagesex

This is the minimum proportion in each age/sex group. Unlikely to be needed to change, but to relax make the values lower

```
[uptakeagesex]
d = 0.035, 0.04, 0.05, 0.06, 0.06, 0.03, 0.035, 0.04, 0.05, 0.06, 0.06, 0.03
```

advcancer

This parameter controls how much the program shifts the booking distribution towards higher risk (likely older) people based on the primary outcome event rate. If you reduce this parameter, then it will be easier for the program to find a solution. Again, if constraints cannot be met it might be decreased in small increments until a feasible solution is found.

[advcancer]

cstar = 0.0003

4.5 Troubleshooting - summary

If the program does not run, first inspect the input files (in `./prog/input`) to look for anomalies. Then consider relaxing constraints, particularly,

1. Relax **g**=0.5 by increasing in small intervals to 1.0.
2. Relax **cstar** by decreasing in small intervals.

In general, if you have a long list of GPs then you should not have problems with constraints.