**Instructions for Dexter**

Running the simulations for varying uptake parameter:

1. Change the ‘p\_case\_detection’ (this is the uptake parameter; it gives the probability of case detection in each year of intervention) in the ‘global settings’ chunk to the required uptake.
2. In the ‘run simulations’ chunk, we run the simulation (using the ‘BIA\_simul()’ function) for each of the different strategies.
3. In the ‘produce results tables’ chunk, we then combine the results into two different data frames: ‘all\_overall’ has more general results (we haven’t discussed using these yet but might be useful later) then ‘all\_results’ has all the results we need for calculating the budget impact.
4. Also in the ‘produce results tables’ chunk, we run the bia\_table() function for each strategy. This calculates the budget impact of the strategy compared to the baseline scenario of no case detection, S10. Results are then combined into ‘bia\_all\_long’.
5. Save ‘all\_overall’ and ‘all\_results’ and ‘bia\_all\_long’ as .Rda files (or in another format if you prefer), making sure you name it something that tells you what uptake parameter the results correspond to.
6. Repeat steps 1-5 for varying uptake parameters.
7. You will then want to combine all of the ‘all\_results’ tables for varying uptake parameters into one larger table and combine all of the ‘bia\_all\_long’ tables into one table. You may need to add a new column for uptake parameter. These are the data frames that the Shiny app should be pulling from.

To produce the plots for a given uptake parameter:

1. Filter the combined ‘bia\_all\_long’ table by the required uptake parameter.
2. The ‘budget impact results’ chunk then gives you all the code you need to produce the plots (what I used in the paper is also in the ‘budget impact results’ chunk but you’re more than welcome to come up with your own).

How to adapt results for varying cost inputs:

1. For different cost inputs, you need to use the combined ‘all\_results’ table, filter for the required uptake parameter, and recalculate the cost groups. In the ‘Adapting output for changing costs’ chunk I give an example of how to calculate the cost subgroups for one strategy and using the unit costs in the paper. You will need to work out how to adapt this for use in the app with different cost inputs but hopefully it’s a starting point.
2. Once costs are recalculated, the bia\_table() function will need re-running for the new costs and bia\_all\_long will need to be reproduced.
3. Then follow the steps to make the plots.