**Bladder Cancer Screening Model Code Check**

December 2022

*Main\_model*

* Looks fine. Not able to fully check the code relating to running on HPC. Assume this will be updated when want to run different scenarios e.g. screening vs no screening.

*Load\_all\_files*

* Row 16: why is risk\_age in the model? This was used in MIMIC\_bowel to determine when someone would get risk scored. This isn’t happening in bladder model, so should be removed (or renamed if used for a different purpose).
* Row 26: This isn’t about the code, but I had a look at the cancer mortality data in the model and it is unclear to me why mortality rate increases for all stages between year 2-5, then increases again (at a much lower rate) between year 6-10. This doesn’t seem plausible particularly when mortality after year 10 is assumed to be 0. This doesn’t happen for CRC mortality rates which all decrease with year since diagnosis (although in some cases year 2 is higher than year 1 mortality, which could make sense if initial treatment keeps some people alive). I can see for simplicity assuming a constant rate of mortality between year 2-5 and a different (lower one) between year 6-10 – have done this myself in other projects and it is a reasonable assumption to make when the data for each year is not available, but if you are doing something more complex than that, then would expect a reducing rate of mortality. I noted there was nothing in the word document about calculating mortality from survival.
* Row 42: This is processing the mortality data is it not rather than loading it (just for clarity in the commenting)?
* Row 101: I notice that you have generate\_random here and again in the Main\_model at line 82. Is one of them redundant? Normally it would differ for each PSA loop, so probably shouldn’t be in the Load\_all\_files script.

*Set\_params*

* In general I would keep this script for the functions directly related to setting parameters (the generate\_parameters one is OK here too as related), and put the other functions inside different scripts appropriately labelled. Some of the other functions seem more related to updating population in each simulation cycle or assigning health states to population at model start, so it would make sense to group them within a different script/multiple scripts appropriately labelled e.g. setup\_pop and update\_pop.
* Row 6 and Row 22 and Row 43: I’m wondering why these functions seem to provide info for each person separately rather than working in a vectorised way. Vectorisation will speed them up. It should be simple to create a vector output rather than a single one for the first function and a matrix output rather than a vector one for the second function, instead of using a loop. I notice in Main\_model you say you are replacing to use apply or map, but much better to vectorise instead as apply will be no faster than loop (main reason it is used is just for neater code, but I struggle to understand/use it properly) and vectorising is consistent with rest of model.
* Row 22: These arguments are very long. For the purposes of a function best to use shorter ones, and then specify what they are when the function is called. If the same arguments are being used every time you call the function, then they probably don’t need to be included as arguments at all. If they only differ by PSA/calibration run, then I would suggest specifying them as parameters in the set\_parameters function, so the current value for that PSA run is already specified in the global environment. In fact, seeing that the f.stage.assign function which calls this function doesn’t include those things as arguments, means that the arguments are not needed in this function either.
* Row 43: This function seems a bit redundant as just does a little bit more than f.stage, using just one set of f.stage arguments. I would just combine them into one with both bits of code, unless you need f.stage elsewhere without f.stage.assign (but I don’t think you do).
* Row 149: Is p.onset\_sex basically a RR? It has been treated as such in this equation. If it is that is fine, but I would suggest relabelling it RR.sex to make this clear (a relative risk is very different from a probability of onset, which is more like an absolute risk). Whilst the actual param value for this needs calibrating, I would suggest using the lifetime incidence differences as both the prior and calibration target (maybe this is what the 1.25 value you have for this param already represents) and I don’t see a problem in calibrating it together with the other RRs. For the sake of consistency I would also suggest if you want to, putting the age effect (which I assume is also a RR for each increasing year of age so should be labelled appropriately) in this equation, as in total this represents the personal risk. I remember you saying in the report that the age effect wasn’t linear. This is fine, although it would be good to label the parameter appropriately if it is a transformed variable.
* Row 152: Here P.onset represents I assume a calibrated absolute risk (rather than a RR). This is fine, I just want to check that the labelling of params is consistent (in this case, the param has a good name for an absolute risk). Ideally this equation would just multiply absolute risk by relative risk. I would also suggest labelling it (row 151) probability of onset of BC at time t, to make it clear that it is an onset probability for the current year that is being produced and not (for example) a time to onset – the return text in line 132 is also slightly confusing in this way.
* Row 176: You appear to be using rbinom to generate your random numbers for smoking cessation rather than pre-specifying them in m.rand. I would strongly advise against using random numbers that are not pre-specified within the simulation loops as things will end up changing between intervention and control i.e. different people will stop smoking. This means you have introduced extra uncertainty that shouldn’t be there and could affect the stability of your results. I don’t see any reason why this can’t be done in the usual way we use to decide randomly who has events.

*Natural\_history*

* Row 18: I don’t think this row should be needed. Instead it should be possible to calculate (not calibrate) RRs for all three smoking groups based on your existing RRs for current/former smokers and the HSE population prevalence of smoking. This could be done outside the model, just using the HSE population to get the parameter values. Then in Row 19 you can just adjust so that the no\_smoke group get their RR (lower than 1) applied to the original TP.OC values too. The adjusted RRs for the other two groups will each become smaller than currently. It is not much different than what we do normally in the risk.calc function for RRs with cancer risk.
* Row 81: I assume that the parameter P.onset\_low.risk represents the proportion of total cancers that are low grade rather than high grade at onset. I was just wondering how the value of 0.192 fits with your low grade incidence data, which seems to show a similar incidence for high grade and low grade. Are you calibrating the P.onset\_low.risk parameter? I would actually expect a lot more at onset than incidence given that it would be expected that a lot wouldn’t get diagnosed (I assume). I went back to the report and noted there that there was no info on where this param value had come from, and it doesn’t say it is calibrated.

*Sympt*

* Rows 18-19: I don’t think there is an error, but I would find it easier to understand if the two eligibility criteria were specified in a similar way e.g. if both used m.M but ==3 for BC and ==2 for LG\_BC that would be more logical. Is there any reason why you can’t do that? Also, why does one criteria use BC\_diag in m.Diag but the other doesn’t use LG\_BC\_diag to be equivalent?
* Row 25: I was wondering why you are multiplying two probabilities, one of which is to the power of year since onset, but I think I see now. There wasn’t much about this in the report, but I assume both the probabilities are calibrated, and you are assuming that the time relationship is not linear, hence the pre-specified form of this equation. Maybe my maths is wrong, but won’t this result in probabilities getting smaller with increasing time if P.sympt.diag\_B is a probability under 1? I thought probability of diagnosis would get bigger with increasing time? Should the relationship be ^(1/yr\_onset) instead? You may find that you don’t need P.sympt.diag A, but I suppose in this case it will become 1. Are you constraining the calibration so both values are under 1 (will need to make sure of this as would be wrong to have probabilities above 1)? I see in your parameters sheet that you actually have P.sympt.diag\_B as a value of 1.2, which means the relationship you have does result in bigger values for more years after diagnosis. But it seems like this could risk ending up with probabilities above 1 when combined with P.sympt.diag\_A in some situations – for example if someone has had cancer for 20 years, then you will end up multiplying P.sympt.diag\_A by 38.3, so unless it is very small this will go above 1 – your value for it is 0.1, so in this case you end up with probability of 3.8 for diagnosis – makes no sense! In fact what it means with the parameter values you are currently using is that diagnosis becomes definite 10 years after onset. I’m not sure this is a good starting place for calibration unless you have evidence this is so. I would suggest trying to constrain the probability parameters to be under 1 to avoid this – it means it will never happen and your calibration should work (and no one will ever definitely be diagnosed after a certain number of years since onset).
* Row 28: This appears to be modifying risk of symptomatic diagnosis for everyone, not just for people aged above 80. Is this meant to be the case? As you have it the equation would result in smaller decrements for people aged a lot of years above 80 than only two years above 80 which I think isn’t right – I think like above it might be ^(1/age-80) that you want instead. Also, for people aged under 80 it is likely to be producing decrements that are above 1 even with the suggested change in equation. I think you need to add a condition that it is only done in people aged above 80.
* Row 34: Just double checking that a single parameter for time to LG diagnosis is suitable – fine if you have decided it is.
* Row 47: Wondering why you need the extra condition of multiplying new\_diag\_LG by m.State[, “BC\_LG”]. Isn’t it the case that everyone with a new diagnosis will be in that state anyway due to the eligibility conditions (m.M[ ,t+1]==2) set in row 19 and then acted on in row 34? If redundant I would suggest removing it for ease of code review.

*Run\_simulation*

* Rows 59-70 + 79: I am feeling that maybe there is slightly too much code in here that would be better moved to functions. This all affects m.Diag so could be moved to a different function called something like f.can\_onset maybe. The idea is to try to streamline run\_simulation so it contains a series of different high level updating tasks that are clearly described. Each of these tasks should be short and separate and self-contained (so only updating one of the matrices).
* Row 67: You have rbinom in here that is introducing extra uncertainty between arms, rather than using m.Rand to supply your pre-specified random numbers. This should be removed and random sampling of who has recurrence done in the usual way based on m.Rand.
* Rows 70 & 79: Just for neatness I would put these just after row 59 to 62.

*OS\_helper*

* Don’t know what this does but assume related to the model running on HPC, so will ignore.

**Model Running**

* Model runs OK (I didn’t include the bits for multiple cores/HPC).
* Checked outputs for one model run (m.Diag, m.M and m.M\_8) and they seem to make logical sense (although obviously need correct parameters for proper validation). Might be useful in testing mode to also output m.State just to check progression of states over time is logical, and of course m.Out when that is working.