LTBI Bayesian MPES with PREDICT data

Nathan Green (Imperial College London)

14/02/2020

Introduction

We will fit the MPES model detailed elsewhere to a PREDICT study data extract.

Analysis in R

We used vague normal priors for the LTBI prevalence and TB progression. A prior sensitivity analysis is given at the end of this document.

The data have been grouped so that they are sufficiently coarse for anonymisation and relevant for the DES model.

```r
library(readr)
library(R2jags)
library(R2WinBUGS)
library(purrr)
library(dplyr)
library(forcats)
```

```r
data0 <- read_csv(here::here("raw-data", "aggregated_data.csv"),
                  col_types = list(X1 = col_integer(),
                                   age_grp  = col_factor(levels =
                                                           c("(15,35]",
                                                             "(35,55]",
                                                             "(55,100]")),
                                   sex  = col_factor(levels =
                                                       c("Female",
                                                         "Male")),
                                   ethnicity  = col_factor(levels =
                                                             c("White",
                                                               "Black African or Caribbean",
                                                               "South Asian",
                                                               "Other")),
                                   inc_cob_participant2 = col_factor(levels =
                                                                       c('<40',
                                                                         '41-100',
                                                                         '100-300',
                                                                         '>300')),
                                   yearssinceentry_grp = col_factor(levels =
                                                                      c('(0,5]',
                                                                        '(5,10]',
                                                                        '(10,100]')),
                                   prevbcg = col_factor(levels =
                                                          c("No", "Yes")),
                                   reasonforscreening = col_factor(levels =
                                                                     c("Contact",
                                                                       "Migrant")),
                                   pop = col_double(),
                                   tb = col_double(),
                                   ltbi = col_double()))
```

```
# inspect the raw relationships in the data
library(scales)

data0 <-
  data0 %>%
  mutate(ltbi_prev = round(ltbi/pop, 3),
         prog_rate = round(tb/ltbi, 3))

# ltbi vs tb counts by grouping
# add some jitter
plot(data0$ltbi + rnorm(nrow(data0),0,0.1), data0$tb + rnorm(nrow(data0),0,0.1),
     col = alpha(as.numeric(data0$reasonforscreening), 0.2), pch = 19,
     xlab = "LTBI", ylab = "TB", main = "reason for screening")

legend("bottomright", legend = c("Contact", "Migrant"),
       col = 1:2, pch = 19, bty = "n")

hist(data0$ltbi_prev[data0$pop > 20],
     breaks = 30,
     main = "",
     xlab = "LTBI prevalence (sample size over 20 people)")
```

```
head(data0)
```

```
## # A tibble: 6 x 11
##      X1 age_grp sex   ethnicity inc_cob_partici~ yearssinceentry~ prevbcg
##   <int> <fct>   <fct> <fct>     <fct>            <fct>            <fct>
## 1     1 (15,35] Fema~ Black Af~ <40              (0,5]            <NA>
## 2     2 (15,35] Fema~ Black Af~ <40              (0,5]            No
## 3     3 (15,35] Fema~ Black Af~ <40              (0,5]            Yes
## 4     4 (15,35] Fema~ Black Af~ <40              (10,100]         Yes
## 5     5 (15,35] Fema~ Black Af~ <40              (10,100]         Yes
## 6     6 (15,35] Fema~ Black Af~ <40              (5,10]           <NA>
## # ... with 4 more variables: reasonforscreening <fct>, pop <dbl>,
## #   tb <dbl>, ltbi <dbl>
```

```
# dat <- data0[1:100, ]

## aggregate some groups
dat <-
  data0 %>%
  mutate(
    inc_cob_participant2 = fct_explicit_na(inc_cob_participant2),
    yearssinceentry_grp  = fct_explicit_na(yearssinceentry_grp),
    prevbcg = fct_explicit_na(prevbcg)) %>%
  group_by(age_grp,
           ethnicity,
           inc_cob_participant2,
           reasonforscreening,
           prevbcg,
           yearssinceentry_grp) %>%
  summarise(pop = sum(pop),
            tb  = sum(tb),
            ltbi = sum(ltbi))

save(dat, file = "dat.RData")
```

```
# stratified model

jags_dat_input <-
  list(
    len_gp = nrow(dat),  # number of groups
    Xm  = dat$pop,       # number of migrants/contacts
    Xp  = dat$ltbi,      # number of positive test results
    Xtb = dat$tb         # number of observed active tb cases
  )

jags_dat_input

params <-
  c("sens", "spec",
    "lambda",
    "p_latent")

BUGS_file_name <- "BUGS_code_Xl_fn.txt"
```

```
# regression model

# transform to levels to integers
# aggregate by covariates of interest
dat_regn <-
  dat %>%
  mutate(
    rfs = as.numeric(reasonforscreening) - 1,
    inc = as.numeric(inc_cob_participant2),
    eth = as.numeric(ethnicity),
    age = as.numeric(age_grp),
    bcg = as.numeric(prevbcg),
    yse = as.numeric(yearssinceentry_grp)
  ) %>%  # transform to level integer
  group_by(rfs,
           inc,
           eth,
           age,
           bcg,
           yse
  ) %>%
  summarise(pop = sum(pop),
            tb  = sum(tb),
            ltbi = sum(ltbi))
```

```r
##TODO:
# remove error row
# why is this causing an error?
# pop = ltbi = tb = 1
# maybe its because its prob = 1?
dat_regn <- dat_regn[-235, ]
```

BUGS inputs.

```r
# create list of input data
jags_dat_input <-
  list(
    len_gp = nrow(dat_regn),  # number of groups
    len_rfs = max(dat_regn$rfs) + 1,
    len_inc = max(dat_regn$inc),
    len_eth = max(dat_regn$eth),
    len_age = max(dat_regn$age),
    len_bcg = max(dat_regn$bcg),
    len_yse = max(dat_regn$yse),
    rfs = dat_regn$rfs,
    inc = dat_regn$inc,
    eth = dat_regn$eth,
    age = dat_regn$age,
    bcg = dat_regn$bcg,
    yse = dat_regn$yse,
    Xm  = dat_regn$pop,       # number of migrants/contacts
    Xp  = dat_regn$ltbi,      # number of positive test results
    Xtb = dat_regn$tb         # number of observed active tb cases
  )

jags_dat_input
```

```
## $len_gp
## [1] 586
##
## $len_rfs
## [1] 2
##
## $len_inc
## [1] 5
##
## $len_eth
## [1] 4
##
## $len_age
## [1] 3
##
## $len_bcg
## [1] 3
##
## $len_yse
## [1] 4
```

```
##
## $rfs
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [176] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [211] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [246] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [281] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [316] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [351] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [386] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [421] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [456] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [491] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [526] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [561] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## $inc
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [106] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3
## [141] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [176] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [211] 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [246] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [281] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [316] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [351] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [386] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3
## [421] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [456] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [491] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [526] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
## [561] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
##
## $eth
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
##  [36] 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [71] 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3
## [106] 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 2 2 2 2 2
## [141] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [176] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [211] 4 4 4 4 4 4 4 4 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [246] 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [281] 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [316] 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3
## [351] 3 3 3 3 3 3 3 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
## [386] 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1
## [421] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3
## [456] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4
## [491] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [526] 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
## [561] 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4
##
## $age
##   [1] 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 1 1 1 1 2
##  [36] 2 2 3 3 3 1 1 1 1 1 1 1 1 1 1 2 3 3 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
##  [71] 2 3 3 3 3 1 1 1 1 1 1 1 1 2 2 2 2 3 1 1 1 1 1 1 1 1 2 2 2 3 3 3 1 1 2
## [106] 2 3 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 1 1 1 1 1 2 2 2 1 1 1 1 1
## [141] 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
## [176] 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
## [211] 2 3 3 3 3 3 3 3 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3
## [246] 2 2 2 2 3 3 1 1 1 1 1 1 1 2 2 2 2 3 3 3 3 1 1 1 1 1 1 1 1 1 2 2 2 2 2
## [281] 3 3 3 1 1 1 1 1 1 1 1 1 1 2 2 2 2 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
## [316] 3 3 1 1 1 1 1 1 1 1 1 1 2 2 2 3 3 3 1 1 1 1 1 2 2 2 3 3 1 1 1 1 2 2 1
## [351] 1 1 1 2 2 2 3 1 1 1 1 2 2 1 1 1 1 1 1 2 2 2 2 2 3 3 1 1 1 1 1 2 2 2 2
## [386] 3 3 1 2 2 2 2 3 3 3 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 1
## [421] 3 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1
## [456] 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 2 2 2 2
## [491] 2 2 2 2 2 2 3 3 3 3 3 3 3 3 1 1 1 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3
## [526] 1 1 2 2 2 2 3 3 1 1 1 1 1 1 2 2 2 3 3 1 1 1 1 2 2 3 1 1 1 1 1 1 2 2 2
## [561] 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 1 1 1 2 2
##
## $bcg
##   [1] 1 1 1 1 2 2 2 2 3 3 3 1 1 2 2 2 2 3 3 3 1 2 2 3 3 1 1 2 2 2 2 3 3 3 2
##  [36] 2 3 1 2 3 1 1 1 2 2 2 3 3 3 3 3 1 3 1 1 1 2 2 2 2 3 3 3 1 2 2 2 2 3 3
##  [71] 3 1 1 2 3 1 1 2 2 2 3 3 3 2 2 2 3 2 1 1 1 2 2 2 3 3 1 2 2 1 2 2 2 2 2
## [106] 3 2 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 1 2 2 3 1 2 2 2 3 2 2 3 1 1 1 2 2
## [141] 2 2 3 3 3 1 1 1 2 2 2 2 3 3 1 2 2 2 3 3 1 1 1 1 2 2 2 2 3 3 3 3 1 1 1
## [176] 2 2 2 2 3 3 3 1 1 2 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 2 3 3 3
## [211] 3 1 1 2 2 2 3 3 2 2 3 1 2 2 1 1 2 2 2 3 3 3 3 1 2 2 2 2 3 3 1 2 2 2 3
## [246] 1 2 2 3 2 3 1 1 2 2 2 3 3 2 2 2 3 1 2 3 3 1 1 2 2 2 2 3 3 3 1 2 2 3 3
## [281] 1 2 3 1 1 1 2 2 2 2 3 3 3 1 2 2 3 2 2 3 1 1 1 1 2 2 2 2 3 3 3 3 1 2 3
## [316] 2 3 1 1 1 2 2 2 2 3 3 3 1 2 2 2 3 3 1 1 2 2 3 1 2 2 2 3 1 1 2 2 2 2 1
## [351] 1 2 3 1 2 2 2 1 2 2 3 1 2 1 1 2 2 3 3 1 2 2 2 3 1 2 1 1 2 2 3 2 2 3 3
## [386] 2 3 2 1 2 2 3 1 2 3 1 1 2 2 2 3 3 3 3 1 1 1 2 2 2 2 3 3 3 2 2 3 3 3 3
## [421] 2 1 1 2 2 2 2 3 3 3 1 2 2 2 3 3 3 1 2 2 2 3 3 1 1 1 1 2 2 2 2 3 3 3 3
## [456] 1 1 1 2 2 2 2 3 3 3 3 1 1 1 2 2 2 2 3 3 3 3 1 1 1 2 2 2 3 3 3 1 1 1 2
## [491] 2 2 2 3 3 3 1 1 2 2 2 3 3 3 1 2 3 1 1 2 1 1 2 2 2 3 3 3 1 1 2 2 2 3 2
## [526] 1 2 1 2 3 3 2 3 1 1 2 2 2 3 1 2 2 2 3 1 2 2 3 2 3 2 1 2 2 2 3 3 1 2 3
## [561] 3 1 1 1 1 2 2 2 3 3 1 2 2 2 3 3 1 2 2 3 3 1 1 2 1 2
##
## $yse
##   [1] 1 2 3 4 1 2 3 4 1 2 3 1 3 1 2 3 4 1 2 4 3 1 3 2 3 2 3 1 2 3 4 1 2 4 2
##  [36] 3 3 3 3 3 2 3 4 1 2 3 1 2 3 4 3 3 3 1 2 3 1 2 3 4 1 2 3 3 1 2 3 4 1 2
##  [71] 3 1 3 3 3 1 2 1 2 3 1 2 3 1 2 3 3 3 1 2 3 1 2 3 1 2 3 2 3 3 2 3 1 2 3
## [106] 3 3 1 3 1 2 3 1 2 3 1 2 3 1 2 3 2 3 1 2 3 3 3 1 2 3 3 2 3 2 1 2 3 1 2
## [141] 3 4 1 2 3 1 2 3 1 2 3 4 1 3 3 1 2 3 2 3 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3
## [176] 1 2 3 4 1 2 3 1 3 1 2 3 4 1 3 4 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 4 1 2 3
## [211] 4 2 3 1 3 4 1 3 2 3 1 3 2 3 1 2 1 2 3 1 2 3 4 1 1 2 3 4 2 3 2 1 2 3 3
## [246] 3 3 4 3 3 3 1 2 1 2 3 1 2 1 2 3 3 2 3 1 3 1 4 1 2 3 4 1 2 4 4 3 4 3 4
## [281] 4 4 4 1 3 4 1 2 3 4 1 3 4 4 3 4 4 3 4 3 1 2 3 4 1 2 3 4 1 2 3 4 4 4 4
## [316] 4 4 1 2 4 1 2 3 4 1 2 4 4 3 4 4 3 4 1 2 1 2 1 2 2 3 1 1 1 3 2 3 1 3 1
## [351] 3 1 1 3 1 3 4 1 1 2 1 1 3 1 2 1 2 1 2 1 1 2 3 1 1 2 1 2 1 2 1 1 2 1 2
## [386] 3 2 1 1 1 3 3 3 3 3 1 2 1 2 3 1 2 3 4 1 2 4 1 2 3 4 1 2 3 1 3 1 2 3 1
## [421] 1 1 2 1 2 3 4 1 2 3 1 1 2 3 1 2 3 1 1 2 3 3 4 1 2 3 4 1 2 3 4 1 2 3 4
## [456] 1 2 3 1 2 3 4 1 2 3 4 1 2 3 1 2 3 4 1 2 3 4 1 2 3 1 2 3 1 2 3 1 2 3 1
## [491] 2 3 4 1 2 3 1 3 1 2 3 1 2 3 1 1 1 1 3 1 1 2 1 2 4 1 2 3 1 3 1 2 3 1 1
## [526] 2 3 2 3 1 3 3 3 1 2 1 2 4 1 1 1 3 1 1 4 2 4 4 4 4 4 4 1 3 4 1 4 1 4 1
## [561] 4 1 2 3 4 1 2 4 1 4 4 1 3 4 3 4 4 3 4 3 4 1 4 4 4 4
##
## $Xm
##   [1]  12   7   1   2  33  27  22   2   9   5   6   3   6   5  14  39   1
##  [18]   2   3   1   2   1  16   1   6   1   2   6   8  19   3   2   2   1
##  [35]   3  22   1   4  26   6   1   2   1   3   4   3   1   2   1   1   1
##  [52]   1   2   4   3   1   5  10   9   1   4   1   2   1   3   3  20   1
##  [69]   1   1   1   1   1   5   2   3   1  27  16   4   4   1   1   9   5
##  [86]  12   1   2   2   4   2  13   8   5   4   2   1   4  12   1   1   2
## [103]   1   1   2   1   5   2   4  32  18  18   4   4   4   1   1   2   2
## [120]   8  23   1   5   1   3   7   1   1   2   2   1   1   3   5   1   6
## [137]   4   9  30  34  84   1   3   4   5   2   2   9  10  24  91   1   2
## [154]   8   2   1   6  11   1   2  64  19  12   3 276 140 103  10  58  31
## [171]  16   1   7   8  14  40  71 205   3   5   7  36   1   6   5   8 108
## [188]   3   1  32   1  11   5   4  73  40  32   9   1   6   1   1   4  14
## [205]  17  45   1   1   4   4   1   2   2   4  11   1   1   5   1   1   1
## [222]   1   1   1   2   2  14  18  28   5   2   4   1   1   5   8  43   1
## [239]   1   7   1   1   2  11   1   2   6   1   2   4   4   1   1  10  10
## [256]   5   1   2   2   7  11   3   1   6   1   1   2  47   5   6   2 174
## [273]   3   2  16  24   1 140   2  18  12  56   8   2   5  27  11   7   9
## [290]  93   3   3  20   4   1  47  10   2   2   2  10   7   1  36  45  21
## [307]   8 285  10   2   2  44   2  48   3   3   1   5   3  27  12   6   4
## [324] 118   2   2  18   2   1  31   2   1   1   3   1   5   1   3   1   2
## [341]   3   1   1   2   1   1   1   1   1   1   1   2   1   1   1   1   1
## [358]   1   3   1   2   1   1  14   3  78  10  11   3   5  23   6   2   4
## [375]   1   1   1   1  18   1  13   1   1   1   1   1   1   3   1   2   3
## [392]   1   3   5   3  17   1  64   4   1  12   1   1   1   2   1   1  24
## [409]   4   3   1   1   1   1   4   4   1   2   4   2   1  18   4  76   5
## [426]   5   2  17   4   1   7  22   4   7   3   2   1   2   8   2   3   1
## [443]   1 397  32  11   6 882  82  35  20 274  26   7   4  57   5  41 227
## [460]  52 209  12  52   8  72   2  14   3 114  31   6 366   9   9   1 143
## [477]   5  15   2   4  31   6   3   8   1   1   2   1   4   9   5  12   1
## [494]   5   2   2   5   2   2   3   1   2   1   3   1   2   1   1   1   1
## [511]  10   1  40   3   1   9   5   1   2   1  15   2   4   4   1   1   1
## [528]   2   2   2   4  24   2   1   1  14   1   1   2   1   4   4   1   1
## [545]   3   1   6   1   5   1   4   1   4   1   5   2   1   1   2   1   2
## [562]   3   1   1   9  11   3  42   2   5   3   1   1  23   2   2   2   4
## [579]   9   1   3   2   6   5   1   3
##
## $Xp
##   [1]   1   0   0   0   6   1   3   0   2   2   1   1   1   4   7   9   0
##  [18]   1   0   0   0   0   5   0   1   0   1   1   1   5   0   0   0   1
##  [35]   1   3   1   0   6   0   1   1   1   0   0   0   0   1   0   0   0
##  [52]   1   1   0   0   0   0   4   1   1   2   0   1   0   1   1   4   0
##  [69]   0   0   0   1   0   0   0   1   1   7   5   3   2   0   0   3   4
##  [86]   4   0   0   1   3   0   3   5   0   1   0   0   1   5   0   1   1
## [103]   0   1   0   0   2   0   0   4   5   3   0   1   0   1   0   1   1
## [120]   4   9   0   3   0   2   3   0   0   1   0   0   1   0   1   1   1
## [137]   1   1   7  11  22   0   0   3   3   1   0   5   2  13  43   0   1
## [154]   5   2   0   4   5   1   0  24   7   1   1  66  34  13   2  15  13
## [171]   7   0   5   3   5  19  31  63   2   1   1  15   0   3   2   5  33
## [188]   0   1  14   1   5   3   1  28  12   5   3   0   2   0   1   0  11
## [205]   7  22   0   1   3   2   1   0   1   4   6   0   0   3   0   0   0
## [222]   0   0   1   1   0   4   3  10   2   2   1   0   1   3   4  16   0
## [239]   1   3   0   0   2   6   0   0   1   0   1   1   2   1   0   2   4
## [256]   3   0   1   1   0   3   1   1   4   0   0   2   7   5   5   1  27
## [273]   2   2   6   6   0  18   0   5   4  11   5   2   5   6  10   6   9
## [290]  17   3   3   3   0   1   8   2   0   1   1   9   6   1   8  43  17
## [307]   7  44   9   2   2   9   0   7   2   1   1   5   3   6  12   6   2
## [324]  20   2   2   1   0   1   5   1   0   0   0   0   0   0   0   0   0
## [341]   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## [358]   0   0   0   0   0   0   1   0  13   2   2   0   3  10   3   1   2
## [375]   0   0   1   0   6   0   4   0   0   1   1   1   0   0   0   2   1
## [392]   0   0   1   0   3   0   7   1   0   1   0   0   0   0   0   0   6
## [409]   1   0   0   0   0   0   1   2   0   0   0   0   1   6   1  28   2
## [426]   2   0   4   2   0   2  10   1   1   2   1   1   0   3   0   2   1
## [443]   1  98   8   2   2 220  18   6   7  69   4   0   1  18   3  10  84
## [460]  14  48   1  16   3  15   1   8   1  20  15   2  79   1   0   0  40
## [477]   0   5   1   2   5   1   0   5   1   0   1   0   2   4   1   4   0
## [494]   4   1   2   2   2   1   1   0   2   0   1   0   1   0   0   0   0
## [511]   3   0  10   1   0   4   1   1   2   0   3   1   1   3   0   0   0
## [528]   0   0   0   1   3   0   1   1   4   0   1   0   0   1   1   0   1
## [545]   0   0   0   0   1   0   0   0   1   1   2   2   0   1   0   1   0
## [562]   2   1   1   2   8   3   6   2   1   0   0   1   3   0   0   0   2
## [579]   2   0   2   2   1   0   0   0
## 
## $Xtb
##   [1]  0  0  0  0  0  0  1  0  0  0  1  0  0  1  1  2  0  0  0  0  0  0  0
##  [24]  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##  [47]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##  [70]  0  0  0  0  1  0  0  0  1  0  0  0  0  0  1  1  0  0  0  0  0  0  0
##  [93]  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0
## [116]  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
## [139]  0  2  2  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  1
## [162]  0  0  0 10  9  0  0  0  1  1  0  0  0  1  1  2  3  0  0  0  1  0  0
## [185]  0  0  1  0  0  0  0  0  0  0  2  0  0  1  0  0  0  0  0  1  0  0  0
## [208]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
## [231]  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## [254]  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  1
## [277]  0  1  0  1  0  1  0  0  0  3  0  1  0  1  0  0  0  0  0  1  0  0  0
## [300]  0  0  0  0  1  1  1  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [323]  0  3  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [346]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [369]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [392]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [415]  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## [438]  0  0  0  0  0  0  5  0  0  1 15  0  0  1  2  1  0  0  1  0  0  3  2
## [461]  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
## [484]  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [507]  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
## [530]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [553]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
## [576]  0  0  0  0  0  0  0  0  0  0  0
```

```r
save(jags_dat_input, file = "jags_dat_input.RData")

params <-
  c("sens", "spec",                 # test performance
    "lambda",                       # progression proportion
    "ppred",
    "pred_Xtb", "pred_X_latent",    # posterior predictive distns
    "prior_Xtb", "prior_X_latent"   # prior posterior distns
    # "p_contact", "p_migrant"      # marginalised
  )

BUGS_file_name <- "BUGS_code_regn.txt"
```

```r
# n_iter <- 1e6
# n_burnin <- 1e3
# n_thin <- 1e2

n_iter <- 1000#0
n_burnin <- 10#0
n_thin <- 1
```

The BUGS code is

```
## LTBI screening evidence synthesis model ----

model {

  for (j in 1:len_gp) {

      X_latent[j] <- trunc(p_latent[j] * Xm[j])            # functional relationship

      p_pos[j] <- (p_latent[j]*sens) + (1 - p_latent[j])*(1 - spec)

      Xp[j] ~ dbin(p_pos[j], Xm[j])

      Xtb[j] ~ dbin(lambda, X_latent[j])                   # time independent

      logit(p_latent[j]) <- alpha + betas*rfs[j] + beta_inc[inc[j]] + beta_eth[eth[j]] +
                            beta_age[age[j]] + beta_bcg[bcg[j]] + beta_yse[yse[j]]
    }

  ## prior distributions for inference

  alpha ~ dnorm(0, 0.368)
  betas ~ dnorm(0, 0.368)

  #beta_inc[1] = 0           # set as baseline
  for (i in 1:len_inc) {
    beta_inc[i] ~ dnorm(0, 0.368)
  }

  for (i in 1:len_eth) {
    beta_eth[i] ~ dnorm(0, 0.368)
```

```
}

for (i in 1:len_age) {
  beta_age[i] ~ dnorm(0, 0.368)
}

for (i in 1:len_bcg) {
  beta_bcg[i] ~ dnorm(0, 0.368)
}

for (i in 1:len_yse) {
  beta_yse[i] ~ dnorm(0, 0.368)
}

sens ~ dbeta(100, 5)   # good: mean~=0.9
spec ~ dbeta(100, 5)
lambda ~ dbeta(5, 100) # mean~= 0.1

## priors used in prior predictive distn

prior_alpha ~ dnorm(0, 0.368)
prior_s ~ dnorm(0, 0.368)
for (i in 1:len_inc) {
  prior_inc[i] ~ dnorm(0, 0.368)
}
for (i in 1:len_eth) {
  prior_eth[i] ~ dnorm(0, 0.368)
}
for (i in 1:len_age) {
  prior_age[i] ~ dnorm(0, 0.368)
}
for (i in 1:len_bcg) {
  prior_bcg[i] ~ dnorm(0, 0.368)
}
for (i in 1:len_yse) {
  prior_yse[i] ~ dnorm(0, 0.368)
}
prior_lambda ~ dbeta(5, 100)

## prior predictive distn
## for sample covariates
##TODO: include positivity, sens, spec
for (j in 1:len_gp) {
  logit(prior_ltbi[j]) <- prior_alpha + prior_s*rfs[j] + prior_inc[inc[j]] + prior_eth[eth[j]] +
                          prior_age[age[j]] + prior_bcg[bcg[j]] + prior_yse[yse[j]]
  prior_X_latent[j] <- trunc(prior_ltbi[j] * Xm[j])
  prior_Xtb[j] ~ dbin(prior_lambda, prior_X_latent[j])
}


## posterior predictions

## complete grid of covariate values
for (i in 1:len_rfs) {
    for (k in 1:len_inc) {
      for (m in 1:len_eth) {
        for (a in 1:len_age) {
          for (b in 1:len_bcg) {
            for (y in 1:len_yse) {
              logit(ppred[i,k,m,a,b,y]) <- alpha + betas*i + beta_inc[k] + beta_eth[m] +
                                            beta_age[a] + beta_bcg[b] + beta_yse[y]
            }
          }
        }
      }
    }
  }

  ## for sample covariates
  for (j in 1:len_gp) {
    logit(ppred_grp[j]) <- alpha + betas*rfs[j] + beta_inc[inc[j]] + beta_eth[eth[j]] +
                            beta_age[age[j]] + beta_bcg[bcg[j]] + beta_yse[yse[j]]
    pred_X_latent[j] <- trunc(ppred_grp[j] * Xm[j])
    pred_Xtb[j] ~ dbin(lambda, pred_X_latent[j])
  }

  ## posterior predictive distn
  #c <- 1   # pick an individual to replicate; ##TODO: loop over whole sample?
  #pred_X_latent <- trunc(ppred[rfs[c] + 1, inc[c], eth[c], age[c], bcg[c], yse[c]] * Xm[c])

  ## marginalise
  ##TODO: how to do this?
  #p_contact <- sum(ppred[1, , , , , ])
  #p_migrant <- 1 - p_contact

}
```

```r
out <- jags(jags_dat_input,
            parameters.to.save = params,
            model.file = file_loc,
            n.chains = 2,
            n.iter = n_iter,
            n.burnin = n_burnin,
            n.thin = n_thin,
            DIC = TRUE,
            working.directory = here::here("scripts"),
            progress.bar = "text")

save(out, file = "../../data output/out_regn.RData")
```

References