Fit Stan model to TN and St. Louis data

Chris Hoover

6/1/2020

```
TN_all <- readRDS("../TN/Data/TN_all_2020-06-01.RDS")
TN_nsh <- readRDS("../TN/Data/TN_counties_2020-06-01.RDS") %>%
  filter(COUNTY == "Davidson")
StL <- read_xlsx("Data/StLMay20-adjusted priors.xlsx", sheet = "Data",
                 skip = 8)
```

```
start.date <- as.Date("2020-01-23")
end.date <- as.Date("2020-07-01")

stl_inputs <- list()

# the number of age groups
  stl_inputs[['nage']] = 1

# the population for each age group
  stl_inputs[['npop']] = array(400000)
# the fraction of hospitalized cases (ICU + non-ICU)
  stl_inputs[['frac_hosp']] = array(0.07)
# the fraction of ICU cases
  stl_inputs[['frac_icu']] = array(0.02)
# the death rate of infected
  stl_inputs[['frac_mort']] = array(0.01)
# the fraction of asymptomatic cases
  stl_inputs[['frac_asym']] = array(0.178)

# a parameter modifying the prior uncertainty in the fractions above (set to 1000.0 for `nage` > 1)
  stl_inputs[['alpha_multiplier']] = 100.0

# mean duration in "exposed" stage
  stl_inputs[['mu_duration_lat']] = 5.0
# standard deviation (sd) of duration in "exposed" stage
  stl_inputs[['sigma_duration_lat']] = 2.0
# mean duration in "infectious" stage for asymptomatic cases
  stl_inputs[['mu_duration_rec_asym']] = 7.0
# sd of duration in "infectious" stage for asymptomatic cases
  stl_inputs[['sigma_duration_rec_asym']] = 5.0
# mean duration in "infectious" stage for mild cases
  stl_inputs[['mu_duration_rec_mild']] = 7.0
# sd of duration in "infectious" stage for mild cases
  stl_inputs[['sigma_duration_rec_mild']] = 5.0
# mean duration in "infectious" stage for hospitalized cases
  stl_inputs[['mu_duration_pre_hosp']] = 5.0
# sd of duration in "infectious" stage for hospitalized cases
  stl_inputs[['sigma_duration_pre_hosp']] = 1.0
# mean duration in hospital for non-ICU cases
  stl_inputs[['mu_duration_hosp_mod']] = 12.0
# sd of duration in hospital for non-ICU cases
```

```
  stl_inputs[['sigma_duration_hosp_mod']] = 2.0
# mean duration in hospital for ICU cases
  stl_inputs[['mu_duration_hosp_icu']] = 12.0
# sd of duration in hospital for ICU cases
  stl_inputs[['sigma_duration_hosp_icu']] = 2.0

# lambda parameter for initial conditions of "exposed"
  stl_inputs[['lambda_ini_exposed']] = 0.3

# mean initial beta estimate
  stl_inputs[['mu_beta1']] = 0.2
# sd initial beta estimate
  stl_inputs[['sigma_beta1']] = 0.02

# distance in time between the knots used to construct the splines
  stl_inputs[['dknot']] = 10

# spline mode (must be 1 or 2)
# splinemode = 1: estimate beta up to today
# splinemode = 2: estimate beta up to dknot days before today, then assume constant value up to today
  stl_inputs[['splinemode']] = 1

# number of interventions
  stl_inputs[['ninter']] = 1

# length of each intervention
  stl_inputs[['len_inter']] = array(10)
# mean change in beta through intervention
  stl_inputs[['mu_beta_inter']] = array(1.0)
# sd change in beta through intervention
  stl_inputs[['sigma_beta_inter']] = array(0.5)


tn_inputs <- stl_inputs
nsh_inputs <- stl_inputs

#Pop of TN
tn_inputs[["npop"]] <- array(6829174)

#Pop of Davidson county
nsh_inputs[["npop"]] <- array(694144)
```

Fit to StL data

```
# Timeframe for simulation. Can replace Sys.Date() with any date at which to cut off model fitting
  itoday_1based = Sys.Date()-start.date

# specify the date up to when beta is estimated
  stl_inputs[['itoday']] = as.numeric(itoday_1based)

# the number of days to run the model for
```

```r
  stl_inputs[['nt']] = as.numeric(as.Date(end.date) - as.Date(start.date))

# Put data into list
  stl_inputs[['nhobs']] <- nrow(StL)
  stl_inputs[['thobs']] <- as.numeric(as.Date(StL$date) - start.date)
  stl_inputs[['obs_Hmod']] <- StL$hosp.lower/0.9

# start time of each interventions
  stl_inputs[['t_inter']] = array(as.numeric(itoday_1based+5))

# Run the stan model
  stan_hosp_only <- "SEIR_Hosp_Only.stan"

StL_fit <- stan(
  file = stan_hosp_only,
  data = stl_inputs,
  thin = 3,
  chains = 3,
  warmup = 200,
  iter = 1700,
  cores = 3,
  refresh = 100,
  control = list(adapt_delta = 0.95)
)
```

```
## DIAGNOSTIC(S) FROM PARSER:
## Info: integer division implicitly rounds to integer. Found int division: itoday - 1 / dknot
##  Positive values rounded down, negative values rounded up or down in platform-dependent way.
## Info: integer division implicitly rounds to integer. Found int division: itoday / dknot
##  Positive values rounded down, negative values rounded up or down in platform-dependent way.
##
## itoday/dknot = 13, nbeta = 19
## beta_knots = [1,10,20,30,40,50,60,70,80,90,100,110,120,130,135,136,144,145,160]
```

```
## Warning: There were 1321 transitions after warmup that exceeded the maximum treedepth. Increase max_
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
## Warning: The largest R-hat is NA, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

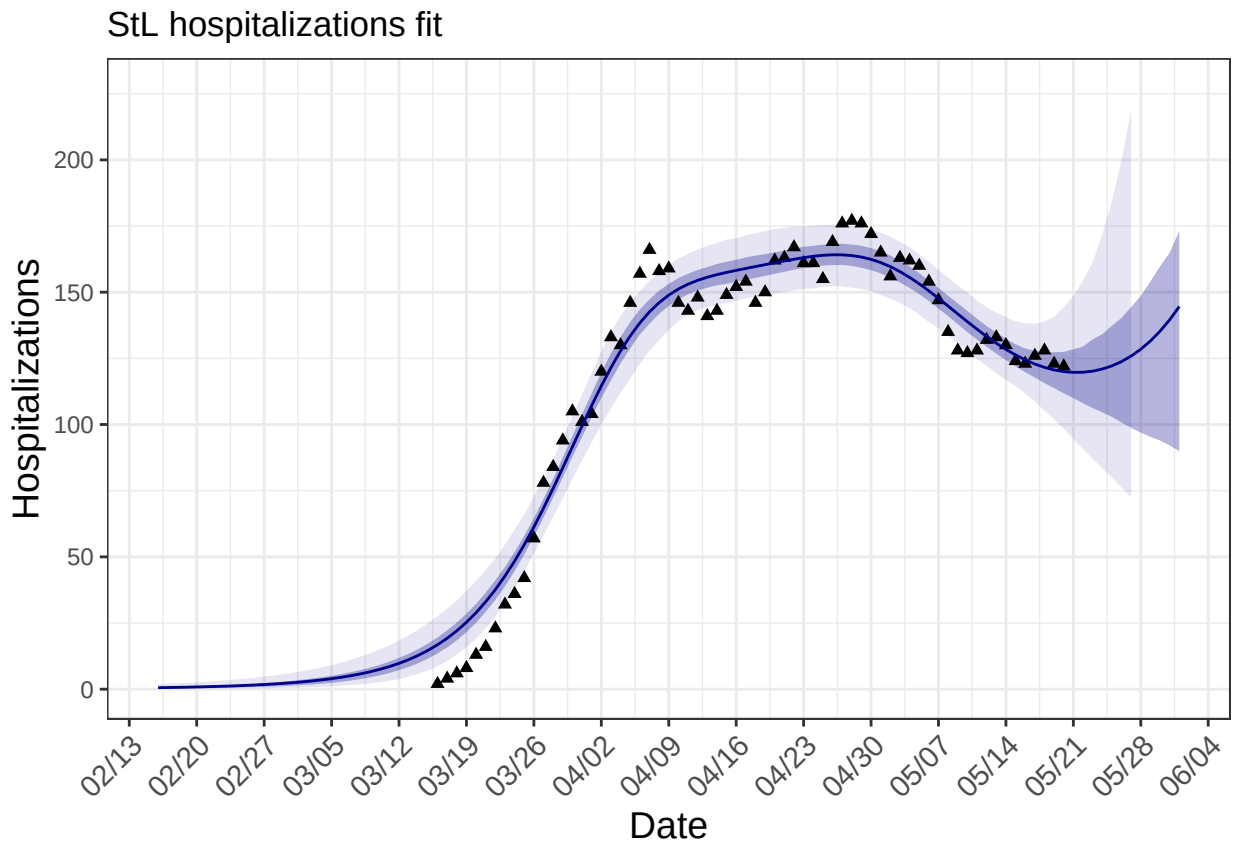
```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```r
stan_StL <- rstan::extract(StL_fit, permuted = T)

# State variables summary from model
state_vars <- c("S", "E", "Ia", "Im", "Ip", "Hosp", "Hicu", "Rlive", "Rdead")

StL_state_sums <- do.call(rbind, lapply(1:dim(stan_StL$x)[2], function(s){
  as_tibble(t(apply(stan_StL$x[,s,],2,stan_post_sum))) %>%
    mutate(State = state_vars[s],
           obs = row_number(),
           Date = as.Date(start.date+obs))
}))

colnames(StL_state_sums)[1:6] <- c("var_mean", "var_med", "var_75th", "var_25th", "var_975th", "var_025

# Observed infections (since we don't observe all infections output by actual model)
StL_hosp <- stan_StL$hospitalized

StL_hosp_sum <- t(apply(StL_hosp, 2, stan_post_sum))

colnames(StL_hosp_sum) <- c("hosp_mean", "hosp_med", "hosp_75th", "hosp_25th", "hosp_975th", "hosp_025t

StL_hosp_sum <- StL_hosp_sum %>% as_tibble() %>%
  mutate(obs = row_number(),
         Date = as.Date(start.date+obs))
```

```r
ggplot() +
  theme_bw() +
  theme(axis.text.x = element_text(size = 11,
                                   angle = 45,
                                   hjust = 1),
        axis.title = element_text(size = 14)) +
# Hospitalizations uncertainty intervals
  geom_ribbon(data = StL_hosp_sum,
              aes(x = Date,
                  ymax = hosp_75th,
                  ymin = hosp_25th),
              fill = "darkblue",
              alpha = 0.3) +
  geom_ribbon(data = StL_hosp_sum,
              aes(x = Date,
                  ymax = hosp_975th,
                  ymin = hosp_025th),
              fill = "darkblue",
              alpha = 0.1) +
# Observed data
  geom_point(data = StL,
             aes(x = as.Date(date), y = hosp.lower/0.9),
             pch = 17) +
# Model means
  geom_line(data = StL_hosp_sum,
            aes(x = Date, y = hosp_mean),
            col = "darkblue") +
# Formatting
```

```
  scale_x_date(date_labels = "%m/%d",
               date_breaks = "7 day",
               limits = as.Date(c("2020-02-16",
                                  "2020-06-01")))+
  ylim(c(0, max(StL$hosp.lower/0.9, na.rm = T)+50)) +
  labs(y = "Hospitalizations",
       title = "StL hospitalizations fit")
```

```
## Warning: Removed 53 rows containing missing values (geom_path).
```



Fit to TN data