CMV transient infection probability estimates

Contents

This looks at quantifying the probability of a blip using geometric distribution of blips

Estimation of blip probability

use all_blips because it includes infants without transient infections (zero blips). The most updated method fits a KM curve then estimates the geometric probability using nls

```
pos_count_data = all_blips %>% filter((!consecutive_swab | is.na(consecutive_swab))) %>%
  group_by(PatientID2, Virus, infantInfection) %>%
  summarize(total_blips = sum(pos))
```

```
#nls estimation of probability using cumulative distribution
pest_fun = function(data_in){
  survm_all = survfit(Surv(time = total_blips, event = infantInfection) ~ 1,
                      data = data_in) %>%
    broom::tidy() %>%
    rename(blips = time)
  nls(estimate ~  (1-p)^(blips + 1),
                     data = survm_all,
                     start = list(p = 0.25), lower = 1e-6, upper = 1 - 1e-6, algorithm = "port") %>%
    coef() %>% unname()
}

bootstrap_fun = function(data_in){
  plyr::ldply(1:1000, function(i){
      boot_data = data_in %>% sample_frac(1, T)

      data.frame(
        run = i,
        est = pest_fun(boot_data)
      )
  })
}
```

```
#this only reruns if it has to
if(file.exists("data/virus_prob_est.RData")) {
  load("data/virus_prob_est.RData")
  print("Did not re-run simulations, loaded previous results Delete the virus_prob_est.RData file in the
  }else{
  viral_probs = plyr::ldply(unique(subset(pos_count_data, !Virus %in% c("HHV8"))$Virus), function(virus
    analysis_data = subset(pos_count_data, Virus == virus)
    total_infected = sum(analysis_data$infantInfection)
    total_blips = sum(analysis_data$total_blips)
    total_blips_inf = sum(analysis_data$total_blips[analysis_data$infantInfection == 1])

    bootstrap_est = bootstrap_fun(ungroup(subset(pos_count_data, Virus == virus)))
```

```
    if(virus == "HHV6") bootstrap_est_infonly = data.frame(run = 1, est = NA) else{
          bootstrap_est_infonly = bootstrap_fun( ungroup(subset(pos_count_data,
                                                                Virus == virus & infantInfection == 1)))
    }

    estimate = median(bootstrap_est$est)
    est_ci = quantile(bootstrap_est$est, c(0.975, 0.025))

    estimate_inf = median(bootstrap_est_infonly$est)
    est_ci_inf = quantile(bootstrap_est_infonly$est, c(0.975, 0.025), na.rm = T)
    #browser()

    data.frame(
      Virus = virus,
      total_infected = total_infected,
      total_blips = total_blips,
      total_blips_inf = total_blips_inf,
      p_est = 1 - estimate,
      est_lower = 1 - est_ci[1],
      est_upper = 1 - est_ci[2],
      p_est_inf = 1 - estimate_inf,
      est_lower_inf = 1 - est_ci_inf[1],
      est_upper_inf = 1 - est_ci_inf[2]
    )
  })
  save(viral_probs, file = "data/virus_prob_est.RData")
}
```

[1] "Did not re-run simulations, loaded previous results Delete the virus_prob_est.RData file in the data/ folder to run new set."

```
prob_table = viral_probs %>%
  mutate(
    est_out = combine_est_interval(p_est, est_lower, est_upper),
    est_out_complete = combine_est_interval(p_est_inf, est_lower_inf, est_upper_inf)
  )
```

```
## Warning in if (is.na(estimate)) return(missing_char): the condition has
## length > 1 and only the first element will be used

## Warning in if (is.na(estimate)) return(missing_char): the condition has
## length > 1 and only the first element will be used
```

```
prob_table$Virus = factor(prob_table$Virus, levels = c("CMV", "EBV", "HSV", "HHV6"), ordered = T)

select(prob_table, Virus, total_blips, est_out, total_blips_inf,  est_out_complete) %>%
  arrange(Virus) %>% xtable() %>% print()
```

Virus	total_blips	est_out	total_blips_inf	est_out_complete
CMV	136	0.88 (0.8, 0.92)	66	0.76 (0.65, 0.84)
EBV	69	0.81 (0.73, 0.87)	47	0.7 (0.56, 0.8)
HSV	70	0.89 (0.84, 0.94)	24	0.72 (0.52, 0.83)
HHV6	6	0.34 (0.11, 0.65)	0	NA (NA, NA)

Cumulative distribution plot

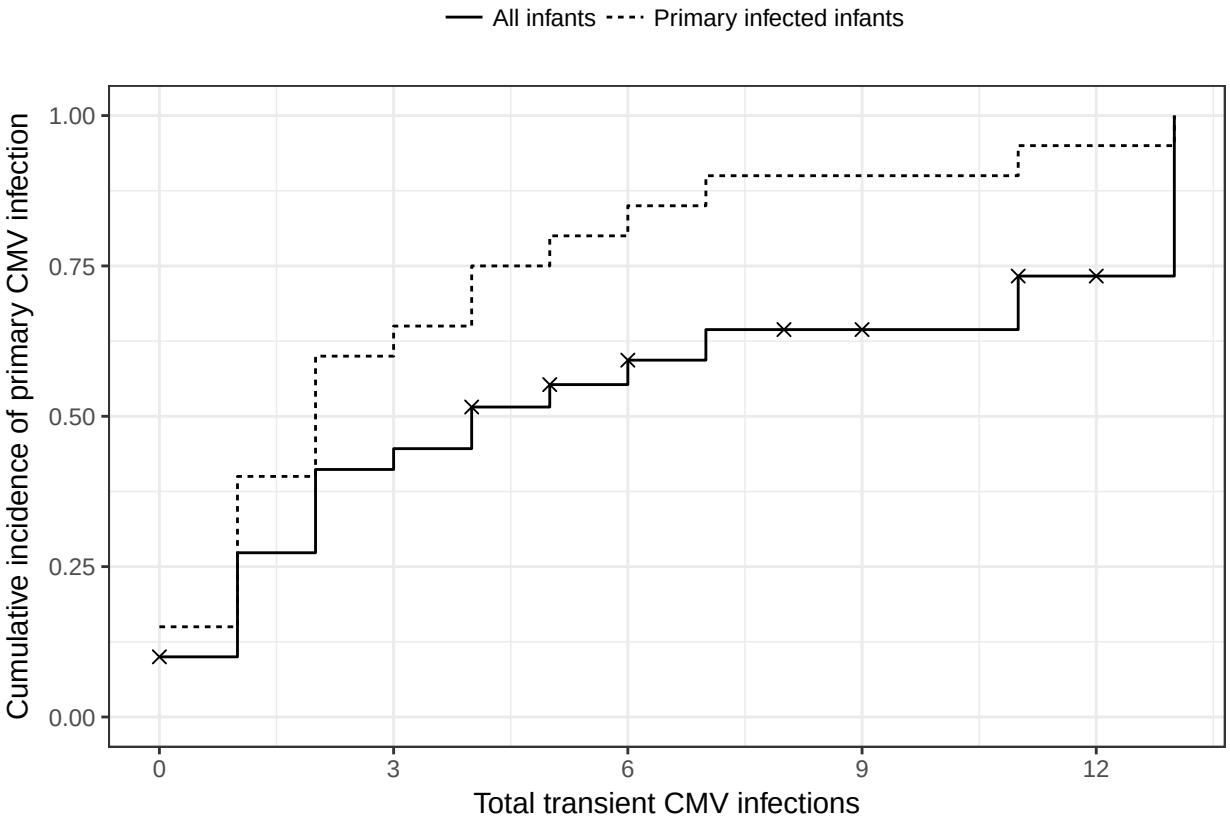
```
cmv_surv_data = plyr::ldply(0:1, function(i){
  survm = survfit(Surv(time = total_blips, event = infantInfection) ~ 1,
                  data = subset(pos_count_data, Virus == "CMV" & infantInfection >= i))

  output = broom::tidy(survm) %>%
    rename(blips = time) %>%
    mutate(censored = n.censor > 0)
  output$infected_only = i == 1
  output
})

test1 = data.frame(blips = 0:13, estimate = pgeom(0:13, .1232, lower.tail = F), infected_only = 1) %>%
  mutate(
    est2 = 1 - estimate,
    upper = est2 + 1/sqrt(20/(est2^2 * (1 - est2))),
         lower = est2 - 1/sqrt(20/(est2^2 * (1 - est2))))
test2 = data.frame(blips = 0:13, estimate = pgeom(0:13, 0.2357, lower.tail = F), infected_only = 1) %>%
  mutate(
        est2 = 1 - estimate,
    upper = est2 + 1/sqrt(20/(est2^2 * (1 - est2))),
         lower = est2 - 1/sqrt(20/(est2^2 * (1 - est2))))

cmv_km = ggplot(data = arrange(cmv_surv_data, blips),
       aes(x = blips, y = 1-estimate, linetype = factor(infected_only))) +
  geom_step() +
  geom_point(data = subset(cmv_surv_data, censored), shape = 4, size = 2) +
  scale_linetype("", breaks = c("FALSE","TRUE"), labels = c("All infants", "Primary infected infants")) +
  scale_y_continuous("Cumulative incidence of primary CMV infection", breaks = 0:4/4, limits = c(0,1)) +
  scale_x_continuous("Total transient CMV infections", breaks = 0:6 * 3)

cmv_km + theme(legend.position = "top")
```




```
#geom_line(data = test1) +  geom_ribbon(data = test1, aes(ymin = lower, ymax = upper), alpha =0.25) +
#geom_line(data = test2) +  geom_ribbon(data = test2, aes(ymin = lower, ymax = upper), alpha =0.25)
```