Sampling in aging test

M.S.

2023-02-21

```
meta <- read_tsv(PATH("metadata.txt"))
```

```
## Rows: 4 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): file.name, donor
## dbl (1): age
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
meta
```

```
## # A tibble: 4 x 3
##   file.name        donor    age
##   <chr>            <chr>  <dbl>
## 1 A6-I200ob.txt.gz donor1    27
## 2 A6-I201ob.txt.gz donor1    30
## 3 A6-I202ob.txt.gz donor2    47
## 4 A5-S23.txt.gz    donor2    50
```

```
data <- meta %>%
  group_by(donor, age) %>%
  group_modify(~read_tsv(PATH(.x$file.name))) %>%
  ungroup %>%
  mutate(tcrkey = paste(v, cdr3nt)) %>%
  group_by(donor, age, tcrkey) %>%
  summarise(count = sum(count)) %>%
  ungroup
```

```
## Rows: 1812062 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): cdr3nt, cdr3aa, v, d, j
## dbl (6): count, freq, VEnd, DStart, DEnd, JStart
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 758985 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): cdr3nt, cdr3aa, v, d, j
## dbl (6): count, freq, VEnd, DStart, DEnd, JStart
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 954170 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): cdr3nt, cdr3aa, v, d, j
## dbl (6): count, freq, VEnd, DStart, DEnd, JStart
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 137624 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): cdr3nt, cdr3aa, v, d, j
## dbl (6): count, freq, VEnd, DStart, DEnd, JStart
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## `summarise()` has grouped output by 'donor', 'age'. You can override using the `.groups` argument.
```

```
data.1 <- data %>%
  left_join(data %>%
              inner_join(data,
                         by = c("donor", "tcrkey")) %>%
              filter(age.x != age.y) %>%
              select(-age.x, -age.y) %>%
              select(donor, tcrkey) %>%
              unique %>%
              mutate(overlapping = T)) %>%
  mutate(overlapping = !is.na(overlapping))
```

```
## Warning in inner_join(., data, by = c("donor", "tcrkey")): Each row in `x` is expected to match at m
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

```
## Joining with `by = join_by(donor, tcrkey)`
```

```
set.seed(42)

na_to_null <- function(x) {
  as.integer(ifelse(is.na(x), 0, x))
}

counts <- rpois(100, 2)

get_freq_table <- function(x) {
  tibble(count = x) %>%
    group_by(count) %>%
    summarise(species = n() %>% as.numeric())
}

get_freq_table(counts)
```

```
## # A tibble: 7 x 2
##   count species
```

```
##   <int>   <dbl>
## 1     0      13
## 2     1      25
## 3     2      25
## 4     3      20
## 5     4      11
## 6     5       5
## 7     6       1
```

```
rarefy <- function(freq_tbl, count_star) {
  count_total = sum(freq_tbl$count * freq_tbl$species)

  areas <- tibble(
    count_star = count_star,
    count_total = count_total,
    phi = count_star / count_total,
    interpolation = count_total >= count_star
  ) %>%
    unique

  freq_tbl <- freq_tbl %>%
    mutate(F1 = sum(species[which(count == 1)]),
           F2 = sum(species[which(count == 2)]),
           Sobs = sum(species),
           Sunseen = F1 * F1 / 2 / F2,
           Sest = Sobs + Sunseen)

  rbind(
    freq_tbl %>%
      cross_join(areas %>% filter(interpolation)) %>%
      group_by(count_star, interpolation, Sobs, Sest, count_total) %>%
      summarise(Sarea = sum(species * (1 - (1 - phi) ^ count)),
                VarSarea = sum(species * (1 - (1 - phi) ^ count)^2) -
                  Sarea[1] * Sarea[1] / Sest[1]),
    freq_tbl %>%
      cross_join(areas %>% filter(!interpolation)) %>%
      group_by(count_star, interpolation, Sobs, Sest, count_total) %>%
      summarise(Sarea = Sobs[1] +
                  Sunseen[1] * (1 - exp(-(phi[1] - 1) * F1[1] / Sunseen[1])),
                VarSarea = NA)
  ) %>%
    ungroup
}

rarefy(get_freq_table(counts), c(10, 50, 500) %>% as.integer)
```

```
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
```

```
## # A tibble: 3 x 7
##   count_star interpolation  Sobs  Sest count_total  Sarea VarSarea
##        <int> <lgl>         <dbl> <dbl>       <dbl>  <dbl>    <dbl>
## 1         10 TRUE            100  112.         210   9.52    0.482
```

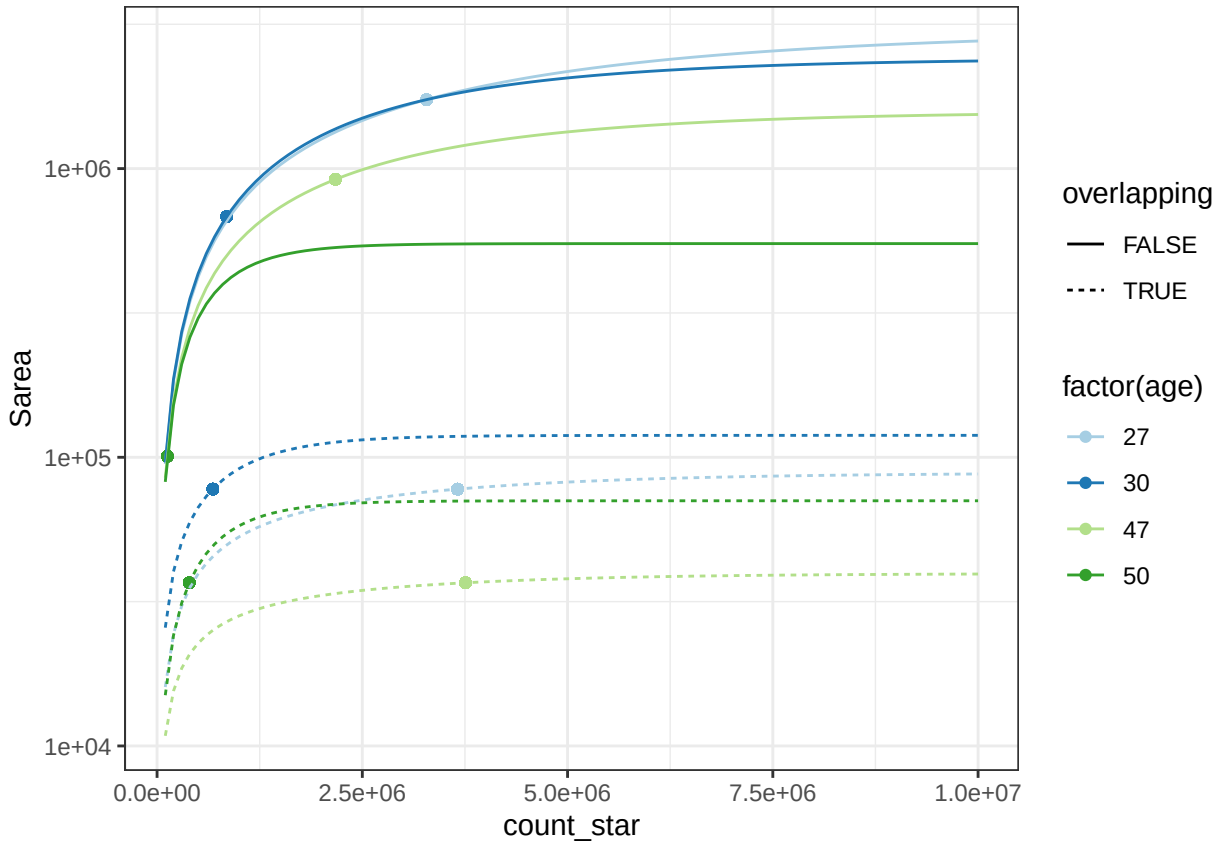
```
## 2         50 TRUE            100  112.         210  39.4     6.48
## 3        500 FALSE           100  112.         210 112.     NA
```

```
points <- seq(1, 10000000, length.out = 101) %>% as.numeric()

data.1 %>%
  group_by(donor, age, overlapping) %>%
  group_modify(~get_freq_table(.x$count) %>%
                 rarefy(points)) -> data.r
```

```
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'count_star', 'interpolation', 'Sobs',
## 'Sest'. You can override using the `.groups` argument.
```

```
data.r %>%
  filter(count_star > 100) %>%
  ggplot(aes(x = count_star, y = Sarea,
             group = paste(age, overlapping),
             color = factor(age), linetype = overlapping)) +
  geom_point(data = data.r, aes(x = count_total, y = Sobs)) +
  geom_path() +
  scale_y_log10() +
  scale_color_brewer(palette = "Paired") +
  theme_bw()
```


Sarea
1e+06
1e+05
1e+04
0.0e+00
2.5e+06
5.0e+06
7.5e+06
1.0e+07
count_star
overlapping
FALSE
TRUE
factor(age)
27
30
47
50