Visualizing Pangolin Uncertainty (a.k.a. Pangoluncertainty)

Devan Becker

2021-02-10

```
library(here)
library(ggplot2)
library(dplyr)
library(tidyr)
library(stringr)
```

Goal

Visualize/summarize the results of sending resampled uncertainty matrices through pangolin.

Vis Setup

Each row in the dataset represents a single sample from one uncertainty matrix (only dealing with one accession number for now). The columns are as folows:

- `taxon`: the accession name, followed by ".0" for the consensus sequence and "0.1" to "0.1000" for the 1,000 samples from the matrix.
  - This is split into `taxon`, which is all identical for one file, and `sample`, which is 0 for the conseq and 1:1000 for the 1,000 samples.
- `lineage`: The called lineage from pangolin
- `probability`: the bootstrap support for this lineage call
- `pangoLEARN_version`, `status`, and `note`: extra info from pangolin

```
unc1 <- read.csv(here("data/pangolineages/", "ERR4085809_pangolineages.csv"))
unc1 <- unc1 %>%
    separate(col = "taxon", sep = "\\.",
        into = c("taxon", "sample")) %>%
    mutate(taxon = str_replace(taxon, "\\_", ""))
head(unc1)
```

```
##        taxon sample lineage probability pangoLEARN_version    status note
## 1 ERR4085809      0     B.1           1         2021-02-21 passed_qc   NA
## 2 ERR4085809      1     A.2           1         2021-02-21 passed_qc   NA
## 3 ERR4085809      2       B           1         2021-02-21 passed_qc   NA
## 4 ERR4085809      3       B           1         2021-02-21 passed_qc   NA
## 5 ERR4085809      4  B.1.98           1         2021-02-21 passed_qc   NA
## 6 ERR4085809      5       B           1         2021-02-21 passed_qc   NA
```

To prep the data, I calculate another column called "prop", which represents the number of samples that were assigned the sam lineage as the one in that row. So, for a lineage assigned B.1, prop would be the total number of B.1s in the sampled lineages divided by 1000.

```
unc2 <- unc1 %>%
    group_by(lineage) %>%
    summarise(prop = n() / (nrow(unc1) - 1)) %>%
    right_join(unc1, by = "lineage") %>%
    arrange(as.numeric(sample))
```

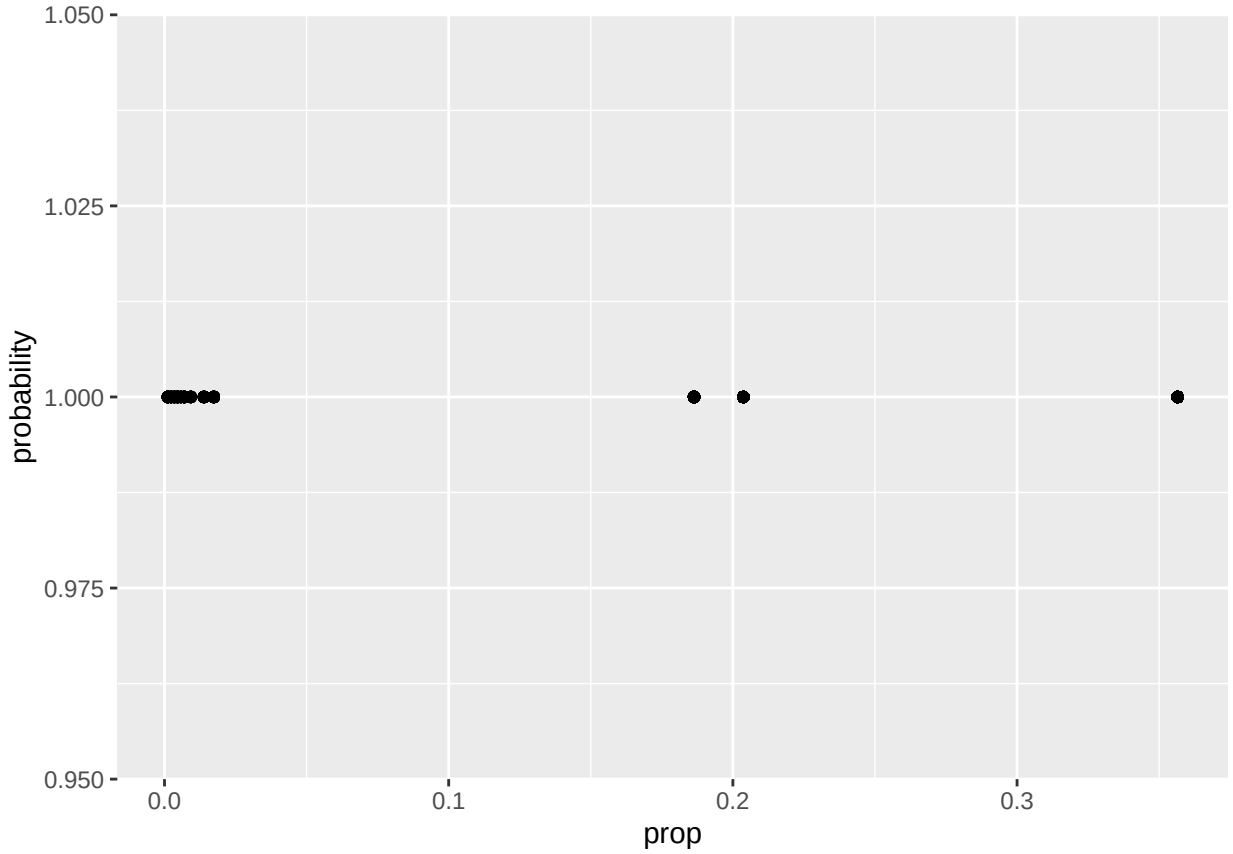
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(unc2)
```

```
## # A tibble: 6 x 8
##   lineage    prop taxon     sample probability pangoLEARN_version status   note
##   <chr>     <dbl> <chr>     <chr>        <dbl> <chr>              <chr>    <lgl>
## 1 B.1     0.186   ERR40858~ 0                1 2021-02-21         passed_~ NA
## 2 A.2     0.00694 ERR40858~ 1                1 2021-02-21         passed_~ NA
## 3 B       0.204   ERR40858~ 2                1 2021-02-21         passed_~ NA
## 4 B       0.204   ERR40858~ 3                1 2021-02-21         passed_~ NA
## 5 B.1.98  0.00231 ERR40858~ 4                1 2021-02-21         passed_~ NA
## 6 B       0.204   ERR40858~ 5                1 2021-02-21         passed_~ NA
```

Now we can view the differences between the pangolin bootstrap support and the actual proportion of samples with that lineage designation!

```
ggplot(unc2) +
    aes(x = prop, y = probability) +
    geom_point()
```



Right now it's not very interesting because all of the pangolin probabilities are 1. I changed something in my code, and I don't know why this broke it. It's on my TODO.