Results of PangoVis

Devan Becker

2021-04-10

Load Packages and Data

```
# Packages that Art hates
library(dplyr)
library(tidyr)
library(ggplot2)
library(stringr)
library(here)

dirich <- params$dirich

# Read in CSV files
csvs <- list.files(here("data/", "pangolineages"),
    pattern = ifelse(dirich, "*_d.csv", "*.csv"),
    full.names = TRUE)

# Remove any copies
csvs <- csvs[!grepl("-1", csvs)]

# Bring them into one data frame
lins <- bind_rows(lapply(csvs, read.csv))

# Taxon is encoded as _ACCSESSIONNUMBER.ID, split into ACCESSIONNUMBER and ID
lins <- lins %>%
    separate(col = "taxon", sep = "\\.",
        into = c("taxon", "sample")) %>%
    mutate(taxon = str_replace(taxon, "\\_", ""))

badlins <- table(lins$taxon)
badlins <- names(badlins[which(badlins < 5000)])
cat(length(badlins), " runs were removed for having too few samples.")
```

```
## 11  runs were removed for having too few samples.
```

```
lins <- filter(lins, !taxon %in% badlins)

#### Visualize the uncertainty in the base calls ----
taxons <- sort(unique(lins$taxon))
length(taxons)
```

```
## [1] 80
```

Abstract Info

```
summs <- lins %>%
    group_by(taxon) %>%
    summarise(
        maxperc = mean(lineage == names(sort(table(lineage),
            decreasing = TRUE)[1])),
        uniques = length(unique(lineage)),
        minpango = min(probability),
        maxpango = max(probability),
        menpango = mean(probability),
        max = names(sort(table(lineage), decreasing = TRUE))[1])
```

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

```
print("summary info")
```

```
## [1] "summary info"
```

```
print(summs)
```

```
## # A tibble: 80 x 7
##    taxon      maxperc uniques minpango maxpango menpango max
##    <chr>        <dbl>   <int>    <dbl>    <dbl>    <dbl> <chr>
##  1 ERR4363387   0.938      12        1        1        1 B.1.222
##  2 ERR4364007   0.871      43        1        1        1 B.1.1.29
##  3 ERR4664555   0.945      18        1        1        1 B.1.1.253
##  4 ERR4667618   0.990       4        1        1        1 B.1.1.315
##  5 ERR4692364   0.914      20        1        1        1 B.1
##  6 ERR4693034   0.847      32        1        1        1 B.1.1.310
##  7 ERR4693061   0.941       9        1        1        1 B.23
##  8 ERR4693079   0.866      41        1        1        1 B.1.1.310
##  9 ERR4693537   0.972       6        1        1        1 B.1.177.7
## 10 ERR4693605   0.970      11        1        1        1 B.1.177.3
## # ... with 70 more rows
```

```
1 - mean(summs$maxperc); 1 - mean(summs$menpango)
```

```
## [1] 0.09633206
```

```
## [1] 0.0249975
```

```
#print(summs, n = Inf)
```

Stacked Bar Plots

```
max_label <- 250
other_label <- 100

par(mfrow = c(17, 1), mar = c(0.05, 7.75, 0.05, 0.05))
if (exists("seq_info")) rm(seq_info)
for (i in seq_along(taxons)) {
    pang <- lins[lins$taxon == taxons[i], ]
    called <- pang$lineage[pang$sample == 0][1]
    pangtab <- sort(table(pang$lineage), decreasing = TRUE)
```

```
    seq_info_i <- data.frame(
        called = called,
        mode = names(pangtab)[1],
            mode_count = pangtab[1],
        runner_up = names(pangtab)[2],
            runner_up_count = pangtab[2],
        unique = length(pangtab), atoms = sum(pangtab == 1))
    seq_info_i$taxon <- taxons[i]

    if (!exists("seq_info")) {
        seq_info <- seq_info_i
    } else {
        seq_info <- bind_rows(seq_info, seq_info_i)
    }

    colvec <- rep("grey", length(pangtab))
    colvec[which(names(pangtab) == called)] <- "red"

    n <- sum(pangtab > max_label)
    if (n > 1) {
        add_other <- FALSE
        if (sum(pangtab < other_label) > 10) {
            add_other <- TRUE
            other_count <- sum(pangtab <= other_label)
            pangtab <- c(pangtab[pangtab > other_label],
                c("other" = sum(pangtab[pangtab <= other_label])))
            colvec[which(names(pangtab) == "other")] <- "black"
        }
        barlabx <- c(0, cumsum(pangtab[1:(n - 1)])) +
            pangtab[1:n] / 2
        barlabels <- names(pangtab)[1:n]
        barlens <- sapply(gregexpr("\\.", barlabels), length)
        for (j in seq_along(barlabels)) {
            if (pangtab[j] < 400 & barlens[j] >= 2) {
                barsplit <- strsplit(barlabels[j], split = "\\.")[[1]]
                barn <- length(barsplit)
                half <- floor(barn / 2)
                barlabels[j] <- paste0(
                    paste(barsplit[1:half], collapse = "."),
                    ".\n",
                    paste(barsplit[(half + 1):barn], collapse = ".")
                )
            }
        }

        barplot(as.matrix(pangtab),
            col = colvec, hori = TRUE, axes = FALSE)
        text(barlabx, 0.7, barlabels, cex = 1.5)
        if (add_other) {
            text(x = sum(pangtab) - pangtab["other"] / 2,
            y = 0.7, col = "white", cex = 1.5,
            label = paste0("Others:\n", other_count))
        }
```

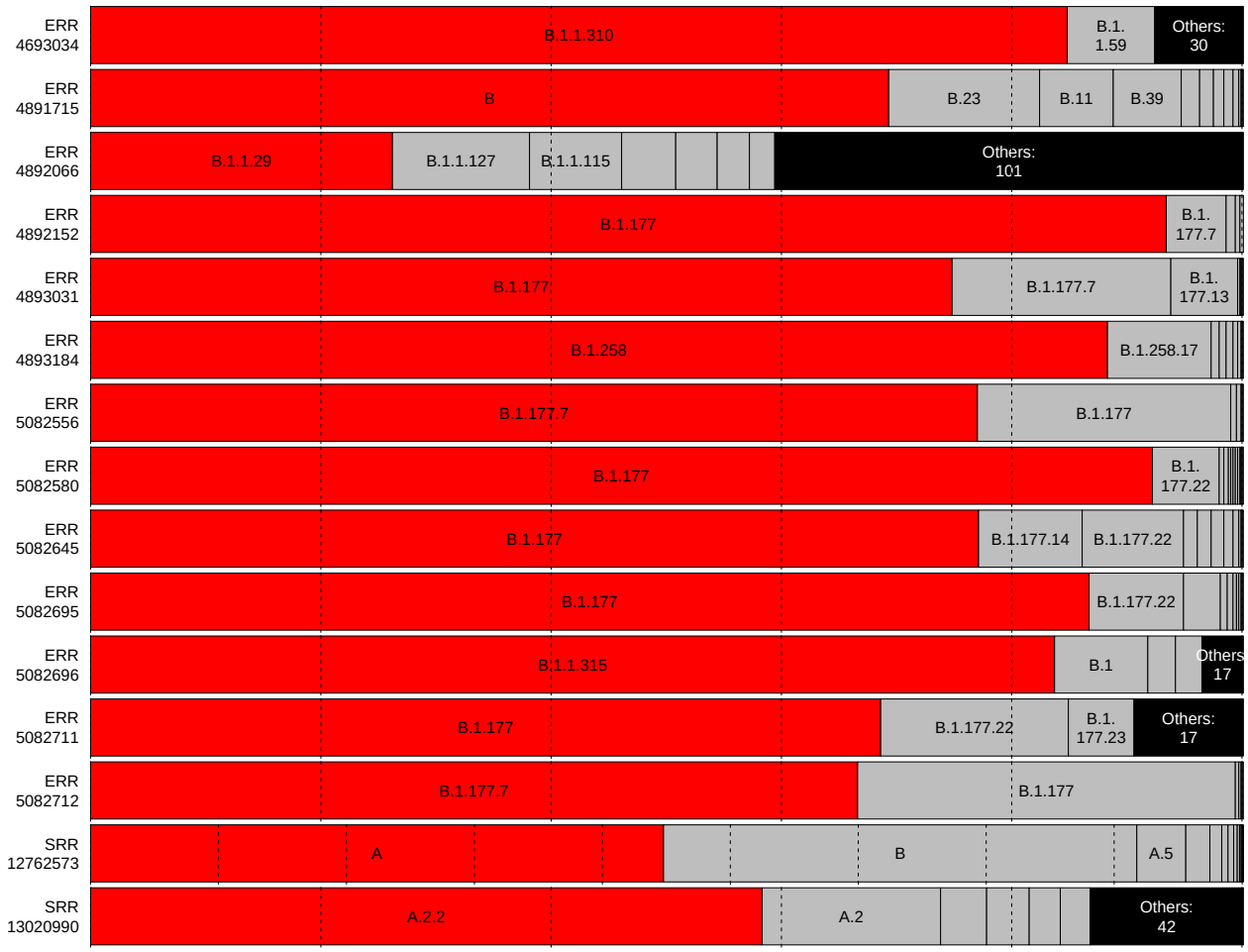
```
        mtext(side = 2, cex = 1, las = 1,
            text = paste(substr(taxons[i], 1, 3),
                substr(taxons[i], 4, 20), sep = "\n"))
        abline(v = seq(0, 10000, 1000), lty = 2)
        "pretty_labels <- seq(0, sum(pangtab),
                by = ifelse(sum(pangtab) < 2000, 100, 1000))
        mtext(side = 1,
            at = pretty_labels,
            text = pretty_labels,
            line = 0,
            cex = 0.75
        )"
    }
}

seq_info$taxon <- taxons
knitr::kable(seq_info, row.names = FALSE)
```

called	mode	mode_count	runner_up	runner_up_count	unique	atoms	taxon
B.1.222	B.1.222	5635	B.1	174	12	0	ERR4363387
B.1.1.29	B.1.1.29	5233	B.1.1	84	43	0	ERR4364007
B.1.1.253	B.1.1.253	5677	B.1	90	18	0	ERR4664555
B.1.1.315	B.1.1.315	5947	B.1	42	4	0	ERR4667618
B.1	B.1	5491	B.1.367	132	20	0	ERR4692364
B.1.1.310	B.1.1.310	4241	B.1.1.59	380	32	0	ERR4693034
B.23	B.23	4711	B.48	130	9	0	ERR4693061
B.1.1.310	B.1.1.310	4336	B.1.1.29	180	41	0	ERR4693079
B.1.177.7	B.1.177.7	4866	B.1.177	115	6	0	ERR4693537
B.1.177.3	B.1.177.3	4856	B.1.177	60	11	0	ERR4693605
B.1.36	B.1.36	4826	B.1.36.9	35	15	0	ERR4891711
B	B	3466	B.23	655	12	0	ERR4891715
B.1.177.6	B.1.177.6	4911	B.1.177	35	7	0	ERR4891805
B.1	B.1	3656	B.1.243	95	97	0	ERR4891841
B.1.1.216	B.1.1.216	4186	B.1	90	58	0	ERR4891863
B.1.98	B.1.98	4321	B.1	240	22	0	ERR4891889
B.1.1.304	B.1.1.304	4841	B.1.1.70	20	16	0	ERR4891898
None	None	5002	B.1	1	2	1	ERR4891916
B	B	4361	B.1	125	35	0	ERR4891988
B.1.1.307	B.1.1.307	4946	B.1	25	8	0	ERR4892048
B.1.1.29	B.1.1.29	1311	B.1.1.127	595	108	0	ERR4892066
B.39	B.39	4871	B	70	13	0	ERR4892112
B.1.177	B.1.177	4671	B.1.177.7	260	5	0	ERR4892152
B.1.177.17	B.1.177.17	4961	B.1.177	45	2	0	ERR4892200
B.1.1.216	B.1.1.216	4521	B.1.1.29	60	32	0	ERR4892203
B.1.1.162	B.1.1.162	3016	B.1.1.141	235	90	0	ERR4892293
B.1.177	B.1.177	4806	B.1.177.22	85	8	0	ERR4892339
B.40	B.40	4981	B	10	5	0	ERR4892386
B.1.177	B.1.177	4836	B.1.177.22	55	10	0	ERR4892392
B.1.523	B.1.523	4556	B.1.471	225	13	0	ERR4892423
B.1	B.1	4501	B.1.36	105	18	0	ERR4893013
B.1.177	B.1.177	3741	B.1.177.7	950	7	0	ERR4893031
B.1.1.307	B.1.1.307	4936	B.1	25	8	0	ERR4893033
B.1.1.29	B.1.1.29	2711	B.1.1.37	85	124	0	ERR4893037

called	mode	mode_count	runner_up	runner_up_count	unique	atoms	taxon
B.1.177	B.1.177	4786	B.1.177.22	135	8	0	ERR4893080
B.1.177.16	B.1.177.16	4961	B.1.177	40	3	0	ERR4893138
B.1.258	B.1.258	4416	B.1.258.17	450	9	0	ERR4893184
B.1.1.216	B.1.1.216	4576	B.1.1.141	90	35	0	ERR4893186
B.1.177	B.1.177	4686	B.1.177.23	205	7	0	ERR4893197
B.1.177	B.1.177	4916	B.1.177.22	55	8	0	ERR4893242
B.1.1.307	B.1.1.307	4966	B.1.1.311	35	3	0	ERR4893353
B.1.523	B.1.523	4656	B.1	145	13	0	ERR4893393
B.1.177	B.1.177	4811	B.1.177.22	120	8	0	ERR5062571
B.1.177.19	B.1.177.19	4936	B.1.177	60	3	0	ERR5063165
B.1.177	B.1.177	4686	B.1.177.22	205	8	0	ERR5064166
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5069584
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5069616
B.1	B.1	4821	B.1.247	65	21	0	ERR5069624
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5069871
B.1.177	B.1.177	4866	B.1.177.22	90	6	0	ERR5070060
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5070294
B.1.160	B.1.160	4801	B.1.160.7	120	6	0	ERR5074314
B.1.177.19	B.1.177.19	4906	B.1.177	40	6	0	ERR5076163
B.1.177.19	B.1.177.19	4931	B.1.177	70	3	0	ERR5076748
B.1.177	B.1.177	4761	B.1.177.22	150	12	0	ERR5077151
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5077411
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5077618
B.1.36.17	B.1.36.17	4806	B.1	85	9	0	ERR5080897
B.1.1.311	B.1.1.311	4806	B.1	100	8	0	ERR5080913
B.1.177	B.1.177	4681	B.1.177.22	135	15	0	ERR5080918
B.1.177	B.1.177	4761	B.1.177.22	130	12	0	ERR5081301
B.1.177.4	B.1.177.4	4931	B.1.177	25	9	0	ERR5081304
B.1.177.15	B.1.177.15	4756	B.1.177	170	9	0	ERR5081316
B.1.177.7	B.1.177.7	3851	B.1.177	1100	6	0	ERR5082556
B.1.160	B.1.160	4851	B.1.160.7	70	11	0	ERR5082569
B.1.177	B.1.177	4611	B.1.177.22	290	12	0	ERR5082580
B.1.177.4	B.1.177.4	4991	B.1.177.2	15	2	0	ERR5082590
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5082610
B.1.177.7	B.1.177.7	4776	B.1.177	185	5	0	ERR5082630
B.1.177	B.1.177	3856	B.1.177.14	450	11	0	ERR5082645
B.1.1.315	B.1.1.315	4901	B.1	45	8	0	ERR5082664
B.1.177.6	B.1.177.6	4891	B.1.177	70	7	0	ERR5082674
B.1.177	B.1.177	4336	B.1.177.22	410	10	0	ERR5082695
B.1.1.315	B.1.1.315	4186	B.1	405	21	0	ERR5082696
B.1.177	B.1.177	3431	B.1.177.22	815	20	0	ERR5082711
B.1.177.7	B.1.177.7	3331	B.1.177	1640	6	0	ERR5082712
None	None	5006	NA	NA	1	0	SRR12639958
A	A	4478	B	3699	13	0	SRR12762573
A.2.2	A.2.2	2916	A.2	775	48	0	SRR13020990
A.1	A.1	8902	B.40	58	11	0	SRR13092002

```
rm(seq_info)
```


ERR 4693034
B.1.1.310
B.1. 1.59
Others: 30
ERR 4891715
B
B.23
B.11
B.39
ERR 4892066
B.1.1.29
B.1.1.127
B.1.1.115
Others: 101
ERR 4892152
B.1.177
B.1. 177.7
ERR 4893031
B.1.177
B.1.177.7
B.1. 177.13
ERR 4893184
B.1.258
B.1.258.17
ERR 5082556
B.1.177.7
B.1.177
ERR 5082580
B.1.177
B.1. 177.22
ERR 5082645
B.1.177
B.1.177.14
B.1.177.22
ERR 5082695
B.1.177
B.1.177.22
ERR 5082696
B.1.1.315
B.1
Others 17
ERR 5082711
B.1.177
B.1.177.22
B.1. 177.23
Others: 17
ERR 5082712
B.1.177.7
B.1.177
SRR 12762573
A
B
A.5
SRR 13020990
A.2.2
A.2
Others: 42