Results of PangoVis

Devan Becker

2021-04-12

Load Packages and Data

```
# Packages that Art hates
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

```
## here() starts at /home/devan/OneDriveUWO/0postdoc/sup
```

```
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)])
```

```r
cat(length(badlins), " runs were removed for having too few samples.")
```

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

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

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

```
## [1] 93
```

Abstract Info

```r
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)
```

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

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

```r
print(summs)
```

```
## # A tibble: 93 x 7
##    taxon      maxperc uniques minpango maxpango menpango max
##    <chr>        <dbl>   <int>    <dbl>    <dbl>    <dbl> <chr>
##  1 ERR4305816   0.949      15        1        1        1 B.3.1
##  2 ERR4307842   0.950      31        1        1        1 B.1.1.289
##  3 ERR4363387   0.950      24        1        1        1 B.1.222
##  4 ERR4364007   0.839      76        1        1        1 B.1.1.29
##  5 ERR4664555   0.964      28        1        1        1 B.1.1.253
##  6 ERR4667618   0.994       8        1        1        1 B.1.1.315
##  7 ERR4692364   0.926      62        1        1        1 B.1
##  8 ERR4693034   0.895      76        1        1        1 B.1.1.310
##  9 ERR4693061   0.950      23        1        1        1 B.23
## 10 ERR4693079   0.875     117        1        1        1 B.1.1.310
## # ... with 83 more rows
```

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

```
## [1] 0.1003398
```

```
## [1] 0.05376129
```

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

Stacked Bar Plots

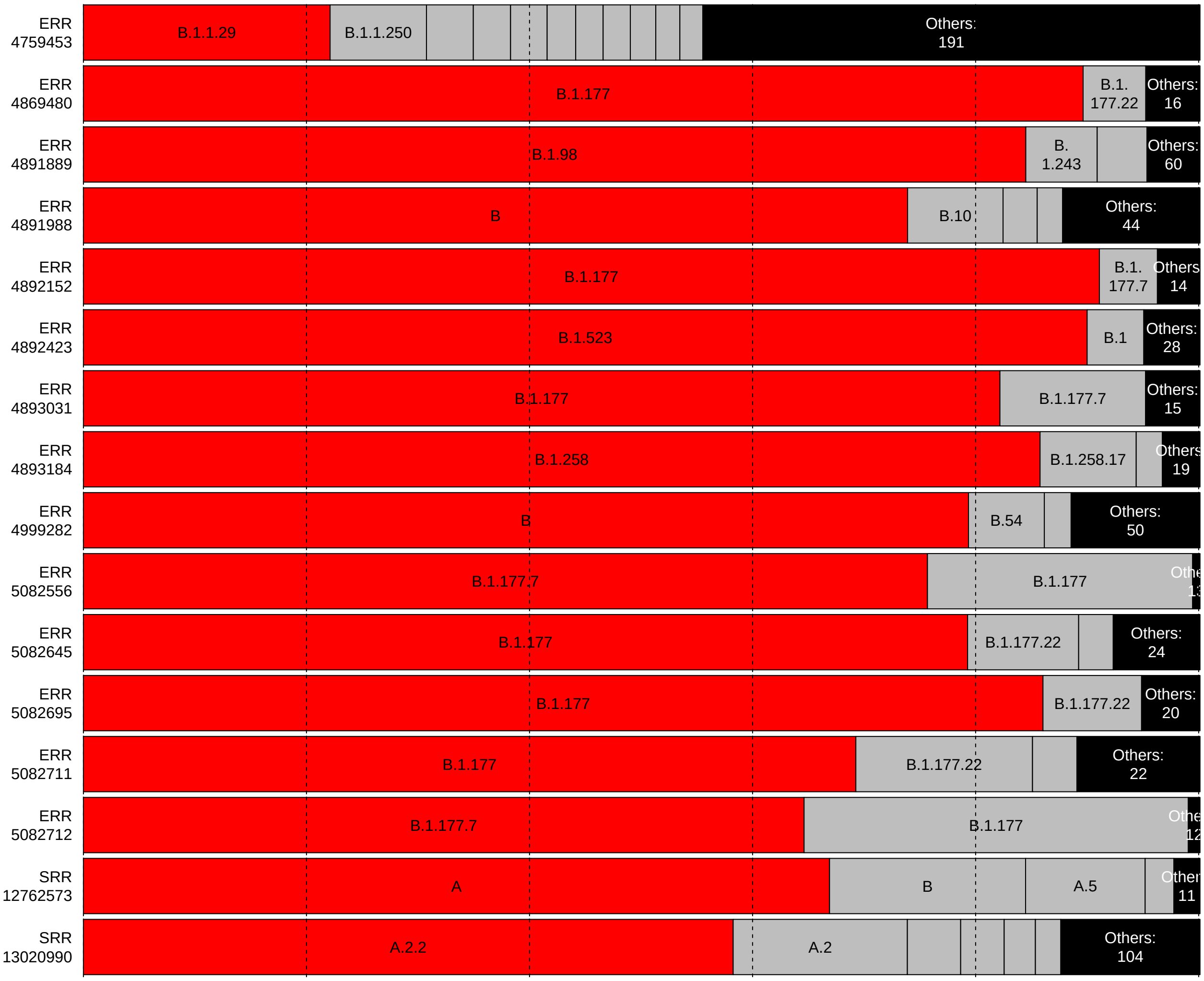
```
max_label <- 250
other_label <- 100

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],
            perc = round(100*pangtab[1] / sum(pangtab), 2),
        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
seq_info <- arrange(seq_info, mode, mode_count)
knitr::kable(seq_info, row.names = FALSE)
```

called	mode	mode_count	perc	runner_up	runner_up_count	unique	atoms	taxon
A	A	3345	66.82	B	879	15	4	SRR12762573
A.1	A.1	4925	98.38	B.40	34	14	3	SRR13092002
A.2.2	A.2.2	2913	58.19	A.2	781	110	52	SRR13020990
B	B	3695	73.81	B.10	428	48	14	ERR4891988
B	B	3968	79.26	B.54	340	53	7	ERR4999282
B	B	4290	85.70	B.23	196	30	8	ERR4891715
B.1	B.1	3786	75.63	B.1.243	62	211	43	ERR4891841
B.1	B.1	4530	90.49	B.1.88	103	49	15	ERR4893013
B.1	B.1	4638	92.65	B.1.400	76	62	24	ERR4692364
B.1	B.1	4821	96.30	B.1.247	67	46	18	ERR5069624
B.1.1.162	B.1.1.162	3004	60.01	B.1.1.274	122	204	51	ERR4892293
B.1.1.216	B.1.1.216	4176	83.42	B.1	118	120	45	ERR4891863
B.1.1.216	B.1.1.216	4458	89.05	B.1.1.208	83	92	37	ERR4893186
B.1.1.216	B.1.1.216	4525	90.39	B.1	43	88	27	ERR4892203
B.1.1.251	B.1.1.251	2909	58.11	B.1	183	160	43	ERR5080893
B.1.1.253	B.1.1.253	4824	96.36	B.1	49	28	12	ERR4664555
B.1.1.289	B.1.1.289	4757	95.03	B.1	54	31	11	ERR4307842
B.1.1.29	B.1.1.29	1106	22.09	B.1.1.250	432	202	41	ERR4759453
B.1.1.29	B.1.1.29	1440	28.77	B.1.1.127	185	191	26	ERR4892066
B.1.1.29	B.1.1.29	2520	50.34	B.1.1.39	86	228	29	ERR4893037
B.1.1.29	B.1.1.29	4201	83.92	B.1.1	60	76	21	ERR4364007

called	mode	mode_count	perc	runner_up	runner_up_count	unique	atoms	taxon
B.1.1.304	B.1.1.304	4832	96.52	B.1	31	34	12	ERR4891898
B.1.1.307	B.1.1.307	4871	97.30	B.1	53	44	28	ERR4893033
B.1.1.307	B.1.1.307	4962	99.12	B.1	31	9	6	ERR4893353
B.1.1.307	B.1.1.307	4978	99.44	B.1.1.37	8	14	8	ERR4892048
B.1.1.310	B.1.1.310	4380	87.50	B.1.1.29	196	117	57	ERR4693079
B.1.1.310	B.1.1.310	4482	89.53	B.1.1.59	103	76	27	ERR4693034
B.1.1.311	B.1.1.311	4872	97.32	B.1	43	20	14	ERR5080913
B.1.1.315	B.1.1.315	4587	91.63	B.1.1.281	203	27	9	ERR5082696
B.1.1.315	B.1.1.315	4857	97.02	B.1	55	14	6	ERR5082664
B.1.1.315	B.1.1.315	4956	99.00	B.1	18	11	5	ERR4869497
B.1.1.315	B.1.1.315	4976	99.40	B.1	18	8	3	ERR4667618
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5069584
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5069616
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5069871
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5070294
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5077411
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5077618
B.1.1.7	B.1.1.7	5006	100.00	NA	NA	1	0	ERR5082610
B.1.160	B.1.160	4775	95.39	B.1.160.8	119	18	9	ERR5074314
B.1.160	B.1.160	4894	97.76	B.1.160.5	28	22	9	ERR5082569
B.1.160.7	B.1.160.7	4944	98.76	B.1.160	46	5	1	ERR4869446
B.1.177	B.1.177	3463	69.18	B.1.177.22	792	25	2	ERR5082711
B.1.177	B.1.177	3964	79.18	B.1.177.22	498	27	9	ERR5082645
B.1.177	B.1.177	4109	82.08	B.1.177.7	653	17	4	ERR4893031
B.1.177	B.1.177	4302	85.94	B.1.177.22	442	22	7	ERR5082695
B.1.177	B.1.177	4482	89.53	B.1.177.22	281	18	0	ERR4869480
B.1.177	B.1.177	4555	90.99	B.1.177.7	260	16	3	ERR4892152
B.1.177	B.1.177	4578	91.45	B.1.177.23	142	19	6	ERR4892339
B.1.177	B.1.177	4621	92.31	B.1.177.22	242	18	2	ERR5082580
B.1.177	B.1.177	4651	92.91	B.1.177.22	188	19	1	ERR5064166
B.1.177	B.1.177	4656	93.01	B.1.177.22	185	19	1	ERR5081301
B.1.177	B.1.177	4662	93.13	B.1.177.22	200	20	4	ERR4869458
B.1.177	B.1.177	4707	94.03	B.1.177.22	140	22	5	ERR5080918
B.1.177	B.1.177	4709	94.07	B.1.177.22	137	17	6	ERR4893242
B.1.177	B.1.177	4716	94.21	B.1.177.22	175	17	1	ERR4869487
B.1.177	B.1.177	4755	94.99	B.1.177.22	95	17	5	ERR4892392
B.1.177	B.1.177	4790	95.69	B.1.177.22	138	20	3	ERR5077151
B.1.177	B.1.177	4802	95.92	B.1.177.22	88	13	3	ERR5070060
B.1.177	B.1.177	4803	95.94	B.1.177.22	107	19	3	ERR5062571
B.1.177	B.1.177	4849	96.86	B.1.177.22	84	17	4	ERR4893080
B.1.177	B.1.177	4889	97.66	B.1.177.22	48	15	5	ERR4893197
B.1.177.15	B.1.177.15	4777	95.43	B.1.177	167	22	12	ERR5081316
B.1.177.16	B.1.177.16	4970	99.28	B.1.177	26	7	4	ERR4893138
B.1.177.17	B.1.177.17	4967	99.22	B.1.177	36	4	1	ERR4892200
B.1.177.19	B.1.177.19	4859	97.06	B.1.177	102	13	4	ERR5076163
B.1.177.19	B.1.177.19	4893	97.74	B.1.177	80	11	4	ERR5076748
B.1.177.19	B.1.177.19	4915	98.18	B.1.177	68	10	2	ERR5063165
B.1.177.3	B.1.177.3	4827	96.42	B.1.177	56	20	6	ERR4693605
B.1.177.4	B.1.177.4	4967	99.22	B.1.177.2	24	11	7	ERR5081304
B.1.177.4	B.1.177.4	4980	99.48	B.1.177.2	21	6	3	ERR5082590
B.1.177.6	B.1.177.6	4876	97.40	B.1.177	75	13	3	ERR5082674
B.1.177.6	B.1.177.6	4912	98.12	B.1.177	34	11	4	ERR4891805

called	mode	mode_count	perc	runner_up	runner_up_count	unique	atoms	taxon
B.1.177.7	B.1.177.7	3231	64.54	B.1.177	1722	14	6	ERR5082712
B.1.177.7	B.1.177.7	3784	75.59	B.1.177	1189	15	7	ERR5082556
B.1.177.7	B.1.177.7	4822	96.32	B.1.177	161	13	7	ERR5082630
B.1.177.7	B.1.177.7	4883	97.54	B.1.177	105	9	3	ERR4693537
B.1.222	B.1.222	4756	95.01	B.1	102	24	9	ERR4363387
B.1.258	B.1.258	4289	85.68	B.1.258.17	431	22	5	ERR4893184
B.1.36	B.1.36	4758	95.05	B.1.36.9	87	28	12	ERR4891711
B.1.36.17	B.1.36.17	4745	94.79	B.1	161	25	12	ERR5080897
B.1.523	B.1.523	4458	89.05	B.1	126	29	9	ERR4893393
B.1.523	B.1.523	4500	89.89	B.1	254	30	11	ERR4892423
B.1.98	B.1.98	4225	84.40	B.1.243	320	63	31	ERR4891889
B.23	B.23	4754	94.97	B.48	111	23	9	ERR4693061
B.3.1	B.3.1	4751	94.91	B.3	173	15	3	ERR4305816
B.39	B.39	4661	93.11	B	192	38	18	ERR4892112
B.40	B.40	4953	98.94	B	18	11	3	ERR4892386
None	None	5002	99.98	B.1	1	2	1	ERR4891916
None	None	5006	100.00	NA	NA	1	0	ERR4999251
None	None	5006	100.00	NA	NA	1	0	ERR4999255
None	None	5006	100.00	NA	NA	1	0	ERR4999275
None	None	5006	100.00	NA	NA	1	0	SRR12639958

```
rm(seq_info)
```


ERR 4759453
B.1.1.29
B.1.1.250
Others: 191
ERR 4869480
B.1.177
B.1. 177.22
Others: 16
ERR 4891889
B.1.98
B. 1.243
Others: 60
ERR 4891988
B
B.10
Others: 44
ERR 4892152
B.1.177
B.1. 177.7
Others 14
ERR 4892423
B.1.523
B.1
Others: 28
ERR 4893031
B.1.177
B.1.177.7
Others: 15
ERR 4893184
B.1.258
B.1.258.17
Others 19
ERR 4999282
B
B.54
Others: 50
ERR 5082556
B.1.177.7
B.1.177
Othe 13
ERR 5082645
B.1.177
B.1.177.22
Others: 24
ERR 5082695
B.1.177
B.1.177.22
Others: 20
ERR 5082711
B.1.177
B.1.177.22
Others: 22
ERR 5082712
B.1.177.7
B.1.177
Othe 12
SRR 12762573
A
B
A.5
Other 11
SRR 13020990
A.2.2
A.2
Others: 104