

# Results of PangoVis

Devan Becker

2021-04-10

## 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 _ACCESSIONNUMBER.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 0.950     19      1      1      1 B.1.222
## 2 ERR4364007 0.821     84      1      1      1 B.1.1.29
## 3 ERR4664555 0.963     27      1      1      1 B.1.1.253
## 4 ERR4667618 0.993      7      1      1      1 B.1.1.315
## 5 ERR4692364 0.914     55      1      1      1 B.1
## 6 ERR4693034 0.869     77      1      1      1 B.1.1.310
## 7 ERR4693061 0.954     19      1      1      1 B.23
## 8 ERR4693079 0.874     99      1      1      1 B.1.1.310
## 9 ERR4693537 0.973      9      1      1      1 B.1.177.7
## 10 ERR4693605 0.968     15      1      1      1 B.1.177.3
## # ... with 70 more rows

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

## [1] 0.09735665
## [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	4756	B.1	108	19	3	ERR4363387
B.1.1.29	B.1.1.29	4112	B.1.1.44	130	84	27	ERR4364007
B.1.1.253	B.1.1.253	4819	B.1	52	27	9	ERR4664555
B.1.1.315	B.1.1.315	4969	B.1	22	7	2	ERR4667618
B.1	B.1	4577	B.1.400	92	55	12	ERR4692364
B.1.1.310	B.1.1.310	4352	B.1.1.59	185	77	31	ERR4693034
B.23	B.23	4776	B.48	112	19	6	ERR4693061
B.1.1.310	B.1.1.310	4375	B.1.1.29	212	99	40	ERR4693079
B.1.177.7	B.1.177.7	4871	B.1.177	125	9	4	ERR4693537
B.1.177.3	B.1.177.3	4847	B.1.177	62	15	1	ERR4693605
B.1.36	B.1.36	4724	B.1.36.9	71	26	8	ERR4891711
B	B	4408	B.23	189	30	7	ERR4891715
B.1.177.6	B.1.177.6	4944	B.1.177.9	31	8	1	ERR4891805
B.1	B.1	3736	B.1.243	91	216	49	ERR4891841
B.1.1.216	B.1.1.216	4179	B.1	135	132	48	ERR4891863
B.1.98	B.1.98	4037	B.1	311	75	36	ERR4891889
B.1.1.304	B.1.1.304	4856	B.1.1.4	16	34	11	ERR4891898
None	None	5002	B.1	1	2	1	ERR4891916
B	B	3401	B.10	609	79	32	ERR4891988
B.1.1.307	B.1.1.307	4975	B.1	10	14	8	ERR4892048
B.1.1.29	B.1.1.29	1384	B.1.1.59	239	213	41	ERR4892066
B.39	B.39	4817	B	113	22	8	ERR4892112
B.1.177	B.1.177	4372	B.1.177.7	427	15	3	ERR4892152

called	mode	mode_count	runner_up	runner_up_count	unique	atoms	taxon
B.1.177.17	B.1.177.17	4967	B.1.177	39	2	0	ERR4892200
B.1.1.216	B.1.1.216	4528	B.1.1.29	38	76	18	ERR4892203
B.1.1.162	B.1.1.162	3001	B.1.1.209	112	198	66	ERR4892293
B.1.177	B.1.177	4529	B.1.177.23	181	17	0	ERR4892339
B.40	B.40	4974	B	13	7	1	ERR4892386
B.1.177	B.1.177	4830	B.1.177.22	81	17	1	ERR4892392
B.1.523	B.1.523	4501	B.1	228	30	6	ERR4892423
B.1	B.1	4378	B.1.88	140	50	16	ERR4893013
B.1.177	B.1.177	3779	B.1.177.7	985	13	0	ERR4893031
B.1.1.307	B.1.1.307	4885	B.1	46	40	22	ERR4893033
B.1.1.29	B.1.1.29	2445	B.1.1.39	101	231	29	ERR4893037
B.1.177	B.1.177	4819	B.1.177.22	98	12	4	ERR4893080
B.1.177.16	B.1.177.16	4964	B.1.177	29	4	1	ERR4893138
B.1.258	B.1.258	4256	B.1.258.17	468	24	6	ERR4893184
B.1.1.216	B.1.1.216	4441	B.1.1.208	52	100	43	ERR4893186
B.1.177	B.1.177	4844	B.1.177.23	68	17	6	ERR4893197
B.1.177	B.1.177	4724	B.1.177.22	131	13	3	ERR4893242
B.1.1.307	B.1.1.307	4957	B.1	20	15	10	ERR4893353
B.1.523	B.1.523	4598	B.1	122	32	7	ERR4893393
B.1.177	B.1.177	4779	B.1.177.22	133	21	7	ERR5062571
B.1.177.19	B.1.177.19	4914	B.1.177	64	8	1	ERR5063165
B.1.177	B.1.177	4783	B.1.177.22	115	18	3	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	4829	B.1.247	75	43	14	ERR5069624
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5069871
B.1.177	B.1.177	4841	B.1.177.22	115	14	2	ERR5070060
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5070294
B.1.160	B.1.160	4663	B.1.160.8	184	17	7	ERR5074314
B.1.177.19	B.1.177.19	4875	B.1.177	74	13	2	ERR5076163
B.1.177.19	B.1.177.19	4852	B.1.177	113	12	6	ERR5076748
B.1.177	B.1.177	4802	B.1.177.22	114	19	4	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	4785	B.1	125	15	5	ERR5080897
B.1.1.311	B.1.1.311	4876	B.1	54	14	8	ERR5080913
B.1.177	B.1.177	4710	B.1.177.22	130	22	4	ERR5080918
B.1.177	B.1.177	4667	B.1.177.22	182	20	5	ERR5081301
B.1.177.4	B.1.177.4	4983	B.1.177.2	11	13	10	ERR5081304
B.1.177.15	B.1.177.15	4845	B.1.177	130	11	3	ERR5081316
B.1.177.7	B.1.177.7	3702	B.1.177	1256	12	1	ERR5082556
B.1.160	B.1.160	4857	B.1.160.5	61	15	5	ERR5082569
B.1.177	B.1.177	4620	B.1.177.22	240	21	2	ERR5082580
B.1.177.4	B.1.177.4	4980	B.1.177.2	15	8	5	ERR5082590
B.1.1.7	B.1.1.7	5006	NA	NA	1	0	ERR5082610
B.1.177.7	B.1.177.7	4879	B.1.177	113	10	4	ERR5082630
B.1.177	B.1.177	4014	B.1.177.22	391	26	9	ERR5082645
B.1.1.315	B.1.1.315	4853	B.1	61	15	6	ERR5082664
B.1.177.6	B.1.177.6	4875	B.1.177	80	11	2	ERR5082674
B.1.177	B.1.177	4463	B.1.177.22	227	19	2	ERR5082695
B.1.1.315	B.1.1.315	4403	B.1.1.281	244	39	18	ERR5082696
B.1.177	B.1.177	3586	B.1.177.22	793	30	7	ERR5082711

called	mode	mode_count	runner_up	runner_up_count	unique	atoms	taxon
B.1.177.7	B.1.177.7	3240	B.1.177	1704	15	1	ERR5082712
None	None	5006	NA	NA	1	0	SRR12639958
A	A	5708	B	2156	16	0	SRR12762573
A.2.2	A.2.2	2950	A.2	730	108	47	SRR13020990
A.1	A.1	8879	B.40	57	20	6	SRR13092002

rm(seq\_info)

