

all_strain_genome_stats

2023-07-12

Assembly stats for 13 *Emiliania huxleyi* genomes

```
genstats <- list()
gc <- list()
stats <- list()
strains <- as.character(read.csv('data/strains.csv',header=FALSE))
for (strain in strains){
  genstats[[strain]] <- read.table(paste0("data/2023-genome-stats/stats_after_pilon_round_2_decontam/",
                                           strain,".genstats.txt"),sep="\t",header=F)
  colnames(genstats[[strain]]) <- c('Contig_name','Avg_fold', 'Length', 'Ref_GC',
                                     'Covered_percent', 'Covered_bases','Plus_reads',
                                     'Minus_reads', 'Read_GC', 'Median_fold', 'Std_Dev')
  gc[[strain]] <- read.table(paste0("data/2023-genome-stats/stats_after_pilon_round_2_decontam/",
                                    strain,".gcscaffold.txt"),sep="\t",header=F)
  colnames(gc[[strain]]) <- c('Contig_name', 'Length', 'A', 'C', 'G', 'T', 'N',
                               'IUPAC','Other', 'GC')
  stats[[strain]] <- inner_join(genstats[[strain]],gc[[strain]],by="Contig_name")
}
```

Read in assembly stats data

```
n50s <- c()
lengths <- c()
contig_counts <- c()
mins <- c()
maxs <- c()
for (strain in strains){
  contig_lengths <- stats[[strain]]$Length.x
  contig_counts <- c(contig_counts,length(contig_lengths))
  total_assembled_length=sum(contig_lengths)
  contig_lengths <- sort(contig_lengths,decreasing=TRUE)

  sum <- 0
  for (length in contig_lengths){
    sum <- sum+length
    if (sum>=total_assembled_length/2){
      n50s <- c(n50s,as.numeric(length))
      lengths <- c(lengths,total_assembled_length)
```

```

    maxs <- c(maxs,contig_lengths[1])
    mins <- c(mins,contig_lengths[length(contig_lengths)])
    break
  }
}

}

global_stats <- data.frame(n50s,lengths,contig_counts,mins,maxs)
rownames(global_stats) <- strains
colnames(global_stats) <- c('N50','Total assembled length','Contig count',
                           'Min contig length','Max contig length')

```

Calculate globals assembly stats including N50 and total assembled length

```

name_translation <- read.table('data/genomescope/illumina-run-conversions.txt',sep=' ')
temp <- name_translation$V2
names(temp) <- name_translation$V1
name_translation <- temp

```

```

global_stats$genome_haploid_length <- seq(1,nrow(global_stats))
global_stats$genome_unique_length <- seq(1,nrow(global_stats))
for (folder in list.dirs('data/genomescope/')){
  if (grepl('HA',folder)){
    key <- str_split(folder,'_',simplify=TRUE)[,1]
    key <- str_split(key,'/',simplify=TRUE)[,4]
    temp <- read.csv(paste0(folder,"/summary.txt_fixed.csv"))
    global_stats[name_translation[key],'genome_haploid_length'] <-
      mean(temp[2,'min'],temp[2,'max'])
    global_stats[name_translation[key],'genome_unique_length'] <-
      mean(temp[4,'min'],temp[4,'max'])
  }
}
#print(temp)
print(global_stats)

```

Add in estimated genome size stats, calculated using Genomescope

	N50	Total assembled length	Contig count	Min contig length
## CCMP371	167293	299709025	5373	671
## CCMP375	82247	247806427	8312	505
## CCMP377	70514	297357770	9460	493
## CCMP1280	104319	241936265	7168	510
## RCC874	3015506	152968590	396	522
## RCC914	63139	210245484	9395	508
## RCC1222	55476	179814701	10077	160
## RCC1239	364457	171886557	2195	506
## RCC1256	43379	236745271	10971	393
## RCC3492	70125	211559349	9734	496

```

## RCC3963    146371          281844499      5463        544
## RCC6071    511805          200137934      1929        545
## RCC6856    78893           333916182      9838        528
## RCC1212    NA              NA            NA          NA
## RCC1215    NA              NA            NA          NA
##          Max contig length genome_haploid_length genome_unique_length
## CCMP371      1476961         98806191       72440020
## CCMP375      1947962         103218139      69423426
## CCMP377      606376          138388017      76045285
## CCMP1280     4348365         111401730      64523269
## RCC874       7853977         118376237      91959391
## RCC914       653764          86242315       53701350
## RCC1222     1451258          107508536      66175986
## RCC1239     1376033          116873978      83660661
## RCC1256     579311          93127971       69271484
## RCC3492     1245722          77328042       62355972
## RCC3963     1638904          107212419      77275117
## RCC6071     2885505          131960871      81986586
## RCC6856     509900           123410186      76317769
## RCC1212    NA              123605384       80280702
## RCC1215    NA              -1             -1

```

```

global_stats$Strain <- rownames(global_stats)
ggplot(global_stats,aes(x=N50,y=genome_haploid_length,label=Strain))+ 
  geom_point()+
  theme_bw()+
  scale_x_log10()+
  geom_smooth(method=lm,colour="black")+
  ylab("Haploid genome length")+
  xlab("Contig N50")+
  labs(caption=str_wrap("Figure 1. Relationship between predicted haploid genome
                        length and contig N50 for 13 Emiliania huxleyi
                        genomes.",75))+
  geom_text_repel(size=3)+
  theme(plot.caption = element_text(hjust = 0,size=12))

```

Genomescope predicted haploid genome length vs N50

```

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite values ('stat_smooth()').

## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 2 rows containing missing values ('geom_point()').

```

```
## Warning: Removed 2 rows containing missing values ('geom_text_repel()').
```

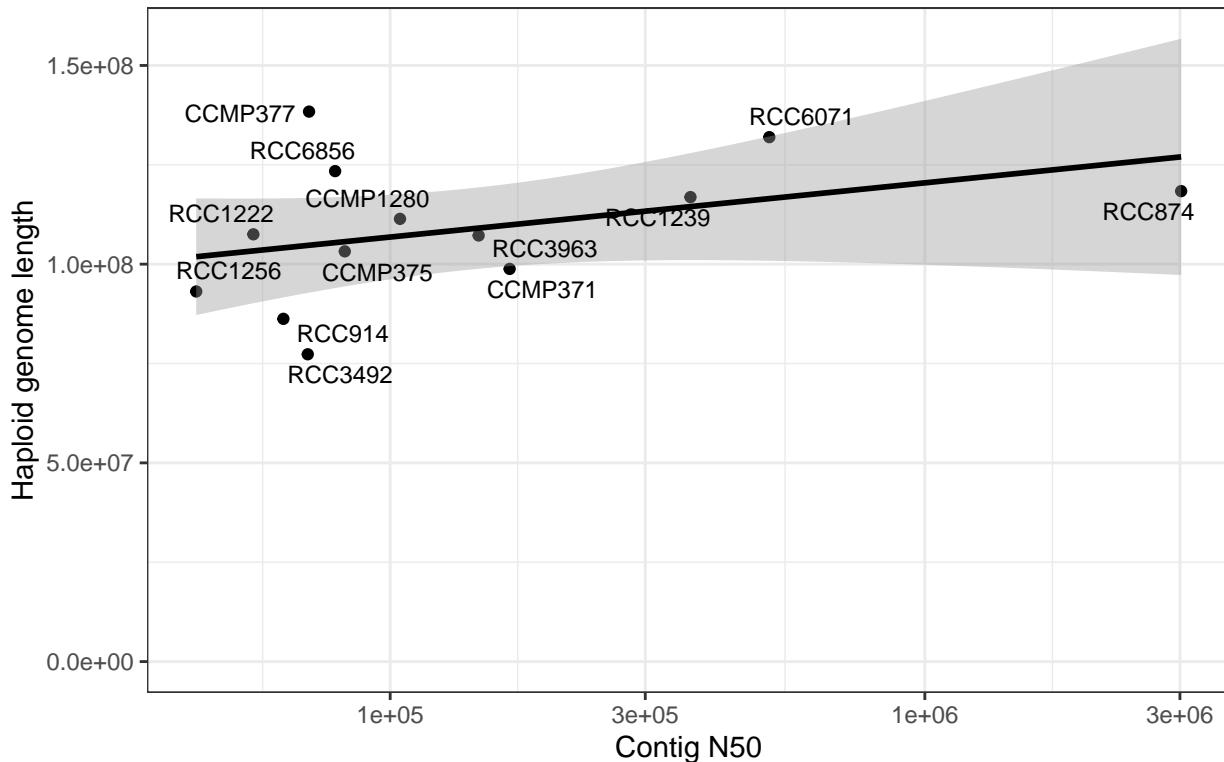


Figure 1. Relationship between predicted haploid genome length and contig N50 for 13 *Emiliania huxleyi* genomes.

```
ggsave("plots/haploid_genome_length_vs_N50.png")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite values ('stat_smooth()').

## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 2 rows containing missing values ('geom_point()').

## Warning: Removed 2 rows containing missing values ('geom_text_repel()').
```

```
global_stats$Strain <- rownames(global_stats)
ggplot(global_stats,aes(x=N50,y=genome_unique_length,label=Strain))+
```

```

geom_point()+
theme_bw()+
scale_x_log10()+
geom_smooth(method=lm, colour="black")+
ylab("Unique genome length")+
xlab("Contig N50")+
labs(caption=str_wrap("Figure 2. Relationship between predicted unique genome
length and contig N50 for 13 Emiliania huxleyi
genomes.", 75))+
geom_text_repel(size=3)+
theme(plot.caption = element_text(hjust = 0, size=12))

```

Genomescope predicted unique genome length vs N50

```

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite values ('stat_smooth()').

## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 2 rows containing missing values ('geom_point()').

## Warning: Removed 2 rows containing missing values ('geom_text_repel()').

```

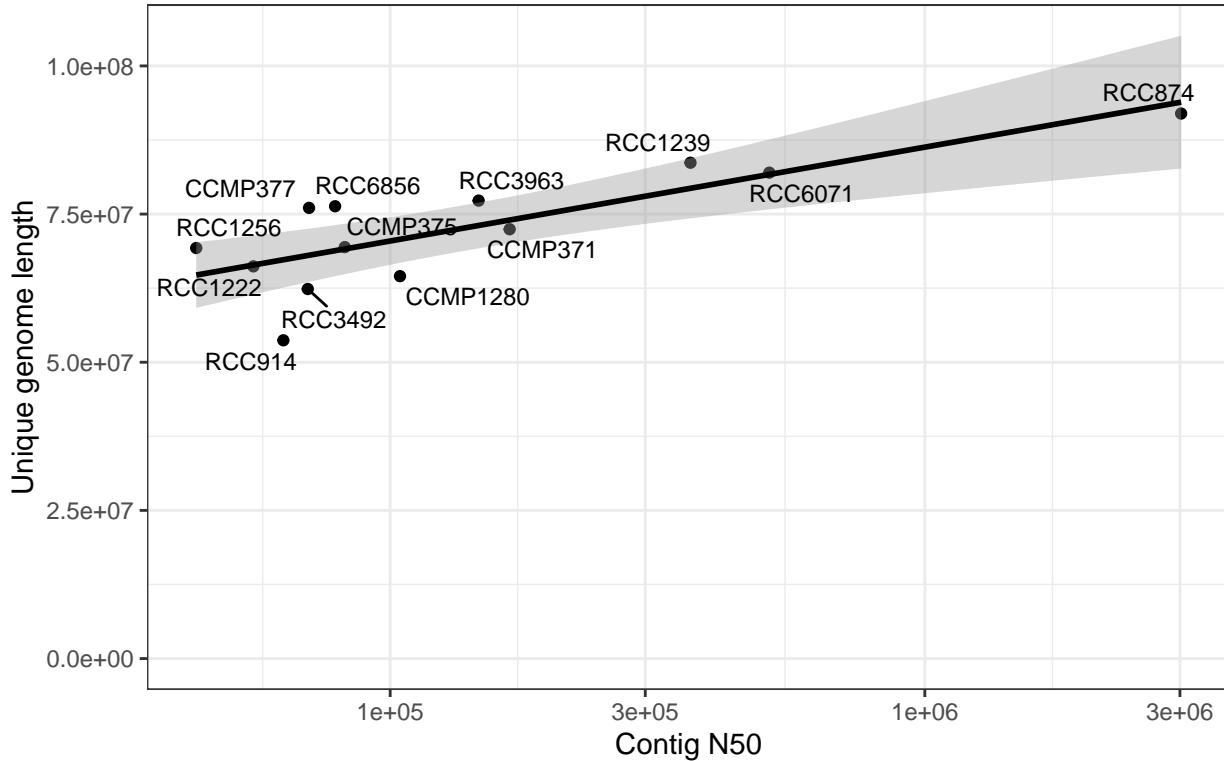


Figure 2. Relationship between predicted unique genome length and contig N50 for 13 *Emiliania huxleyi* genomes.

```

ggsave("plots/unique_genome_length_vs_N50.png")

## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite values ('stat_smooth()').

## Warning: The following aesthetics were dropped during statistical transformation: label
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?

## Warning: Removed 2 rows containing missing values ('geom_point()').

## Warning: Removed 2 rows containing missing values ('geom_text_repel()').

```

```

stats_joined <- ldply(stats, rbind)
ggplot(stats_joined, aes(x=Length.x)) + geom_histogram() + scale_x_log10() +
  theme_bw() +
  xlab("Contig size") +

```

```

ylab("Count") + facet_wrap(vars(.id)) +
  labs(caption = str_wrap("Figure 3. Contig length distribution for 13 Emiliania huxleyi genomes.", 75)) +
  theme(plot.caption = element_text(hjust = 0, size = 12))

```

Contig length distribution for each strain

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

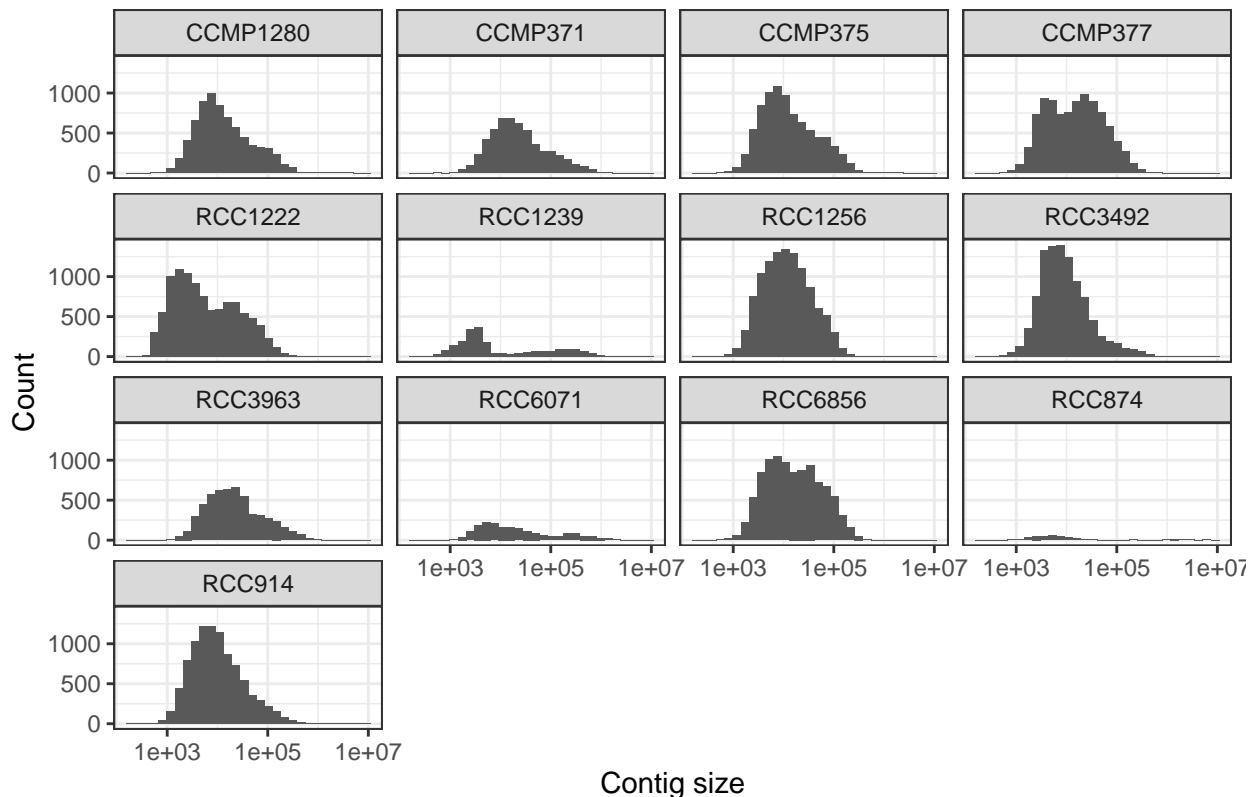


Figure 3. Contig length distribution for 13 *Emiliania huxleyi* genomes.

```
ggsave("plots/contig_length_distributions.png")
```

Saving 6.5 x 4.5 in image
'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

```

ggplot(data = stats_joined, aes(x = GC, y = Avg_fold)) +
  geom_point(size = 0.5) + ylim(0, 300) + facet_wrap(vars(.id)) +
  theme_bw() +
  ylab("Contig GC%") +
  xlab("Average Fold Coverage") +
  labs(caption = str_wrap("Figure 4. Contig Fold coverage vs GC percentage for 13 Emiliania huxleyi genomes.", 75)) +
  theme(plot.caption = element_text(hjust = 0, size = 12))

```

Scatterplots of fold coverage vs GC percentage (each point represents a contig)

```
## Warning: Removed 150 rows containing missing values ('geom_point()').
```

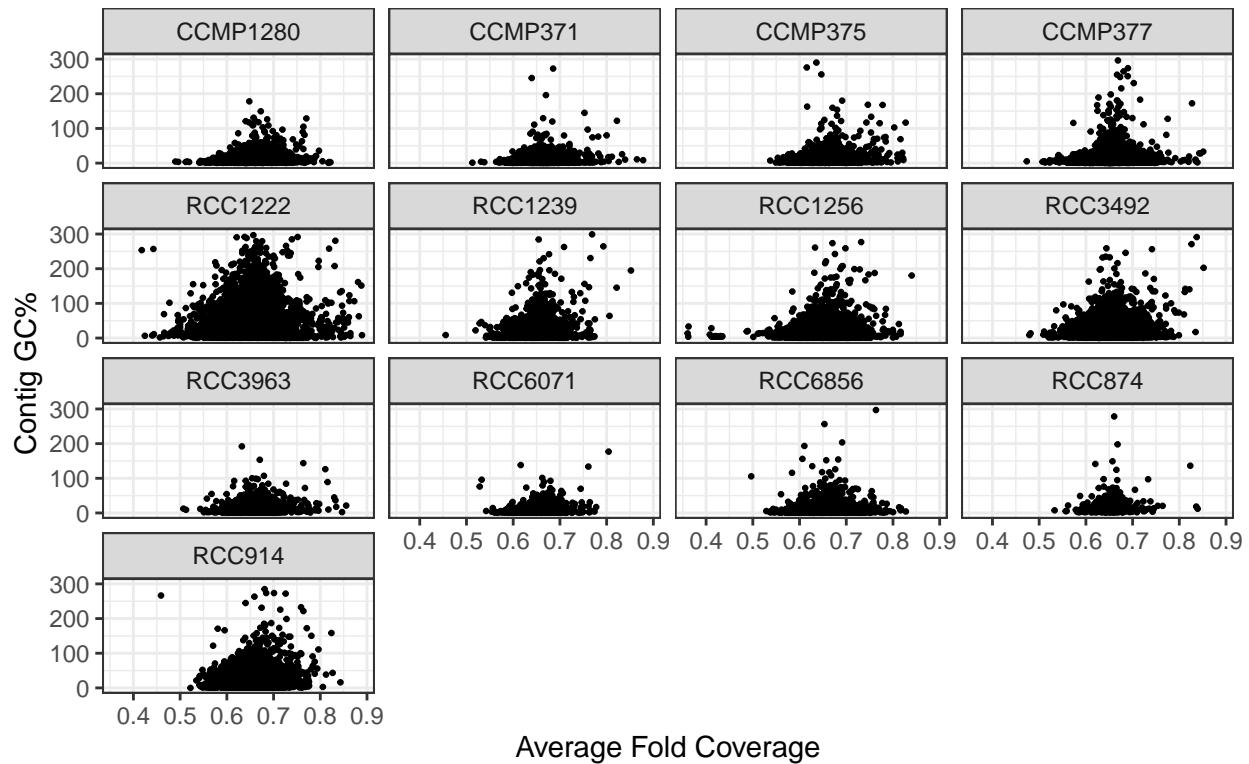


Figure 4. Contig Fold coverage vs GC percentage for 13 *Emiliana huxleyi* genomes.

```
ggsave("plots/coverage_vs_GC.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 150 rows containing missing values ('geom_point()').
```