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')

    #genstats[[strain]]$Contig_name <- str_split(genstats[[strain]]$Contig_name,regex("_1$"),simplify =

  stats[[strain]] <- inner_join(genstats[[strain]],gc[[strain]],by="Contig_name")


}
```

```
genstats <- list()
gc <- list()
rmdup_stats <- list()
for (strain in strains){
  genstats[[strain]] <- read.table(paste0("data/stats_after_pilon_round_2_decontam_rmdup/"
                                          ,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/stats_after_pilon_round_2_decontam_rmdup/"
                                    ,strain,".gcscaffold.txt"),sep="\t",header=F)
  colnames(gc[[strain]]) <- c('Contig_name',    'Length',   'A',    'C',    'G',    'T',    'N',
                              'IUPAC','Other',  'GC')
```

```
    #genstats[[strain]]$Contig_name <- str_split(genstats[[strain]]$Contig_name,regex("_1$"),simplify =

  rmdup_stats[[strain]] <- inner_join(genstats[[strain]],gc[[strain]],by="Contig_name")


}
```

Read in assembly stats data for non-dedupped assemblies

```
n50s <- c()
lengths <- c()
contig_counts <- c()
mins <- c()
maxs <- c()
l50s <- 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
count <- 0
for (length in contig_lengths){
  sum <- sum+length
  count <- count+1
  if (sum>=total_assembled_length/2){
    l50s <- c(l50s,count)
    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,l50s,lengths,contig_counts,mins,maxs)
rownames(global_stats) <- strains
colnames(global_stats) <- c('N50','L50',  'Total assembled length','Contig count',
                            'Min contig length','Max contig length')
```

Calculate globals assembly stats including N50 and total assembled length

```
n50s <- c()
lengths <- c()
contig_counts <- c()
```

```
l50s <- c()


for (strain in strains){
contig_lengths <- rmdup_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
count <- 0
for (length in contig_lengths){
  sum <- sum+length
  count <- count+1
  if (sum>=total_assembled_length/2){
    l50s <- c(l50s,count)
    n50s <- c(n50s,as.numeric(length))
    lengths <- c(lengths,total_assembled_length)
    break
  }
}

}
temp <- data.frame(n50s,l50s,lengths,contig_counts)
rownames(temp) <- strains
colnames(temp) <- c('rmdup_N50','rmdup_L50',  'rmdup Total assembled length','rmdup Contig count')
global_stats <- cbind(global_stats,temp)
global_stats
```

Add in rmdup stats

```
##              N50  L50 Total assembled length Contig count Min contig length
## CCMP371   167293  442              299709025         5373               671
## CCMP375    82247  791              247806427         8312               505
## CCMP377    70514 1125              297357770         9460               493
## CCMP1280  104319  573              241936265         7168               510
## RCC874   3015506   16              152968590          396               522
## RCC914     63139  769              210245484         9395               508
## RCC1222    55476  873              179814701        10077               160
## RCC1239   364457  140              171886557         2195               506
## RCC1256    43379 1463              236745271        10971               393
## RCC3492    70125  586              211559349         9734               496
## RCC3963   146371  484              281844499         5463               544
## RCC6071   511805  108              200137934         1929               545
## RCC6856    78893 1210              333916182         9838               528
##          Max contig length rmdup_N50 rmdup_L50 rmdup Total assembled length
## CCMP371            1476961    175654       255                     178311892
## CCMP375            1947962     87072       535                     184716260
## CCMP377             606376     78015       871                     252888127
## CCMP1280           4348365    113435       361                     176639739
## RCC874             7853977   3369442        15                     152306371
## RCC914              653764     77641       462                     147848163
## RCC1222            1451258     55027       677                     145640905
```

```
## RCC1239            1376033    370740       138                    170164584
## RCC1256             579311     44965      1147                    198119043
## RCC3492            1245722    106475       366                    171317621
## RCC3963            1638904    158060       295                    183446030
## RCC6071            2885505    546189       101                    192603649
## RCC6856             509900     83052       795                    238298182
##          rmdup Contig count
## CCMP371                3749
## CCMP375                7179
## CCMP377                8513
## CCMP1280               6028
## RCC874                  383
## RCC914                 7788
## RCC1222                9374
## RCC1239                2174
## RCC1256                9915
## RCC3492                8474
## RCC3963                4042
## RCC6071                1770
## RCC6856                8184
```

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

Add in estimated genome size stats, calculated using Genomescope

```
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]
  if (sum(grepl(name_translation[key],rownames(global_stats)))==1){
  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)
```

```
##              N50  L50 Total assembled length Contig count Min contig length
## CCMP371   167293  442              299709025         5373                671
## CCMP375    82247  791              247806427         8312                505
## CCMP377    70514 1125              297357770         9460                493
## CCMP1280  104319  573              241936265         7168                510
## RCC874   3015506   16              152968590          396                522
## RCC914     63139  769              210245484         9395                508
## RCC1222    55476  873              179814701        10077                160
## RCC1239   364457  140              171886557         2195                506
## RCC1256    43379 1463              236745271        10971                393
## RCC3492    70125  586              211559349         9734                496
## RCC3963   146371  484              281844499         5463                544
## RCC6071   511805  108              200137934         1929                545
## RCC6856    78893 1210              333916182         9838                528
##          Max contig length rmdup_N50 rmdup_L50 rmdup Total assembled length
## CCMP371            1476961    175654       255                     178311892
## CCMP375            1947962     87072       535                     184716260
## CCMP377             606376     78015       871                     252888127
## CCMP1280           4348365    113435       361                     176639739
## RCC874             7853977   3369442        15                     152306371
## RCC914              653764     77641       462                     147848163
## RCC1222            1451258     55027       677                     145640905
## RCC1239            1376033    370740       138                     170164584
## RCC1256             579311     44965      1147                     198119043
## RCC3492            1245722    106475       366                     171317621
## RCC3963            1638904    158060       295                     183446030
## RCC6071            2885505    546189       101                     192603649
## RCC6856             509900     83052       795                     238298182
##          rmdup Contig count genome_haploid_length genome_unique_length
## CCMP371                3749              98806191             72440020
## CCMP375                7179             103218139             69423426
## CCMP377                8513             138388017             76045285
## CCMP1280               6028             111401730             64523269
## RCC874                  383             118376237             91959391
## RCC914                 7788              86242315             53701350
## RCC1222                9374             107508536             66175986
## RCC1239                2174             116873978             83660661
## RCC1256                9915              93127971             69271484
## RCC3492                8474              77328042             62355972
## RCC3963                4042             107212419             77275117
## RCC6071                1770             131960871             81986586
## RCC6856                8184             123410186             76317769
```

```
global_stats$Strain <- rownames(global_stats)
colors <- c("Initial" = "black", "Rmdup" = "blue")

ggplot(global_stats,aes(x=N50,y=genome_haploid_length,label=Strain,color="Initial"))+
  geom_point()+
  geom_point(aes(x=rmdup_N50,y=genome_haploid_length,label=Strain, color = "Rmdup"))+
  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))+
  labs(color = "Legend")+   scale_color_manual(values = colors)
```

Genomescope predicted haploid genome length vs N50

```
## Warning in geom_point(aes(x = rmdup_N50, y = genome_haploid_length, label =
## Strain, : Ignoring unknown aesthetics: label
```

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

```
## 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?
```

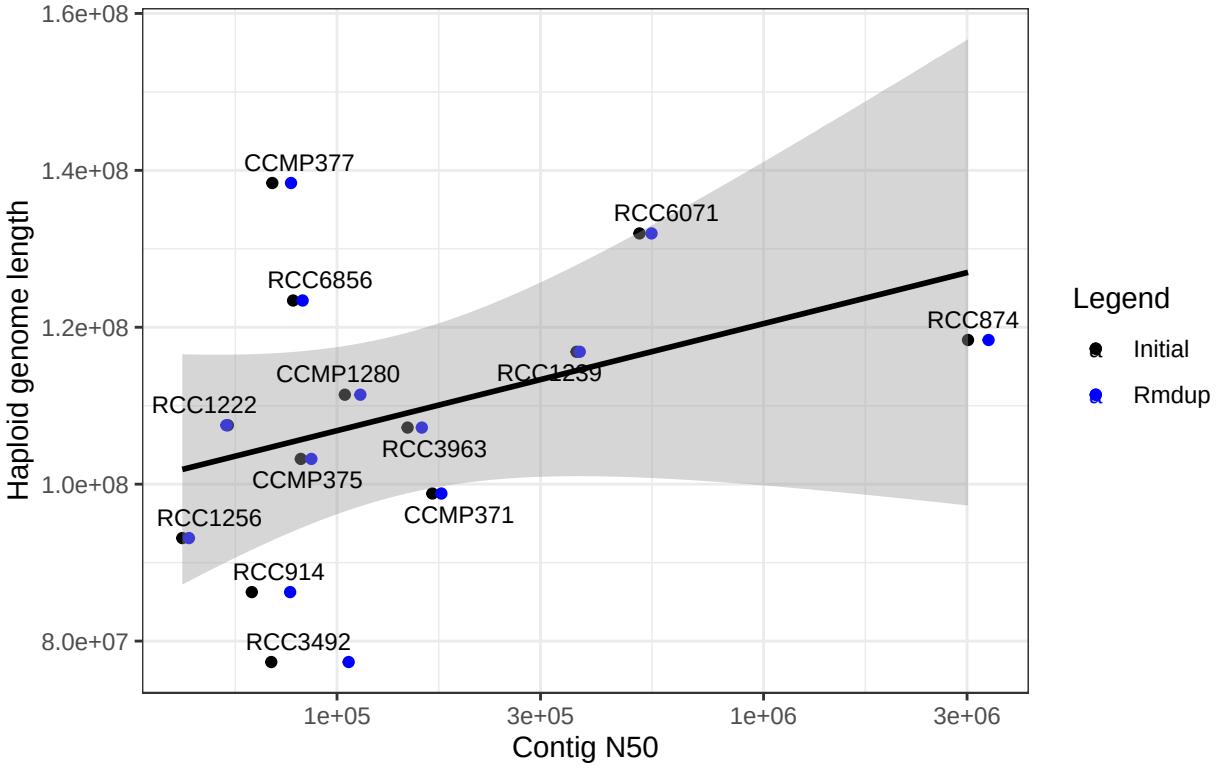


Figure 1. Relationship between predicted haploid genome length and contig N50 for 13 Emiliania huxleyi genomes.

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

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

```
## Warning: 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?
```

```
global_stats$Strain <- rownames(global_stats)
ggplot(global_stats,aes(x=N50,y=genome_unique_length,label=Strain,color="Initial"))+
  geom_point()+
  geom_point(aes(x=rmdup_N50,y=genome_unique_length,label=Strain, color = "Rmdup"))+
  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))+
  labs(color = "Legend")+
  scale_color_manual(values = colors)
```

Genomescope predicted unique genome length vs N50

```
## Warning in geom_point(aes(x = rmdup_N50, y = genome_unique_length, label =
## Strain, : Ignoring unknown aesthetics: label
```

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

```
## 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?
```

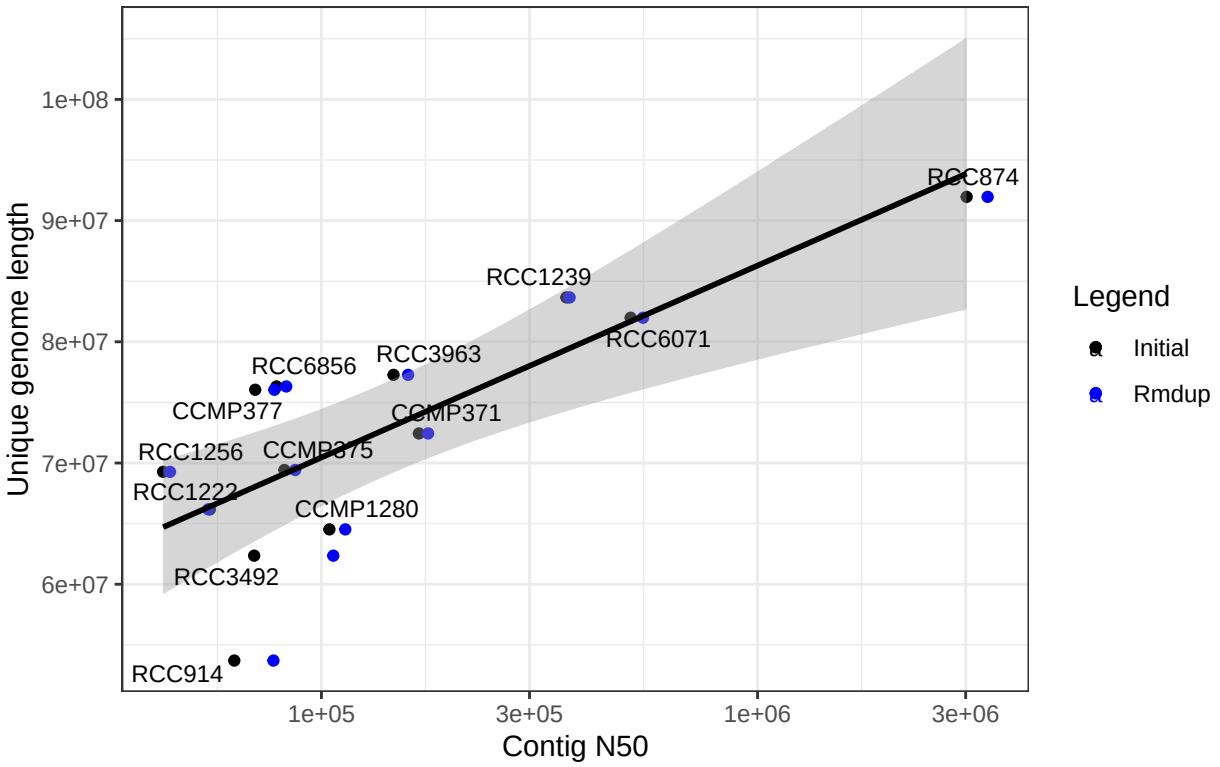


Figure 2. Relationship between predicted unique genome length and contig N50 for 13 Emiliania huxleyi genomes.

```
ggsave("plots/unique_genome_length_vs_N50.png")
```

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

```
## Warning: 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?
```

```
stats_joined <- ldply(stats,rbind)
initial <- ggplot(stats_joined,aes(x=Length.x))+geom_histogram()+scale_x_log10()+
           theme_bw()+
  xlab("Contig size")+
  ylab("Count")+facet_wrap(vars(.id))+
  theme(plot.caption = element_text(hjust = 0,size=12), axis.text.x = element_text(angle = 90))

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

```
  ylab("Count")+facet_wrap(vars(.id))+
  theme(plot.caption = element_text(hjust = 0,size=12),
        axis.text.x = element_text(angle = 90))

combined <- plot_grid(initial,rmdup, labels = c('Initial', 'Rmdup'), label_size = 12,hjust=-5,vjust=30)
```

Contig length distribution for each strain

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

```
title <- ggdraw() + draw_label("Figure 3. Contig length distribution for 13 Emiliania huxleyi genomes",
plot_grid(combined,title, ncol=1, rel_heights=c(1, 0.1))
```

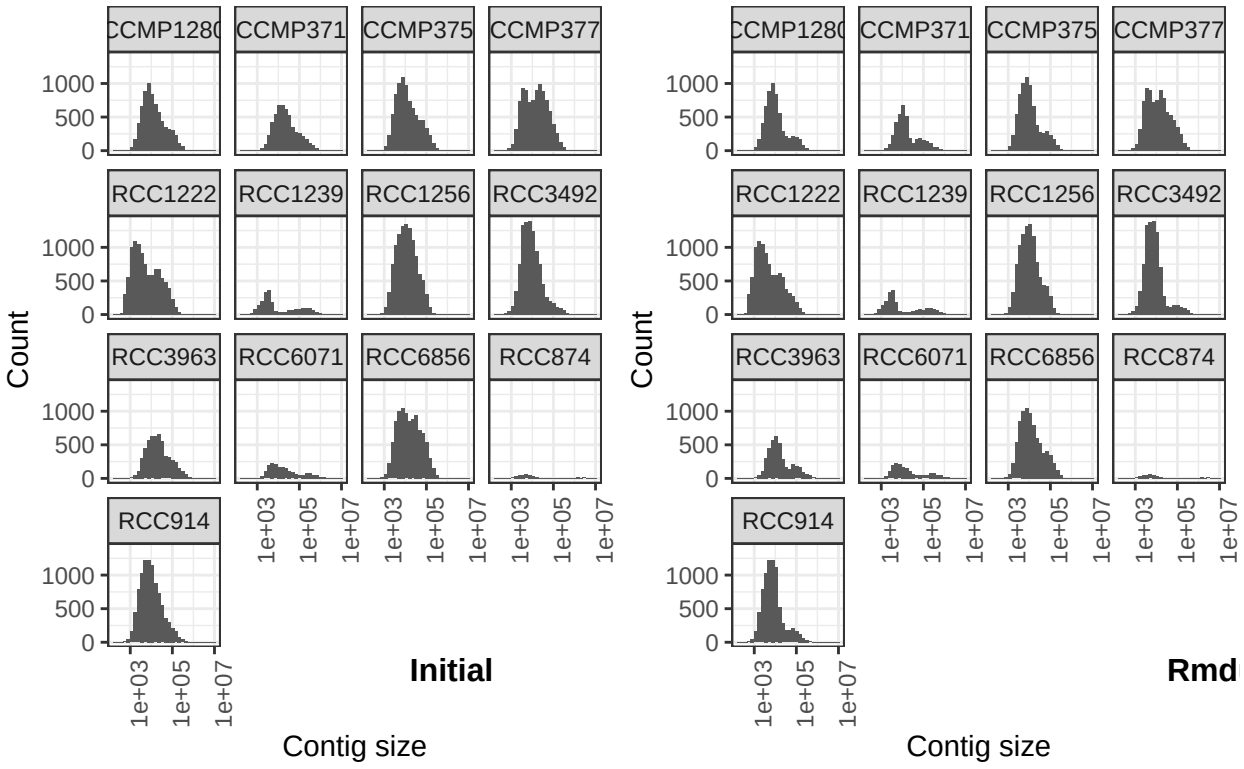


Figure 3. Contig length distribution for 13 Emiliania huxleyi genomes

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

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

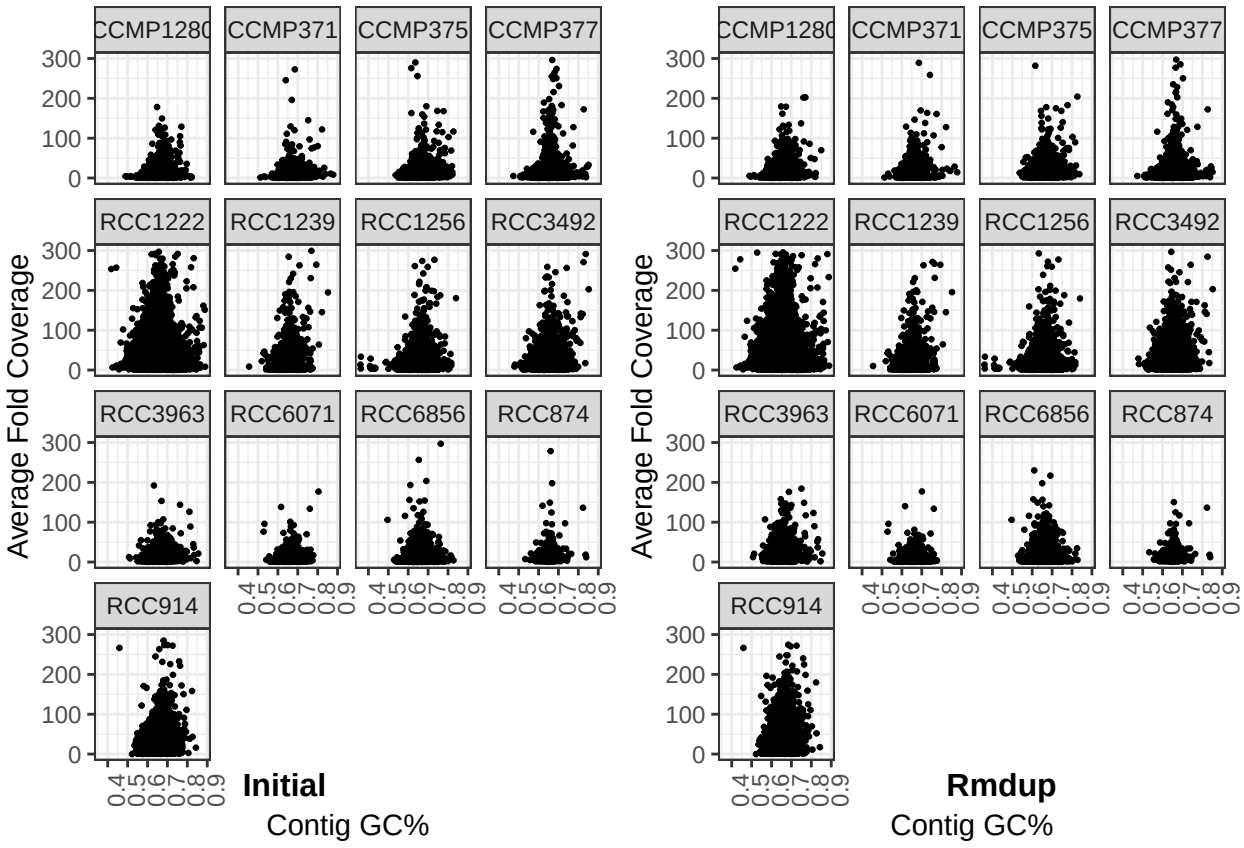
```
stats_joined <- ldply(stats,rbind)
initial <- ggplot(data = stats_joined, aes(x = GC,y=Avg_fold)) +
  geom_point(size=0.5)+
```

```
  ylim(0,300)+
  #scale_y_log10()+
  facet_wrap(vars(.id))+
            theme_bw()+
  xlab("Contig GC%")+
  ylab("Average Fold Coverage")+
  theme(plot.caption = element_text(hjust = 0,size=12), axis.text.x = element_text(angle = 90))


stats_joined <- ldply(rmdup_stats,rbind)
rmdup <- ggplot(data = stats_joined, aes(x = GC,y=Avg_fold)) +
  geom_point(size=0.5)+
  ylim(0,300)+
  #scale_y_log10()+
  facet_wrap(vars(.id))+
            theme_bw()+
  xlab("Contig GC%")+
  ylab("Average Fold Coverage")+
  theme(plot.caption = element_text(hjust = 0,size=12),
        axis.text.x = element_text(angle = 90))

plot_grid(initial,rmdup, labels = c('Initial', 'Rmdup'), label_size = 12,hjust=-3,vjust=35)
```

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

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

```
## Warning: Removed 174 rows containing missing values (‘geom_point()‘).
```




```
combined <- plot_grid(initial,rmdup, labels = c('Initial', 'Rmdup'), label_size = 12,hjust=-5,vjust=30)
```

```
## Warning: Removed 150 rows containing missing values (‘geom_point()‘).
## Removed 174 rows containing missing values (‘geom_point()‘).
```

```
title <- ggdraw() + draw_label("Figure 4. Contig fold coverage vs GC percentage for 13 Emiliania huxley
plot_grid(combined,title, ncol=1, rel_heights=c(1, 0.1))
```

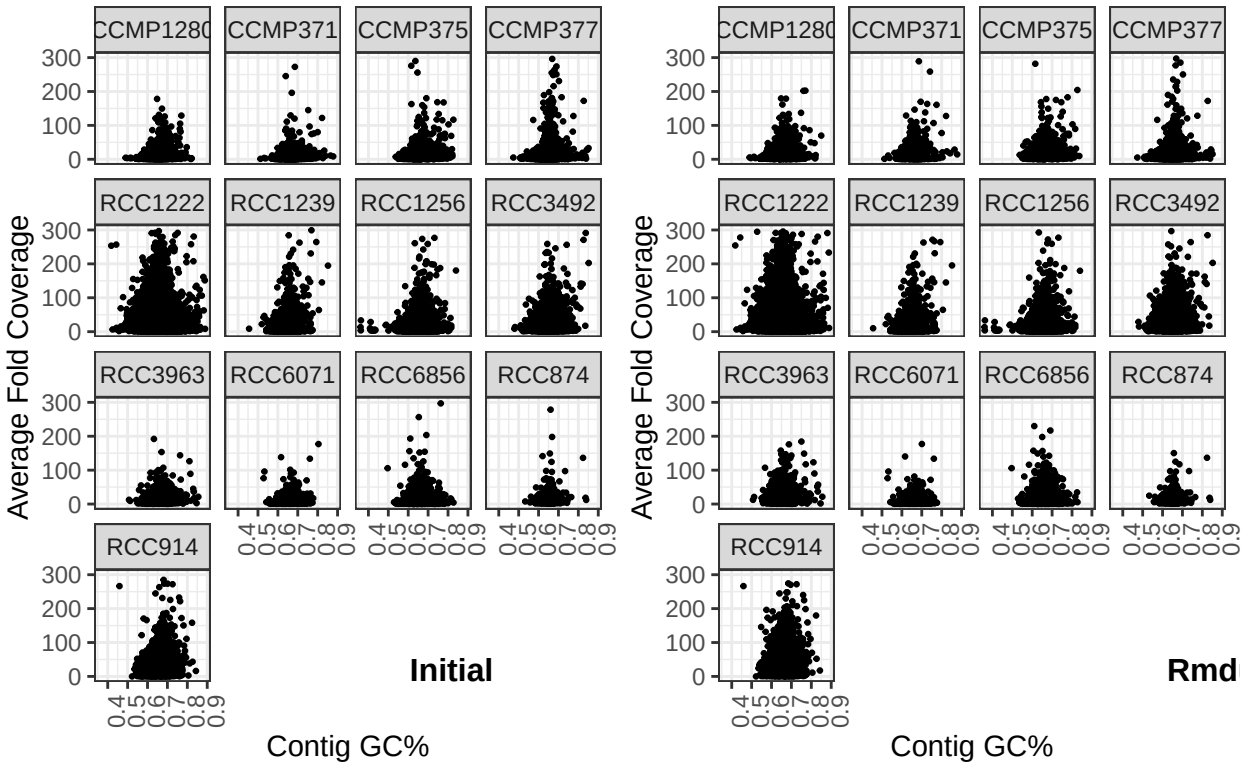


Figure 4. Contig fold coverage vs GC percentage for 13 Emiliania huxleyi genomes

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

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

```
global_stats$rmdup_total_assembled_length <- global_stats[,"rmdup Total assembled length"]
global_stats$total_assembled_length <- global_stats[,"Total assembled length"]

ggplot(global_stats,aes(x=rmdup_total_assembled_length,y=genome_unique_length,label=Strain,color="Rmdup
  geom_point()+
  geom_point(aes(x=total_assembled_length,y=genome_unique_length,label=Strain, color = "Initial"))+
  theme_bw()+
  geom_abline(color="black",slope=1,linetype = "dashed")+
    geom_abline(color="black",slope=0.5,linetype = "dashed")+
  xlim(0,3e+8)+
  ylim(0,1.5e+8)+
  ylab("Unique genome length")+
  xlab("Total assembled length")+
  labs(caption=str_wrap("Figure 5.  Relationship between predicted unique genome
                        length and total assembled length for 13 Emiliania huxleyi
                        genomes.",75))+
    geom_text_repel(size=3,nudge_x=10,min.segment.length=0)+
    annotate("text",x=1e8,y=1e8,label="y=x",color="red",size=5)+
```

```
    annotate("text",x=2.5e8,y=1e8,label="y=0.5x",color="red",size=5)+
  theme(plot.caption = element_text(hjust = 0,size=12))+
  labs(color = "Legend")+
  scale_color_manual(values = colors)
```

Genomescope predicted unique genome length vs assembled length

```
## Warning in geom_point(aes(x = total_assembled_length, y = genome_unique_length,
## : Ignoring unknown aesthetics: label
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```

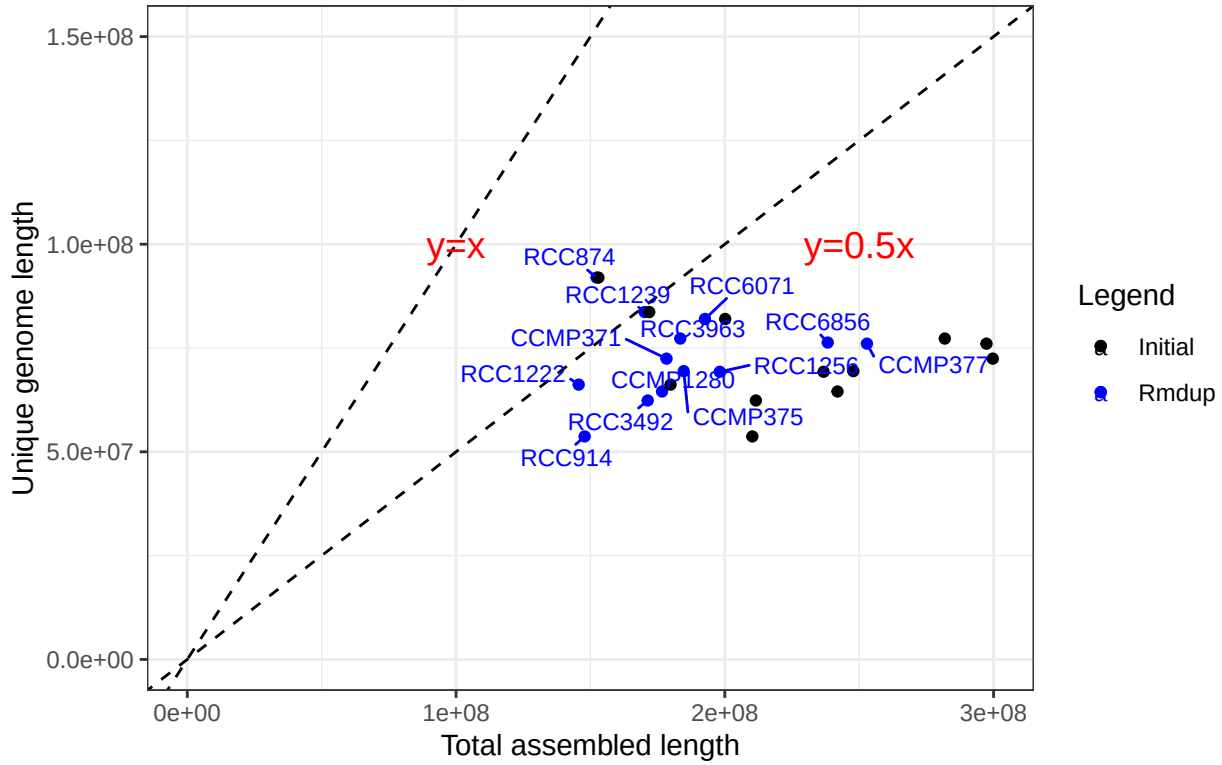


Figure 5. Relationship between predicted unique genome length and total assembled length for 13 Emiliania huxleyi genomes.

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

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

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```

```
ggplot(global_stats,aes(x=rmdup_total_assembled_length,y=genome_haploid_length,label=Strain,color="Rmdup
  geom_point()+
  geom_point(aes(x=total_assembled_length,y=genome_haploid_length,label=Strain, color = "Initial"))+
  theme_bw()+
  geom_abline(color="black",slope=1,linetype = "dashed")+
```

```
  geom_abline(color="black",slope=0.5,linetype = "dashed")+
 xlim(0,3e+8)+
 ylim(0,1.5e+8)+
 ylab("Haploid genome length")+
 xlab("Total assembled length")+
 labs(caption=str_wrap("Figure 6.  Relationship between predicted haploid genome
                       length and total assembled length for 13 Emiliania huxleyi
                       genomes.",75))+
   geom_text_repel(size=3,nudge_x=10,min.segment.length=0)+
   annotate("text",x=1e8,y=1e8,label="y=x",color="red",size=5)+
     annotate("text",x=1.3e8,y=5e7,label="y=0.5x",color="red",size=5)+
 theme(plot.caption = element_text(hjust = 0,size=12))+
 labs(color = "Legend")+
 scale_color_manual(values = colors)
```

```
## Warning in geom_point(aes(x = total_assembled_length, y =
## genome_haploid_length, : Ignoring unknown aesthetics: label
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```

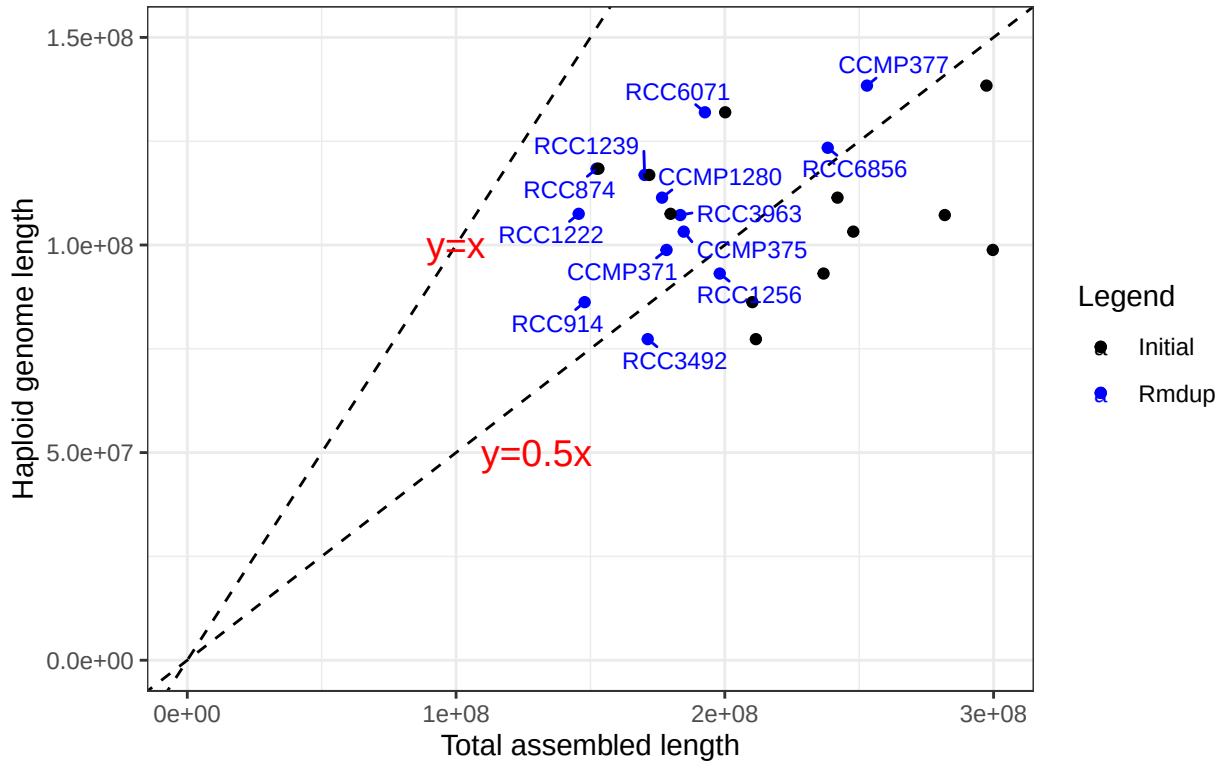


Figure 6. Relationship between predicted haploid genome length and total assembled length for 13 Emiliania huxleyi genomes.

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

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

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```