# Predicting Endosporulation Among Expanding Firmicutes Phylogeny

*Jordan Bird*

*December 13, 2018*

### Setup Libraries

```
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

```
library("stringr")
library("reshape2")
```

```
## Warning: package 'reshape2' was built under R version 3.4.3
```

```
library("plyr")
```

```
## Warning: package 'plyr' was built under R version 3.4.3
```

```
library("gplots")
```

```
## Warning: package 'gplots' was built under R version 3.4.1
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library("openxlsx")
```

```
## Warning: package 'openxlsx' was built under R version 3.4.4
```

### Download BLAST results and metadata

```
REF_HITS <- read.table("../data/WW_REFERENCE_BLAST_OUT.txt", header = F, sep="\t")
REF_META <- read.xlsx(xlsxFile = "../data/WWRefence_Metadata.xlsx", sheet = 1)
UBA_HITS_DATA <- read.xlsx("../data/tyson_genome_list_1201_firms.xlsx")
UBA_HITS <- read.table("../data/B_subtilis_spo_157_gene_in_PARKS_FIRMICUTES.txt",
                       header = F, sep="\t")
```

### Combine Data

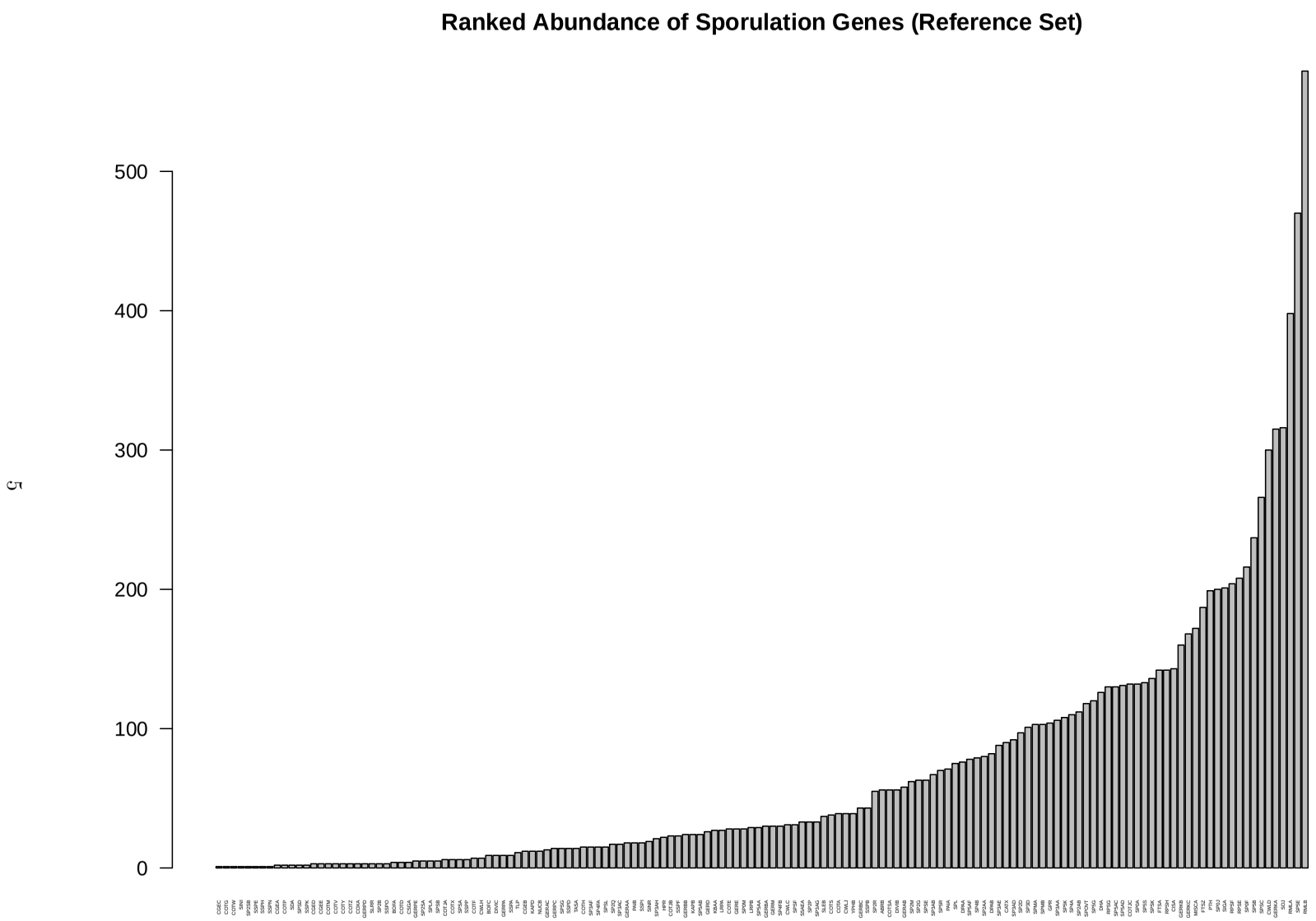
```
colnames(REF_HITS) <- c("qseqid","sseqid","pident","length","qlen","slen","mismatch",
                        "gapopen","qstart","qend","sstart","send","qcov","evalue","bitscore")
REF_HITS <- unique(REF_HITS)
REF_HITS <- REF_HITS[REF_HITS$qcov >= 80,]
REF_HITS <- REF_HITS[REF_HITS$bitscore >= 80,]
REF_HITS <- REF_HITS[order(REF_HITS$bitscore, decreasing = T),]
REF_HITS <- REF_HITS[!duplicated(REF_HITS$sseqid),]
```

```
REF_HITS <- REF_HITS[order(REF_HITS$bitscore, decreasing = T),]
REF_GENOMES <- strsplit(as.vector(unlist(strsplit(as.vector(REF_HITS$sseqid),
                                                 "\\|"))[grep("_", unlist(strsplit(as.vector(REF_HITS$s
                                                                                   "\\|")))])),"_")
Nuccore_ID = ""
for(i in 1:length(REF_GENOMES)){
  if(REF_GENOMES[[i]][1] == "NC" | REF_GENOMES[[i]][1] == "NZ"){
    p <- str_c(REF_GENOMES[[i]][c(1,2)],"", collapse = "_")
  }
  else(
    p <- str_c(REF_GENOMES[[i]][c(1)],"")
  )
  Nuccore_ID[i] <- p
}
REF_HITS$Nuccore_ID <- Nuccore_ID

feature_type = ""
for(i in 1:length(REF_GENOMES)){
  if(REF_GENOMES[[i]][1] == "NC" | REF_GENOMES[[i]][1] == "NZ"){
    p <- REF_GENOMES[[i]][c(3)]
  }
  else(
    p <- REF_GENOMES[[i]][c(2)]
  )
  feature_type[i] <- p
}
REF_HITS$feature_type <- feature_type
RefSeq_ID = ""
for(i in 1:length(REF_GENOMES)){
  if(REF_GENOMES[[i]][1] == "NC" | REF_GENOMES[[i]][1] == "NZ"){
    p <- str_c(REF_GENOMES[[i]][c(4,5)],"", collapse = "_")
  }
  else(
    p <- str_c(REF_GENOMES[[i]][c(3)],"", collapse = "_")
  )
  RefSeq_ID[i] <- p
}
REF_HITS$RefSeq_ID <- RefSeq_ID

gene_names <- unlist(strsplit(as.vector(REF_HITS$qseqid),
"\\|"))[grep("_",
unlist(strsplit(as.vector(REF_HITS$qseqid), "\\|")))]
gene_names_spl <- strsplit(gene_names, "_")
unlist(strsplit(gene_names, "_"))
is.odd <- function(x) x %% 2 != 0
gene_names <- unlist(strsplit(gene_names,
"_"))[seq(from=1,
to=length(unlist(strsplit(gene_names, "_"))))[is.odd(seq(from=1,
to=length(unlist(strsplit(gene_names, "_"))))]]
REF_HITS$Gene_Names <- gene_names
REF_HITS <- merge(REF_HITS, REF_META)
#Check to make sure all the genes are in REF_HITS set after filtering
list_genes<-unique(REF_HITS$Gene_Names)
```

```
setdiff(list_genes, REF_HITS$Gene_Names)
```

**plot abundances of genes in reference dataset**

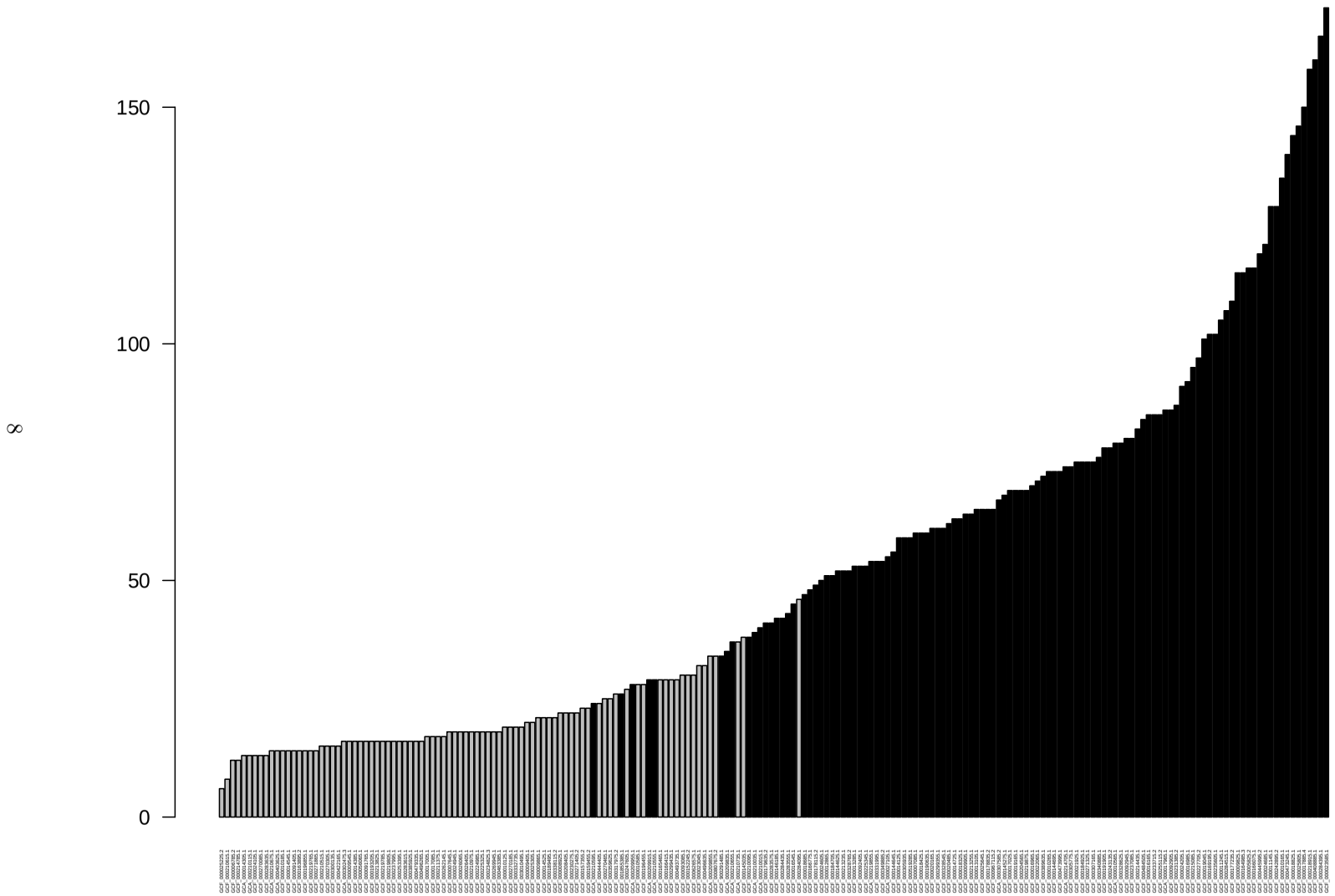
```r
barplot(table(REF_HITS$Gene_Names)[order(table(REF_HITS$Gene_Names))], cex.names = 0.25,
        las= 2, main = "Ranked Abundance of Sporulation Genes (Reference Set)")
```


Ranked Abundance of Sporulation Genes (Reference Set)
500
400
300
200
100
0
g

**Label spore formers and non spore formers in reference set**

```
spore_cols <- REF_HITS[,c(23,27)]
genome_gene_count <- as.data.frame(table(spore_cols$Assembly))
colnames(genome_gene_count) <- c("Assembly", "Freq")
spore_cols <- unique(spore_cols)
spore_cols<- merge(spore_cols, genome_gene_count)
spore_cols <- spore_cols[order(spore_cols$Freq),]
spore_cols$cols <- spore_cols$`Spore.Forming.(Weller.and.Wu)`
spore_cols[spore_cols$cols == "Y",]$cols <- "black"
spore_cols[spore_cols$cols == "N",]$cols <- "grey"
```

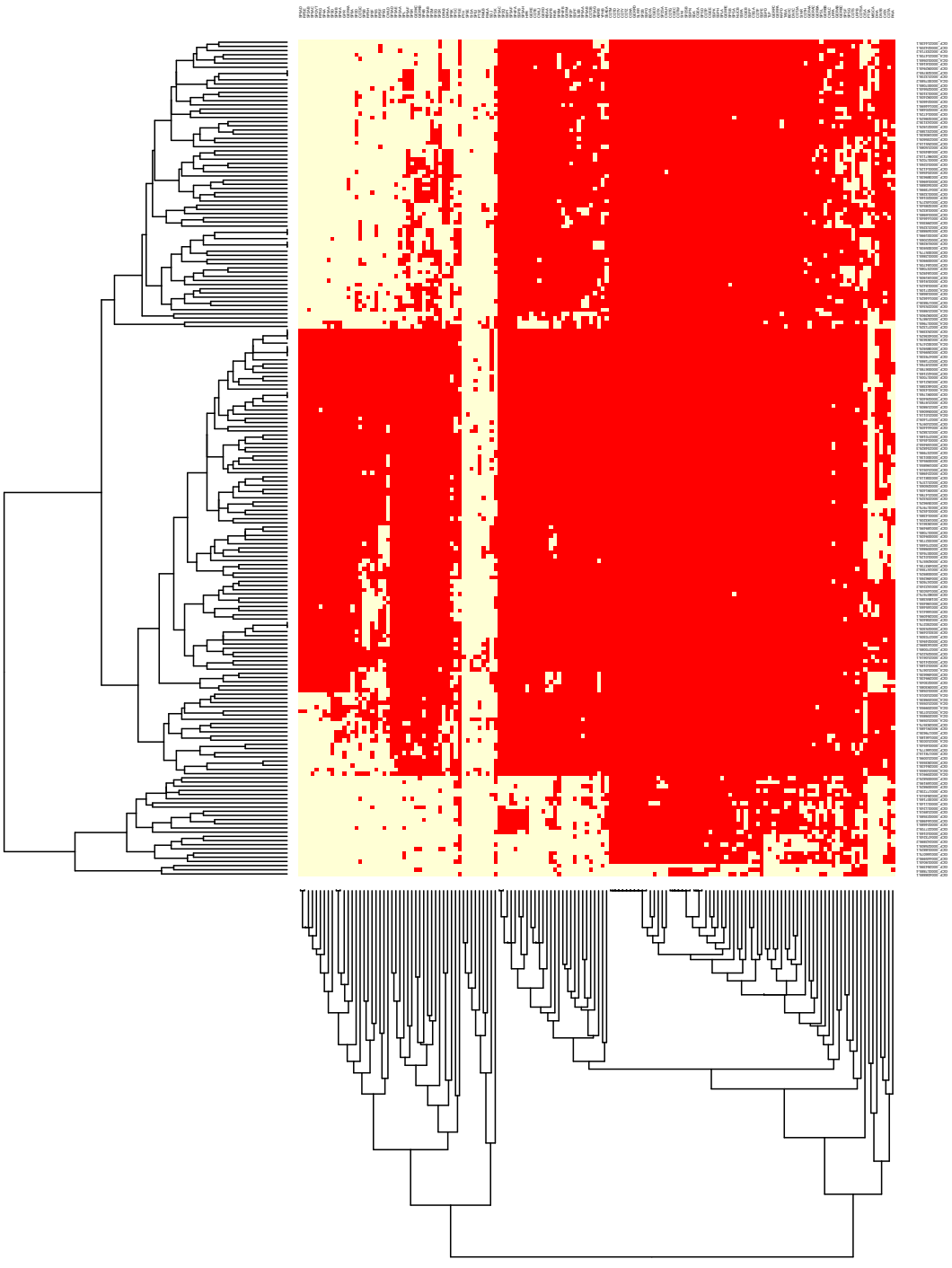
```r
barplot(spore_cols$Freq, las= 2, cex.axis = 1, cex.names = 0.25,
        names.arg = spore_cols$Assembly, col=spore_cols$cols,
        main="Ranked Abundance of Sporulation Genes in Genomes (Reference Set)")
```


Ranked Abundance of Sporulation Genes in Genomes (Reference Set)
150
100
50
0
∞

**Convert abundance to presence/abscene**

```
gene_by_genome <- table(REF_HITS$Gene_Names, REF_HITS$Assembly)
gene_by_genome[gene_by_genome > 0] <- 1
gene_by_genome_df <- as.data.frame(gene_by_genome)
colnames(gene_by_genome_df) <- c("Gene_Names", "Assembly", "Presence")
gene_by_genome_df <- merge(gene_by_genome_df, spore_cols)
gene_by_genome_df[order(gene_by_genome_df$Freq),]
```

```
heatmap.2(gene_by_genome, trace="none", margins = c(10,10), cexRow=0.25,
          cexCol=0.25,
          key = FALSE, sepwidth=c(0.5,0.5), sepcolor="black")
```



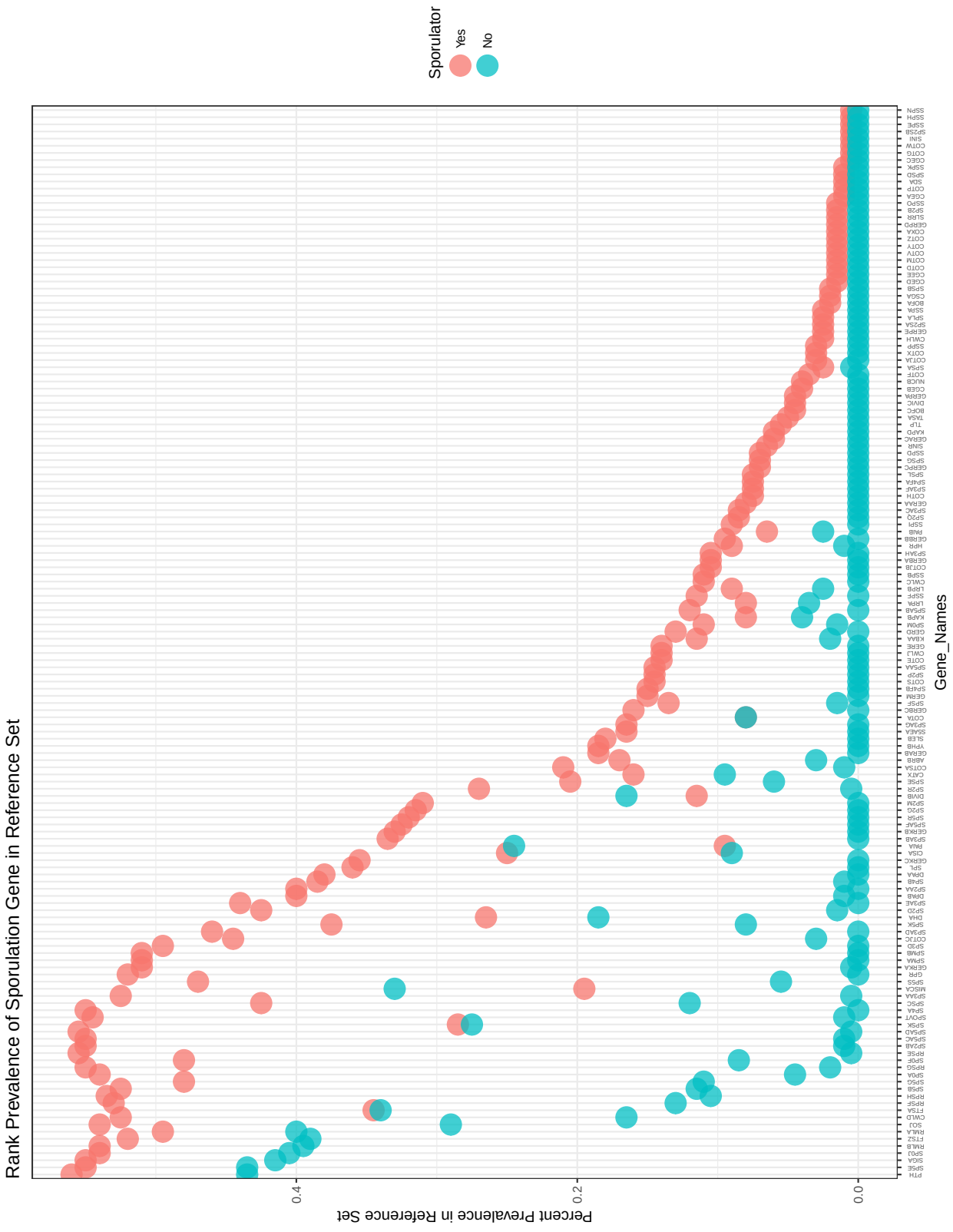
```
gene_presence <- ddply(gene_by_genome_df, .(Gene_Names), summarise, sum_gene_presence = sum(Presence))
gene_by_genome_df <- merge(gene_by_genome_df, gene_presence)
gene_by_genome_df <- gene_by_genome_df[order(gene_by_genome_df$sum_gene_presence),]
```

**Convert presence of Gene to percent of presence in among the 200 Genomes**

```
gene_by_genome_df_hits <- gene_by_genome_df[gene_by_genome_df$Presence > 0,]
sporu_vs_non <- table(gene_by_genome_df_hits$Gene_Names ,gene_by_genome_df_hits$cols)
sporu_vs_non <- melt(sporu_vs_non)

sporu_vs_non$Percent_Core_Set = sporu_vs_non$value/200
colnames(sporu_vs_non) <- c("Gene_Names", "cols", "hits", "Percent_Core_Set")
```

### Plot prevalence of genes in reference dataset in ranked order

```
ggplot(sporu_vs_non,
       aes(y=Percent_Core_Set,
                        x=reorder(Gene_Names,
                                  -rep(sporu_vs_non[sporu_vs_non$cols == "black",]$Percent_Core_Set +
                                                                                    sporu_vs_non[sporu_vs_non$cols == "grey",]$Percent_Core_Set, 2))
           color=cols)) +
  geom_point(stat="identity", size=6, alpha=0.75) +
  ggtitle("Rank Prevalence of Sporulation Gene in Reference Set") + xlab("Gene_Names") +
  ylab("Percent Prevalence in Reference Set") +
  theme_bw(base_size = 10) + scale_color_discrete("Sporulator", labels=c("Yes", "No")) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.4, size = 4))
```


Rank Prevalence of Sporulation Gene in Reference Set
Percent Prevalence in Reference Set
0.0
0.2
0.4
Gene_Names
Sporulator
Yes
No

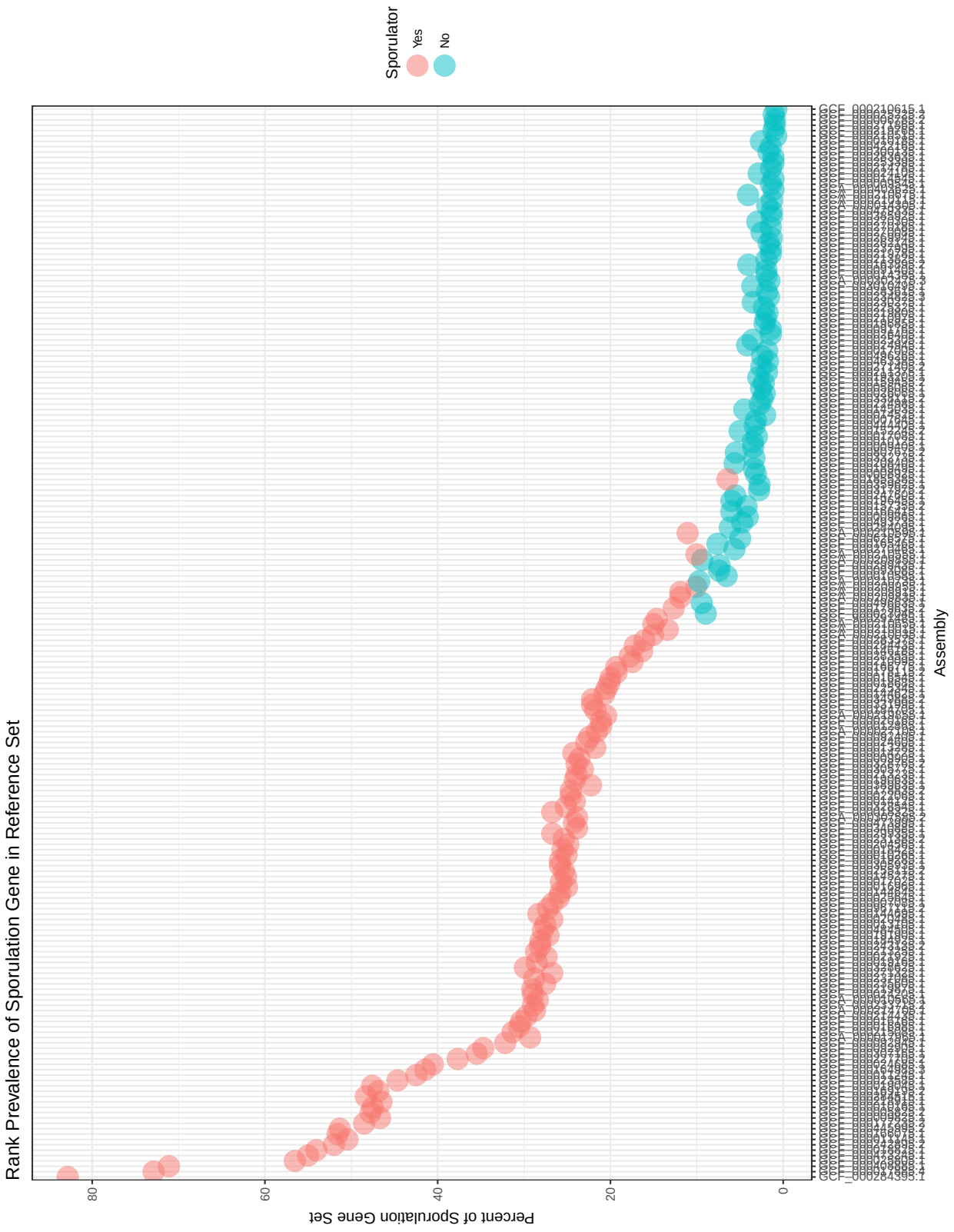
### Weight the prevalence of genes in the reference dataset

The weight is equal to the prevalence of a gene in the reference dataset among sporulators minus the prevalence of a gene in the reference dataset among nonsporulators all divided by the prevalence of a gene in the reference dataset among sporulators. This allows for a gene that is a present in exclusively sporulator to count as one and a gene with some representation among non-sporulators to be discounted by that prevalence. In some cases the genes more prevalent in nonsporulator than sporulators and that negative value was returned to a value of zero.

```
gene_weights <- as.data.frame(cbind(as.vector(sporu_vs_non[sporu_vs_non$cols == "black",]$Gene_Names),
                                    as.vector(as.numeric((sporu_vs_non[sporu_vs_non$cols ==
                                                                       "black",]$Percent_Core_Set-spo
colnames(gene_weights) <- c("Gene_Names", "weight")
gene_weights$weight <- as.numeric(as.character(gene_weights$weight))
gene_weights[gene_weights$weight < 0,]$weight <- 0
gene_by_genome_df_hits <- merge(gene_by_genome_df_hits, gene_weights)

weighted_percents <- ddply(gene_by_genome_df_hits, .(Assembly), summarise, weighted_percent= sum(weight
sporu_vs_non <- table(gene_by_genome_df_hits$Assembly ,gene_by_genome_df_hits$cols)
sporu_vs_non <- melt(sporu_vs_non)
sporu_vs_non <- sporu_vs_non[sporu_vs_non$value != 0, ]
sporu_vs_non$Percent_Core_Set = sporu_vs_non$value/150
colnames(sporu_vs_non) <- c("Assembly", "cols", "hits", "Percent_Core_Set")

sporu_vs_non <- merge(sporu_vs_non, weighted_percents)
```

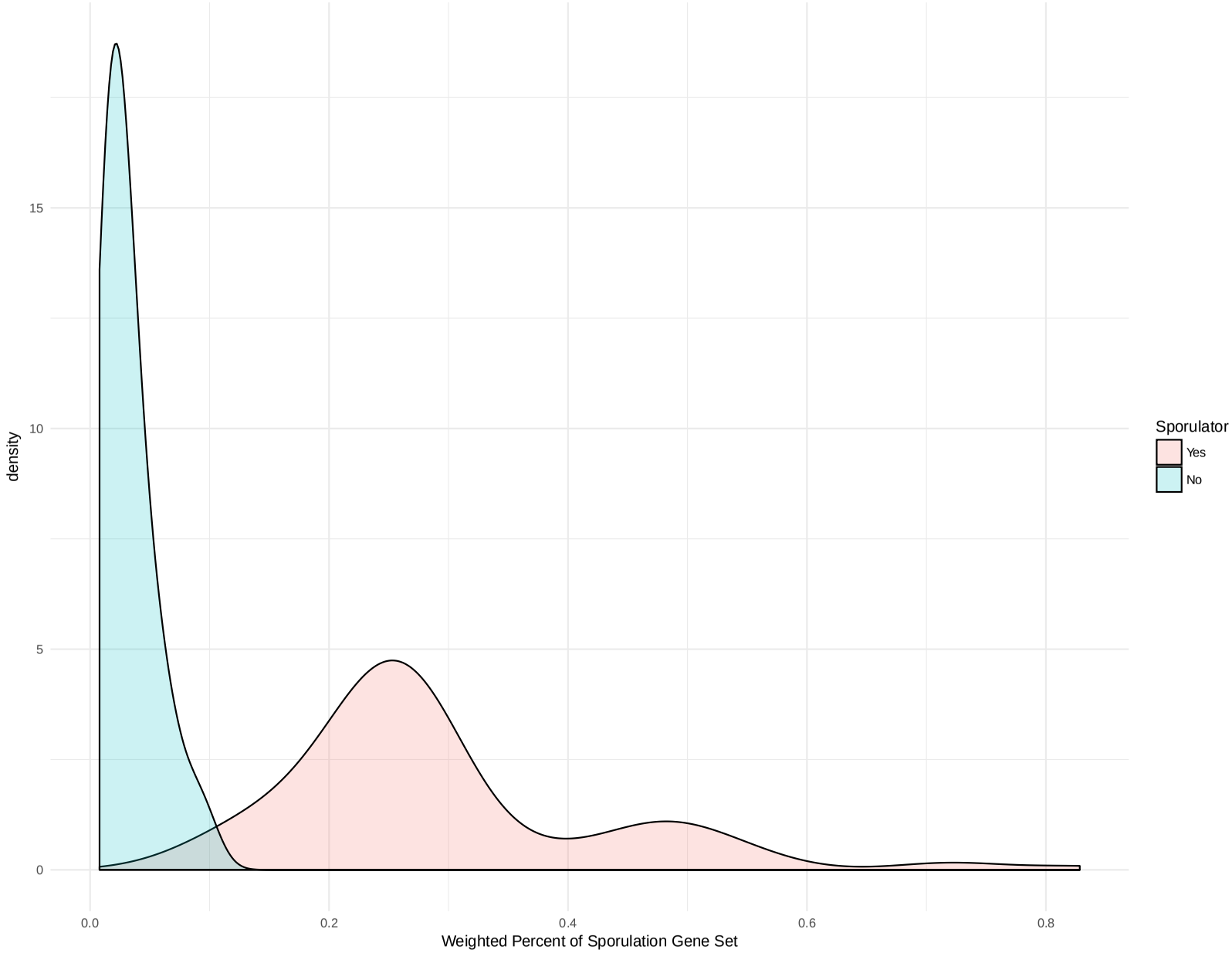
### Plot

```
ggplot(sporu_vs_non, aes(y=weighted_percent*100, x=reorder(Assembly, -(hits)), color=cols)) +
  geom_point(stat="identity", size=6, alpha = 0.5) +
  ggtitle("Rank Prevalence of Sporulation Gene in Reference Set") + xlab("Assembly") +
  ylab("Percent of Sporulation Gene Set") +
  theme_bw(base_size = 10) + scale_color_discrete("Sporulator", labels=c("Yes", "No")) +
  theme(axis.text.x = element_text(angle = -90, vjust = 0.4, hjust = 0, size = 8))
```


Rank Prevalence of Sporulation Gene in Reference Set
Percent of Sporulation Gene Set
0
20
40
60
80
Assembly
Sporulator
Yes
No

**Plot**

```
p <- ggplot(sporu_vs_non) +
  geom_density(aes(x=weighted_percent, fill=cols), alpha=0.2, adjust=2) +
  xlab("Weighted Percent of Sporulation Gene Set") +
  scale_fill_discrete("Sporulator", labels=c("Yes", "No", "?")) +
  theme_minimal(base_size = 10)
p
```


density
15
10
5
0
0.0
0.2
0.4
0.6
0.8
Weighted Percent of Sporulation Gene Set
Sporulator
Yes
No

**Bring in Tyson dataset**

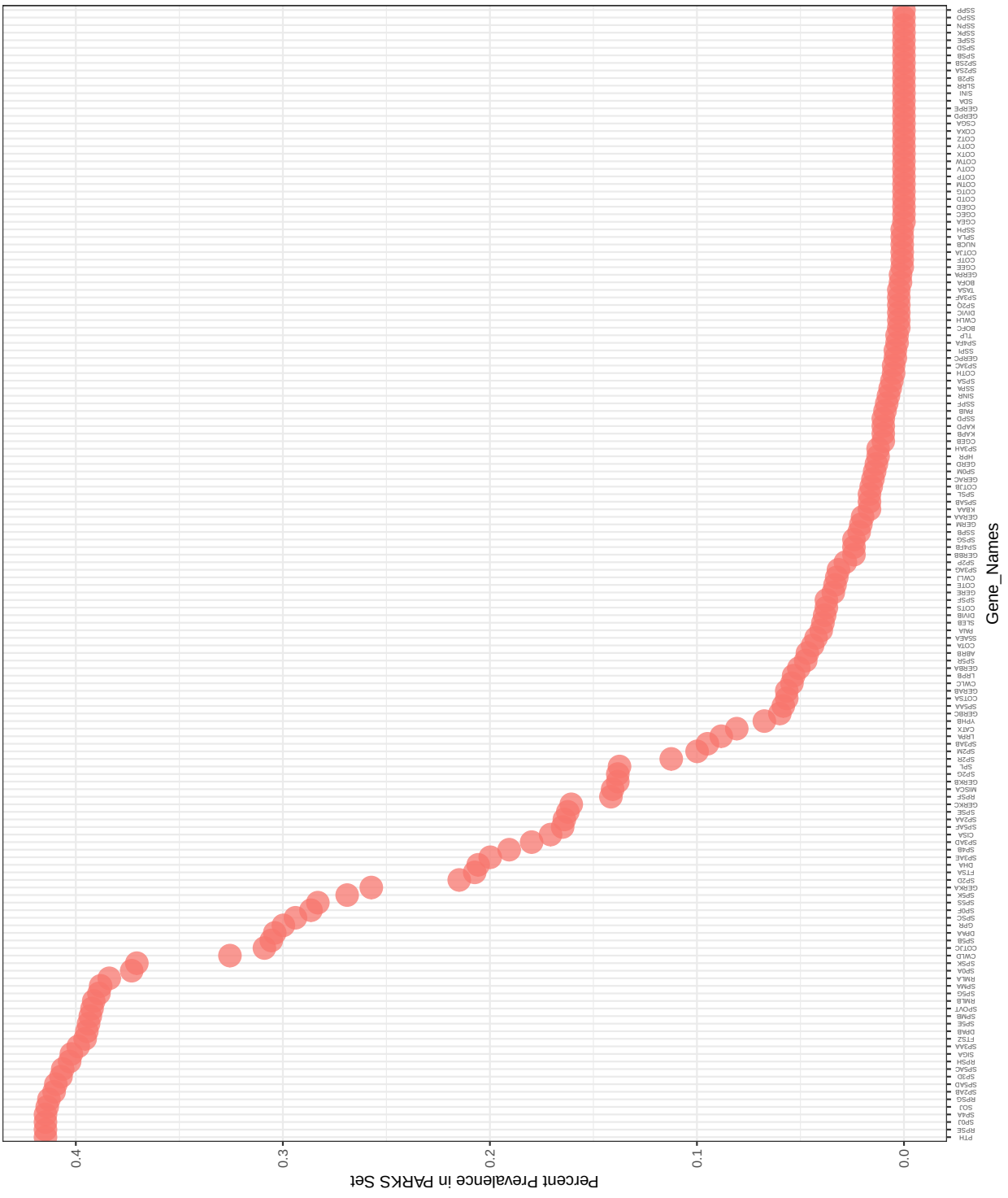
```
colnames(UBA_HITS) <- c("qseqid","sseqid","pident","length","qlen","slen","mismatch",
                        "gapopen","qstart","qend","sstart","send","qcov","evalue","bitscore")
length(unique(UBA_HITS$Assembly))
UBA_HITS$Genome = str_sub(UBA_HITS$sseqid,1,4)
list_genomes<-unique(UBA_HITS$Genome)


gene_names <- unlist(strsplit(as.vector(UBA_HITS$qseqid),
                              "\\|"))[grep("_", unlist(strsplit(as.vector(UBA_HITS$qseqid),
                                                                "\\|")))]
gene_names_spl <- strsplit(gene_names, "_")

is.odd <- function(x) x %% 2 != 0
gene_names <- unlist(strsplit(gene_names, "_"))[seq(from=1,
to=length(unlist(strsplit(gene_names, "_"))))[is.odd(seq(from=1,
to=length(unlist(strsplit(gene_names, "_"))))]]

UBA_HITS$Gene_Names <- gene_names
list_genes<-unique(UBA_HITS$Gene_Names)

UBA_HITS <- unique(UBA_HITS)
UBA_HITS <- UBA_HITS[UBA_HITS$qcov >= 80,]
UBA_HITS <- UBA_HITS[UBA_HITS$bitscore >= 80,]

UBA_HITS <- UBA_HITS[order(UBA_HITS$bitscore, decreasing = T),]

UBA_HITS <- UBA_HITS[!duplicated(UBA_HITS$sseqid),]

genomes_not_found <- setdiff(list_genomes, UBA_HITS$Genome)
genes_not_found <- setdiff(list_genes, UBA_HITS$Gene_Names)


UBA_HITS$Genome = str_sub(UBA_HITS$sseqid,1,4)

UBA_HITS_DATA$Genome = str_sub(UBA_HITS_DATA$`DDBJ/ENA/GenBank.Accession`,1,4)
UBA_HITS <- merge(UBA_HITS, UBA_HITS_DATA)
UBA_GENES_GENOMES <-UBA_HITS[,c(17,19)]



setarr <- setdiff(UBA_GENES_GENOMES$Gene_Names, REF_HITS$Gene_Names)

UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[1],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[2],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[3],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[4],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[5],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[6],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[7],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[8],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[9],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[10],]
```

```
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[11],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[12],]
UBA_GENES_GENOMES <- UBA_GENES_GENOMES[UBA_GENES_GENOMES$Gene_Names != setarr[13],]

genes_not_found <- setdiff(REF_HITS$Gene_Names, UBA_GENES_GENOMES$Gene_Names)

count_by_gene <- as.data.frame(table(UBA_GENES_GENOMES$UBA.Genome.ID,UBA_GENES_GENOMES$Gene_Names))

count_by_gene[count_by_gene$Freq > 0,]$Freq <- 1
colnames(count_by_gene) <- c("Assembly", "Gene_Names", "Presence")

genes_not_found <- as.data.frame(cbind(genes_not_found, rep(0, length(genes_not_found))))
colnames(genes_not_found) <- c("Gene_Names","Presence")


genomes_not_found <-setdiff(UBA_HITS_DATA$UBA.Genome.ID, count_by_gene$Assembly)
genomes_not_found <- as.data.frame(cbind(genomes_not_found,
                                         rep(0, length(genomes_not_found))))
colnames(genomes_not_found) <- c("Assembly","Presence")


not_founds <- merge(c(as.vector(unique(genes_not_found$Gene_Names)),
                      as.vector(unique(count_by_gene$Gene_Names))),
                    c(as.vector(unique(count_by_gene$Assembly)),
                      as.vector(unique(genomes_not_found$Assembly))))



not_founds$Presence <- 0
not_founds <- not_founds[c(2,1,3)]
colnames(not_founds) <- c("Assembly","Gene_Names","Presence")
count_by_gene <- rbind(count_by_gene, not_founds)
```

```
freq_Parks<- ddply(count_by_gene, .(Gene_Names), summarise, Freq=sum(as.numeric(Presence)))
count_by_gene <- merge(count_by_gene, freq_Parks)
count_by_gene$`Spore.Forming.(Weller.and.Wu)` <- "?"
count_by_gene$cols <- "blue"
gene_presence_Parks<- ddply(count_by_gene, .(Assembly),
                            summarise, sum_gene_presence=sum(as.numeric(Presence)))
count_by_gene <- merge(count_by_gene, gene_presence_Parks)

count_by_gene <- count_by_gene[c(2,1,3,5,4,6,7)]

count_by_gene_hits <- count_by_gene[count_by_gene$Presence > 0,]
count_by_gene_hits <- merge(count_by_gene_hits, gene_weights)


weighted_percents <- ddply(count_by_gene_hits, .(Assembly),
                           summarise, weighted_percent= sum(weight)/150)

count_by_gene_hits_count <- table(count_by_gene_hits$Gene_Names ,count_by_gene_hits$cols)
count_by_gene_hits_count <- melt(count_by_gene_hits_count)
count_by_gene_hits_count$Percent_Core_Set = count_by_gene_hits_count$value/1201
colnames(count_by_gene_hits_count) <- c("Gene_Names", "cols", "hits", "Percent_Core_Set")
```

### Plot

```
ggplot(count_by_gene_hits_count,
       aes(y=Percent_Core_Set, x=reorder(Gene_Names, -hits), color=cols)) +
  geom_point(stat="identity", size=6, alpha=0.75) +
  xlab("Gene_Names") +
  ylab("Percent Prevalence in PARKS Set") +
  theme_bw(base_size = 10) +
  scale_color_discrete("Sporulator", labels=c("?")) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.4, size = 4))
```

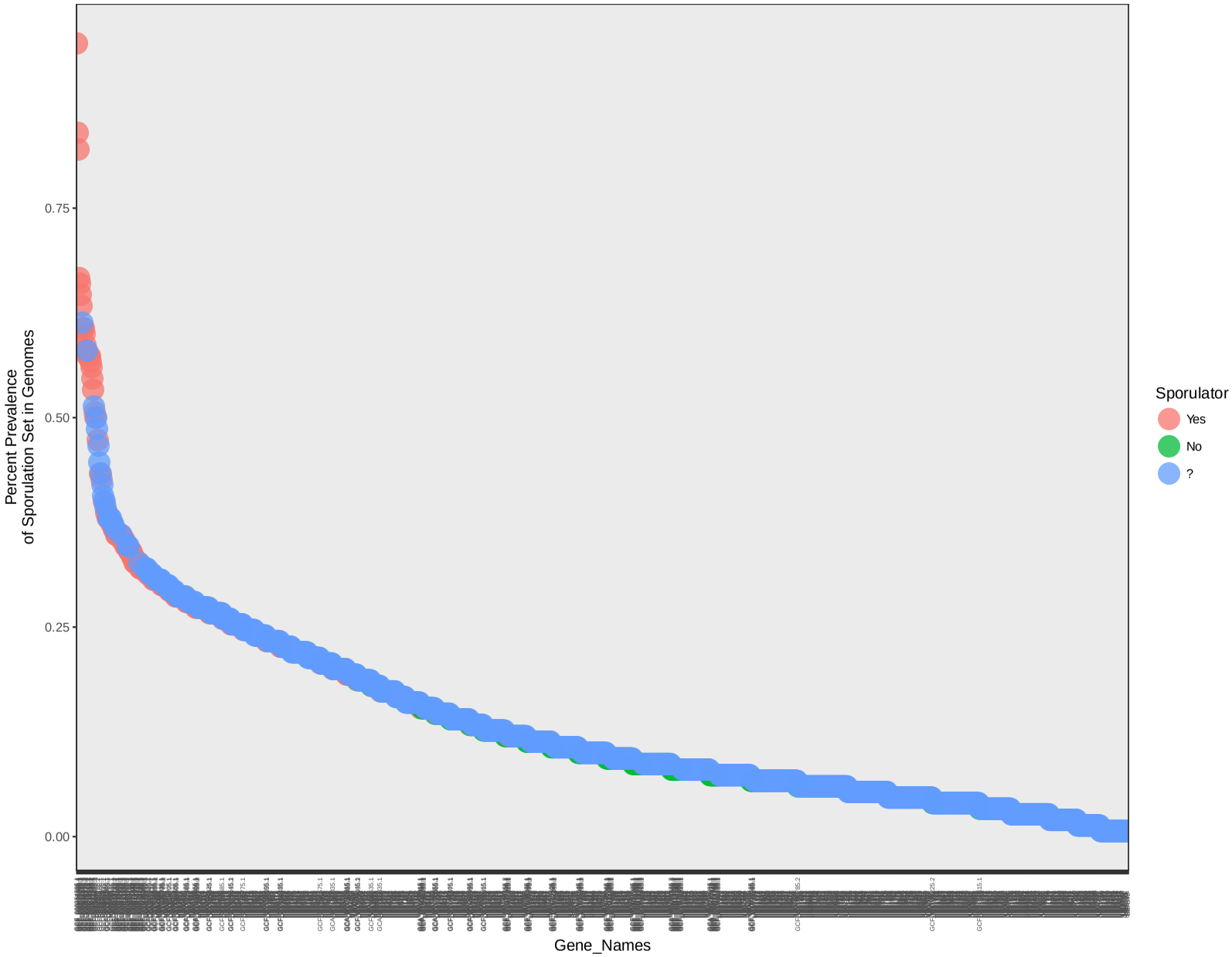

Sporulator
?
Percent Prevalence in PARKS Set
0.0
0.1
0.2
0.3
0.4
Gene_Names

```
count_by_gene_hits_count <- count_by_gene[count_by_gene$Presence > 0,]
count_by_gene_hits_count <- table(count_by_gene_hits_count$Assembly ,count_by_gene_hits_count$cols)
count_by_gene_hits_count <- melt(count_by_gene_hits_count)
colnames(count_by_gene_hits_count) <- c("Assembly", "cols", "hits")
count_by_gene_hits_count$Percent_Core_Set = count_by_gene_hits_count$hits/150
count_by_gene_hits_count <-merge(count_by_gene_hits_count, weighted_percents)
```

```
PARKS_and_REF <- rbind(sporu_vs_non, count_by_gene_hits_count)
```

### Plot

```
ggplot(PARKS_and_REF, aes(y=Percent_Core_Set, x=reorder(Assembly, -hits), color=cols)) +
  geom_point(stat="identity", size=6, alpha=0.75) +
  xlab("Gene_Names") +
  ylab("Percent Prevalence\nof Sporulation Set in Genomes") +
  theme_bw(base_size = 10) +
  scale_color_discrete("Sporulator", labels=c("Yes","No","?")) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.4, size = 4))
```

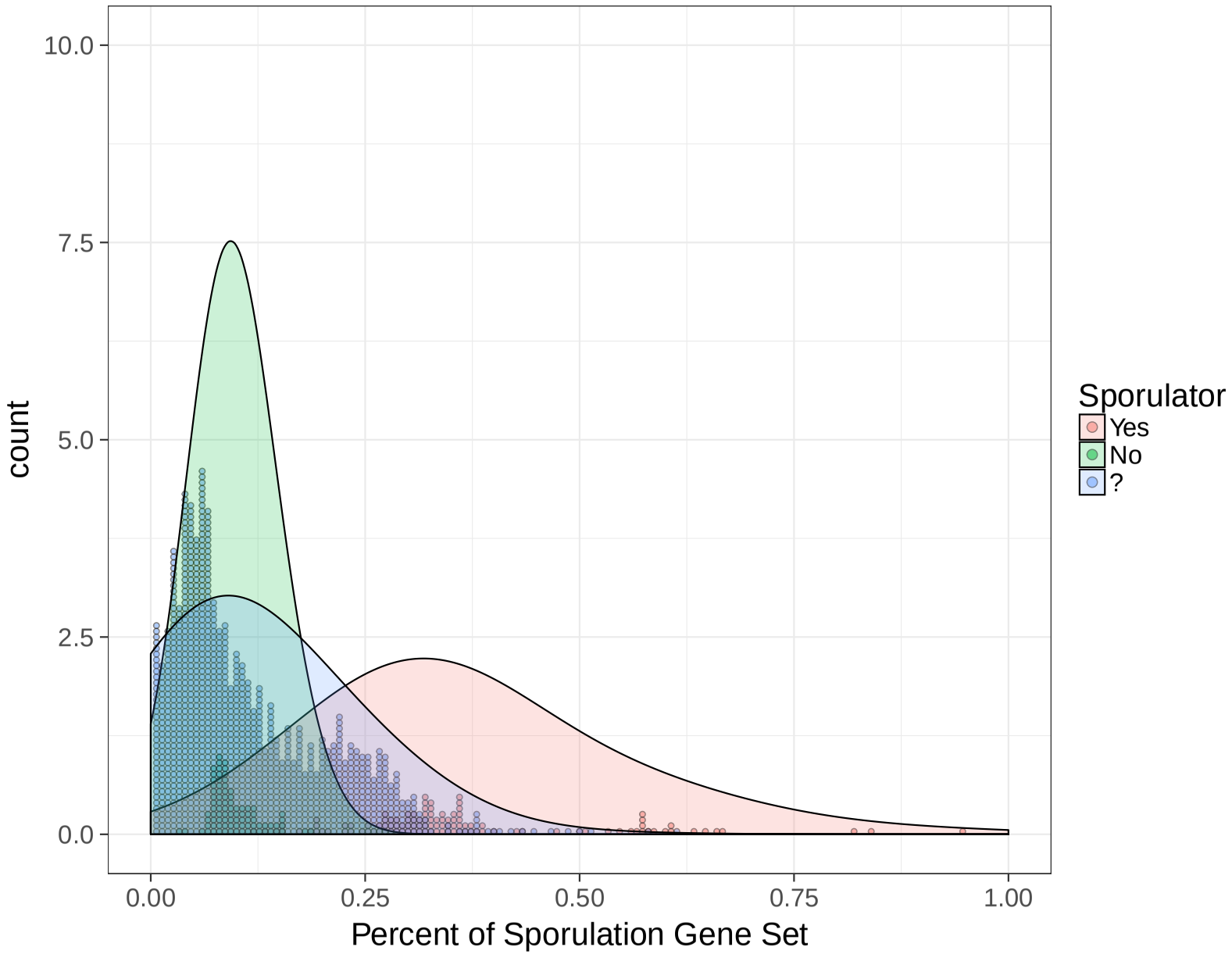

Percent Prevalence
of Sporulation Set in Genomes
0.75
0.50
0.25
0.00
Gene_Names
Sporulator
Yes
No
?

### Plot

```
p <- ggplot(PARKS_and_REF) +
  geom_dotplot(data=PARKS_and_REF, aes(x=Percent_Core_Set, fill=cols),
               alpha=0.5, dotsize = 1.2, binwidth = 1/180) +
  scale_fill_discrete("Sporulator", labels=c("Yes", "No", "?")) +
  theme_bw(base_size = 20)

p <- p + geom_density(aes(x=Percent_Core_Set, fill=cols), alpha=0.2, adjust=5) +
  xlab("Percent of Sporulation Gene Set") +
  scale_fill_discrete("Sporulator", labels=c("Yes", "No", "?")) +
  theme_bw(base_size = 20) + scale_x_continuous(limits=c(0,1)) +
  scale_y_continuous(limits=c(0,10))
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

```
p
```

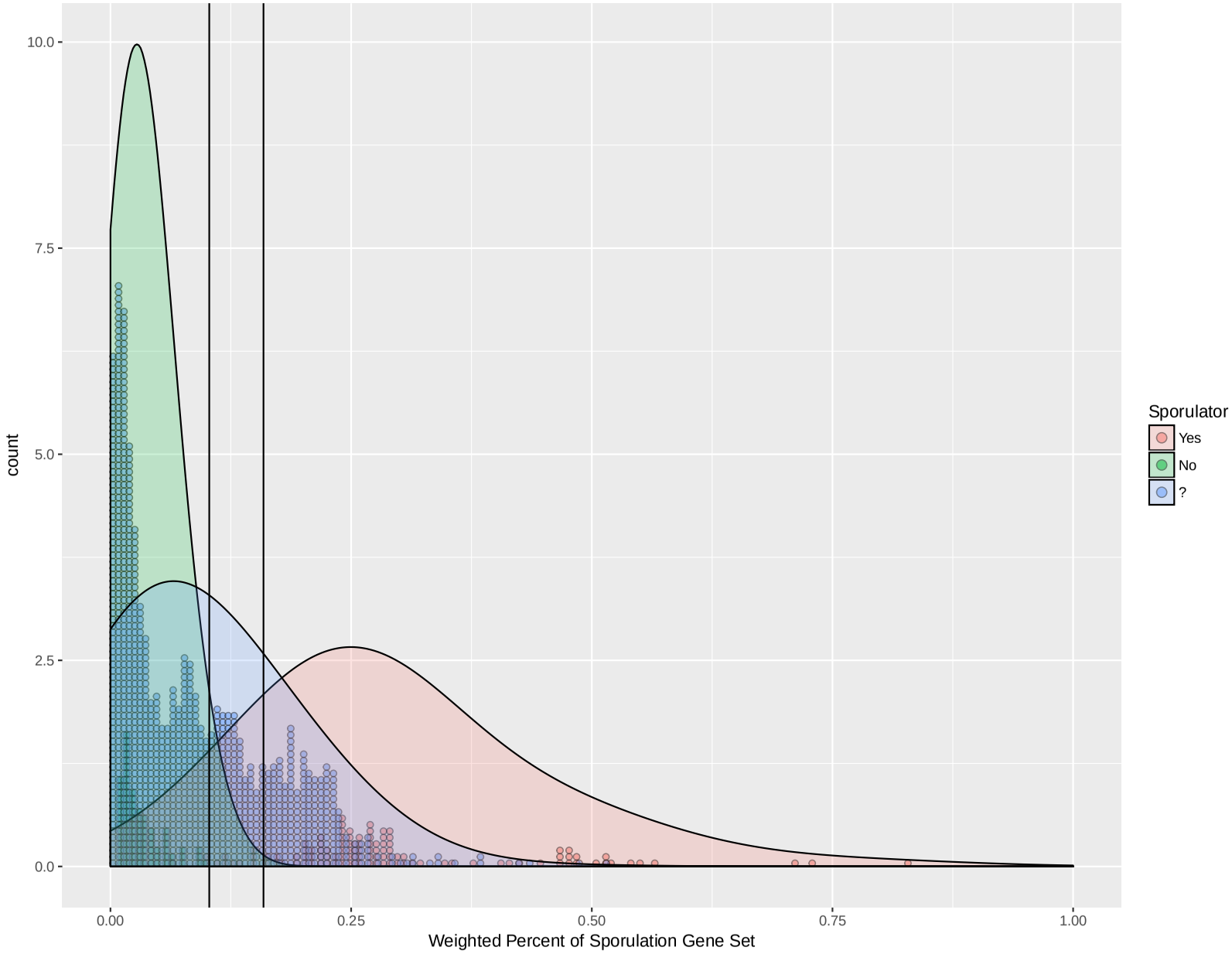

10.0
7.5
5.0
2.5
0.0
count
0.00
0.25
0.50
0.75
1.00
Percent of Sporulation Gene Set
Sporulator
Yes
No
?

**Plot**

```
p <- ggplot(PARKS_and_REF) +
  geom_dotplot(data=PARKS_and_REF, aes(x=weighted_percent, fill=cols),
               alpha=0.5,dotsize = 1.2, binwidth = 1/180) +
  scale_fill_discrete("Sporulator", labels=c("Yes", "No"))

p <- p +
  geom_density(aes(x=as.numeric(weighted_percent), fill=cols), alpha=0.2, adjust=5) +
  xlab("Weighted Percent of Sporulation Gene Set") +
  scale_fill_discrete("Sporulator", labels=c("Yes", "No", "?")) +
  scale_x_continuous(limits=c(0,1))
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

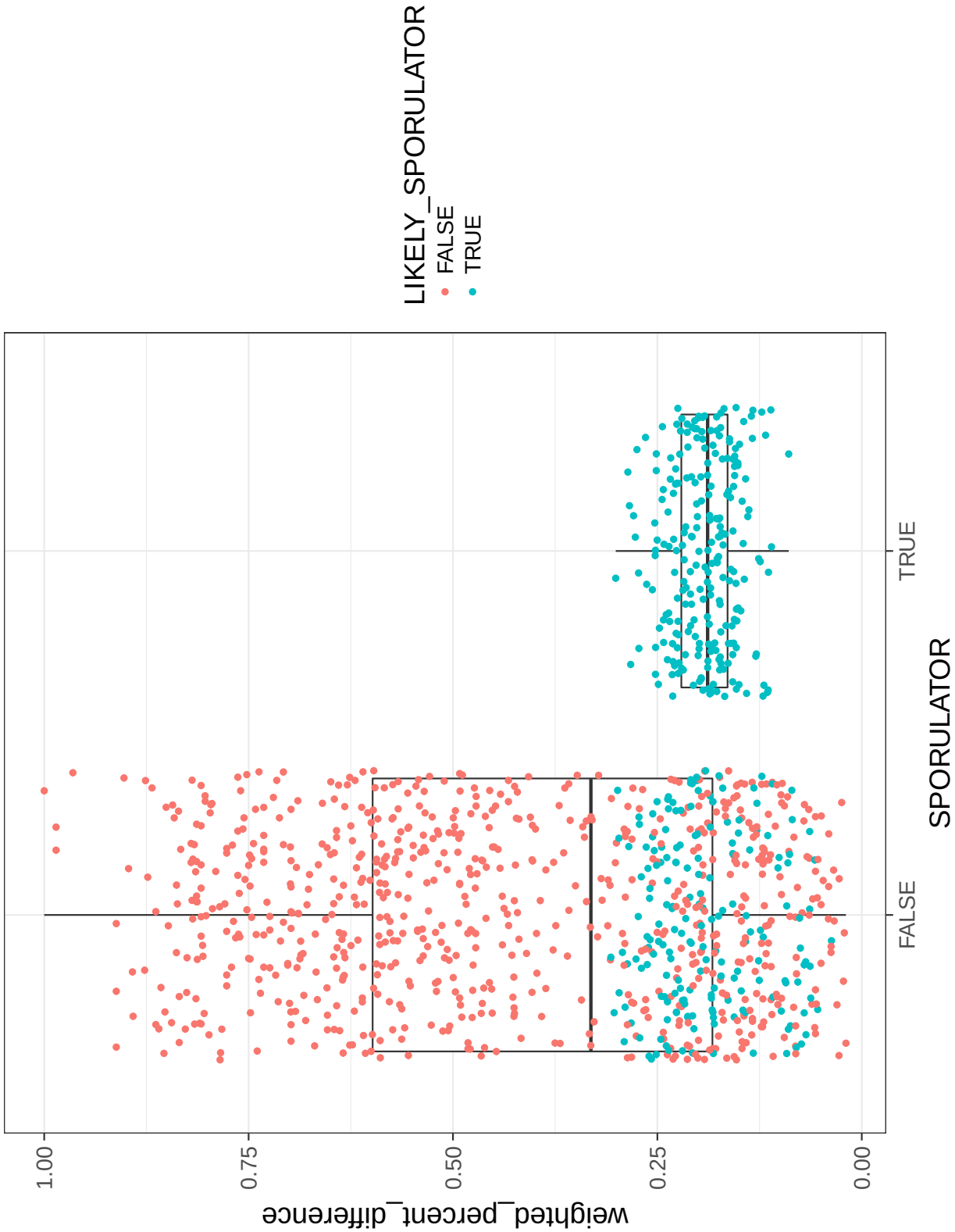
```
mu <- ddply(PARKS_and_REF, "cols", summarise,
            grp.mean=mean(Percent_Core_Set), grp.sd= sd(Percent_Core_Set))

weighted_mu <- ddply(PARKS_and_REF, "cols", summarise,
                     grp.mean=mean(weighted_percent), grp.sd= sd(weighted_percent))
p <- p +
  geom_vline(aes(xintercept=weighted_mu[2,2] +
                   3.290*weighted_mu[2,3]))
p <- p +
  geom_vline(aes(xintercept=weighted_mu[1,2] - 1*weighted_mu[1,3]))
p
```


10.0
7.5
5.0
2.5
0.0
count
0.00
0.25
0.50
0.75
1.00
Weighted Percent of Sporulation Gene Set
Sporulator
Yes
No
?

```
PARKS <- PARKS_and_REF[PARKS_and_REF$cols == "blue", ]
PARKS$LIKELY_SPORULATOR <- PARKS_and_REF[PARKS_and_REF$cols == "blue", ]$weighted_percent > weighted_mu
  3.290*weighted_mu[2,3]
PARKS$SPORULATOR <- PARKS_and_REF[PARKS_and_REF$cols == "blue", ]$weighted_percent > weighted_mu[1,2] -

PARKS <- ddply(PARKS, .(Assembly), mutate,
               weighted_percent_difference = (Percent_Core_Set - weighted_percent)/Percent_Core_Set,
               weighted_difference = (Percent_Core_Set - weighted_percent))
```

### Plot

```
ggplot(PARKS) +
  geom_boxplot(aes(x=SPORULATOR, y=weighted_percent_difference), position = "dodge") +
  geom_jitter(aes(x=SPORULATOR, y=weighted_percent_difference, color=LIKELY_SPORULATOR)) +
  theme_bw(base_size = 18)
```


LIKELY_SPORULATOR
FALSE
TRUE
weighted_percent_difference
1.00
0.75
0.50
0.25
0.00
FALSE
TRUE
SPORULATOR

```
REFS <- PARKS_and_REF[PARKS_and_REF$cols == "black" | PARKS_and_REF$cols == "grey", ]

REFS <- ddply(REFS, .(Assembly, cols), mutate,
              weighted_percent_difference = (Percent_Core_Set - weighted_percent)/Percent_Core_Set,
              weighted_difference = (Percent_Core_Set - weighted_percent))
```

### Plot

```
ggplot(REFS) +
  geom_boxplot(aes(x=cols, y=weighted_percent_difference), position = "dodge") +
  geom_jitter(aes(x=cols, y=weighted_percent_difference)) + theme_bw(base_size = 30) +
  scale_x_discrete(labels=(c("Sporulators","Non Sporulators"))) + xlab("")
```

weighted_percent_difference
0.75
0.50
0.25
Sporulators
Non Sporulators