Bayesian Analysis of Beta-RD Statistic

Rauf Salamzade

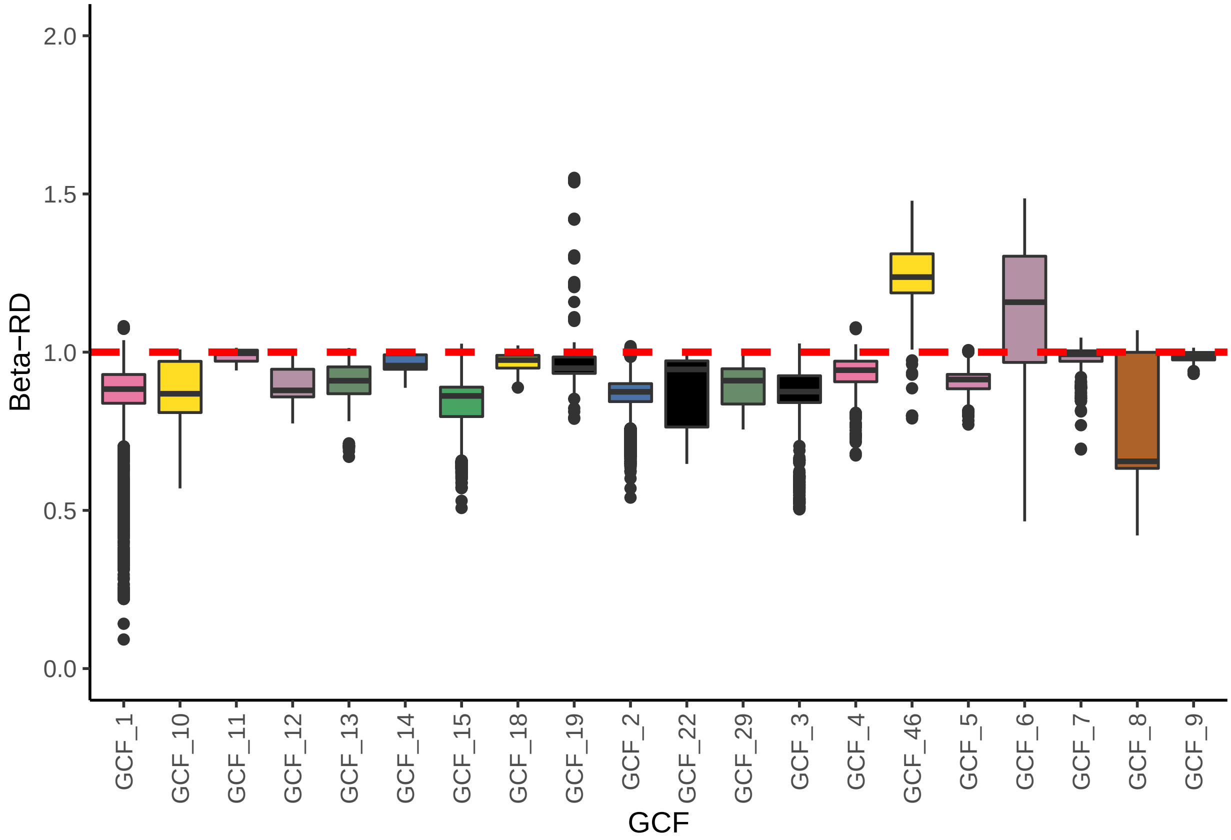
April 01, 2022

Setup of R libraries

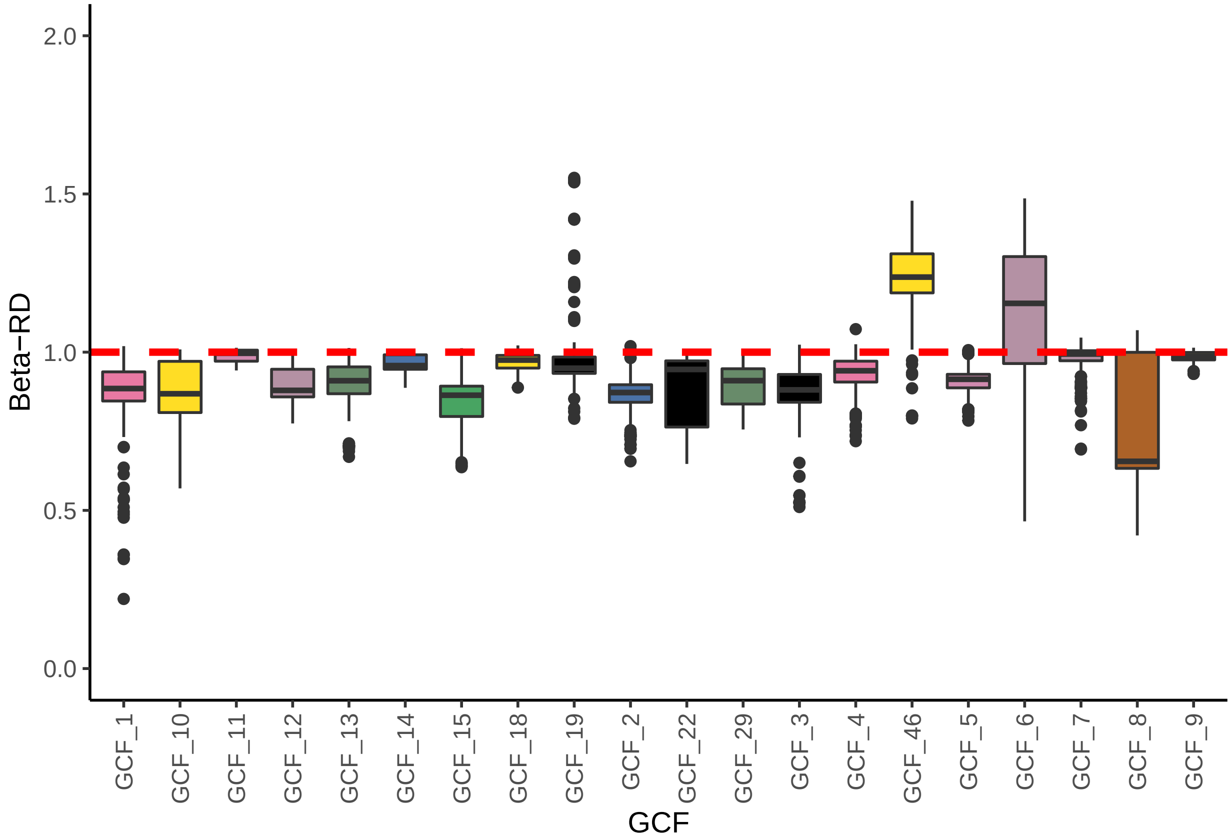
```r
library(knitr)
library(rstan)
library(ggplot2)
library(corrplot)
library(RColorBrewer)
library(dplyr)
set.seed(12345)
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

Quick and Simple Preliminary Exploration of the Dataset

```r
data <- read.table('Staphylococcus_BetaRD.txt', header=T, sep='\t')

nb.cols <- 28
gcf.cols <- c(colorRampPalette(brewer.pal(9, "Set1"))(nb.cols), c("#000000"))
names(gcf.cols) <- c("betalactone", "butyrolactone", "CDPS",
                     "cyclic-lactone-autoinducer", "ectoine","epipeptide",
                     "hserlactone", "ladderane", "lanthipeptide-class-i",
                     "lanthipeptide-class-ii", "lanthipeptide-class-iii",
                     "lanthipeptide-class-iv", "lanthipeptide-class-v", "LAP",
                     "linaridin", "NAPAA", "NRPS", "NRPS-like", "nucleoside",
                     "phenazine", "RiPP-like", "RRE-containing", "sactipeptide",
                     "siderophore", "T1PKS", "T3PKS", "terpene", "thiopeptide",
                     "hybrid")

ggplot(data, aes(x=gcf_name, y=beta_rd)) + theme_classic() +
  geom_boxplot(aes(fill=gcf_class),show.legend=F) +
  scale_fill_manual(values=gcf.cols) +
  geom_hline(yintercept=1.0, color='red', linetype='dashed', size=1.3) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("GCF") + ylab("Beta-RD") + ylim(0,2)
```



```
# downsample data using dplyr and select 1000 beta-rd from each gcf
downsampled_data <- data %>% group_by(gcf_id) %>% slice_sample(n=500)

ggplot(downsampled_data, aes(x=gcf_name, y=beta_rd)) + theme_classic() +
  geom_boxplot(aes(fill=gcf_class),show.legend=F) +
  scale_fill_manual(values=gcf.cols) +
  geom_hline(yintercept=1.0, color='red', linetype='dashed', size=1.3) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("GCF") + ylab("Beta-RD") + ylim(0,2)
```



# Specificying the Model and Hierarchical Structuring

$$
\begin{aligned}
&\overline{\mu} \sim Norm(\mu_0, \sigma_0^2)\\
&\tau \sim half - t_7(1)\\
&\mu_1 ... \mu_J \sim Norm(\overline{\mu}, \tau)\\
&\sigma \sim half - t_7(1)\\
&y_j \sim Norm(\mu_j, \sigma^2)
\end{aligned}
$$

Setting Priors and Getting Input Ready for MCMC Analysis

```
y <- downsampled_data$beta_rd
gcf_id <- downsampled_data$gcf_id

y_mean <- mean(y)
y_sd <- sd(y)
n <- length(y)
J <- length(unique(gcf_id))

# set parameters
mu0 <- 1.0
sigma0 <- y_sd
```

```
R1 <- 1
R2 <- 1
```

Run MCMC Analysis

```
#hier_model <- stan_model(file = "Hierarchical_MCMC.stan")
#save(hier_model, file=’hier_model.Rdata’)
load("hier_model.Rdata")
```

And here is the code to use the STAN model above:

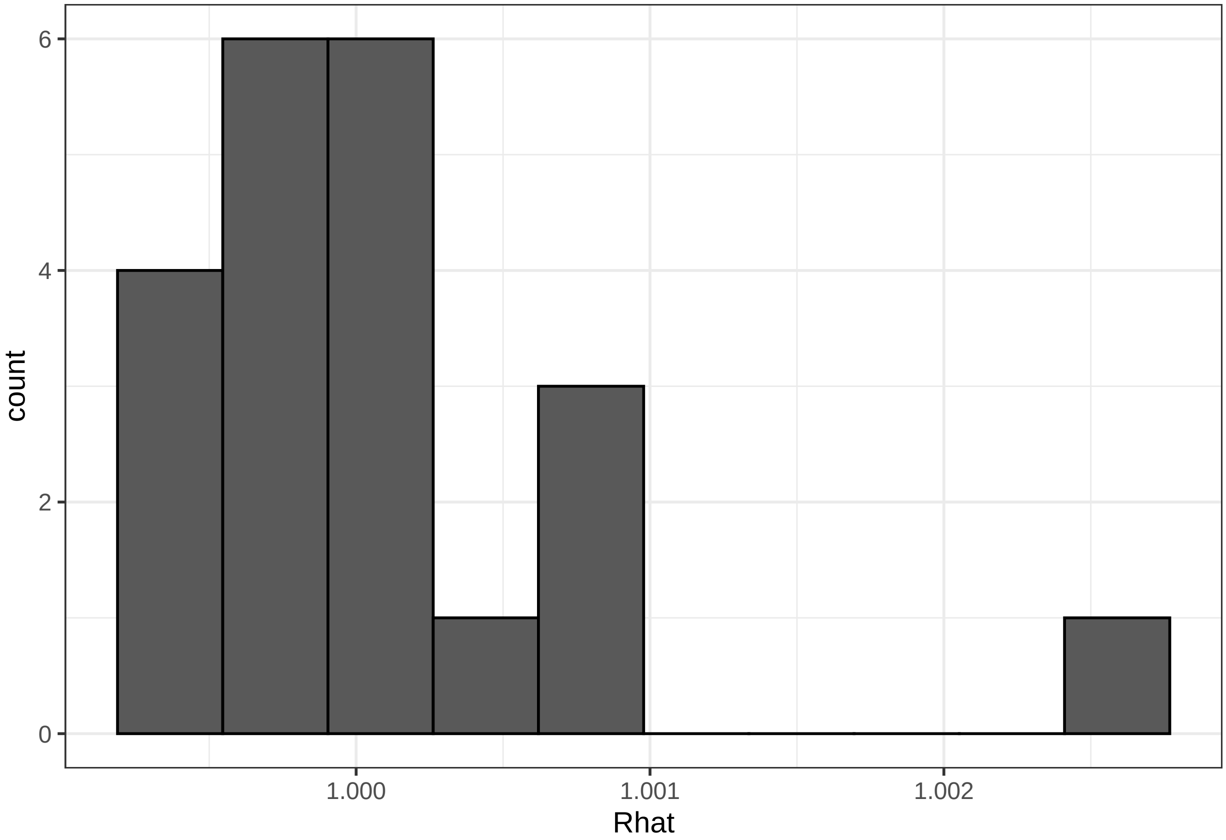
```
stan_input_data <- list(n = n, J = J, y = y, gcf_id = gcf_id, mu0 = mu0,
                        sigma0 = sigma0, R1 = R1, R2 = R2)

#fit <- sampling(object = hier_model, data = stan_input_data, cores=4, iter=2000,
#                pars = c("post_y"))
#save(fit, file=’staphylococcus_analysis_fit.RData’)
load("staphylococcus_analysis_fit.RData")

# Let’s examine the Rhat values first (to see if the chains converged enough)
fit_summary <- summary(fit)
fit_summary_df <- as.data.frame(fit_summary$summary)

# print range of Rhats
print(’Range of Rhats:’)
```

```
## [1] "Range of Rhats:"
```

```
print(range(fit_summary_df$Rhat))
```

```
## [1] 0.9992798 1.0025024
```

```
# Uneccessary code to visualize histogram of Rhat
ggplot(fit_summary_df, aes(x=Rhat)) + geom_histogram(color=’black’, bins=10) + theme_bw()
```

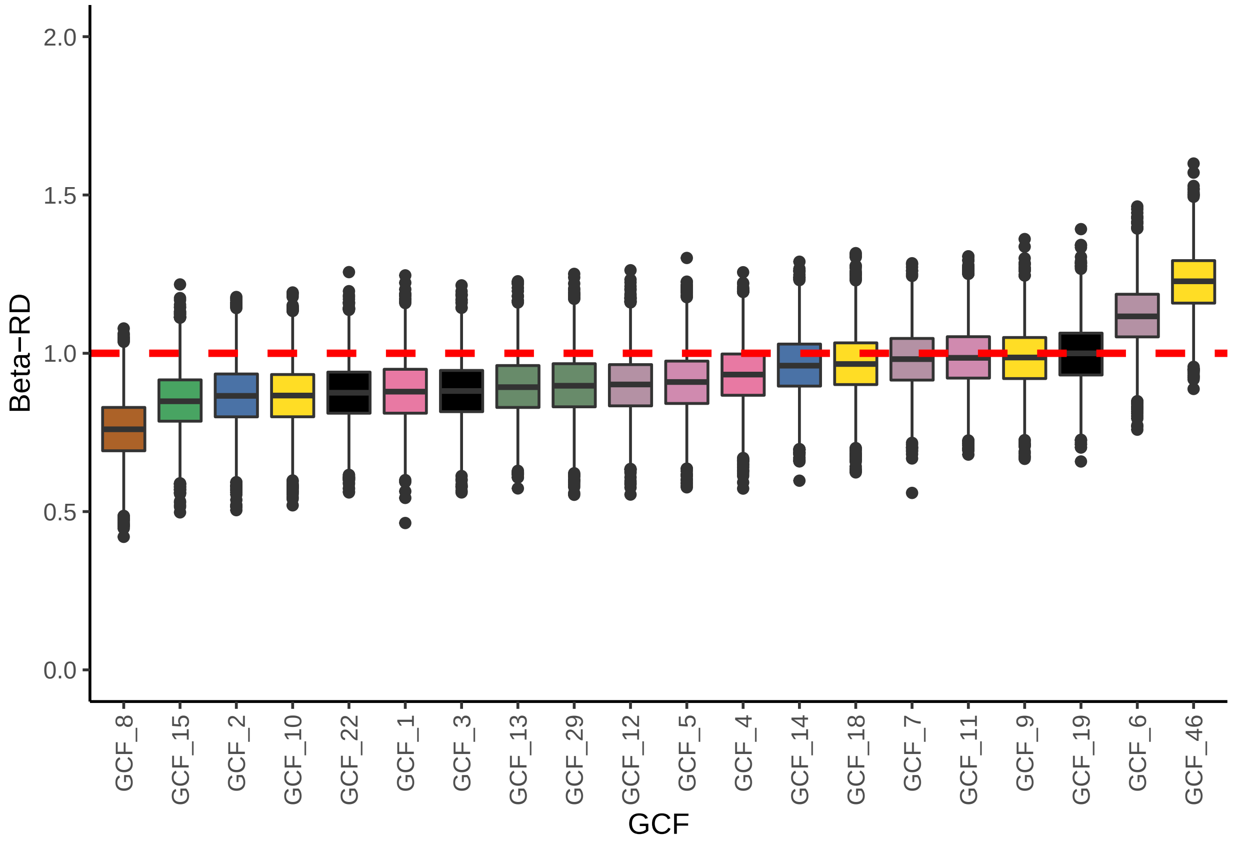


We see Rhat values are well below 1.1, so we believe that the chains have converged!

Visualizing the Posterior Distributions

```r
post_y <- as.data.frame(extract(object = fit, pars = "post_y")[["post_y"]])
gcf_info_df <- distinct(data.frame(gcf_id = downsampled_data$gcf_id,
                    gcf_name = downsampled_data$gcf_name,
                    gcf_class = downsampled_data$gcf_class))
gcf_info_df <- gcf_info_df[order(gcf_id),]
gcf_names_ordered <- gcf_info_df$gcf_name
gcf_class_ordered <- gcf_info_df$gcf_class

posterior.data <- data.frame()
for (j in 1:J) {
  beta_rd_posterior <- as.vector(post_y[j][,1])
  gcf_name <- rep(gcf_names_ordered[j], length(beta_rd_posterior))
  gcf_class <- rep(gcf_class_ordered[j], length(beta_rd_posterior))
  median_beta_rd <- rep(median(beta_rd_posterior), length(beta_rd_posterior))
  posterior.row <- data.frame(beta_rd_posterior = beta_rd_posterior,
                              gcf_name = gcf_name,
                              gcf_class = gcf_class,
                              median_beta_rd = median_beta_rd)
  posterior.data <- rbind(posterior.data, posterior.row)
}
```

```
ggplot(posterior.data, aes(x=reorder(gcf_name, median_beta_rd),
                           y=beta_rd_posterior, fill=gcf_class)) +
  theme_classic() + geom_boxplot(show.legend=F) +
  scale_fill_manual(values=gcf.cols) +
  geom_hline(yintercept=1.0, color='red', linetype='dashed', size=1.3) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("GCF") + ylab("Beta-RD") + ylim(0,2)
```



```
pdf("Staphylococcus_Posterior_BetaRD.pdf", height=5, width=9.38)
ggplot(posterior.data, aes(x=reorder(gcf_name, median_beta_rd),
                           y=beta_rd_posterior, fill=gcf_class)) +
  theme_classic() + geom_boxplot(show.legend=F) +
  scale_fill_manual(values=gcf.cols) +
  geom_hline(yintercept=1.0, color='red', linetype='dashed', size=1.3) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("GCF") + ylab("Beta-RD") + ylim(0,2) + theme(text = element_text(size=20))
dev.off()
```

```
## pdf
##   2
```