# Tuberculosis_Team_A_Dewei_LIN

Dewei Lin

2023-05-16

```
library(ggplot2)
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(patchwork) # To display 2 charts together
library(hrbrthemes)
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
```

```
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
```

```
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(xtable)
library(poisbinom)
library(readxl)
```

## 1. Import data and create columns we need

No perfect test: Measure of one's performance requires a gold standard while the gold standard doesn't apply in our data set because microscopes and PCRs have low sensitivity. Our solution is to use resoultion method to label each sample in our data.
HIT: whether rat hits
LEVEL: bacteria load level of sample
STATUS: POS/NEG label using resolution method

```
#work with df "rats"
TBdetectionRats_Tanzania <- read_excel("TBdetectionRats_Tanzania.xlsx",
    sheet = "Rat Session")
#View(TBdetectionRats_Tanzania)
rats <- TBdetectionRats_Tanzania
df.sample.results <- rats[,c(14:17)]
#df.sample.results
#code HIT;STATUS_BLINDPOS into binary variables, for future convenience
rats$HIT <- as.numeric(rats$HIT==1)
rats$STATUS_BLINDPOS <- as.numeric(rats$STATUS_BLINDPOS==1)
```

```r
rats$LEVEL <- ifelse(is.na(rats$ID_BL_APOPO)==1,rats$ID_BL_DOTS,rats$ID_BL_APOPO)
rats$STATUS <- as.numeric(rats$LEVEL !=1)
df.display <- head(cbind(df.sample.results,rats$HIT,rats$RAT_NAME))
colnames(df.display) <- c("BL_DOTS","GXP_DOTS","BL_APOPO","GXP_APOPO","HIT","RAT_NAME")
#df.display
head(rats)
```

```
## # A tibble: 6 x 28
##   SESSION_DATE        PROGRAM ID_EVALUATION_SESSION ID_SAMPLE RAT_NAME     RUN
##   <dttm>              <chr>                   <dbl>     <dbl> <chr>        <chr>
## 1 2022-08-01 00:00:00 DAR                     14272    820815 Freddy       F
## 2 2022-08-01 00:00:00 DAR                     14272    820815 Carolina     F
## 3 2022-08-01 00:00:00 DAR                     14272    820815 Serena       F
## 4 2022-08-01 00:00:00 DAR                     14272    820815 Princess Le~ F
## 5 2022-08-01 00:00:00 DAR                     14272    820815 Violet       F
## 6 2022-08-01 00:00:00 DAR                     14272    820816 Freddy       B
## # i 22 more variables: HOLE <dbl>, HIT <dbl>, STATUS_BLINDPOS <dbl>,
## #   ID_PATIENT <dbl>, POT_NUMBER <dbl>, DOTS_NAME <chr>, DATE_INCOMING <dttm>,
## #   ID_BL_DOTS <dbl>, ID_GXP_DOTS <dbl>, ID_BL_APOPO <dbl>, ID_GXP_APOPO <lgl>,
## #   START_TIME <dttm>, END_TIME <dttm>, `DURATION (MINS)` <dttm>,
## #   TRAINER <chr>, DOCUMENTER <chr>, HANDLER <chr>, `TEMPERATURE (*C)` <dbl>,
## #   ID_STORAGE <dbl>, REUSED <dbl>, LEVEL <dbl>, STATUS <dbl>
```

## 2. Define sensitivity and specificity formula

```r
sensitivity <- function(df, test, truth) {
  TP <- sum(as.numeric(df[test] == 1 & df[truth] == 1))
  FP <- sum(as.numeric(df[test] == 1 & df[truth] == 0))
  TN <- sum(as.numeric(df[test] == 0 & df[truth] == 0))
  FN <- sum(as.numeric(df[test] == 0 & df[truth] == 1))
  #sens = Sensitivity = TP/(TP+FP)
  sensitivity <- TP / (TP + FN)
  return(sensitivity)
}
specificity <- function(df, test, truth) {
  TP <- sum(as.numeric(df[test] == 1 & df[truth] == 1))
  FP <- sum(as.numeric(df[test] == 1 & df[truth] == 0))
  TN <- sum(as.numeric(df[test] == 0 & df[truth] == 0))
  FN <- sum(as.numeric(df[test] == 0 & df[truth] == 1))
  #spec = Specificity = TN/(TN+FN)
  specificity <- TN / (TN + FP)
  return(specificity)
}
# Overall sensitivity
sens <- sensitivity(rats, "HIT", "STATUS")
# Overall specificity
spec <- specificity(rats, "HIT", "STATUS")
sens;spec
```

```
## [1] 0.6193429
```

```
## [1] 0.9119767
```

## 3. Clean Data

Show how performance of rats differs with REUSED:

```
reuse0 <- subset(rats,rats$REUSED==0)
reuse1 <- subset(rats,rats$REUSED==1)
reuse2 <- subset(rats,rats$REUSED==2)
reuse3 <- subset(rats,rats$REUSED==3)
reuse4 <- subset(rats,rats$REUSED==4)
# Overall sensitivity
sens0 <- sensitivity(reuse0, "HIT", "STATUS")
sens1 <- sensitivity(reuse1, "HIT", "STATUS")
sens2 <- sensitivity(reuse2, "HIT", "STATUS")
sens3 <- sensitivity(reuse3, "HIT", "STATUS")
sens4 <- sensitivity(reuse4, "HIT", "STATUS")
# Overall specificity
spec0 <- specificity(reuse0, "HIT", "STATUS")
spec1 <- specificity(reuse1, "HIT", "STATUS")
spec2 <- specificity(reuse2, "HIT", "STATUS")
spec3 <- specificity(reuse3, "HIT", "STATUS")
spec4 <- specificity(reuse4, "HIT", "STATUS")
df0 <- rbind(length(reuse0$SESSION_DATE),
             length(reuse1$SESSION_DATE),
             length(reuse2$SESSION_DATE),
             length(reuse3$SESSION_DATE),
             length(reuse4$SESSION_DATE))
df1 <- rbind(sens0,sens1,sens2,sens3,sens4)
df2 <- rbind(spec0,spec1,spec2,spec3,spec4)
df3 <- rbind(sum(reuse0$STATUS)/length(reuse0$SESSION_DATE),
             sum(reuse1$STATUS)/length(reuse1$SESSION_DATE),
             sum(reuse2$STATUS)/length(reuse2$SESSION_DATE),
             sum(reuse3$STATUS)/length(reuse3$SESSION_DATE),
             sum(reuse4$STATUS)/length(reuse4$SESSION_DATE)
)
overall <- cbind(nrow(rats),sum(rats$STATUS)/nrow(rats),sens,spec)
by.reuse <- cbind(df0,df3,df1,df2)
by.reuse <- rbind(by.reuse,overall)
rownames(by.reuse) <- c("Never Reused",
                        "Reused Once",
                        "Reused 2 Times",
                        "Reused 3 Times",
                        "Reused 4 Times",
                        "All Samples")
colnames(by.reuse)<- c("# of Samples","Positive Rate","Sensitivity","Specificity")
by.reuse
```

```
##                # of Samples Positive Rate Sensitivity Specificity
## Never Reused             9     0.0000000         NaN   1.0000000
## Reused Once          87034     0.1092562   0.4581975   0.9135634
## Reused 2 Times        7764     0.4811953   0.8525161   0.8815789
## Reused 3 Times        1308     0.9770642   0.9522692   0.8666667
## Reused 4 Times         564     1.0000000   0.9911348         NaN
## All Samples          96750     0.1566718   0.6193429   0.9119767
```

Performance of rats varies as number of times of the sample being reused.

Use data with REUSED<=1:
cdata: clean data

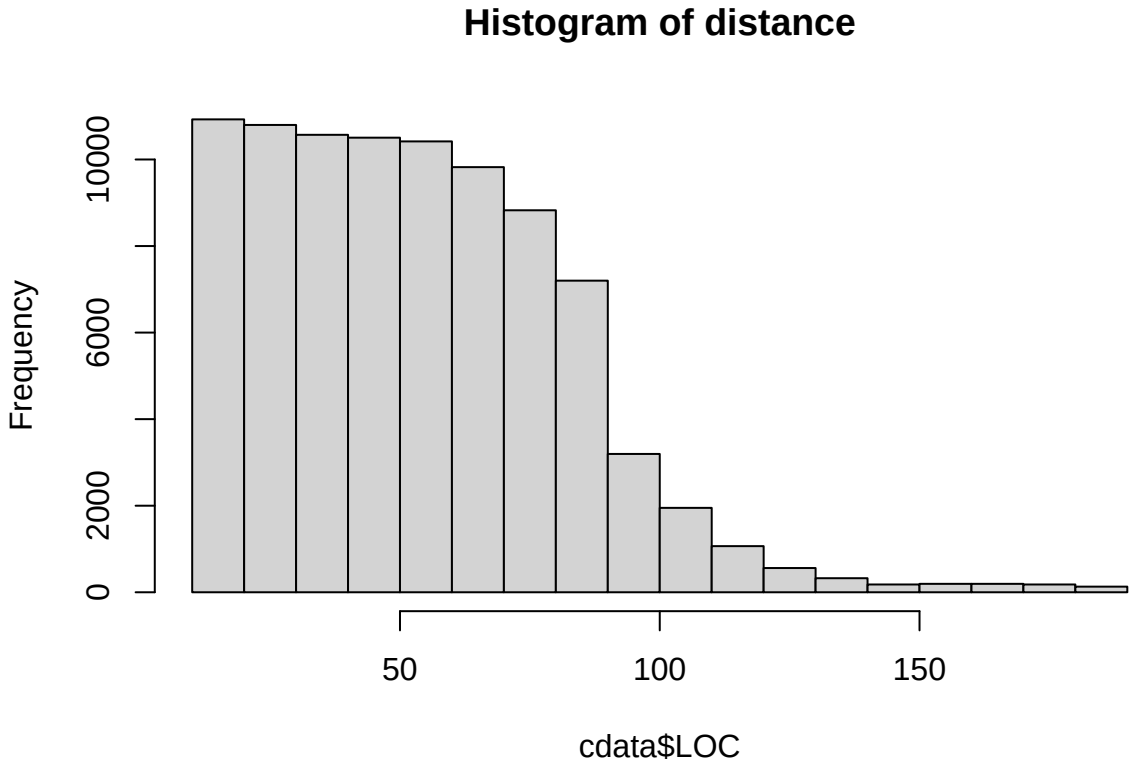
```
#cdata contains reused <= 1 samples
cdata <- subset(rats,rats$REUSED<=1)
cdata$LEVEL <- ifelse(is.na(cdata$ID_BL_APOPO)==1,cdata$ID_BL_DOTS,cdata$ID_BL_APOPO)
cdata$STATUS <- as.numeric(cdata$LEVEL !=1)
```

## 4. Potential Analysis

### 4.1 Precise Location of Hole

distribution of distance and compare performance across distance

```
#cdata <- rats
alphabet <- sort(unique(cdata$RUN))
cdata$RUN.new <- match(cdata$RUN,alphabet)
cdata$LOC <- cdata$RUN.new*10 + cdata$HOLE
cdata$result <- as.numeric(cdata$HIT==cdata$STATUS)
#sum(is.na(cdata$LOC)==1)
hist(cdata$LOC,main="Histogram of distance")
```




```
loc.min <- min(cdata$LOC)
loc.max <- max(cdata$LOC)
df.loc <- NA
for (distance in loc.min:loc.max){
  sub <- subset(cdata,cdata$LOC==distance)
  sens <- sensitivity(sub,"HIT","STATUS")
  spec <- specificity(sub,"HIT","STATUS")
  count <- length(sub$SESSION_DATE)
```

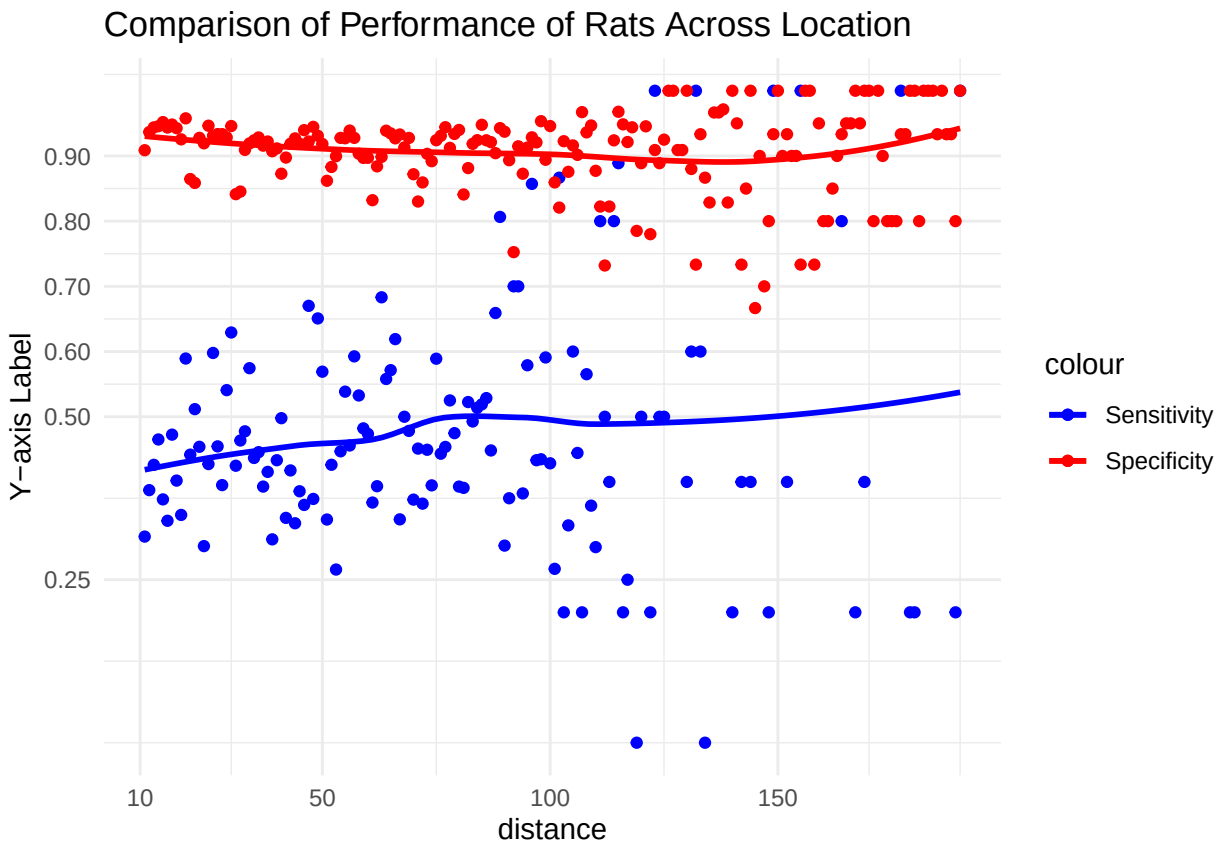
```
  temp <- data.frame(distance,count,sens,spec)
  df.loc <- rbind(df.loc,temp)
}
df.loc <- subset(df.loc,is.na(df.loc)==0)
df.loc <- subset(df.loc,df.loc$distance>=loc.min&df.loc$distance<=loc.max)
df.loc.count <- subset(df.loc,df.loc$count>=100&is.na(df.loc$sens)==0)
#df.loc.count
check <- subset(cdata,cdata$LOC==118)
#sum(check$STATUS==1)
ggplot(df.loc, aes(x = distance)) +
  geom_point(aes(y = sens, color = "Sensitivity")) +
  geom_smooth(aes(y = sens, color = "Sensitivity"), se = FALSE) +
  geom_point(aes(y = spec, color = "Specificity")) +
  geom_smooth(aes(y = spec, color = "Specificity"), se = FALSE) +
  ylab("Y-axis Label") +
  scale_color_manual(values = c("Sensitivity" = "blue", "Specificity" = "red")) +
  scale_x_continuous(breaks = c(10, 50, 100, 150)) +
  scale_y_continuous(breaks = c(0.25, 0.5, 0.6, 0.7, 0.8, 0.9)) +
  ggtitle("Comparison of Performance of Rats Across Location") +
  theme_minimal()
```

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

```
## Warning: Removed 47 rows containing non-finite values (`stat_smooth()`).
```

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

```
## Warning: Removed 47 rows containing missing values (`geom_point()`).
```

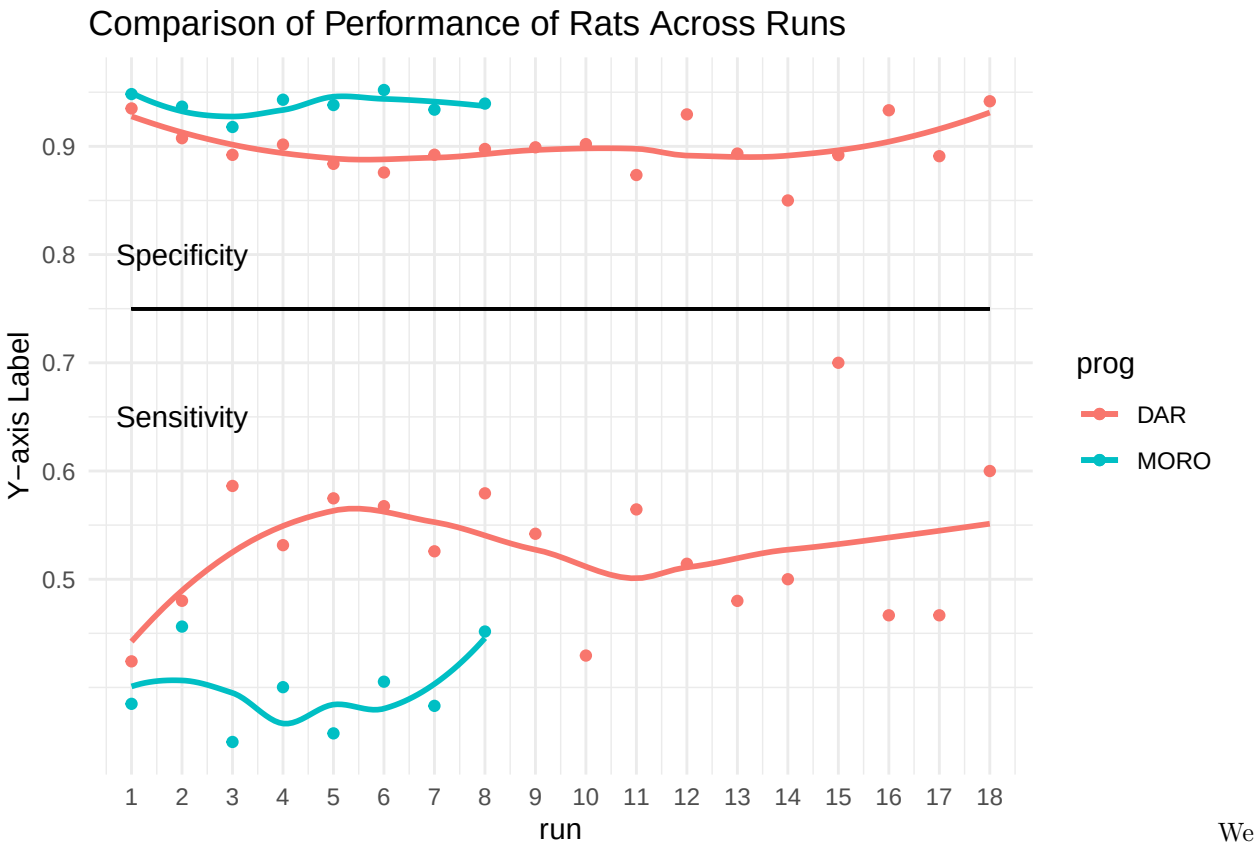


The performance of rats seem not to have some monotone trends when distances of samples increases.

```
df.run <- NA
PROGRAM <- c("DAR","MORO")
for (prog in PROGRAM){
  sub.prog <- subset(cdata,cdata$PROGRAM==prog)
  run.min <- min(sub.prog$RUN.new)
  run.max <- max(sub.prog$RUN.new)
      for (run in run.min:run.max){
        sub.run <- subset(sub.prog,sub.prog$RUN.new==run)
        sens <- sensitivity(sub.run,"HIT","STATUS")
        spec <- specificity(sub.run,"HIT","STATUS")
        count <- nrow(sub.run)
        temp <- data.frame(prog,run,count,sens,spec)
        df.run <- rbind(df.run,temp)
      }
}
df.run <- subset(df.run,is.na(df.run$prog)==0)
ggplot(df.run, aes(x = run)) +
  geom_point(aes(y = sens, color = prog,group=prog)) +
  geom_smooth(aes(y = sens, color = prog), se = FALSE) +
  geom_point(aes(y = spec, color = prog,group=prog)) +
  geom_smooth(aes(y = spec, color = prog), se = FALSE) +
  ylab("Y-axis Label") +
  #scale_color_manual(values = c("Sensitivity" = "blue", "Specificity" = "red")) +
  scale_x_continuous(breaks = c(1:18)) +
  scale_y_continuous(breaks = c(0.25, 0.5, 0.6, 0.7, 0.8, 0.9)) +
  ggtitle("Comparison of Performance of Rats Across Runs") +
  geom_segment(aes(x = 1, y = 0.75, xend = 18, yend = 0.75))+
  annotate("text", x = 2, y = 0.8, label = "Specificity")+
  annotate("text", x = 2, y = 0.65, label = "Sensitivity")+
  theme_minimal()
```

**by different programs**

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



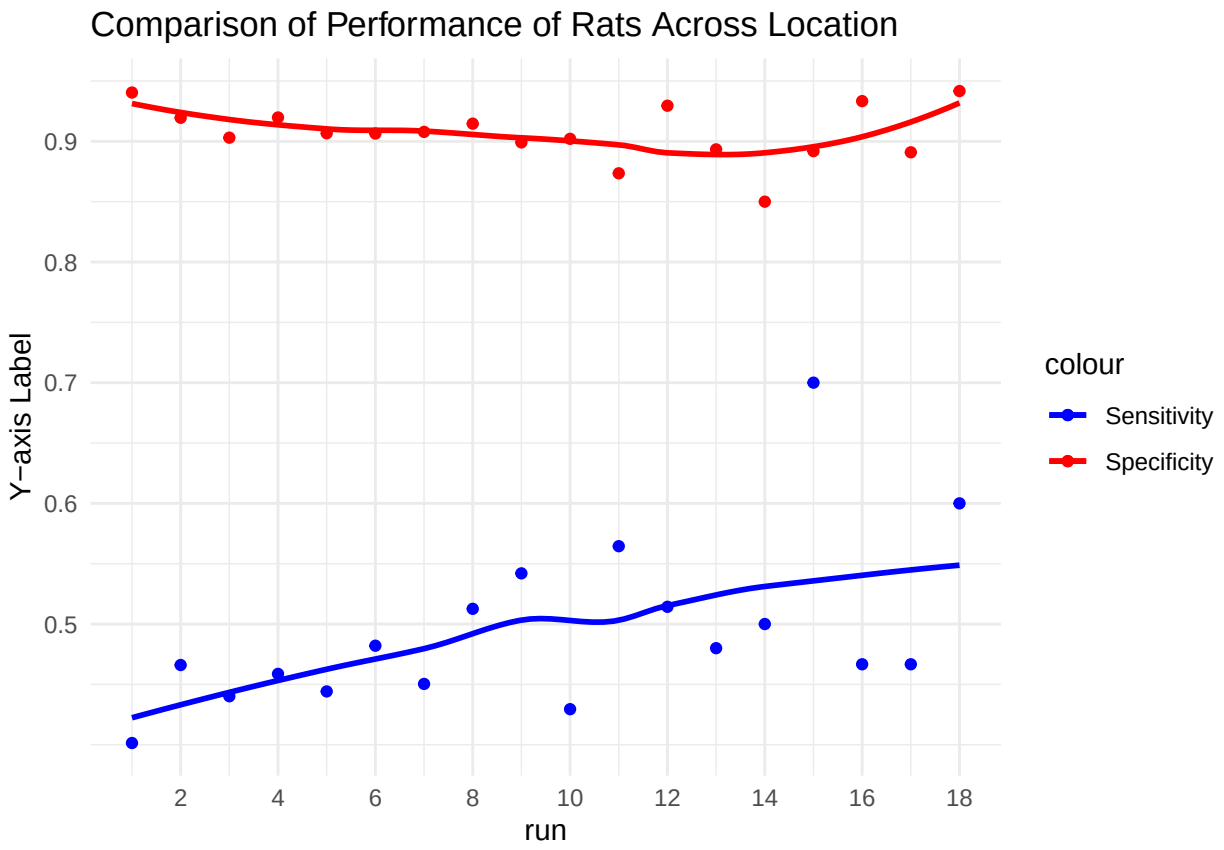

We furthur stratify the data by two programs, still no obvious pattern found.

### 4.2 Less Precise Location of Hole

```
run.min <- min(cdata$RUN.new)
run.max <- max(cdata$RUN.new)
df.run <- NA
for (run in run.min:run.max){
  sub <- subset(cdata,cdata$RUN.new==run)
  sens <- sensitivity(sub,"HIT","STATUS")
  spec <- specificity(sub,"HIT","STATUS")
  count <- length(sub$SESSION_DATE)
  temp <- data.frame(run,count,sens,spec)
  df.run <- rbind(df.run,temp)
}
df.run <- subset(df.run,is.na(df.run)==0)
ggplot(df.run, aes(x = run)) +
  geom_point(aes(y = sens, color = "Sensitivity")) +
  geom_smooth(aes(y = sens, color = "Sensitivity"), se = FALSE) +
  geom_point(aes(y = spec, color = "Specificity")) +
  geom_smooth(aes(y = spec, color = "Specificity"), se = FALSE) +
  ylab("Y-axis Label") +
  scale_color_manual(values = c("Sensitivity" = "blue", "Specificity" = "red")) +
  scale_x_continuous(breaks = c(2,4,6,8,10,12,14,16,18)) +
  scale_y_continuous(breaks = c(0.25, 0.5, 0.6, 0.7, 0.8, 0.9)) +
  ggtitle("Comparison of Performance of Rats Across Location") +
  theme_minimal()
```

**compare performance across RUN:**

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## Warning: Removed 54 rows containing non-finite values (`stat_smooth()`).
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## Warning: Removed 54 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 54 rows containing missing values (`geom_point()`).
## Removed 54 rows containing missing values (`geom_point()`).
```




```
summary(lm(run~sens,data=df.run))
```

```
##
## Call:
## lm(formula = run ~ sens, data = df.run)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.255 -3.091 -1.820  3.885  8.719
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -11.444      7.785  -1.470   0.1609
## sens          42.268     15.561   2.716   0.0153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.552 on 16 degrees of freedom
##   (54 observations deleted due to missingness)
## Multiple R-squared:  0.3156, Adjusted R-squared:  0.2728
## F-statistic: 7.379 on 1 and 16 DF,  p-value: 0.01525
```

This is surprising. There might exist positive correlation between sensitivity and run position.

What's more surprising is that, the run that has highest performance actually has low positive rate

```
run.15 <- subset(cdata,cdata$RUN.new==15)
sum(run.15$STATUS)/length(run.15$SESSION_DATE)
```

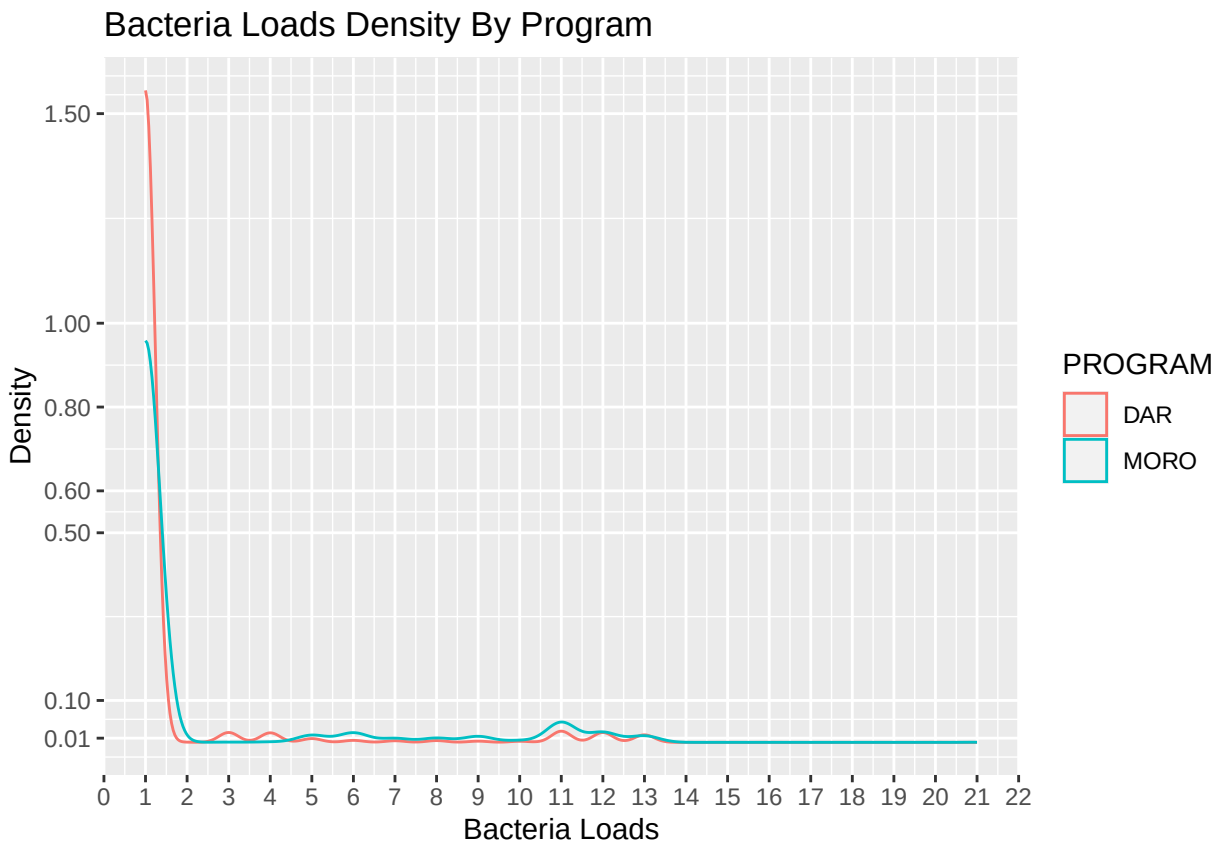
```
## [1] 0.05128205
```

My thoughts are that even less precise calculation of distance doesn't show any obvious pattern, I would consider distance as an insignificant factor in terms of performance of rats.

### 4.3 Bacterial Loads

distribution of Bacterial Loads in 2 programs:

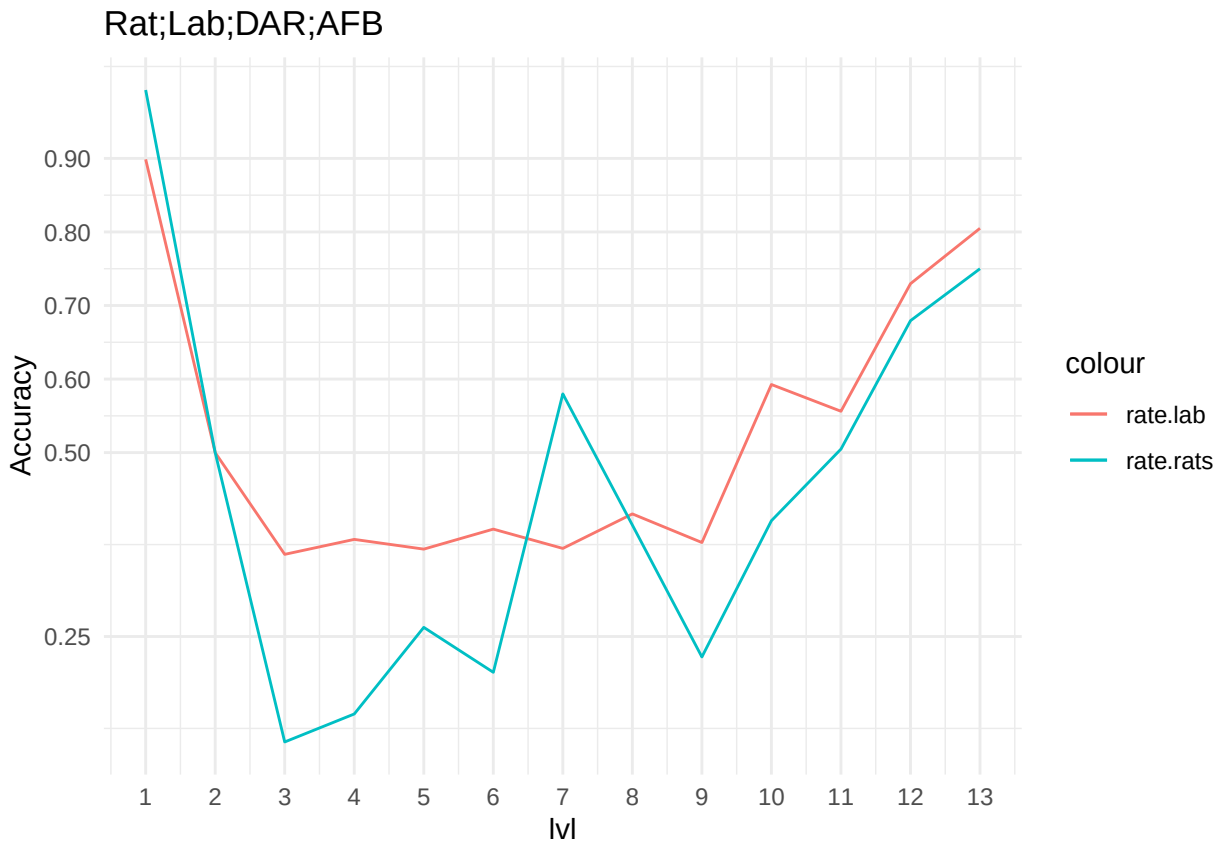
```
becteria.load <-  ggplot(cdata, aes(x=LEVEL, color=PROGRAM)) +
                  scale_y_continuous(breaks = c(0.01,0.1,0.5, 0.6, 0.8,1.0,1.5)) +
                  scale_x_continuous(breaks = 0:24) +
                  xlab("Bacteria Loads")+
                  ylab("Density")+
                  ggtitle("Bacteria Loads Density By Program")+
                  geom_density()
becteria.load
```



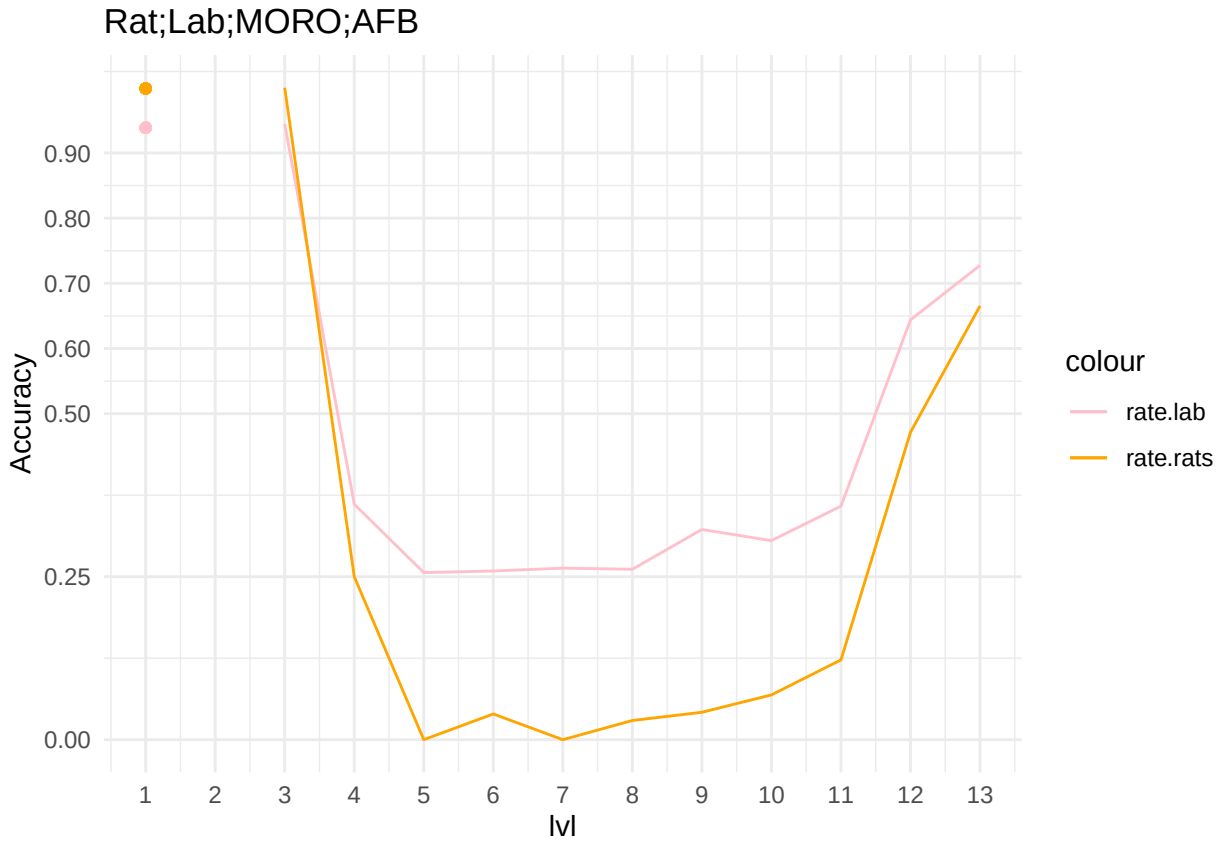
Comment: The majority of samples has low bacteria loads.

lab vs rat based on the bacterial loads (by two different locations)

```
cdata$LAB.RESULT <- as.numeric(cdata$ID_BL_DOTS!=1)
df <- data.frame()
DAR <- subset(cdata,cdata$PROGRAM=="DAR")
range <- 1:13
for (lvl in range){
    lvl.df <- subset(DAR,DAR$LEVEL==lvl)
    rate.lab <- sum(lvl.df$HIT==lvl.df$STATUS)/nrow(lvl.df)
    rate.rats <- sum(lvl.df$LAB.RESULT==lvl.df$STATUS)/nrow(lvl.df)
    count <- sum(cdata$PROGRAM=="DAR"&cdata$LEVEL==lvl)
    temp <- data.frame(Prog = "DAR",lvl,count,rate.lab,rate.rats)
    df <- rbind(df,temp)
}
df.DAR <- df
df <- data.frame()
MORO <- subset(cdata,cdata$PROGRAM=="MORO")
for (lvl in range){
    lvl.df <- subset(MORO,MORO$LEVEL==lvl)
    rate.lab <- sum(lvl.df$HIT==lvl.df$STATUS)/nrow(lvl.df)
    rate.rats <- sum(lvl.df$LAB.RESULT==lvl.df$STATUS)/nrow(lvl.df)
    count <- sum(cdata$PROGRAM=="MORO"&cdata$LEVEL==lvl)
    temp <- data.frame(Prog = "MORO",lvl,count,rate.lab,rate.rats)
    df <- rbind(df,temp)
}
df.MORO <- df
dar <- ggplot(df.DAR, aes(x = lvl)) +
  geom_line(aes(y = rate.lab, color = "rate.lab")) +
  geom_line(aes(y = rate.rats, color = "rate.rats")) +
  ylab("Accuracy") +
  scale_x_continuous(breaks = 1:13) +
  scale_y_continuous(breaks = c(0,0.25,0.5,0.6,0.7,0.8,0.9)) +
  ggtitle("Rat;Lab;DAR;AFB")+
  theme_minimal()
moro <- ggplot(df.MORO, aes(x = lvl)) +
  geom_line(aes(y = rate.lab, color = "rate.lab")) +
  geom_line(aes(y = rate.rats, color = "rate.rats")) +
  scale_color_manual(values = c("rate.lab" = "pink", "rate.rats" = "orange")) +
  ylab("Accuracy") +
  scale_x_continuous(breaks = 1:13) +
  scale_y_continuous(breaks = c(0,0.25,0.5,0.6,0.7,0.8,0.9)) +
  ggtitle("Rat;Lab;MORO;AFB")+
  geom_point(aes(x=1,y=0.9386080),colour="pink")+
  geom_point(aes(x=1,y=0.99882271),colour="orange")+
  theme_minimal()
dar
```


Rat;Lab;DAR;AFB
Accuracy
0.90
0.80
0.70
0.60
0.50
0.25
1
2
3
4
5
6
7
8
9
10
11
12
13
lvl
colour
rate.lab
rate.rats

```
moro
```


Rat;Lab;MORO;AFB
Accuracy
0.90
0.80
0.70
0.60
0.50
0.25
0.00
1
2
3
4
5
6
7
8
9
10
11
12
13
lvl
colour
rate.lab
rate.rats

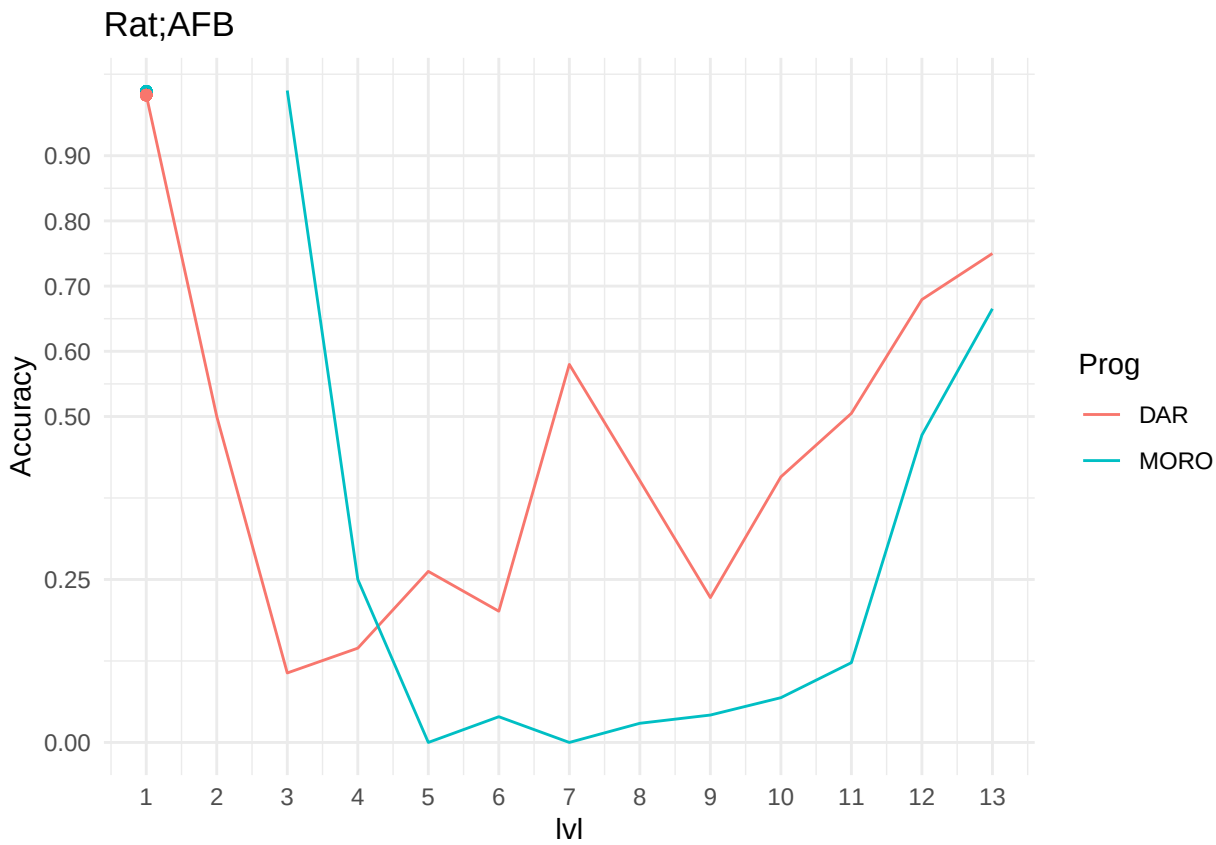
Comment: Rats are better than microscopes at certain interval of bacteria loads. However, there are few samples at some levels, and this can't be ignored.

**performance of rats across different levels of bectaria loads in two programs** We also want to compare how two programs differ from each other to help improve quality of experiments.

```
df.comb <- rbind(df.DAR,df.MORO)
#df.comb
plot.comb <- ggplot(df.comb, aes(x = lvl)) +
  geom_line(aes(y = rate.rats, group=Prog,color = Prog)) +
  ylab("Accuracy") +
  xlim(1,13)+
  scale_x_continuous(breaks = 1:13) +
  scale_y_continuous(breaks = c(0,0.25,0.5,0.6,0.7,0.8,0.9)) +
  ggtitle("Rat;AFB")+
  geom_point(aes(x=1,y=0.99882271),colour="#00BFC4")+
  geom_point(aes(x=1,y=0.9929913),colour="#F8766D")+
  theme_minimal()
```

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

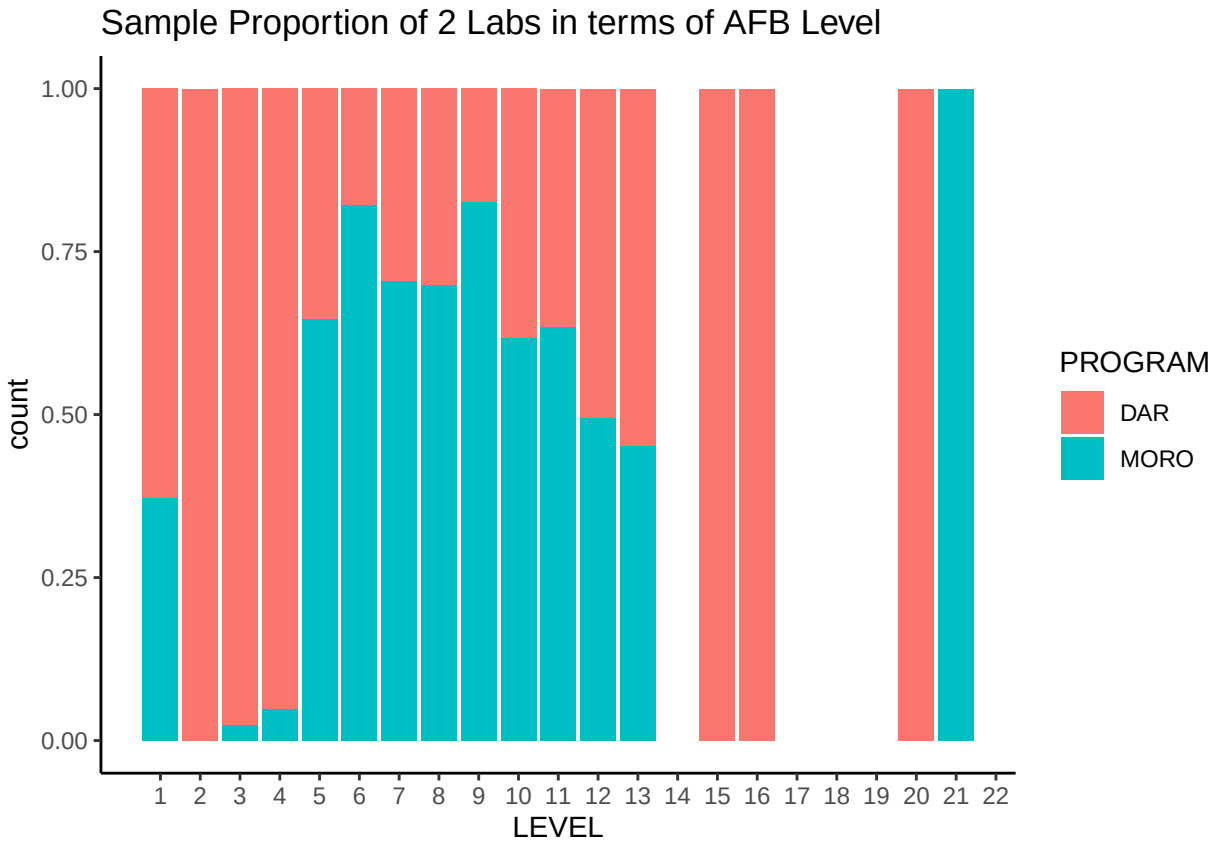
```
plot.comb
```




sample Proportion of 2 Labs in terms of AFB Level

```
ggplot(cdata, aes(x = LEVEL, fill = PROGRAM)) +
    geom_bar(position = "fill") +
```

```
    scale_x_continuous(breaks = 1:24) +
    ggtitle("Sample Proportion of 2 Labs in terms of AFB Level")+
    theme_classic()
```

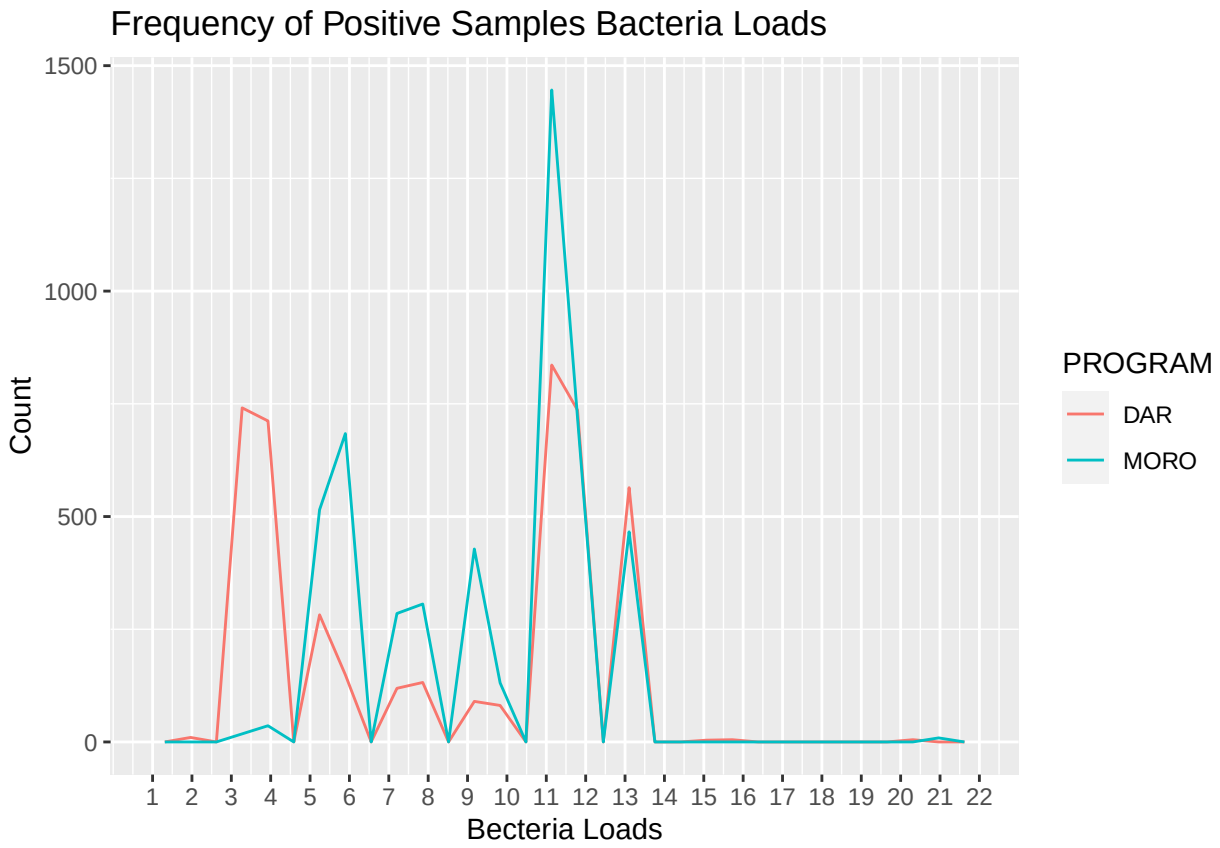



We suspect the difference is a result of different structures of samples in two programs in term of proportion of samples at each level.

**frequency of positive samples becteria loads** As requested in potential analysis, we are also interested in frequency of positive samples becteria loads.

```
pos.samples <- subset(cdata,cdata$STATUS==1)
ggplot(pos.samples, aes(x = LEVEL,color=PROGRAM)) +
          ggtitle("Frequency of Positive Samples Bacteria Loads")+
          scale_x_continuous(breaks = 1:28) +
          xlab("Becteria Loads")+
          ylab("Count")+
          geom_freqpoly()
```

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



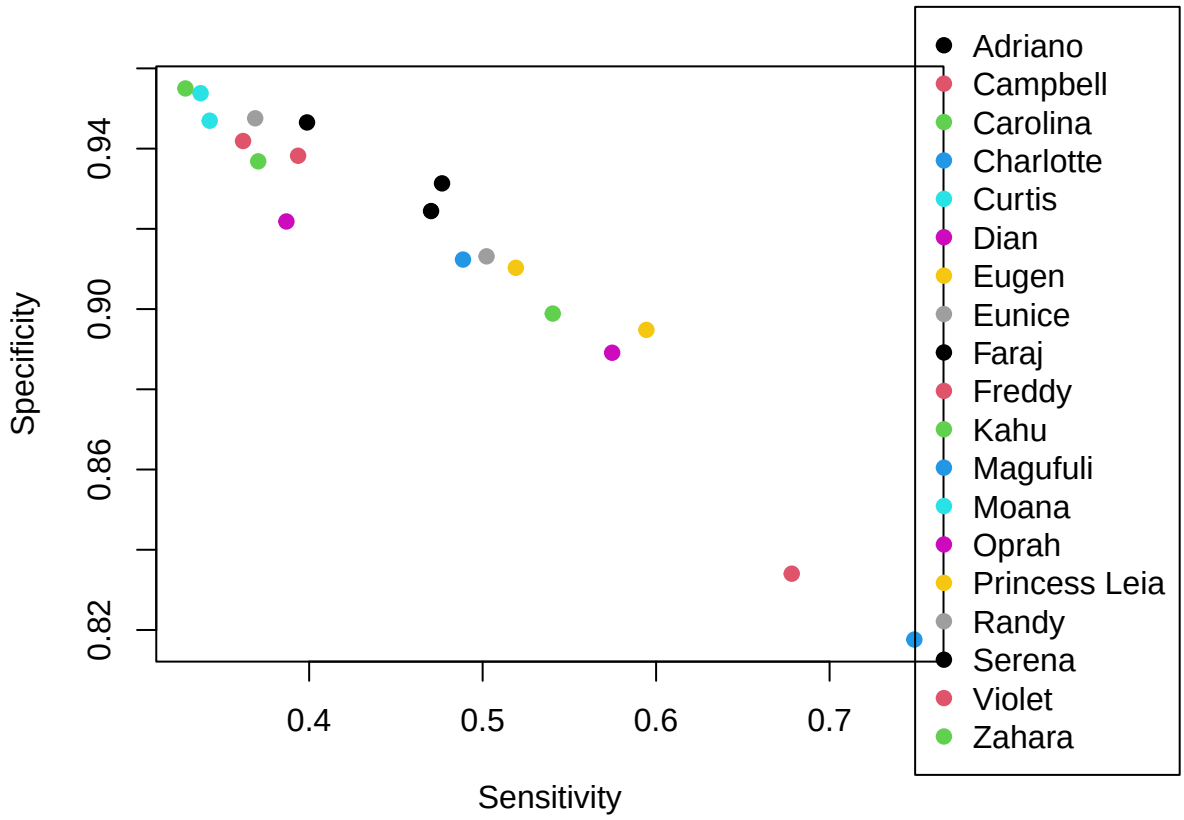

Comment: There is indeed some evidence the frequences are different in lower levels of becteria loads.

### 4.4 trade-off between sensitvity and specificity

```
rats <- subset(rats, rats$REUSED <= 1)
rats_names <- unique(rats$RAT_NAME)
sen_spec_rat <- data.frame(matrix(nrow = length(rats_names), ncol = 2),
                           row.names = rats_names)
colnames(sen_spec_rat) <- c("Sensitivity", "Specificity")
for (name in rats_names) {
  # A subset of rats data with specified rat name
  specific_rat <- subset(rats, rats$RAT_NAME == name)
  # rat sensitivity
  rat_sens <- sensitivity(specific_rat, "HIT", "STATUS")
  # rat specificity
  rat_spec <- specificity(specific_rat, "HIT", "STATUS")
  sen_spec_rat[name, ] <- c(rat_sens, rat_spec)
}
sen_spec_rat["RatName"] <- rats_names
par(mar = c(4, 4, 3, 8), xpd = TRUE)
plot(sen_spec_rat$Sensitivity, sen_spec_rat$Specificity,
     xlab = "Sensitivity", ylab = "Specificity", pch=19,
     col = as.factor(sen_spec_rat$RatName))

legend("topright", inset=c(-0.3, -0.1),
       legend = levels(factor(sen_spec_rat$RatName)),
       pch = 19,
```

```
      col = factor(levels(factor(sen_spec_rat$RatName))))
```




Comment: There is a nagetive correlation between sensitivity and specificity

### 4.5 Individual rats sensitivity and specificity

```
head(sen_spec_rat)
```

```
##                Sensitivity Specificity       RatName
## Freddy           0.6781915   0.8340342        Freddy
## Carolina         0.5404255   0.8988786      Carolina
## Serena           0.4765957   0.9313398        Serena
## Princess Leia    0.5191489   0.9102892 Princess Leia
## Violet           0.3936170   0.9382255        Violet
## Charlotte        0.7488688   0.8176126     Charlotte
```

### 4.6 Optimal Team Selection by Poisson-Binomial Model

More details can be found on: https://www.researchgate.net/publication/257017356_On_computing_the_distribution_function_for_the_Poisson_binomial_distribution/

Algorithm:
1. Input $\theta_j$ and $\gamma_j$ along with names of rats;
2. Drop rats with lowest $\theta_j$ and $\gamma_j$ to prevent R from running out of memory;
3. Find all possible combination of team of rats with team size ranging from 5 to 18;
4. Compute $\theta_{team}$ and $\gamma_{team}$ for all possible teams;
5. Compute prevalence rate $p$;
6. Compute $team\ accuracy_i = p\theta_{team_i} + (1-p)\gamma_{team_i}$;
7. Rank the teams by team accuracy (highest to lowest).

```
best.team <- function(df) {
    team.df <- data.frame() # Initialize an empty data frame
    team.number <- 0
    # find all possible combinations of team of rats
    combinations <- lapply(2:nrow(df), function(n) {
        t(combn(rownames(df), n))
    })
    for (m in 5:length(combinations)-1) {
    # m is team size starts from 5
    # comb.sub contains combinations of teams with size m
    comb.sub <- combinations[[m]]
        for (i in 1:dim(comb.sub)[1]) {
        # ith team
        team <- comb.sub[i, ]
        team.member <- toString(team)
        # pp is vector that contains individual sensitivity in the team
        pp <- df[team, "Sensitivity"]
        # pp2 is vector that contains individual specificity in the team
        pp2 <- df[team, "Specificity"]
        team.se <- ppoisbinom(2-1, pp,lower_tail=FALSE )
        team.sp <- ppoisbinom(m-2, pp2, lower_tail = FALSE)
        team.number <- team.number+1
        #temp <- data.frame(team.se,team.sp, team.member,thershold,team.number)
        temp <- data.frame(team.se,team.sp, team.member)
        team.df <- rbind(team.df,temp)
        }
    }
return(team.df)
}
```

```
df.raw <- sen_spec_rat
df <- df.raw %>%
    filter(Sensitivity > min(Sensitivity), Specificity > min(Specificity))
team.df <- best.team(df)
head(team.df)
```

**compute team sensitivity and specificity**

```
##     team.se   team.sp                                        team.member
## 1 0.8437486 0.9927360   Freddy, Carolina, Serena, Princess Leia, Violet
## 2 0.8650750 0.9914081   Freddy, Carolina, Serena, Princess Leia, Eunice
## 3 0.8624104 0.9913653 Freddy, Carolina, Serena, Princess Leia, Magufuli
## 4 0.8792864 0.9901354    Freddy, Carolina, Serena, Princess Leia, Oprah
## 5 0.8424258 0.9918680     Freddy, Carolina, Serena, Princess Leia, Dian
## 6 0.8327160 0.9935614   Freddy, Carolina, Serena, Princess Leia, Curtis
```

```
unique_rows <- rats[rats$ID_SAMPLE %in% unique(rats$ID_SAMPLE), ]
#p <- sum(unique_rows$STATUS)/nrow(unique_rows)
p <- sum(cdata$STATUS)/nrow(cdata)
team.ac <- team.df$team.se* p  + team.df$team.sp * (1-p)
team.df$team.ac <- team.ac
team.df <- team.df %>%
```

```
  arrange(desc(team.ac))
head(team.df)
```

**compute team accuracy and select best team**

```
##     team.se   team.sp
## 1 0.9034658 0.9905560
## 2 0.9077458 0.9899515
## 3 0.9193147 0.9885317
## 4 0.9113461 0.9894871
## 5 0.9223782 0.9881035
## 6 0.9189552 0.9885186
##                                                    team.member   team.ac
## 1  Serena, Princess Leia, Curtis, Eugen, Randy, Adriano, Faraj 0.9810418
## 2 Serena, Princess Leia, Eunice, Curtis, Eugen, Randy, Adriano 0.9809710
## 3 Serena, Princess Leia, Eunice, Curtis, Eugen, Adriano, Faraj 0.9809701
## 4          Serena, Oprah, Curtis, Eugen, Randy, Adriano, Faraj 0.9809506
## 5  Serena, Princess Leia, Eunice, Eugen, Randy, Adriano, Faraj 0.9809233
## 6  Serena, Violet, Curtis, Eugen, Moana, Randy, Adriano, Faraj 0.9809192
```