

# Checking Discriminant Taxa By OTU

I am using this script to check the consistency of the discriminant taxa across the different comparisons. I will be checking the discriminant taxa for each comparison and then checking to see if the same OTUs are discriminant in the other comparisons.

For example, if an OTU is discriminant for SF in the SJ v. SF shallow comparison, I will check to see if it is also discriminant for SF in the SF v. TP shallow comparison. If it is, then I will consider it a consistent discriminant taxa for SF in the 0-5 cm depth.

16S

## CORRECTED CODE

CHECKED BY EMILY BECHTOLD APRIL 2025

- sf v. sj shallow: when the coef is negative it is discriminant for SF and when its positive its discriminant for SJ
- sf v. tp shallow: when the coef is negative it is discriminant for SF and when its positive its discriminant for TP
- tp v. sj shallow: when the coef is negative it is discriminant for SJ and when its positive its discriminant for TP
- sj v. sf deep: when the coef is negative it is discriminant for SF and when its positive its discriminant for SJ
- sf v. tp deep: when the coef is negative it is discriminant for SF and when its positive its discriminant for TP
- tp v. sj deep: when the coef is negative it is discriminant for SJ and when its positive its discriminant for TP
- sj v. sf OM: when the coef is negative it is discriminant for SF and when its positive its discriminant for SJ
- sf v. tp OM: when the coef is negative it is discriminant for SF and when its positive its discriminant for TP
- tp v. sj OM: when the coef is negative it is discriminant for SJ and when its positive its discriminant for TP

## Step 1: Load the required Libraries

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.4      v readr      2.1.5
## v forcats   1.0.0      v stringr   1.5.1
## v lubridate 1.9.3      v tibble   3.2.1
## v purrr     1.0.2      v tidyr    1.3.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(phyloseq)
library(dplyr)
library(readxl)
library(ggrepel)
library(openxlsx)
```

## Step 2: Load the data

Note: Read in all comparisons from the Excel file. The data is organized by shallow, deep, and OM samples. Each comparison is stored in a separate sheet within the Excel file.

```
# Load data
data_SJSF_shallow <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_1.xlsx")
data_SFTP_shallow <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_1.xlsx")
data_SJTP_shallow <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_1.xlsx")

data_SJSF_deep <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_2.xlsx")
data_SFTP_deep <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_2.xlsx")
data_SJTP_deep <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_2.xlsx")

data_SJSF_OM <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_3.xlsx")
data_SFTP_OM <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_3.xlsx")
data_SJTP_OM <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_3.xlsx")
```

## Step 3: Create functions to classify discriminant OTUs

Note: The function `classify_discriminant` takes a data frame and two group names as input. It filters the data for significant OTUs ( $qval < 0.05$ ) and classifies them based on the sign of the coefficient. If the coefficient is negative, it assigns the first group name; if positive, it assigns the second group name.

```
# Function to clasfsify discriminant OTUs
classify_discriminant <- function(data, group1, group2) {
  data %>%
    filter(qval < 0.05) %>% # Consider only significant OTUs
    mutate(discriminant_for = ifelse(coef < 0, group1, group2))
}
```

## Step 4: Apply the classification function to each dataset

```
# Apply classification
data_SJSF_shallow <- classify_discriminant(data_SJSF_shallow, "SF", "SJ")
data_SFTP_shallow <- classify_discriminant(data_SFTP_shallow, "SF", "TP")
data_SJTP_shallow <- classify_discriminant(data_SJTP_shallow, "SJ", "TP")

data_SJSF_deep <- classify_discriminant(data_SJSF_deep, "SF", "SJ")
data_SFTP_deep <- classify_discriminant(data_SFTP_deep, "SF", "TP")
data_SJTP_deep <- classify_discriminant(data_SJTP_deep, "SJ", "TP")

data_SJSF_OM <- classify_discriminant(data_SJSF_OM, "SF", "SJ")
data_SFTP_OM <- classify_discriminant(data_SFTP_OM, "SF", "TP")
data_SJTP_OM <- classify_discriminant(data_SJTP_OM, "SJ", "TP")
```

## Step 5: Filter and select relevant columns

Note: The filter function is used to select only the rows where the discriminant is for a specific group. The select function is used to keep only the relevant columns (OTUID and taxonomy). This filters the discriminant OTUs for each group and selects the relevant columns (OTUID and taxonomy).

### 16S SF 0-5 cm comparison

```
data_SJSF_shallow_filt <- data_SJSF_shallow %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

data_SFTP_shallow_filt <- data_SFTP_shallow %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

SF5_discrim <- inner_join(data_SJSF_shallow_filt, data_SFTP_shallow_filt, by = "OTUID")
```

Looking at discriminant taxa for SF 0-5cm: We do this by looking at the OTUs that are discriminant for SF in the SJ v. SF shallow comparison and the OTUs that are discriminant for SF in the SF v. TP shallow comparison.

### 16S TP 0-5 cm comparison

```
data_SJTP_shallow_filt <- data_SJTP_shallow %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

data_SFTP_shallow_filt <- data_SFTP_shallow %>%
  filter(discriminant_for == "TP") %>%
```

```
select(OTUID, taxonomy)

TP5_discrim <- inner_join(data_SJTP_shallow_filt, data_SFTP_shallow_filt, by = "OTUID")
```

Looking at discriminant taxa for TP 0-5cm: We do this by looking at the OTUs that are discriminant for TP in the SJ v. TP shallow comparison and the OTUs that are discriminant for TP in the SF v. TP shallow comparison.

#### 16S SJ 0-5 cm comparison

```
data_SJTP_shallow_filt <- data_SJTP_shallow %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

data_SJSF_shallow_filt <- data_SJSF_shallow %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

SJ5_discrim <- inner_join(data_SJTP_shallow_filt, data_SJSF_shallow_filt, by = "OTUID")
```

Looking at discriminant taxa for SJ 0-5cm: We do this by looking at the OTUs that are discriminant for SJ in the SJ v. TP shallow comparison and the OTUs that are discriminant for SJ in the SF v. SJ shallow comparison.

#### 16S SF 5-15 cm comparison

```
data_SJSF_deep_filt <- data_SJSF_deep %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

data_SFTP_deep_filt <- data_SFTP_deep %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

SF15_discrim <- inner_join(data_SJSF_deep_filt, data_SFTP_deep_filt, by = "OTUID")
```

Looking at discriminant taxa for SF 5-15cm: We do this by looking at the OTUs that are discriminant for SF in the SJ v. SF deep comparison and the OTUs that are discriminant for SF in the SF v. TP deep comparison.

#### 16S TP 5-15 cm comparison

```
data_SJTP_deep_filt <- data_SJTP_deep %>%
  filter(discriminant_for == "TP") %>%
```

```

select(OTUID, taxonomy)

data_SFTP_deep_filt <- data_SFTP_deep %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

TP15_discrim <- inner_join(data_SJTP_deep_filt, data_SFTP_deep_filt, by = "OTUID")

```

Looking at discriminant taxa for TP 5-15cm: We do this by looking at the OTUs that are discriminant for TP in the SJ v. TP deep comparison and the OTUs that are discriminant for TP in the SF v. TP deep comparison.

#### 16S SJ 5-15 cm comparison

```

data_SJTP_deep_filt <- data_SJTP_deep %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

data_SJSF_deep_filt <- data_SJSF_deep %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

SJ15_discrim <- inner_join(data_SJTP_deep_filt, data_SJSF_deep_filt, by = "OTUID")

```

Looking at discriminant taxa for SJ 5-15cm: We do this by looking at the OTUs that are discriminant for SJ in the SJ v. TP deep comparison and the OTUs that are discriminant for SJ in the SF v. SJ deep comparison.

#### 16S SF OM comparison

```

data_SJSF_OM_filt <- data_SJSF_OM %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

data_SFTP_OM_filt <- data_SFTP_OM %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

SFOM_discrim <- inner_join(data_SJSF_OM_filt, data_SFTP_OM_filt, by = "OTUID")

```

Looking at discriminant taxa for SF 5-15cm: We do this by looking at the OTUs that are discriminant for SF in the SJ v. SF deep comparison and the OTUs that are discriminant for SF in the SF v. TP deep comparison.

#### 16S TP OM comparison

```

data_SJTP_OM_filt <- data_SJTP_OM %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

data_SFTP_OM_filt <- data_SFTP_OM %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

TPOM_discrim <- inner_join(data_SJTP_OM_filt, data_SFTP_OM_filt, by = "OTUID")

```

Looking at discriminant taxa for TP 5-15cm: We do this by looking at the OTUs that are discriminant for TP in the SJ v. TP OM comparison and the OTUs that are discriminant for TP in the SF v. TP OM comparison.

#### 16S SJ OM comparison

```

data_SJTP_OM_filt <- data_SJTP_OM %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

data_SJSF_OM_filt <- data_SJSF_OM %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

SJOM_discrim <- inner_join(data_SJTP_OM_filt, data_SJSF_OM_filt, by = "OTUID")

```

Looking at discriminant taxa for SJ 5-15cm: We do this by looking at the OTUs that are discriminant for SJ in the SJ v. TP OM comparison and the OTUs that are discriminant for SJ in the SF v. SJ OM comparison.

#### Step 6: Save the results to an Excel file

Note: The openxlsx package is used to create an Excel workbook and add sheets for each comparison. The results are written to the respective sheets.

```

# Save results to Excel
wb <- createWorkbook()

# Add sheets
addWorksheet(wb, "SF5_discrim")
addWorksheet(wb, "TP5_discrim")
addWorksheet(wb, "SJ5_discrim")
addWorksheet(wb, "SF15_discrim")
addWorksheet(wb, "TP15_discrim")
addWorksheet(wb, "SJ15_discrim")
addWorksheet(wb, "SFOM_discrim")

```

```
addWorksheet(wb, "TPOM_discrim")
addWorksheet(wb, "SJOM_discrim")
```

```
# Write data
writeData(wb, "SF5_discrim", SF5_discrim)
writeData(wb, "TP5_discrim", TP5_discrim)
writeData(wb, "SJ5_discrim", SJ5_discrim)
writeData(wb, "SF15_discrim", SF15_discrim)
writeData(wb, "TP15_discrim", TP15_discrim)
writeData(wb, "SJ15_discrim", SJ15_discrim)
writeData(wb, "SFOM_discrim", SFOM_discrim)
writeData(wb, "TPOM_discrim", TPOM_discrim)
writeData(wb, "SJOM_discrim", SJOM_discrim)
```

```
saveWorkbook(wb, "/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_ks/16S/16S")
```

## ITS

### CORRECTED CODE

CHECKED BY EMILY BECHTOLD APRIL 2025

- sf v. sj shallow: when the coef is negative it is discriminant for SF and when its positive its discriminant for SJ
- sf v. tp shallow: when the coef is negative it is discriminant for SF and when its positive its discriminant for TP
- tp v. sj shallow: when the coef is negative it is discriminant for SJ and when its positive its discriminant for TP
- sj v. sf deep: when the coef is negative it is discriminant for SF and when its positive its discriminant for SJ
- sf v. tp deep: when the coef is negative it is discriminant for SF and when its positive its discriminant for TP
- tp v. sj deep: when the coef is negative it is discriminant for SJ and when its positive its discriminant for TP
- sj v. sf OM: when the coef is negative it is discriminant for SF and when its positive its discriminant for SJ
- sf v. tp OM: when the coef is negative it is discriminant for SF and when its positive its discriminant for TP
- tp v. sj OM: when the coef is negative it is discriminant for SJ and when its positive its discriminant for TP

### Step 1: Load the required Libraries

```
library(ggplot2)
library(tidyverse)
library(phyloseq)
library(dplyr)
library(readxl)
library(ggrepel)
library(openxlsx)
```

## Step 2: Load the data

Note: Read in all comparisons from the Excel file. The data is organized by shallow, deep, and OM samples. Each comparison is stored in a separate sheet within the Excel file.

```
# Load data
data_SJSF_shallow <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
data_SFTP_shallow <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
data_SJTP_shallow <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")

data_SJSF_deep <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
data_SFTP_deep <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
data_SJTP_deep <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")

data_SJSF_OM <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
data_SFTP_OM <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
data_SJTP_OM <- read_excel("/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS.xlsx")
```

## Step 3: Create functions to classify discriminant OTUs

Note: The function `classify_discriminant` takes a data frame and two group names as input. It filters the data for significant OTUs ( $qval < 0.05$ ) and classifies them based on the sign of the coefficient. If the coefficient is negative, it assigns the first group name; if positive, it assigns the second group name.

```
# Function to clasfsify discriminant OTUs
classify_discriminant <- function(data, group1, group2) {
  data %>%
    filter(qval < 0.05) %>% # Consider only significant OTUs
    mutate(discriminant_for = ifelse(coef < 0, group1, group2))
}
```

## Step 4: Apply the classification function to each dataset

```
# Apply classification
data_SJSF_shallow <- classify_discriminant(data_SJSF_shallow, "SF", "SJ")
data_SFTP_shallow <- classify_discriminant(data_SFTP_shallow, "SF", "TP")
data_SJTP_shallow <- classify_discriminant(data_SJTP_shallow, "SJ", "TP")

data_SJSF_deep <- classify_discriminant(data_SJSF_deep, "SF", "SJ")
data_SFTP_deep <- classify_discriminant(data_SFTP_deep, "SF", "TP")
data_SJTP_deep <- classify_discriminant(data_SJTP_deep, "SJ", "TP")

data_SJSF_OM <- classify_discriminant(data_SJSF_OM, "SF", "SJ")
data_SFTP_OM <- classify_discriminant(data_SFTP_OM, "SF", "TP")
data_SJTP_OM <- classify_discriminant(data_SJTP_OM, "SJ", "TP")
```



## Step 5: Filter and select relevant columns

Note: The filter function is used to select only the rows where the discriminant is for a specific group. The select function is used to keep only the relevant columns (OTUID and taxonomy). This filters the discriminant OTUs for each group and selects the relevant columns (OTUID and taxonomy).

### ITS SF 0-5 cm comparison

```
data_SJSF_shallow_filt <- data_SJSF_shallow %>%  
  filter(discriminant_for == "SF") %>%  
  select(OTUID, taxonomy)  
  
data_SFTP_shallow_filt <- data_SFTP_shallow %>%  
  filter(discriminant_for == "SF") %>%  
  select(OTUID, taxonomy)  
  
SF5_discrim <- inner_join(data_SJSF_shallow_filt, data_SFTP_shallow_filt, by = "OTUID")
```

Looking at discriminant taxa for SF 0-5cm: We do this by looking at the OTUs that are discriminant for SF in the SJ v. SF shallow comparison and the OTUs that are discriminant for SF in the SF v. TP shallow comparison.

### ITS TP 0-5 cm comparison

```
data_SJTP_shallow_filt <- data_SJTP_shallow %>%  
  filter(discriminant_for == "TP") %>%  
  select(OTUID, taxonomy)  
  
data_SFTP_shallow_filt <- data_SFTP_shallow %>%  
  filter(discriminant_for == "TP") %>%  
  select(OTUID, taxonomy)  
  
TP5_discrim <- inner_join(data_SJTP_shallow_filt, data_SFTP_shallow_filt, by = "OTUID")
```

Looking at discriminant taxa for TP 0-5cm: We do this by looking at the OTUs that are discriminant for TP in the SJ v. TP shallow comparison and the OTUs that are discriminant for TP in the SF v. TP shallow comparison.

### ITS SJ 0-5 cm comparison

```
data_SJTP_shallow_filt <- data_SJTP_shallow %>%  
  filter(discriminant_for == "SJ") %>%  
  select(OTUID, taxonomy)
```

```
data_SJSF_shallow_filt <- data_SJSF_shallow %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

SJ5_discrim <- inner_join(data_SJTP_shallow_filt, data_SJSF_shallow_filt, by = "OTUID")
```

Looking at discriminant taxa for SJ 0-5cm: We do this by looking at the OTUs that are discriminant for SJ in the SJ v. TP shallow comparison and the OTUs that are discriminant for SJ in the SF v. SJ shallow comparison.

#### ITS SF 5-15 cm comparison

```
data_SJSF_deep_filt <- data_SJSF_deep %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

data_SFTP_deep_filt <- data_SFTP_deep %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

SF15_discrim <- inner_join(data_SJSF_deep_filt, data_SFTP_deep_filt, by = "OTUID")
```

Looking at discriminant taxa for SF 5-15cm: We do this by looking at the OTUs that are discriminant for SF in the SJ v. SF deep comparison and the OTUs that are discriminant for SF in the SF v. TP deep comparison.

#### ITS TP 5-15 cm comparison

```
data_SJTP_deep_filt <- data_SJTP_deep %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

data_SFTP_deep_filt <- data_SFTP_deep %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

TP15_discrim <- inner_join(data_SJTP_deep_filt, data_SFTP_deep_filt, by = "OTUID")
```

Looking at discriminant taxa for TP 5-15cm: We do this by looking at the OTUs that are discriminant for TP in the SJ v. TP deep comparison and the OTUs that are discriminant for TP in the SF v. TP deep comparison.

#### ITS SJ 5-15 cm comparison

```

data_SJTP_deep_filt <- data_SJTP_deep %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

data_SJSF_deep_filt <- data_SJSF_deep %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

SJ15_discrim <- inner_join(data_SJTP_deep_filt, data_SJSF_deep_filt, by = "OTUID")

```

Looking at discriminant taxa for SJ 5-15cm: We do this by looking at the OTUs that are discriminant for SJ in the SJ v. TP deep comparison and the OTUs that are discriminant for SJ in the SF v. SJ deep comparison.

#### ITS SF OM comparison

```

data_SJSF_OM_filt <- data_SJSF_OM %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

data_SFTP_OM_filt <- data_SFTP_OM %>%
  filter(discriminant_for == "SF") %>%
  select(OTUID, taxonomy)

SFOM_discrim <- inner_join(data_SJSF_OM_filt, data_SFTP_OM_filt, by = "OTUID")

```

Looking at discriminant taxa for SF 5-15cm: We do this by looking at the OTUs that are discriminant for SF in the SJ v. SF deep comparison and the OTUs that are discriminant for SF in the SF v. TP deep comparison.

#### ITS TP OM comparison

```

data_SJTP_OM_filt <- data_SJTP_OM %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

data_SFTP_OM_filt <- data_SFTP_OM %>%
  filter(discriminant_for == "TP") %>%
  select(OTUID, taxonomy)

TPOM_discrim <- inner_join(data_SJTP_OM_filt, data_SFTP_OM_filt, by = "OTUID")

```

Looking at discriminant taxa for TP 5-15cm: We do this by looking at the OTUs that are discriminant for TP in the SJ v. TP OM comparison and the OTUs that are discriminant for TP in the SF v. TP OM comparison.

```
## Warning in inner_join(data_SJTP_OM_filt, data_SFTP_OM_filt, by = "OTUID"): Detected an unexpected ma
## i Row 17 of 'x' matches multiple rows in 'y'.
## i Row 9 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
## "many-to-many" to silence this warning.
```

## ITS SJ OM comparison

```
data_SJTP_OM_filt <- data_SJTP_OM %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

data_SJSF_OM_filt <- data_SJSF_OM %>%
  filter(discriminant_for == "SJ") %>%
  select(OTUID, taxonomy)

SJOM_discrim <- inner_join(data_SJTP_OM_filt, data_SJSF_OM_filt, by = "OTUID")
```

Looking at discriminant taxa for SJ 5-15cm: We do this by looking at the OTUs that are discriminant for SJ in the SJ v. TP OM comparison and the OTUs that are discriminant for SJ in the SF v. SJ OM comparison.

## Step 6: Save the results to an Excel file

Note: The openxlsx package is used to create an Excel workbook and add sheets for each comparison. The results are written to the respective sheets.

```
# Save results to Excel
wb <- createWorkbook()

# Add sheets
addWorksheet(wb, "SF5_discrim")
addWorksheet(wb, "TP5_discrim")
addWorksheet(wb, "SJ5_discrim")
addWorksheet(wb, "SF15_discrim")
addWorksheet(wb, "TP15_discrim")
addWorksheet(wb, "SJ15_discrim")
addWorksheet(wb, "SFOM_discrim")
addWorksheet(wb, "TPOM_discrim")
addWorksheet(wb, "SJOM_discrim")
```

```
# Write data
writeData(wb, "SF5_discrim", SF5_discrim)
writeData(wb, "TP5_discrim", TP5_discrim)
writeData(wb, "SJ5_discrim", SJ5_discrim)
writeData(wb, "SF15_discrim", SF15_discrim)
writeData(wb, "TP15_discrim", TP15_discrim)
writeData(wb, "SJ15_discrim", SJ15_discrim)
writeData(wb, "SFOM_discrim", SFOM_discrim)
```

```
writeData(wb, "TPOM_discrim", TPOM_discrim)
writeData(wb, "SJOM_discrim", SJOM_discrim)
```

```
saveWorkbook(wb, "/Users/kyasparks/Desktop/CSU 2023-2024/Projects/ASCC/ASCC_July2024_USETHIS_ks/ITS/ITS.
```