

# MitoGraph\_simplified\_width\_filtered.R

mcleland

2022-05-26

```
#  
# Summarize MitoGraph Metrics using Box Plots  
#  
  
# First set working directory to source file location  
# Run with command + option + R  
  
library(tidyverse)  
  
## -- Attaching packages ----- tidyverse 1.3.1 --  
  
## v ggplot2 3.3.5      v purrr  0.3.4  
## v tibble  3.1.6      v dplyr  1.0.7  
## v tidyr   1.2.0      v stringr 1.4.0  
## v readr   2.1.2      v forcats 0.5.1  
  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()  
  
library(reshape2)  
  
##  
## Attaching package: 'reshape2'  
  
## The following object is masked from 'package:tidyr':  
##  
## smiths  
  
library(stringr)  
library(formattable)  
library(igraph)  
  
##  
## Attaching package: 'igraph'  
  
## The following object is masked from 'package:formattable':  
##  
## normalize
```

```
## The following objects are masked from 'package:dplyr':
##
##   as_data_frame, groups, union
```

```
## The following objects are masked from 'package:purrr':
##
##   compose, simplify
```

```
## The following object is masked from 'package:tidyr':
##
##   crossing
```

```
## The following object is masked from 'package:tibble':
##
##   as_data_frame
```

```
## The following objects are masked from 'package:stats':
##
##   decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##   union
```

```
# Import data from log files (*.txt)
```

```
datafolder1 <- "~/mount/home110/mcharwig/20210903_Ntermtrun_repeat4/mitoYFP_DrpIF/combined_drp1_days/Mi
```

```
# Import data
```

```
# with read_csv command below
```

```
data <- read_csv(file = "output-summary_filtered.csv")
```

```
## Rows: 496 Columns: 19
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (4): file, job, isoform, treatment
```

```
## dbl (15): Total_Nodes, Total_Edges, Total_Length_um, Total_Connected_Compone...
```

```
##
```

```
## i Use 'spec()' to retrieve the full column specification for this data.
```

```
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
data2 <- data %>%
```

```
  mutate(job = factor(job, levels = c("WT_RPE_pcDNA",
                                       "Fis1_RPE_pcDNA",
                                       "Fis1_RPE_hFis1_WT",
                                       "Fis1_RPE_hFis1_dN8"), ordered = T)) #RENAME with your sample ord
```

```
data3 <- data2 %>%
```

```
  mutate(treatment = factor(treatment, levels = c("pcDNA",
```

```

                                "hFis1_WT",
                                "hFis1_dN8"), ordered = T)) #RENAME with your sample
data3

```

```

## # A tibble: 496 x 19
##   file          job  isoform treatment Total_Nodes Total_Edges Total_Length_um
##   <chr>         <ord> <chr>   <ord>         <dbl>         <dbl>         <dbl>
## 1 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           175           144           240.
## 2 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           460           429           716.
## 3 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           716           625           942.
## 4 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           451           386           530.
## 5 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           269           237           426.
## 6 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           193           168           322.
## 7 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           302           276           424.
## 8 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           284           294           442.
## 9 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           428           363           538.
## 10 /Users/mclel~ Fis1~ Fis1_R~ pcDNA           563           439           703.
## # ... with 486 more rows, and 12 more variables:
## #   Total_Connected_Components <dbl>, PHI <dbl>, Avg_Edge_Length_um <dbl>,
## #   Total_Edge_Norm_to_Length_um <dbl>, Total_Node_Norm_to_Length_um <dbl>,
## #   Total_Connected_Components_Norm_to_Length_um <dbl>, Free_Ends <dbl>,
## #   three_way_junction <dbl>, four_way_junction <dbl>, Avg_Degree <dbl>,
## #   MitoGraph_Connectivity_Score <dbl>, Average_width_um <dbl>

```

*# Here is a loop that creates 11 graphs where the X and Y variable are listed accordingly.*

```

PlotsToBeMade <- c("job",
                   "treatment",
                   "PHI",
                   "Avg_Edge_Length_um",
                   "Total_Node_Norm_to_Length_um",
                   "Total_Connected_Components_Norm_to_Length_um",
                   "Free_Ends",
                   "three_way_junction",
                   "four_way_junction",
                   "Avg_Degree",
                   "MitoGraph_Connectivity_Score",
                   "Average_width_um")

```

```

AxisLabels <- c("job",
                "treatment",
                "PHI",
                "Avg Edge Length um",
                "Total Node Norm to Length um",
                "Total Connected Components Norm to Length um",
                "Free Ends",
                "three way junction",
                "four way junction",
                "Avg Degree",
                "MitoGraph Connectivity Score",
                "Average width um")

```

```

Titles <- c("job",

```

```

    "treatment",
    "PHI",
    "Avg Edge Length um",
    "Total Node Norm to Length um",
    "Total Connected Components Norm to Length um",
    "Free Ends",
    "Three way junction",
    "Four way junction",
    "Avg Degree",
    "MitoGraph Connectivity Score",
    "Average width um")

# Uncomment to modify specific axis settings
yAxisMinimum <- c(0,
                  0,
                  0,
                  0,
                  0,
                  0,
                  0,
                  0,
                  0,
                  0,
                  0)

#
# yAxisMaximum <- c(1,
#                   1,
#                   150,
#                   150,
#                   150,
#                   150,
#                   150,
#                   1.5,
#                   10000,
#                   20000,
#                   10000,
#                   20000)

for (p in seq(3,length(PlotsToBeMade))) {

  yaxis <- PlotsToBeMade[p]
  xaxis <- PlotsToBeMade[1]
  fill <- PlotsToBeMade[1]

  # Adjust to modify error bar width
  # error <- c(0.03,0.05,
  #            0.03,0.025,
  #            0.02,0.05,
  #            8,0.005)

  plot(ggplot(data=data3,aes_string(x=xaxis, y=yaxis, fill=fill)) +

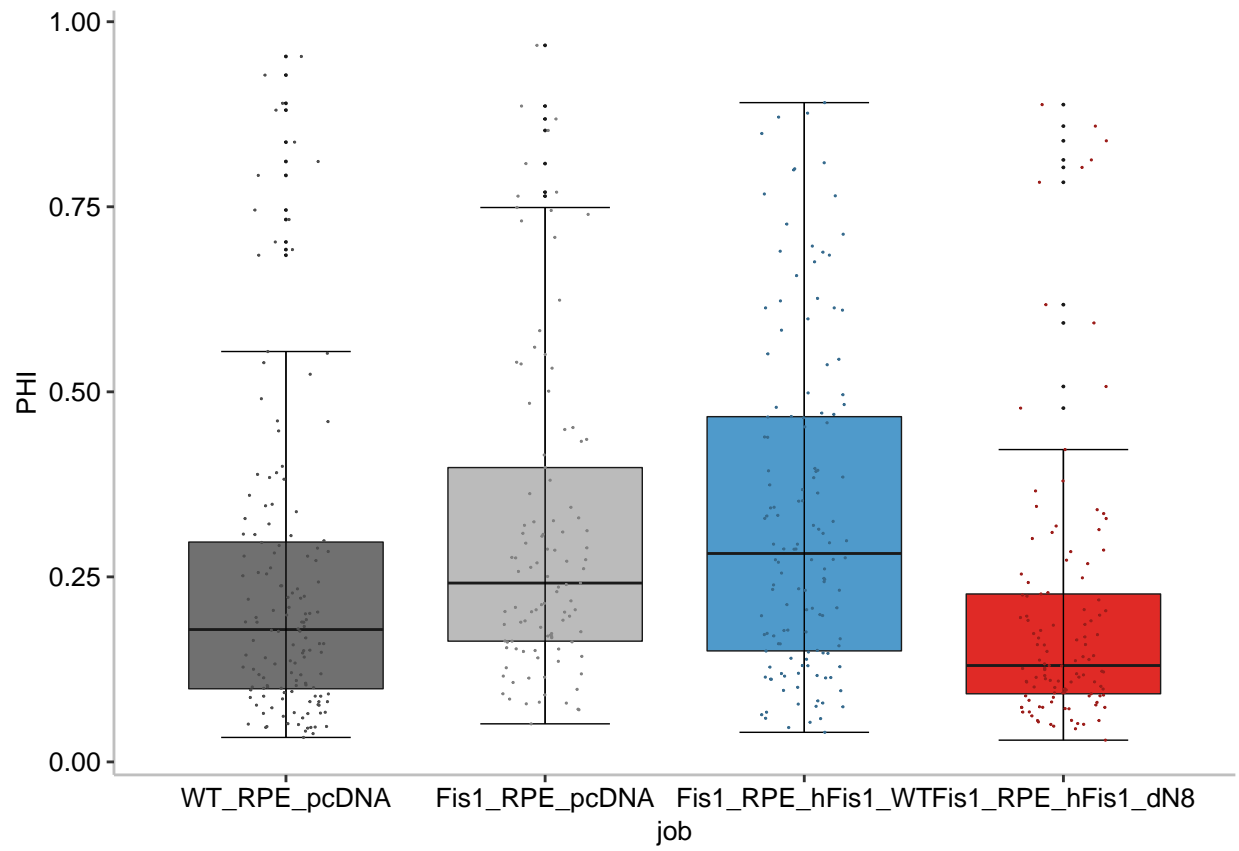
```

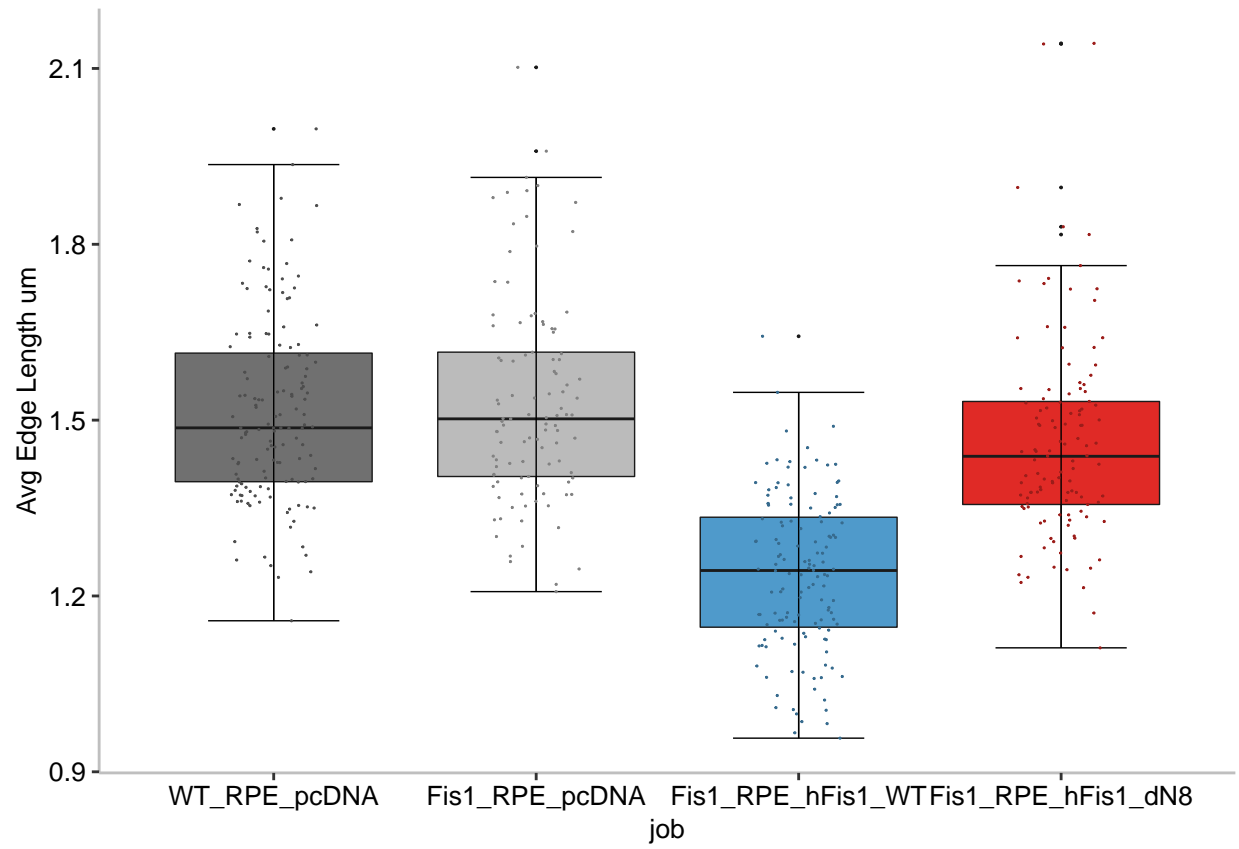
```

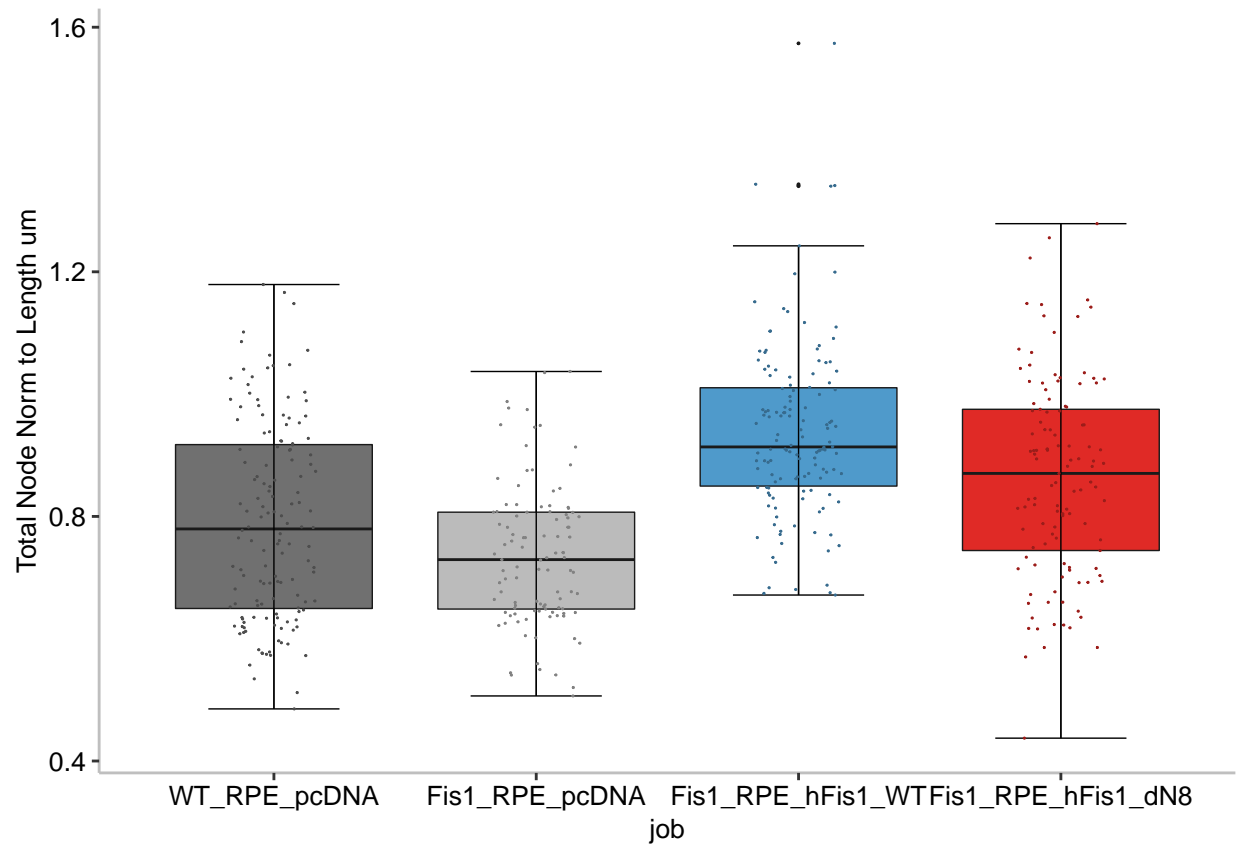
geom_boxplot(outlier.size = 0, colour = "grey10", position = position_dodge (width = 1), size = 0.25) +
stat_boxplot(geom="errorbar", position = position_dodge (width = 1), width = 0.5, size = 0.25) +
scale_fill_manual(values = c("#707070", "#BBBBBB", "#4F9ACB", "#E02A26")) +
geom_point(data=data3, aes_string(colour = fill, x=xaxis, y=yaxis), pch=20, size=0.01, position = position_dodge (width = 1)) +
scale_colour_manual(values=c("#4e4e4e", "#828282", "#376b8e", "#9c1d1a")) +
labs(title = Titles[p],
      x = xaxis,
      y = AxisLabels[p]) +
#ylab(AxisLabels[p]) +
theme_bw() +
theme(axis.text.x = element_text(size = 10, color = "black"),
      axis.title.x = element_text(size = 10, color = "black"),
      axis.text.y = element_text(size = 10, color = "black"),
      axis.title.y = element_text(size = 10, color = "black"),
      plot.title = element_blank(),
      #plot.title = element_text(size = 8, color = "black", hjust = 0.5, face = "bold"),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank(),
      panel.background = element_blank(),
      axis.line.x = element_line(color = "grey75", size = 0.5, linetype = 1),
      axis.line.y = element_line(color = "grey75", size = 0.5, linetype = 1),
      legend.position = "none"
      #legend.title = element_blank(),
      #legend.justification = c(0, 1),
      #legend.position = "right",
      #legend.text = element_text(size = 6, color = "black")
    )
# Add a "+" above and uncomment the below 2 lines to add custom axis scales.
#ylim(yAxisMinimum[p], yAxisMaximum[p])
)

ggsave(paste(datafolder1, "Plot-filtered", yaxis, ".png", sep=""), width = 5.8, height = 8, units = "cm")
ggsave(paste(datafolder1, "Plot-filtered", yaxis, ".eps", sep=""), width = 5.8, height = 8, units = "cm")
}

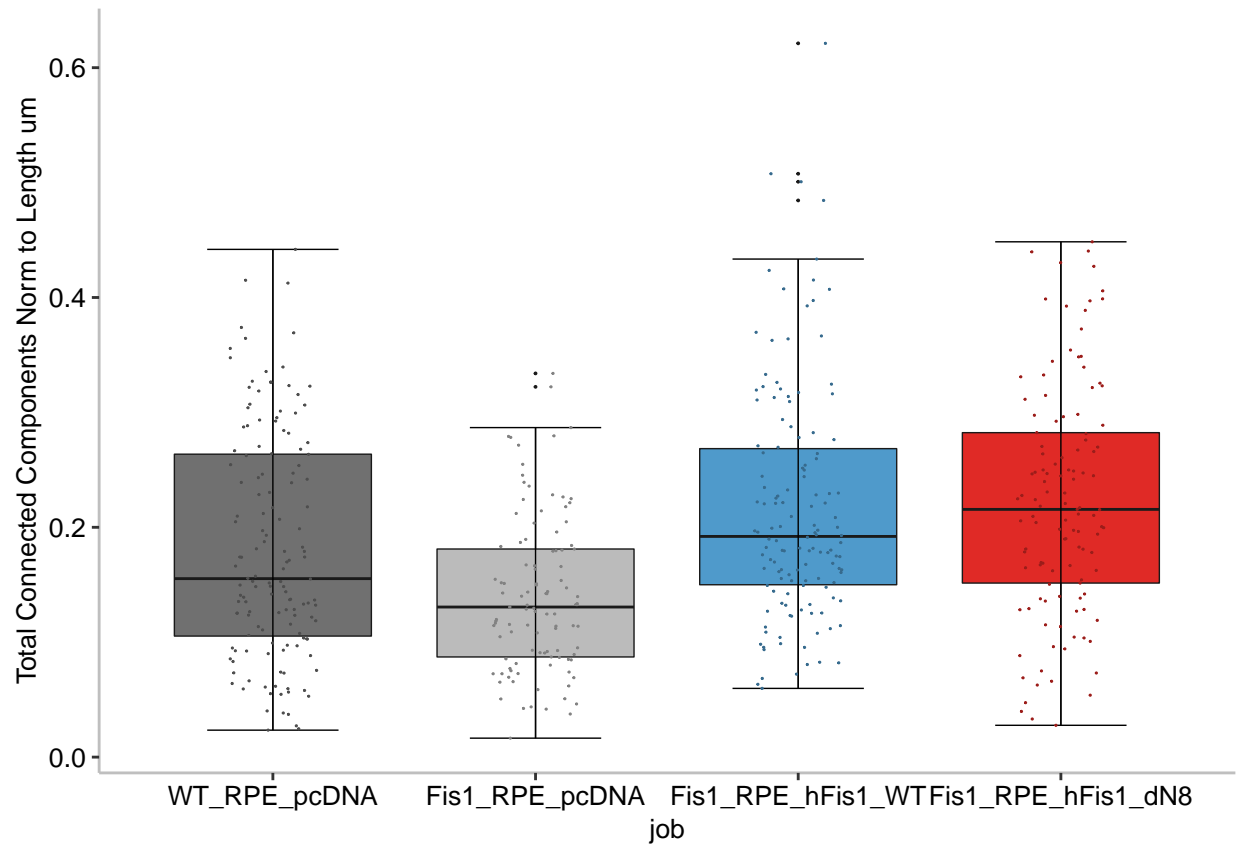
```

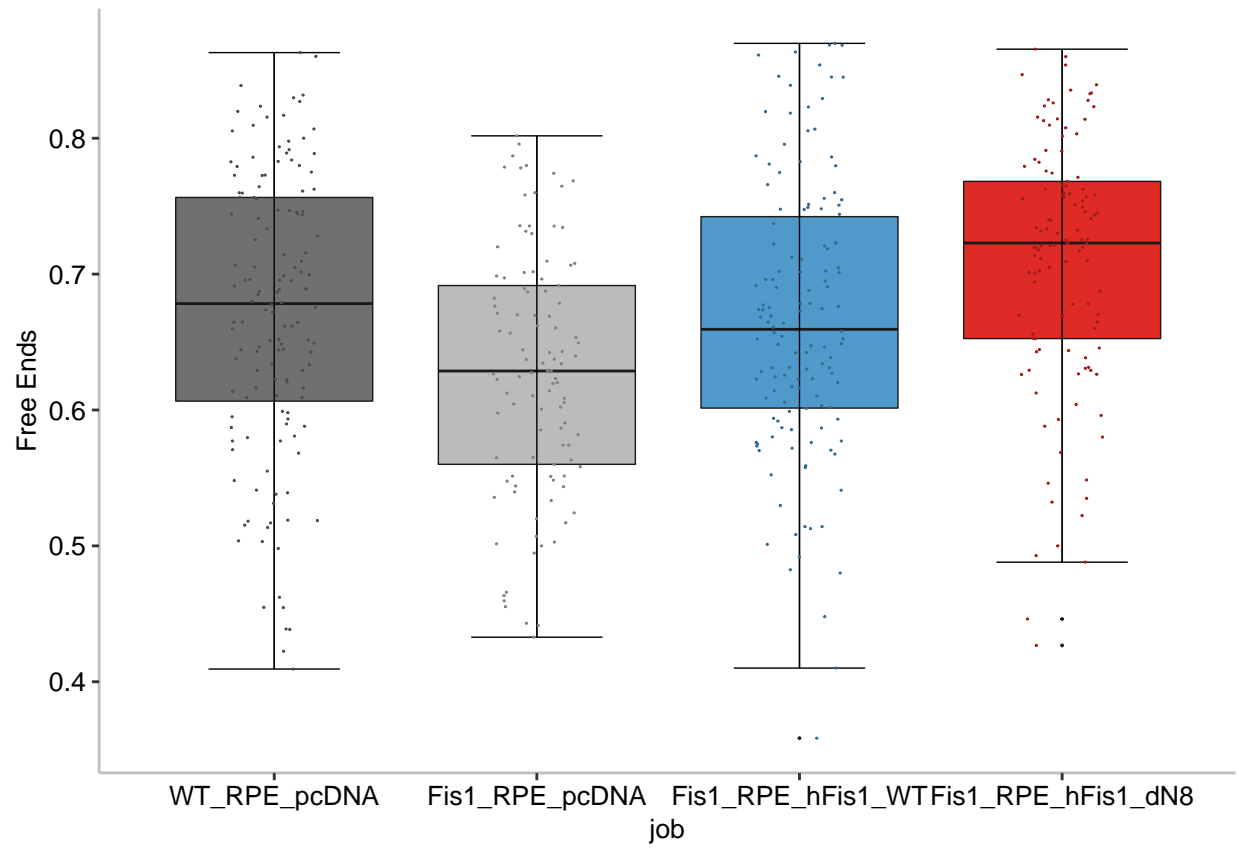


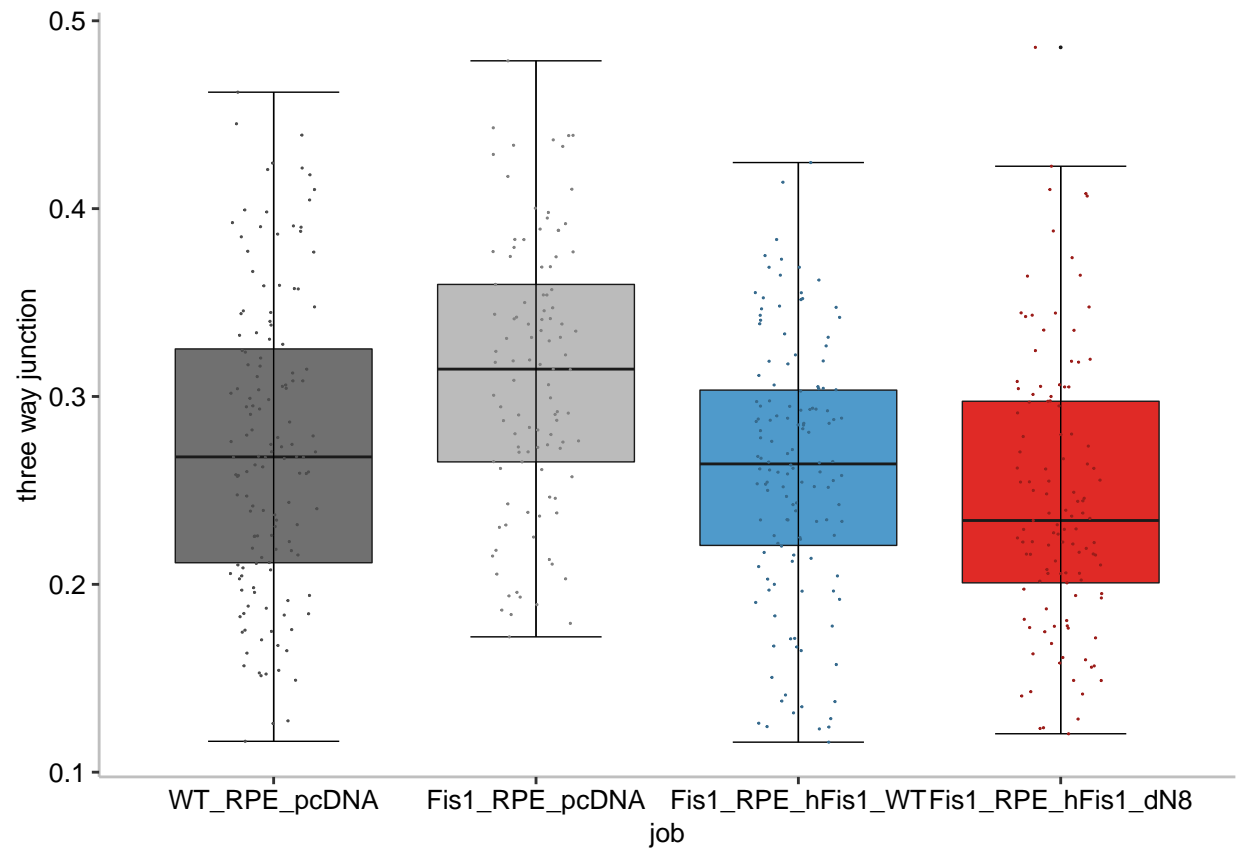


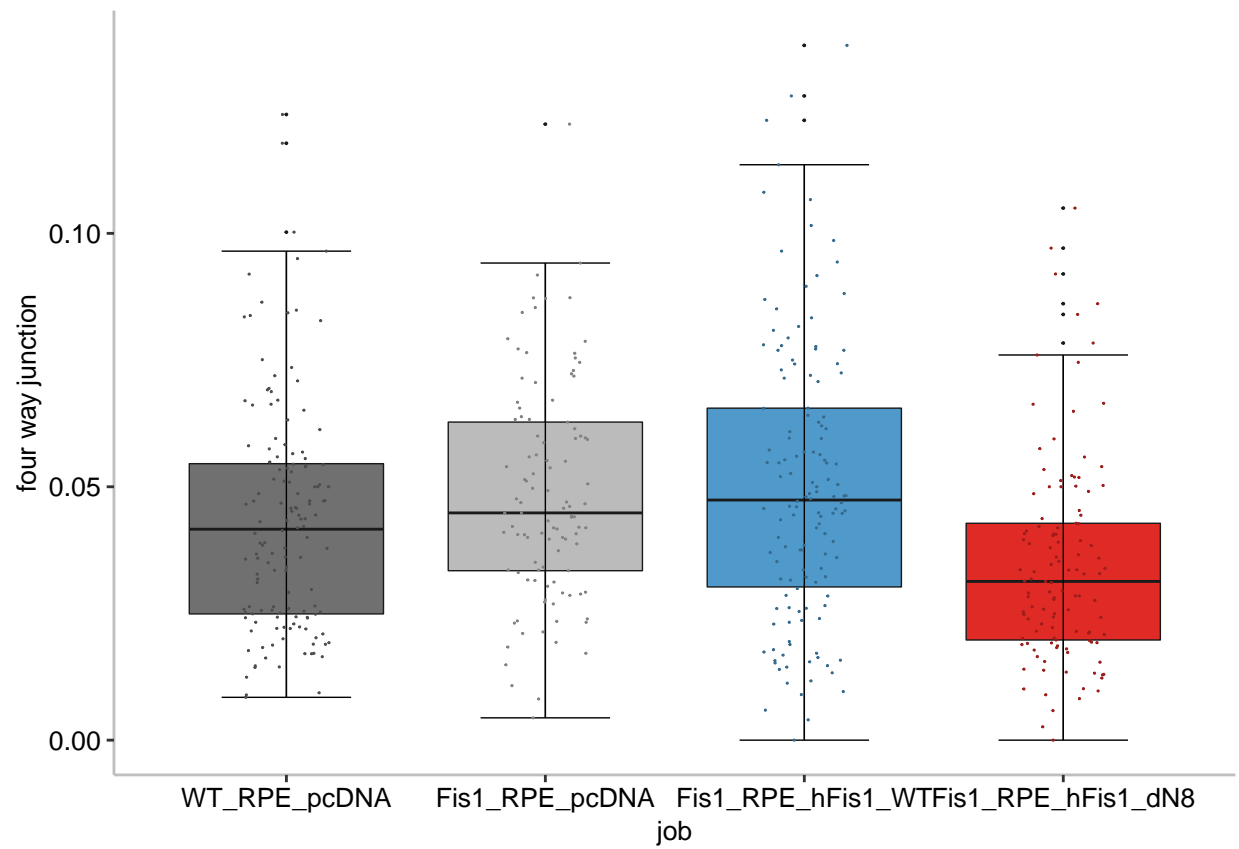


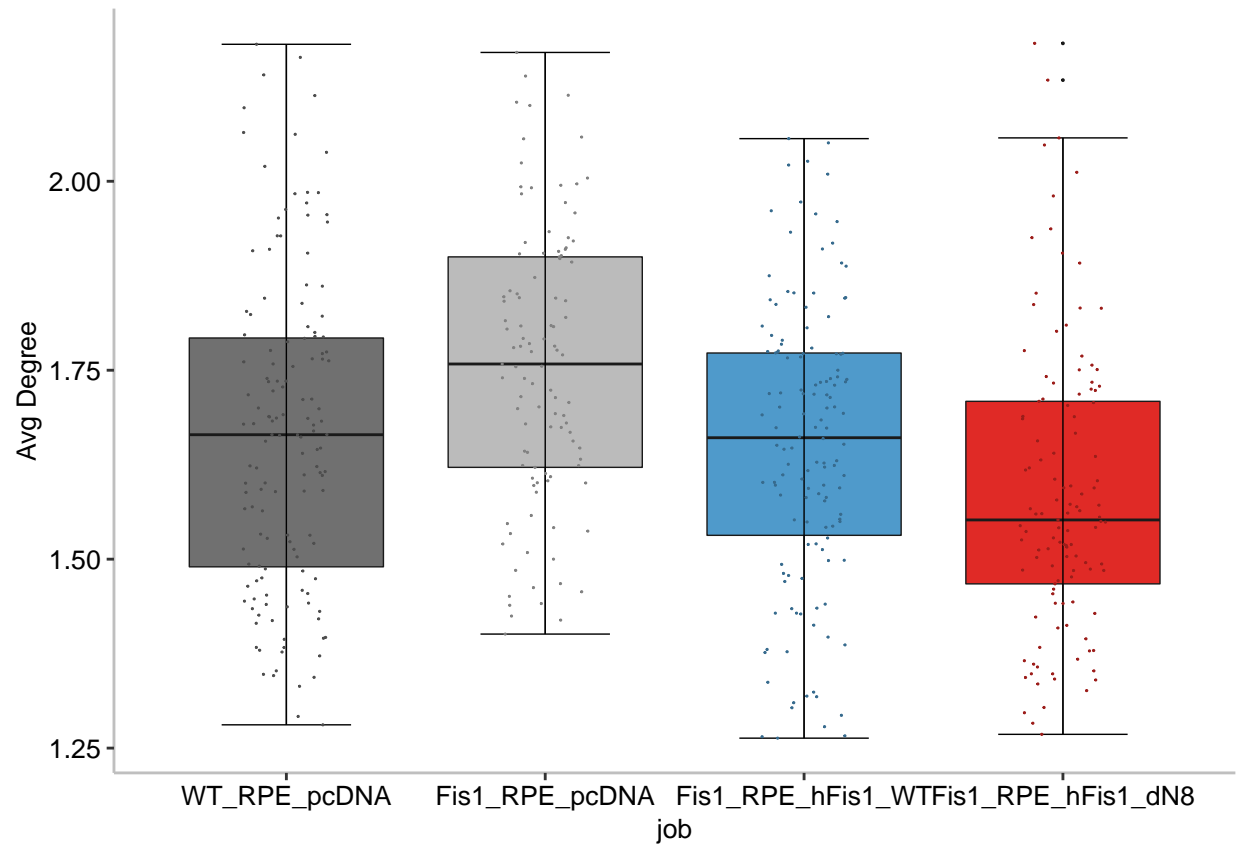


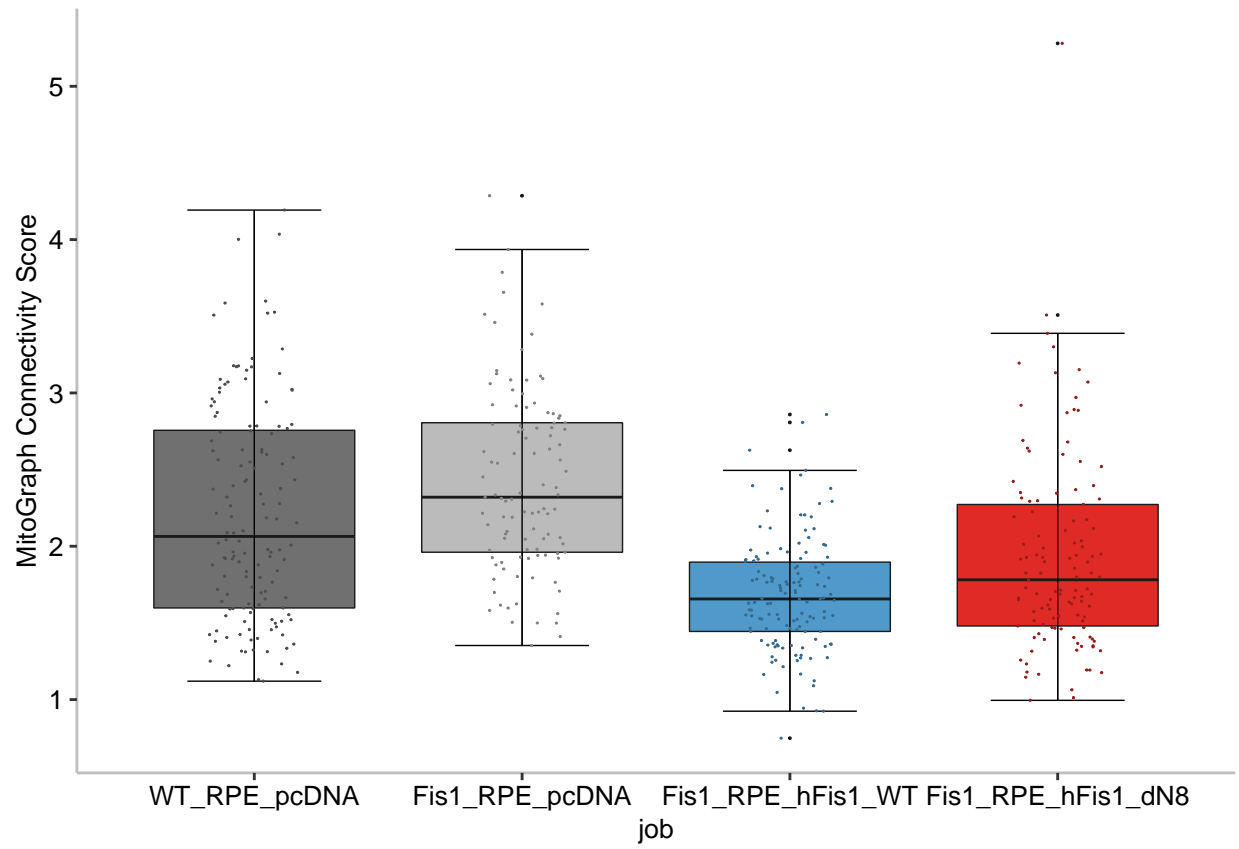


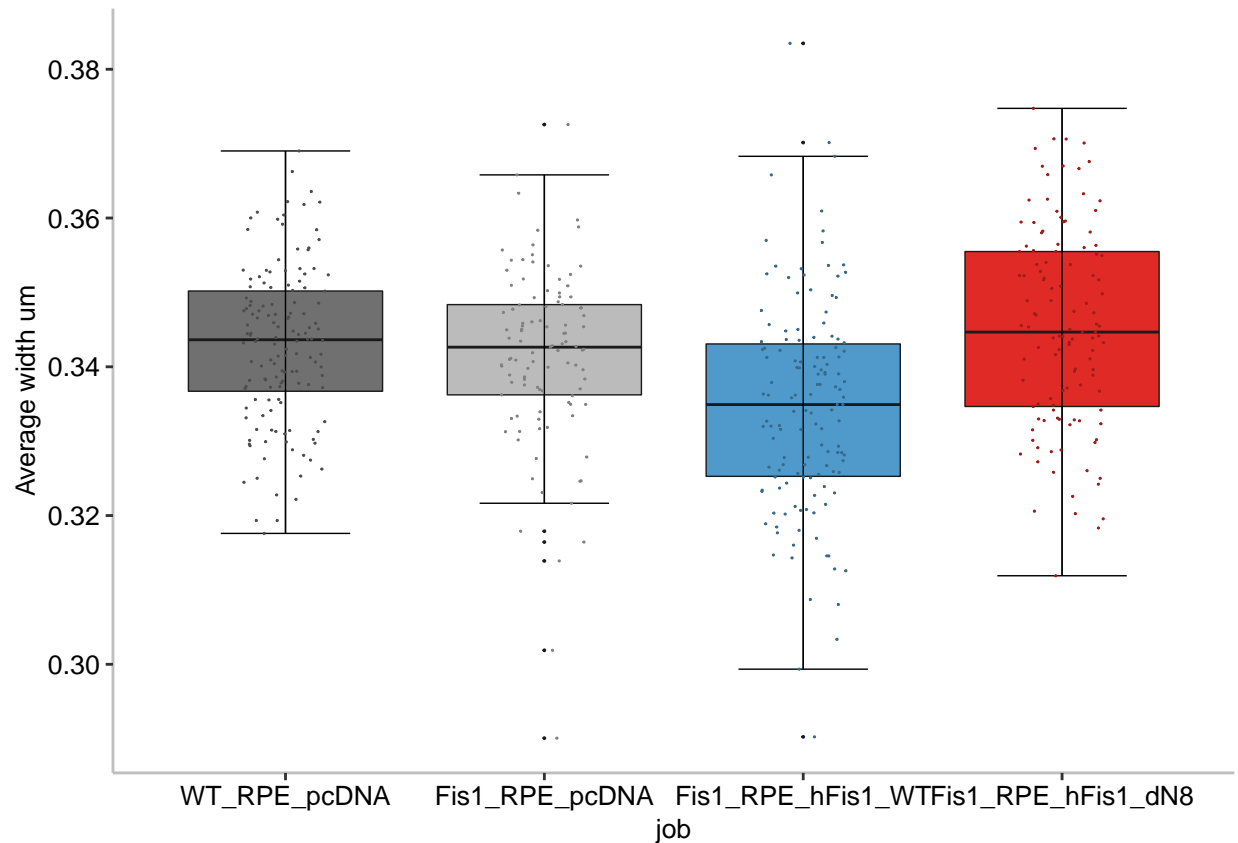












```
#test
```

```
data4 <- read_csv(file = "output-summary_filtered.csv")
```

```
## Rows: 496 Columns: 19
## -- Column specification -----
## Delimiter: ","
## chr (4): file, job, isoform, treatment
## dbl (15): Total_Nodes, Total_Edges, Total_Length_um, Total_Connected_Compone...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
data5 <- data4 %>%
  mutate(job = factor(job, levels = c("WT_RPE_pcDNA",
    "Fis1_RPE_pcDNA",
    "Fis1_RPE_hFis1_WT",
    "Fis1_RPE_hFis1_dN8"), ordered = T)) #RENAME with your sample ord

data5_long <- reshape2::melt(data = data5, id.vars = "job", measure.vars = c("PHI",
  "Total_Edge_Norm_to_Length",
  "Avg_Edge_Length_um",
  "Total_Node_Norm_to_Length",
  "three_way_junction",
```

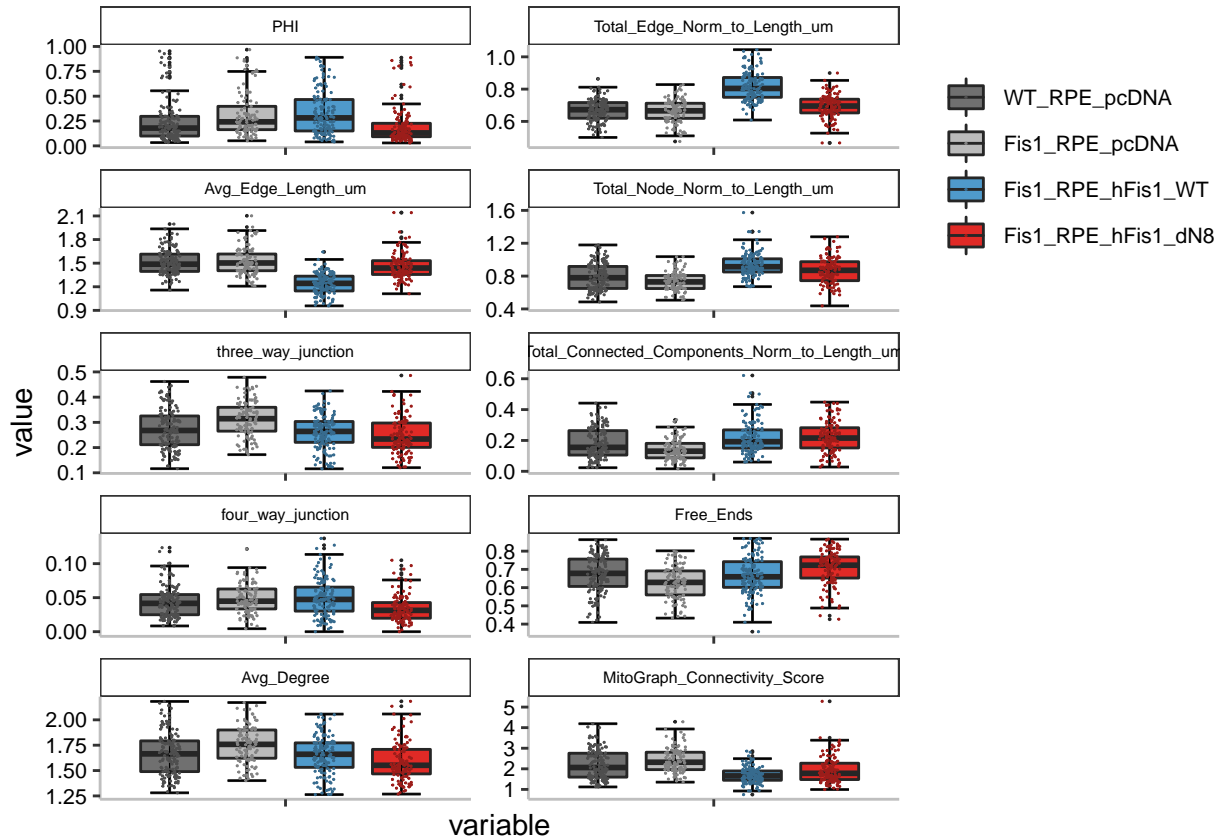
```

    "Total_Connected_Component",
    "four_way_junction",
    "Free_Ends",
    "Avg_Degree",
    "MitoGraph_Connectivity_Score"
  ))

ggplot(data5_long, aes(fill=job, x=variable, y=value)) +
  stat_boxplot(geom = "errorbar", colour = "grey15", width = 0.5, position = position_dodge (width = 1)) +
  geom_boxplot (outlier.size = 0, colour = "grey15", position = position_dodge (width = 1)) +
  stat_boxplot(geom="errorbar", position = position_dodge (width = 1), width = 0.5, size = 0.25) +
  scale_fill_manual(values = c( "#707070", "#BBBBBB", "#4F9ACB", "#E02A26")) +
  facet_wrap(~variable, scales = "free", ncol = 2) +
  geom_point (aes(colour = job, x=variable, y=value), pch=20, position=position_jitterdodge(jitter.width=1)) +
  scale_colour_manual(values=c("#4e4e4e", "#828282", "#376b8e", "#9c1d1a")) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_text(size = 8, color = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.line.x = element_line(color = "grey75", size = 0.5, linetype = 1),
        axis.line.y = element_line(color = "grey75", size = 0.5, linetype = 1),
        #legend.position = "none",
        legend.title = element_blank(),
        legend.justification = c(0, 1),
        legend.position = "right",
        legend.text = element_text(size = 8, color = "black"),
        strip.background = element_rect(fill = "white"),
        strip.text.x = element_text(size = 6, colour = "black")
  )

```





```

ggsave(paste("All_metrics-filtered.eps",sep=""), width = 9, height = 13, units = "cm", dpi = 300)
ggsave(paste("All_metrics-filtered.png",sep=""), width = 9, height = 13, units = "cm", dpi = 300)

#AOV_STATS (ANOVA and TUKEY Post Hoc analysis)

AOV_STATS <- TukeyHSD(aov(data=data, MitoGraph_Connectivity_Score~job))

AOV_Table <- (AOV_STATS)

AOV_Table_summary <- as.data.frame(AOV_Table[1:1])

write.csv(AOV_Table_summary, paste("MitoGraph_CS_AOV_stats.csv",sep=""))

formattable(AOV_Table_summary, Condition.p.adj=formatter("span", style = x~style(color=ifelse(x < 0.05

```

job.diff

job.lwr

job.upr

job.p.adj

Fis1\_RPE\_hFis1\_WT-Fis1\_RPE\_hFis1\_dN8

-0.2516827

-0.44550351

-0.05786193  
4.854263e-03  
Fis1\_RPE\_pcDNA-Fis1\_RPE\_hFis1\_dN8  
0.4916388  
0.28571918  
0.69755842  
9.459815e-09  
WT\_RPE\_pcDNA-Fis1\_RPE\_hFis1\_dN8  
0.2807320  
0.08885694  
0.47260698  
1.041522e-03  
Fis1\_RPE\_pcDNA-Fis1\_RPE\_hFis1\_WT  
0.7433215  
0.54367547  
0.94296757  
9.724466e-11  
WT\_RPE\_pcDNA-Fis1\_RPE\_hFis1\_WT  
0.5324147  
0.34728856  
0.71754079  
1.004776e-10  
WT\_RPE\_pcDNA-Fis1\_RPE\_pcDNA  
-0.2109068  
-0.40866445  
-0.01314924  
3.137726e-02