

```

---
title: "SexAndAgingImmunity"
output: html_document
---

``{r}
rm(list=ls())

...

#SETUP LIBRARY LOADS
``{r}
library(knitr)
library(gtools)
library(data.table)
library(rmeta)
library(forestplot)
library(plyr)
# load my library

opts_chunk$set(echo=FALSE, include=TRUE, warning=T, message=F, out.width= 600)
opts_knit$set(stop_on_error=2)

...

#INPUT AND DATA PREPARATION
``{r}
age_data = readRDS("/labs/uNet_dt_with_preliminaryAge.rds")
age_data = age_data[analyticalGroup== "adult_organ_rna", ]
age_data1 = age_data
###age_data1 = age_data[(age_data$gene_tag=="ligand" & age_data$level1=="cytokine"),]
age_data1$sex[age_data1$sex == "male"] = 0
age_data1$sex[age_data1$sex=="female"] = 1

#getting rid of technical replicate groups that are NA or blank
age_data2 = age_data1
age_data2 = age_data2[!is.na(age_data2$technical_replicate_group)]
age_data2 = age_data2[!age_data2$technical_replicate_group==" "]
age_data2 = age_data2[order(age_data2$technical_replicate_group)]

age_data3 = age_data2
age_data3$technical_replicate_group=gsub("(GTEx-\\w{4,5}-\\w{4,5})-.*", "\\1",
age_data3$technical_replicate_group)
#_____

bbc <- read.csv("/labs/biobank_collection_20170613_014541.csv", stringsAsFactors = FALSE)
bbc1 = data.table(bbc)

#converting categorical hardyScale values to numerical
attach(bbc1)
bbc1$hardyScale[bbc1$hardyScale=="Ventilator case"]=0
bbc1$hardyScale[hardyScale=="Slow death"]=1

```

```
bbc1$hardyScale[hardyScale=="Intermediate death"]=2
bbc1$hardyScale[hardyScale=="Fast death - natural causes"]=3
bbc1$hardyScale[hardyScale=="Fast death - violent"]=4
```

```
bbc1$autolysisScore[autolysisScore=="None"] = 0
bbc1$autolysisScore[autolysisScore=="Mild"] = 1
bbc1$autolysisScore[autolysisScore=="Moderate"] = 2
bbc1$autolysisScore[autolysisScore=="Severe"] = 3
```

```
bbc1 = bbc1[order(sampleId)]
bbc1 = bbc1[order(bbc1$sampleId)]
bbc2 = bbc1[bbc1$materialType=="RNA:Total RNA"]
bbc2 = bbc2[!duplicated(bbc2$tissueSampleId)]
# _____
```

```
sample_att = read.csv("/labs/GTEX_Data_V6_Annotations_SampleAttributesDS (1).csv")
sample_att1 = sample_att
sample_att1$SAMPID=gsub("(GTEX-\\w{4,5}-\\w{4,5})-.*", "\\1", sample_att1$SAMPID)
sample_att1 = sample_att1[, c("SAMPID", "SMTSISCH")]
sample_att2 = sample_att1[!is.na(sample_att1$SMTSISCH),]
sample_att3 = sample_att2[!duplicated(sample_att2$SAMPID),]
# _____
```

```
#merge
age_data4 = merge(age_data3, bbc2, by.x = c("technical_replicate_group"), by.y = c("tissueSampleId"))
age_data4$sex.y[age_data4$sex.y == "male"] = 0
age_data4$sex.y[age_data4$sex.y=="female"] = 1
age_data4[age_data4$sex.x == age_data4$sex.y]
age_data4$autolysisScore = as.numeric(age_data4$autolysisScore)
age_data4$hardyScale=as.numeric(age_data4$hardyScale)
age_data4 = merge(age_data4, sample_att3, by.x = c("technical_replicate_group"), by.y=c("SAMPID"))
...`
```

#STEP 1: RUNNING ADJUSTMENT ON SEX, AGE, ISCHEMIC TIME, HARDYSCALE, AUTOLYSIS SCORE

```
```{r}
age_data5 = ddply(age_data4,c("hgnc_symbol", "organism_part"),
  function(x) {

    #sex coefficient
    w1 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,1],

    error = function(err){
      print(paste("MY_ERROR: ",err))
      return(NA)},
    finally={print("hi")})

    #age coefficient
    w2 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,1],

    error = function(err){
      print(paste("MY_ERROR: ",err))
      return(NA)},
```

```

        finally={print("hi")})

#hardyScale coefficient
w3 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,1],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#autolysisScore coefficient
w4 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,1],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#ischemic time coefficient
w10 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,1],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#intercept
w5 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[1,1],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#sex t value
w6 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,3],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#age t value
w7 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,3],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#hardy scale t value
w8 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,3],

```

```

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#autolysis score t value
w9 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,3],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#ischemic time t value
w11 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,3],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#sex error
w12 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,2],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#age error
w13 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,2],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#hardy scale error
w14 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,2],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

#autolysis score error
w15 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,2],

        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)},
        finally={print("hi")})

```

```

#ischemic time error
w16 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,2],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#sex p value
w17 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,4],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#age p value
w18 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,4],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#hardy scale p value
w19 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,4],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#autolysis score p value
w20 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,4],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#ischemic time p value
w21 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,4],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

```

```

        data.frame(sex.coef = w1, age.coef = w2, hardy.coef = w3, auto.coef = w4, isch.coef = w10, intercept =
w5, sex.t = w6, age.t = w7, hardy.t = w8, auto.t = w9, isch.t = w11,
        sex.e = w12, age.e = w13, hardy.e = w14, auto.e = w15, isch.e = w16, sex.p = w17, age.p = w18,
hardy.p = w19, auto.p = w20, isch.p = w21)
    })

```

```

age_data6 = age_data5
age_data6 = age_data6[!is.na(age_data6$sex.coef),]
age_data6 = age_data6[ order(age_data6$sex.t),]
#age_data6$q.value= p.adjust(age_data6$sex.p)

```

```

#calculating adjusted pvalue

```

```

age_data7 = merge(age_data6, age_data4, by = c("hgnc_symbol", "organism_part"))
age_data8 = age_data7

```

```

temp = c("gene_tag", "individual", "sample", "disease", "amount", "mass", "materialType",

```

```

"originalMaterialType", "rin", "pathologyNotes", "sampleId", "volume", "ageBracket", "tissueSiteDetail", "synonym", "c
oncentration",
"subjectId")

```

```

age_data8 = age_data8[!(names(age_data8) %in% temp)]
age_data8$adjustedRPKM = with(age_data8, meanRPKM - (age.coef*age) - (hardy.coef*hardyScale) -
(auto.coef*autolysisScore) - (isch.coef*SMTSISCH))
saveRDS(age_data5, "/labs/201707_Unetdt_Adjustment/output/age_data5")
saveRDS(age_data8, "/labs/201707_Unetdt_Adjustment/output/age_data8")
...

```

```

#STEP 1: CREATING HEATMAP FOR SEX, AGE, ISCHEMIC TIME

```

```

```{r}

```

```

#SEX

```

```

sex = age_data6
sex = dcast(sex, organism_part~hgnc_symbol,value.var = "sex.t")
rownames(sex) = sex$organism_part
sex[,1]=NULL
sex = data.matrix(sex)
rbColor = colorRampPalette(c("Blue", "White", "Red"))
heatmap.2(sex, col=rbColor, main = "Sex Difference by Organ by Gene", sepwidth=c(0,0), tracecol=NULL, scale=
"none", cexCol = .3)

```

```

#AGE

```

```

age = age_data6
age = dcast(age, organism_part~hgnc_symbol,value.var = "age.t")
rownames(age) = age$organism_part
age[,1]=NULL
age = data.matrix(age)
rbColor = colorRampPalette(c("Blue", "White", "Red"))
heatmap.2(age, col=rbColor, main = "Age Difference by Organ by Gene", sepwidth=c(0,0), tracecol=NULL, scale=
"none", cexCol = .3)

```

```

#ISCHEMIC TIME

```

```

isch = age_data6
isch = dcast(isch, organism_part~hgnc_symbol,value.var = "isch.t")

```

```

isch = data.table(isch)
isch = isch[-nrow(isch)]
rownames(isch) = isch$organism_part
isch[,1]=NULL

```

```

isch = data.matrix(isch)
rbColor = colorRampPalette(c("Blue", "White", "Red"))
heatmap.2(isch, col=rbColor, main = "Ischemic Time Difference by Organ by Gene", sepwidth=c(0,0),
tracecol=NULL, scale= "none", cexCol = .3)

```

```

...

```

```

#AGE HEATMAP ONLY CYTOKINE

```

```

```{r}
heatmap_age_cytokine = age_data6_cytokine
heatmap_age_cytokine = dcast(heatmap_age_cytokine, organism_part~hgnc_symbol,value.var = "age.t")
rownames(heatmap_age_cytokine) = heatmap_age_cytokine$organism_part
heatmap_age_cytokine[,1]=NULL
heatmap_age_cytokine = data.matrix(heatmap_age_cytokine)
#rbColor = colorRampPalette(c("Blue", "White", "Red"))
heatmap.bar(data_mx= heatmap_age_cytokine, impute_byRow_TF = T, col_midRepeat_n = 2, fontsize_col = 7.2,
fontsize_row = 13, fontsize = 8, legend = TRUE, col_mid_txt = "white", border_color = "white", col_neg_txt =
"darkblue", col_pos_txt = "firebrick3")

```

```

...

```

```

#SEX HEATMAP ONLY CYTOKINE

```

```

```{r}
heatmap_sex_cytokine = age_data6_cytokine
heatmap_sex_cytokine = dcast(heatmap_sex_cytokine, organism_part~hgnc_symbol,value.var = "sex.t")
rownames(heatmap_sex_cytokine) = heatmap_sex_cytokine$organism_part
heatmap_sex_cytokine[,1]=NULL
heatmap_sex_cytokine = data.matrix(heatmap_sex_cytokine)
#rbColor = colorRampPalette(c("Blue", "White", "Red"))
heatmap.bar(data_mx= heatmap_sex_cytokine, impute_byRow_TF = T, col_midRepeat_n = 2, fontsize_col = 7.2,
fontsize_row = 13, fontsize = 8, legend = TRUE, col_mid_txt = "white", border_color = "white", col_neg_txt =
"darkblue", col_pos_txt = "firebrick3")

```

```

...

```

```

#UNADJUSTED LINEAR CORRELATION

```

```

```{r}
unadj_age = age_data3[,N:=N, by=(organism_part, hgnc_symbol, sex)]
unadj_age=unadj_age[!(unadj_age$N<20),]
#unadj_age$sex[unadj_age$sex==0] = "male"
#unadj_age$sex[unadj_age$sex==1] = "female"

unadjusted_age_data = ddply(unadj_age,c("hgnc_symbol", "organism_part"),

```

```

function(x) {
  #print(paste(x[x$sex=="female"],]$meanRPKM))
  # w = tryCatch(t.test(x[x$sex == "female", ]$meanRPKM,x[x$sex == "male", ]$meanRPKM)$p.value,
  # error = function(err){
  #   print(paste("MY_ERROR: ",err))
  # print((x[x$sex == "female", ]$meanRPKM))
  #print( paste(x$hgnc_symbol[1], x$organism_part[1]))
  # return(NA)},
  # finally={print(paste(x$hgnc_symbol[1], x$organism_part[1]))})

  #w1 = tryCatch(t.test(x[x$sex == "female", ]$meanRPKM,x[x$sex == "male", ]$meanRPKM)$statistic,
  #   error = function(err){
  #     print(paste("MY_ERROR: ",err))
  #     print((x[x$sex == "female", ]$meanRPKM))
  #     print( paste(x$hgnc_symbol[1], x$organism_part[1]))
  #     return(NA)},
  #     finally={print(paste(x$hgnc_symbol[1], x$organism_part[1]))})

  #w2 = tryCatch(t.test(x[x$sex == "female", ]$meanRPKM,x[x$sex == "male", ]$meanRPKM)$p.value,
  #   error = function(err){
  #     print(paste("MY_ERROR: ",err))
  #     print((x[x$sex == "female", ]$meanRPKM))
  #     print( paste(x$hgnc_symbol[1], x$organism_part[1]))
  #     return(NA)},
  #     finally={print(paste(x$hgnc_symbol[1], x$organism_part[1]))})

  #sex coef
  w3 = tryCatch( coef(summary(lm(meanRPKM~sex, data = x)))[2,1],
    error = function(err){
      print(paste("MY_ERROR: ",err))
      return(NA)},
    finally={print("hi")})

  #sex error
  w4 = tryCatch( coef(summary(lm(meanRPKM~sex, data = x)))[2,2],
    error = function(err){
      print(paste("MY_ERROR: ",err))
      return(NA)},
    finally={print("hi")})

  #age coef
  w5 = tryCatch( coef(summary(lm(meanRPKM~age, data = x)))[2,1],
    error = function(err){
      print(paste("MY_ERROR: ",err))
      return(NA)},
    finally={print("hi")})

  #age error
  w6 = tryCatch( coef(summary(lm(meanRPKM~age, data = x)))[2,2],
    error = function(err){
      print(paste("MY_ERROR: ",err))
      return(NA)},
    finally={print("hi")})

  data.frame(sex.coef = w3, sex.e = w4, age.coef = w5, age.e = w6)

```



```
}}
```

```
unadjusted_age_data1 = unadjusted_age_data
unadjusted_age_data1 = unadjusted_age_data1[!is.na(unadjusted_age_data1$sex.coef),]
unadjusted_age_data1 = unadjusted_age_data1[!is.na(unadjusted_age_data1$age.coef),]
#age_data6= age_data6[ order(age_data6$sex.t),]
#age_data6$q.value= p.adjust(age_data6$sex.p)

#calculating adjusted pvalue
unadjusted_age_data1 = merge(unadjusted_age_data1, age_data, by = c("hgnc_symbol", "organism_part"))

saveRDS(unadjusted_age_data1, "/labs/201707_Unetdt_Adjustment/output/unadjusted_age_data1")
```

```
...
```

#STEP 2: BARPLOT, DIFFERENCE BETWEEN MALES AND FEMALES BY ORGAN. ADDING CYTOKINE AND CHEMOKINE ADJUSTED RPKM

Question: What is the sex difference of adjustedRPKM levels in both cytokine and chemokine by organ?  
Observations: breast is the most different (higher expression in females). coronary artery more highly expressed in males

```
```{r}
step2_byorgan = age_data8
step2_byorgan = data.table(step2_byorgan)
step2_byorgan = step2_byorgan[step2_byorgan$level1=="cytokine"|step2_byorgan$level1=="chemokine"]
step2_byorgan = step2_byorgan[!is.na(step2_byorgan$adjustedRPKM)]
step2_byorgan_m = step2_byorgan[step2_byorgan$sex.x == 0, mean(adjustedRPKM), by = organism_part]
step2_byorgan_f = step2_byorgan[step2_byorgan$sex.x == 1, mean(adjustedRPKM), by = organism_part]
step2_byorgan_diff = merge(step2_byorgan_m, step2_byorgan_f, by = c("organism_part"))
step2_byorgan_diff$diff = step2_byorgan_diff$V1.x - step2_byorgan_diff$V1.y
step2_byorgan_diff = step2_byorgan_diff[order(step2_byorgan_diff$diff)]

attach(step2_byorgan_diff)
step2_byorgan_diff = step2_byorgan_diff[order(step2_byorgan_diff$diff)]
barplot = ggplot(step2_byorgan_diff, aes(x = organism_part, y = diff, fill = diff)) + geom_bar(stat = "identity")
barplot + coord_flip() + scale_x_discrete(limits = step2_byorgan_diff$organism_part) +
  xlab("") + ylab("KINDEX Difference") + scale_fill_gradient2(low = "darkred", mid = "snow3", high = "darkblue") +
  theme(plot.title = element_text(hjust = 0.5)) + theme(legend.position="none") +
  theme(text=element_text(size=13))
```

```
...
```

#STEP 2: BARPLOT AGE. ADDING CYTOKINE AND CHEMOKINE ADJUSTED RPKM

```
```{r}
step2_byorgan = age_data8
step2_byorgan = data.table(step2_byorgan)
step2_byorgan = step2_byorgan[step2_byorgan$level1=="cytokine"|step2_byorgan$level1=="chemokine"]
```

```
step2_byorgan = step2_byorgan[!is.na(step2_byorgan$adjustedRPKM)]
...
```

#### #META INPUT

```
```{r}
meta_sex = age_data8[,c("hgnc_symbol", "organism_part", "sex.coef", "sex.e", "adjustedRPKM", "age")]
meta_age = age_data8[,c("hgnc_symbol", "organism_part", "age.coef", "age.e", "adjustedRPKM", "age")]
meta_isch = age_data8[,c("hgnc_symbol", "organism_part", "isch.coef", "isch.e", "adjustedRPKM", "age")]
meta_isch = meta_isch[!is.na(meta_isch$isch.coef),]
...

```

#### #CREATE META INPUT TABLE FOR UNADJUSTED VALUE SEX

```
```{r}
meta_sex_unadjusted =
unadjusted_age_data1[,c("hgnc_symbol", "organism_part", "sex.coef", "sex.e", "meanRPKM")]
meta_age_unadjusted =
unadjusted_age_data1[,c("hgnc_symbol", "organism_part", "age.coef", "age.e", "meanRPKM")]
#meta_isch_unadjusted =
unadjusted_age_data1[,c("hgnc_symbol", "organism_part", "isch.coef", "isch.e", "meanRPKM")]
...

```

#### #RUNNING THE METAANALYSIS FOR ADJUSTED FUNCTION

```
```{r}
meta = function(var){
  var = data.table(var)
  colnames(var)[3]="coef"
  colnames(var)[4]="se"
  meta = ddply(var, c("hgnc_symbol"), function(x){
    x = x[!is.na(x$adjustedRPKM),]
    x = data.table(x)
    organ = x[,median(adjustedRPKM), by = c("organism_part")]
    organ = organ[order(-organ$V1)]
    val = organ[1]$V1 - log(20)
    organ1 = organ[organ$V1 >= val]
    x = x[x$organism_part %in% organ1$organism_part]

    x = x[!duplicated(x$organism_part)]

    w1 = meta.summaries(x$coef, x$se, method="random")$summary
    w2 = meta.summaries(x$coef, x$se, method="random")$se.summary
    w3 = meta.summaries(x$coef, x$se, method="random")$test[1]
  })
}

```

```

w4 = meta.summaries(x$coef, x$se, method="random")$test[2]
w5 = meta.summaries(x$coef, x$se, method="random")$het[1]
w6 = meta.summaries(x$coef, x$se, method="random")$het[2]
w7 = meta.summaries(x$coef, x$se, method="random")$het[3]
conf = 1.959963986
#upper confidence
w8 = w1 + (conf*w2)
#lower confidence
w9 = w1 - (conf*w2)

w10 = dim(organ)[1]
w11 = dim(organ1)[1]

data.table(ans = w1, se.sum = w2, t = w3, p = w4, het1 = w5, het2 = w6, het3 = w7, upper = w8, lower = w9,
numOfOrgans_before = w10, numOfOrgans_after = w11)
})

meta = meta[order(meta$p, -meta$ans),]
return(meta)
}
```

```

#### #RUNNING THE METAANALYSIS FOR UNADJUSTED

```

```{r}
meta_unadj = function(var){
  var = data.table(var)
  colnames(var)[3]="coef"
  colnames(var)[4]="se"
  meta = ddply(var, c("hgnc_symbol"), function(x){
    x = x[!is.na(x$meanRPKM),]
    x = data.table(x)
    organ = x[,median(meanRPKM), by = c("organism_part")]
    organ = organ[order(-organ$V1)]
    val = organ[1]$V1 - log(20)
    organ1 = organ[organ$V1 >= val]
    x = x[x$organism_part %in% organ1$organism_part]

    x = x[!duplicated(x$organism_part)]

    w1 = meta.summaries(x$coef, x$se, method="random")$summary
    w2 = meta.summaries(x$coef, x$se, method="random")$se.summary
    w3 = meta.summaries(x$coef, x$se, method="random")$test[1]
    w4 = meta.summaries(x$coef, x$se, method="random")$test[2]
    w5 = meta.summaries(x$coef, x$se, method="random")$het[1]
    w6 = meta.summaries(x$coef, x$se, method="random")$het[2]
    w7 = meta.summaries(x$coef, x$se, method="random")$het[3]
    conf = 1.959963986
    #upper confidence
    w8 = w1 + (conf*w2)
    #lower confidence
    w9 = w1 - (conf*w2)

    w10 = dim(organ)[1]

```

```

w11 = dim(organ1)[1]

data.table(ans = w1, se.sum = w2, t = w3, p = w4, het1 = w5, het2 = w6, het3 = w7, upper = w8, lower = w9,
numOfOrgans_before = w10, numOfOrgans_after = w11)
})

meta = meta[order(meta$p, -meta$ans),]
meta$q_unadj = p.adjust(meta$p)
return(meta)
}
...

```

#### #SAVING META-ANALYSIS DATA

```

```{r}
sex = meta(meta_sex)
sex$q_adj = p.adjust(sex$p)
sex_unadj = meta_unadj(meta_sex_unadjusted)
sex_unadj = sex_unadj[,c("hgnc_symbol", "q_unadj")]
sex = merge(sex, sex_unadj, by = c("hgnc_symbol"))
sex = sex[order(sex$q_adj),]
saveRDS(sex, "/labs/201707_MetaAnalysis/meta_sex")

```

```

age = meta(meta_age)
age$q_adj = p.adjust(age$p)
age_unadj = meta_unadj(meta_age_unadjusted)
age_unadj = age_unadj[,c("hgnc_symbol", "q_unadj")]
age = merge(age, age_unadj, by = c("hgnc_symbol"))
age = age[order(age$q_adj),]
saveRDS(age, "/labs/201707_MetaAnalysis/meta_age")

```

```

isch = meta(meta_isch)
isch$q_adj = p.adjust(isch$p)
isch_unadj = meta_unadj(meta_isch_unadjusted)
isch_unadj = isch_unadj[,c("hgnc_symbol", "q_unadj")]
isch = merge(isch, isch_unadj, by = c("hgnc_symbol"))
isch = isch[order(isch$q_adj),]
saveRDS(isch, "/labs/201707_MetaAnalysis/meta_isch")
...

```

#### #FORREST PLOT

```

```{r}
sex_forest = meta(meta_sex)
forestplot(labeltext = sex_forest$hgnc_symbol, mean = sex_forest$ans, upper = sex_forest$upper, lower =
sex_forest$lower, is.summary = FALSE, boxsize=0.75, txt_gp = fpTxtGp(cex = .1))
...

```

#### #FOREST PLOT FUNCTION FOR VERY SIGNIFICANT GENES

```

```{r}
gene_specific_forest_plot = function(gene, x, type){
  x = data.table(x)

```

```

colnames(x)[3]="coef"
colnames(x)[4]="se"
x = x[hgnc_symbol=="PDCD1LG2",]

x = x[!is.na(x$adjustedRPKM),]
x = data.table(x)
organ = x[,median(adjustedRPKM), by = c("organism_part")]
organ = organ[order(-organ$V1)]
val = organ[1]$V1 - log(20)
organ1 = organ[organ$V1 >= val]
x = x[x$organism_part %in% organ1$organism_part]
x = x[order(-x$coef)]
x = x[!duplicated(x$organism_part)]
print(unique(x$organism_part))

meta = meta.summaries(x$coef, x$se, names = x$organism_part, method="random")
meta$test[2]
plot(meta, summary = TRUE, summlabel = "Summary", xlab = "", ylab = (""), main = "PDCD1LG2",
col=meta.colors(box="darkred",line="black", summary = "darkred"))

# meta = ddply(x, c("organism_part"), function(y){
#   w1 = meta.summaries(y$coef, y$se, method="random")$summary
#   w2 = meta.summaries(y$coef, y$se, method="random")$se.summary
#   data.table(upper = w1 + w2, ans = w1, lower = w1 - w2)
# })

#forestplot(labeltext = meta$organism_part, mean = meta$ans, upper = meta$upper, lower = meta$lower)
}

```

Question: is HLA regulated by sex hormones?

Assumption: HLA is differentially expressed between males and females, with males having the higher expression

Step 1: Run the meta-analysis including age

```

```{r}
meta_with_age = function(var){
  var = data.table(var)
  colnames(var)[3]="coef"
  colnames(var)[4]="se"

  var$age_group[var$age < 40] = 0
  var$age_group[var$age > 40 & var$age < 60] = 1
  var$age_group[var$age > 60] = 2

```

```

meta = ddply(var, c("hgnc_symbol", "age_group"), function(x){
  x = x[!is.na(x$adjustedRPKM),]
  x = data.table(x)
  organ = x[,median(adjustedRPKM), by = c("organism_part")]
  organ = organ[order(-organ$V1)]
  val = organ[1]$V1 - log(20)
  organ1 = organ[organ$V1 >= val]
  x = x[x$organism_part %in% organ1$organism_part]

  x = x[!duplicated(x$organism_part)]

  w1 = meta.summaries(x$coef, x$se, method="random")$summary
  w2 = meta.summaries(x$coef, x$se, method="random")$se.summary
  w3 = meta.summaries(x$coef, x$se, method="random")$test[1]
  w4 = meta.summaries(x$coef, x$se, method="random")$test[2]
  w5 = meta.summaries(x$coef, x$se, method="random")$het[1]
  w6 = meta.summaries(x$coef, x$se, method="random")$het[2]
  w7 = meta.summaries(x$coef, x$se, method="random")$het[3]
  conf = 1.959963986
  #upper confidence
  w8 = w1 + (conf*w2)
  #lower confidence
  w9 = w1 - (conf*w2)

  w10 = dim(organ)[1]
  w11 = dim(organ1)[1]

  data.table(ans = w1, se.sum = w2, t = w3, p = w4, het1 = w5, het2 = w6, het3 = w7, upper = w8, lower = w9,
numOfOrgans_before = w10, numOfOrgans_after = w11)
})

meta = meta[order(meta$p, -meta$ans),]
return(meta)
}
```



```

```{r}
hormone = meta_with_age(meta_sex)
hormone = data.table(hormone)
hormone_hlag = hormone[hgnc_symbol=="HLA-G"]
young = test[age_group==0]
old = test[age_group==2]
diff_of_mean = (young$ans - old$ans)
standard_dev = sqrt((young$se.sum*young$se.sum) + (old$se.sum*old$se.sum))
tval = diff_of_mean / standard_dev
```

```{r}
ttest_sex_byage = function(var){
hormone_hlag = hormone[hgnc_symbol==var]
young = hormone_hlag[age_group==0]

```


```

```

old = hormone_hlag[age_group==2]
diff_of_mean = young$ans - old$ans
standard_dev = sqrt((young$se.sum*young$se.sum) + (old$se.sum*old$se.sum))
tval = diff_of_mean / standard_dev
print(tval)
}
...

```

#UNADJUSTED FOREST PLOT FOR VERY SIGNIFICANT GENES

```

```{r}
gene_specific_forest_plot_unadj = function(gene, x, type){
  x = data.table(x)
  colnames(x)[3]="coef"
  colnames(x)[4]="se"
  x = x[hgnc_symbol==gene]

  x = x[!is.na(x$meanRPKM),]
  organ = x[,median(meanRPKM), by = c("organism_part")]
  organ = organ[order(-organ$V1)]
  val = organ[1]$V1 - log(100)
  organ1 = organ[organ$V1 >= val]
  x = x[x$organism_part %in% organ1$organism_part]
  x = x[order(-x$coef)]
  x = x[!duplicated(x$organism_part)]
  print(unique(x$organism_part))

  meta = meta.summaries(x$coef, x$se, names = x$organism_part, method="random")
  meta$test[2]
  #plot(meta, summary = TRUE, summlabel = "Summary")
  plot(meta, summary = TRUE, summlabel = "Summary", xlab = paste(type, "Difference"), main = paste(gene, type,
"Difference"))

  # meta = ddply(x, c("organism_part"), function(y){
  # w1 = meta.summaries(y$coef, y$se, method="random")$summary
  # w2 = meta.summaries(y$coef, y$se, method="random")$se.summary
  # data.table(upper = w1 + w2, ans = w1, lower = w1 - w2)
  # })

  #forestplot(labeltext = meta$organism_part, mean = meta$ans, upper = meta$upper, lower = meta$lower)
}
...

```

#STEP 1: RUNNING ADJUSTMENT ON SEX, AGE, ISCHEMIC TIME, HARDYSCALE, AUTOLYSIS SCORE WITH ONLY CYTOKINE

```

``{r}
age_data4_cytokine = age_data4[age_data4$level1=="cytokine",]
age_data5_cytokine = ddpdy(age_data4_cytokine,c("hgnc_symbol", "organism_part"),
  function(x) {

    #sex coefficient
    w1 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,1],
      error = function(err){
        print(paste("MY_ERROR: ",err))
        return(NA)},
      finally={print("hi")})

    #age coefficient
    w2 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,1],
      error = function(err){
        print(paste("MY_ERROR: ",err))
        return(NA)},
      finally={print("hi")})

    #hardyScale coefficient
    w3 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,1],
      error = function(err){
        print(paste("MY_ERROR: ",err))
        return(NA)},
      finally={print("hi")})

    #autolysisScore coefficient
    w4 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,1],
      error = function(err){
        print(paste("MY_ERROR: ",err))
        return(NA)},
      finally={print("hi")})

    #ischemic time coefficient
    w10 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,1],
      error = function(err){
        print(paste("MY_ERROR: ",err))
        return(NA)},
      finally={print("hi")})

    #intercept
    w5 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[1,1],
      error = function(err){
        print(paste("MY_ERROR: ",err))
        return(NA)},
      finally={print("hi")})
  }

```



```

#sex t value
w6 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,3],
              error = function(err){
                print(paste("MY_ERROR: ",err))
                return(NA)},
              finally={print("hi")})

#age t value
w7 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,3],
              error = function(err){
                print(paste("MY_ERROR: ",err))
                return(NA)},
              finally={print("hi")})

#hardy scale t value
w8 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,3],
              error = function(err){
                print(paste("MY_ERROR: ",err))
                return(NA)},
              finally={print("hi")})

#autolysis score t value
w9 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,3],
              error = function(err){
                print(paste("MY_ERROR: ",err))
                return(NA)},
              finally={print("hi")})

#ischemic time t value
w11 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,3],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#sex error
w12 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,2],
               error = function(err){
                 print(paste("MY_ERROR: ",err))
                 return(NA)},
               finally={print("hi")})

#age error
w13 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,2],
               error = function(err){
                 print(paste("MY_ERROR: ",err))

```

```

        return(NA)),
        finally={print("hi")})

#hardy scale error
w14 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,2],
        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)),
        finally={print("hi")})

#autolysis score error
w15 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[5,2],
        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)),
        finally={print("hi")})

#ischemic time error
w16 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[6,2],
        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)),
        finally={print("hi")})

#sex p value
w17 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[2,4],
        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)),
        finally={print("hi")})

#age p value
w18 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[3,4],
        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)),
        finally={print("hi")})

#hardy scale p value
w19 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x))))[4,4],
        error = function(err){
            print(paste("MY_ERROR: ",err))
            return(NA)),
        finally={print("hi")})

#autolysis score p value

```

```

w20 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x)))[5,4],
                error = function(err){
                  print(paste("MY_ERROR: ",err))
                  return(NA)},
                finally={print("hi")})

#ischemic time p value
w21 = tryCatch( coef(summary(lm(meanRPKM~sex.x+age+hardyScale+autolysisScore+SMTSISCH, data =
x)))[6,4],
                error = function(err){
                  print(paste("MY_ERROR: ",err))
                  return(NA)},
                finally={print("hi")})

data.frame(sex.coef = w1, age.coef = w2, hardy.coef = w3, auto.coef = w4, isch.coef = w10, intercept =
w5, sex.t = w6, age.t = w7, hardy.t = w8, auto.t = w9, isch.t = w11,
           sex.e = w12, age.e = w13, hardy.e = w14, auto.e = w15, isch.e = w16, sex.p = w17, age.p = w18,
           hardy.p = w19, auto.p = w20, isch.p = w21)
})

age_data6_cytokine = age_data5_cytokine
age_data6_cytokine = age_data6_cytokine[!is.na(age_data6_cytokine$sex.coef),]
age_data6_cytokine = age_data6_cytokine[!is.na(age_data6_cytokine$age.coef),]
saveRDS(age_data6_cytokine, "/labs/201707_Unetdt_Adjustment/output/age_data6_only_cytokine")
```

#MAKING EXTRA FIGURES
Gtex Figure --> Number of Samples per Organ
```{r}
test = uNet_dt
test = test[!is.na(test$technical_replicate_group)]
attach(test)
test$technical_replicate_group=gsub("(GTEx-\\w{4,5}-\\w{4,5})-.*", "\\1", test$technical_replicate_group)
test = test[!duplicated(test$technical_replicate_group)]
test = test[,N, by = organism_part][order(N)]
test = test[N > 50]

ggplot(test, aes(x = organism_part, y = N)) + geom_bar(stat = "identity", fill = "gray68") +
theme(axis.text.x=element_text(angle=55,hjust=1,vjust=1)) + ylab("N of Samples") + xlab("") +
theme(text=element_text(size=15)) + theme(legend.position="none") + scale_x_discrete(limits =
test$organism_part)
```

```