#Load Packages

library(geomorph)

LM.Data <- read.table ("combined\_fgf8\_P0\_lm.txt",header=T,row.names=1)

Class.Data <- read.table ("combined\_fgf8P0\_geno&cs.txt",header=T,row.names=1)

Class <- Class.Data.ord

#find mean shape by group, average group and control means, group center all samples

ref<-mshape(Y.gpa$coords)

droplevels(Class)

gpsp0<-droplevels(Class)

lsmeans<-arrayspecs((rowsum(two.d.array(Y.gpa$coords), gps)/as.vector(table(gps))),76,3) #gp means

controlmeans<-lsmeans [,,c(4,8)]#get means of control groups, (floxwt = 5, wt = 9)

avemean <-apply (controlmeans, c(1,2), mean) #average means

meandif <- controlmeans - as.vector(avemean) #subtract the average control from each control groupm

crectdif <- meandif[,,1]

neodif <- meandif[,,2]

Y.neo<-((Y.gpa$coords[,,1:96]) - as.vector(neodif))

Y.crect <- ((Y.gpa$coords [,,97:169]) -as.vector (crectdif))

library(abind)

Y.dta <- abind (Y.neo, Y.crect)

#PCA plot (tangent space)

PCAnoreg <- plotTangentSpace(Y.dta, warpgrids =TRUE, groups = Class)

P#generate allometric resdiuals

Allom <- plotAllometry(Y.dta ~ Y.gpa$Csize,~Class,method="CAC", iter=999,RRPP=TRUE, verbose = TRUE)

plotAllometry(Y.dta ~ Y.gpa$Csize,~Class, method="PredLine", iter=15)

Allom$ProcDist.lm

Y.resid <- arrayspecs(Allom$resid.shape, 76, 3)

procD.lm(Y.dta~ Y.gpa$Csize\*Class)

advanced.procD.lm(Y.dta ~ Y.gpa$Csize\*Class,

 Y.dta ~ Y.gpa$Csize+Class,

 groups=~ Class,

 iter = 999)

#set-up (Homogenetity of slopes test)

procD.lm(Y.dta ~ log(Y.gpa$Csize)\*Class, iter = 999, RRPP=T)

# Compare slopes

advanced.procD.lm(Y.dta ~ log(Y.gpa$Csize)\*Class,

 ~ log(Y.gpa$Csize) + Class,

 slope = ~ log(Y.gpa$Csize),

 groups = ~ Class,

 iter = 999)

#calculate trace

p0gdf <- geomorph.data.frame(Y.dta, csize = Y.gpa$Csize, Genotype=Class)

morphol.disparity(Y.dta~1, ~Genotype, data=p0gdf)

morphol.disparity(Y.resid, Class, iter = 999)

mshape(Y.dta)

plotAllometry()

write.table((two.d.array(Y.dta)), file = "P0alignednoregression.txt")

Y <- two.d.array(Y.dta)

fit1 <- lm(Y~ 1) # effects = intercept

fit2 <- lm(Y ~ Class) # effects = intercept + species

advanced.procD.lm(fit1, fit2)

morphol.disparity(Y.dta, Class)

#make warps

surface <- read.ply("fgf8crect\_83\_7.ply", addNormals = FALSE)

#points <- as.matrix(read.table("rg\_fgf8crect\_p1\_83\_7\_fjs\_Landmarks.landmarkAscii"))

mesh <- surface

findMeanSpec(Y.dta)

mean.mesh <- read.ply("fgf8crect\_P1\_89\_8.ply", addNormals = FALSE)

mean.points <- LM.Array.ord[,,186]

mean.refwarp <- warpRefMesh(mean.mesh, mean.points, ref)

PC1\_Min <- plotRefToTarget(ref,regroup.verbose$pc.shapes$PC1min, mesh = mean.refwarp, method="surface")

rgl.snapshot("PC1min.png", fmt="png", top=TRUE)

PC1\_Min\_dots <- plotRefToTarget(ref,regroup.verbose$pc.shapes$PC1min, method="vector")

rgl.snapshot("PC1mindots.png", fmt="png", top=TRUE)

PC1\_Max <-plotRefToTarget(ref,regroup.verbose$pc.shapes$PC1max, mesh = mean.refwarp, method="surface")

rgl.snapshot("PC1max.png", fmt="png", top=TRUE)

PC1\_Max\_dots <-plotRefToTarget(ref,regroup.verbose$pc.shapes$PC1max, method="vector")

rgl.snapshot("PC1maxdots.png", fmt="png", top=TRUE)

PC2\_Min <- plotRefToTarget(ref,regroup.verbose$pc.shapes$PC2min, mesh = mean.refwarp, method="surface")

rgl.snapshot("PC2min.png", fmt="png", top=TRUE)

PC2\_Min\_dots <- plotRefToTarget(ref,regroup.verbose$pc.shapes$PC2min, method="vector")

rgl.snapshot("PC2mindots.png", fmt="png", top=TRUE)

PC2\_Max <-plotRefToTarget(ref,regroup.verbose$pc.shapes$PC2max, mesh = mean.refwarp, method="surface")

rgl.snapshot("PC2max.png", fmt="png", top=TRUE)

PC2\_Max\_dots <-plotRefToTarget(ref,regroup.verbose$pc.shapes$PC2max, method="vector")

rgl.snapshot("PC2maxdots.png", fmt="png", top=TRUE)

Class <- factor (Class, levels = c("WT","floxwt","neowt","DeltaWt","floxnull","floxwt:crect","floxnull:crect","neo","DeltaNeo" ))

library(ggplot2)

dfp0 <- as.data.frame(regroup$pc.scores, row.names = 1)

p <- ggplot(dfp0, aes(df[,1], df[,2]))

(p + geom\_point(aes(fill = Class),colour="black", pch=21,size=I(5), alpha=I(.8))

 + labs(x= "PC1", y= "PC2")

 + guides(fill =guide\_legend(title="Genotypes")))

fgf8lev <- c(1,1,0.929175984,0.629659273,0.485682854,0.463525824,0.258340585,0.162956662,0.12315139)

fact<- c("WT"= 1.0393736,"floxwt"=1.0971024,"neowt"=0.9601877,"DeltaWt"=0.7024247,"floxnull"=0.7095715,"floxwt:crect"=0.5675576,"floxnull:crect"=0.2018279,"neo"=0.1948377,"DeltaNeo"=0.1485272)

fgf8p0 <- gpsp0

fgf8p0 <- fact[gpsp0]

fgf8p0 <- as.numeric(fgf8p0)

df2p0<- cbind(dfp0,fgf8p0)

fit3 <- lm(dfp0[,1]~fgf8p0, data=dfp0)

fit4 <- lm(df2p0[,1]~fgf8p0, data=df2p0)

q <- qplot(df[,1], df[,2], data=df2, xlab= "PC1", ylab= "PC2")

q + geom\_point(aes(colour = fgf8),size=5) + scale\_colour\_gradient2(low="red", mid="violet", high="blue", midpoint=.2, limits=c(0.1, 1.2))

q <- ggplot(df2, aes(df[,1], df[,2]))

(q + geom\_point(aes(fill = I(fgf8)),colour="black", pch=21,size=I(5), alpha=I(.8))

+ scale\_fill\_gradientn(colours=rainbow(4))

+ labs(x= "PC1", y= "PC2")

+ guides(fill =guide\_colorbar(title="Fgf8 level")))

r <- ggplot(data=df2,aes(fgf8, df[,1]))

(r + geom\_point(aes(fill = Class),colour="black", pch=21,size=I(5), alpha=I(.8)) + geom\_smooth(method = "lm", formula = y ~ splines::bs(x, 3))

+ labs(x= "Fgf8 level", y= "PC1")

+ guides(fill =guide\_legend(title="Genotypes")))

rna <- read.csv("fgf8qRTPCR.csv", col.names = c("Genotype", "sample", "Fgf8"))

Genotype <- factor (rna$Genotype, levels = c("WT", "floxwt","NeoWt", "DeltaWt", "floxnull", "floxwt:crect", "floxnull:crect","Neo", "DeltaNeo"))

library("plyr")

means <- ddply(rna, "Genotype", summarise, mean = mean(Fgf8), sd=sd(Fgf8))

##> means (data as of May 2 2016 - 2 outliers removed)

Genotype mean sd

1 DeltaNeo 0.1485272 0.1005975

2 DeltaWt 0.7024247 0.3703243

3 floxnull 0.7095715 0.6316764

4 floxnull:crect 0.2018279 0.1684020

5 floxwt 1.0971024 0.4974643

6 floxwt:crect 0.5675576 0.3443150

7 Neo 0.1948377 0.1179042

8 NeoWt 0.9601877 0.2544536

9 WT 1.0393736 0.2834377

#Barplot

ggplot(means, aes(x = Genotype, y = mean)) + geom\_bar(stat = "identity")

mean <- read.table("fgf8rtpcrmean.txt", sep = ",", col.names = c("Genotype", "mean", "sd"))

#Gigger plot

means$Genotype = factor(means$Genotype, levels = c("WT","floxwt","NeoWt","floxnull","DeltaWt","floxwt:crect","floxnull:crect","Neo","DeltaNeo" ))

rna$Genotype = factor(rna$Genotype, levels = c("WT","floxwt","NeoWt","floxnull","DeltaWt","floxwt:crect","floxnull:crect","Neo","DeltaNeo" ))

ggplot(rna, aes(x=Genotype, y=Fgf8, colour = Genotype)) +geom\_boxplot(outlier.shape = NA)+ geom\_jitter()+ theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1)) + labs (y = "Relative Fgf8 expression")

mean$Genotype <- factor(mean$Genotype, levels = c("WT","floxwt","NeoWt","floxnull","DeltaWt","floxwt:crect","Neo","floxnull:crect","DeltaNeo" ))

p <- ggplot(mean, aes(x= Genotype, y= mean, fill = Genotype))

p + geom\_bar(stat="identity") + geom\_errorbar(limits, width=0.2) + theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1)) + labs (y = "Relative Fgf8 expression")

limits <- aes(x = Genotype, ymax= mean+sd, ymin = mean-sd, colour = Genotype)

gdfp0 <- geomorph.data.frame(Y.dta, csize = Y.gpa$Csize, Genotype = gps)

traceP0 <- morphol.disparity(Y.dta~1, ~gps, data = gdfp0)

tracedf <- as.data.frame(trace$Procrustes.var, row.names = c("WT","floxwt","NeoWt","DeltaWt","floxnull","floxwt:crect","Neo","floxnull:crect","DeltaNeo" ))

tracedfP0 <- as.data.frame(traceP0$Procrustes.var, row.names = c("DeltaNeo","DeltaWt","floxnull","floxnull:crect","floxwt", "floxwt:crect","Neo","NeoWt","WT" ))

combinedtrace <- merge (tracedfP0,tracedf,by=0)

 ans <- merge(tracedf,tracedfP0,by="row.names",all=T)

 row.names(ans) <- ans[,"Row.names"]

 ans[,!names(ans) %in% "Row.names"]

trace\_fgf8 <- merge(ans, means, by.x="Row.names", by.y = "Genotype")

colnames(trace\_fgf8) <- c("Genotype", "Trace\_e10.5","Trace\_P0","fgf8","sd")

trace\_fgf8$Genotype <- factor (trace\_fgf8$Genotype, levels = c("WT","floxwt","NeoWt","floxnull","DeltaWt","floxwt:crect","floxnull:crect","Neo","DeltaNeo" ))

library(reshape)

tmelt <- melt(trace\_fgf8[,-5], id= c("Genotype","fgf8"))

fit.trace1 <- lm(tmelt$value ~ poly(tmelt$fgf8,3))

summary(fit.trace1)

fit.trace2 <- lm(tmelt$value ~ tmelt$fgf8)

summary(fit.trace2)

v <- ggplot(trace\_fgf8, aes(x= fgf8, y= Trace\_P0))

(v + geom\_point(aes(colour = Genotype), size=3.5)

 + geom\_point(aes(x= fgf8, y= Trace\_e10.5, colour = Genotype), shape = 18, size = 4.5,show.legend=FALSE)

 + coord\_cartesian(ylim = c(0, 0.0300))

 + labs(x= "Fgf8 expression", y= "Trace")

 + scale\_shape\_discrete(name ="Age",

 breaks=c("Trace\_P0", "Trace\_e10.5"),

 labels=c("P0", "E10.5")))

s <- ggplot(tmelt, aes(x= fgf8, y= value))

(s + geom\_point(aes(colour = Genotype, shape=variable),size=5, alpha=I(.8))

 + coord\_cartesian(ylim = c(0, 0.0300))

 + labs(x= "Fgf8 expression", y= "Trace")

 + scale\_shape\_discrete(name ="Age",

 breaks=c("Trace\_P0", "Trace\_e10.5"),

 labels=c("P0", "E10.5")))

 + geom\_smooth(method = "lm", formula = y ~ splines::bs(x, 2)))

##pcaplots from rnaseq data

PCA <- read.table("RNAseq\_ggplot.txt")

PCA$Category <- factor(PCA$Category, levels = c("WT","Neo\_het","Delta\_het","Neo\_homo","Delta\_neo"))

u <- ggplot(PCA,aes(PCA[,1], PCA[,2]))

u + geom\_point(aes(fill = Category), colour="black", pch=21,size=I(5), alpha=I(.8)) + xlab("PC1: 44% variance") +

 ylab("PC2: 16% variance") + labs(title="")

#get PC1 mean by group:

PC1\_RNA <- ddply(PCA, "Category", summarise, PC1 = mean(PC1), sd=sd(PC1))

Category<- PCA$Category

leves<- c("WT"= 1,"Neo\_het"=0.929175984,"Delta\_het"=0.629659273,"Neo\_homo"=0.258340585,"Delta\_neo"=0.12315139)

PC1xRNA<- cbind(PC1\_RNA,leves)

w <- ggplot(PC1xRNA,aes(PC1xRNA$leves, PC1xRNA$PC1))

w + geom\_point(aes(fill = Category),colour="black", pch=21,size=I(5), alpha=I(.8)) + xlab("mean Fgf8 level by RT-PCR") +

 ylab("PC1")

PCAfgf8 <- read.table ("RNAseq\_fgf8\_ggplot.txt", header = TRUE)

PCAfgf8$Genotype <- factor(PCAfgf8$Genotype, levels = c("WT","Neo\_het","Delta\_het","Neo\_homo","Delta\_neo"))

y <- ggplot(PCAfgf8,aes(PCAfgf8$Fgf8, PCAfgf8$PC1))

y + geom\_point(aes(fill = Genotype),colour="black", pch=21,size=I(5), alpha=I(.8)) + xlab("Fgf8 level by RT-PCR") +

 ylab("PC1 (44% Variance)") +

 geom\_smooth(method = "lm", col = "blue") +

 labs(title = paste("Adj R2 = ",signif(summary(lmfit)$adj.r.squared, 5),

 " P =",signif(summary(lmfit)$coef[2,4], 5)))

PCAfgf8bygroup <- ddply(PCAfgf8, "Genotype", summarise, PC1 = mean(PC1), Fgf8 = mean(Fgf8))

lmfit <- lm(PC1 ~ Fgf8, PCAfgf8)

pcr <- read.csv("fgf8\_data\_summary\_322.csv")

pcr$Genotype <- factor(pcr$X, levels = c("WT","floxwt","NeoWt","floxnull","DeltaWt","floxwt:crect","Neo","floxnull:crect","DeltaNeo" ))

b <- ggplot(pcr, aes(x= Genotype, y= Average.fold.change, fill = Genotype))

b + geom\_bar(stat="identity") + geom\_errorbar (limits, width=0.2) + theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1)) + labs (y = "Relative Fgf8 expression")

limits <- aes(x = Genotype, ymax= Average.fold.change+SEM, ymin =Average.fold.change-SEM, colour = Genotype)

countsNA <- read.table("Cell counts \_for RG(1).txt", header=TRUE)

counts <- na.omit(countsNA)

mean\_counts <- ddply(counts, "Genotype", summarise, counts = mean(Tail\_Somites), sd\_counts=sd(Tail\_Somites))

mean\_counts$Genotype <- factor(mean\_counts$Genotype, levels = c("WT","DeltaWt", "NeoWt","DeltaNeo"))

std <- function(x) sd(x)/sqrt(length(x))

#levs<- factor(counts$Genotype, levels=c("DeltaNeo"=0.12315139, "Neo\_het"=0.9601877, "WT"= 1.0393736))

counts\_merge <- cbind(mean\_counts,levs)

counts.geno <- merge(counts\_merge, counts)

c <- ggplot(counts.geno,aes(x=levs,y=Tail\_Somites))

c + geom\_boxplot(aes(colour = Genotype))

#counts$levels<- levels(counts$Genotype, c("Delta\_neo"=0.12315139, "Neo\_het"=0.929175984, "WT"= 1))

levs <- c("DeltaNeo"=0.1485272, "DeltaWt"= 0.7024247,"NeoWt"=0.96018, "WT"= 1.0393)

#countss <- merge (counts,levs, by=counts$Genotype)

#counttable <- (read.csv("cellcountdata.csv"))

plot(counttable$Total\_cells\_per\_ml ~ counttable$Tail\_Somites, data= counttable)

fit1= lm(counttable$Total\_cells\_per\_ml ~ counttable$Tail\_Somites, counttable)

abline(fit1)

fit2 = lm(counttable$Total\_cells\_per\_ml ~ (counttable$Tail\_Somites^2)+counttable$Tail\_Somites, data= counttable)

fit2line = predict(fit2)

lm1 <- lm.fit(counttable$Total\_cells\_per\_ml ~ counttable$Tail\_Somites)

plot(fit2)

norm\_cell <- (counttable$Total\_cells\_per\_ml/(counttable$Tail\_Somites^2))

counttable <- cbind(norm\_cell, counttable)

fit3 = lm((counttable$Total\_cells\_per\_ml^(1/2)) ~ (counttable$Tail\_Somites), data= counttable)

plot(fit3)

#revise of fit from new data Apr 18

fit4 = lm((counts.geno$Cells\_Total^(1/3)) ~ (counts.geno$Tail\_Somites), data= counts.geno)

plot(fit4)

crcell <- (counttable$Total\_cells\_per\_ml^(1/3))

crcell\_norm <- (crcell/(counttable$Tail\_Somites))

counttable <- cbind(crcell, counttable)

counttable <- cbind(crcell\_norm, counttable)

counttable$Genotype <- factor(counttable$Genotype, levels=c( "WT", "DeltaHet", "DeltaNeo"))

d <- ggplot(counttable,aes(x=Genotype,y=crcell\_norm))

d + geom\_boxplot(aes(colour = counttable$Genotype))

#using fit4 (Benedikt's model) residuals in further analysis (cube root cells)

e <- ggplot(counts.geno,aes(x=levs,y=fit4$fitted.values))

e + geom\_boxplot(aes(colour = counts.geno$Genotype))

##subset counts to remove old and young specimens:

subset\_counts <- subset(counts.geno, counts.geno$Somites <= 13 & counts.geno$Somites > 7, select = c(Genotype, counts, levs, Cells\_total, Somites))

fit5 = lm((subset\_counts$Cells\_total^(1/3)) ~ (subset\_counts$Somites), data= subset\_counts)

#plot(fit5)

f <- ggplot(subset\_counts,aes(x=levs,y=fit5$fitted.values))

f + geom\_point (aes(colour = subset\_counts$Genotype))

#Determine if new curve is linear?

fittedcounts <- cbind(counts.geno, fit4$fitted.values)

fit6 = lm(fittedcounts$`fit4$fitted.values` ~ fittedcounts$levs, data= fittedcounts)

summary (fit6)

fit7 <- lm(fittedcounts$`fit4$fitted.values` ~ poly(fittedcounts$levs,3), data=fittedcounts)

summary(fit7)

##revise again jun 30

counts <- read.table("Cell\_counts\_Jun29.txt", header=TRUE)

mean\_counts <- ddply(counts, "Genotype", summarise, counts = mean(Somites), sd\_counts=sd(Somites))

levs <- c("DeltaNeo"=0.1485272, "DeltaWt"= 0.7024247,"Neo"=0.1948377,"NeoWt"=0.96018, "WT"= 1.0393)

counts\_merge <- cbind(mean\_counts,levs)

counts.geno <- merge(counts\_merge, counts)

fit4 = lm((counts.geno$Cells\_Total^(1/3)) ~ (counts.geno$Somites), data= counts.geno)

plot(fit4)

summary(fit4)

fit5 = lm((counts.geno$Cells\_Total^(1/3)) ~ counts.geno$Somites\*Genotype, data= counts.geno)

plot(fit5)

summary(fit5)

counts.geno$Genotype <- factor(counts.geno$Genotype, levels=c( "WT","NeoWt", "DeltaWt","Neo", "DeltaNeo"))

e <- ggplot(counts.geno,aes(x=levs,y=fit5$fitted.values))

(e + geom\_boxplot(aes(colour = counts.geno$Genotype))+ labs(x= "Fgf8 level", y= "Regression of cell number against somite")

 + guides(colour =guide\_legend(title="Genotypes")))

fittedcounts <- cbind(counts.geno, fit4$fitted.values)

fit6 = lm(fittedcounts$`fit4$fitted.values` ~ fittedcounts$levs, data= fittedcounts)

summary (fit6)

fit7 <- lm(fittedcounts$`fit4$fitted.values` ~ poly(fittedcounts$levs,3), data=fittedcounts)

summary(fit7)

#just june 29

counts <- read.table("june29only.txt")

mean\_counts <- ddply(counts, "V4", summarise, counts = mean(V2), sd\_counts=sd(V2))

#regression of Fgf8 level against shape data (duh, thanks Nathan!)

#make new geomorph data frame

allom <- procD.allometry(Y.dta ~ fgf8p0, logsz = FALSE)

all.nlm <- procD.lm(Y.dta ~ fgf8p0+I(fgf8p0^(2)))

plot(allom, method="CAC")

allomdf <- data.frame(allom$CAC, fgf8, Class)

head(allomdf)

r <- ggplot(data=allomdf,aes(fgf8, allom$CAC))

(r + geom\_point(aes(fill = Class),colour="black", pch=21,size=I(5), alpha=I(.8))

+ geom\_smooth(method = "lm", formula = y ~ splines::bs(x, 3))

+ labs(x= "Fgf8 level", y= "Regression Residuals")

+ guides(fill =guide\_legend(title="Genotypes")))

mlin <- procD.lm(Y.dta~fgf8)

mqua <- procD.lm(Y.dta ~ fgf8+I(fgf8^(2)))

mq3 <- procD.lm(Y.dta ~ fgf8+I(fgf8^(3)))

mq3ploy <- procD.lm(Y.dta ~ poly(fgf8,3))

mlin <- anova(lm(Y.dta2D~fgf8))

mqua <- manova(Y.dta2D ~ fgf8+I(fgf8^(2)))

mq3 <- manova(Y.dta2D ~ fgf8+I(fgf8^(3)))

mq3ploy <- lm(Y.dta2D ~ poly(fgf8,3))

anova(mlin,mqua,mq3,mq3ploy)

anova(mlin,mqua,mq3ploy)

anova(mqua,mq3ploy)

comp <- advanced.procD.lm(Y.dta ~fgf8, ~fgf8+I(fgf8^(2)))

comp2 <- advanced.procD.lm(Y.dta ~ fgf8, ~ fgf8+I(fgf8^(3)))

comp3<- advanced.procD.lm(Y.dta ~ fgf8, ~ poly(fgf8,3))

###this doesn't work on the full matrix, trying on PCscores (regroup$pc.scores)

pc.scores <- regroup$pc.scores[,1:180]

mlin <- manova(pc.scores ~ fgf8)

mqua <- manova(pc.scores~ fgf8+I(fgf8^(2)))

mq3 <- manova(pc.scores~ fgf8+I(fgf8^(3)))

mq3ploy <- manova(pc.scores ~ poly(fgf8,3))

anova(mlin,mqua,mq3,mq3ploy)

anova(mlin,mqua,mq3ploy)

anova(mqua,mq3ploy)

anova (lm(allom$CAC ~ fgf8))