Popgen

Loading libraries

Loading previous results

```
setwd("~/GitHub/devnetwork/")
load("results/DEtests.RData")
load("results/collectedPhylo.RData")
tau <- read.csv("results/bee_tau.csv")
sexGenes <- read.csv("results/dmel_sexGenes.csv")
antConn <- read.csv("results/antConnectivity.csv")
beeConn <- read.csv("results/beeConnectivity.csv")
beePi <- read.csv("results/apis.gene.pi.csv")
beeSub <- read.csv("results/substitutions.csv")
antEvol <- read.csv("data/MpharAnn.csv")
antConstraint <- read.table("results/MKtest_constraint_ant.csv")
beeSnipre <- read.csv("results/bayesian_results_apis.csv")
alphaResults <- read.csv("results/collectedAlpha.csv")
colnames(antConstraint) = c("Gene","f")
antSub = antEvol[!is.na(antEvol$Fixed.Non.Synonymous),c(1,13,12,15,14,17,16)]
colnames(antSub) = c("Gene","FN","FS","PN","PS","Trepl","Tsil")
antSub = cbind(antSub,f=as.numeric(as.character((t(antConstraint[2,4:(ncol(antConstraint) - 1)])))))
beeConstraint = read.table("results/MKtest_constraint_bee.csv")
beeSub = cbind(beeSub,f=as.numeric(as.character((t(beeConstraint[2,4:(ncol(beeConstraint) - 1)])))))
antGamma = antEvol[!is.na(antEvol$BSnIPRE.gamma),c(1,19,20)]
beeGamma = beeSnipre[!is.na(beeSnipre$BSnIPRE.gamma),c("gene","BSnIPRE.est","BSnIPRE.gamma")]
colnames(beeGamma)[1] = "Gene"
antSub = merge(antSub,antGamma,by="Gene")
beeAnn = read.csv("results/annotation.csv",header=F)
beeAnn = beeAnn[!duplicated(beeAnn$V5),]
beeGamma = merge(beeGamma,beeAnn,by.x="Gene",by.y="V5")
beeSub = merge(beeSub,beeGamma[,c(2,3,7)],by.x = "Gene",by.y="V4")

beeT <- read.table("data/bees.tpm.txt",header=TRUE)
antT <- read.table("data/ants.tpm.txt",header=TRUE)
modifyDF <- function(data){
  rownames(data)=data[,1]
  return(data[!grepl("ERCC",rownames(data)),-c(1)])
}

genFactor <- function(counts){
  factors <- data.frame(sample=colnames(counts))
  factors$stage = 8
  factors$stage[grepl("_E",factors$sample)]=1
  factors$stage[grepl("L1",factors$sample)]=2
  factors$stage[grepl("L2",factors$sample)]=3
  factors$stage[grepl("L3",factors$sample)]=4
  factors$stage[grepl("L4",factors$sample)]=5
  factors$stage[grepl("L5",factors$sample)]=6
  factors$stage[grepl("P",factors$sample)]=7
```

```r
  factors$tissue="larva"
  factors$tissue[grepl("P",factors$sample)]="pupa"
  factors$tissue[grepl("_E",factors$sample)]="egg"
  factors$tissue[grepl("G\\.",factors$sample)]="abdomen"
  factors$tissue[grepl("H\\.",factors$sample)]="head"
  factors$tissue[grepl("M\\.",factors$sample)]="thorax"
  factors$NF=NA
  factors$NF[grepl("_N",factors$sample)]="nurse"
  factors$NF[grepl("_F",factors$sample)]="forager"
  factors$caste="worker"
  factors$caste[grepl("_S|_V|_AQ|_G",factors$sample)]="queen"
  factors$caste[grepl("_M",factors$sample)]="male"
  factors$VM=NA
  factors$VM[grepl("_V",factors$sample)]="virgin"
  factors$VM[grepl("_AQ",factors$sample)]="mated"
  factors$colony=1
  factors$colony[grepl(".2",factors$sample)]=2
  factors$colony[grepl(".3",factors$sample)]=3
  for (i in 2:7){
    factors[,i]=as.factor(factors[,i])
  }
  rownames(factors)=factors$sample
  factors$caste = factor(factors$caste,levels = c("queen","male","worker")) #Queen genes will always be
  factors$tissue = factor(factors$tissue,levels = c("egg","larva","pupa","head","thorax","abdomen"))
  factors$NF = factor(factors$NF,levels = c("nurse","forager")) #Make nurse genes down-regulated becaus
  return(factors)
}

beeT <- modifyDF(beeT)
antT <- modifyDF(antT)
antT = antT[rowSums(antT) > 0,]
beeT = beeT[rowSums(beeT) > 0,]
factorA <- genFactor(antT)
factorB <- genFactor(beeT)

#Bootstrap for confidence intervals of mean
bootMean <- function(d,boots,stage,species,variable){
  m = mean(d)
  mboot = sapply(1:boots,function(x){
    mean(sample(d,length(d),replace=TRUE))
  })
  return(c(mean=m,c1=quantile(mboot,0.025),c2=quantile(mboot,0.975),
           stage=stage,species=species,value=variable))
}

calcMean <- function(d,col){
  Sum <- ldply(lapply(levels(d$species),function(i){
    ldply(lapply(levels(d$stage),function(j){
      ldply(lapply(levels(d$value),function(k){
        bootMean(d[d$species==i&d$stage==j&d$value==k,col],1000,j,i,k)
      }))
    }))
  }))
  colnames(Sum)[2:3] = c("c1","c2")
  for (i in 1:3){
    Sum[,i] = as.numeric(as.character(Sum[,i]))
  }
  Sum$stage = factor(Sum$stage,levels = c("larva","pupa","head","thorax","abdomen"))
  return(Sum)
}
```

Collecting Data

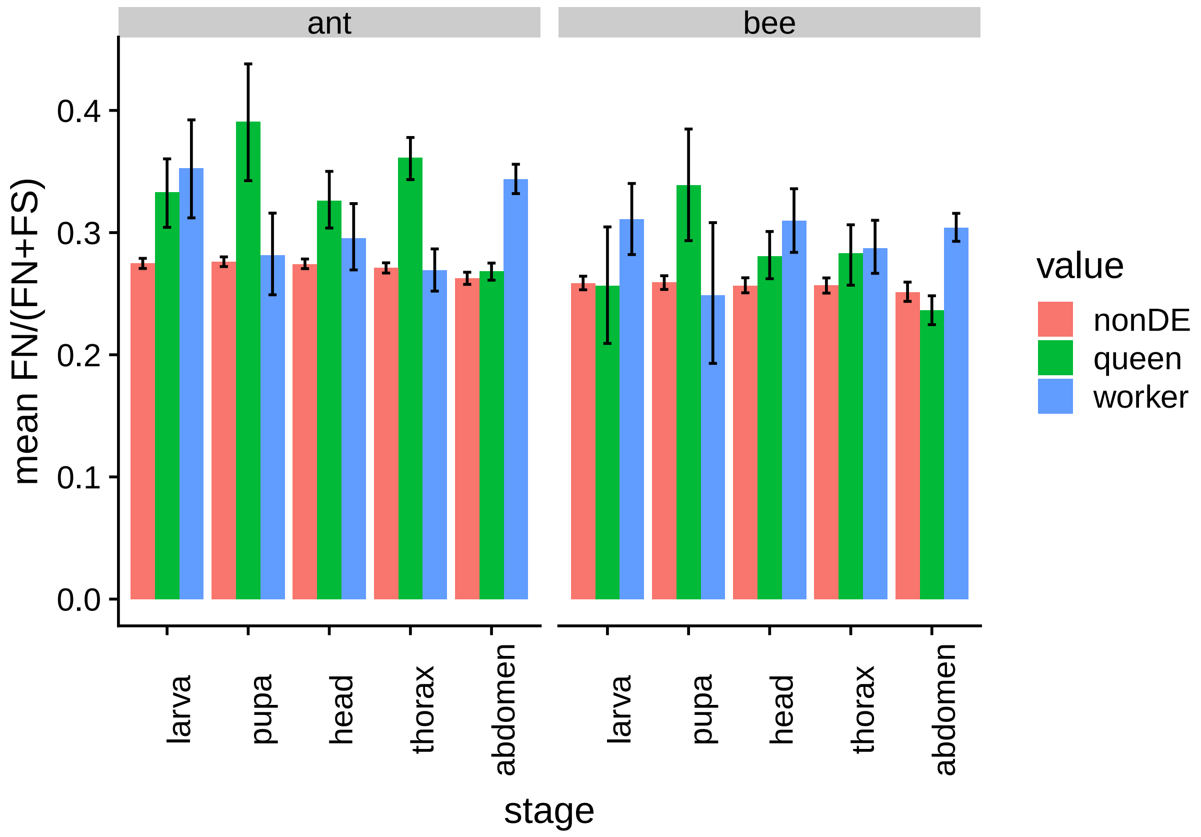
```r
beeS = merge(beeRes[[2]],beeSub,by="Gene") %>% melt(id.vars=colnames(beeSub))
antS = merge(antRes[[2]],antSub,by="Gene") %>% melt(id.vars=colnames(antSub))
beeS$species = "bee"
antS$species = "ant"
beeS$divRank = rank(beeS$FN/(beeS$FS+beeS$FN+1))
antS$divRank = rank(antS$FN/(antS$FS+antS$FN+1))

allS = rbind(beeS,antS)
colnames(allS)[11] = "stage"
allS$species = as.factor(allS$species)
allS$value = as.factor(allS$value)
allS$stage = factor(allS$stage,levels = c("larva","pupa","head","thorax","abdomen"))
```

Part 1: Divergence of worker and queen genes

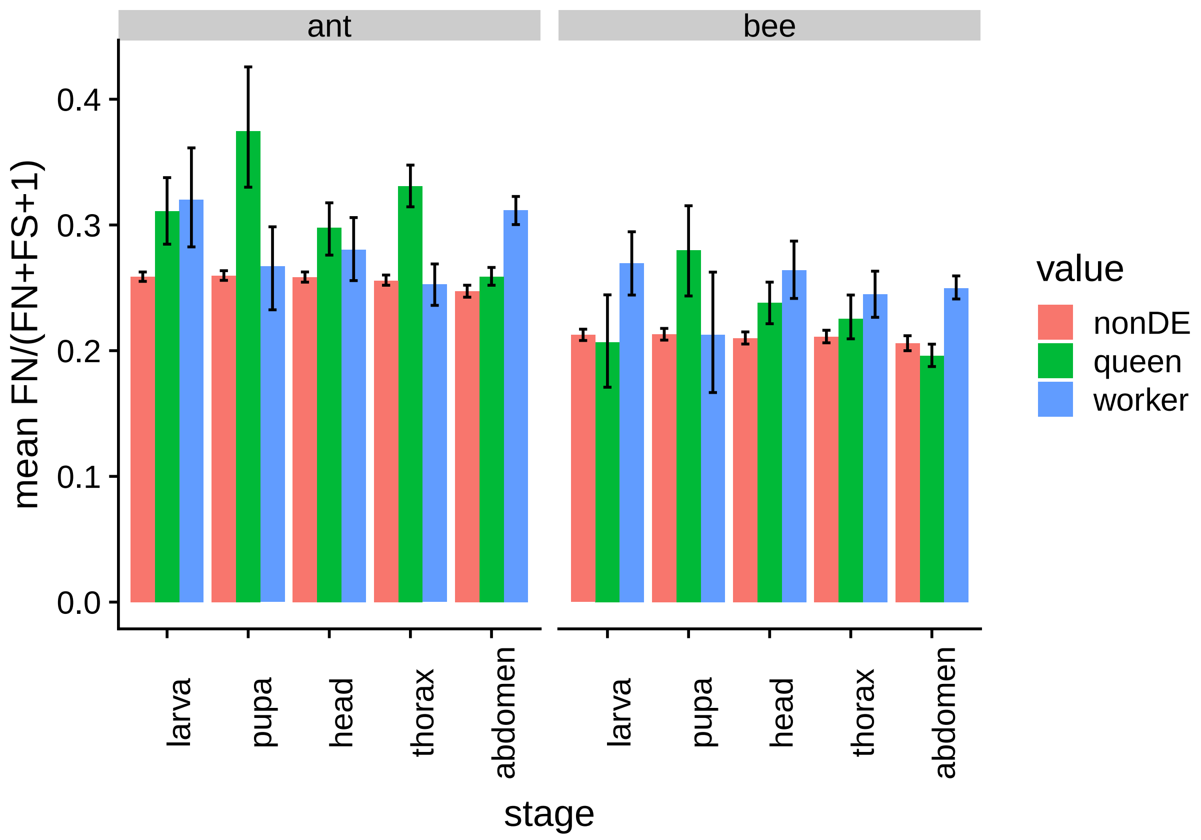
```r
allS$FnFS = allS$FN/(allS$FN+allS$FS)
FnFS = calcMean(allS[!is.na(allS$FnFS),],"FnFS")

ggplot(FnFS,aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  facet_wrap(. ~ species)+
  ylab("mean FN/(FN+FS)")+
  theme(axis.text.x=element_text(angle=90))
```



```
allS$FnFS2 = allS$FN/(allS$FN+allS$FS+1)
FnFS = calcMean(allS,"FnFS2")

##This plot includes genes with zero substitutions
ggplot(FnFS,aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  facet_wrap(. ~ species)+
  ylab("mean FN/(FN+FS+1)")+
  theme(axis.text.x=element_text(angle=90))
```

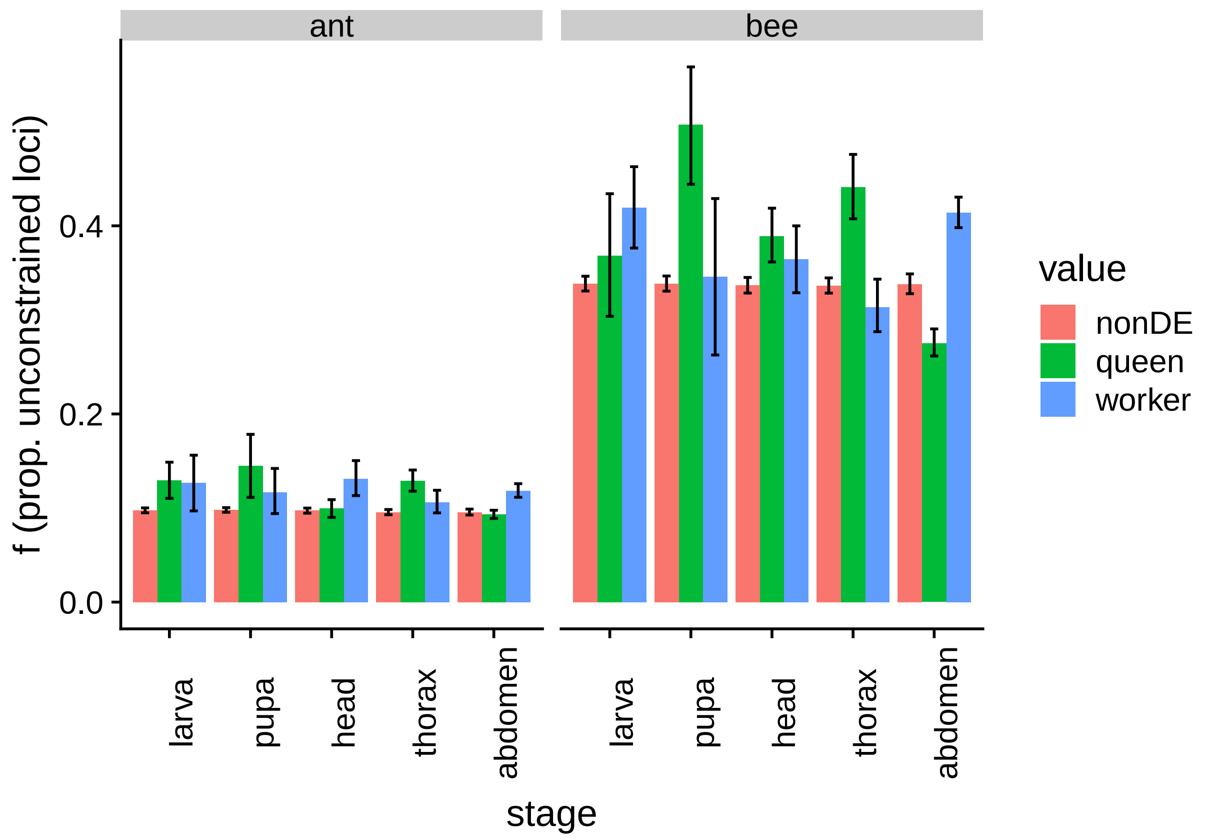


Part 2: Selective Constraint and Pi

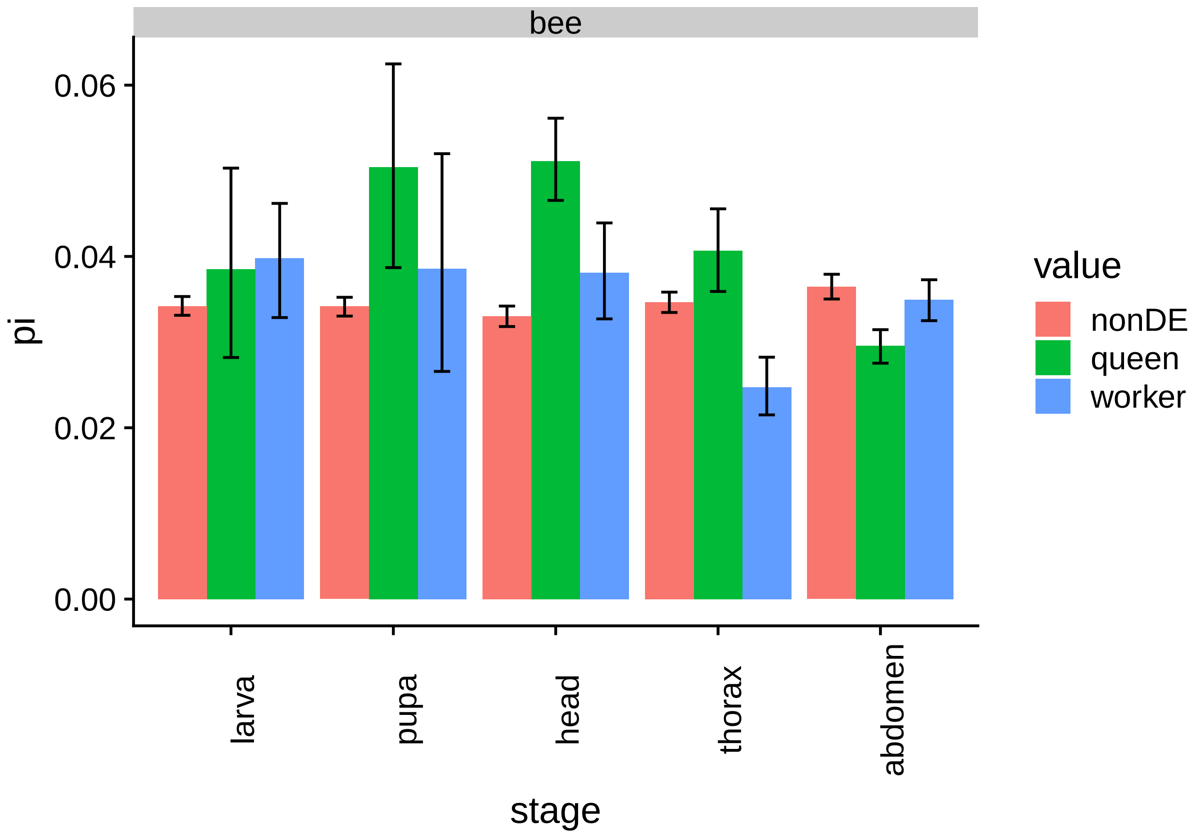
- note that for now only have pi for bees

```
#constraint
f = calcMean(allS,"f")

ggplot(f,aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  facet_wrap(. ~ species)+
  ylab("f (prop. unconstrained loci)")+
  theme(axis.text.x=element_text(angle=90))
```

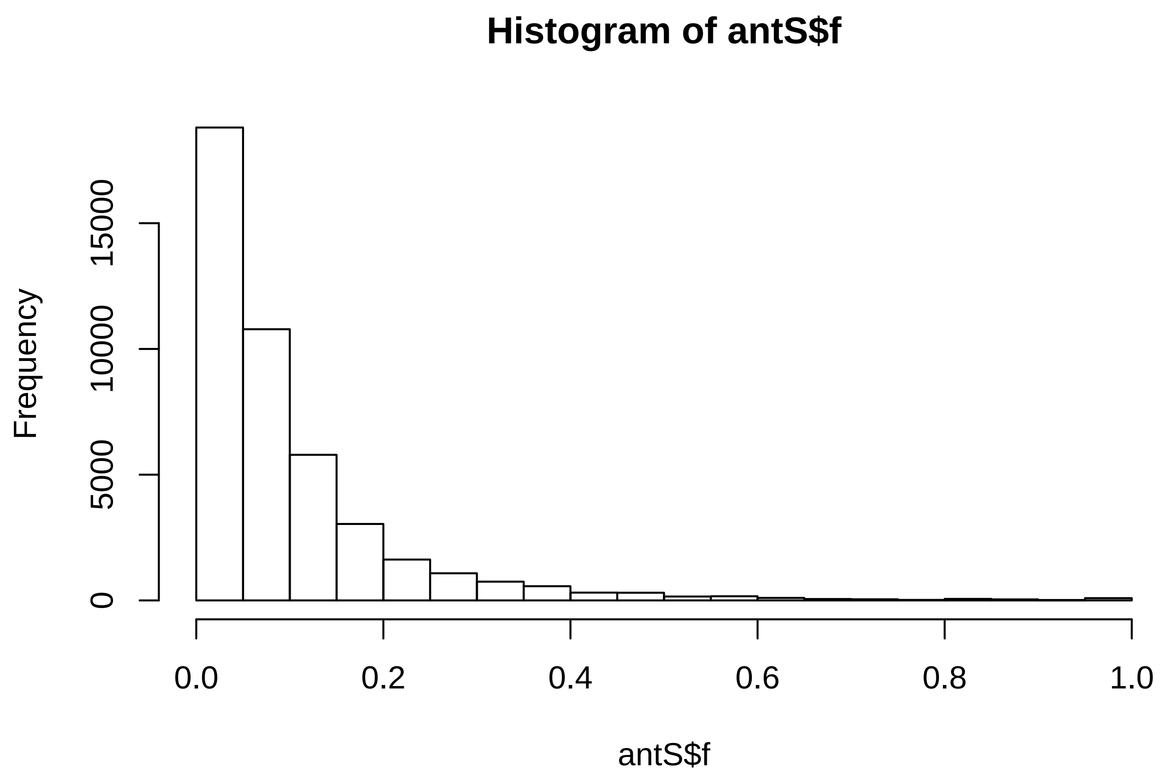


```
# #pi (just bees)
Bpi = merge(beePi,allS,by="Gene")
p = calcMean(droplevels(Bpi),"pi")
ggplot(p,aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  facet_wrap(. ~ species)+
  ylab("pi")+
  theme(axis.text.x=element_text(angle=90))
```

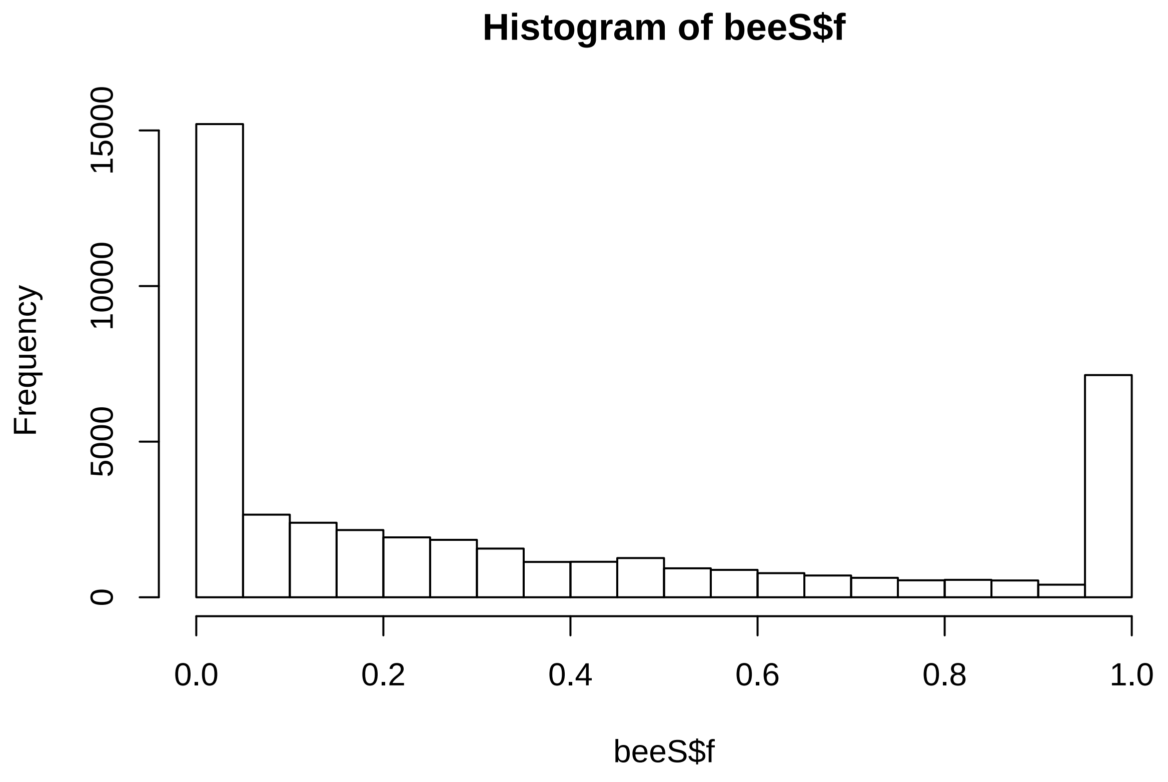


Note: the distributions of f look weird (but are consistent with more diversity in pharaonis sequences)

```
hist(antS$f)
```


Histogram of antS$f
Frequency
0
5000
10000
15000
0.0
0.2
0.4
0.6
0.8
1.0
antS$f

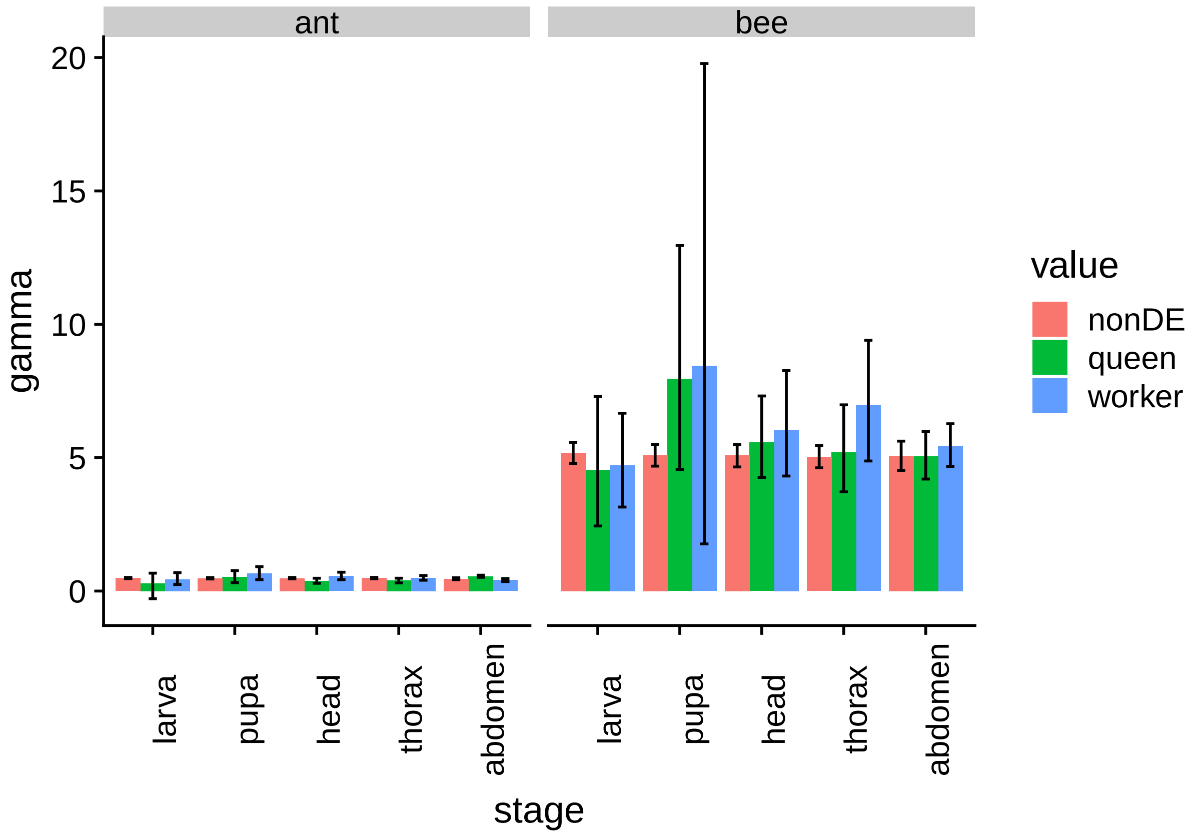
```
hist(beeS$f)
```


Histogram of beeS$f
Frequency
0
5000
10000
15000
0.0
0.2
0.4
0.6
0.8
1.0
beeS$f

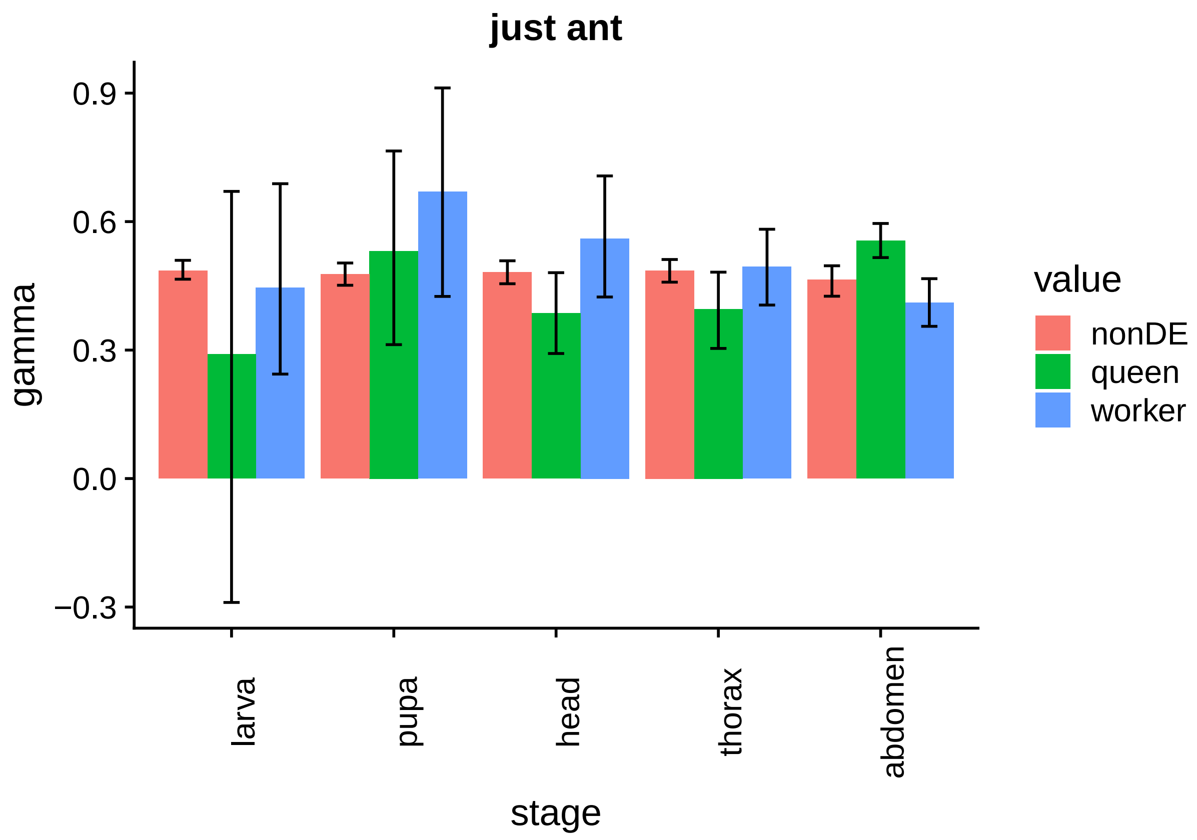
Part 3: Positive selection, Gamma

```
gamma = calcMean(allS,"BSnIPRE.gamma")

ggplot(gamma,aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  facet_wrap(. ~ species)+
  ylab("gamma")+
  theme(axis.text.x=element_text(angle=90))
```

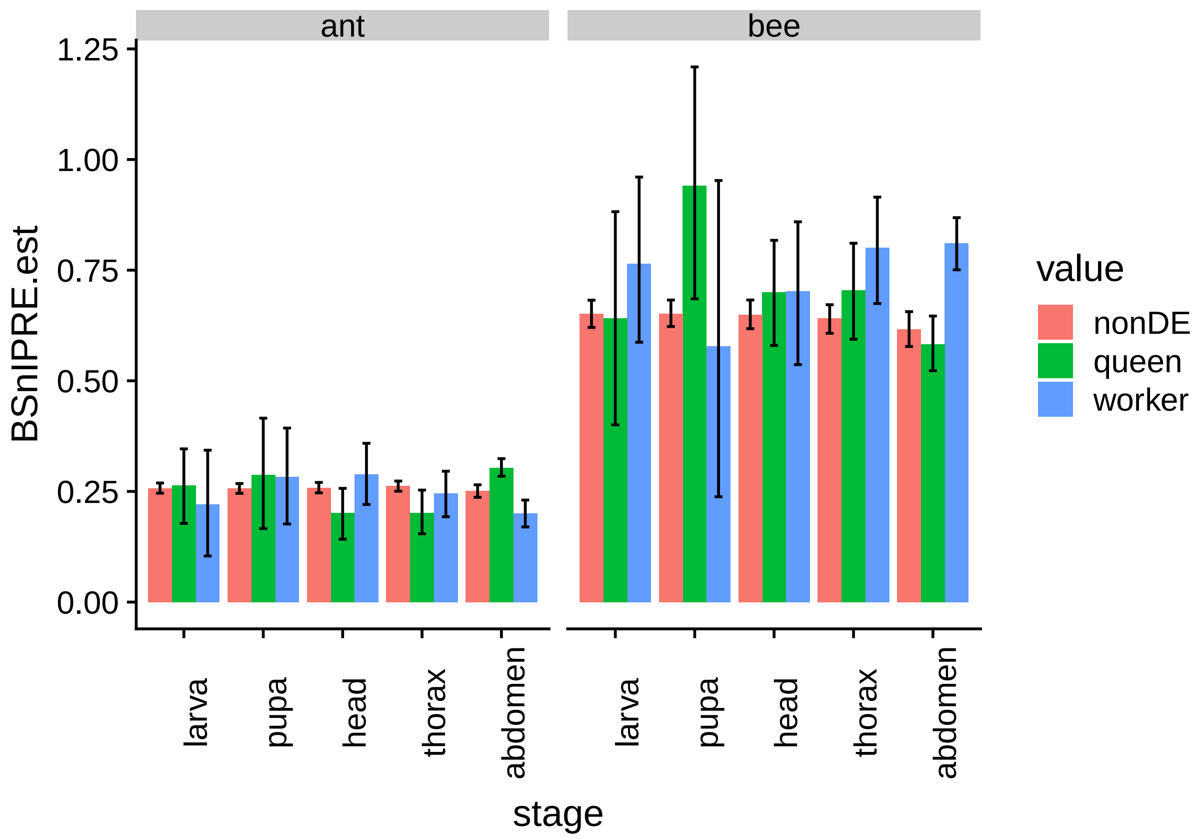


```
ggplot(gamma[gamma$species=="ant",],aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  ylab("gamma")+
  ggtitle("just ant")+
  theme(axis.text.x=element_text(angle=90))
```



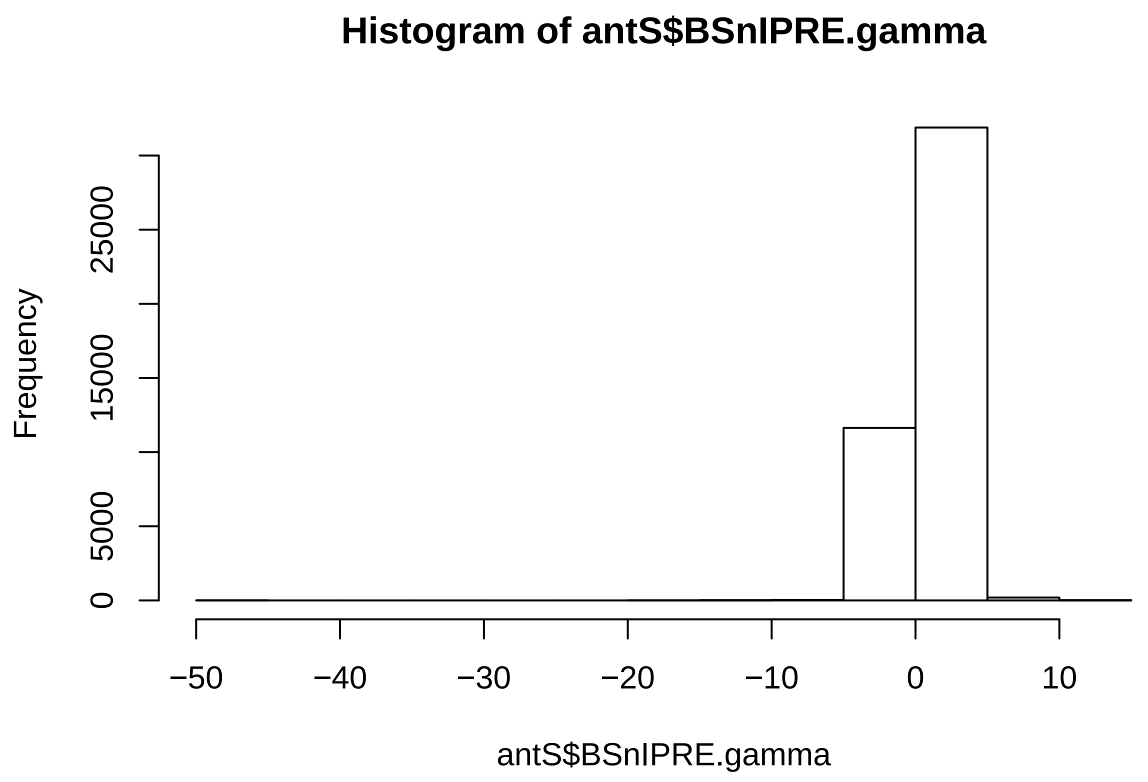
```
est = calcMean(allS,"BSnIPRE.est")

ggplot(est,aes(x = stage,fill=value,y=mean))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position=position_dodge(width=0.9),width=0.3)+
  facet_wrap(. ~ species)+
  ylab("BSnIPRE.est")+
  theme(axis.text.x=element_text(angle=90))
```

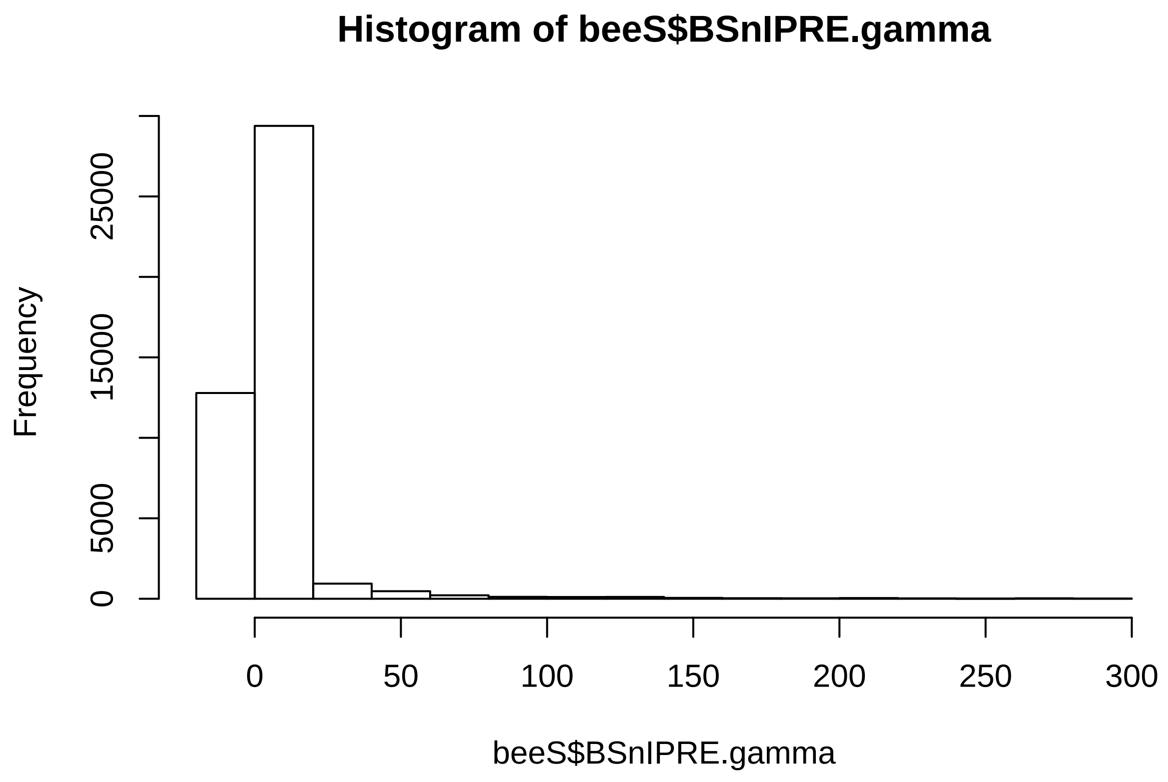


Honey bee gamma distribution also looks weird

```r
hist(antS$BSnIPRE.gamma)
```



```r
hist(beeS$BSnIPRE.gamma)
```



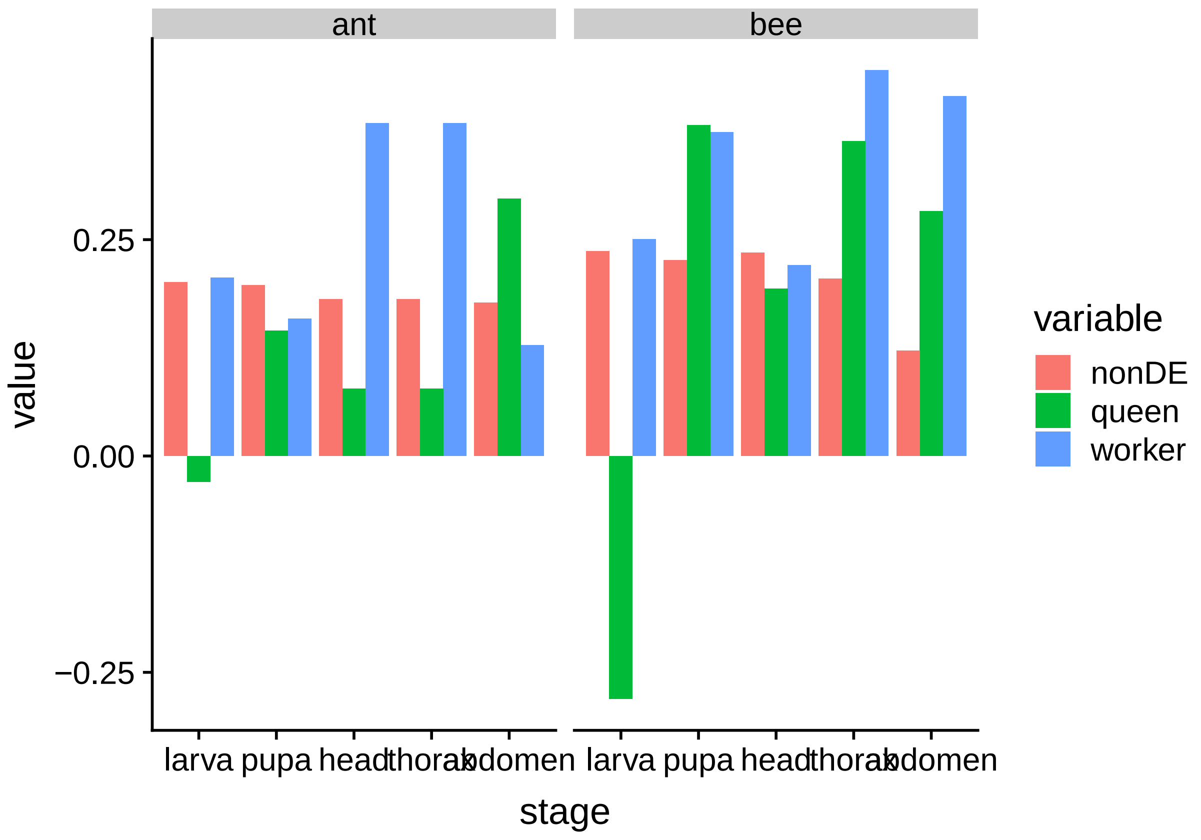
Part 3: Positive selection, alpha

```
alpha <- read.csv("results/alphaHead.csv")
alpha = alpha[alpha$LnL!="LnL",]
alphaM = melt(alpha[,c(4:8)],id.vars =c("stage","species"))
```

```
## Warning: attributes are not identical across measure variables; they will
## be dropped
```

```
alphaM$stage = factor(alphaM$stage,levels = c("larva","pupa","head","thorax","abdomen"))
alphaM$value = as.numeric(as.character(alphaM$value))
levels(alphaM$variable) = c("nonDE","queen","worker")

ggplot(alphaM,aes(x = stage,y=value,fill=variable))+
  geom_bar(stat="identity",position = position_dodge())+
  facet_wrap(. ~ species)
```


ant
bee
value
larva
pupa
head
thorax
abdomen
stage
variable
nonDE
queen
worker