

Mphar_Amel

Loading previous results

```
load("results/DEtests.RData")
load("results/collectedPhylo.RData")
load("results/PlaidResults.RData")
load("results/tau_results.RData")
sexGenes <- read.csv("results/dmel_sexGenes.csv")
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)])
}
beeT <- modifyDF(beeT)
antT <- modifyDF(antT)
antT = antT[rowSums(antT) > 0,]
beeT = beeT[rowSums(beeT) > 0,]
```

Loading libraries, themes

Caste-bias across development

```
compDef <- function(antR,beeR){

  #Define whether or not orthologs exist
  antR$ortholog_found = antR$OGG_found = FALSE
  antR$ortholog_found[antR$Gene %in% AllPS$Gene.x] = TRUE
  antR$OGG_found[antR$Gene %in% ACUogg$gene_Mphar] = TRUE
  aM = melt(antR,id.vars = c("Gene","ortholog_found","OGG_found"))
  aD = ddply(aM,~variable,summarize,
    NDE = sum(value=="nonDE"),
    no_ortholog = sum(value!="nonDE" & !ortholog_found),
    dup = sum(value!="nonDE" & ortholog_found & !OGG_found),
    OGG = sum(value!="nonDE" & OGG_found))

  #Do same thing for apis
  beeR$ortholog_found = beeR$OGG_found = FALSE
  beeR$ortholog_found[beeR$Gene %in% AllPS$Gene.y] = TRUE
  beeR$OGG_found[beeR$Gene %in% ACUogg$gene_Amel] = TRUE
  bM = melt(beeR,id.vars = c("Gene","ortholog_found","OGG_found"))
  bD = ddply(bM,~variable,summarize,
    NDE = sum(value=="nonDE"),
    no_ortholog = sum(value!="nonDE" & !ortholog_found),
    dup = sum(value!="nonDE" & ortholog_found & !OGG_found),
    OGG = sum(value!="nonDE" & OGG_found))
  colnames(bM)[5] = "value_apis"

  #Getting all results together, tabulating
```

```

aM = merge(aM[, -c(2,3)], ACUogg, by.x="Gene", by.y="gene_Mphar")
bM = merge(bM[, -c(2,3)], ACUogg, by.x="Gene", by.y="gene_Amel")
allM = merge(aM, bM, by=c("OGGacu", "variable"))
allD = ddply(allM, ~variable, summarize,
             DEboth = sum(value_apis!="nonDE" & value != "nonDE"))

#Calculate number of genes which are DE, have ortholog, and aren't commonly DEG
aD$DEboth = bD$DEboth = allD$DEboth
aD$OGG = aD$OGG - aD$DEboth
bD$OGG = bD$OGG - bD$DEboth

aDM = melt(aD, id.vars = "variable")
bDM = melt(bD, id.vars = "variable")
colnames(aDM) = colnames(bDM) = c("stage", "DEtype", "value")
aDM$species = "ant"
bDM$species = "bee"

#Get data back together
d = rbind(aDM, bDM)
d$species=as.factor(d$species)
levels(d$species) = c("M. pharaonis", "A. mellifera")
levels(d$DEtype) = c("NDE", "no ortholog", "paralogs present", "ortholog present", "non-conserved caste-biased")
return(d)
}

d = compDef(antRes[[2]], beeRes[[2]])

levels(d$stage)[1] = "larva*"

levels(d$species) = c("ant", "bee")
p1m <- ggplot(d[d$DEtype!="NDE", ],
             aes(x = stage, y = value, fill = DEtype))+
  geom_bar(stat="identity")+
  plot2theme+
  ylim(0,5500)+
  facet_grid(. ~ species)+
  xlab("stage/tissue")+
  scale_fill_manual(values = DE_palette[c(3,5,7,9)])+
  ylab("number of\ncaste-biased genes")+
  theme(axis.text.x = element_text(angle=-25, hjust=0.1),
        legend.position = "top",
        strip.text = element_text(size=22, face="bold"),
        legend.title = element_blank(),
        strip.background = element_rect(color=NA, fill=NA),
        plot.margin = margin(0.5, 2, 0.5, 0.5, "cm"))+
  theme(panel.spacing = unit(2, "lines"))

p2 <- ggplot(d[d$DEtype!="NDE" & d$species=="ant", ],
             aes(x = stage, y = value, fill = DEtype))+
  geom_bar(stat="identity", position = "fill")+
  main_theme+
  scale_fill_manual(values = DE_palette[c(3,5,7,9)])+
  theme(axis.text = element_text(size = 8.5), axis.ticks = element_blank())+

```

```

xlab("")+
ylab("proportion")+
scale_y_continuous(breaks = c(0,0.5,1))+
theme(axis.text.x = element_blank(),
      axis.ticks.x = element_blank(),
      rect = element_rect(fill="transparent"),
      legend.position = "none",
      panel.border = element_blank(),
      panel.background = element_blank(),
      legend.title = element_blank(),
      axis.title.y = element_text(margin=margin(t=0,r=0,l=0,b=0)),
      axis.text.y = element_text(margin=margin(t=0,r=-3,l=0,b=0)),
      axis.title = element_text(size=10),
      plot.margin = margin(0,0,0,0,"cm"))

p3 <- ggplot(d[d$DEtype!="NDE" & d$species=="bee",],
            aes(x = stage, y = value, fill = DEtype))+
geom_bar(stat="identity",position = "fill")+
main_theme+
theme(axis.text = element_text(size = 8.5),axis.ticks = element_blank())+
xlab("")+
scale_fill_manual(values = DE_palette[c(3,5,7,9)])+
ylab("proportion")+
scale_y_continuous(breaks = c(0,0.5,1))+
theme(axis.text.x = element_blank(),
      axis.ticks.x = element_blank(),
      rect = element_rect(fill="transparent"),
      legend.position = "none",
      panel.border = element_blank(),
      panel.background = element_blank(),
      legend.title = element_blank(),
      axis.title.y = element_text(margin=margin(t=0,r=0,l=0,b=0)),
      axis.text.y = element_text(margin=margin(t=0,r=-3,l=0,b=0)),
      axis.title = element_text(size=10),
      plot.margin = margin(0,0,0,0,"cm"))

pQ <- ggdraw()+
draw_plot(p1m+
          theme(legend.text = element_text(size=15),
                legend.key.width = unit(1,"cm"),
                legend.position = "bottom"))+
draw_plot(p2,x=0.15,y=0.69,height=0.18,width=0.18)+
draw_plot(p3,x=0.57,y=0.69,height=0.18,width=0.18)

```

Figure 1a. Caste bias across development

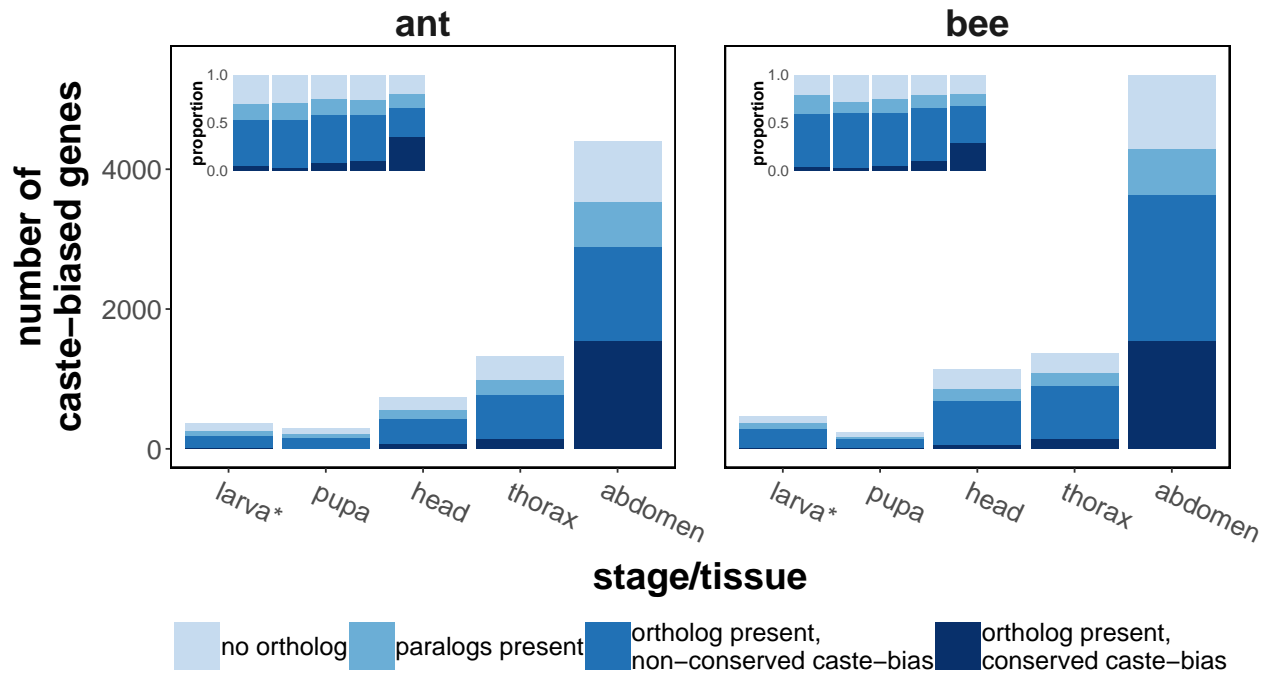


Figure 1b. Nurse/forager bias

```
d = compDef(antSocRes[[2]],beeSocRes[[2]])

p1 <- ggplot(d[d$DEtype!="NDE",],
             aes(x = stage, y = value, fill = DEtype))+
  geom_bar(stat="identity")+
  main_theme+
  xlab("tissue")+
  ylim(0,4500)+
  facet_grid(. ~ species)+
  scale_fill_manual(values = DE_palette[c(3,5,7,9)])+
  ylab("number of behavior-biased genes")+
  theme(axis.text.x = element_text(angle=-25,hjust=0.1),
        legend.position = "top",
        strip.text = element_text(size=20,face="bold.italic"),
        axis.title.y=element_text(margin=margin(t=0,l=0,r=10,b=0)),
        legend.title = element_blank(),
        strip.background = element_rect(color="black",fill="darkgrey"),
        plot.margin = margin(0,2,2,2,"cm"))

p2 <- ggplot(d[d$DEtype!="NDE" & d$species=="M. pharaonis",],
             aes(x = stage, y = value, fill = DEtype))+
  geom_bar(stat="identity",position = "fill")+
  main_theme+
  scale_fill_manual(values = DE_palette[c(3,5,7,9)])+
  theme(axis.text = element_text(size = 8.5),axis.ticks = element_blank())+
  xlab("")+
```

```

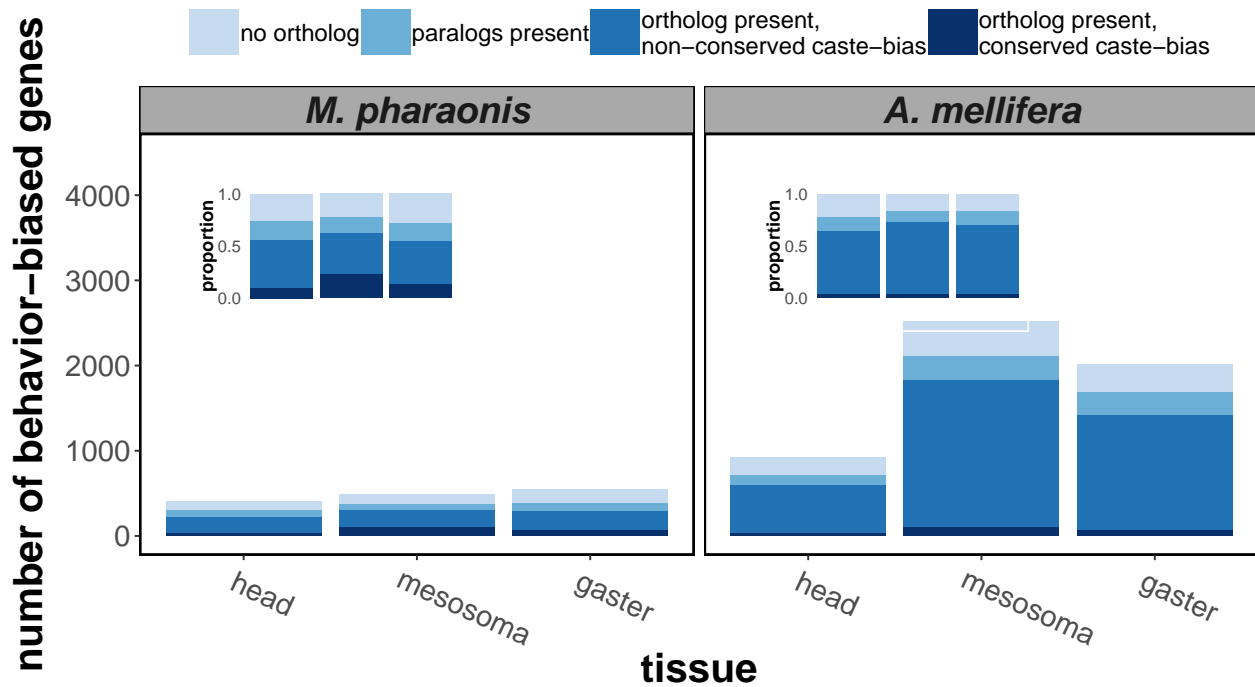
ylab("proportion")+
scale_y_continuous(breaks = c(0,0.5,1))+
theme(axis.text.x = element_blank(),
      axis.ticks.x = element_blank(),
      rect = element_rect(fill="transparent"),
      legend.position = "none",
      panel.border = element_blank(),
      panel.background = element_blank(),
      legend.title = element_blank(),
      axis.title.y = element_text(margin=margin(t=0,r=0,l=0,b=0)),
      axis.text.y = element_text(margin=margin(t=0,r=-3,l=0,b=0)),
      axis.title = element_text(size=10),
      plot.margin = margin(0,0,0,0,"cm"))

p3 <- ggplot(d[d$DEtype!="NDE" & d$species=="A. mellifera",],
            aes(x = stage, y = value, fill = DEtype))+
geom_bar(stat="identity",position = "fill")+
main_theme+
theme(axis.text = element_text(size = 8.5),axis.ticks = element_blank())+
xlab("")+
scale_fill_manual(values = DE_palette[c(3,5,7,9)])+
ylab("proportion")+
scale_y_continuous(breaks = c(0,0.5,1))+
theme(axis.text.x = element_blank(),
      axis.ticks.x = element_blank(),
      rect = element_rect(fill="transparent"),
      legend.position = "none",
      panel.border = element_blank(),
      panel.background = element_blank(),
      legend.title = element_blank(),
      axis.title.y = element_text(margin=margin(t=0,r=0,l=0,b=0)),
      axis.text.y = element_text(margin=margin(t=0,r=-3,l=0,b=0)),
      axis.title = element_text(size=10),
      plot.margin = margin(0,0,0,0,"cm"))

pSoc <- ggdraw()+
draw_plot(p1+
          theme(legend.text = element_text(size=13),
                legend.key.width = unit(1,"cm")))+
draw_plot(p2,x=0.2,y=0.58,height=0.18,width=0.18)+
draw_plot(p3,x=0.59,y=0.58,height=0.18,width=0.18)

```

Figure 1b. nurse/forager bias



Phylostrata of genes with conserved abdominal caste bias

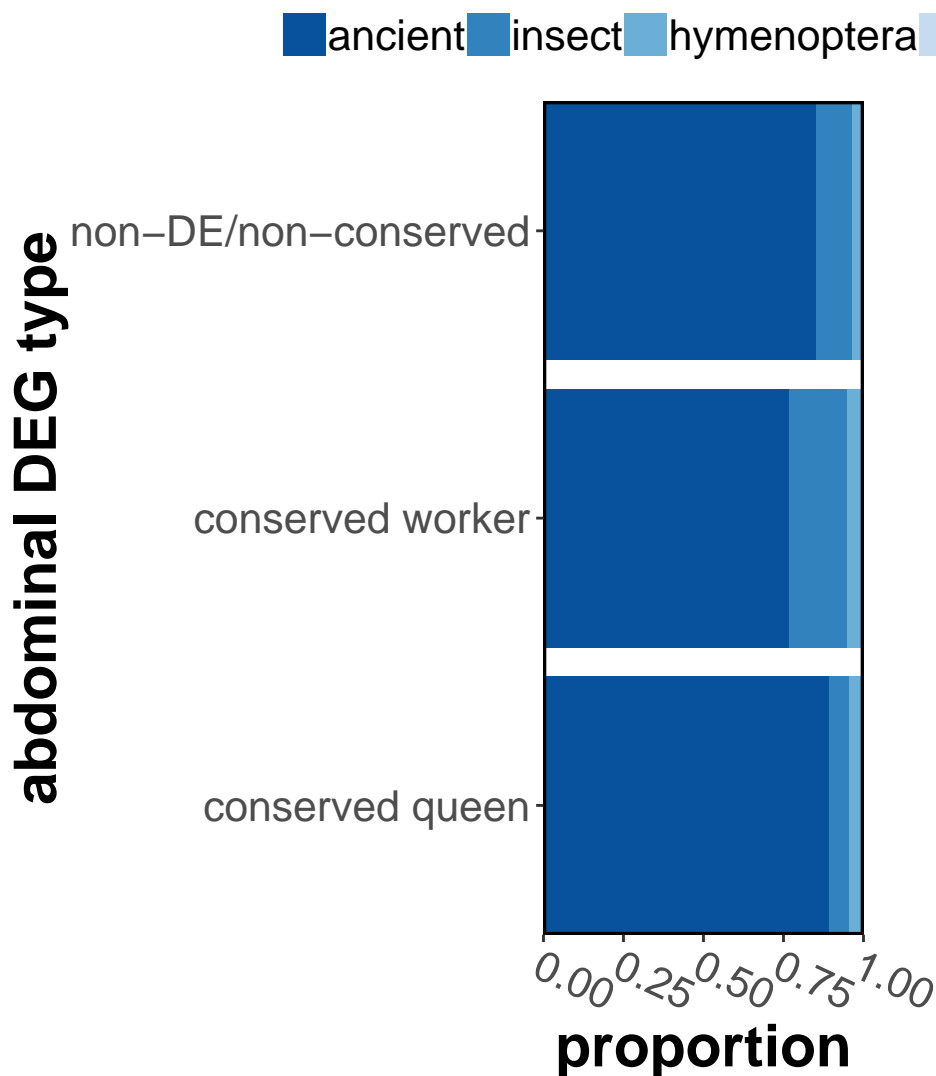
```
ps0 = merge(AllPS_sum, ACUogg, by="OGGacu")
ps0$a2 = "non-DE/non-conserved"

#Find genes with conserved queen or worker bias
ps0$a2[ps0$gene_Amel %in% beeRes[[2]]$Gene[beeRes[[2]]$abdomen=="worker"] &
  ps0$gene_Mphar %in% antRes[[2]]$Gene[antRes[[2]]$abdomen=="worker"]] = "conserved worker"
ps0$a2[ps0$gene_Amel %in% beeRes[[2]]$Gene[beeRes[[2]]$abdomen=="queen"] &
  ps0$gene_Mphar %in% antRes[[2]]$Gene[antRes[[2]]$abdomen=="queen"]] = "conserved queen"

levels(ps0$psName)[1] = "ancient"

p <- ggplot(ps0, aes(x = a2, fill = forcats::fct_rev(psName)))+
  ylab("proportion")+
  geom_bar(stat = "count", position = "fill")+
  scale_fill_manual(values = mypalette2[c(2,4,5,6)], name = "phylostrata")+
  main_theme+
  coord_flip()+
  xlab("abdominal DEG type")+
  scale_x_discrete(expand=c(0,0))+
  guides(fill = guide_legend(reverse=T))+
  scale_y_continuous(expand=c(0,0))+
  theme(axis.text.x = element_text(angle=-25, hjust=0),
        legend.position = "top",
        legend.title = element_blank(),
        plot.margin = unit(c(0.5,2,0.5,2), "cm"))
```

Figure 1c. Estimated evolutionary age of genes with abdominal caste bias



Comparing caste bias to sex bias

```
extractBias <- function(Deres){
  sexQ <- rownames(Deres)[Deres$FDR < 0.1 & Deres$logFC < 0]
  sexM <- rownames(Deres)[Deres$FDR < 0.1 & Deres$logFC > 0]
  sexFC <- data.frame(Gene = rownames(Deres), FC = Deres$logFC)
  return(list(FC = sexFC, Queen = sexQ, nonQueen = sexM))
}

AsexRes <- lapply(ant_sexDE, extractBias)
AcasteRes <- lapply(antTests_oneLarv[c(3:5)], extractBias)
BsexRes <- lapply(bee_sexDE, extractBias)
BcasteRes <- lapply(beeTests_oneLarv[c(3:5)], extractBias)
```

Identify genes with conserved bias for plotting

```
ogg11 = ACUogg
ogg11$abdDE = "non-conserved/non-DE"
ogg11$abdDE[(ogg11$gene_Amel %in% BcasteRes[[3]][[2]] & ogg11$gene_Mphar %in% AcasteRes[[3]][[2]])] = "conserved queen"
ogg11$abdDE[(ogg11$gene_Amel %in% BcasteRes[[3]][[3]] & ogg11$gene_Mphar %in% AcasteRes[[3]][[3]])] = "conserved worker"
```

Construct sex bias vs caste bias plots

```
biasPlot <- function(d1,d2,spec){

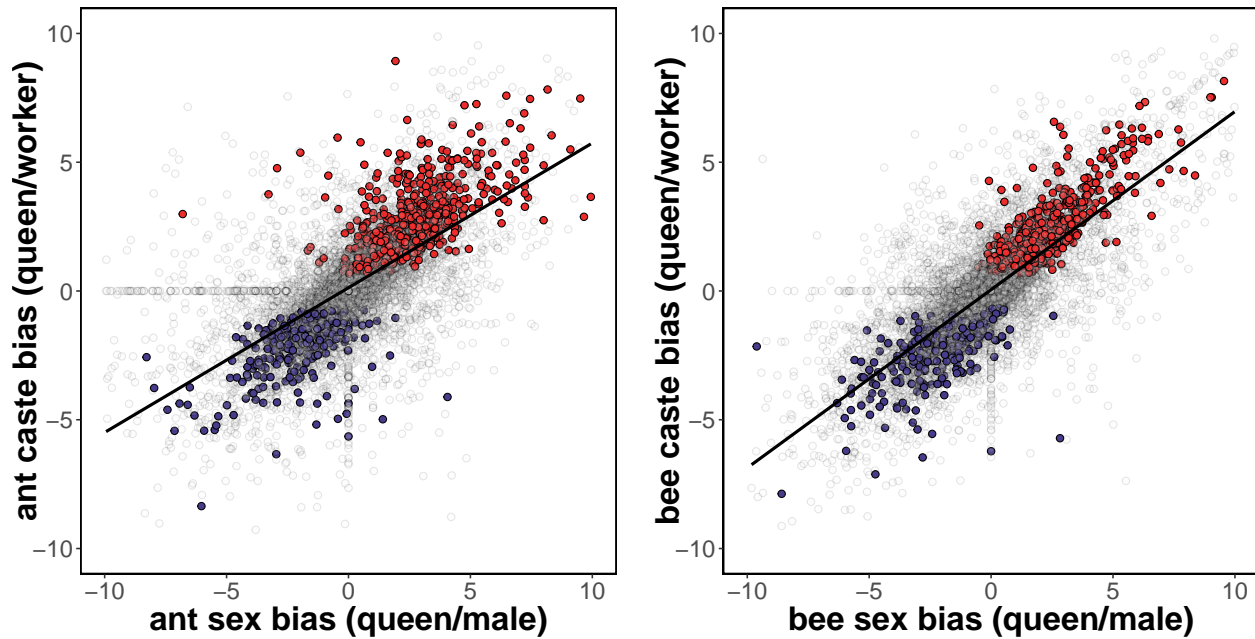
  FC = merge(d1,d2,by = "Gene")
  FC = merge(FC,ogg11,by.x = "Gene",by.y=spec,all.x=TRUE)
  FC$abdDE[is.na(FC$abdDE)] = "non-conserved/non-DE"
  FC$abdDE = factor(FC$abdDE,levels = c("conserved queen","conserved worker","non-conserved/non-DE"))
  FC$alpha = 0.2
  FC$alpha[FC$abdDE!="non-conserved/non-DE"]=0.8

  p1 <- ggplot(FC,aes(x = -FC.x,y = -FC.y))+ #Queen-up will be positive
    geom_point(aes(fill = abdDE,alpha=alpha),pch=21,color="black",size=2)+
    geom_smooth(method="lm",se=FALSE,color="black")+
    scale_fill_manual(values = SexPal[c(1,2,5)],name = "abdominal caste bias")+
    guides(fill = guide_legend(override.aes = list(size=4)))+
    scale_alpha_continuous(guide="none")+
    main_theme+
    ylab("caste bias (queen/worker)")+
    xlab("sex bias (queen/male)")+
    ylim(-10,10)+xlim(-10,10)+
    theme(axis.title.y=element_text(margin = margin(t=0,l=15,r=-5,b=0)))+
    theme(legend.position="none",
          legend.text = element_text(size=17),
          legend.title = element_text(size=19,face="bold"))
  return(p1)

}

p1 = biasPlot(AsexRes[[3]][[1]],AcasteRes[[3]][[1]],"gene_Mphar")+
  ylab("ant caste bias (queen/worker)")+
  xlab("ant sex bias (queen/male)")
p2 = biasPlot(BsexRes[[3]][[1]],BcasteRes[[3]][[1]],"gene_Amel")+
  ylab("bee caste bias (queen/worker)")+
  xlab("bee sex bias (queen/male)")
grid.arrange(p1,p2,nrow=1)

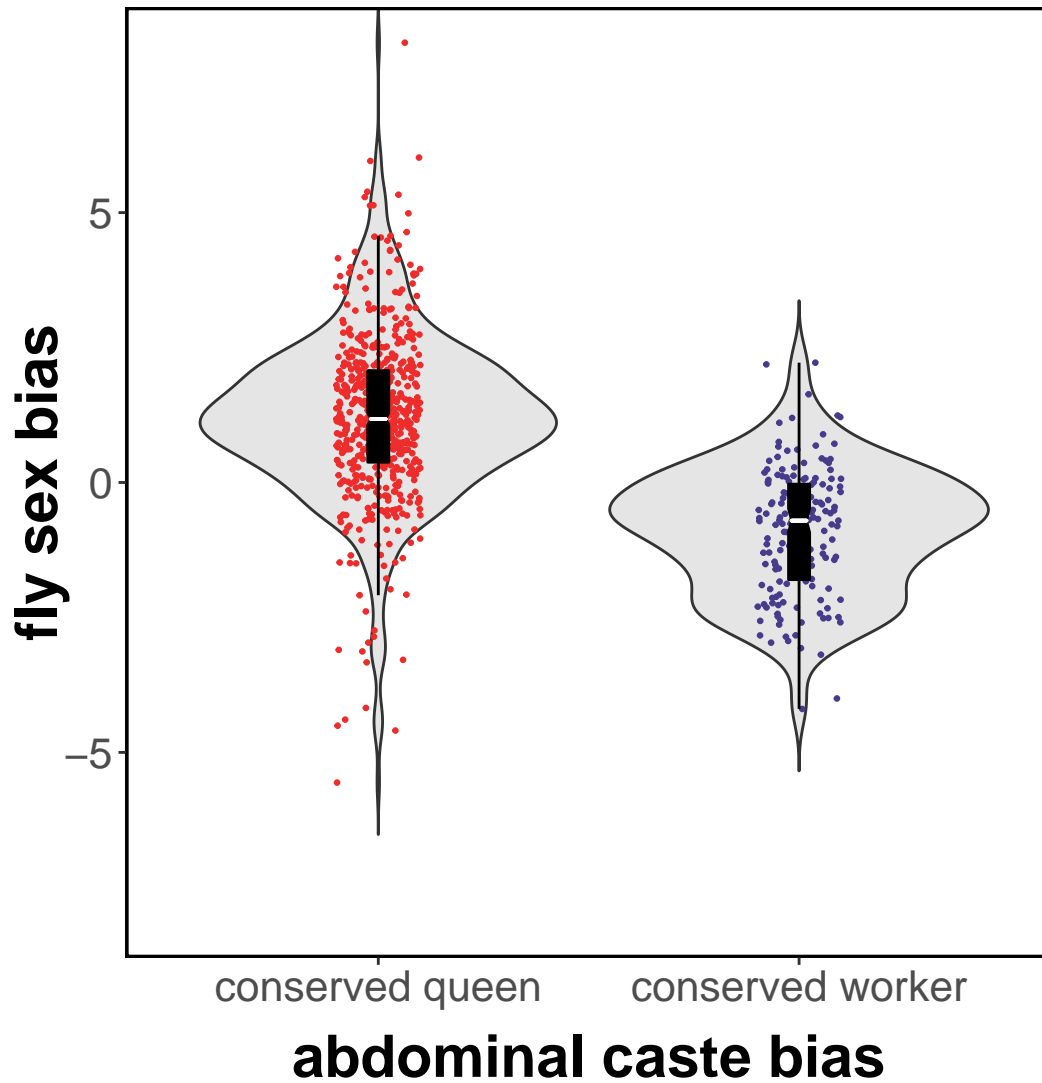
## Warning: Removed 54 rows containing non-finite values (stat_smooth).
## Warning: Removed 54 rows containing missing values (geom_point).
## Warning: Removed 47 rows containing non-finite values (stat_smooth).
## Warning: Removed 47 rows containing missing values (geom_point).
```

Comparison to *D. melanogaster*

```
DmelSC = merge(sexGenes,ogg11,by="gene_Amel")
DmelSC$abdDE = factor(DmelSC$abdDE,levels = c("conserved queen","conserved worker","non-DE/inconsistent"))

p4 <- ggplot(DmelSC[grepl("conserved",DmelSC$abdDE),],aes(x = abdDE,y=-logFC))+
  geom_violin(fill="grey90",trim=FALSE)+
  geom_jitter(width = 0.1,size=0.5,aes(color=abdDE))+
  geom_boxplot(width=0.05,outlier.shape = NA,fill="black",color="black",notch=TRUE,notchwidth = 0.7)+
  plot2theme+
  ylab("fly sex bias")+
  ylim(-10,10)+
  theme(axis.title.y=element_text(margin = margin(t=0,l=15,r=0,b=0)))+
  xlab("abdominal caste bias")+
  scale_fill_manual(values=SexPal)+
  scale_color_manual(values=SexPal)+
  theme(legend.position="none")+
  stat_summary(geom = "crossbar", width=0.035, fatten=0, size=0.7,color="white",
    fun.data = function(x){c(y=median(x), ymin=median(x), ymax=median(x))})+
  coord_cartesian(ylim = c(-8,8))
p4
```



Calculation of overall caste bias

```
#Calculate euclidean distance
euclDist <- function(res){
  cb = apply(res[, -c(1)], 1, function(x) sqrt(sum(x^2))/length(x))
  cb_noAbd = apply(res[, -c(1, ncol(res))], 1, function(x) sqrt(sum(x^2))/length(x))
  cb_noAdult = apply(res[, -c(1, (ncol(res) - 2):ncol(res))], 1, function(x) sqrt(sum(x^2))/length(x))
  cb_larva = apply(res[, -c(1, (ncol(res) - 3):ncol(res))], 1, function(x) sqrt(sum(x^2))/length(x))
  cb_adult = apply(res[, -c(1:(ncol(res) - 3))], 1, function(x) sqrt(sum(x^2))/length(x))
  cb_abd = apply(as.data.frame(res[, -c(1:(ncol(res) - 1))]), 1, function(x) sqrt(sum(x^2))/length(x))
  results = data.frame(Gene = res$Gene, cb=cb, cb_noAbd=cb_noAbd, cb_noAdult=cb_noAdult, cb_larva=cb_larva,
    return(results)
}

#For this analysis, we include all larval stages

antCB = euclDist(antRes_allstage[[1]])
beeCB = euclDist(beeRes_allstage[[1]])
```

```

antSB = euclDist(antSocRes[[1]])
beeSB = euclDist(beeSocRes[[1]])

antSB$type=beeSB$type="behavior"
antCB$type=beeCB$type="caste"
antBias2 = rbind(antCB,antSB)
beeBias2 = rbind(beeCB,beeSB)

#Calculate connectivity
antCor = cor(t(antT[rowSums(antT) > 0,]))^6
beeCor = cor(t(beeT[rowSums(beeT) > 0,]))^6
antConn = data.frame(Gene = rownames(antCor), kTotal = rowSums(antCor))
beeConn = data.frame(Gene = rownames(beeCor), kTotal = rowSums(beeCor))

```

Caste/Behavior bias vs connectivity

```

cbAps = merge(antBias2,Aps,by="Gene")
cbBps = merge(beeBias2,Bps,by="Gene")
cbAps = merge(antConn,cbAps,by="Gene")
cbBps = merge(beeConn,cbBps,by="Gene")

cbBps$kTotal = cbBps$kTotal/max(cbBps$kTotal)
cbAps$kTotal = cbAps$kTotal/max(cbAps$kTotal)
levels(cbAps$psName)[1] = levels(cbBps$psName)[1]= "ancient"

cbPlot <- function(data,type,spec){
  p1C <- ggplot(data[data$type==type,],aes(x = kTotal,y=cb))+
    geom_hex(bins=70)+
    scale_fill_gradient(low = "blue",high="red")+
    plot2theme+
    ylim(0,2.5)+
    geom_smooth(method="lm",size=1.5,se=FALSE,color="black")+
    xlab("scaled network connectivity")+
    ggtitle(spec)+
    ylab(paste("overall",type,"bias",sep = " "))+
    scale_x_log10(breaks = c(0.01,0.1,1))+
    theme(legend.position="none")
  return(p1C)
}

p1 <- cbPlot(cbAps,"caste","ant")
p2 <- cbPlot(cbAps,"behavior","ant")
p3 <- cbPlot(cbBps,"caste","bee")
p4 <- cbPlot(cbBps,"behavior","bee")

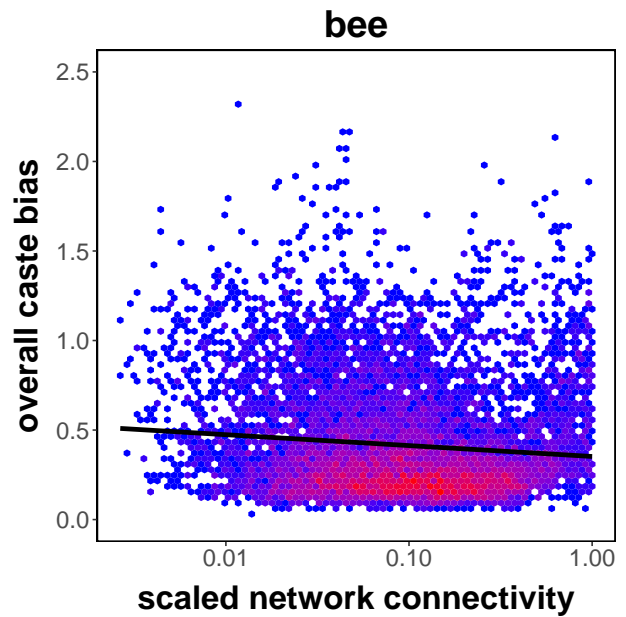
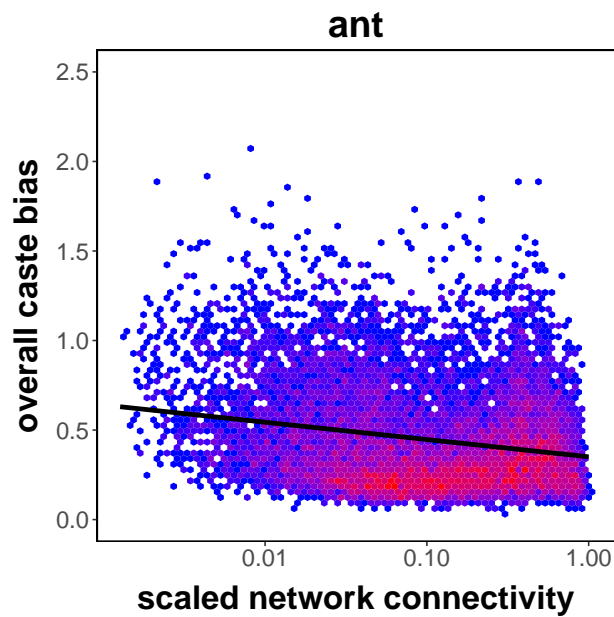
grid.arrange(p1,p3,nrow=1)

```

```

## Warning: package 'hexbin' was built under R version 3.4.3
## Warning: Removed 7 rows containing missing values (geom_hex).
## Warning: Removed 6 rows containing missing values (geom_hex).

```



```
grid.arrange(p2,p4,nrow=1)
```

```
## Warning: Removed 56 rows containing non-finite values (stat_binhex).
```

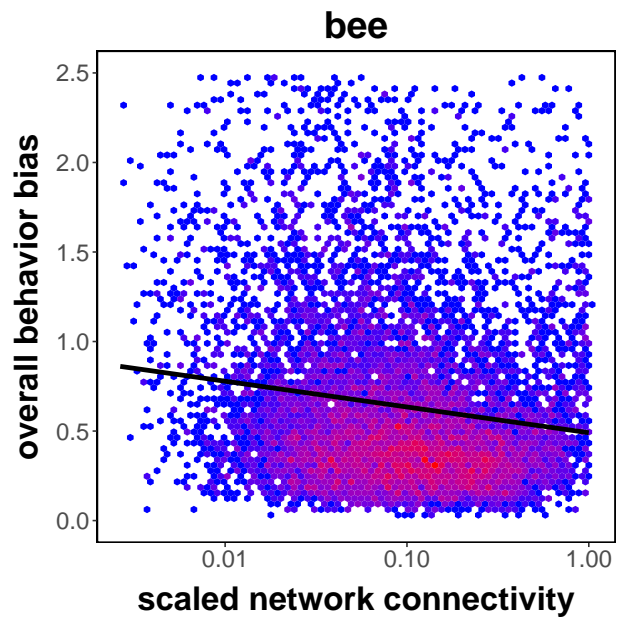
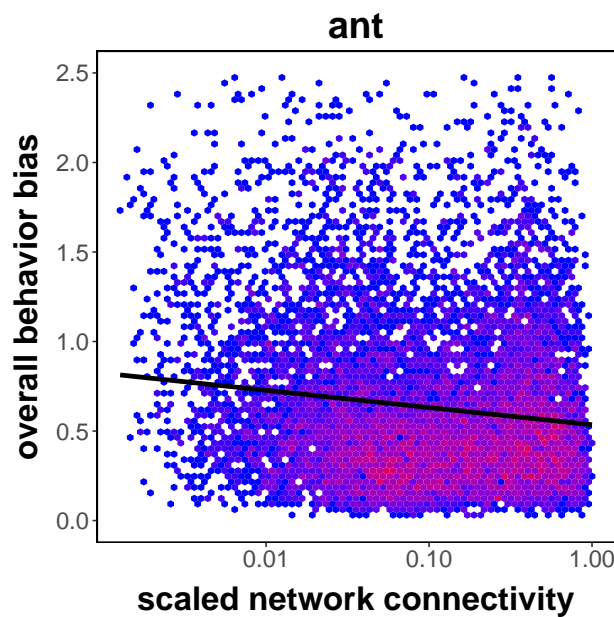
```
## Warning: Removed 56 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 47 rows containing missing values (geom_hex).
```

```
## Warning: Removed 156 rows containing non-finite values (stat_binhex).
```

```
## Warning: Removed 156 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 37 rows containing missing values (geom_hex).
```

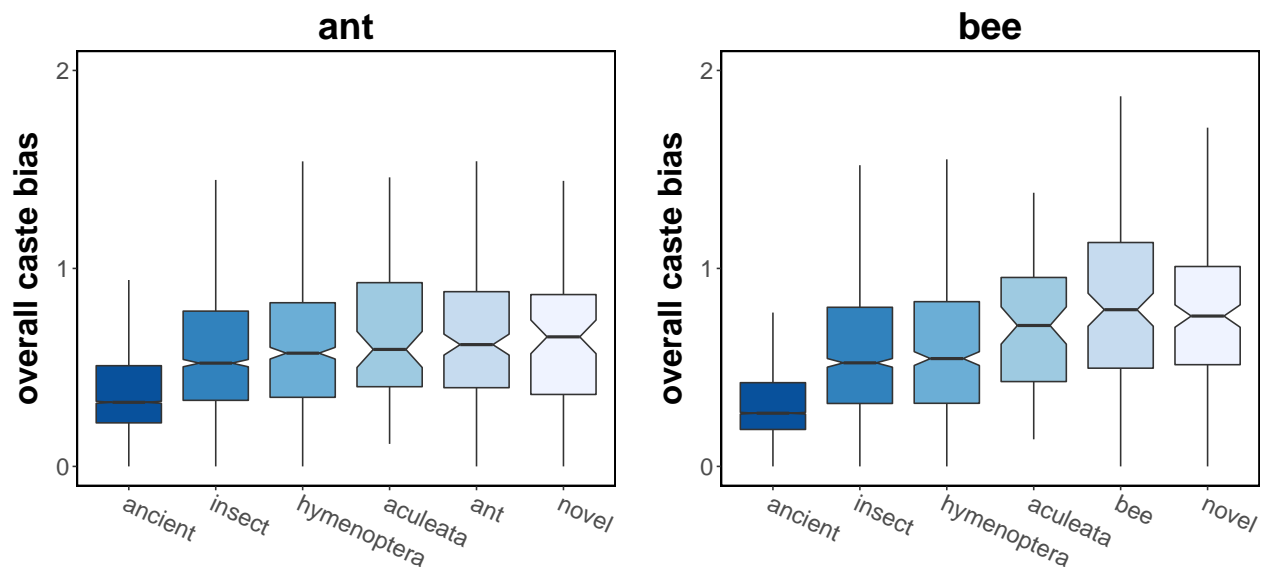


Caste/Behavior bias vs phylostrata

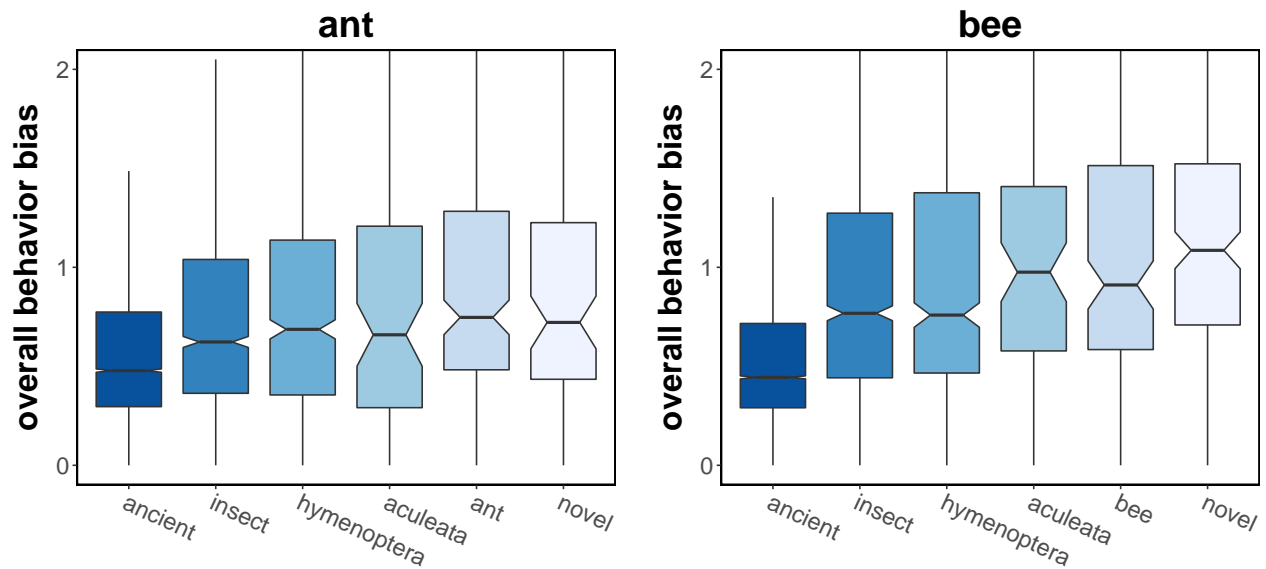
```
phPlot <- function(data,type,spec){
  pA <- ggplot(data[data$type==type & !is.na(data$psName),],aes(x = psName,y=cb,fill=psName))+
    geom_boxplot(notch=TRUE,outlier.shape=NA)+
    plot2theme+
    ggtitle(spec)+
    xlab("")+
    coord_cartesian(ylim=c(0,2))+
    scale_y_continuous(breaks = c(0,1,2,3))+
    scale_fill_manual(values = rev(mypalette2),name = "phylostrata",guide=guide_legend(title.position="top"))+
    ylab(paste("overall",type,"bias",sep = " "))+
    theme(axis.text.x = element_text(angle = -25,hjust=0),
          legend.position="none")
  return(pA)
}

p1 <- phPlot(cbAps,"caste","ant")
p2 <- phPlot(cbAps,"behavior","ant")
p3 <- phPlot(cbBps,"caste","bee")
p4 <- phPlot(cbBps,"behavior","bee")

grid.arrange(p1,p3,nrow=1)
```



```
grid.arrange(p2,p4,nrow=1)
```

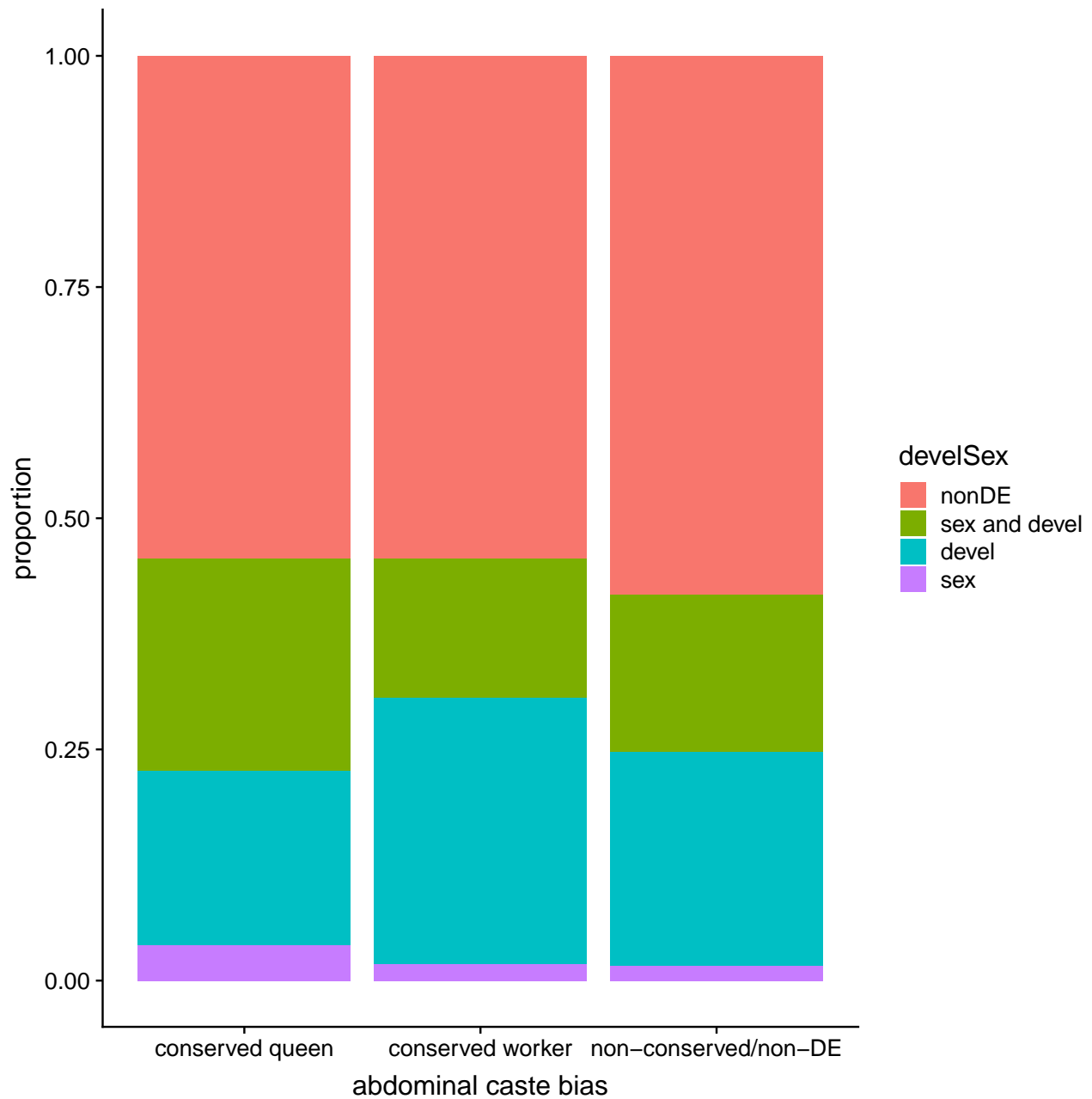


Development section

Comparison to pre-defined sex and development genes

```
DmelSC = merge(sexGenes,ogg11,by="gene_Amel")
key <- read.table("data/DmelKey.txt") #protein-CDS
key2 <- read.table("data/Dmel_CDStoGene_key.txt") #Gene-CDS
devel <- read.table("data/Devel_IDS.txt") #Downloaded from flybase, genes with developmental terms
sex <- read.table("data/Sex_IDS.txt") #Downloaded from flybase, genes with sex terms
key3 = merge(key,key2,by="V1")
DmelSC = merge(DmelSC,key3,by.x="Gene",by.y="V2.y")
DmelSC$develSex = "nonDE"
DmelSC$develSex[DmelSC$V2.x %in% devel$V1] = "devel"
DmelSC$develSex[DmelSC$V2.x %in% sex$V1] = "sex"
DmelSC$develSex[DmelSC$V2.x %in% sex$V1 & DmelSC$V2.x %in% devel$V1] = "sex and devel"

DmelSC$develSex = factor(DmelSC$develSex,levels = c("nonDE","sex and devel","devel","sex"))
p <- ggplot(DmelSC[!grepl("ant",DmelSC$abdDE),],aes(x = abdDE,fill=develSex))+
  geom_bar(stat="count",position="fill")+
  ylab("proportion")+
  xlab("abdominal caste bias")
p
```



-log(P-value) for genes with conserved abdominal differential expression

```
#using DE results
antDevel2$Gene=rownames(antDevel2)
aD = merge(ogg11,antDevel2,by.x="gene_Mphar",by.y="Gene")

p1 <- ggplot(aD[!grepl("ant",aD$abdDE),],aes(x = abdDE,y=-log(FDR)))+
  geom_boxplot(notch = T)+
  scale_y_log10()+
  theme(axis.text.x = element_text(angle=90))+
  ggtitle("devel measured in ant")
```

```

beeDevel2$Gene=rownames(beeDevel2)
bD = merge(ogg11,beeDevel2,by.x="gene_Amel",by.y="Gene")

p2 <- ggplot(bD[!grepl("ant",bD$abdDE),],aes(x = abdDE,y=-log(FDR)))+
  geom_boxplot(notch = T)+
  scale_y_log10()+
  theme(axis.text.x = element_text(angle=90))+
  ggtitle("devel measured in bee")

grid.arrange(p1,p2,nrow=1)

```

```

## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 14 rows containing non-finite values (stat_boxplot).
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 12 rows containing non-finite values (stat_boxplot).

```

