Mphar_Amel

Loading libraries, themes

Loading previous results

```
setwd("~/GitHub/devnetwork/")
load("results/DEtests.RData")
load("results/collectedPhylo.RData")
antPlaid <- read.csv("results/antPlaidGenes.csv")
beePlaid <- read.csv("results/beePlaidGenes.csv")
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")

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,]


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$sample)]=7
  factors$tissue="larva"
  factors$tissue[grepl("P",factors$sample)]="pupa"
  factors$tissue[grepl("_E",factors$sample)]="egg"
  factors$tissue[grepl("G\\.",factors$sample)]="gaster"
  factors$tissue[grepl("H\\.",factors$sample)]="head"
  factors$tissue[grepl("M\\.",factors$sample)]="mesosoma"
  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$tissue,levels = c("egg","larva","pupa","head","mesosoma","gaster"))
  factors$NF = factor(factors$NF,levels = c("nurse","forager")) #Make nurse genes down-regulated becaus
  return(factors)
}

factorA <- genFactor(antT)
factorB <- genFactor(beeT)

TGmap <- read.table("phylostratigraphy/out/TGmap_Amel.txt")
TNmap <- as.data.frame(fread("data/AmelTranName.txt",sep="~",header=FALSE))

AmelName <- merge(TGmap,TNmap,by.x = "V2",by.y = "V1")[,c(2,3)]
```

```
## Warning in merge.data.frame(TGmap, TNmap, by.x = "V2", by.y = "V1"): column
## name 'V2' is duplicated in the result
```

```
colnames(AmelName) = c("Gene","GeneName")
AmelName$GeneName = gsub(" isoform X[0-9]","",AmelName$GeneName)
aName = AmelName[!duplicated(AmelName$Gene),]
```

0.1 Number of times a gene is queen- and worker-biased across development

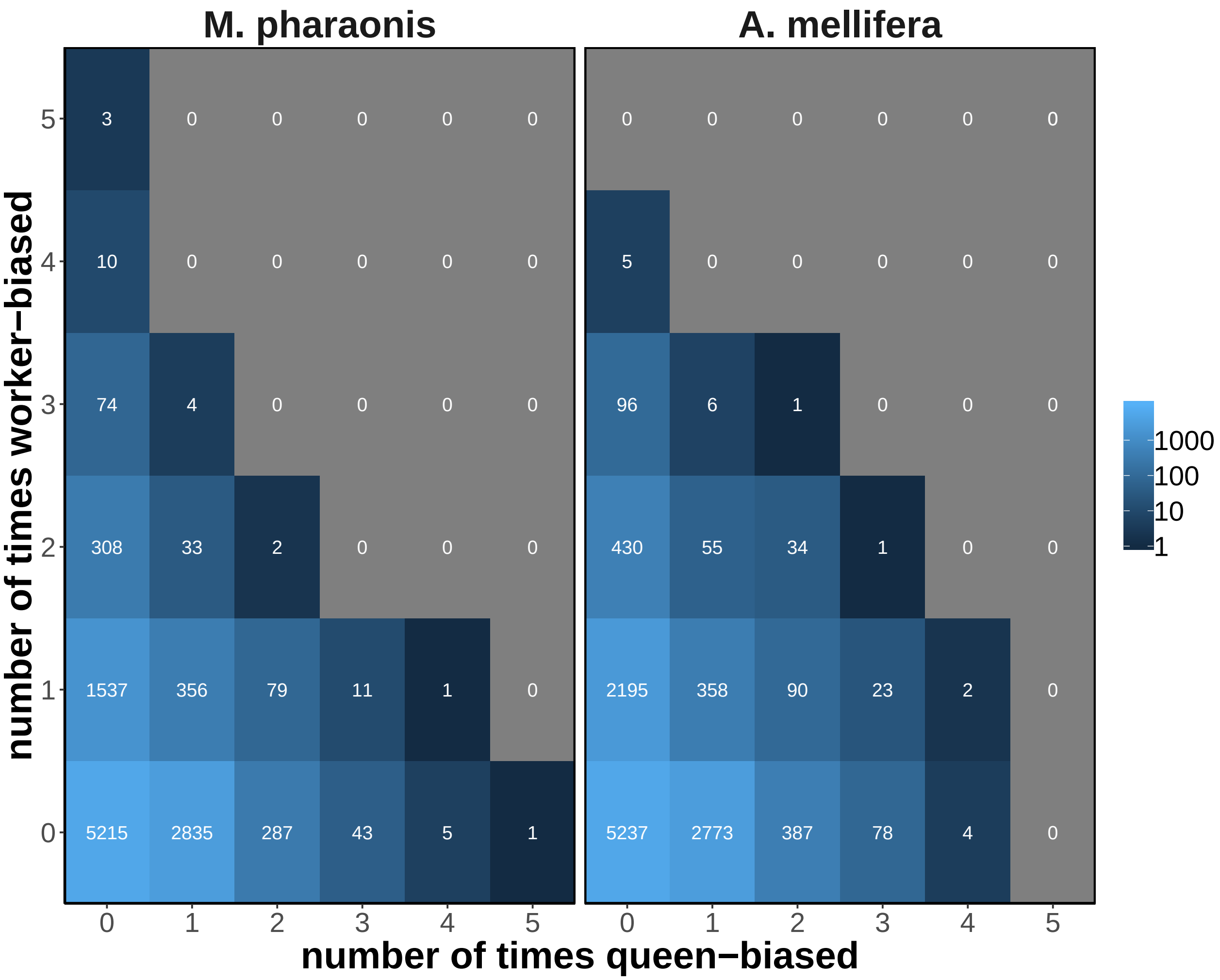
```
#Summarize number of times DE
sumDE <- function(dfDE,type1,type2){
  dfDE$numQueen = apply(dfDE[,c(2:ncol(dfDE))],1,function(x) sum(x == type1))
  dfDE$numWorker = apply(dfDE[,c(2:ncol(dfDE))],1,function(x) sum(x == type2))
  d = table(dfDE$numQueen,dfDE$numWorker)
  m = melt(d)
  colnames(m)[c(1,2)] = c(type1,type2)
  return(m)
}


m1 = sumDE(antRes[[2]],"queen","worker")
m2 = sumDE(beeRes[[2]],"queen","worker")

#Add these since there are no genes DE all five times in Apis
m2E = t(sapply(seq(0,5),function(i) c(queen=i,worker=5,value=0)))
m2Eb = t(sapply(seq(0,5),function(i) c(queen=5,worker=i,value=0)))
m2E = rbind(m2Eb,m2E)
m2 = rbind(m2,m2E)

m1$species = "M. pharaonis"
```

```r
m2$species = "A. mellifera"
mA = rbind(m1,m2)
mA$species = factor(mA$species,levels= c("M. pharaonis","A. mellifera"))


#Create heatmap of differential expression (number of times DE for queens and workers)
p <- ggplot(mA,aes(x=queen,y=worker))+
  geom_tile(aes(fill = value))+
  facet_grid(. ~ species)+
  scale_fill_gradient(name = "number of genes",trans = "log",
                      breaks = c(1,10,100,1000),
                      limits = c(1,10000),
                      labels = c(1,10,100,1000))+
  geom_text(aes(x = queen,y = worker,label = value),color="white")+
  main_theme+
  scale_y_continuous(name = "number of times worker-biased",
                     breaks  = seq(0,5),
                     expand = c(0,0))+
  scale_x_continuous(name = "number of times queen-biased",
                     breaks = seq(0,5),
                     expand = c(0,0))+
  theme(legend.position = "right",
        axis.line=element_line(color="black"),
        axis.text = element_text(size=16),
        axis.title = element_text(size = 22,face="bold"),
        strip.text = element_text(size=22,face="bold"),
        legend.title = element_blank(),
        strip.background = element_rect(color=NA,fill=NA),
        plot.title = element_text(hjust = 0.5,size=25,face = "bold"),
        panel.border = element_rect(size = 1, color = "black",fill = NA))
ggsave(p,file = "figures/p2.pdf",height=8,width=10)
```

```
## Warning: Transformation introduced infinite values in discrete y-axis
```

0.1 Number of times a gene is queen- and worker-biased across development

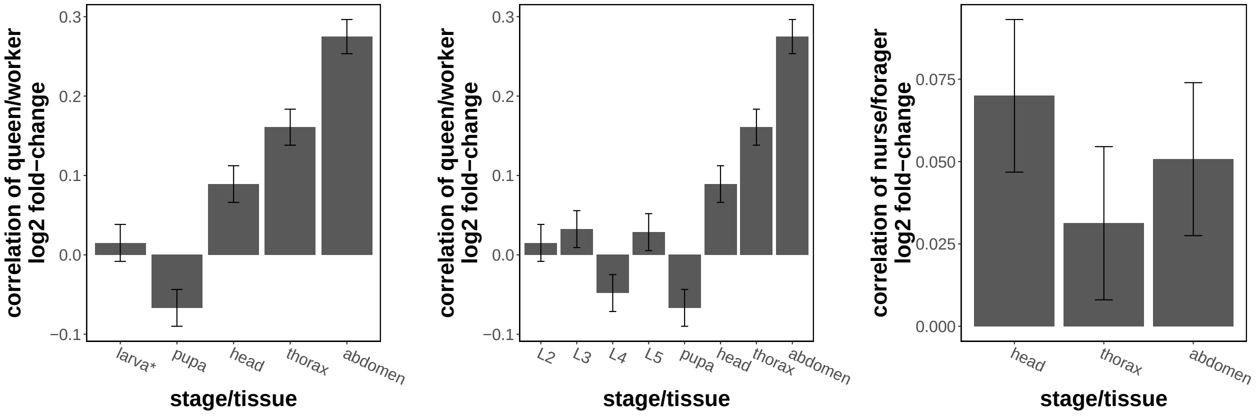
```
## Warning: Transformation introduced infinite values in discrete y-axis
```



0.2 Correlation of log fold-change across stages

```
#Correlation of log fold change across development
lfcCor <- function(antD,beeD){
  nStage = ncol(antD) - 1
  antD <- merge(antD,ACUogg,by.x = "Gene",by.y = "gene_Mphar")
  beeD <- merge(beeD,ACUogg,by.x = "Gene",by.y = "gene_Amel")
  antD = antD[antD$OGG %in% beeD$OGG,]
  beeD = beeD[beeD$OGG %in% antD$OGG,]
  antD = antD[order(antD$OGG),]
  beeD = beeD[order(beeD$OGG),]
  d = data.frame(stage = colnames(antD)[c(2:(nStage+1))])
  dAbs = data.frame(stage = colnames(antD)[c(2:(nStage+1))])
  for (i in 1:nStage){
    t = cor.test(antD[,i+1],beeD[,i+1])
    d[i,2] = t$estimate
    d[i,3] = t$conf.int[1]
    d[i,4] = t$conf.int[2]
    t = cor.test(abs(antD[,i+1]),abs(beeD[,i+1]))
    dAbs[i,2] = t$estimate
    dAbs[i,3] = t$conf.int[1]
    dAbs[i,4] = t$conf.int[2]
  }
  colnames(d) = colnames(dAbs) = c("Stage","cor","c1","c2")
  return(list(d,dAbs))
}
CasteCor <- lfcCor(antRes[[1]],beeRes[[1]])
CasteCor_allStage <- lfcCor(antRes_allstage[[1]],beeRes_allstage[[1]])
BehavCor <- lfcCor(antSocRes[[1]],beeSocRes[[1]])

CasteCor[[1]]$Stage = as.character(CasteCor[[1]]$Stage)
CasteCor[[1]]$Stage[1] = "larva*"
CasteCor_allStage[[1]]$Stage = c("L2","L3","L4","L5","pupa","head","thorax","abdomen")
BehavCor[[1]]$Stage = c("head","thorax","abdomen")
CasteCor[[1]]$Stage = factor(CasteCor[[1]]$Stage,levels = CasteCor[[1]]$Stage)
CasteCor_allStage[[1]]$Stage = factor(CasteCor_allStage[[1]]$Stage,levels = CasteCor_allStage[[1]]$Stag
BehavCor[[1]]$Stage = factor(BehavCor[[1]]$Stage,levels = BehavCor[[1]]$Stage)


pl <- lapply(list(CasteCor[[1]],CasteCor_allStage[[1]],BehavCor[[1]]), function(x){
  ggplot(x,aes(x = Stage,y=cor))+
    geom_bar(stat="identity")+main_theme+
    xlab("stage/tissue")+
    geom_errorbar(aes(ymin=c1,ymax=c2),width=0.2)+
    ylab("correlation of queen/worker\nlog2 fold-change")+
    theme(plot.margin=unit(c(0.5,1.5,0.5,0.5),"cm"),
          axis.text.x = element_text(hjust=0,angle=-25))
})
pl[[3]] = pl[[3]] + ylab("correlation of nurse/forager\nlog2 fold-change")
```

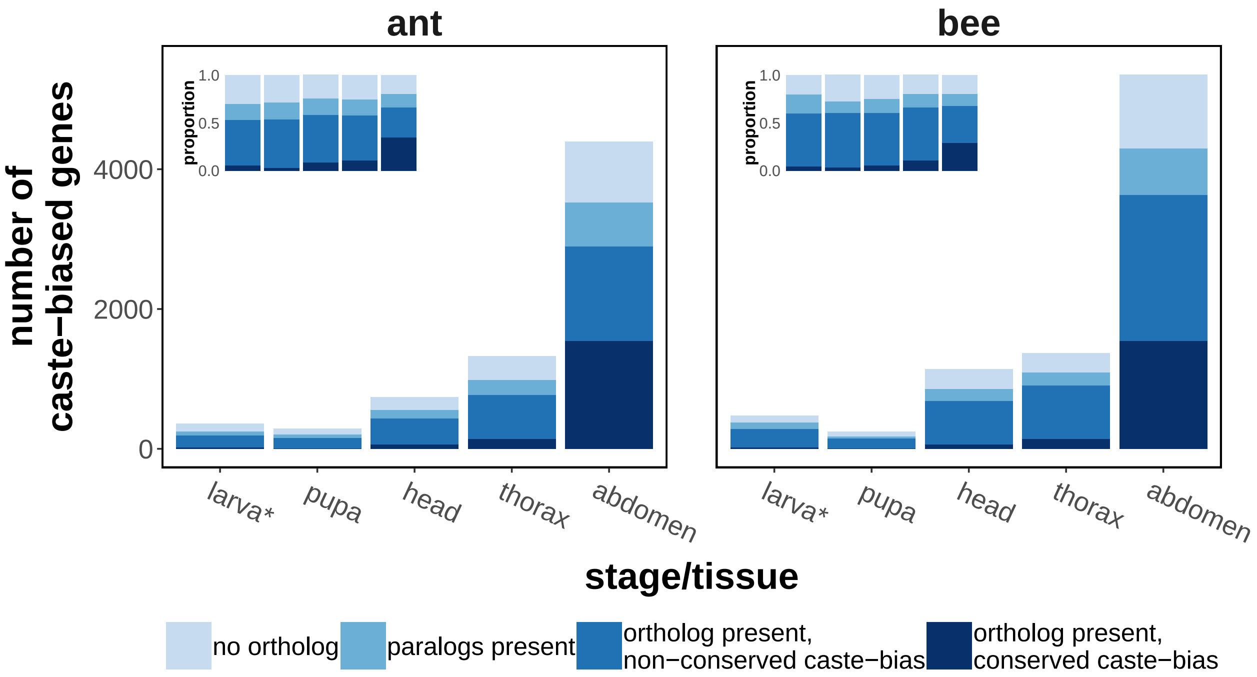
0.2 Correlation of log fold-change across stages



## 1.1 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,\nnon-conserved caste-bi
  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)
```

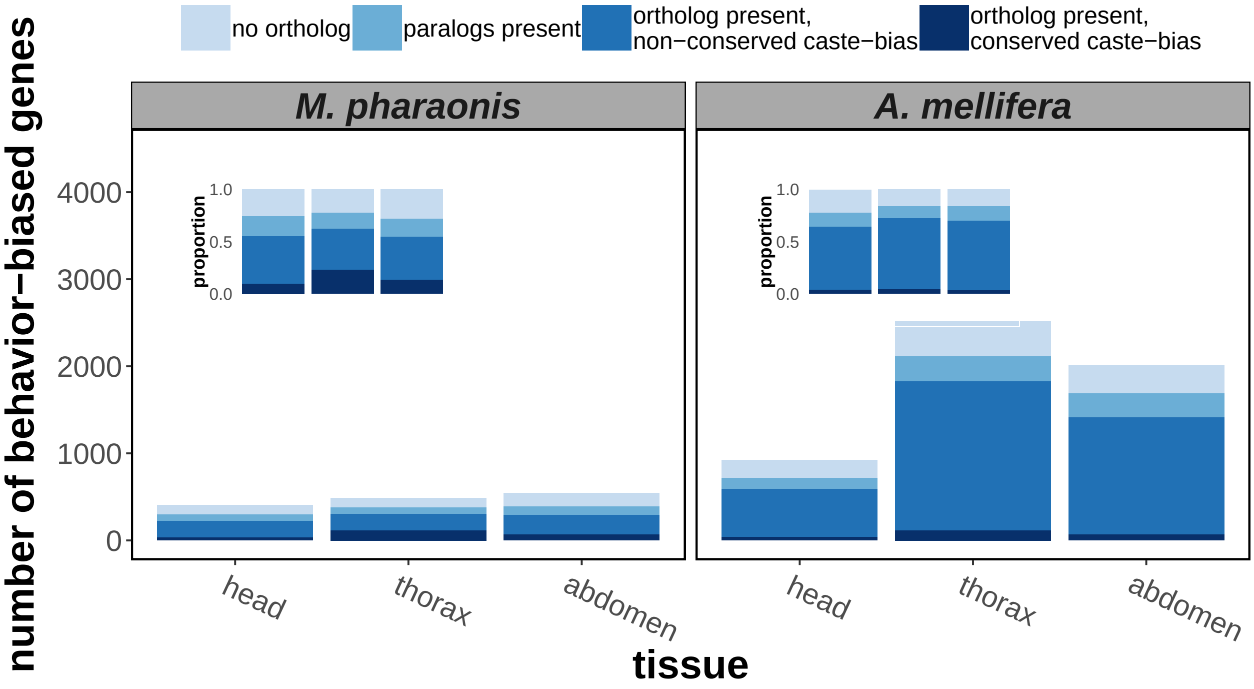
1.1 Caste bias across development



1.2 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)
```

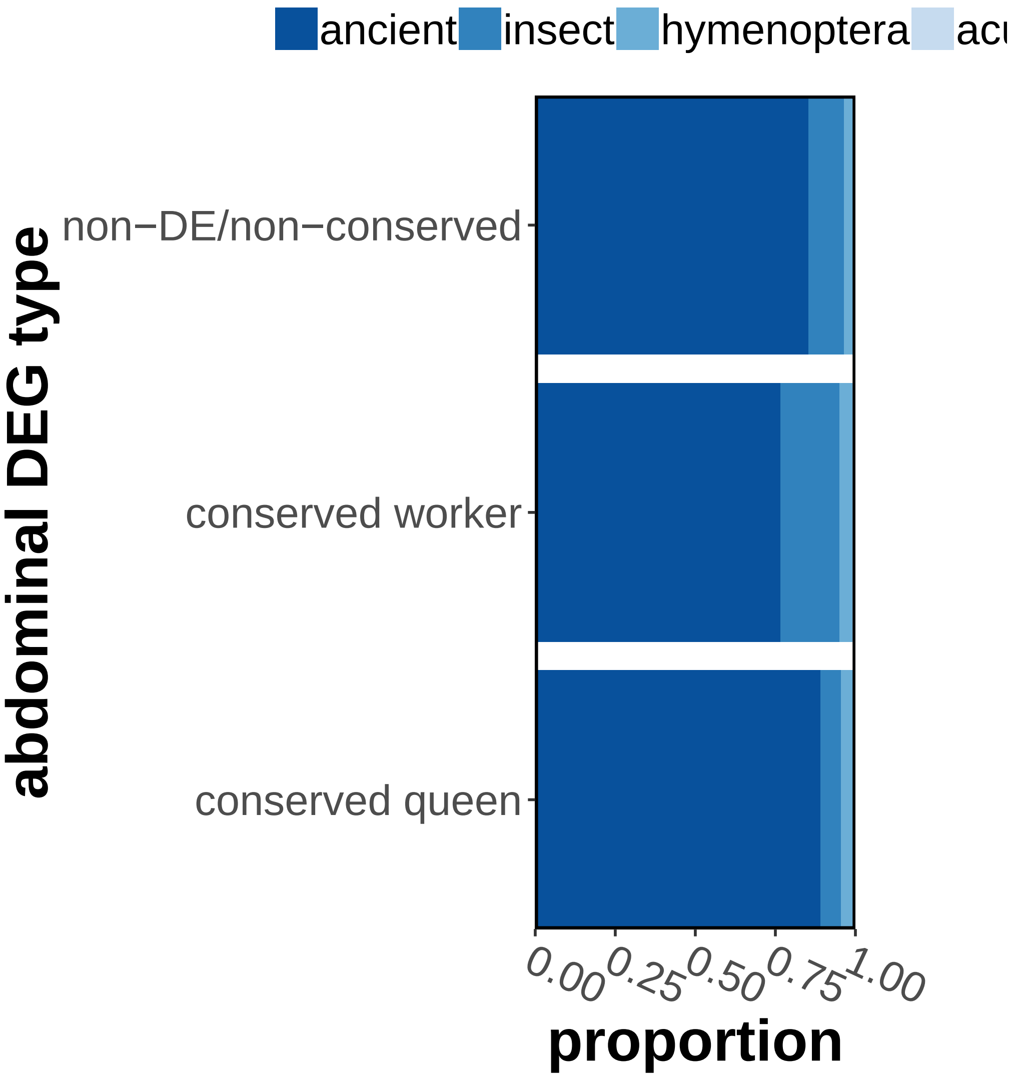
1.2 nurse/forager bias



1.3 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"))
```

1.3 Estimated evolutionary age of genes with abdominal caste bias



Phylostrata across all comparisons

```
ant_PS = merge(antRes[[1]],Aps,by="Gene")
aPM = melt(ant_PS,id.vars = c("Gene","OGGacu","ps","psName","ODBgene"))
aPM$species = "ant"
bee_PS = merge(beeRes[[1]],Bps,by="Gene")
bPM = melt(bee_PS,id.vars = c("Gene","OGGacu","ps","psName","ODBgene"))
bPM$species = "bee"
PM = rbind(aPM,bPM)
```

```
PM$psName = factor(PM$psName,levels = levels(PM$psName)[c(1:5,7,6)])
p <- ggplot(PM[!is.na(PM$psName),],aes(x = variable,y=-value,fill=psName))+
  geom_boxplot(outlier.shape = NA,notch=T)+
  facet_wrap(. ~ species)+
  ylab("log2 fold-change (queen/worker)")+
  coord_cartesian(ylim=c(-5,5))+
  theme(legend.position="bottom")+
  xlab("stage/tissue")+
  geom_hline(yintercept = 0,linetype="dashed")
ggsave(p,file = "figures/p5.pdf",height=6,width=12)

ant_PS = merge(antSocRes[[1]],Aps,by="Gene")
aPM = melt(ant_PS,id.vars = c("Gene","OGGacu","ps","psName","ODBgene"))
aPM$species = "ant"
bee_PS = merge(beeSocRes[[1]],Bps,by="Gene")
bPM = melt(bee_PS,id.vars = c("Gene","OGGacu","ps","psName","ODBgene"))
bPM$species = "bee"
PM = rbind(aPM,bPM)
PM$psName = factor(PM$psName,levels = levels(PM$psName)[c(1:5,7,6)])
p <- ggplot(PM[!is.na(PM$psName),],aes(x = variable,y=-value,fill=psName))+
  geom_boxplot(outlier.shape = NA,notch=T)+
  facet_wrap(. ~ species)+
  ylab("log2 fold-change (nurse/forager)")+
  coord_cartesian(ylim=c(-5,5))+
  theme(legend.position="bottom")+
  xlab("tissue")+
  geom_hline(yintercept = 0,linetype="dashed")
ggsave(p,file = "figures/p18.pdf",height=6,width=12)
```

```
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```

2.1 Bicluster associated with queen abdominal expression

-Previously performed plaid clustering and extracted genes commonly found in a queen abdomen-specific bicluster.

```
load("results/PlaidResults.RData")
freq_set <- function(data,maxMod,factor,tpm,type){
  res <- lapply(data,function(x){
    checkMod = seq(1:nrow(x@NumberxCol))[rowSums(x@NumberxCol) <= maxMod]
    r = list()
    modI = 0
    for (mod in checkMod){
      qp = x@NumberxCol
      samp = factor$sample[x@NumberxCol[mod,]]
      if (sum(grepl(type,samp))==3){
        modI = modI+1
        genes = rownames(tpm)[x@RowxNumber[,mod]]
        r[[modI]]=list(samples=samp,genes=genes)
      }
    }
    if (length(r) > 1){
      rnew = r[[1]]
      rnew$genes = unique(c(r[[2]]$genes,r[[1]]$genes)) #combine genes from queen abdomen module
      r = rnew
    }
    return(r)
  })
  return(res)
}

#Calculate number of significant biclusters for a given set
num_set <- function(set){
  nums = lapply(names(set),function(x){
    length(set[[x]][lapply(set[[x]],length) > 0])
  })
  numDF = data.frame(Type = names(set),Freq=unlist(nums))
}

commonGenes <- function(QGres,tpm){
  df = data.frame(Gene=rownames(tpm),KeepNum=0)
  num = 0
  for (res in QGres){
    if (length(res) == 1){
      num = num+1
      df$KeepNum[df$Gene %in% res[[1]]$genes] = df$KeepNum[df$Gene %in% res[[1]]$genes] + 1
    }
  }
  df$KeepNum = df$KeepNum/num
  return(df)
}

samp_types = sapply(factorA$sample,function(x) gsub(".*_","",x) %>% gsub("\\.[0-9]","",.)) %>%
  unique()

#Output significant biclusters for each sample type
type_bicAmel <- lapply(samp_types,function(x) freq_set(beePl,6,factorB,beeT,x))
type_bicMphar <- lapply(samp_types,function(x) freq_set(antPl,6,factorA,antT,x))
names(type_bicAmel) = names(type_bicMphar) = samp_types
typeNum_Amel = num_set(type_bicAmel)
typeNum_Mphar = num_set(type_bicMphar)

antQG <- type_bicMphar[["AQG"]]
beeQG <- type_bicAmel[["AQG"]]
antQG = antQG[lapply(antQG,length) > 0]
beeQG = beeQG[lapply(beeQG,length) > 0]

aG <- commonGenes(antQG,antT)
bG <- commonGenes(beeQG,beeT)

pdf("figures/p6.pdf")
hist(aG$KeepNum)
dev.off()
```

```
## pdf
##   2
```

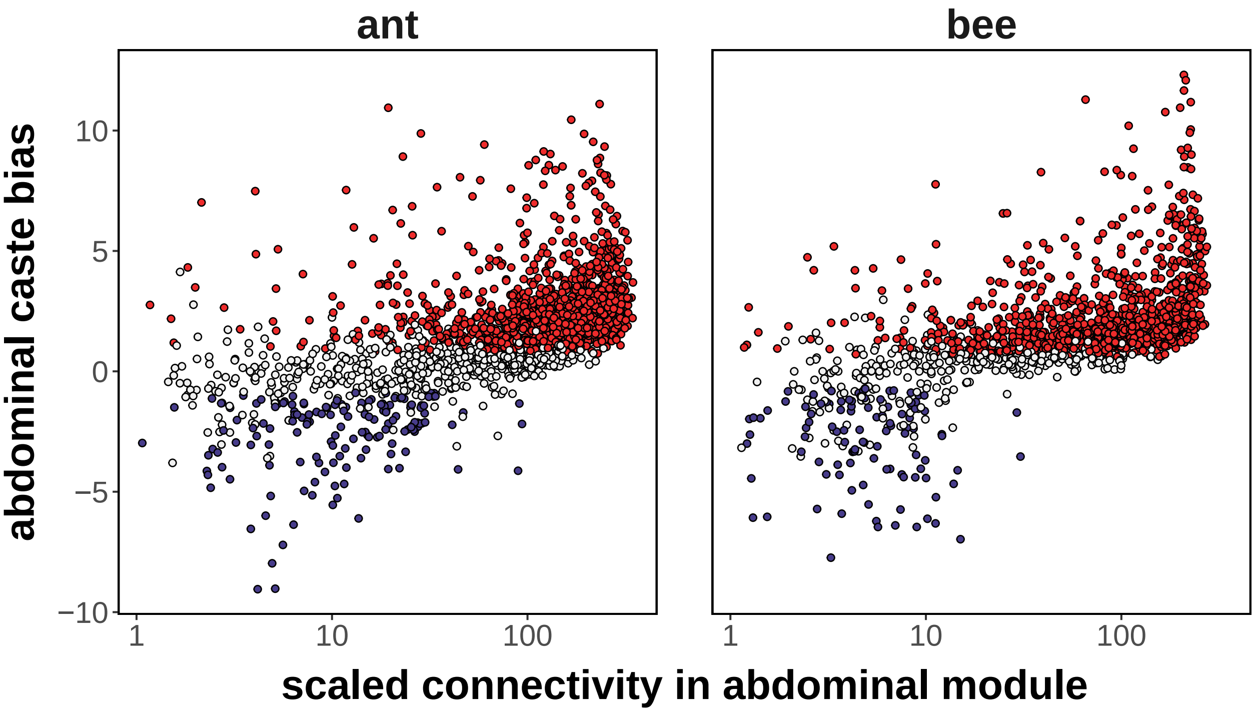
```
pdf("figures/p7.pdf")
hist(bG$KeepNum)
dev.off()
```

```
## pdf
##   2
```

```
#Find out what the genes are
antGrC = merge(antPlaid,Aps,by="Gene")
beeGrC = merge(beePlaid,Bps,by="Gene")
antGrC$species = "ant"
beeGrC$species = "bee"

#Add -log(p)
antGrC = merge(antGrC,antRes[[3]],by="Gene")
beeGrC = merge(beeGrC,beeRes[[3]],by="Gene")

#Make general plot
allC = rbind(antGrC,beeGrC)
allC$DEcat = factor(allC$DEcat,levels = c("queen","worker","non-DE"))

#Note -abdomen because we want queen genes to be upregulated
p1 <- ggplot(allC,aes(x = conn,y=-abdomen.x))+
  geom_point(aes(fill = DEcat),pch=21,color="black")+
  scale_fill_manual(values = SexPal[c(1,2,5)],name="DEG type")+
  scale_alpha_continuous(guide="none")+
  ylab("abdominal caste bias")+
  xlab("scaled connectivity in abdominal module")+
  plot2theme+
  facet_grid(. ~ species)+
  theme(panel.spacing = unit(2, "lines"))+
  guides(fill = guide_legend(override.aes = list(size=4)))+
  scale_x_log10()+
  theme(legend.position ="none",
        legend.text = element_text(size=15),
        strip.text = element_text(size=22,face="bold"),
        axis.title.y=element_text(margin=margin(t=0,l=0,r=10,b=0)),
        legend.title = element_blank(),
        plot.margin = unit(rep(1,4),"cm"),
        strip.background = element_rect(color=NA,fill=NA))


ggsave(p1,file = "figures/p8.pdf",height=6,width=10)

p2 <- ggplot(allC,aes(x = conn,y=-log(abdomen.y)))+
  geom_point(aes(fill = DEcat),pch=21,color="black")+
  scale_fill_manual(values = SexPal[c(1,2,5)],name="DEG type")+
  scale_alpha_continuous(guide="none")+
  ylab("-log(P) of queen/worker\ndifferential expression")+
  xlab("scaled connectivity in abdominal module")+
  plot2theme+
  facet_grid(. ~ species)+
  theme(panel.spacing = unit(2, "lines"))+
  guides(fill = guide_legend(override.aes = list(size=4)))+
  scale_x_log10()+
  theme(legend.position ="none",
        legend.text = element_text(size=15),
        strip.text = element_text(size=22,face="bold"),
        axis.title.y=element_text(margin=margin(t=0,l=0,r=10,b=0)),
        legend.title = element_blank(),
        plot.margin = unit(rep(1,4),"cm"),
        strip.background = element_rect(color=NA,fill=NA))


ggsave(p2,file = "figures/p9.pdf",height=6,width=10)
```

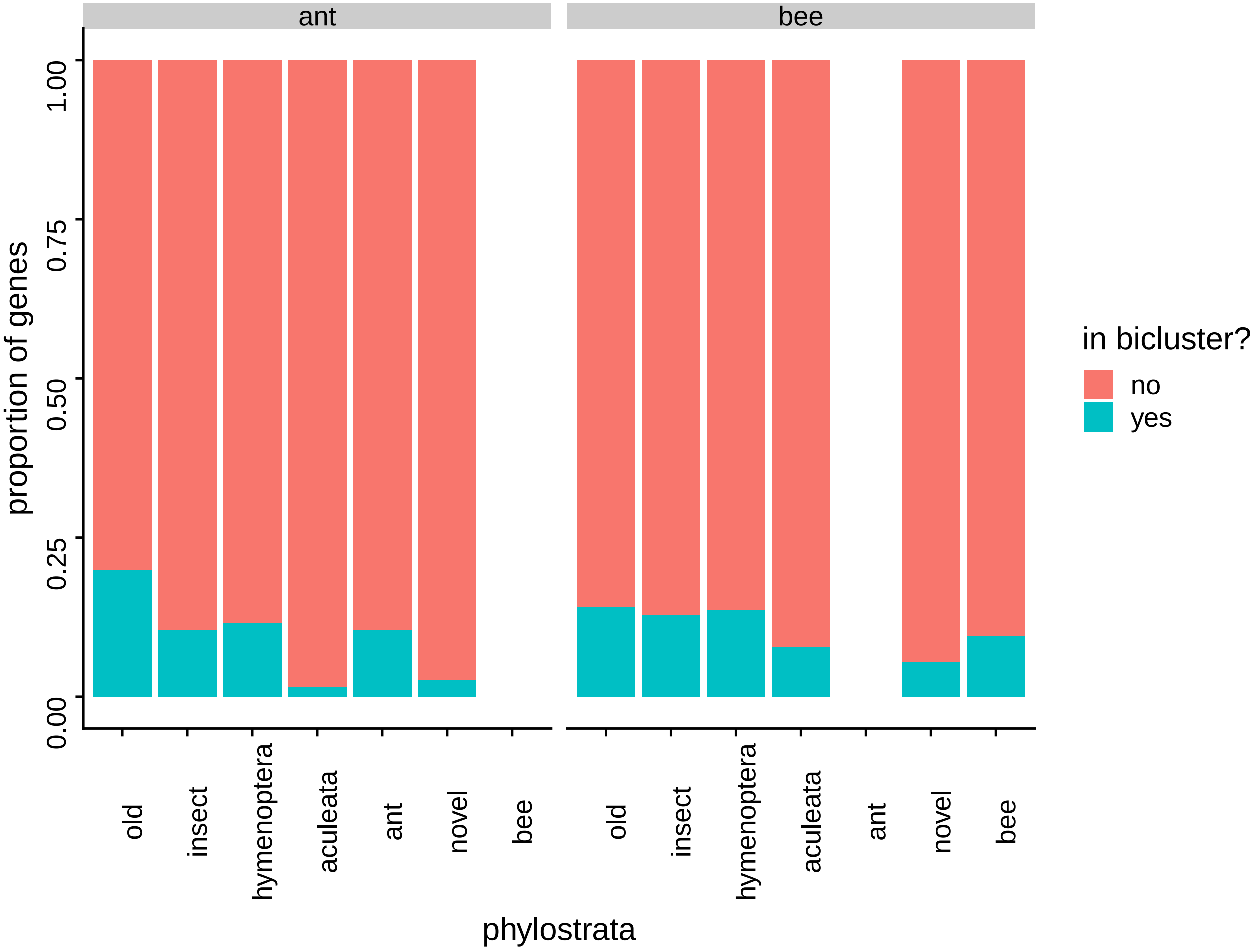
Highly connected bicluster genes are queen-biased

```
p1
```



Age of genes in bicluster

```
Aps$bic = Bps$bic = "no"
Aps$bic[Aps$Gene %in% antGrC$Gene] = "yes"
Bps$bic[Bps$Gene %in% beeGrC$Gene] = "yes"
Aps$species = "ant"
Bps$species = "bee"
psBic = rbind(Aps,Bps)
p <- ggplot(psBic[!is.na(psBic$psName),],aes(x = psName,fill=bic))+
  geom_bar(stat="count",position="fill")+
  facet_wrap(. ~ species)+
  xlab("phylostrata")+
  ylab("proportion of genes")+
  theme(axis.text = element_text(angle=90))+
  scale_fill_discrete(name = "in bicluster?")
p
```



```
ggsave(p,file = "figures/p10.pdf",height=6,width=8)
```

2.2 Specific genes within the bicluster

- holding off plotting specific genes for now

```
beeAnn <- read.csv("data/Amel_Gene_Names.csv")
ann <- read.csv("data/MpharAnn.csv")

#Pick genes with connectivity in the top 90% that are four-fold upregulated in queens
antGenes = antGrC$Gene[antGrC$conn > quantile(antGrC$conn,0.9) &
                          antGrC$abdomen < -2]
beeGenes = beeGrC$Gene[beeGrC$conn > quantile(beeGrC$conn,0.9) &
                          beeGrC$abdomen < -2]
annSwiss = ann$SwissProt[ann$Gene %in% antGenes]
beeSwiss = beeAnn$GeneName[beeAnn$Gene %in% beeGenes]
```

ant Genes

```
annSwiss[!grepl("ncharacterized",annSwiss) & annSwiss!="-"]
```

```
## factor(0)
## 7725 Levels:  Pyruvate dehydrogenase  acetyl-transferring  -phosphatase 1  mitochondrial ...
```

bee Genes

```
beeSwiss[!grepl("ncharacterized",beeSwiss) & beeSwiss!="-"]
```

```
## factor(0)
## 9976 Levels: 18-wheeler precursor ... zinc finger protein 629-like
```

3.1 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)
```

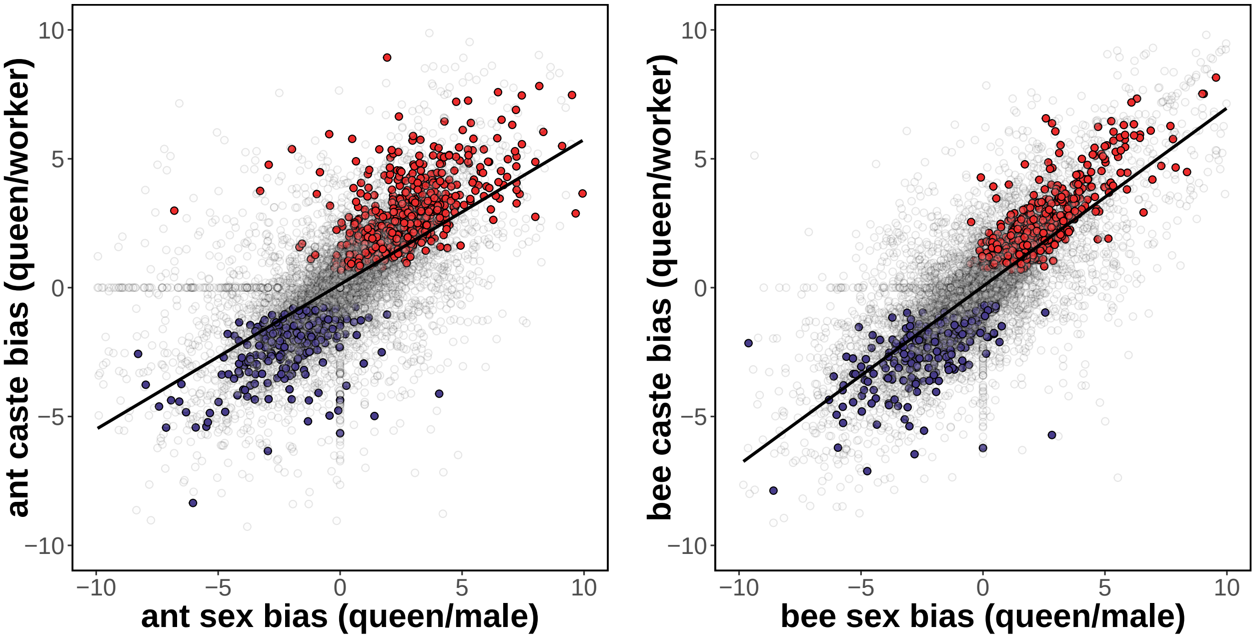
Identify genes with conserved bias for plotting

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

3.1 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).
```



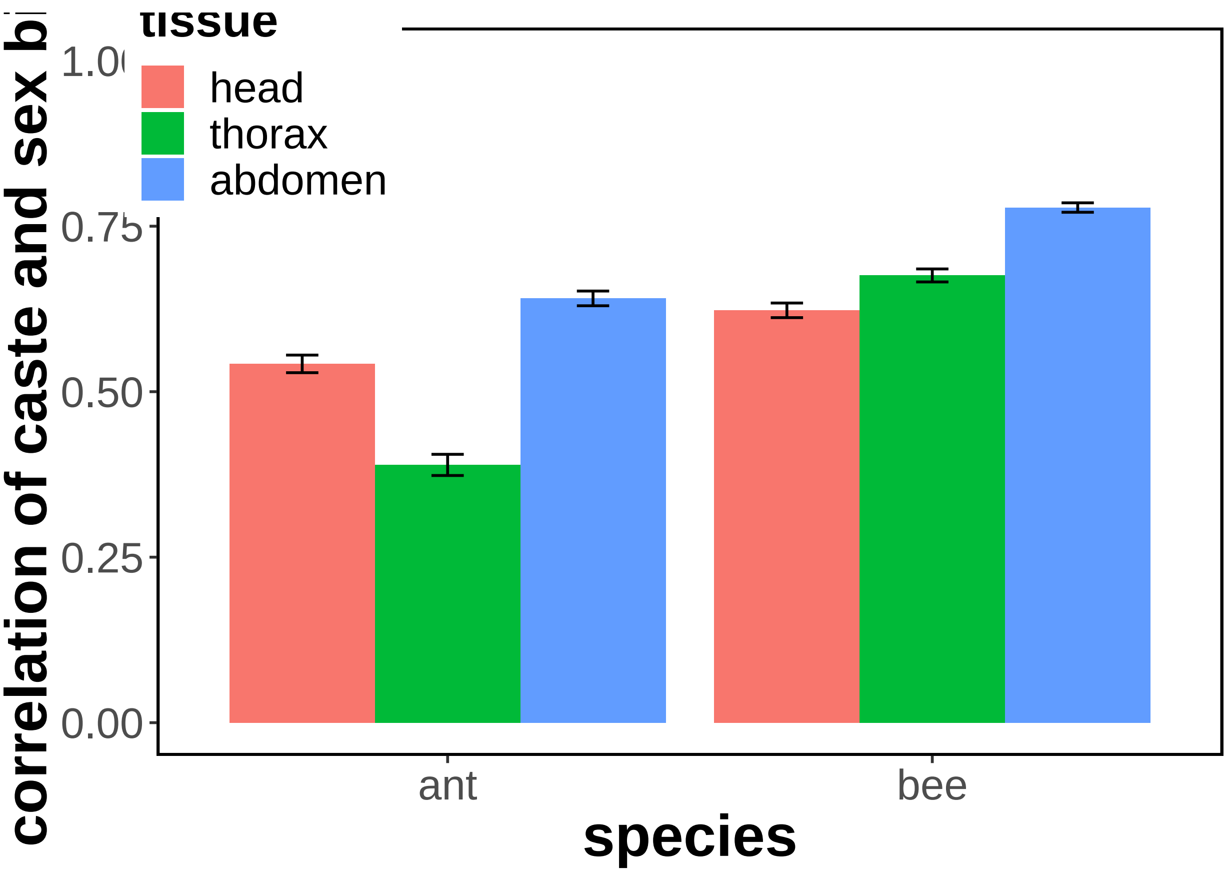
```
p =arrangeGrob(p1,p2,nrow=1)
```

```
## 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).
```

```
ggsave(p,file = "figures/p11.pdf",height=6,width=12)
```

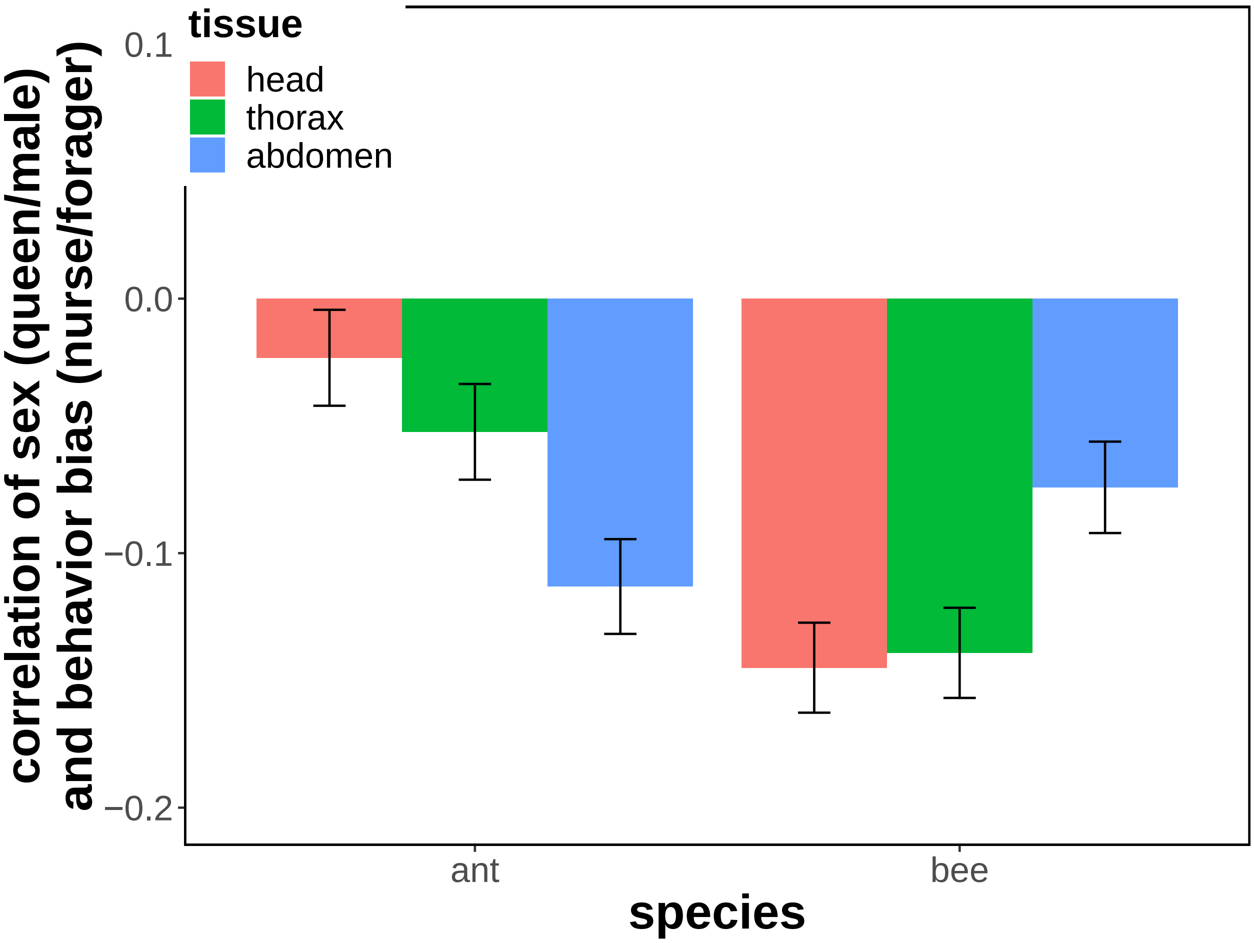
Sex bias vs caste bias in all tissues

```
FCcor <- function(test1,test2){
  FC = merge(test1[[1]],test2[[1]],by = "Gene")
  r = cor.test(FC$FC.x,FC$FC.y)
  return(c(r$estimate,c1=r$conf.int[1],c2=r$conf.int[2]))
}

Acor <- ldply(lapply(seq(1,3),function(i){
  FCcor(AcasteRes[[i]],AsexRes[[i]])
}))

Bcor <- ldply(lapply(seq(1,3),function(i){
  FCcor(BcasteRes[[i]],BsexRes[[i]])
}))

Acor$tissue = Bcor$tissue = c("head","thorax","abdomen")
Acor$species = "ant"
Bcor$species = "bee"
AllCor = rbind(Acor,Bcor)
AllCor$tissue=factor(AllCor$tissue,levels = c("head","thorax","abdomen"))

p <- ggplot(AllCor,aes(x=species,y=cor,fill=tissue))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position = position_dodge(.9),width=0.2)+
  main_theme+
  ylim(0,1)+
  ylab("correlation of caste and sex bias")+
  theme(legend.title = element_text(size=18,face="bold"),
        legend.position = c(0.1,0.9))
ggsave(p,file = "figures/p12.pdf",height=6,width=8)
p
```



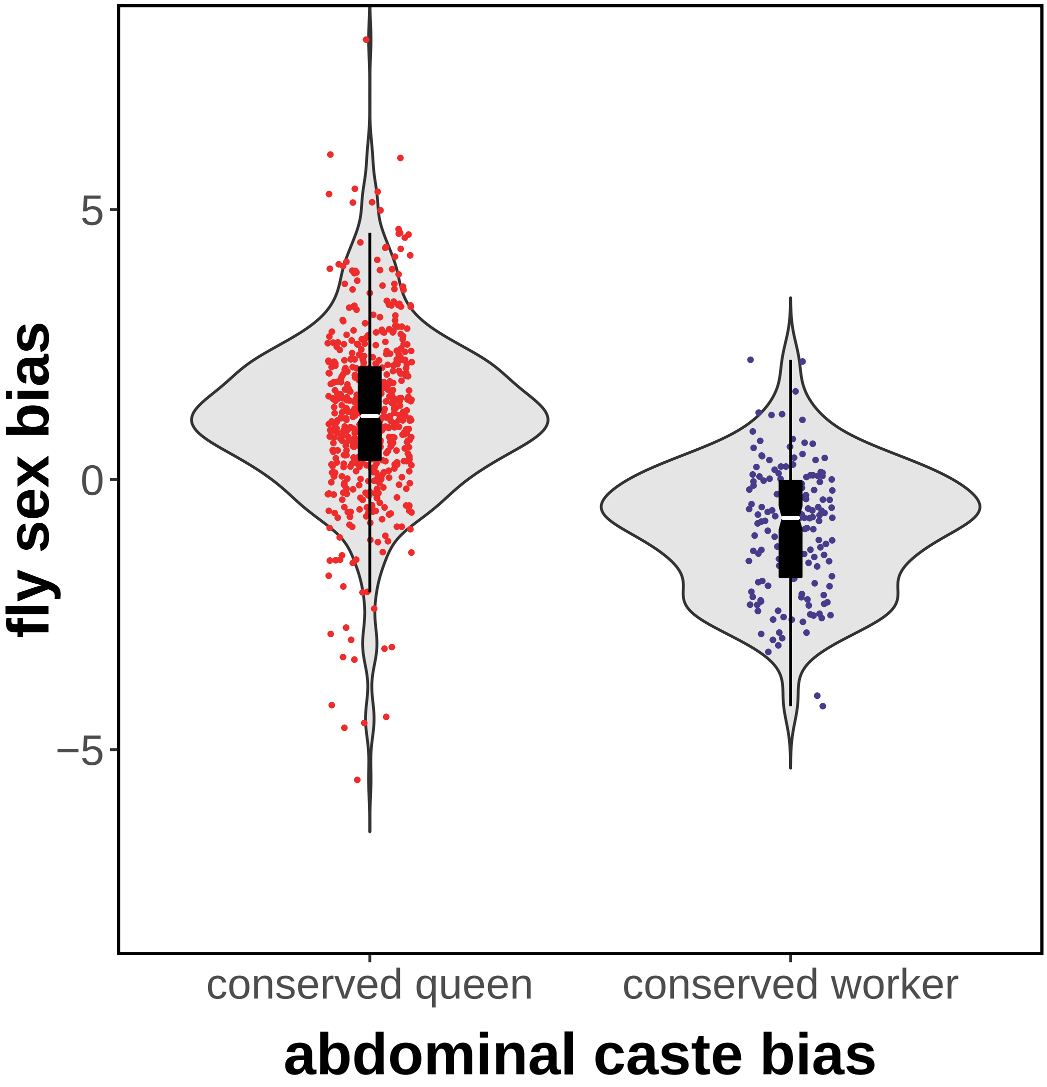
Sex bias vs caste bias for nurses and foragers

```
AsocRes <- lapply(antSocial,extractBias)
BsocRes <- lapply(beeSocial,extractBias)

Acor <- ldply(lapply(seq(1,3),function(i){
  FCcor(AsocRes[[i]],AsexRes[[i]])
}))

Bcor <- ldply(lapply(seq(1,3),function(i){
  FCcor(BsocRes[[i]],BsexRes[[i]])
}))

Acor$tissue = c("head","thorax","abdomen")
Bcor$tissue = c("head","thorax","abdomen")
Acor$species = "ant"
Bcor$species = "bee"
AllCor = rbind(Acor,Bcor)
AllCor$tissue=factor(AllCor$tissue,levels = c("head","thorax","abdomen"))

p <- ggplot(AllCor,aes(x=species,y=cor,fill=tissue))+
  geom_bar(stat="identity",position=position_dodge())+
  geom_errorbar(aes(ymin=c1,ymax=c2),position = position_dodge(.9),width=0.2)+
  main_theme+
  ylim(-0.2,0.1)+
  ylab("correlation of sex (queen/male)\nand behavior bias (nurse/forager)")+
  theme(legend.title = element_text(size=18,face="bold"),
        legend.position = c(0.1,0.9))
ggsave(p,file = "figures/p19.pdf",height=6,width=8)
p
```



3.2 Comparison to D. melanogaster

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

p4 <- ggplot(DmelSC[!grepl("non-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
```



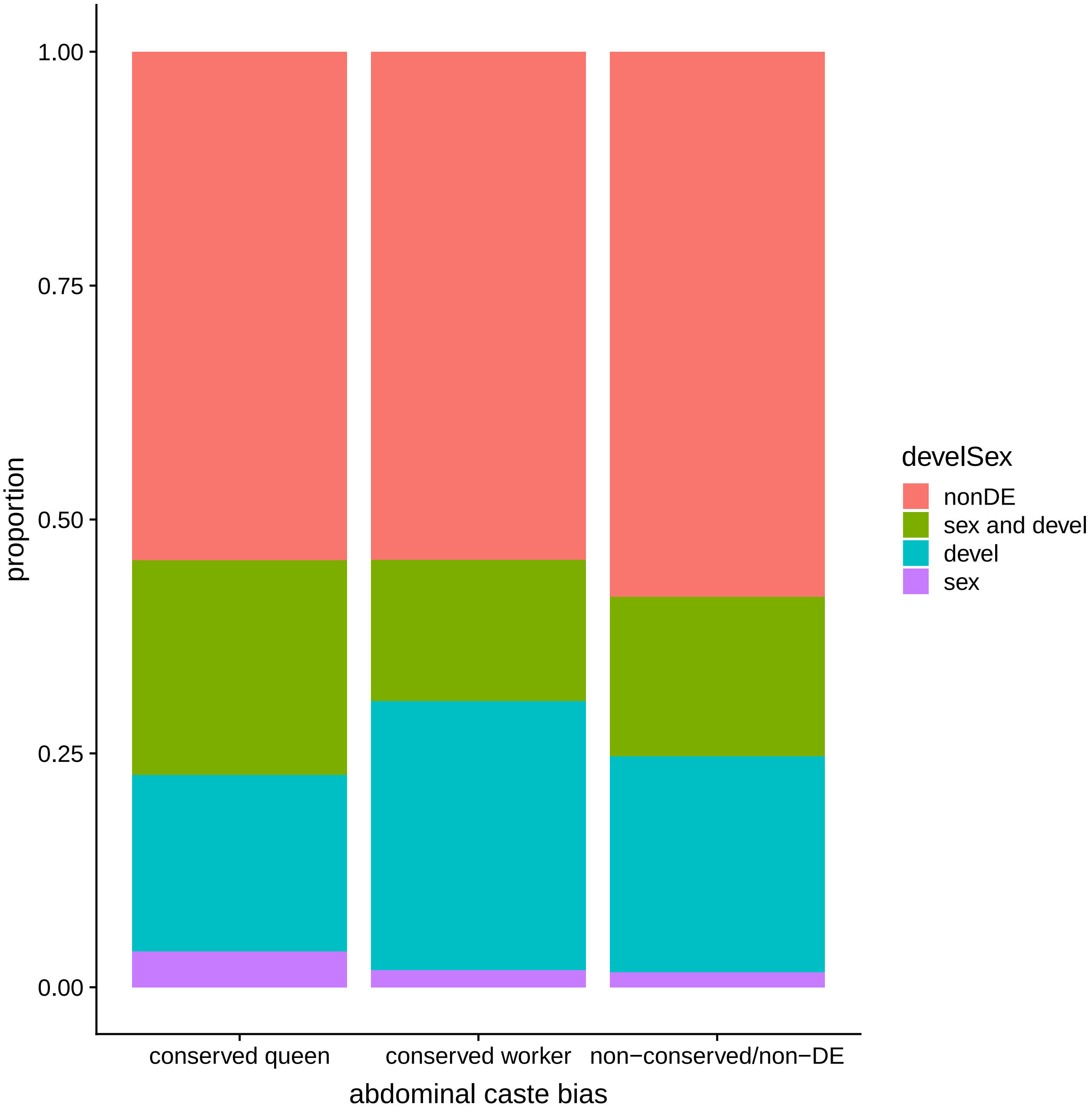
```
ggsave(p4,file = "figures/p13.pdf",height=6,width=6)

Dmel_male = DmelSC$gene_Amel[DmelSC$FDR < 0.05 & DmelSC$logFC > 0]
Dmel_female = DmelSC$gene_Amel[DmelSC$FDR < 0.05 & DmelSC$logFC < 0]
cQ = DmelSC$gene_Amel[DmelSC$abdDE=="conserved queen"]
cW = DmelSC$gene_Amel[DmelSC$abdDE=="conserved worker"]
Gtype = list(Dmel_m=Dmel_male,Dmel_f=Dmel_female,conQ=cQ,conW=cW)

p <- venn.diagram(Gtype,filename=NULL)
ggsave(p,file="figures/p14.pdf")
```

```
## Saving 6 x 6 in image
```

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/Dmel_Devel_IDs.txt") #Downloaded from flybase, genes with developmental terms
sex <- read.table("data/Dmel_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
```

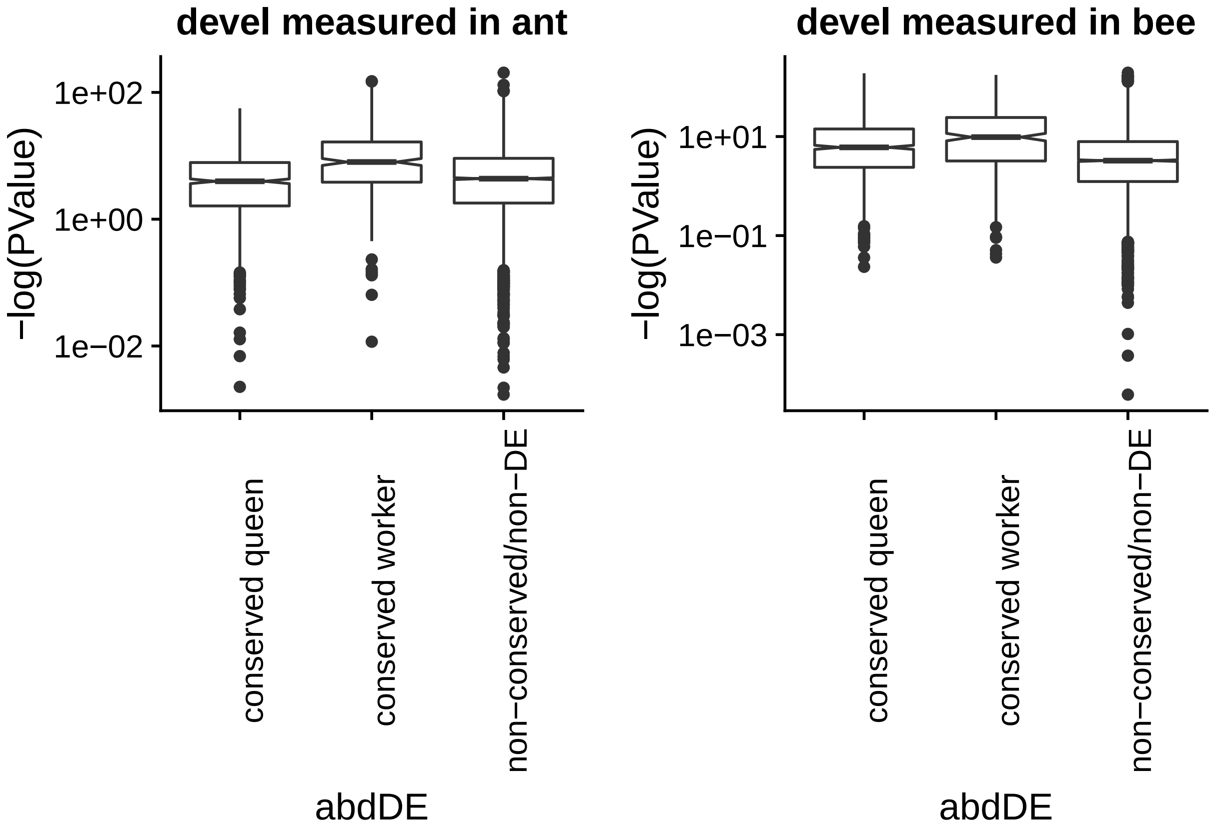


```
ggsave(p,file = "figures/p16.pdf",height=8,width=10)
```

-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(PValue)))+
  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(PValue)))+
  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 11 rows containing non-finite values (stat_boxplot).

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```



```
p = arrangeGrob(p1,p2,nrow=1)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 11 rows containing non-finite values (stat_boxplot).

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```

```r
ggsave(p,file = "figures/p15.pdf",height=8,width=10)
```

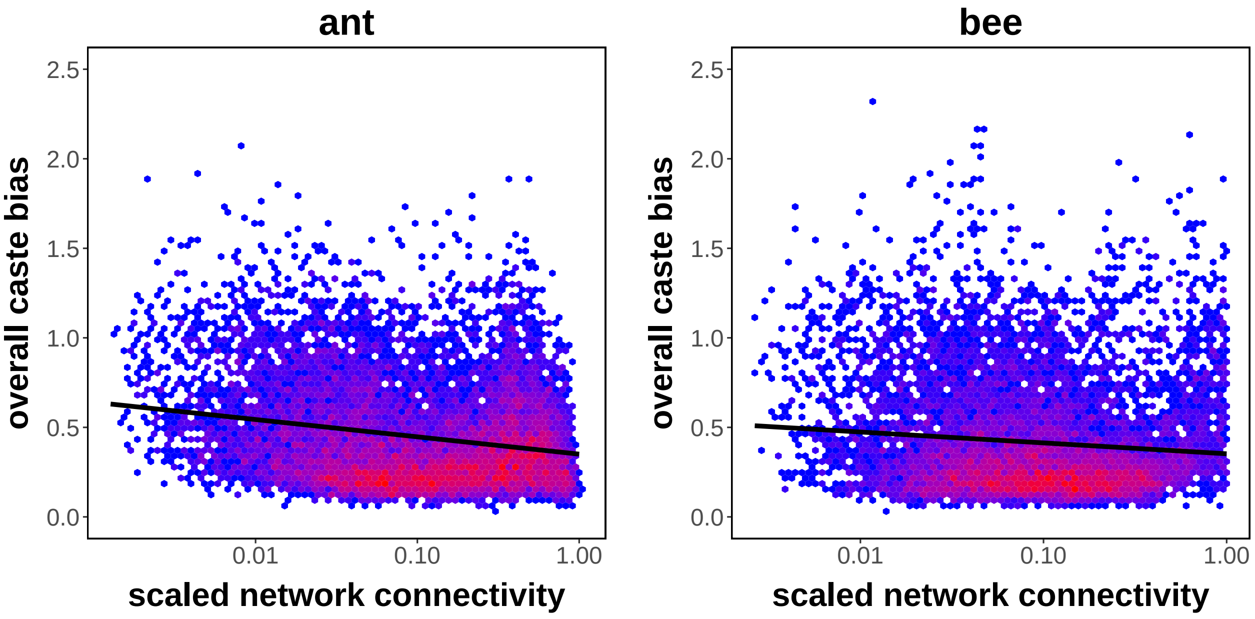
4.1 Calculation of overall caste bias

```r
#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)
```

4.1 Caste/Behavior bias vs connectivity

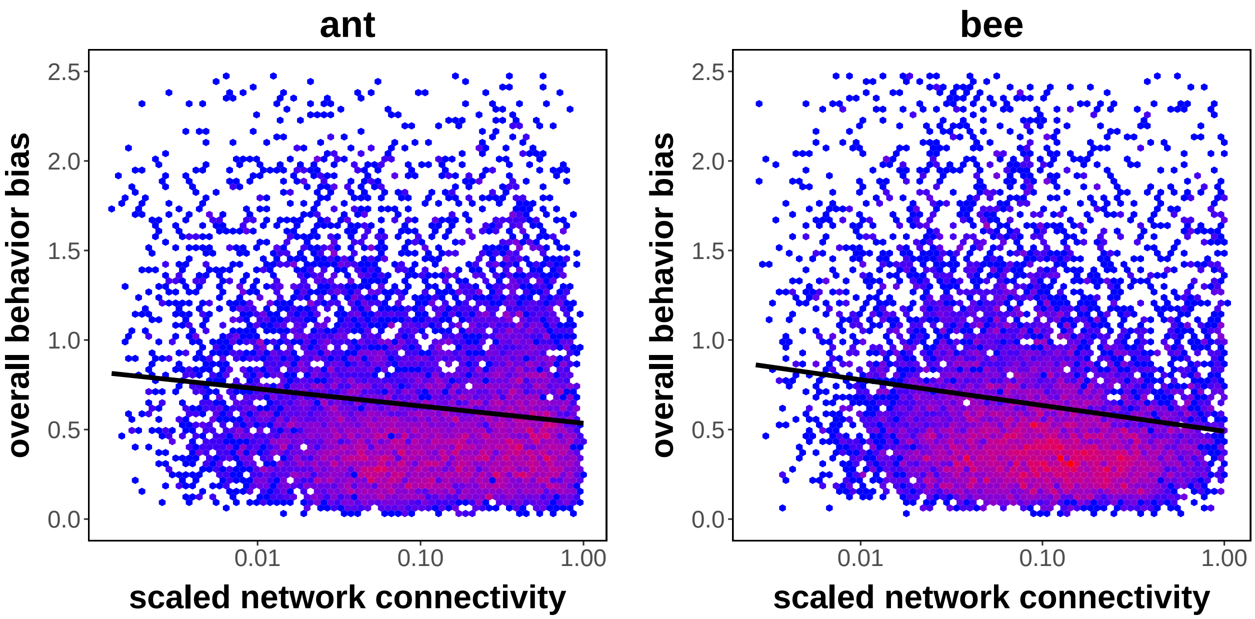
```r
cbAps = merge(antBias2,Aps,by="Gene")
cbBps = merge(beeBias2,Bps,by="Gene")
cbAps = merge(antConn,cbAps,by="Gene")
cbBps = merge(beeConn,cbBps,by="Gene")

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).
```



```r
p <- arrangeGrob(p1,p2,p3,p4,nrow=2)
```

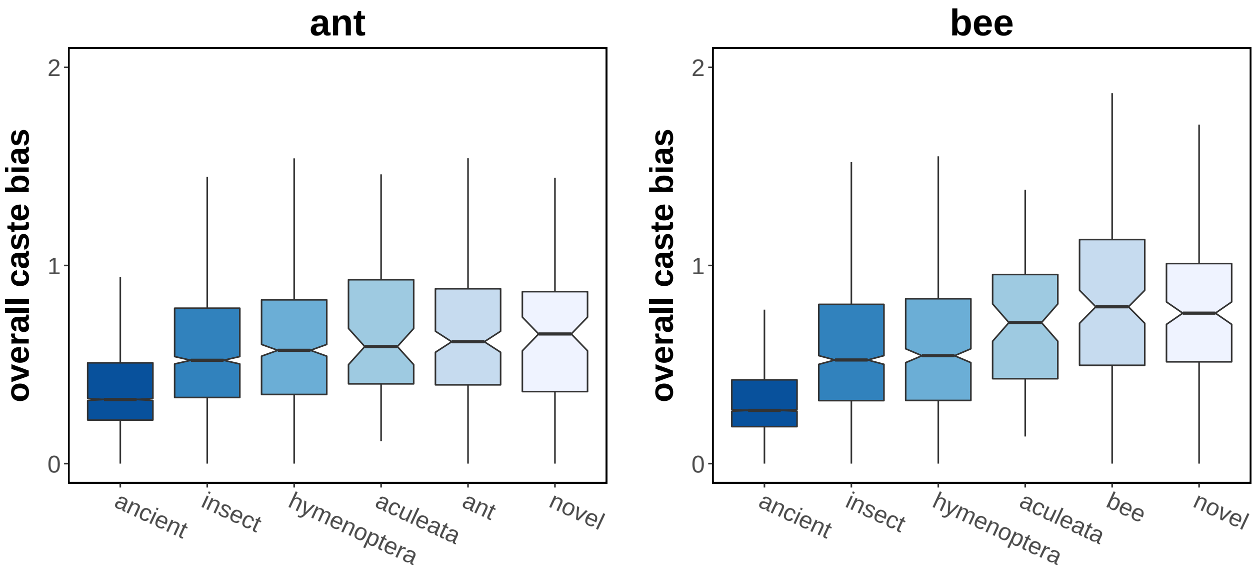
```
## Warning: Removed 7 rows containing missing values (geom_hex).

## 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 6 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).
```

```r
ggsave(p,file="figures/p22.pdf",height=12,width=12)
```

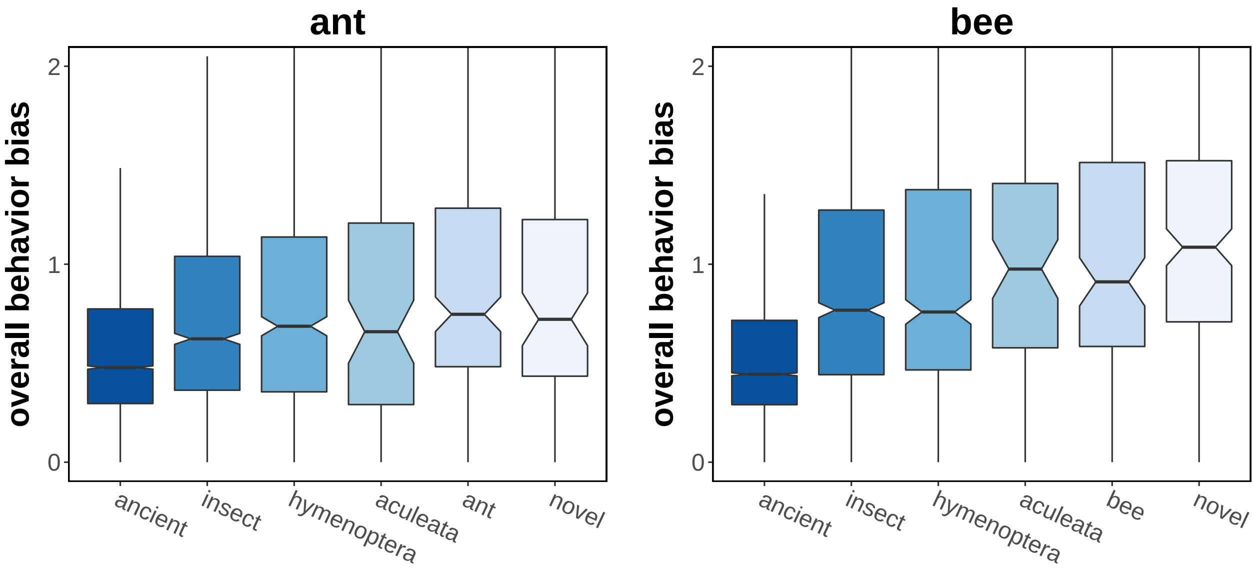
4.2 Caste/Behavior bias vs phylostrata

```r
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="to
  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)
```



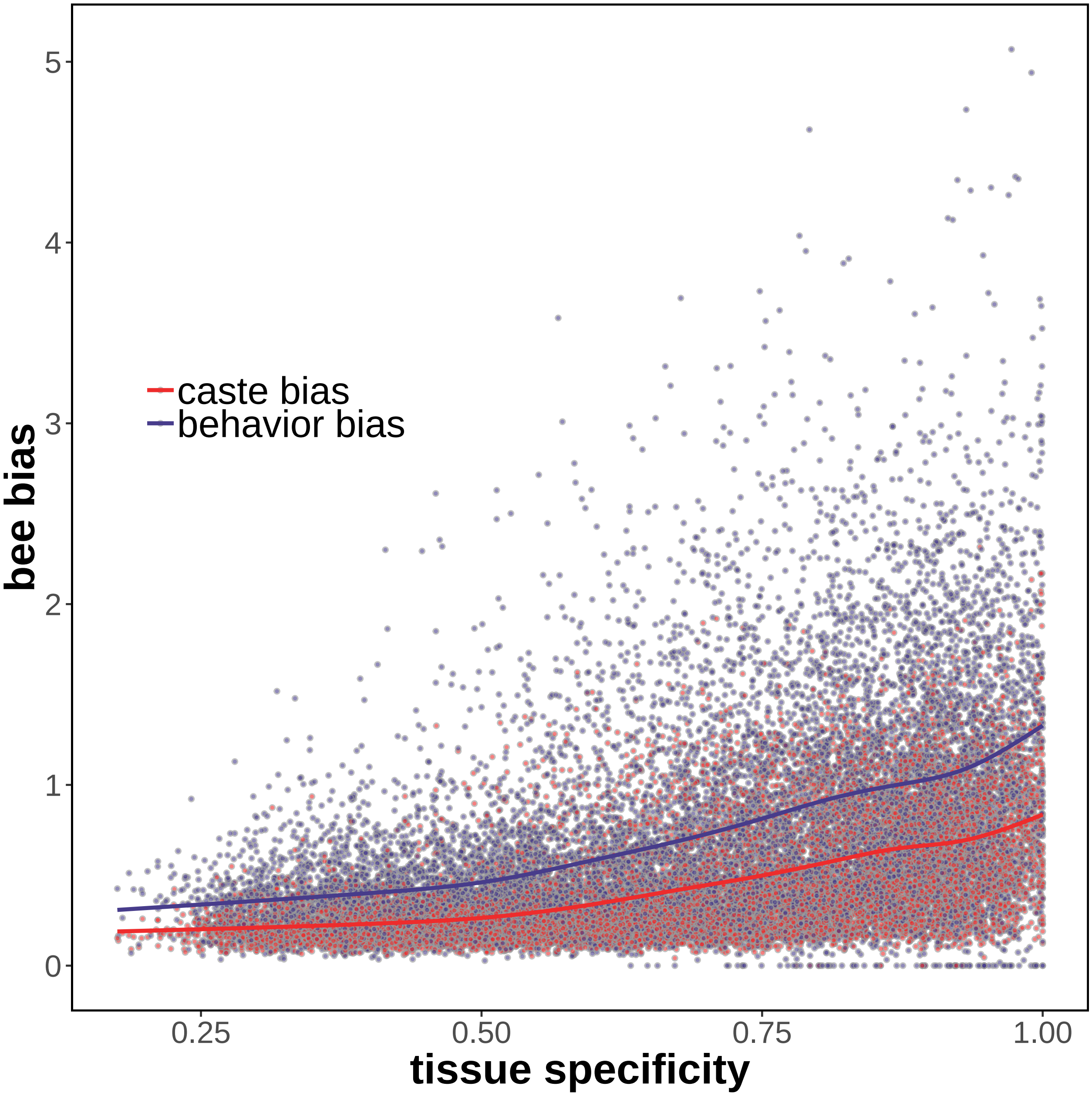
```
p <- arrangeGrob(p1,p2,p3,p4,nrow=2)
ggsave(p,file="figures/p21.pdf",height=12,width=12)
```

4.3 Caste bias vs tissue specificity

-tissue specificity is estimated across 12 honey bee tissues.

```r
bT = merge(tau,beeBias2,by="Gene")
bT$species = "bee"
bT$type = factor(bT$type,levels = c("caste","behavior"))
levels(bT$type) = c("caste bias","behavior bias")

p1 <- ggplot(bT,aes(x=tau,y=cb))+
  geom_point(size=1,alpha=0.6,pch=21,aes(fill=type),color="grey60")+
  geom_smooth(se=FALSE,aes(color=type))+
  scale_color_manual(values = SexPal)+
  scale_fill_manual(values = SexPal)+
  main_theme+xlab("tissue specificity")+ylab("bee bias")+
  theme(legend.position=c(0.2,0.6),
        legend.text = element_text(size=20),
        legend.title=element_blank())
p1
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
ggsave(p1,file="figures/p23.pdf",height=8,width=8)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

Comparison of caste and behavior bias between species

```
cbBias = merge(beeCB,ACUogg,by.x="Gene",by.y="gene_Amel")
cbBias = merge(cbBias,antCB,by.x="gene_Mphar",by.y="Gene")
cbBias = cbBias[!is.na(cbBias$OGGacu),]
socBias = merge(beeSB,ACUogg,by.x="Gene",by.y="gene_Amel")
socBias = merge(socBias,antSB,by.x="gene_Mphar",by.y="Gene")
socBias = socBias[!is.na(socBias$OGGacu),]
p3 <- ggplot(cbBias,aes(x=cb_noAdult.x,y=cb_noAdult.y))+
  geom_point(size=2,alpha=0.3)+
  geom_smooth(method="lm",se=FALSE,color="red")+
  main_theme+xlab("bee caste bias")+ylab("ant caste bias")

p4 <- ggplot(socBias,aes(x=cb.x,y=cb.y))+
  geom_point(size=2,alpha=0.3)+
  geom_smooth(method="lm",se=FALSE,color="red")+
  scale_x_continuous(breaks = c(0,1,2))+
  main_theme+xlab("bee behavioral bias")+ylab("ant behavior bias")

p <- arrangeGrob(p3,p4,nrow=1)
ggsave(p,file="figures/p20.pdf",width=12,height=7)
```