

Supplemental code for Social Scaling of Stressor Impacts

Accompanying code for “Colony size buffers interactions between neonicotinoid exposure and cold stress in bumble bees”

```
#Queenright colony experiments
```

Load and clean data for queenright experiments

Load data

```
# Get colony demographic metadata
setwd("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/colonyInfo")
metadata <- read.csv("csvFiles/colonyMetaData.csv")

### Parse into separate treatments ### initialize
metadata$neonic <- 0
metadata$cold <- 0
metadata$neonic[str_detect(metadata$Treatment, "eonic")] <- 1
metadata$cold[str_detect(metadata$Treatment, "old")] <- 1
metadata$neonic <- as.numeric(metadata$neonic)
metadata$cold <- as.numeric(metadata$cold)

### Load behavioral tracking data
setwd("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/thermalRig")
behData <- read.csv("behData_June2022.csv")
behData <- behData[complete.cases(behData[, c("porTimeMoving",
    "movingVelocity")]), ] ## Remove rows with missing data
# tracking.metadata <- read.csv('trackingMetadata.csv')
tag.data <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/tagData.csv")

# Correct name for one misidentified colony
tag.data$colony[tag.data$colony == "BB246"] <- "BG246"

### Add size at tagging to metadata by reading taglists
### replace nans with 'worker'
tag.data$caste[is.na(tag.data$caste)] <- "worker"
tag.data$caste[tag.data$caste == ""] <- "worker"
tag.data$caste[tag.data$caste == "Worker"] <- "worker"
tag.data$caste[tag.data$caste == "Queen"] <- "queen"
tag.data$caste[tag.data$caste == "Male"] <- "male"

# Add colony position to tag.data
for (i in 1:length(metadata[, 1])) {
    tag.data$thermalRigPosition[tag.data$colony == metadata$colonyID[i]] <- metadata$positionInThermalRig[i]
```

```

tag.data$unique_id <- paste(tag.data$block, tag.data$thermalRigPosition,
                           tag.data$id, sep = "_")

for (i in 1:length(metadata[, 1])) {
  # Figure out how many bees in this colony
  metadata$colony.size.at.tagging[i] <- dim(subset(tag.data,
    colony == metadata$colonyID[i]))[1]
  metadata$new_queens[i] <- dim(subset(tag.data, colony ==
    metadata$colonyID[i] & caste == "queen" & founding_queen !=
    "Y"))[1]
  metadata$males[i] <- dim(subset(tag.data, colony == metadata$colonyID[i] &
    caste == "male"))[1]
}

metadata$reproductive <- as.factor((metadata$new_queens + metadata$males) >
  0)

reproductive.vs.size.model <- glm(reproductive ~ log10(colony.size.at.tagging),
  data = subset(metadata, colony.size.at.tagging > 0), family = "binomial")
summary(reproductive.vs.size.model)

## Call:
## glm(formula = reproductive ~ log10(colony.size.at.tagging), family = "binomial",
##      data = subset(metadata, colony.size.at.tagging > 0))
##
## Deviance Residuals:
##       Min      1Q   Median      3Q      Max
## -1.1884 -0.9461 -0.7742  1.3922  1.6155
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.4415     0.7391  -1.950  0.0511 .
## log10(colony.size.at.tagging)  0.6478     0.5418   1.196  0.2318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 74.726 on 57 degrees of freedom
## Residual deviance: 73.239 on 56 degrees of freedom
## AIC: 77.239
##
## Number of Fisher Scoring iterations: 4

## Load thermal data
thermData <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressor")
thermData <- thermData[!is.na(thermData$block), ] #Remove some missing data

behData$colony.id <- paste(behData$block, behData$rigPosition,
                           sep = "_")
behData$uniqueID <- paste(behData$colony.id, behData$id, sep = "_")

```

```

## Fill in metadata to individual behavioral and colony
## level thermal data frames
for (i in 1:length(metadata[, 1])) {
  # print(i)
  bd.ind <- behData$rigPosition == metadata$positionInThermalRig[i] &
    behData$block == metadata$experimentalBlock[i]

  if (sum(bd.ind) > 0) {
    behData$neonic[bd.ind] <- as.logical(metadata$neonic[i])
    behData$cold[bd.ind] <- as.logical(metadata$cold[i])
    behData$colony.size.at.tagging[bd.ind] <- metadata$colony.size.at.tagging[i]
    behData$species[bd.ind] <- metadata$Species[i]
    behData$experimental.day[bd.ind] <- floor(behData[bd.ind,
      ]$dateNum) - min(floor(behData[bd.ind, ]$dateNum)) +
      1
    behData$time.from.start[bd.ind] <- behData[bd.ind, ]$dateNum -
      min(behData[bd.ind, ]$dateNum)
    worker.brood.ratio <- unique(behData$colony.size.at.tagging[bd.ind]/behData$numBrood[bd.ind])
    behData$worker.brood.ratio[bd.ind] <- worker.brood.ratio
    # Add reproductive state (binary)
    behData$reproductive[bd.ind] <- metadata$reproductive[i]

    # Identify first exposure
    rs <- behData$rampState[bd.ind]
    tfs <- behData$time.from.start[bd.ind]
    first.cold.time <- tfs[which(rs == 3)[1]]
    pre.exposure <- tfs < first.cold.time
    behData$pre.exposure[bd.ind] <- pre.exposure
    td.ind <- thermData$rigPosition == metadata$positionInThermalRig[i] &
      thermData$block == metadata$experimentalBlock[i]
    thermData$neonic[td.ind] <- as.logical(metadata$neonic[i])
    thermData$cold[td.ind] <- as.logical(metadata$cold[i])
    thermData$colony.size.at.tagging[td.ind] <- metadata$colony.size.at.tagging[i]
    thermData$species[td.ind] <- metadata$Species[i]
    thermData$experimental.day[td.ind] <- floor(thermData[td.ind,
      ]$dateNum) - min(floor(thermData[td.ind, ]$dateNum)) +
      1
    thermData$worker.brood.ratio[td.ind] <- worker.brood.ratio
  } else {
    # print(paste('no tracking data for colony ', i, ',
    # skipping', sep=''))
  }
}

# Remove missing values from thermData
thermData <- thermData[!is.na(thermData$block), ]

# Remove errant readings from thermData
thermData <- subset(thermData, (meanTempBrood_all - nestAirTemp) >
  -2)

# Add unique colony identifier to thermData

```

```

thermData$colony.id <- paste(thermData$block, thermData$rigPosition,
  sep = "_")
behData$tod.hours <- (behData$dateNum%%1) * 24 #Convert timestamp to daily hours
behData <- subset(behData, framesTracked > 10) #Select out low-quality tracking

# Variable conversion to factors
behData$id <- as.factor(behData$id)
behData$experimental.day <- as.factor(behData$experimental.day)
behData$neonic <- as.factor(behData$neonic)
behData$cold <- as.factor(behData$cold)

behData$trial <- paste(behData$colony.id, behData$date, sep = "_")

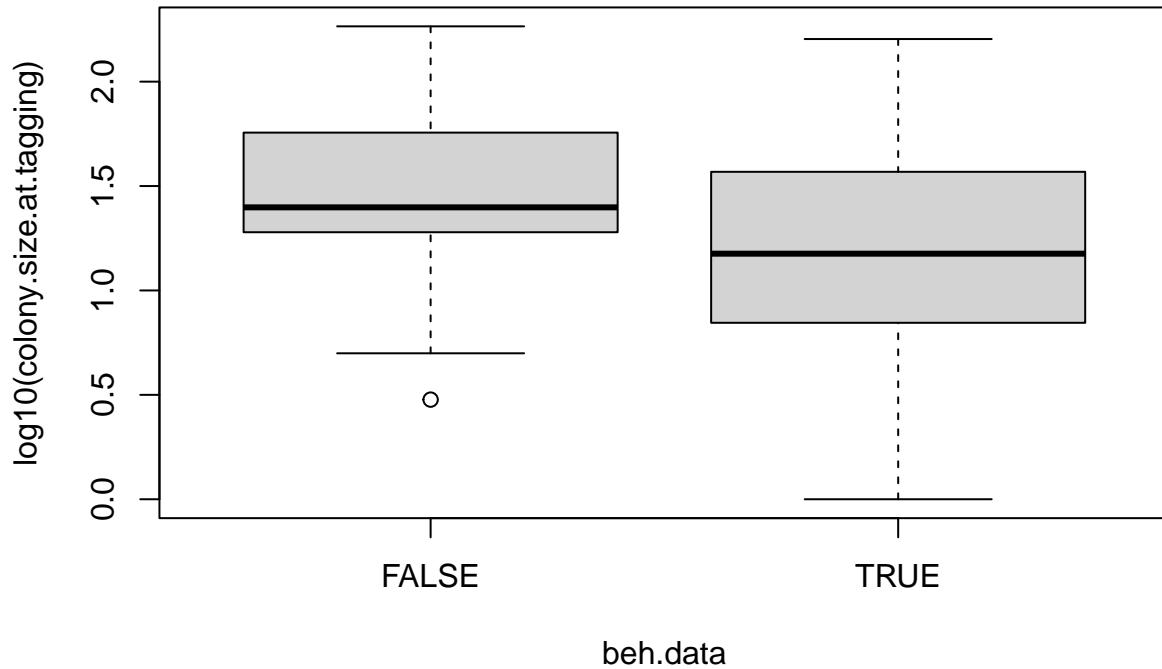
# Add marker for surface nesting species
behData$surfaceNesting <- behData$species == "griseocolis"

for (i in 1:length(tag.data$unique_id)) {
  behData$caste[behData$uniqueID == tag.data$unique_id[i]] <- tag.data$caste[i]
  behData$callow[behData$uniqueID == tag.data$unique_id[i]] <- tag.data$callow[i]
  behData$foundling.queen[behData$uniqueID == tag.data$unique_id[i]] <- tag.data$foundling_queen[i]
}

# create discrete rampState variable
behData$rampState_ch <- as.character(behData$rampState)
rmps <- as.character(seq(0, 5))
rep_names <- c("floating", "a_before", "ramp_down", "cold", "ramp_up",
  "b_after")
for (i in 1:length(rmps)) {
  behData$rampState_ch[behData$rampState_ch == rmps[i]] <- rep_names[i]
}
behData$rampState_ch <- as.factor(behData$rampState_ch)

# Examine colonies with missing behavioral data relative to
# rest of colonies
metadata$c.id <- paste(metadata$experimentalBlock, metadata$positionInThermalRig,
  sep = "_")
metadata$beh.data <- metadata$c.id %in% unique(behData$colony.id)
boxplot(log10(colony.size.at.tagging) ~ beh.data, data = subset(metadata,
  colony.size.at.tagging > 0))

```



```
wilcox.test(colony.size.at.tagging ~ beh.data, data = subset(metadata,
  colony.size.at.tagging > 0))
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties

##
## Wilcoxon rank sum test with continuity correction
##
## data: colony.size.at.tagging by beh.data
## W = 274.5, p-value = 0.2503
## alternative hypothesis: true location shift is not equal to 0
```

Perform principal components analysis on within-nest behavioral metrics

```
# Run principal components analysis on nest behavior
pc.vars <- c("distanceToCenterInstantaneous", "movingVelocity",
  "medianDistanceToClosestBrood", "medianDistanceToClosestWaxpot",
  "medianDistanceToAllBrood", "medianDistanceToAllWaxpots",
  "degreeCentrality", "meanInteractionRate", "spatCorAll",
  "broodInfoRichness", "waxpotInfoRichness")

behData <- behData[complete.cases(behData[, pc.vars]), ]

# Subset to only target variables
```

```

pca.data.sub <- behData[, pc.vars]

pca.data.sub$spatCorAll[pca.data.sub$spatCorAll < 0] <- 0 #Make negative correlations zero

# Transform raw variables
sqrt.tf.vars <- c("distanceToCenterInstantaneous", "medianDistanceToAllBrood",
  "medianDistanceToAllWaxpots")
pca.data.sub[, sqrt.tf.vars] <- pca.data.sub[, sqrt.tf.vars]^0.5

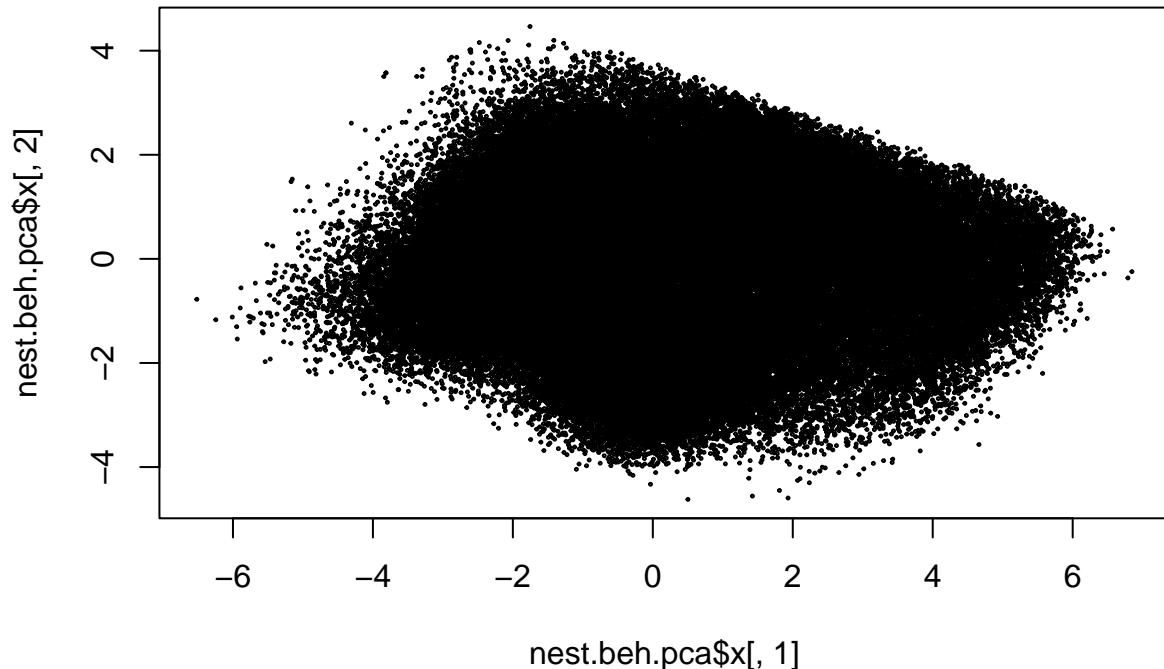
n.root.tf.vars <- c("medianDistanceToClosestBrood", "medianDistanceToClosestWaxpot")
pca.data.sub[, n.root.tf.vars] <- pca.data.sub[, n.root.tf.vars]^0.2

lg.tf.vars <- c("meanInteractionRate", "spatCorAll", "broodInfoRichness",
  "waxpotInfoRichness")
pca.data.sub[, lg.tf.vars] <- log10(0.01 + pca.data.sub[, lg.tf.vars])
pca.data.sub$degreeCentrality <- pca.data.sub$degreeCentrality^0.5

# for(i in 1:length(pc.vars)){ hist(pca.data.sub[, i], main
# = pc.vars[i]) }

nest.beh.pca <- prcomp(pca.data.sub, center = TRUE, scale = TRUE)
plot(nest.beh.pca$x[, 1], nest.beh.pca$x[, 2], cex = 0.2)

```



```

behData$PC1 <- nest.beh.pca$x[, 1]
behData$PC1 <- -behData$PC1 #NB Switch sign of PC1 to reflect 'spatial centrality' in other findings

# Yeo-Johnson transformation
tf <- yeojohnson(behData$PC1)
behData$PC1 <- predict(tf)

behData$PC2 <- nest.beh.pca$x[, 2]
behData$PC3 <- nest.beh.pca$x[, 3]

# Subtract baseline PC1 for each colony
colonies <- unique(behData$colony.id)
baseline.pcs <- data.frame(matrix(nrow = length(colonies), ncol = 3))
colnames(baseline.pcs) <- c("colony", "pc1", "pc2")

for (i in 1:length(colonies)) {
  c.id <- behData$colony.id == colonies[i]
  first.ramp.time <- min(subset(behData, colony.id == colonies[i] &
    rampState > 0)$time.from.start)
  baseline.data <- subset(behData, colony.id == colonies[i] &
    rampState == 0 & time.from.start < first.ramp.time)
  if (length(baseline.data[, 1]) == 0) {
    print("no baseline data")
    print(i)
    print(colonies[i])
  }
  mean.pc1 <- mean(baseline.data$PC1)
  behData$PC1.res[c.id] <- behData$PC1[c.id] - mean.pc1

  mean.pc2 <- mean(baseline.data$PC2)
  behData$PC2.res[c.id] <- behData$PC2[c.id] - mean.pc2
  behData$first.ramp.time <- first.ramp.time
}

# pc loadings visualization prep
pc.loadings <- -nest.beh.pca$rotation[, 1:2]
pc.loadings <- pc.loadings[order(pc.loadings[, 1]), ]
pc.loadings <- as.data.frame(pc.loadings)
pc.loadings$variable <- rownames(pc.loadings)

setwd(output_folder)

pdf("FigS4a.pdf")
ggplot(data = pc.loadings, aes(x = reorder(variable, +PC1), y = PC1,
  fill = PC1)) + geom_bar(stat = "identity") + coord_flip() +
  scale_fill_distiller(palette = "RdYlBu") + theme_minimal()
dev.off()

## pdf
## 2

```

```

pdf("FigS4b.pdf")
ggplot(data = pc.loadings, aes(x = reorder(variable, +PC1), y = PC2,
  fill = PC2)) + geom_bar(stat = "identity") + coord_flip() +
  scale_fill_distiller(palette = "RdYlBu") + theme_minimal()
dev.off()

## pdf
## 2

behData$caste <- as.factor(behData$caste)
behData$uniqueID <- as.factor(behData$uniqueID)
behData$colony.id <- as.factor(behData$colony.id)

# Remove perplexus colony from analysis, and exclude
# observations with low tracking quality (less than 30
# frames within a video)
behData <- subset(behData, species %in% c("impatiens", "bimaculatus",
  "griseoculis") & framesTracked > 30)

```

Body temperature validation

```

body_temp_ref_data <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScaling0

temp_val_plots <- function(body_temp_ref_data) {
  par(mfcol = c(2, 2))

  plot(ground_truth_radius ~ predicted_radius, data = body_temp_ref_data,
    axes = FALSE, pch = 19, col = "coral3", xlab = ("Predicted Body Temperature"),
    ylab = "Manual body temperature")
  abline(a = 0, b = 1, lty = 2)
  axis(1)
  axis(2)
  cc = cor(body_temp_ref_data$ground_truth_radius, body_temp_ref_data$predicted_radius,
    method = "pearson", use = "complete.obs")
  text(12, 27, paste("Pearson correlation\n = ", signif(cc,
    2), sep = ""))
}

plot(ground_truth_pt ~ predicted_pt, data = body_temp_ref_data,
  axes = FALSE, pch = 19, col = "coral3", xlab = ("Predicted Body Temperature"),
  ylab = "Manual body temperature")
abline(a = 0, b = 1, lty = 2)
axis(1)
axis(2)
cc = cor(body_temp_ref_data$ground_truth_pt, body_temp_ref_data$predicted_pt,
  method = "pearson", use = "complete.obs")
text(12, 27, paste("Pearson correlation\n = ", signif(cc,
  2), sep = ""))

plot(ground_truth_pt ~ ground_truth_radius, data = body_temp_ref_data,
  axes = FALSE, pch = 19, col = "coral3", ylab = ("Body Temperature (point)"),
  xlab = "Body Temperature (radius)")

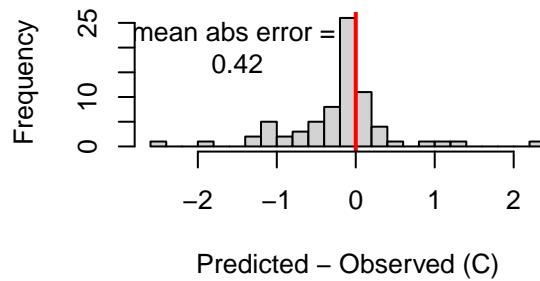
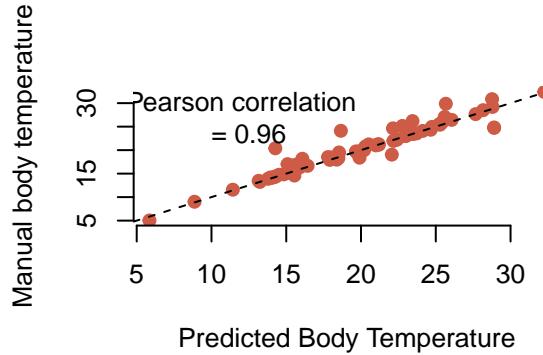
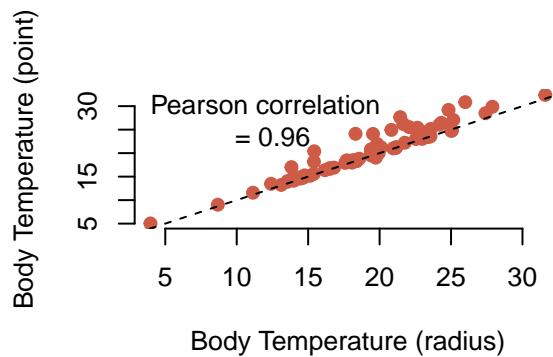
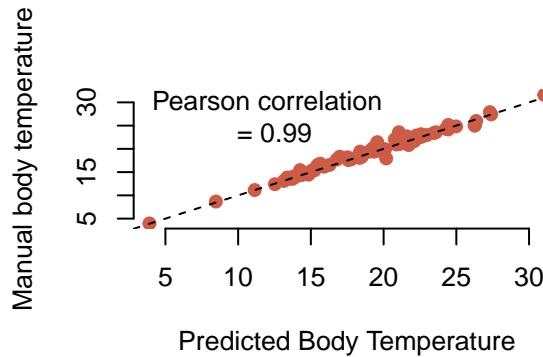
```

```

abline(a = 0, b = 1, lty = 2)
axis(1)
axis(2)
cc = cor(body_temp_ref_data$ground_truth_pt, body_temp_ref_data$ground_truth_radius,
         method = "pearson", use = "complete.obs")
text(12, 27, paste("Pearson correlation\n = ", signif(cc,
2), sep = ""))
hist(body_temp_ref_data$predicted_radius - body_temp_ref_data$ground_truth_radius,
     axes = FALSE, xlab = "Predicted - Observed (C)", ylab = "Frequency",
     breaks = 20, main = "")
axis(2)
abline(v = 0, lwd = 2, col = "red")
mean_error = mean(abs(body_temp_ref_data$predicted_radius -
body_temp_ref_data$ground_truth_radius), na.rm = TRUE)
text(-1.5, 20, paste("mean abs error = \n", signif(mean_error,
2), sep = ""))
axis(1)
}

# Plot inline
temp_val_plots(body_temp_ref_data)

```



```

setwd(output_folder)
# Export figure panel

```

```

pdf("FigS3.pdf")
temp_val_plots(body_temp_ref_data)
dev.off()

## pdf
## 2

tmp <- subset(behData, cold == TRUE & neonic == FALSE & colony.size.at.tagging >
  4 & framesTracked > 20)
tmp$distanceToCenterInstantaneous <- tmp$distanceToCenterInstantaneous/pixel_conversion
# Get residuals values for relevant metrics
pc1.res.model <- lmer(PC1 ~ 1 + (1 | uniqueID) + (1 | colony.id),
  data = tmp)
tmp$pc1.ind.residual <- residuals(pc1.res.model)
bor.res.model <- lmer(broodOccupancyRate ~ 1 + (1 | uniqueID) +
  (1 | colony.id), data = tmp)
tmp$bor.ind.residual <- residuals(bor.res.model)
dist.res.model <- lmer(distanceToCenterInstantaneous ~ 1 + (1 |
  uniqueID) + (1 | colony.id), data = tmp)
tmp$dist.ind.residual <- residuals(dist.res.model)

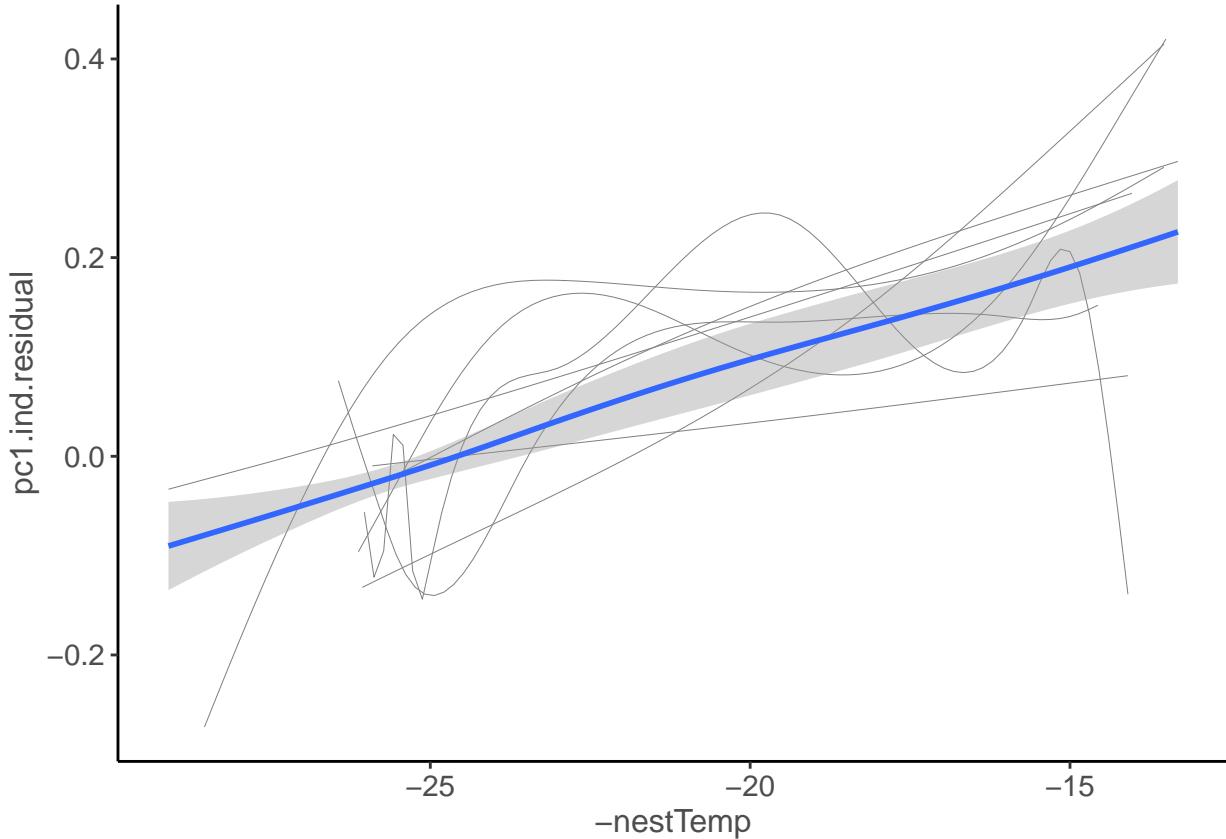
p1 <- ggplot(tmp, aes(x = -nestTemp, y = pc1.ind.residual)) +
  geom_smooth(method = "gam", se = F, aes(group = colony.id),
  size = 0.1, colour = "grey50") + geom_smooth(method = "gam")

## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.

p1

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

```



```

p2 <- ggplot(tmp, aes(x = -nestTemp, y = bor.ind.residual)) +
  geom_smooth(method = "gam", se = F, aes(group = colony.id),
  size = 0.1, colour = "grey50") + geom_smooth(method = "gam")
p3 <- ggplot(tmp, aes(x = -nestTemp, y = dist.ind.residual)) +
  geom_smooth(method = "gam", se = F, aes(group = colony.id),
  size = 0.1, colour = "grey50") + geom_smooth(method = "gam")
setwd(output_folder)

# Export
pdf("Fig2_ABC.pdf", width = 2, height = 6)
grid.arrange(p1, p2, p3, nrow = 3)

```

```

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

dev.off()

```

```

## pdf
## 2

```

Active response model - effects of treatment during active cooling

```
setwd(output_folder)

response.data = subset(behData, rampState %in% c(1, 2, 3) & caste %in%
  c("worker", "queen"))

pc1.active.response.model <- lmer(PC1 ~ log10(colony.size.at.tagging) *
  nestTemp * neonic + species + (1 | caste) + (1 | experimental.day) +
  (1 | block) + (1 | colony.id/uniqueID), data = response.data,
  REML = FALSE)

pc1.active.response.model.s1 <- lmer(PC1 ~ log10(colony.size.at.tagging) *
  neonic + nestTemp + species + (1 | caste) + (1 | experimental.day) +
  (1 | block) + (1 | colony.id/uniqueID), data = response.data,
  REML = FALSE)
pc1.active.response.model.s2 <- lmer(PC1 ~ log10(colony.size.at.tagging) *
  nestTemp + neonic + species + (1 | caste) + (1 | experimental.day) +
  (1 | block) + (1 | colony.id/uniqueID), data = response.data,
  REML = FALSE)
pc1.active.response.model.s3 <- lmer(PC1 ~ log10(colony.size.at.tagging) *
  neonic + nestTemp + species + (1 | caste) + (1 | experimental.day) +
  (1 | block) + (1 | colony.id/uniqueID), data = response.data,
  REML = FALSE)
pc1.active.response.model.s4 <- lmer(PC1 ~ log10(colony.size.at.tagging) +
  neonic + nestTemp + species + (1 | caste) + (1 | experimental.day) +
  (1 | block) + (1 | colony.id/uniqueID), data = response.data,
  REML = FALSE)
pc1.active.response.model.s5 <- lmer(PC1 ~ log10(colony.size.at.tagging) +
  neonic * nestTemp + species + (1 | caste) + (1 | experimental.day) +
  (1 | block) + (1 | colony.id/uniqueID), data = response.data,
  REML = FALSE)
pc1.active.response.model.s6 <- lmer(PC1 ~ log10(colony.size.at.tagging) *
  neonic * nestTemp + nestTemp * species + (1 | caste) + (1 |
  experimental.day) + (1 | block) + (1 | colony.id/uniqueID),
  data = response.data)

capture.output(anova(pc1.active.response.model, pc1.active.response.model.s1,
  pc1.active.response.model.s2, pc1.active.response.model.s3,
  pc1.active.response.model.s4, pc1.active.response.model.s5,
  pc1.active.response.model.s6), file = "TableS1.comp.txt")

## refitting model(s) with ML (instead of REML)

write.csv(as.data.frame(anova(pc1.active.response.model, pc1.active.response.model.s1,
  pc1.active.response.model.s2, pc1.active.response.model.s3,
  pc1.active.response.model.s4, pc1.active.response.model.s5,
  pc1.active.response.model.s6)), file = "TableS1.comp.csv")
```

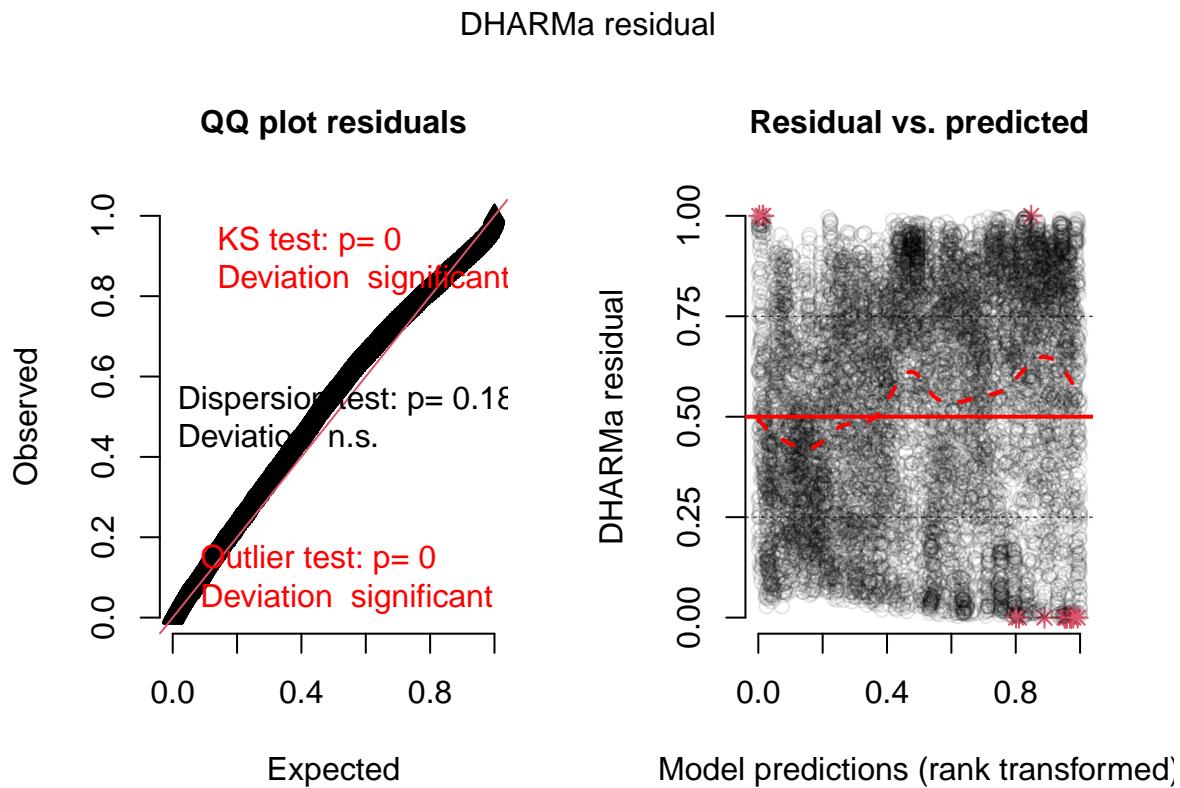
```
## refitting model(s) with ML (instead of REML)
```

```

# Best model -> full interaction model
capture.output((summary(pc1.active.response.model)), file = "TableS1.final.txt")
write.csv(signif(summary(pc1.active.response.model)$coefficients,
  3), file = "TableS1.final.csv")

sim.res <- simulateResiduals(pc1.active.response.model)
plot(sim.res)

```



```

# Export figure panel
pdf("Fig2_de.pdf")
plot_model(pc1.active.response.model, type = "pred", terms = c("nestTemp",
  "neonic", "colony.size.at.tagging [4, 80]", ci.lvl = 0.5,
  colors = list(ctrl_color, imid_color), title = "", axis.title = c(expression(paste("Nest Temperatur
  degree, "C)), sep = "")), "Spatial Centrality (PC1)",
  show.legend = F)
dev.off()

## pdf
## 2

```

Recovery model - effects of treatment on behavioral recovery after cold exposure

```

setwd(output_folder)

sub.dat <- subset(behData, rampState %in% c(1, 5) & cold == TRUE &
  caste %in% c("worker", "queen"))
sub.dat$rampState <- as.factor(sub.dat$rampState)

pc1.recovery.model.tmp.sp.int <- lmer(PC1 ~ rampState_ch * neonic *
  log10(colony.size.at.tagging) + rampState_ch * species +
  (1 | colony.id/uniqueID), data = sub.dat, REML = FALSE)
pc1.recovery.model <- lmer(PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) +
  species + (1 | colony.id/uniqueID), data = sub.dat, REML = FALSE)
anova(pc1.recovery.model, pc1.recovery.model.tmp.sp.int) #No evidence for temperature * species interaction

## Data: sub.dat
## Models:
## pc1.recovery.model: PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) + species + (1 | colony.id/uniqueID)
## pc1.recovery.model.tmp.sp.int: PC1 ~ rampState_ch * neonic * log10(colony.size.at.tagging) + rampState_ch * species + (1 | colony.id/uniqueID)
## pc1.recovery.model.s1: PC1 ~ rampState_ch + neonic * log10(colony.size.at.tagging) + species + (1 | colony.id/uniqueID)
## pc1.recovery.model.s2: PC1 ~ rampState_ch * neonic + log10(colony.size.at.tagging) + species + (1 | colony.id/uniqueID)
## pc1.recovery.model.s3: PC1 ~ neonic * log10(colony.size.at.tagging) + rampState_ch + species + (1 | colony.id/uniqueID)
## pc1.recovery.model.s4: PC1 ~ neonic + log10(colony.size.at.tagging) + rampState_ch + species + (1 | colony.id/uniqueID)

## Model selection
capture.output(anova(pc1.recovery.model.tmp.sp.int, pc1.recovery.model,
  pc1.recovery.model.s1, pc1.recovery.model.s2, pc1.recovery.model.s3,
  pc1.recovery.model.s4), file = "TableS2.comp.txt")
write.csv(as.data.frame(anova(pc1.recovery.model.tmp.sp.int,
  pc1.recovery.model, pc1.recovery.model.s1, pc1.recovery.model.s2,
  pc1.recovery.model.s3, pc1.recovery.model.s4)), file = "TableS2.comp.csv")

write.csv(signif(as.data.frame(summary(pc1.recovery.model)$coefficients),
  3), file = "TableS2.best.csv")
capture.output(summary(pc1.recovery.model), file = "TableS2.best.txt")
# Export panels
pdf("Fig2_cd.pdf")
plot_model(pc1.recovery.model, type = "pred", terms = c("rampState_ch",
  "neonic", "colony.size.at.tagging [4, 80]"), ci.lvl = 0.5,
  colors = list(ctrl_color, imid_color), title = "", axis.title = c("Observation Period",
  "Spatial Centrality (PC1)"), show.legend = F)
dev.off()

```

```

## pdf
## 2

#Worker group models Characterize treatment effects by initial behavioral groups (bees that are on vs. off
the nest at beginning of trials)

setwd(output_folder)

#### look at effects during ramp down by whether bees were
#### already on the nest
sub.dat <- subset(behData, rampState %in% c(1, 2, 3) & cold ==
  TRUE & caste %in% c("worker", "queen")) #Subset to data in initial cold exposure

colony.size.cutoff <- 30 #Define workers that were on nest at ambient temperatures
sd <- subset(sub.dat, rampState < 2 & nestTemp > 23)
tmp <- tapply(sd$medianDistancetoClosestBrood, sd$uniqueID, median,
  na.rm = TRUE) #Get averages in this time period for all bees

incubator.list <- names(tmp)[which(tmp < pixel_conversion)]
non.incubator.list <- names(tmp)[which(tmp > pixel_conversion *
  2)]

# Model responses with 'incubator' as an explicit
# interaction term
sub.dat$incubator <- NA
sub.dat$incubator[sub.dat$uniqueID %in% incubator.list] <- "inc"
sub.dat$incubator[sub.dat$uniqueID %in% non.incubator.list] <- "ninc"

# Subset data to small (<30 worker) colonies of impatiens
# to isolate individual variation
tmp.dat <- subset(sub.dat, caste == "worker" & incubator %in%
  c("inc", "ninc") & colony.size.at.tagging < colony.size.cutoff)
pc1.response.model.by.incubator <- lmer(PC1 ~ nestTemp * neonic *
  incubator + species + (1 | colony.id/uniqueID), data = tmp.dat,
  REML = FALSE)
summary(pc1.response.model.by.incubator)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula:
## PC1 ~ nestTemp * neonic * incubator + species + (1 | colony.id/uniqueID)
## Data: tmp.dat
##
##      AIC      BIC  logLik deviance df.resid
##  2732.9  2798.5 -1353.4   2706.9     1138
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -4.3735 -0.4875 -0.0016  0.5917  3.5714
##
## Random effects:
## Groups            Name        Variance Std.Dev.
## uniqueID:colony.id (Intercept) 0.26520  0.5150

```

```

## colony.id          (Intercept) 0.05578  0.2362
## Residual           0.52808  0.7267
## Number of obs: 1151, groups: uniqueID:colony.id, 94; colony.id, 13
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                1.603e+00 3.360e-01 2.381e+01  4.772
## nestTemp                  -2.463e-02 9.588e-03 1.073e+03 -2.568
## neonicTRUE                 -4.995e-02 3.381e-01 4.092e+01 -0.148
## incubatorninc              8.506e-02 3.592e-01 4.615e+02  0.237
## speciesgriseocolis        -3.158e-01 2.945e-01 1.010e+01 -1.072
## speciesimpatiens           5.778e-02 2.738e-01 8.098e+00  0.211
## nestTemp:neonicTRUE         5.123e-03 1.221e-02 1.071e+03  0.420
## nestTemp:incubatorninc     -6.072e-02 1.567e-02 1.094e+03 -3.875
## neonicTRUE:incubatorninc   -1.781e+00 4.836e-01 4.227e+02 -3.682
## nestTemp:neonicTRUE:incubatorninc 4.859e-02 2.018e-02 1.108e+03  2.408
##                               Pr(>|t|)
## (Intercept)                7.56e-05 ***
## nestTemp                   0.010350 *
## neonicTRUE                 0.883277
## incubatorninc              0.812908
## speciesgriseocolis        0.308601
## speciesimpatiens           0.838091
## nestTemp:neonicTRUE         0.674895
## nestTemp:incubatorninc     0.000113 ***
## neonicTRUE:incubatorninc   0.000261 ***
## nestTemp:neonicTRUE:incubatorninc 0.016211 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                               (Intr) nstTmp nnTRUE incbtr spcsgr spcsmp nsT:TRUE nstTm: nTRUE:
## nestTemp      -0.510
## neonicTRUE    -0.544  0.511
## incubatrnnnc -0.480  0.477  0.412
## specsgrscls -0.519 -0.005 -0.192  0.066
## specismptns -0.672 -0.006  0.060  0.113  0.662
## nstTmp:TRUE   0.400 -0.785 -0.654 -0.375  0.007  0.004
## nstTmp:ncbt   0.298 -0.612 -0.305 -0.801  0.007  0.019  0.480
## nncTRUE:ncb   0.273 -0.355 -0.561 -0.732  0.109  0.022  0.456   0.596
## nstTm:TRUE:   -0.236  0.475  0.384  0.622  0.008 -0.009 -0.605  -0.777 -0.788

```

```

pc1.response.model.by.incubator.s1 <- lmer(PC1 ~ nestTemp * neonic +
  incubator + species + (1 | colony.id/uniqueID), data = tmp.dat,
  REML = FALSE)
pc1.response.model.by.incubator.s2 <- lmer(PC1 ~ nestTemp + neonic *
  incubator + species + (1 | colony.id/uniqueID), data = tmp.dat,
  REML = FALSE)
pc1.response.model.by.incubator.s3 <- lmer(PC1 ~ nestTemp * incubator +
  neonic + species + (1 | colony.id/uniqueID), data = tmp.dat,
  REML = FALSE)
pc1.response.model.by.incubator.s4 <- lmer(PC1 ~ nestTemp + neonic +
  incubator + species + (1 | colony.id/uniqueID), data = tmp.dat,
  REML = FALSE)

```

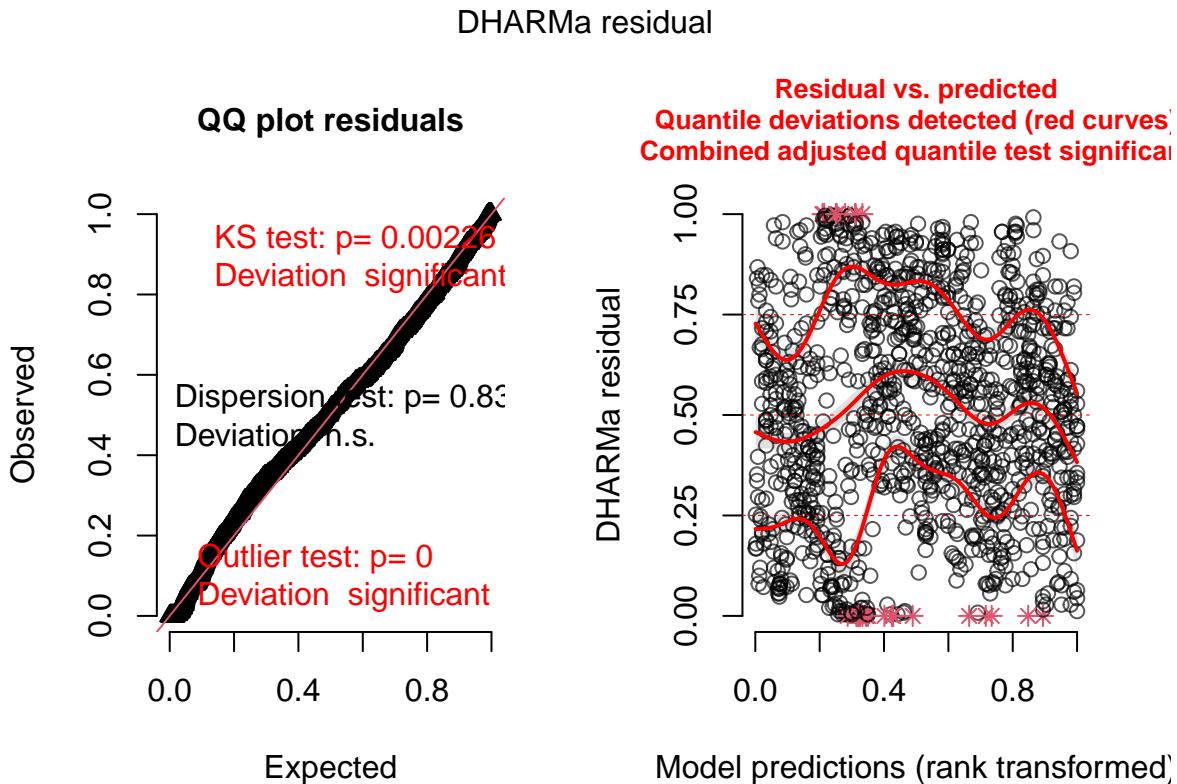
```

capture.output(anova(pc1.response.model.by.incubator, pc1.response.model.by.incubator.s1,
pc1.response.model.by.incubator.s2, pc1.response.model.by.incubator.s3,
pc1.response.model.by.incubator.s4), file = "TableS3.comp.txt")
# Full model preferred
write.csv(as.data.frame(anova(pc1.response.model.by.incubator,
pc1.response.model.by.incubator.s1, pc1.response.model.by.incubator.s2,
pc1.response.model.by.incubator.s3, pc1.response.model.by.incubator.s4)),
file = "TableS3.comp.csv")

capture.output(summary(pc1.response.model.by.incubator), file = "TableS3.best.txt")
write.csv(signif(as.data.frame(summary(pc1.response.model.by.incubator)$coefficients),
3), file = "TableS3.best.csv")

# Visualize residuals
res <- simulateResiduals(pc1.response.model.by.incubator)
plot(res)

```



```

pdf("Fig4A.pdf")
dat <- subset(sub.dat, rampState == 3 & bodyTemp_localAv > 10 &
colony.size.at.tagging < colony.size.cutoff)
beeswarm(bodyTemp_localAv ~ incubator, data = dat, pch = 19,
cex = 0.75, col = c("darkgoldenrod2", "mediumpurple3"), axes = FALSE,
ann = FALSE)
bxplot(bodyTemp_localAv ~ incubator, data = dat, add = TRUE,
probs = 0.5)

```

```

# Experimental block dropped as a random effect for model
# convergence
axis(2)
dev.off()

## pdf
## 2

# Model for above
body.temp.by.group.model <- lmer(bodyTemp_localAv ~ incubator +
  neonic + species + (1 | colony.id/uniqueID) + (1 | experimental.day),
  data = subset(dat, incubator %in% c("inc", "ninc")))
summary(body.temp.by.group.model)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## bodyTemp_localAv ~ incubator + neonic + species + (1 | colony.id/uniqueID) +
##   (1 | experimental.day)
## Data: subset(dat, incubator %in% c("inc", "ninc"))
##
## REML criterion at convergence: 2808.5
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.9044 -0.4702 -0.0190  0.5465  2.9700
##
## Random effects:
## Groups           Name        Variance Std.Dev.
## uniqueID:colony.id (Intercept) 3.360    1.833
## colony.id         (Intercept) 5.587    2.364
## experimental.day  (Intercept) 3.074    1.753
## Residual          4.631    2.152
## Number of obs: 605, groups:
## uniqueID:colony.id, 89; colony.id, 13; experimental.day, 3
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 19.6422   2.2788 11.0351   8.620 3.12e-06 ***
## incubatorninc -3.5470   0.6297 86.0763  -5.633 2.19e-07 ***
## neonicTRUE   -0.2803   1.5882  9.0950  -0.176  0.864
## speciesgriseocolis -0.5001   2.1723  8.6642  -0.230  0.823
## speciesimpatiens  1.3172   2.1125  9.0561   0.624  0.548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) incbtr nnTRUE spcsgr
## incubatrnnc -0.254
## neonicTRUE   -0.358  0.022
## spcsgrscls  -0.600  0.152 -0.209
## specismptns -0.738  0.159  0.141  0.676

```

```

pdf("Fig4B.pdf")
plot_model(pc1.response.model.by.incubator, type = "pred", terms = c("nestTemp",
  "neonic", "incubator"), ci.lvl = 0.5, colors = list(ctrl_color,
  imid_color), axis.title = c(expression(paste("Nest Temperature (",
  degree, "C)", sep = "")), "Spatial Centrality (PC1)"), show.legend = F,
  title = ""))
dev.off()

## pdf
## 2

setwd(output_folder)

# Subset data
sub.dat <- subset(behData, rampState %in% c(1, 5) & cold == TRUE)
sub.dat$rampState <- as.factor(sub.dat$rampState)

# Explicit model for incubator interaction effect, subset
# to small colonies
sub.dat$incubator <- NA
sub.dat$incubator[sub.dat$uniqueID %in% incubator.list] <- "inc"
sub.dat$incubator[sub.dat$uniqueID %in% non.incubator.list] <- "ninc"

tmp.dat <- subset(sub.dat, caste == "worker" & incubator %in%
  c("inc", "ninc") & colony.size.at.tagging < colony.size.cutoff)
pc1.recovery.model.by.incubator <- lmer(PC1 ~ rampState_ch *
  neonic * incubator + species + (1 | colony.id/uniqueID),
  data = tmp.dat, REML = FALSE)
pc1.recovery.model.by.incubator.s1 <- lmer(PC1 ~ rampState_ch *
  neonic + incubator + species + (1 | colony.id/uniqueID),
  data = tmp.dat, REML = FALSE)
pc1.recovery.model.by.incubator.s2 <- lmer(PC1 ~ rampState_ch +
  neonic * incubator + species + (1 | colony.id/uniqueID),
  data = tmp.dat, REML = FALSE)
pc1.recovery.model.by.incubator.s3 <- lmer(PC1 ~ rampState_ch *
  incubator + neonic + species + (1 | colony.id/uniqueID),
  data = tmp.dat, REML = FALSE)
pc1.recovery.model.by.incubator.s4 <- lmer(PC1 ~ rampState_ch +
  neonic + incubator + species + (1 | colony.id/uniqueID),
  data = tmp.dat, REML = FALSE)

# Evaluate models using AIC
capture.output(anova(pc1.recovery.model.by.incubator, pc1.recovery.model.by.incubator.s1,
  pc1.recovery.model.by.incubator.s2, pc1.recovery.model.by.incubator.s3,
  pc1.recovery.model.by.incubator.s4), file = "TableS4.comp.txt")

write.csv(as.data.frame(anova(pc1.recovery.model.by.incubator,
  pc1.recovery.model.by.incubator.s1, pc1.recovery.model.by.incubator.s2,
  pc1.recovery.model.by.incubator.s3, pc1.recovery.model.by.incubator.s4)),
  file = "TableS4.comp.csv")

capture.output(summary(pc1.recovery.model.by.incubator), file = "TableS4.best.txt")
write.csv(as.data.frame(signif(summary(pc1.recovery.model.by.incubator)$coefficients,

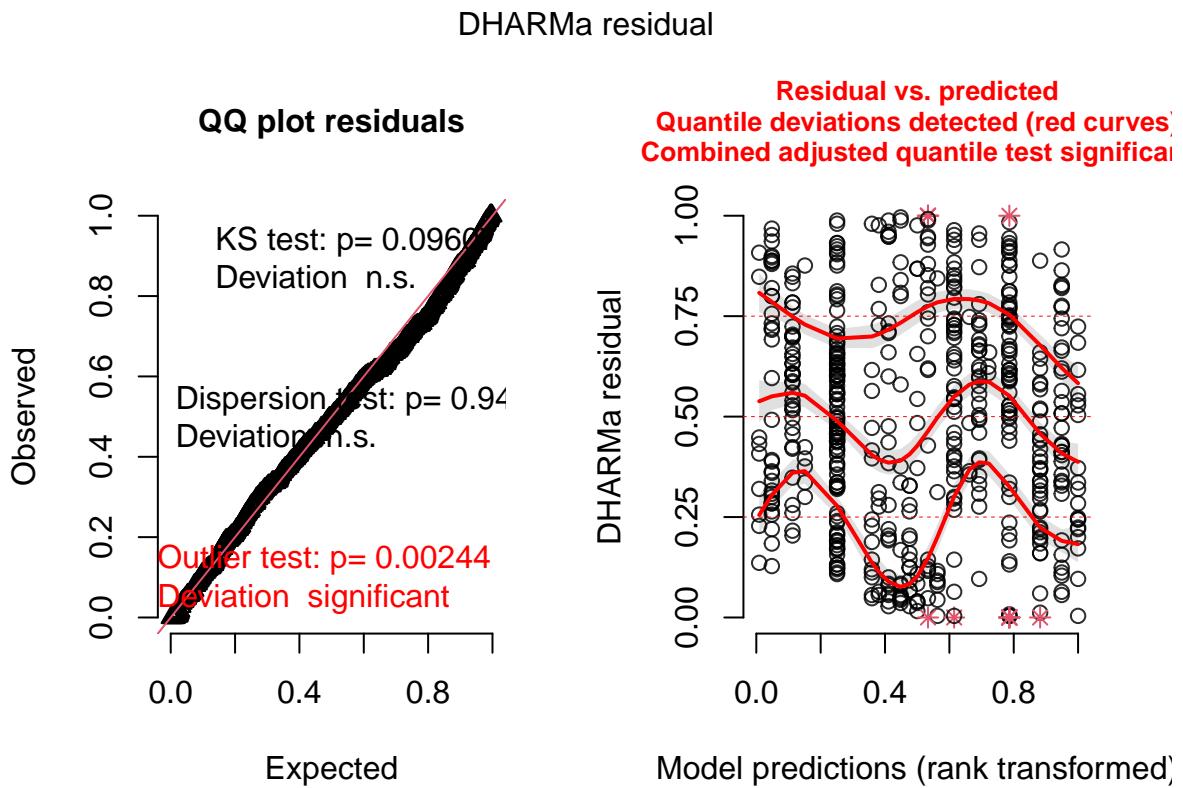
```

```

 3)), file = "TableS4.best.csv")

res <- simulateResiduals(pc1.recovery.model.by.incubator)
plot(res)

```



```

pdf("Fig4C.pdf")
plot_model(pc1.recovery.model.by.incubator, type = "pred", terms = c("rampState_ch",
  "neonic", "incubator"), ci.lvl = 0.5, colors = list(ctrl_color,
  imid_color), axis.title = c("Observation Period", "Spatial Centrality (PC1)"),
  show.legend = F, title = "Recovery response by worker group")
dev.off()

```

```

## pdf
## 2

```

Brood surface temperature analysis

```

setwd(output_folder)
thermData$neonic <- as.factor(thermData$neonic) #Convert to factor
thermData <- subset(thermData, species %in% c("impatiens", "bimaculatus",
  "griseocolis"))
dat <- subset(thermData, tempRampState %in% c(1, 2, 3) & cold ==

```

```

TRUE)
brood.temp.model <- lmer(meanTempBrood_all ~ nestAirTemp * neonic *
  log10(colony.size.at.tagging) + species + (1 | colony.id) +
  (1 | experimental.day), data = dat, REML = FALSE)
brood.temp.model.s1 <- lmer(meanTempBrood_all ~ nestAirTemp *
  neonic + log10(colony.size.at.tagging) + species + (1 | colony.id) +
  (1 | experimental.day), data = dat, REML = FALSE)
brood.temp.model.s2 <- lmer(meanTempBrood_all ~ nestAirTemp +
  neonic * log10(colony.size.at.tagging) + species + (1 | colony.id) +
  (1 | experimental.day), data = dat, REML = FALSE)
brood.temp.model.s3 <- lmer(meanTempBrood_all ~ nestAirTemp *
  log10(colony.size.at.tagging) + neonic + species + (1 | colony.id) +
  (1 | experimental.day), data = dat, REML = FALSE)
brood.temp.model.s4 <- lmer(meanTempBrood_all ~ nestAirTemp +
  neonic + log10(colony.size.at.tagging) + species + (1 | colony.id) +
  (1 | experimental.day), data = dat, REML = FALSE)
brood.temp.model.s5 <- lmer(meanTempBrood_all ~ nestAirTemp *
  log10(colony.size.at.tagging) + species + (1 | colony.id) +
  (1 | experimental.day), data = dat)
brood.temp.model.s6 <- lmer(meanTempBrood_all ~ nestAirTemp +
  log10(colony.size.at.tagging) + species + (1 | colony.id) +
  (1 | experimental.day), data = dat)
brood.temp.model.s7 <- lmer(meanTempBrood_all ~ nestAirTemp +
  species + (1 | colony.id) + (1 | experimental.day), data = dat)

anova(brood.temp.model, brood.temp.model.s1, brood.temp.model.s2,
  brood.temp.model.s3, brood.temp.model.s4, brood.temp.model.s5,
  brood.temp.model.s6, brood.temp.model.s7)

```

```
## refitting model(s) with ML (instead of REML)
```

```

## Data: dat
## Models:
## brood.temp.model.s7: meanTempBrood_all ~ nestAirTemp + species + (1 | colony.id) + (1 | experimental
## brood.temp.model.s6: meanTempBrood_all ~ nestAirTemp + log10(colony.size.at.tagging) + species + (1
## brood.temp.model.s4: meanTempBrood_all ~ nestAirTemp + neonic + log10(colony.size.at.tagging) + speci
## brood.temp.model.s5: meanTempBrood_all ~ nestAirTemp * log10(colony.size.at.tagging) + species + (1
## brood.temp.model.s1: meanTempBrood_all ~ nestAirTemp * neonic + log10(colony.size.at.tagging) + speci
## brood.temp.model.s2: meanTempBrood_all ~ nestAirTemp + neonic * log10(colony.size.at.tagging) + speci
## brood.temp.model.s3: meanTempBrood_all ~ nestAirTemp * log10(colony.size.at.tagging) + neonic + speci
## brood.temp.model: meanTempBrood_all ~ nestAirTemp * neonic * log10(colony.size.at.tagging) + species
##           npar      AIC      BIC  logLik deviance   Chisq Df Pr(>Chisq)
## brood.temp.model.s7    7 1882.0 1912.8 -934.02    1868.0
## brood.temp.model.s6    8 1882.4 1917.5 -933.19    1866.4  1.6722  1  0.1959630
## brood.temp.model.s4    9 1882.7 1922.2 -932.33    1864.7  1.7225  1  0.1893720
## brood.temp.model.s5    9 1866.4 1905.9 -924.21    1848.4 16.2322  0
## brood.temp.model.s1   10 1879.1 1923.0 -929.57    1859.1  0.0000  1  1.0000000
## brood.temp.model.s2   10 1879.5 1923.4 -929.73    1859.5  0.0000  0
## brood.temp.model.s3   10 1866.8 1910.7 -923.40    1846.8 12.6626  0
## brood.temp.model     13 1853.7 1910.8 -913.85    1827.7 19.1093  3  0.0002595
##
## brood.temp.model.s7

```

```

## brood.temp.model.s6
## brood.temp.model.s4
## brood.temp.model.s5
## brood.temp.model.s1
## brood.temp.model.s2
## brood.temp.model.s3
## brood.temp.model      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(brood.temp.model)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula:
## meanTempBrood_all ~ nestAirTemp * neonic * log10(colony.size.at.tagging) +
##   species + (1 | colony.id) + (1 | experimental.day)
## Data: dat
##
##          AIC      BIC  logLik deviance df.resid
## 1853.7  1910.8 -913.8   1827.7     583
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -5.1137 -0.6020 -0.0538  0.5481  4.2937
##
## Random effects:
## Groups           Name        Variance Std.Dev.
## colony.id       (Intercept) 1.2846   1.133
## experimental.day (Intercept) 0.1347   0.367
## Residual         1.1136   1.055
## Number of obs: 596, groups: colony.id, 19; experimental.day, 3
##
## Fixed effects:
##                                         Estimate Std. Error
## (Intercept)                         -5.64799  1.77277
## nestAirTemp                           1.16796  0.05017
## neonicTRUE                            7.30623  2.59121
## log10(colony.size.at.tagging)        6.35811  1.19849
## speciesgriseocolis                  1.13070  0.86862
## speciesimpatiens                     2.37426  0.83510
## nestAirTemp:neonicTRUE                -0.16501  0.07221
## nestAirTemp:log10(colony.size.at.tagging) -0.19438  0.03779
## neonicTRUE:log10(colony.size.at.tagging) -6.87673  1.88551
## nestAirTemp:neonicTRUE:log10(colony.size.at.tagging) 0.16892  0.05554
##                                         df t value Pr(>|t|)
## (Intercept)                      35.00426 -3.186 0.003030
## nestAirTemp                       576.81893 23.282 < 2e-16
## neonicTRUE                         33.60466  2.820 0.008004
## log10(colony.size.at.tagging)     39.77042  5.305 4.53e-06
## speciesgriseocolis               19.40981  1.302 0.208252
## speciesimpatiens                  19.03252  2.843 0.010386
## nestAirTemp:neonicTRUE            577.01269 -2.285 0.022660
## nestAirTemp:log10(colony.size.at.tagging) 576.50162 -5.144 3.69e-07

```

```

## neonicTRUE:log10(colony.size.at.tagging)           36.32269 -3.647 0.000825
## nestAirTemp:neonicTRUE:log10(colony.size.at.tagging) 577.31600  3.042 0.002460
##
## (Intercept)                                **
## nestAirTemp                                 ***
## neonicTRUE                                  **
## log10(colony.size.at.tagging)                ***
## speciesgriseocolis                         *
## speciesimpatiens                           *
## nestAirTemp:neonicTRUE                      *
## nestAirTemp:log10(colony.size.at.tagging)    ***
## neonicTRUE:log10(colony.size.at.tagging)     ***
## nestAirTemp:neonicTRUE:log10(colony.size.at.tagging) **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) nstArT nnTRUE l10(.. spcsgr spcsmp nsAT:TRUE nAT:10 nTRUE:
## nestAirTemp -0.494
## neonicTRUE -0.661  0.339
## lg10(cl...) -0.870  0.524  0.588
## specsgrscls -0.271 -0.004 -0.148  0.030
## specismptns -0.417 -0.001  0.319  0.027  0.519
## nstArT:TRUE  0.346 -0.692 -0.494 -0.364 -0.003 -0.002
## nAT:10(...)  0.470 -0.952 -0.321 -0.550  0.002  0.002  0.659
## nTRUE:10(..  0.638 -0.333 -0.964 -0.636  0.121 -0.269  0.502      0.348
## nAT:TRUE:10 -0.323  0.644  0.474  0.373  0.005  0.001 -0.959      -0.678 -0.524

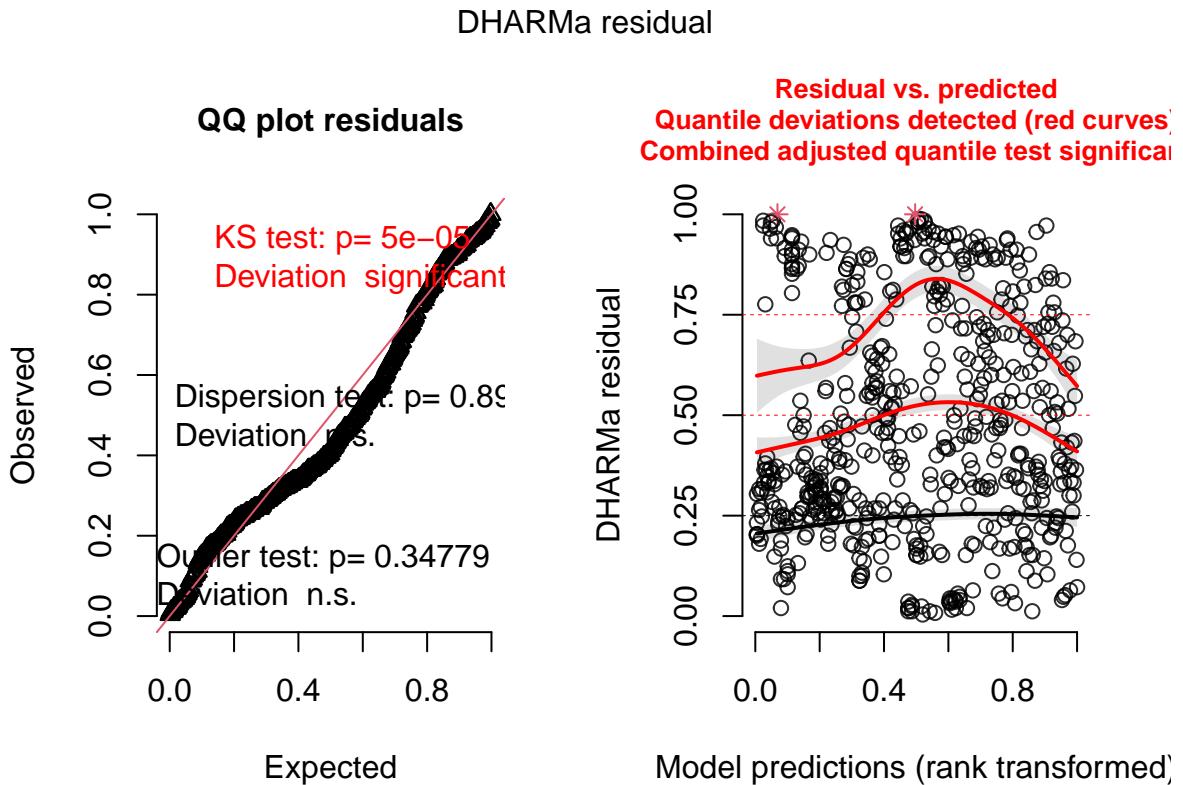
capture.output(anova(brood.temp.model, brood.temp.model.s1, brood.temp.model.s2,
                     brood.temp.model.s3, brood.temp.model.s4, brood.temp.model.s5,
                     brood.temp.model.s6, brood.temp.model.s7), file = "brood.temp.model.txt")

## refitting model(s) with ML (instead of REML)

capture.output(summary(brood.temp.model), file = "brood.temp.best.model.txt")

res <- simulateResiduals(brood.temp.model.s5)
plot(res)

```



```
# Recovery model
dat <- subset(thermData, tempRampState %in% c(1, 5) & cold ==
  TRUE)
dat$tempRampState <- as.factor(dat$tempRampState)
brood.temp.recovery.model <- lmer(meanTempBrood_all ~ tempRampState *
  neonic * log10(colony.size.at.tagging) + species + (1 | colony.id),
  data = dat)
brood.temp.recovery.model.s1 <- lmer(meanTempBrood_all ~ tempRampState *
  neonic + log10(colony.size.at.tagging) + species + (1 | colony.id),
  data = dat)
brood.temp.recovery.model.s2 <- lmer(meanTempBrood_all ~ tempRampState +
  neonic * log10(colony.size.at.tagging) + species + (1 | colony.id),
  data = dat)
brood.temp.recovery.model.s3 <- lmer(meanTempBrood_all ~ tempRampState *
  log10(colony.size.at.tagging) + neonic + species + (1 | colony.id),
  data = dat)
brood.temp.recovery.model.s4 <- lmer(meanTempBrood_all ~ tempRampState +
  neonic + log10(colony.size.at.tagging) + species + (1 | colony.id),
  data = dat)

anova(brood.temp.recovery.model, brood.temp.recovery.model.s1,
  brood.temp.recovery.model.s2, brood.temp.recovery.model.s3,
  brood.temp.recovery.model.s4)
```

```
## refitting model(s) with ML (instead of REML)
```

```

## Data: dat
## Models:
## brood.temp.recovery.model.s4: meanTempBrood_all ~ tempRampState + neonic + log10(colony.size.at.tagg
## brood.temp.recovery.model.s1: meanTempBrood_all ~ tempRampState * neonic + log10(colony.size.at.tagg
## brood.temp.recovery.model.s2: meanTempBrood_all ~ tempRampState + neonic * log10(colony.size.at.tagg
## brood.temp.recovery.model.s3: meanTempBrood_all ~ tempRampState * log10(colony.size.at.tagging) + ne
## brood.temp.recovery.model: meanTempBrood_all ~ tempRampState * neonic * log10(colony.size.at.tagging)
##          npar   AIC   BIC logLik deviance Chisq Df
## brood.temp.recovery.model.s4     8 637.49 664.42 -310.75   621.49
## brood.temp.recovery.model.s1     9 637.26 667.55 -309.63   619.26  2.2323  1
## brood.temp.recovery.model.s2     9 639.47 669.76 -310.73   621.47  0.0000  0
## brood.temp.recovery.model.s3     9 629.74 660.03 -305.87   611.74  9.7294  0
## brood.temp.recovery.model      12 622.73 663.12 -299.36   598.73 13.0102  3
##          Pr(>Chisq)
## brood.temp.recovery.model.s4
## brood.temp.recovery.model.s1  0.135149
## brood.temp.recovery.model.s2
## brood.temp.recovery.model.s3
## brood.temp.recovery.model    0.004615 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(brood.temp.recovery.model)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## meanTempBrood_all ~ tempRampState * neonic * log10(colony.size.at.tagging) +
##   species + (1 | colony.id)
## Data: dat
##
## REML criterion at convergence: 590.9
##
## Scaled residuals:
##   Min    1Q Median    3Q   Max
## -4.0791 -0.5072  0.0107  0.4618  3.2758
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## colony.id (Intercept) 3.0702   1.7522
## Residual           0.7156   0.8459
## Number of obs: 214, groups: colony.id, 20
##
## Fixed effects:
##                               Estimate Std. Error
## (Intercept)                  25.86630  2.37515
## tempRampState5                -0.11623  0.68565
## neonicTRUE                      2.06990  3.46287
## log10(colony.size.at.tagging)  1.01149  1.57061
## speciesgriseocolis            0.09543  1.32472
## speciesimpatiens               0.62465  1.28283
## tempRampState5:neonicTRUE      -3.40789  0.94830
## tempRampState5:log10(colony.size.at.tagging) -0.07475  0.52883
## neonicTRUE:log10(colony.size.at.tagging)      -2.01936  2.44738

```

```

## tempRampState5:neonicTRUE:log10(colony.size.at.tagging) 2.39553 0.72980
##                                         df t value
## (Intercept)                      14.43720 10.890
## tempRampState5                   191.41616 -0.170
## neonicTRUE                        14.53909 0.598
## log10(colony.size.at.tagging)    14.86865 0.644
## speciesgriseocolis              13.29464 0.072
## speciesimpatiens                 13.40236 0.487
## tempRampState5:neonicTRUE        191.18241 -3.594
## tempRampState5:log10(colony.size.at.tagging) 191.85525 -0.141
## neonicTRUE:log10(colony.size.at.tagging)   14.90490 -0.825
## tempRampState5:neonicTRUE:log10(colony.size.at.tagging) 191.49642 3.282
##                                         Pr(>|t|)
## (Intercept)                      2.37e-08 ***
## tempRampState5                   0.865570
## neonicTRUE                        0.559209
## log10(colony.size.at.tagging)    0.529378
## speciesgriseocolis              0.943642
## speciesimpatiens                 0.634172
## tempRampState5:neonicTRUE        0.000415 ***
## tempRampState5:log10(colony.size.at.tagging) 0.887741
## neonicTRUE:log10(colony.size.at.tagging)   0.422319
## tempRampState5:neonicTRUE:log10(colony.size.at.tagging) 0.001223 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) tmpRS5 nnTRUE l10(.. spcsgr spcsmp tmRS5:TRUE tRS5:1 nTRUE:
## tempRmpStt5 -0.179
## neonicTRUE -0.679  0.122
## lg10(cl...) -0.854  0.198  0.579
## specsgrscls -0.308 -0.001 -0.159  0.036
## specismptns -0.484 -0.003  0.382  0.040  0.515
## tmpRS5:TRUE  0.123 -0.723 -0.178 -0.142  0.013  0.013
## tRS5:10(...  0.169 -0.956 -0.115 -0.205 -0.001  0.001  0.691
## nTRUE:10(..  0.673 -0.126 -0.966 -0.648  0.131 -0.336  0.185      0.131
## tRS5:TRUE:1 -0.118  0.693  0.170  0.148 -0.010 -0.009 -0.960      -0.725 -0.192

capture.output(anova(brood.temp.recovery.model, brood.temp.recovery.model.s1,
                     brood.temp.recovery.model.s2, brood.temp.recovery.model.s3,
                     brood.temp.recovery.model.s4), file = "brood.temp.reocovery.model.txt")

## refitting model(s) with ML (instead of REML)

write.csv(as.data.frame(anova(brood.temp.recovery.model, brood.temp.recovery.model.s1,
                               brood.temp.recovery.model.s2, brood.temp.recovery.model.s3,
                               brood.temp.recovery.model.s4)), file = "brood.temp.reocovery.model.csv")

## refitting model(s) with ML (instead of REML)

capture.output(summary(brood.temp.recovery.model), file = "brood.temp.recovery.best.model.txt")
write.csv(signif(summary(brood.temp.recovery.model)$coefficients,

```

```

  3), file = "brood.temp.recovery.best.model.txt")

# Export panels
setwd(output_folder)
pdf("FigS9.pdf")
plot_model(brood.temp.model, type = "pred", terms = c("nestAirTemp",
  "neonic", "colony.size.at.tagging [4, 80]"), colors = list(ctrl_color,
  imid_color), axis.title = c("Nest Temperature (C0", "Spatial Centrality (PC1)"),
  show.legend = F, title = "Brood temperature during cold exposure")
dev.off()

## pdf
## 2

pdf("FigS10.pdf")
plot_model(brood.temp.recovery.model, type = "pred", terms = c("tempRampState",
  "neonic", "colony.size.at.tagging [4, 80]"), colors = list(ctrl_color,
  imid_color), axis.title = c("Observation Period", "Spatial Centrality (PC1)"),
  show.legend = F, title = "Brood temperature during recovery")
dev.off()

## pdf
## 2

```

Proximate behavioral mechanisms

```

setwd(output_folder)

# temp vs distance vs colony size Create subset of data
# during cold exposure
dist.sub <- subset(behData, bodyTemp_localAv > 10 & cold == TRUE &
  rampState == 3 & caste %in% c("worker", "queen"))
dist.sub$distanceToCenterInstantaneous <- dist.sub$distanceToCenterInstantaneous/pixel_conversion
dist.sub$medianDistancetoClosestBrood <- dist.sub$medianDistancetoClosestBrood/pixel_conversion

temp.model <- lmer(bodyTemp_localAv ~ neonic + distanceToCenterInstantaneous +
  log10(colony.size.at.tagging) + (1 | caste) + (1 | colony.id/uniqueID),
  data = dist.sub)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00639594 (tol = 0.002, component 1)

summary(temp.model)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## bodyTemp_localAv ~ neonic + distanceToCenterInstantaneous + log10(colony.size.at.tagging) +

```

```

##      (1 | caste) + (1 | colony.id/uniqueID)
## Data: dist.sub
##
## REML criterion at convergence: 10778.9
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.5674 -0.5938 -0.0675  0.5656  3.5679
##
## Random effects:
## Groups           Name        Variance Std.Dev.
## uniqueID:colony.id (Intercept) 5.0223   2.2410
## colony.id         (Intercept) 4.7105   2.1704
## caste             (Intercept) 0.9879   0.9939
## Residual          7.3867   2.7178
## Number of obs: 2110, groups: uniqueID:colony.id, 384; colony.id, 19; caste, 2
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                14.87078  2.11163 14.96654 7.042 4.04e-06
## neonicTRUE                 -0.47283  1.07283 16.01319 -0.441 0.66530
## distanceToCenterInstantaneous -0.71476  0.04214 2023.06717 -16.961 < 2e-16
## log10(colony.size.at.tagging)  5.15183  1.41992 15.94974  3.628 0.00227
##
## (Intercept) ***
## neonicTRUE
## distanceToCenterInstantaneous ***
## log10(colony.size.at.tagging) **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) nnTRUE dstTCI
## neonicTRUE -0.264
## dstncTCntrI -0.017 -0.041
## lg10(cl...) -0.859 -0.002 -0.058
## optimizer (nloptwrap) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00639594 (tol = 0.002, component 1)

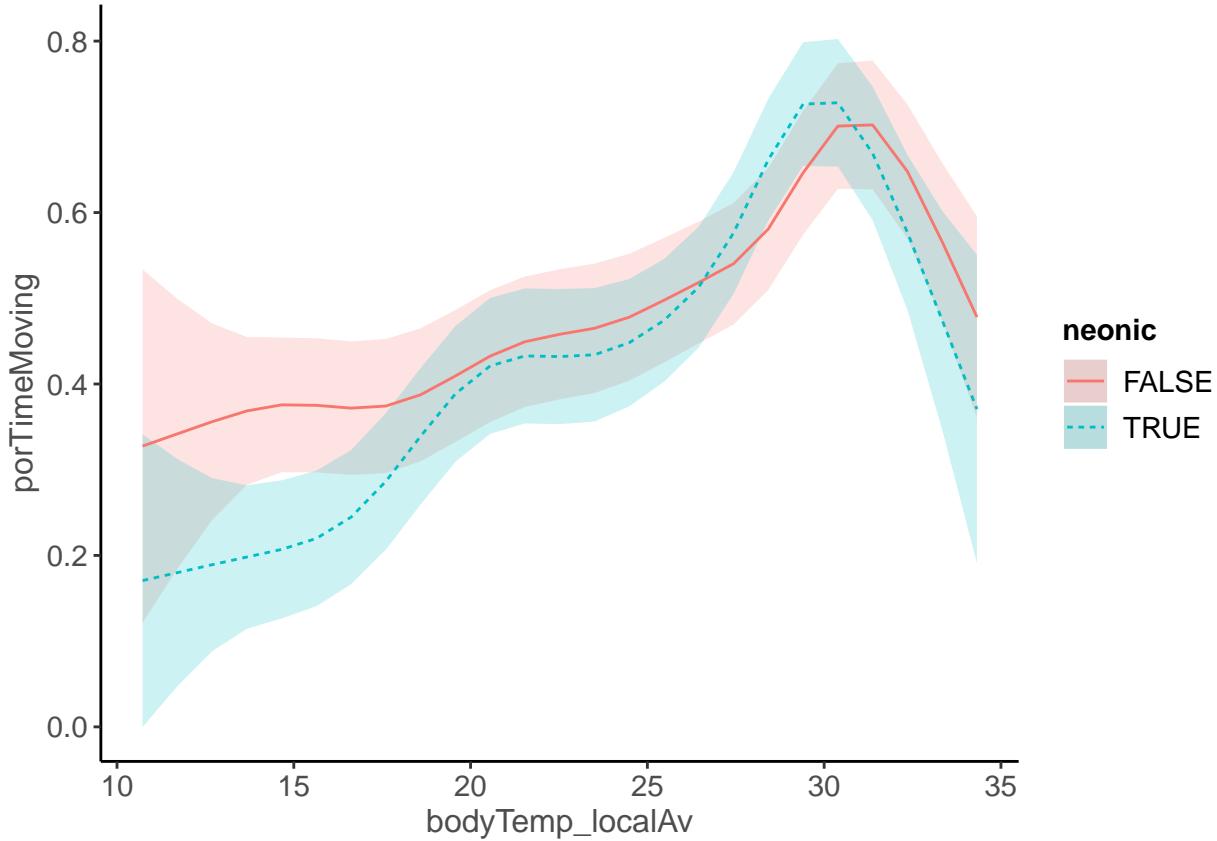
```

```

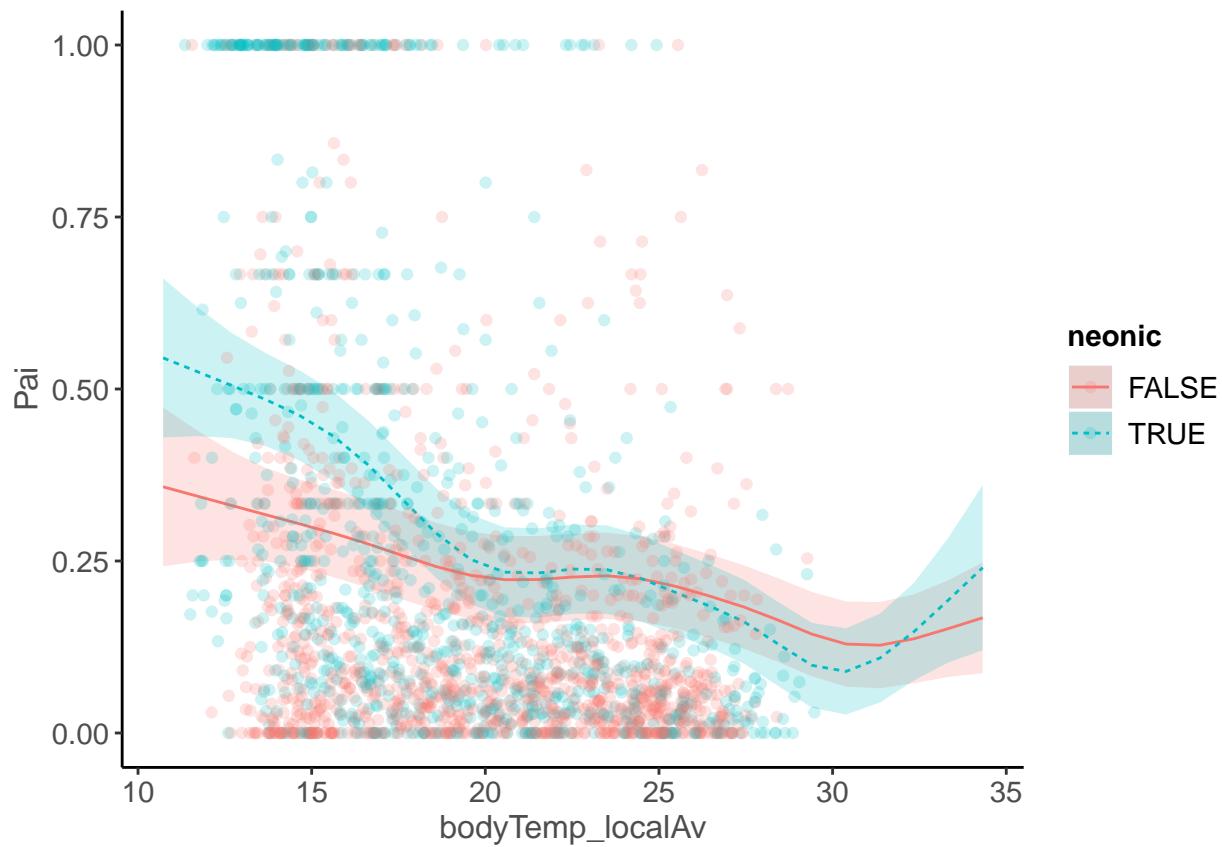
capture.output(summary(temp.model), file = "bodyTemperatureByDistanceModel.txt")

# Overall activity by
tmp <- subset(behData, bodyTemp_localAv > 10 & rampState %in%
  c(1, 2, 3))
tmp$species <- as.factor(tmp$species)
activity.body.temp.model <- bam(porTimeMoving ~ s(colony.id,
  bs = "re") + s(uniqueID, bs = "re") + s(species, bs = "re") +
  s(bodyTemp_localAv, by = neonic), data = tmp)
plot_smooths(activity.body.temp.model, bodyTemp_localAv, neonic)

```



```
# Activity state switching vs body temperature
pai.body.temp.model <- bam(Pai ~ s(colony.id, bs = "re") + s(uniqueID,
  bs = "re") + s(bodyTemp_localAv, by = neonic) + s(species,
  bs = "re"), data = tmp)
plot_smooths(pai.body.temp.model, bodyTemp_localAv, neonic) +
  geom_point(data = dist.sub, aes(x = bodyTemp_localAv, y = Pai,
    colour = neonic), alpha = 0.2)
```



```

### Figure 3
pdf("Fig3_a.pdf")
# plot_smooths(body.temp.by.dist.model,
# distanceToCenterInstantaneous, small.colony)
plot_model(temp.model, type = "pred", terms = c("colony.size.at.tagging",
  "distanceToCenterInstantaneous[1,5]"), ci.lvl = 0.5, title = "Colony size and body temperature",
  axis.title = c("Colony Size", "Body temperature (C)"), colors = list("goldenrod1",
  "sienna"))
dev.off()

## pdf
## 2

pdf("Fig3_b.pdf")
plot_smooths(pai.body.temp.model, bodyTemp_localAv, neonic) +
  geom_point(data = dist.sub, aes(x = bodyTemp_localAv, y = Pai,
    colour = neonic), alpha = 0.2, size = 0.5) + xlim(11,
  30)

## Warning: Removed 12 rows containing missing values ('geom_path()').

dev.off()

```

```

## pdf
## 2

```

```

pdf("Fig3_c.pdf")
plot_smooths(activity.body.temp.model, bodyTemp_localAv, neonic) +
  ylim(0, 1) + geom_point(data = dist.sub, aes(x = bodyTemp_localAv,
    y = porTimeMoving, colour = neonic), alpha = 0.2, size = 0.5) +
  xlim(11, 30)

## Warning: Removed 12 rows containing missing values ('geom_path()').

dev.off()

## pdf
## 2

# Colony size vs. activity
colony.sizes <- aggregate(behData$colony.size.at.tagging, list(behData$colony.id),
  FUN = mean)[, 2]
mean.activity <- aggregate(behData$porTimeMoving, list(behData$colony.id),
  FUN = mean)[, 2]
activity.size.model <- gam(mean.activity ~ s(colony.sizes))
summary(activity.size.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean.activity ~ s(colony.sizes)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.66940   0.01714   39.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df F p-value
## s(colony.sizes) 4.082 4.913 1.97 0.112
##
## R-sq.(adj) = 0.159 Deviance explained = 24.5%
## GCV = 0.013743 Scale est. = 0.01204 n = 41

# Colony size vs. interaction rate
colony.sizes <- aggregate(behData$colony.size.at.tagging, list(behData$colony.id),
  FUN = mean)[, 2]
interaction.rate <- aggregate(behData$meanInteractionRate * behData$colony.size.at.tagging,
  list(behData$colony.id), FUN = mean)[, 2]
interaction.size.model <- gam(interaction.rate ~ s(colony.sizes))
summary(interaction.size.model)

##
## Family: gaussian

```

```

## Link function: identity
##
## Formula:
## interaction.rate ~ s(colony.sizes)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.44636   0.02894 15.43 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df    F p-value
## s(colony.sizes) 1     1 72.59 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.642 Deviance explained =  65%
## GCV = 0.036095 Scale est. = 0.034334 n = 41

setwd(output_folder)
png("FigS8.png", width = 640, height = 300)
a <- plot_smooths(activity.size.model, colony.sizes) + xlab("Colony Size") +
  ylab("Mean activity \n(proportion of time moving)")
b <- plot_smooths(interaction.size.model, colony.sizes) + xlab("Colony Size") +
  ylab("Interaction Rate \n(physical contact/frame)")
grid.arrange(a, b, nrow = 1)
dev.off()

## pdf
## 2

```

Microcolony size-manipulation experiments

Load and clean data

```

block_1 <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
block_2 <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
block_3 <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
metadata <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/
mc_data <- rbind(block_1, block_2, block_3)
head(mc_data)

##   X mean_min_distance_to_brood num_brood_interactions mean_dist_other_beans
## 1 0           1.617829                  0            2.430178
## 2 1           6.518312                  0            8.073369
## 3 2           1.622784                  0            3.783518
## 4 3           0.817224                 73            4.001826
## 5 4           1.681611                  0            4.743056
## 6 5           6.861974                  0            6.188042

```

```

##   total_social_interactions countable_interaction_frames activity
## 1                      10                  22        0
## 2                      0                   18        0
## 3                     67                  153        0
## 4                     67                  170        0
## 5                      0                   44        0
## 6                      0                   22        0
##   speed_when_moving distance_to_social_center frames_tracked tag_ids
## 1            NA          0.5888912           5       16
## 2            NA          7.6145221           6       17
## 3            NA          0.5811764           62      22
## 4            NA          0.4717314           73      33
## 5            NA          3.3990098           14      36
## 6            NA          5.5870597           5       37
##   video_file block colony timestamp
## 1 microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 2 microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 3 microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 4 microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 5 microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 6 microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
##   time_num nest_temp
## 1 272.6542     16.583
## 2 272.6542     16.583
## 3 272.6542     16.583
## 4 272.6542     16.583
## 5 272.6542     16.583
## 6 272.6542     16.583

### Add metadata to data
for (i in 1:length(metadata[, 1])) {
  idx = mc_data$block == metadata$block[i] & mc_data$colony ==
    metadata$colony_number[i]
  mc_data$treatment[idx] <- metadata$treatment[i]
  mc_data$colony_size[idx] <- metadata$colony_size[i]
  mc_data$source_colony[idx] <- metadata$source_colony[i]

}

# Convert to factors
mc_data$block <- as.factor(mc_data$block)
mc_data$colony_size <- as.factor((mc_data$colony_size))
mc_data$source_colony <- as.factor(mc_data$source_colony)
mc_data$unique_colony <- paste(mc_data$block, mc_data$colony,
  sep = "_")
mc_data$tag_ids <- as.factor(mc_data$tag_ids)
mc_data$unique.id <- paste(mc_data$tag_ids, mc_data$unique_colony,
  sep = "_")
mc_data$unique.id <- as.factor(mc_data$unique.id)

### Clean up mc_data Clear out faulty temperature readings
mc_data$nest_temp[mc_data$nest_temp < 5 | mc_data$nest_temp >
  35] <- NaN
# Log transform speed

```

```

mc_data$speed_when_moving <- log10(mc_data$speed_when_moving)

# Square root transform brood interactions
mc_data$num_brood_interactions <- sqrt(mc_data$num_brood_interactions)

# Get tag lists from experiments
mc_taglist <- read.csv("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressor")
# Convert to factors
attach(mc_taglist)
valid.ids <- paste(tag.ID, block, colony, sep = "_")
detach(mc_taglist)
mc_data <- subset(mc_data, unique.id %in% valid.ids & frames_tracked >
  20)
dim(mc_data) #Size of data -> # of unique observations

## [1] 106164      22

length(unique(mc_data$unique.id)) #unique bees

## [1] 237

# Remove two sets of erroneous data
rm.ind.1 <- mc_data$unique_colony == "1_12" & mc_data$time_num <
  270 #Some erroneous timestamps
rm.ind.2 <- mc_data$unique_colony == "2_3" & mc_data$time_num >
  280 #Period of artificially high readings, removing
rm.ind <- rm.ind.1 | rm.ind.2
mc_data <- mc_data[!rm.ind, ]

# Remove some abnormally high temp readings (a few hundred)
# mc_data <- subset(mc_data, nest_temp < 28) Separate pre
# and post
cols <- unique(mc_data$unique_colony)
rampThresh <- 20

define_ramp_states <- function(sd, plt = FALSE) {
  minT <- min(sd$nest_temp, na.rm = TRUE)
  minT_time <- sd$time_num[sd$nest_temp == minT][1]
  sd$rampState[sd$nest_temp > 20 & sd$time_num < minT_time] <- "pre"
  sd$rampState[sd$nest_temp > 20 & sd$time_num > minT_time] <- "post"
  sd$rampState[sd$nest_temp <= 20] <- "ramp"

  sd$cl[sd$ramp == "pre"] <- "coral1"
  sd$cl[sd$ramp == "ramp"] <- "cadetblue"
  sd$cl[sd$ramp == "post"] <- "coral4"
  plt_data <- subset(sd, rampState %in% c("pre", "ramp", "post"))
  if (plt == TRUE) {
    plot(nest_temp ~ time_num, data = plt_data, col = plt_data$cl,
      pch = 19, cex = 0.5)
    title(unique(sd$unique_colony))
  }
  return(sd$rampState)
}

```

```

}

for (i in 1:length(cols)) {
  cur.ind <- mc_data$unique_colony == cols[i]
  sd <- mc_data[cur.ind, ]
  rampState <- define_ramp_states(sd)
  mc_data$rampState[cur.ind] <- rampState
}

# mc_data <- subset(mc_data, rampState %in% c('pre',
# 'ramp'))

```

PCA analysis

```

spat.vars <- c("mean_dist_other_beans", "mean_min_distance_to_brood",
  "distance_to_social_center", "total_social_interactions",
  "num_brood_interactions")

pc.dat <- mc_data[, spat.vars]
cmplt_ind <- complete.cases(pc.dat)
pc.dat <- pc.dat[cmplt_ind, ]
mc_data <- mc_data[cmplt_ind, ]
mc.nest.beh.pca <- prcomp(pc.dat, scale = TRUE, center = TRUE)
mc_data$pc1 <- -mc.nest.beh.pca$x[, 1] #Inverse sign for consistency with other analysis, so higher pc

# Yeo-Johnson transformation to a normal distribution
tf <- yeojohnson(mc_data$pc1)
mc_data$pc1 <- predict(tf)

head(mc_data)

##      X mean_min_distance_to_brood num_brood_interactions mean_dist_other_beans
## 3     2           1.6227837          0.0000000            3.783518
## 4     3           0.8172240          8.544004            4.001826
## 8     7           2.7178458          0.0000000            5.169691
## 11    10          0.9112662          8.774964            4.162098
## 12    11          1.1629358          8.831761            4.142163
## 14    13          3.9095601          0.0000000            5.309894
##      total_social_interactions countable_interaction_frames activity
## 3                 67                  153 0.0000000
## 4                 67                  170 0.0000000
## 8                 0                  170 0.0000000
## 11                17                  108 0.8409091
## 12                20                  140 0.7391304
## 14                0                  140 0.0000000
##      speed_when_moving distance_to_social_center frames_tracked tag_ids
## 3                 NA          0.5811764          62        22
## 4                 NA          0.4717314          73        33
## 8                 NA          1.9373203          73        47
## 11               -0.2440059         2.2086506          47        22

```

```

## 12      -0.6191297          0.7177818      72      33
## 14           NA          2.1745005      74      47
##               video_file block colony      timestamp
## 3  microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 4  microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 8  microcol_1-2021-09-30_15-42-01.jpeg    1      1 2021-09-30_15-42-01
## 11 microcol_1-2021-09-30_12-42-01.jpeg    1      1 2021-09-30_12-42-01
## 12 microcol_1-2021-09-30_12-42-01.jpeg    1      1 2021-09-30_12-42-01
## 14 microcol_1-2021-09-30_12-42-01.jpeg    1      1 2021-09-30_12-42-01
##   time_num nest_temp treatment colony_size source_colony unique_colony
## 3  272.6542    16.583  neonic        4            3       1_1
## 4  272.6542    16.583  neonic        4            3       1_1
## 8  272.6542    16.583  neonic        4            3       1_1
## 11 272.5292    22.529  neonic        4            3       1_1
## 12 272.5292    22.529  neonic        4            3       1_1
## 14 272.5292    22.529  neonic        4            3       1_1
##   unique.id rampState      pc1
## 3     22_1_1      ramp 1.0926685
## 4     33_1_1      ramp 1.4592548
## 8     47_1_1      ramp 0.4305023
## 11    22_1_1      pre  1.0670588
## 12    33_1_1      pre  1.1926445
## 14    47_1_1      pre  0.2890360

# plot(pc1~nest_temp, data = mc_data, pch = 19, cex = 0.1,
# col=rgb(0.6,0.2, 0.3, 0.1))

mc.pc.loadings <- -mc.nest.beh.pca$rotation[, 1:2]
mc.pc.loadings <- mc.pc.loadings[order(mc.pc.loadings[, 1]), ,
  ]
mc.pc.loadings <- as.data.frame(mc.pc.loadings)
mc.pc.loadings$variable <- rownames(mc.pc.loadings)
setwd(output_folder)
pdf("FigS5_mc_pcloadings.pdf")
ggplot(data = mc.pc.loadings, aes(x = reorder(variable, +PC1),
  y = PC1, fill = PC1)) + geom_bar(stat = "identity") + coord_flip() +
  scale_fill_distiller(palette = "RdYlBu") + theme_minimal()
dev.off()

## pdf
## 2

##Models Statistical models

# Overall temperature response model in control colonies
mc.pc1.response.model <- lmer(pc1 ~ nest_temp + (1 | block) +
  (1 | colony) + (1 | unique.id), data = subset(mc_data, treatment ==
  "control"), REML = FALSE)
summary(mc.pc1.response.model)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: pc1 ~ nest_temp + (1 | block) + (1 | colony) + (1 | unique.id)

```

```

##      Data: subset(mc_data, treatment == "control")
##
##      AIC      BIC  logLik deviance df.resid
## 105180.9 105233.6 -52584.4 105168.9     48830
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -4.5512 -0.5910  0.0348  0.6047  4.1271
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## unique.id (Intercept) 0.3645    0.6037
## colony    (Intercept) 0.1067    0.3266
## block     (Intercept) 0.2837    0.5326
## Residual           0.4976    0.7054
## Number of obs: 48836, groups: unique.id, 115; colony, 10; block, 3
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 7.825e-01 3.324e-01 3.517e+00 2.354   0.0871 .
## nest_temp   -2.072e-02 7.361e-04 4.873e+04 -28.148 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## nest_temp -0.043

# Spatial Centrality model
mc.pc1.full.model <- lmer(pc1 ~ nest_temp * treatment * colony_size +
  (1 | block) + (1 | colony/unique.id) + (1 | source_colony) +
  (1 | rampState), data = subset(mc_data, rampState %in% c("pre",
  "ramp")), REML = FALSE)
mc.pc1.full.model.s1 <- lmer(pc1 ~ nest_temp * treatment + colony_size +
  (1 | block) + (1 | colony/unique.id) + (1 | source_colony) +
  (1 | rampState), data = subset(mc_data, rampState %in% c("pre",
  "ramp")), REML = FALSE)
mc.pc1.full.model.s2 <- lmer(pc1 ~ nest_temp + treatment * colony_size +
  (1 | block) + (1 | colony/unique.id) + (1 | source_colony) +
  (1 | rampState), data = subset(mc_data, rampState %in% c("pre",
  "ramp")), REML = FALSE)
mc.pc1.full.model.s3 <- lmer(pc1 ~ nest_temp * colony_size +
  treatment + (1 | block) + (1 | colony/unique.id) + (1 | source_colony) +
  (1 | rampState), data = subset(mc_data, rampState %in% c("pre",
  "ramp")), REML = FALSE)
mc.pc1.full.model.s4 <- lmer(pc1 ~ nest_temp + treatment + colony_size +
  (1 | block) + (1 | colony/unique.id) + (1 | source_colony) +
  (1 | rampState), data = subset(mc_data, rampState %in% c("pre",
  "ramp")), REML = FALSE)

anova(mc.pc1.full.model, mc.pc1.full.model.s1, mc.pc1.full.model.s2,
  mc.pc1.full.model.s3, mc.pc1.full.model.s4)

## Data: subset(mc_data, rampState %in% c("pre", "ramp"))

```

```

## Models:
## mc.pc1.full.model.s4: pc1 ~ nest_temp + treatment + colony_size + (1 | block) + (1 | colony/unique.id)
## mc.pc1.full.model.s1: pc1 ~ nest_temp * treatment + colony_size + (1 | block) + (1 | colony/unique.id)
## mc.pc1.full.model.s2: pc1 ~ nest_temp + treatment * colony_size + (1 | block) + (1 | colony/unique.id)
## mc.pc1.full.model.s3: pc1 ~ nest_temp * colony_size + treatment + (1 | block) + (1 | colony/unique.id)
## mc.pc1.full.model: pc1 ~ nest_temp * treatment * colony_size + (1 | block) + (1 | colony/unique.id) -
##                      npar   AIC   BIC logLik deviance Chisq Df Pr(>Chisq)
## mc.pc1.full.model.s4  10 75525 75611 -37753    75505
## mc.pc1.full.model.s1  11 75527 75621 -37752    75505  0.5088  1     0.4757
## mc.pc1.full.model.s2  11 75520 75615 -37749    75498  6.1940  0
## mc.pc1.full.model.s3  11 75522 75617 -37750    75500  0.0000  0
## mc.pc1.full.model     14 75505 75625 -37738    75477 23.3999  3  3.333e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(mc.pc1.full.model)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: pc1 ~ nest_temp * treatment * colony_size + (1 | block) + (1 |
##           colony/unique.id) + (1 | source_colony) + (1 | rampState)
## Data: subset(mc_data, rampState %in% c("pre", "ramp"))
##
##      AIC      BIC logLik deviance df.resid
## 75504.5 75625.3 -37738.3 75476.5     41302
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -5.4378 -0.5369  0.0138  0.5468  4.8640
##
## Random effects:
## Groups            Name        Variance Std.Dev.
## unique.id:colony (Intercept) 0.349369 0.59107
## colony            (Intercept) 0.048065 0.21924
## source_colony     (Intercept) 0.157256 0.39655
## block             (Intercept) 0.483008 0.69499
## rampState         (Intercept) 0.003474 0.05894
## Residual          0.353137 0.59425
## Number of obs: 41316, groups:
## unique.id:colony, 235; colony, 13; source_colony, 3; block, 3; rampState, 2
##
## Fixed effects:
##                               Estimate Std. Error      df
## (Intercept)                1.101e+00  4.865e-01 4.102e+00
## nest_temp                  -1.174e-02  1.871e-03 5.246e+03
## treatmentneonic             -7.023e-01  1.500e-01 2.328e+02
## colony_size16              -5.694e-01  1.661e-01 1.152e+02
## nest_temp:treatmentneonic  5.430e-03  1.931e-03 4.112e+04
## nest_temp:colony_size16     2.379e-03  1.738e-03 4.111e+04
## treatmentneonic:colony_size16 7.068e-01  2.066e-01 1.403e+02
## nest_temp:treatmentneonic:colony_size16 -9.565e-03  2.382e-03 4.112e+04
##                                         t value Pr(>|t|)
## (Intercept)                    2.263  0.084694 .
## nest_temp                     -6.274 3.81e-10 ***
```

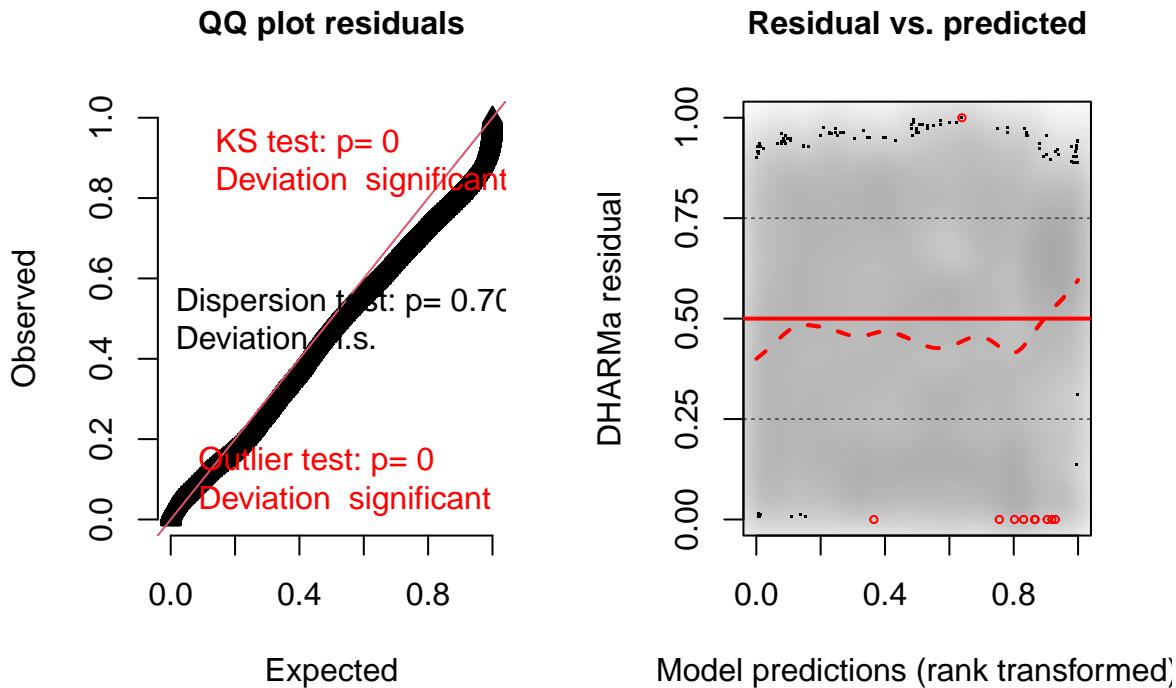
```

## treatmentneonic          -4.683 4.80e-06 ***
## colony_size16            -3.428 0.000843 ***
## nest_temp:treatmentneonic 2.812 0.004923 **
## nest_temp:colony_size16    1.369 0.171075
## treatmentneonic:colony_size16 3.421 0.000817 ***
## nest_temp:treatmentneonic:colony_size16 -4.016 5.94e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) nst_tm trtmnt cln_16 nst_t: n:_16 tr:_16
## nest_temp      -0.067
## treatmntnnc   -0.141  0.126
## colony_sz16   -0.155  0.117  0.533
## nst_tmprt     0.037 -0.567 -0.218 -0.108
## nst_tm:_16    0.041 -0.635 -0.132 -0.177  0.600
## trtmntn:_16   0.112 -0.093 -0.722 -0.742  0.159  0.142
## nst_tm::_16   -0.030  0.463  0.177  0.129 -0.811 -0.730 -0.195

# Evaluate model fit
res <- simulateResiduals(mc.pc1.full.model)
plot(res)

```

DHARMA residual



```
setwd(output_folder)
```

```

capture.output(anova(mc.pc1.full.model, mc.pc1.full.model.s1,
  mc.pc1.full.model.s2, mc.pc1.full.model.s3, mc.pc1.full.model.s4),
  file = "TableS6.comp.txt")
write.csv(as.data.frame(anova(mc.pc1.full.model, mc.pc1.full.model.s1,
  mc.pc1.full.model.s2, mc.pc1.full.model.s3, mc.pc1.full.model.s4)),
  "TableS6.comp.csv")

capture.output(summary(mc.pc1.full.model), file = "TableS6.best")
write.csv(signif(as.data.frame(summary(mc.pc1.full.model)$coefficients),
  3), file = "TableS6.best.csv")

pdf("Fig6_de.pdf")
p1 <- plot_model(mc.pc1.full.model, type = "pred", terms = c("nest_temp",
  "treatment", "colony_size"), ci.lvl = 0, colors = list(ctrl_color,
  imid_color), axis.title = c("Nest temperature (C)", "Spatial Centrality (PC1)"),
  show.legend = F, title = "Microcolony responses")
dev.off()

## pdf
## 2

pdf("FigS7.pdf")
plot_model(mc.pc1.response.model, type = "eff", title = "microcolony temperature reponse",
  axis.title = c("Nest temperature (C)", "Spatial Centrality (PC1)"))

## $nest_temp

# ggplot(subset(mc_data, frames_tracked>30), aes(x =
# nest_temp, y=pc1)) + geom_smooth()
# plot_smooths(mc.pc1.gam.resp.model, nest_temp)
dev.off()

## pdf
## 2

```

Colony demography

Load and clean data

```

## Visualize growth data

## Growth models

gr.plot <- function(sd) {
  plot(1, 1, xlim = c(0.8, 2.2), ylim = c(0, 2.2), axes = FALSE,
    ann = FALSE)
  for (i in 1:length(sd[, 1])) {
    lines(c(1, 2), log10(c(sd$colony.size.at.tagging[i],
      sd$total[i])), lwd = 1.5)
    points(c(1, 2), log10(c(sd$colony.size.at.tagging[i],

```

```

        sd$total[i])), pch = 19)
axis(2, at = log10(c(seq(1, 10, seq(10, 100, 10), seq(100,
1000, 100))), labels = FALSE)
axis(2, at = log10(c(1, 10, 100, 1000)), labels = as.character(c(1,
10, 100, 1000)), lwd = 0, las = 2)
}
setwd(output_folder)
pdf("FigS6_growth_charts.pdf")
par(mfcol = c(1, 4))
sd <- subset(demog.data, cold == 0 & neonic == 0)
gr.plot(sd)
title("control")

sd <- subset(demog.data, cold == 1 & neonic == 0)
gr.plot(sd)
title("no cold + imidacloprid")

sd <- subset(demog.data, cold == 0 & neonic == 1)
gr.plot(sd)
title("cold + imidacloprid-free")

sd <- subset(demog.data, cold == 1 & neonic == 1)
gr.plot(sd)
title("cold + imidacloprid")
dev.off()

```

```

## pdf
##    2

```

```

custom_plot_log <- function(out, ylims, yticks) {
  print(ylims)
  print(yticks)
  out$data$predicted <- log10(out$data$predicted)
  out$data$conf.low <- log10(out$data$conf.low)
  out$data$conf.high <- log10(out$data$conf.high)
  # print(out$data$predicted) plot(out, colors =
  # c(ctrl_color, imid_color))
  ctrl_ind <- c(1, 3)  #ctrl first, cold second
  neonic_ind <- c(2, 4)  #ctrl first, cold second
  offset <- 0.05
  plot(c(1, 2) - offset, out$data$predicted[ctrl_ind], col = ctrl_color,
       pch = 19, xlim = c(0.7, 2.3), ylim = ylims, axes = FALSE,
       xlab = ("Temperature Treatment"), ylab = ("Final colony size"))
  axis(1, labels = c("Control", "Cold"), at = c(1, 2))
  axis(2, at = log10(c(seq(1, 10, seq(10, 100, 10), seq(100,
1000, 100))), labels = FALSE)
  axis(2, at = yticks, labels = as.character(yticks), lwd = 0,
       las = 2)
  lines(c(1, 2) - offset, out$data$predicted[ctrl_ind], col = ctrl_color,
       lty = 2)
  # Add CIs
  lines(c(1, 1) - offset, c(out$data$conf.low[ctrl_ind[1]],

```

```

        out$data$conf.high[ctrl_ind[1]]), col = ctrl_color)
lines(c(2, 2) - offset, c(out$data$conf.low[ctrl_ind[2]],
                           out$data$conf.high[ctrl_ind[2]]), col = ctrl_color)

points(c(1, 2) + offset, out$data$predicted[neonic_ind],
       col = imid_color, pch = 19)
axis(1, labels = c("Control", "Cold"), at = c(1, 2))
lines(c(1, 2) + offset, out$data$predicted[neonic_ind], col = imid_color,
      lty = 2)
# Add CIs
lines(c(1, 1) + offset, c(out$data$conf.low[neonic_ind[1]],
                           out$data$conf.high[neonic_ind[1]]), col = imid_color)
lines(c(2, 2) + offset, c(out$data$conf.low[neonic_ind[2]],
                           out$data$conf.high[neonic_ind[2]]), col = imid_color)
}

setwd(output_folder)

gen.model <- glmer(total ~ log10(colony.size.at.tagging) * cold *
  neonic + (1 + log10(colony.size.at.tagging) | Species) +
  (1 | experimentalBlock) + (1 | beePiPosition), data = subset(demog.data,
  colony.size.at.tagging > 1 & alive.at.tagging == 1 & Species !=
  "perplexus"), family = "poisson")
summary(gen.model)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: total ~ log10(colony.size.at.tagging) * cold * neonic + (1 +
##   log10(colony.size.at.tagging) | Species) + (1 | experimentalBlock) +
##   (1 | beePiPosition)
## Data: subset(demog.data, colony.size.at.tagging > 1 & alive.at.tagging ==
##   1 & Species != "perplexus")
##
##      AIC      BIC  logLik deviance df.resid
##      535.8    561.9   -254.9     509.8      42
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -2.9859 -0.9363  0.2517  0.7687  3.0713
##
## Random effects:
##   Groups            Name        Variance Std.Dev. Corr
##   experimentalBlock (Intercept) 0.39236  0.6264
##   beePiPosition     (Intercept) 0.02637  0.1624
##   Species           (Intercept) 0.69290  0.8324
##                   log10(colony.size.at.tagging) 0.14039  0.3747  -0.64
##   Number of obs: 55, groups: experimentalBlock, 15; beePiPosition, 4; Species, 3
##
## Fixed effects:
##                               Estimate Std. Error z value
## (Intercept)                  0.7170    0.5800  1.236
## log10(colony.size.at.tagging) 2.1625    0.3001  7.207

```

```

## cold1                      0.8116    0.1929   4.208
## neonic1                     0.2694    0.1970   1.367
## log10(colony.size.at.tagging):cold1      -0.5054    0.1270  -3.981
## log10(colony.size.at.tagging):neonic1      -0.2294    0.1430  -1.604
## cold1:neonic1                 -1.1464    0.2932  -3.910
## log10(colony.size.at.tagging):cold1:neonic1  0.7007    0.1932   3.628
##
## Pr(>|z|)
## (Intercept)                  0.216336
## log10(colony.size.at.tagging) 5.72e-13 ***
## cold1                         2.58e-05 ***
## neonic1                        0.171500
## log10(colony.size.at.tagging):cold1 6.87e-05 ***
## log10(colony.size.at.tagging):neonic1 0.108622
## cold1:neonic1                9.24e-05 ***
## log10(colony.size.at.tagging):cold1:neonic1 0.000286 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                               (Intr) lg10(...) cold1  neonic1
## lg10(cl...)              -0.697
## cold1                      -0.200  0.222
## neonic1                     -0.209  0.260   0.474
## lg10(clny.sz.t.tggng):c1  0.173 -0.191  -0.964 -0.431
## lg10(clny.sz.t.tggng):n1  0.192 -0.273  -0.356 -0.935
## cold1:nenc1                 0.076 -0.089  -0.568 -0.587
## l10(...):1:                  -0.065  0.085   0.523  0.594
##                               lg10(clny.sz.t.tggng):c1 lg10(clny.sz.t.tggng):n1
## lg10(cl...)
## cold1
## neonic1
## lg10(clny.sz.t.tggng):c1
## lg10(clny.sz.t.tggng):n1  0.325
## cold1:nenc1                 0.530          0.527
## l10(...):1:                  -0.515         -0.603
##                               cld1:1
## lg10(cl...)
## cold1
## neonic1
## lg10(clny.sz.t.tggng):c1
## lg10(clny.sz.t.tggng):n1
## cold1:nenc1
## l10(...):1:                  -0.960

# Best fitting model based on DHARMA examination of
# residuals, but currently removing 'growth length' as a
# factor

gen.model.s1 <- glmer(total ~ log10(colony.size.at.tagging) *
  cold + neonic + (1 + log10(colony.size.at.tagging) | Species) +
  (1 | experimentalBlock) + (1 | beePiPosition), data = subset(demog.data,
  colony.size.at.tagging > 1 & alive.at.tagging == 1 & Species !=
  "perplexus"), family = "poisson")

```

```

gen.model.s2 <- glmer(total ~ log10(colony.size.at.tagging) +
  cold * neonic + (1 + log10(colony.size.at.tagging) | Species) +
  (1 | experimentalBlock) + (1 | beePiPosition), data = subset(demog.data,
  colony.size.at.tagging > 1 & alive.at.tagging == 1 & Species !=
  "perplexus"), family = "poisson")

gen.model.s3 <- glmer(total ~ log10(colony.size.at.tagging) *
  neonic + cold + (1 + log10(colony.size.at.tagging) | Species) +
  (1 | experimentalBlock) + (1 | beePiPosition), data = subset(demog.data,
  colony.size.at.tagging > 1 & alive.at.tagging == 1 & Species !=
  "perplexus"), family = "poisson")

gen.model.s4 <- glmer(total ~ log10(colony.size.at.tagging) +
  neonic + cold + (1 + log10(colony.size.at.tagging) | Species) +
  (1 | experimentalBlock) + (1 | beePiPosition), data = subset(demog.data,
  colony.size.at.tagging > 1 & alive.at.tagging == 1 & Species !=
  "perplexus"), family = "poisson")
gen.model.s5 <- glmer(total ~ log10(colony.size.at.tagging) +
  cold + (1 + log10(colony.size.at.tagging) | Species) + (1 |
  experimentalBlock) + (1 | beePiPosition), data = subset(demog.data,
  colony.size.at.tagging > 1 & alive.at.tagging == 1 & Species !=
  "perplexus"), family = "poisson")

anova(gen.model, gen.model.s1, gen.model.s2, gen.model.s3, gen.model.s4,
  gen.model.s5)

```

```

## Data: subset(demog.data, colony.size.at.tagging > 1 & alive.at.tagging == ...  

## Models:  

## gen.model.s5: total ~ log10(colony.size.at.tagging) + cold + (1 + log10(colony.size.at.tagging) | Sp  

## gen.model.s4: total ~ log10(colony.size.at.tagging) + neonic + cold + (1 + log10(colony.size.at.tagg  

## gen.model.s1: total ~ log10(colony.size.at.tagging) * cold + neonic + (1 + log10(colony.size.at.tagg  

## gen.model.s2: total ~ log10(colony.size.at.tagging) + cold * neonic + (1 + log10(colony.size.at.tagg  

## gen.model.s3: total ~ log10(colony.size.at.tagging) * neonic + cold + (1 + log10(colony.size.at.tagg  

## gen.model: total ~ log10(colony.size.at.tagging) * cold * neonic + (1 + log10(colony.size.at.tagging  

##           npar   AIC   BIC logLik deviance   Chisq Df Pr(>Chisq)  

## gen.model.s5    8 557.20 573.26 -270.60    541.20  

## gen.model.s4    9 548.62 566.69 -265.31    530.62 10.5805  1  0.0011429 **  

## gen.model.s1   10 545.46 565.53 -262.73    525.46  5.1653  1  0.0230427 *  

## gen.model.s2   10 549.59 569.66 -264.79    529.59  0.0000  0  

## gen.model.s3   10 550.42 570.49 -265.21    530.42  0.0000  0  

## gen.model     13 535.81 561.90 -254.90    509.81 20.6123  3  0.0001267 ***  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

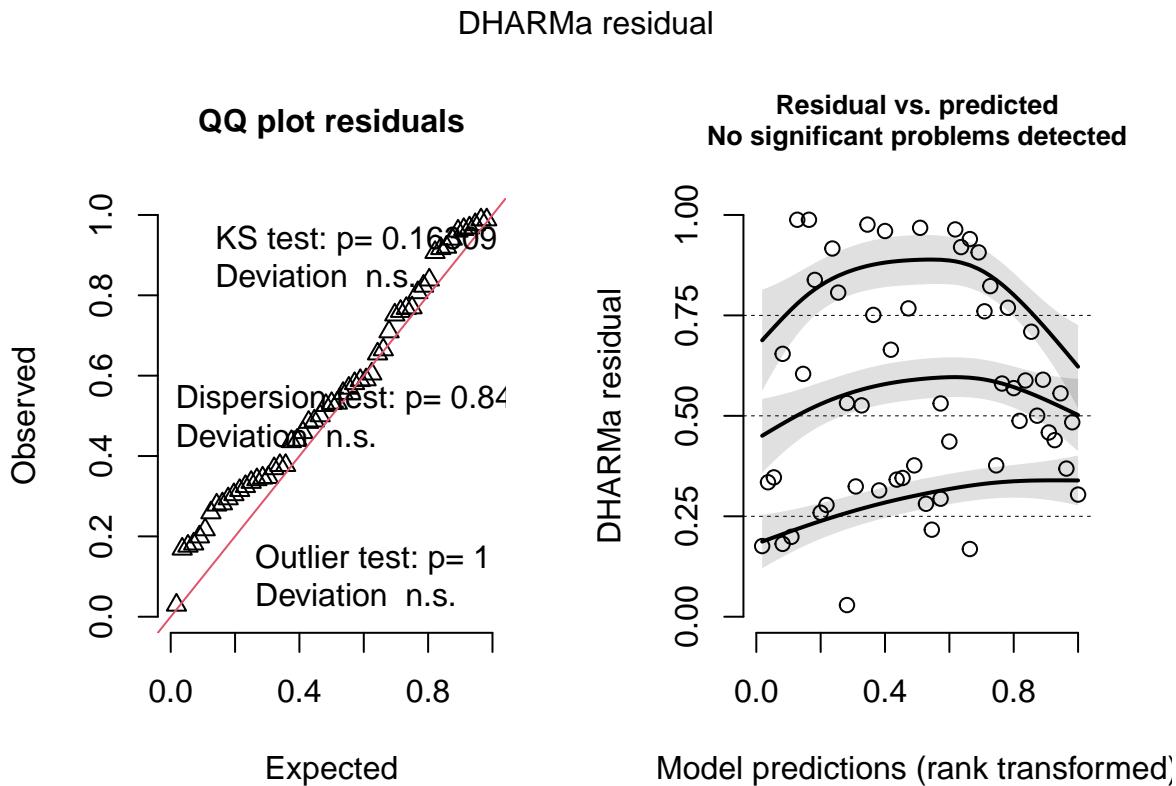
```

capture.output(anova(gen.model, gen.model.s1, gen.model.s2, gen.model.s3,
  gen.model.s4, gen.model.s5), file = "TableS5.comp.txt")
write.csv(as.data.frame(anova(gen.model, gen.model.s1, gen.model.s2,
  gen.model.s3, gen.model.s4, gen.model.s5)), file = "TableS5.comp.csv")

capture.output(summary(gen.model), file = "TableS5.best.txt")
write.csv(signif(summary(gen.model)$coefficients, 3), file = "TableS5.best.csv")
# Preferred model -> complete

```

```
res <- simulateResiduals(gen.model)
plot(res)
```



```
gen.model.exclude.large.colonies <- glmer(total ~ log10(colony.size.at.tagging) *
  neonic * cold + growth.length + (1 | experimentalBlock) +
  (1 | Species), data = subset(demog.data, colony.size.at.tagging >
  0 & colony.size.at.tagging < 40 & alive.at.tagging == 1 &
  Species != "perplexus"), family = "poisson")
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00259194 (tol = 0.002, component 1)
```

```
# NB beePiPosition removed as a random effect in this model
# to improve convergence, removal does not qualitatively
# affect results
```

```
gen.model.exclude.singleton.colonies <- glmer(total ~ log10(colony.size.at.tagging) *
  neonic * cold + growth.length + (1 | experimentalBlock) +
  (1 | Species), data = subset(demog.data, colony.size.at.tagging >
  1 & alive.at.tagging == 1 & Species != "perplexus"), family = "poisson")
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0021387 (tol = 0.002, component 1)
```

```

# NB beePiPosition removed as a random effect in this model
# to improve convergence, removal does not qualitatively
# affect results

gen.model.exclude.large.and.single.colonies <- glmer(total ~
  log10(colony.size.at.tagging) * neonic * cold + growth.length +
  (1 | experimentalBlock) + (1 | Species), data = subset(demog.data,
  colony.size.at.tagging > 1 & colony.size.at.tagging < 40 &
  alive.at.tagging == 1 & Species != "perplexus"), family = "poisson")

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00649863 (tol = 0.002, component 1)

# NB beePiPosition removed as a random effect in this model
# to improve convergence, removal does not qualitatively
# affect results

# Figure 2: growth
pdf("Fig5.pdf", width = 8, height = 4)
par(mfcol = c(1, 2))
out <- plot_model(gen.model.exclude.large.colonies, type = "pred",
  terms = c("cold", "neonic", "colony.size.at.tagging [4]"),
  transform = "exp")
# custom_plot(out, c(0,30), c(0,10,20,30))
custom_plot_log(out, log10(c(1, 40)), log10(c(1, 10)))

## [1] 0.000000 1.60206
## [1] 0 1

out <- plot_model(gen.model.exclude.large.colonies, type = "pred",
  terms = c("cold", "neonic", "colony.size.at.tagging [40]"),
  transform = "exp")
# custom_plot(out, c(0,150), c(0,50,100,150))
custom_plot_log(out, log10(c(1, 150)), log10(c(1, 10, 100)))

## [1] 0.000000 2.176091
## [1] 0 1 2

dev.off()

## pdf
## 2

## [1] 0.14156

# Confirm achieved nest temperature in manipulations above
tmp <- subset(behData, cold == TRUE & rampState == 3) #Create a subset with data in cold-exposed colon

# Nest temperatures
nest.tmps <- aggregate(tmp$nestTemp, by = list(tmp$date), FUN = mean)
colnames(nest.tmps) <- c("timestamp", "nest_temp")
mean(nest.tmps$nest_temp)

```

```

## [1] 14.49346

median(nest.tmps$nest_temp)

## [1] 14.21875

# Chamber temps
ch.tmps <- aggregate(tmp$chamberTemp, by = list(tmp$date), FUN = mean)
colnames(ch.tmps) <- c("timestamp", "nest_temp")
mean(ch.tmps$nest_temp)

## [1] 10.19269

median(ch.tmps$nest_temp)

## [1] 9.9375

setwd("/Users/jamescrall/Dropbox/Work/_Research/Bees/Bumblebees/socialScalingOfStressors/Data/AppletonT"
filelist <- list.files(pattern = ".csv")

par(mfcol = c(4, 1))
for (i in 1:length(filelist)) {

  tempdata <- read.csv(filelist[i], skip = 1)
  tempdata <- tempdata[, 1:6]
  colnames(tempdata) <- c("Num", "Time", "air.temp", "ground.temp",
    "shallow.soil.temp", "deep.soil.temp")
  tempdata$air.temp.diff <- NaN
  tempdata$air.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$air.temp)
  tempdata$ground.temp.diff <- NaN
  tempdata$ground.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$ground.temp)
  tempdata$shallow.soil.temp.diff <- NaN
  tempdata$shallow.soil.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$shallow.soil.temp)
  tempdata$deep.soil.temp.diff <- NaN
  tempdata$deep.soil.temp.diff[2:length(tempdata[, 1])] <- diff(tempdata$deep.soil.temp)
  tempdata$unit <- i
  if (i == 1) {
    m.dat <- tempdata
  } else {
    m.dat <- rbind(m.dat, tempdata)
  }
}

temp.data <- gather(m.dat[, c("Time", "air.temp", "ground.temp",
  "shallow.soil.temp", "deep.soil.temp", "unit")], key = "location",
  value = "Temperature", air.temp, ground.temp, shallow.soil.temp,
  deep.soil.temp)
## Single colony
diff.data <- gather(m.dat[, c("Time", "air.temp.diff", "ground.temp.diff",
  "shallow.soil.temp.diff", "deep.soil.temp.diff", "unit")],

```

```

key = "location", value = "Temperature", air.temp.diff, ground.temp.diff,
shallow.soil.temp.diff, deep.soil.temp.diff)

# Clean out errant readings and nans
temp.data$Temperature[temp.data$Temperature < -10 | temp.data$Temperature >
50] <- NA
diff.data$Temperature[diff.data$Temperature < -5 | diff.data$Temperature >
5] <- NA

temp.data$location <- factor(temp.data$location, levels = c("air.temp",
"ground.temp", "shallow.soil.temp", "deep.soil.temp"))
diff.data$location <- factor(diff.data$location, levels = c("air.temp.diff",
"ground.temp.diff", "shallow.soil.temp.diff", "deep.soil.temp.diff"))
diff.data$Temperature <- abs(diff.data$Temperature)
# hist(temp.data$Temperature) hist(diff.data$Temperature)

setwd(output_folder)
pdf("FigS11.pdf")
par(mfcol = c(2, 1))
p1 <- ggplot(temp.data, aes(x = location, y = Temperature, fill = location)) +
  geom_violin() + geom_boxplot(width = 0.1, outlier.shape = NA) +
  ylim(0, 40) + geom_hline(yintercept = 14.3) + scale_fill_discrete(labels = c("Air",
"Surface", "2 cm belowground", "10 cm belowground"), name = "Temp probe location") +
  theme(legend.position = "none")

p2 <- ggplot(diff.data, aes(x = Temperature, fill = location)) +
  geom_histogram() + xlim(0, 0.5) + scale_fill_discrete(labels = c("Air",
"Surface", "2 cm belowground", "10 cm belowground"), name = "Temp probe location")
grid.arrange(p1, p2)

## Warning: Removed 12637 rows containing non-finite values ('stat_ydensity()').
## Warning: Removed 12637 rows containing non-finite values ('stat_boxplot()').
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## Warning: Removed 100721 rows containing non-finite values ('stat_bin()').
## Warning: Removed 8 rows containing missing values ('geom_bar()').

dev.off()

## pdf
## 2

# Plot for console
grid.arrange(p1, p2)

## Warning: Removed 12637 rows containing non-finite values ('stat_ydensity()').
## Warning: Removed 12637 rows containing non-finite values ('stat_boxplot()').

```

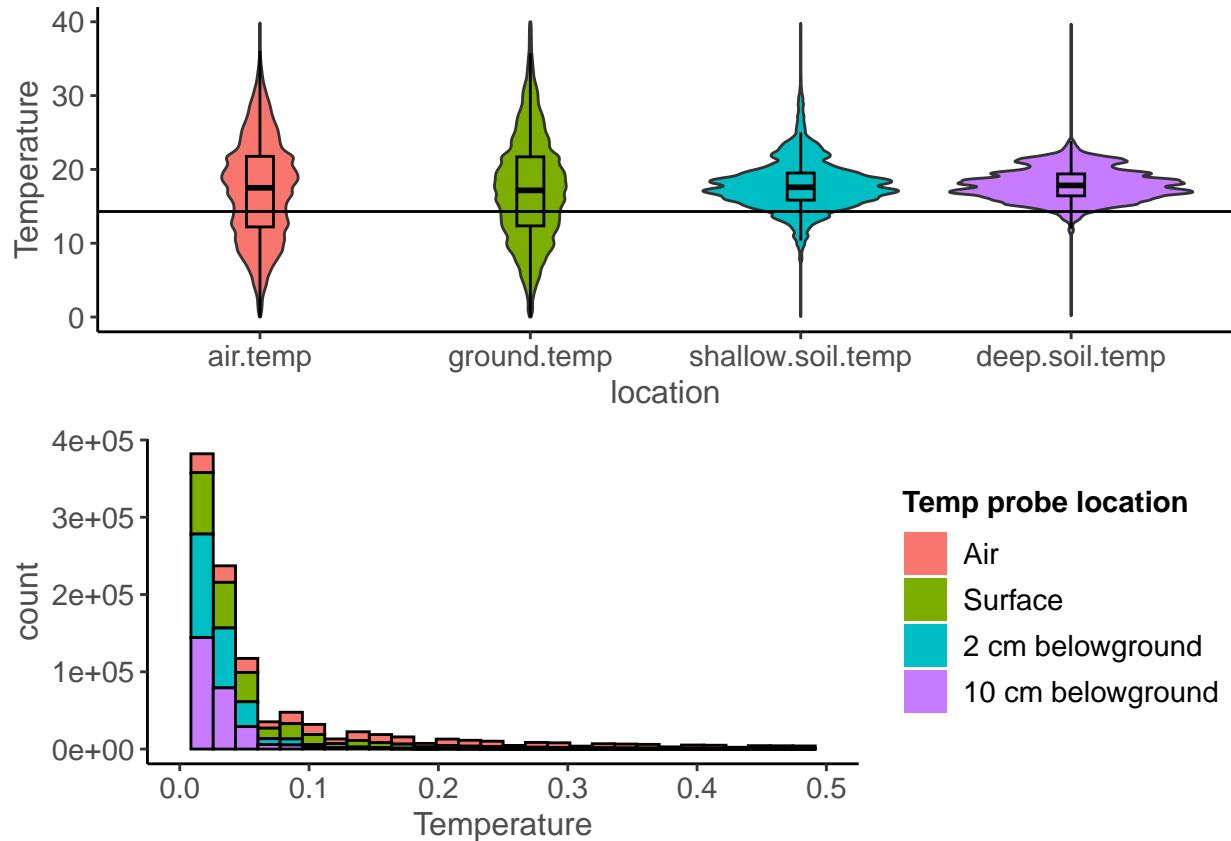
```

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 100721 rows containing non-finite values ('stat_bin()').

## Warning: Removed 8 rows containing missing values ('geom_bar()').

```



```

sm <- diff.data %>%
  group_by(location) %>%
  summarize(quant50 = quantile(Temperature, probs = 0.5, na.rm = TRUE),
            quant95 = quantile(Temperature, probs = 0.95, na.rm = TRUE),
            quant99 = quantile(Temperature, probs = 0.99, na.rm = TRUE))

# Get range of dates
dates <- unique(parse_date_time(m.dat$Time, "mdy HMS"))
min(dates)

```

```
## [1] "2019-08-08 05:54:02 UTC"
```

```
max(dates)
```

```
## [1] "2019-10-02 12:59:35 UTC"
```

```
difftime(min(dates), max(dates))
```

```
## Time difference of -55.29552 days
```