<table>
<thead>
<tr>
<th>Habitat Variable</th>
<th>PC 1</th>
<th>PC 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenvalue</td>
<td>1.54</td>
<td>1.35</td>
</tr>
<tr>
<td>Percent of variance explained</td>
<td>33.75</td>
<td>26.16</td>
</tr>
<tr>
<td>Shrub cover</td>
<td>-0.435</td>
<td>0.125</td>
</tr>
<tr>
<td>Number of saplings</td>
<td>-0.0582</td>
<td>0.439</td>
</tr>
<tr>
<td>Number of beech trees</td>
<td>0.601</td>
<td>0.0103</td>
</tr>
<tr>
<td>Total number of trees</td>
<td>0.579</td>
<td>-0.0745</td>
</tr>
<tr>
<td>Total basal area</td>
<td>0.143</td>
<td>0.341</td>
</tr>
<tr>
<td>Mean tree basal area</td>
<td>-0.244</td>
<td>-0.582</td>
</tr>
<tr>
<td>Percent canopy cover</td>
<td>0.174</td>
<td>-0.574</td>
</tr>
</tbody>
</table>

Table S1: Results of principle component analysis of seven habitat variables measured at experimental playback locations. The first two components, shown here, explained over half of the variance and were included as covariates in the settlement model. Values for habitat variables indicate factor loadings for each component.
Appendix S2  Additional information about stable isotope analysis and dispersal status determination

Stable Isotope Analysis

Feathers and claws were cleaned in 2:1 chloroform:methanol solution and air dried in a fume hood for 48 hours. Samples were transported to the Smithsonian Stable Isotope Mass Spectrometry Laboratory in Suitland, Maryland where the feathers were allowed to equilibrate with the local atmosphere for > 72 hours. After equilibration, a 0.3-0.4mg sample was clipped from the distal end of each feather and loaded into a silver capsule. For claw samples, 0.3-0.4mg of each sample was loaded into a tin capsule. Samples were then crushed, pyrolized at 1350°C in an elemental analyzer (Thermo TC/EA), and introduced in a continuous-flow isotope ratio mass spectrometer (Thermo Delta V Advantage) via a Conflo IV interface. Calculations of raw isotope values were performed with Isodat 3.0 software. All runs included a set of standards for every 10-12 samples. The stable hydrogen (δ²H) values reported include only non-exchangeable hydrogen, as determined by a 3-point linear correction using keratin standards [1]. Three keratin standards were used (expected value): CBS (-197‰), KHS (-121.6‰), & Spectrum Keratin (-54.1‰). All values are expressed in the typical delta notation in units of per mil (‰) normalized on the Vienna Standard Mean Ocean Water–Standard Light Antarctic Precipitation (VSMOW-SLAP) scale for hydrogen and the Vienna PeeDee Belemnite scale for carbon. Analytical error was better than 2‰ for hydrogen samples and 0.2‰ for carbon samples based on replicate analyses of standards.

Determining dispersal status

We probabilistically determined the origin of all unknown-origin individuals in our population using year-specific distributions of local δ²Hf values [2]. We first estimated the expected local δ²Hf value for each year using the mean δ²Hf values from either breeding adult males (2009) or from individuals known to have bred at the study site the previous year (2010-2012). Next,
we centered the $\delta^2 H_f$ values from all recaptured individuals from 2010-2012 on the year-specific means and, because the mean-centered values were normally distributed (Shapiro-Wilk test: $W = 0.972$, $P = 0.5123$), we used the standard deviation of these values (7.179‰) as a measure of local variation in $\delta^2 H_f$ values. Finally, we used the year-specific means and pooled standard deviation to probabilistically assign all unbanded individuals into one of three dispersal categories based on a predefined odds ratio for correctly classifying individuals as local. This odds ratio was used to calculate the range of $\delta^2 H_f$ values capturing a given area under each year-specific local distribution (e.g. 80%) and individuals with $\delta^2 H_f$ values within this range were classified as local while individuals with $\delta^2 H_f$ values more negative or positive than the threshold range were classified as “northern” or “southern,” respectively [2]. To test the sensitivity of our results to the threshold used to classify dispersal status, we carried out the classifications using three progressively stringent thresholds (4:1 odds, 9:1 odds and 19:1 odds) and performed all analyses under each scenario.

Previous studies using hydrogen isotopes to determine the origin of songbirds [3, 4] have applied a correction factor to hydrogen values from yearling individuals to account for possible age-specific isotope discrimination. Over the course of our study, we recaptured six yearlings that were originally banded at our study site as nestlings and hydrogen values from these individuals did not differ from the values of recaptured adults individuals (yearling mean = -64.76‰, adult mean = -64.72‰, $t = 0.016$, $df = 5$, $P = 0.988$). Therefore, we chose not to apply any age-correction to the hydrogen values from unbanded yearlings.

**Literature Cited**


Appendix S3  Posterior predictive checks for the dispersal model

Posterior predictive checks

Figure S1: Results of posterior predictive checks for the immigration model. The histogram shows the simulated number of local individuals based on draws from the posterior distribution for $p_{PB}$ for the 4:1 odds ratio model. The red line shows the observed number of local individuals at experimental conspecific attraction points.
R script associated the manuscript entitled *Habitat features and long-distance dispersal modify the use of social information by a long-distance migratory bird*

Clark S. Rushing  
April 29, 2015

This R Markdown document provides all R code used to analyze playback experiment and dispersal data and to generate the figures included in the manuscript.

Install the crushingr package, which will automatically load all packages required for the script

```r
devtools::install_github('crushing05/crushingr')
require(crushingr)
```

Analysis of settlement data from playback experiment

(note: data files must be saved in current working directory for the script to run)

Read playback data and add occupancy columns for yearlings and adults

```r
pb <- read.csv("pb_complete.csv")
pb$year <- pb$year-2012
pb$trt <- factor(pb$trt, levels(pb$trt)[c(2,3,1)])
pb$occ <- ifelse(pb$abun > 0, 1, 0)  # Convert abundance to occupancy
pb$abun.asy <- pb$abun - pb$abun.sy  # ASY abundance
pb$occ.sy <- ifelse(pb$abun > 0 & pb$abun.sy > 0, 1, 0)  # SY occupancy
pb$occ.asy <- ifelse(pb$abun.asy > 0, 1, 0)  # ASY occupancy
pb$trt1 <- as.numeric(factor(pb$trt, levels=c("control", "ca", "pi")))
```

Yearling settlement models

Fit full model with all predictors and interactions

```r
full.sy <- glm(occ.sy ~ trt + PC1*trt + PC2*trt + year*trt, data = pb, family = "binomial")
summary(full.sy)
```

Test for interactions; if not significant, remove to test for main effects

```r
fit1.sy <- glm(occ.sy ~ trt + PC1*trt + PC2 + year*trt, data = pb, family = "binomial")  # Test for PC2
anova(fit1.sy, full.sy, test="Chisq")  # NS

fit2.sy <- glm(occ.sy ~ trt + PC1 + PC2*trt + year*trt, data = pb, family = "binomial")  # Test for PC1
anova(fit2.sy, full.sy, test="Chisq")  # NS

fit3.sy <- glm(occ.sy ~ trt + PC1*trt + PC2*trt + year, data = pb, family = "binomial")  # Test for yr
anova(fit3.sy, full.sy, test="Chisq")  # NS
```
No evidence of interactions, so new ‘full’ model

```r
full2.sy <- glm(occ.sy ~ trt + PC1 + PC2 + year, data = pb, family = "binomial") # No interactions
anova(full2.sy, full.sy, test="Chisq") # NS
summary(full2.sy)
```

Test for main effects of predictors by dropping and comparing to full model using likelihood ratio tests

```r
fit4.sy <- glm(occ.sy ~ trt + PC1 + PC2, data = pb, family = "binomial") # Test for year effect
anova(fit4.sy, full2.sy, test = "Chisq") # NS

fit5.sy <- glm(occ.sy ~ trt + PC2 + year, data = pb, family = "binomial") # Test for PC1 effect
anova(fit5.sy, full2.sy, test = "Chisq") # NS

fit6.sy <- glm(occ.sy ~ trt + PC1 + year, data = pb, family = "binomial") # Test for PC2 effect
anova(fit6.sy, full2.sy, test = "Chisq") # p = 0.015

fit7.sy <- glm(occ.sy ~ PC1 + PC2 + year, data = pb, family = "binomial") # Test for treatment effect
anova(fit7.sy, full2.sy, test = "Chisq") # p = 0.0033
```

Final ‘top’ model used for parameter estimates

```r
fit8.sy <- glm(occ.sy ~ trt + PC2, data = pb, family = "binomial") # Top model
summary(fit8.sy)
```

**Adult settlement models**

Fit full model with all predictors and interactions

```r
full.asy <- glm(occ.asy ~ trt + PC1*trt+PC2*trt + year*trt, data = pb, family = "binomial")
```

Test for interactions

```r
fit2.asy <- glm(occ.asy ~ trt + PC1*trt + PC2 + year*trt, data = pb, family = "binomial") # Test for for interactions
anova(fit2.asy, full.asy, test="Chisq") # NS

fit3.asy <- glm(occ.asy ~ trt + PC1 + PC2*trt + year*trt, data = pb, family = "binomial") # Test for PC1 interaction
anova(fit3.asy, full.asy, test="Chisq") # NS

fit4.asy <- glm(occ.asy ~ trt + PC1*trt + PC2*trt + year, data = pb, family = "binomial") # Test for PC2 interaction
anova(fit4.asy, full.asy, test="Chisq") # NS
```

Remove non-significant interactions, new ‘full’ model

```r
full2.asy <- glm(occ.asy ~ trt + PC1 + PC2 + year, data = pb, family = "binomial") # NS
anova(full2.asy, full.asy, test="Chisq") # NS
summary(full2.asy)
```

Test for main effects of predictors by dropping and comparing to full model using likelihood ratio tests
Final 'top' model used for parameter estimates

```r
fit9.asy <- glm(occ.asy ~ trt + PC1*trt, data = pb, family = "binomial") # Top model
summary(fit9.asy)
```

Outlier tests

Several points for both age classes appear to be outliers. Remove these points and rerun models to test whether results are driven by these points

Adult Outlier test occupancy

```r
pb2 <- pb[pb$PC1<4,]

# Full model
full.asy <- glm(occ.asy ~ trt + PC1*trt+PC2*trt + year*trt, data = pb2, family = "binomial")

## LRT for interactions
fit2.asy <- glm(occ.asy ~ trt + PC1*trt + PC2 + year*trt, data = pb2, family = "binomial") # Test for PC2*trt
anova(fit2.asy, full.asy, test="Chisq") # NS

fit3.asy <- glm(occ.asy ~ trt + PC1 + PC2*trt + year*trt, data = pb2, family = "binomial") # Test for PC1*trt
anova(fit3.asy, full.asy, test="Chisq") # NS

fit4.asy <- glm(occ.asy ~ trt + PC1*trt + PC2*trt + year, data = pb2, family = "binomial") # Test for yr*trt
anova(fit4.asy, full.asy, test="Chisq") # NS

## Remove non-significant interactions, new 'full' model
full2.asy <- glm(occ.asy ~ trt + PC1 + PC2 + year, data = pb2, family = "binomial") # NS
summary(full2.asy)

fit5.asy <- glm(occ.asy ~ trt + PC1 + PC2, data = pb2, family = "binomial") # Test for year effect
anova(fit5.asy, full2.asy, test= "Chisq") # NS

fit6.asy <- glm(occ.asy ~ trt + PC2 + year, data = pb2, family = "binomial") # Test for PC1 effect
anova(fit6.asy, full2.asy, test= "Chisq") # 0.014
```
Yearling Outlier test occupancy

```r
pb3 <- pb[pb$PC2 < 4,]
# Full model
full.sy <- glm(occ.sy ~ trt + PC1*trt + PC2*trt + year*trt, data = pb3, family = "binomial")

## LRT for interactions
fit2.sy <- glm(occ.sy ~ trt + PC1*trt + PC2 + year*trt, data = pb3, family = "binomial") # Test for PC2*trt
anova(fit2.sy, full.sy, test="Chisq") # NS

fit3.sy <- glm(occ.sy ~ trt + PC1 + PC2*trt + year*trt, data = pb3, family = "binomial") # Test for PC1*trt
anova(fit3.sy, full.sy, test="Chisq") # NS

fit4.sy <- glm(occ.sy ~ trt + PC1*trt + PC2*trt + year, data = pb3, family = "binomial") # Test for yr*trt
anova(fit4.sy, full.sy, test="Chisq") # NS

## Remove non-significant interactions, new 'full' model
full2.sy <- glm(occ.sy ~ trt + PC1 + PC2 + year, data = pb3, family = "binomial")
anova(full2.sy, full.sy, test="Chisq") # NS

summary(full2.sy)

fit5.sy <- glm(occ.sy ~ trt + PC1 + PC2, data = pb3, family = "binomial") # Test for year effect
anova(fit5.sy, full2.sy, test = "Chisq") # NS

fit6.sy <- glm(occ.sy ~ trt + PC2 + year, data = pb3, family = "binomial") # Test for PC1 effect
anova(fit6.sy, full2.sy, test = "Chisq") # NS

fit7.sy <- glm(occ.sy ~ trt + PC1 + year, data = pb3, family = "binomial") # Test for PC2 effect
anova(fit7.sy, full2.sy, test = "Chisq") # p = 0.064

fit8.sy <- glm(occ.sy ~ PC1 + PC2 + year, data = pb3, family = "binomial") # Test for trt effect
anova(fit8.sy, full2.sy, test = "Chisq") # p = 0.002

fit9.sy <- glm(occ.sy ~ trt + PC2, data = pb3, family = "binomial") # Top model
summary(fit9.sy)
```

Analysis of dispersal data

Create bayesian binomial model

```r
sink("Disp_binom.txt")
cat(""
model { # Binomial GLM
  # Priors
  for (i in 1:2){
    p.imm[i] ~ dunif(0,1) # p.imm[1] = population immigration rate
    p.imm[i] ~ dunif(0,1) # p.imm[2] = playback immigration rate
  }
  # Likelihood
  for (i in 1:2) {
    C[i] ~ dbin(p.imm[i], n[i])
  }
  # Derived quantities
}
", fill = TRUE)
sink()
```

Read data and set MCMC settings

```r
disp.df <- read.csv("disp_df.csv")

# JAGS settings
nc <- 3
ni <- 25000
nb <- 10000
nt <- 2

inits <- function(){list(p.imm=runif(2, .3,.7))}
parameters<- c("p.imm", "p.diff")
```

Analysis under 4:1 odds ratio

```r
jags.disp4 <- list(C=disp.df$imm4, n=disp.df$N)

jm.disp4 <- jags.model("Disp_binom.txt", data = jags.disp4, inits = inits,
                       n.chains = nc, n.adapt = nb)
zm.disp4 <- coda.samples(jm.disp4, variable.names = parameters, n.iter = ni,
                         n.thin=nt)
summary(zm.disp4)
#traceplot(zm.disp4) # Check traceplots for convergence
#gelman.diag(zm.disp4, transform = TRUE) # Check Gelman-Rubin diagnostics for convergence
```
Use posterior predictive checks to assess model fit

```
n.sims <- 1500
y.rep.pop <- array(NA, c(n.sims))
for (sim in 1:n.sims){
    y.rep.pop[sim] <- rbinom(n=1, size = disp.df[1,1], prob = df.tot4[sim,2])
    par(mfrow=c(1,1))
    hist(y.rep.pop, xlim=c(60,100),xlab="Number of Local Individuals", ylab = "Count", main="")
    abline(v=disp.df[1,2], col="red", lwd=2)
}

y.rep.pb <- array(NA, c(n.sims))
for (sim in 1:n.sims){
    y.rep.pb[sim] <- rbinom(n=1, size = disp.df[2,1], prob = (df.tot4[sim,3]))
    hist(y.rep.pb, xlim=c(0,16),xlab="Number of Local Individuals", ylab = "Count", main="")
    abline(v=(disp.df[2,2]), col="red", lwd=2)
```

Analysis under 9:1 odds ratio

```
jags.disp9 <- list(C=disp.df$imm9, n=disp.df$N)
jm.disp9 <- jags.model("Disp_binom.txt", data = jags.disp9, inits = inits, n.chains = nc, n.adapt = nb)
zm.disp9 <- coda.samples(jm.disp9, variable.names = parameters, n.iter = ni, n.thin=nt)
summary(zm.disp9)
#traceplot(zm.disp9)
#gelman.diag(zm.disp9, transform = TRUE)
```

9:1 Posterior Predictive Checks

```
n.sims <- 1500
y.rep.pop <- array(NA, c(n.sims))
for (sim in 1:n.sims){
    y.rep.pop[sim] <- rbinom(n=1, size = disp.df[1,1], prob = df.tot9[sim,2])
    par(mfrow=c(1,1))
    hist(y.rep.pop, xlim=c(60,100),xlab="Number of Local Individuals", ylab = "Count", main="")
    abline(v=disp.df[1,3], col="red", lwd=2)
}

y.rep.pb <- array(NA, c(n.sims))
for (sim in 1:n.sims){
    y.rep.pb[sim] <- rbinom(n=1, size = disp.df[2,1], prob = (df.tot9[sim,3]))
    hist(y.rep.pb, xlim=c(0,16),xlab="Number of Local Individuals", ylab = "Count", main="")
    abline(v=(disp.df[2,2]), col="red", lwd=2)
```
Analysis under 19:1 odds ratio

```r
jags.disp19 <- list(C=disp.df$imm19, n=disp.df$N)

jm.disp19 <- jags.model("Disp_binom.txt", data = jags.disp19, inits = inits,
        n.chains = nc, n.adapt = nb)
zm.disp19 <- coda.samples(jm.disp19, variable.names = parameters, n.iter = ni,
        n.thin=nt)
summary(zm.disp19)
```

#traceplot(zm.disp.tot19)
#gelman.diag(zm.disp.tot19, transform = TRUE)

df.tot19 <- as.data.frame(zm.disp19[[1]], zm.disp19[[2]], zm.disp19[[3]]) # Save and write MCMC for figure
write.csv(df.tot19, "disp19_mcmc.csv", row.names = FALSE)

19:1 Posterior Predictive Check

```r
n.sims <- 1500
y.rep.pop <- array(NA, c(n.sims))
for (sim in 1:n.sims){
y.rep.pop[sim] <- rbinom(n=1, size = disp.df[1,1], prob = df.tot19[sim,2])
}
par(mfrow=c(1,1))
hist(y.rep.pop, xlim=c(60,100),xlab="Number of Local Individuals", ylab = "Count", main="")
abline(v=disp.df[1,4], col="red", lwd=2)

y.rep.pb <- array(NA, c(n.sims))
for (sim in 1:n.sims){
y.rep.pb[sim] <- rbinom(n=1, size = disp.df[2,1], prob = (df.tot19[sim,3]))
}
hist(y.rep.pb, xlim=c(0,16),xlab="Number of Local Individuals", ylab = "Count", main="")
abline(v=(disp.df[2,4]), col="red", lwd=2)
```

Figures

If necessary, load crushingr package

```r
#devtools::install_github('crushing05/crushingr)
require(crushingr)
```

Settlement Probability Plot

First, fit ‘means’ parameterization of top models

```r
fit.sy <- glm(occ.sy ~ trt + PC2 ~ 1, data = pb, family = "binomial") # Top yearling model
fit.asy <- glm(occ.asy ~ trt + PC1 ~ 1, data = pb, family = "binomial") # Top adult model
```

Next, estimate settlement probabilities, CI’s and store in data.frame
sy.pred <- invlogit(cbind(coef(fit.sy)[1:3], confint(fit.sy)[1:3,]))
asy.pred <- invlogit(cbind(coef(fit.asy)[1:3], confint(fit.asy)[1:3,]))
settlement.pred <- data.frame(rbind(sy.pred, asy.pred))
names(settlement.pred) <- c("Estimate", "LCI", "UCI")
settlement.pred$Treatment <- rep(c("Control", "Public information", "Location cues"), 2)
settlement.pred$Age <- rep(c("Yearling", "Adult"), each = 3)

Trick to plot tick marks on inside plot in ggplot2

```
rug_breaksx <- data.frame(x=c(0, 1, 2, 3), # Hack to get tick marks on the inside of figure
y=c(0,0.25, 0.5, 0.75, 1.00))
```

Create figure

```
ggplot(data = settlement.pred, aes(x=Treatment, y=Estimate)) +
  geom_errorbar(aes(ymax=UCI, ymin=LCI, group=Age, linetype=Age),
               position=position_dodge(width=.5), width=0) +
  geom_point(aes(group=Age, fill=Age), size=6, position=position_dodge(width=.5), shape=21) +
  scale_fill_manual(values = c("black", "white")) +
  scale_line_type_manual(values=c("solid","dashed"))+
  scale_y_continuous("Probability of settlement", lim=c(0,1))+
  scale_x_discrete("Treatment", labels=c("Control \n", "Location \ncues", "Public \ninformation"))+
  geom_rug(data=rug_breaksx, aes(x=x, y=y))+
  theme(panel.grid.major = element_line(color="grey95"),
        panel.grid.major.x = element_blank(),
        axis.ticks = element_blank(),
        legend.text = element_text(size=28),
        legend.title = element_blank(),
        legend.key.size = unit(1.75, "cm"),
        legend.position=c(.85,.84),
        plot.margin=unit(c(1,1,1,1), "cm"),
        legend.key = element_blank())
```

Habitat effect plots

Yearling plot

First, estimate treatment-specific slopes to overlay on figure

```
hab.sy <- glm(occ.sy ~ trt*PC2, data = pb, family = "binomial")
print(coef(hab.sy), digits=1)
lc.lab.sy <- paste("beta == ", -1.22) # Location cue slope
pi.lab.sy <- paste("beta == ", -0.73) # Public information slope
co.lab.sy <- paste("beta == ", -0.11) # Control slope
rug.sy <- data.frame(x = c(-2, 0, 2, 4, 6), y = c(0, 0.25, 0.5, 0.75, 1))
```

Plot
Adult plot

First, estimate treatment-specific slopes to overlay on figure

```r
hab.asy <- glm(occ.asy ~ trt*PC1, data = pb, family = "binomial")
print(coef(hab.asy), digits=2)
lc.lab.asy <- paste("beta == ", -0.42) # Location cue slope
c0.lab.asy <- paste("beta == ", -0.27) # Control slope (*Public info slope = 0, so not shown*)
rug.asy <- data.frame(x = c(-2, 0, 2, 4, 6), y = c(1, 0.25, 0.5, 0.75, 1))
```

Plot

```r
ggplot(pb, aes(x=PC1, y=occ.asy))+ geom_point(aes(shape=trt))+
  stat_smooth(aes(linetype=trt), method="glm", family="binomial", color="grey75",
              fullrange=TRUE, se=FALSE, size=0.6) +
  stat_smooth(aes(linetype=trt), method="glm", family="binomial", se=FALSE, fullrange=TRUE, color = "black", size=0.8)+
  scale_x_continuous("Habitat PC2")+
  scale_y_continuous("Yearling settlement probability")+
  scale_linetype_manual(values = c("F1", "longdashed", "dashed"),
                         labels = c("Control", "Public information", "Location cues"))+
  scale_shape_manual(values = c(1,19,4),
                     labels = c("Control", "Public information", "Location cues"))+
  guides(shape=guide_legend())+
  annotate("text", x = -0.2, y = 0.75, label = lc.lab.asy, color="grey40", parse=TRUE)+
  annotate("text", x = -2, y = 0.12, label = co.lab.asy, color="grey40", parse=TRUE)+
  theme(panel.grid.major = element_line(color="grey95"), panel.grid.major.x = element_blank(),
        strip.text.x=element_text(size = 18), strip.text.y=element_text(size = 18), legend.position=c(1, 0.8),
        legend.text = element_text(size = 18), legend.key = element_blank(), legend.key.size = unit(1, "cm"),
        axis.ticks = element_blank())+
  geom_rug(data=rug.asy, aes(x=x, y=y))
```
Plot of posterior distributions from dispersal model

Read in MCMC files

```r
mcmc4 <- read.csv("disp4_mcmc.csv")  # Read MCMC output from dispersal model
mcmc9 <- read.csv("disp9_mcmc.csv")
mc19 <- read.csv("disp19_mcmc.csv")
```

Create legend data.frame and hack for tick marks

```r
disp.legend <- data.frame(x=c(0,0,0), y=c(10,10,10), Odds = factor(c("4:1", "9:1", "19:1"), # Hack to add legend
levels=c("4:1", "9:1", "19:1")))
rug.disp <- data.frame(x=c(-.25, 0, .25, .5, .75), y=c(0,2,4,6,8))  # Hack to add tick marks
```

Plot

```r
ggplot(mcmc4, aes(x=p.diff)) + geom_density(adjust=5, size = 1) +
  geom_segment(aes(x= 0.00694, y= 0, xend = 0.00694, yend= 0.9), linetype="dotted", size=0.6) +
  geom_segment(aes(x= 0.462, y= 0, xend = 0.462, yend= 0.7), linetype="dotted", size=0.6) +
  geom_density(data= mcmc9, aes(x=p.diff), adjust=5, size = 1, linetype="longdash") +
  geom_density(data= mc19, aes(x=p.diff), adjust=5, size = 1, linetype="dashed") +
  scale_y_continuous("Density", lim=c(0,8)) + scale_x_continuous(lim=c(-.3,.75)) +
  labs(x=bquote(italic(p[diff]))) +
  geom_errorbar(data=disp.legend, aes(x=-0.3, ymin=y, ymax=y, linetype=Odds), size=1.3) +
  scale_linetype_manual(values=c("solid", "longdash", "dashed")) +
  geom_rug(data=rug.disp, aes(x=x, y=y)) +
  theme(axis.ticks = element_blank(),
        legend.text = element_text(size=28),
        legend.title = element_blank(),
        legend.key.size = unit(2, "cm"),
        legend.position = c(.65,.75),
        plot.margin = unit(c(1,1,1,1), "cm"),
        legend.key = element_blank())
```

10