Published in "Methods in Ecology and Evolution 5(11): 1255–1263, 2014"
which should be cited to refer to this work.

Appendix 1 Data and R code to produce Figure 2.

```
  library(TeachingDemos)

# function to compute small sample version of AIC from output of optim
  aic.opt <- function(x, n) {
    k <- length(x$par)
    aic <- 2*x$value + 2*k
    return(aic + (2*k*(k+1))/(n-k-1))
  }

# Data from Memmot et al (2005)
# establishment success (1) or failure (0) of psyllid introduction after 5 years
  suc <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
           0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1)
# Actual number of individuals released (data supplied by Jane Memmot)
  rel.size <- c(1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 7,
                7, 8, 8, 8, 9, 10, 10, 10, 10, 18, 23, 23, 24, 27, 27, 27, 28,
                30, 30, 69, 70, 76, 79, 80, 82, 82, 82, 82, 90, 152, 176, 185,
                255, 258)
# Number of individuals planned to be released
  plan.rel <- c(rep(2, 10), rep(4, 10), rep(10, 10), rep(30, 10), rep(90, 10), rep(270, 5))


# likelihood function for demographic stochasticity alone (equation 4)
binom.lik <- function(p, y, x) {
  mu <- 1 - (1  - p)^x
  logl <- sum(dbinom(y, 1, mu, log=T))
  return(-logl)
}

# likelihood function for demographic stochasticity + Allee effect from Dennis 1989 (equation 6)
allee.binom.lik <- function(param, y, x) {
  r <- param[2] + 1
  p <- exp(param[1]) / (1 + exp(param[1]))  # logit transformation to ensure 0 < p < 1
  mu <- pnbinom(x, size=r, prob=p)
  logl <- sum(dbinom(y, 1, mu, log=T))
  return(-logl)
}

# likelihood function for demographic stochasticity + among-population heterogeneity (equation 10)
het.binom.lik <- function(param, y, x) {
  disp <- exp(param[2])                     # log transformation to ensure dispersion parameter > 0
  p <- exp(param[1]) / (1 + exp(param[1]))  # logit transformation to ensure 0 < p < 1
  a <- p*disp
  b <- (1-p)*disp
  mu <- 1 - (beta(a, (x+b)) / beta(a, b))
  logl <- sum(dbinom(y, 1, mu, log=T))
  return(-logl)
}

# fit the equations to the data with maximum likelihood
  e4 <- optim(par=c(0.05), fn=binom.lik, method="BFGS", y=suc, x=rel.size, hessian=T)
  e6 <- optim(par=c(-1,1.1), fn=allee.binom.lik, method="BFGS", y=suc, x=rel.size, hessian=T)
  e10 <- optim(par=c(-2, 2), fn=het.binom.lik, method="BFGS", y=suc, x=rel.size, hessian=T)

# fit equation 7 using GLM (=equation 8)
  e7 <- glm(suc ~ log(rel.size), family=binomial(link="cloglog"))


# cacluate the proportion of successes and the mean actual release size for each of the planned
release sizes
  m <- tapply(suc, plan.rel, mean)
```

```
  n <- tapply(rel.size, plan.rel, mean)


# draw the figure
# plot the raw data
  xx <- 1:270
  y <- ifelse(suc==1, 1.05, -0.05)
  plot(jitter(y, 0.15) ~ rel.size, bty="l", pch=3, cex=0.5, col="grey",
    ylab="Establishment probability", xlab="Founding population size", cex.lab=1.4)
  abline(h=0, col="grey")
  abline(h=1, col="grey")
# plot the proportion of successes for each planned release size
  points(m ~ n, pch=19, cex=1.5)

# plot the fitted relationship for equation 4
  p.est.fit <- 1 - (1 - e4$par)^xx
  lines(p.est.fit ~ xx, lwd=2, lty=2)

# plot the fitted relationship for equation 6 and calculate theta
  r <- e6$par[2] + 1
  p <- exp(e6$par[1])/(1+exp(e6$par[1]))
  p.est.allee <- pnbinom(xx, size=r, prob=p)
  lines(p.est.allee ~ xx, lwd=2, lty=3)

# plot the fitted relationship for equation 10
  p <- exp(e10$par[1]) / (1 + exp(e10$par[1]))
  disp <- exp(e10$par[2])
  a <- p*disp
  b <- (1-p)*disp
  p.est.het <- 1 - beta(a, (xx+b)) / beta(a, b)
  lines(p.est.het ~ xx, lwd=2, lty=1)

# plot the fitted relationship for equation 7
  pe7 <- coef(e7)[1] + coef(e7)[2]*log(xx)
  p.est.e7 <- 1 - exp(-exp(pe7))
  lines(p.est.e7 ~ xx, lwd=2, lty=4)

# show the distribution of p values given the estimated a and b parameters of the beta distribution
from equation 8
  h <- rbeta(1000000, a, b)
  return(subplot(hist(h, freq=FALSE, breaks=seq(0, max(h)+0.005, 0.005), col="black",
    main="", xlab=expression(italic(p)), ylab="Density"), x=c(175, 275), y=c(0.25, 0.6),
    pars=list(cex.axis=0.8, cex.lab=0.8, tcl=-0.2, mgp=c(1.5,0.5,0))))

# mean p value and variance
  a/(a+b)
  a*b/(((a+b)^2)*(a+b+1))

# compare AIC values
  n <- length(suc)
  aic.opt(e4, n)
  aic.opt(e6, n)
  aic.opt(e10, n)
  k <- length(coef(e7))
  AIC(e7) + (2*k*(k+1))/(n-k-1)

# function to calculate 95% confidence intervals from optim fit
  ci <- function(x) {
    FI <- solve(x$hessian)
    se <- sqrt(diag(FI))
    ucl <- x$par + 1.96*se
    lcl <- x$par - 1.96*se
    return(cbind(x$par, lcl, ucl))
  }

# 95% confidence intervals
  ci(e6)
  confint(e7)
```

```
# compare the fit of a logit link
  e7.l <- glm(suc ~ log(rel.size), family=binomial(link="logit"))

  AIC(e7) + (2*k*(k+1))/(n-k-1)
  AIC(e7.l) + (2*k*(k+1))/(n-k-1)
```

Appendix 2 R code to produce Figure 3.

```
  library(arm)
  library(TeachingDemos)

# function to back transfrom from logit
  unlogit <- function(x) exp(x) / (1 + exp(x))

# function to compute AIC from output of optim
  aic.opt <- function(x, n) {
    k <- length(x$par)
    aic <- 2*x$value + 2*k
    return(aic)
  }

# function to cut a vector into groups each with n observations
  cut2 <- function(y, x, n) {
    ord <- order(x)
    x <- x[ord]
    y <- y[ord]
    r <- length(x)
    b <- 1:r
    n.group <- round(r/n, 0)
    g <- cut(b, breaks=n.group)
    out.y <- tapply(y, g, mean)
    out.x <- tapply(x, g, mean)
    return(cbind(out.y, out.x))
  }

  setwd("c:\\data\\global propagule pressure")

# read in data from Sol et al
  bird <- read.csv("c:\\data\\global propagule pressure\\Sol et al data.csv")
# exclude observations without estimates of founding population size
  bird <- subset(bird, is.na(Propagule_size)==FALSE)
  bird$Species <- factor(bird$Species)

  table(table(bird$Species))

# option to subset to include species with > n records
  n <- 0
  a <- table(bird$Species)
  bird$n.rec <- a[match(bird$Species, names(a))]
  bird <- subset(bird, n.rec > n)
  bird$Species <- factor(bird$Species)

# extract variables for analysis
  x <- bird$Propagule_size
  y <- bird$Outcome_binary
  N <- length(x)
  species <- as.numeric(factor(bird$Species))
  N.species <- max(species)

# set a limit to truncate the plot
  x.lim <- 300
  xx <- 0:x.lim

# plot raw data
  yy <- ifelse(y==1, 1.05, -0.05)
  plot(jitter(yy, 0.15) ~ x, bty="l", pch=3, cex=0.5, col="grey",
    xlim=c(0,   x.lim),   ylab="Establishment   probability",   xlab="Founding   population   size",
cex.lab=1.4)
  abline(h=0, col="grey")
  abline(h=1, col="grey")
```

```
  out.av <- cut2(y=y, x=x, n=20)
  points(out.av[, 1] ~ out.av[, 2], pch=19, cex=1.5)

# fit by maximum likelihood
# likelihood function for demographic stochasticity alone (equation 4)
binom.lik <- function(param, y, x) {
  p <- exp(param[1]) / (1 + exp(param[1]))  # logit transformation to ensure 0 < p < 1
  mu <- 1 - (1  - p)^x
  logl <- sum(dbinom(y, 1, mu, log=T))
  return(-logl)
}

# likelihood function for demographic stochasticity + Allee effect from Dennis 1989 (equation 6)
allee.binom.lik <- function(param, y, x) {
  r <- param[2] + 1
  p <- exp(param[1]) / (1 + exp(param[1]))  # logit transformation to ensure 0 < p < 1
  mu <- pnbinom(x, size=r, prob=p)
  logl <- sum(dbinom(y, 1, mu, log=T))
  return(-logl)
}

# likelihood function for demographic stochasticity + among-population heterogeneity (equation 10)
het.binom.lik <- function(param, y, x) {
  disp <- exp(param[2])                     # log transformation to ensure dispersion parameter > 0
  p <- exp(param[1]) / (1 + exp(param[1]))  # logit transformation to ensure 0 < p < 1
  a <- p*disp
  b <- (1-p)*disp
  mu <- 1 - (beta(a, (x+b)) / beta(a, b))
  logl <- sum(dbinom(y, 1, mu, log=T))
  return(-logl)
}

# fit the equations to the data with maximum likelihood
  e4 <- optim(par=c(-20), fn=binom.lik, method="BFGS", y=y, x=x, hessian=T)
  e6 <- optim(par=c(-10,-0.5), fn=allee.binom.lik, method="BFGS", y=y, x=x, hessian=T)
  e10 <- optim(par=c(-2, 2), fn=het.binom.lik, method="BFGS", y=y, x=x, hessian=T)

# fit equation 7 using GLM (=equation 8)
  e7 <- glm(y ~ log(x), family=binomial(link="cloglog"))

# plot the fitted relationship for equation 4
  p.est.fit <- 1 - (1 - exp(e4$par))^xx
  lines(p.est.fit ~ xx, lwd=2, lty=2)

# plot the fitted relationship for equation 6 and calculate theta
  r <- e6$par[2] + 1
  p <- exp(e6$par[1]) / (1+exp(e6$par[1]))
  p.est.allee <- pnbinom(xx, size=r, prob=p)
  lines(p.est.allee ~ xx, lwd=2, lty=3)


# plot the fitted relationship bor equation 7
  pe7 <- coef(e7)[1] + coef(e7)[2]*log(xx)
  p.est.e7 <- 1 - exp(-exp(pe7))
  lines(p.est.e7 ~ xx, lwd=2, lty=4)

# plot the fitted relationship for equation 10
  p <- exp(e10$par[1]) / (1 + exp(e10$par[1]))
  disp <- exp(e10$par[2])
  a <- p*disp
  b <- (1-p)*disp
  p.est.het <- 1 - beta(a, (xx+b)) / beta(a, b)
  lines(p.est.het ~ xx, lwd=2, lty=1)

# show the distribution of p values given the estimated a and b parameters of the beta distribution
from equation 8
  h <- rbeta(1000000, a, b)
```

```
  return(subplot(hist(h, freq=FALSE, breaks=seq(0, max(h)+0.005, 0.005), col="black",
    main="", xlab=expression(italic(p)), ylab="Density"), x=c(220, 300), y=c(0.7, 0.95),
    pars=list(cex.axis=0.8, cex.lab=0.8, tcl=-0.2, mgp=c(1.5,0.5,0))))

# compare AIC values
  n <- length(y)
  aic.opt(e4, n)
  aic.opt(e6, n)
  aic.opt(e10, n)
  AIC(e7)

# fit equation 7 (=equation 8) including species as a random effect
# allowing for different intercepts and then different intercepts and slopes
  e7.2 <- glmer(y ~ log(x) + (1|species), family=binomial(link="cloglog"))
  e7.3 <- glmer(y ~ log(x) + (log(x)|species), family=binomial(link="cloglog"))

  AIC(e7)
  AIC(e7.2)
  AIC(e7.3)

# compare with logit link
  AIC(glm(y ~ log(x), family=binomial(link="logit")))
  AIC(glmer(y ~ log(x) + (1|species), family=binomial(link="logit")))

# function to calculate 95% confidence intervals from optim fit
  ci <- function(x) {
    FI <- solve(x$hessian)
    se <- sqrt(diag(FI))
    ucl <- x$par + 1.96*se
    lcl <- x$par - 1.96*se
    return(cbind(x$par, lcl, ucl))
  }

# calculate 95% confidence intervals
  ci(e6)
  confint(e7)

  e7.2
  0.17932 + 1.96*0.03093
  0.17932 - 1.96*0.03093

# compare the fit of a logit link
  e7.2l <- glmer(y ~ log(x) + (1|species), family=binomial(link="logit"))
  AIC(e7.2)
  AIC(e7.2l)
```