Genetic Rescue

Ben Fitzpatrick

2025-10-09

Genetic rescue despite outbreeding depression

This is an R Markdown document to accompany the manuscript by Ben Fitzpatrick. The functions and commands recapitulate the figures in the manuscript.

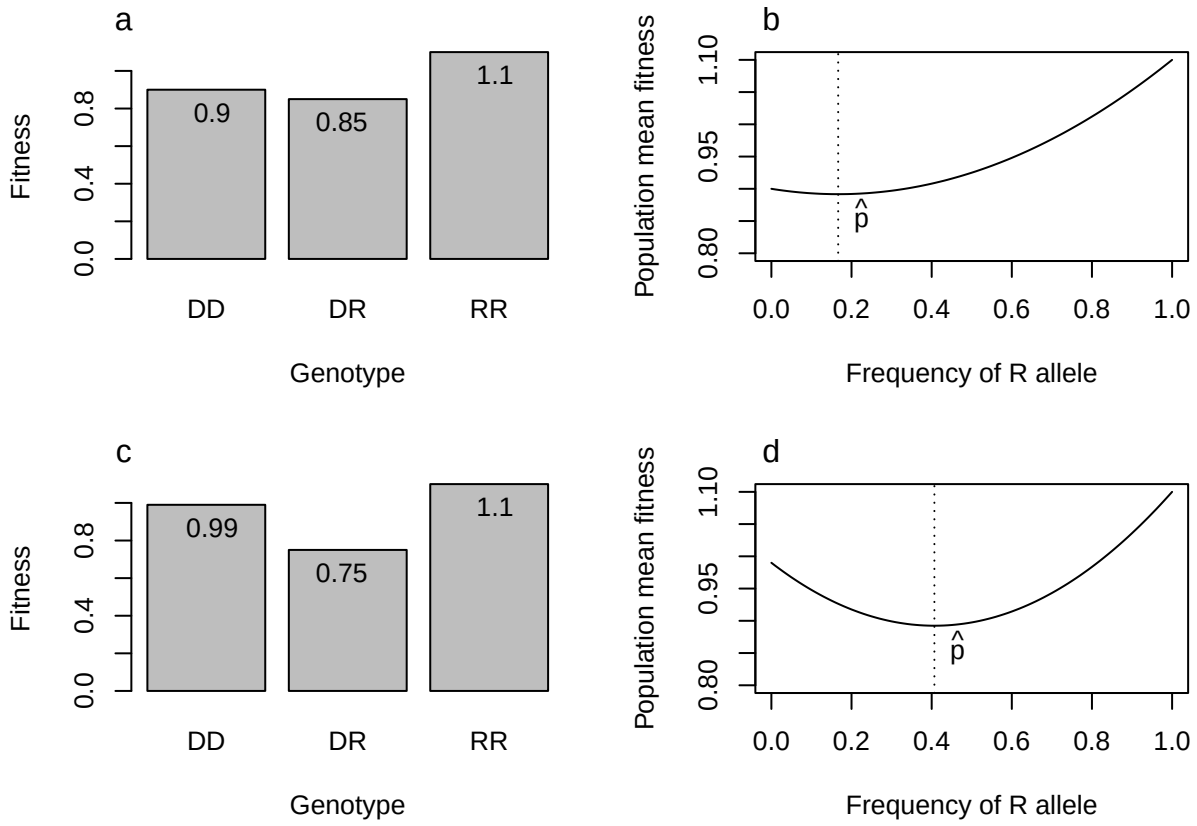
Be sure to set your working directory as appropriate. Activate the 3d plotting library:

```
library(plot3D)
```

Single Locus expectations without population dynamics

```
#quartz("Figure_1", width=5, height=5)
par(mfrow=c(2,2), mar=c(5,5,2,1))

WDD <- 0.90
WDR <- 0.85
WRR <- 1.1

pHat <- (WDD-WDR)/(WRR+WDD-2*WDR)
pvec <- (0:100)/100
wBar <- pvec^2*WRR + 2*pvec*(1-pvec)*WDR + (1-pvec)^2*WDD


barplot(c(WDD,WDR,WRR), ylab="Fitness", xlab="Genotype", names.arg=c("DD", "DR", "RR"))
text(c(WDD,WDR,WRR),x=c(.75,1.85,3.15),y=c(WDD,WDR,WRR),pos=1)
mtext("a", at=.0, line=.5)
plot(pvec, wBar, type="l", xlab="Frequency of R allele", ylab="Population mean fitness", ylim=c(.8,1.1))
abline(v=pHat, lty=3)
text(pHat, 0.85, pos=4, expression(hat(p)))
mtext("b", at=0, line=.5)

WDD <- 0.99
WDR <- 0.75
WRR <- 1.1

pHat <- (WDD-WDR)/(WRR+WDD-2*WDR)
pvec <- (0:100)/100
wBar <- pvec^2*WRR + 2*pvec*(1-pvec)*WDR + (1-pvec)^2*WDD

barplot(c(WDD,WDR,WRR), ylab="Fitness", xlab="Genotype", names.arg=c("DD", "DR", "RR"))
text(c(WDD,WDR,WRR),x=c(.75,1.85,3.15),y=c(WDD,WDR,WRR),pos=1)
mtext("c", at=.0, line=.5)
```

```r
plot(pvec, wBar, type="l", xlab="Frequency of R allele", ylab="Population mean fitness", ylim=c(.8,1.1))
abline(v=pHat, lty=3)
text(pHat, 0.85, pos=4, expression(hat(p)))
mtext("d", at=0, line=.5)
```



Genetic Rescue Simulation: 1 locus

Deterministic finite population model

Define function to iterate classic population genetics model:

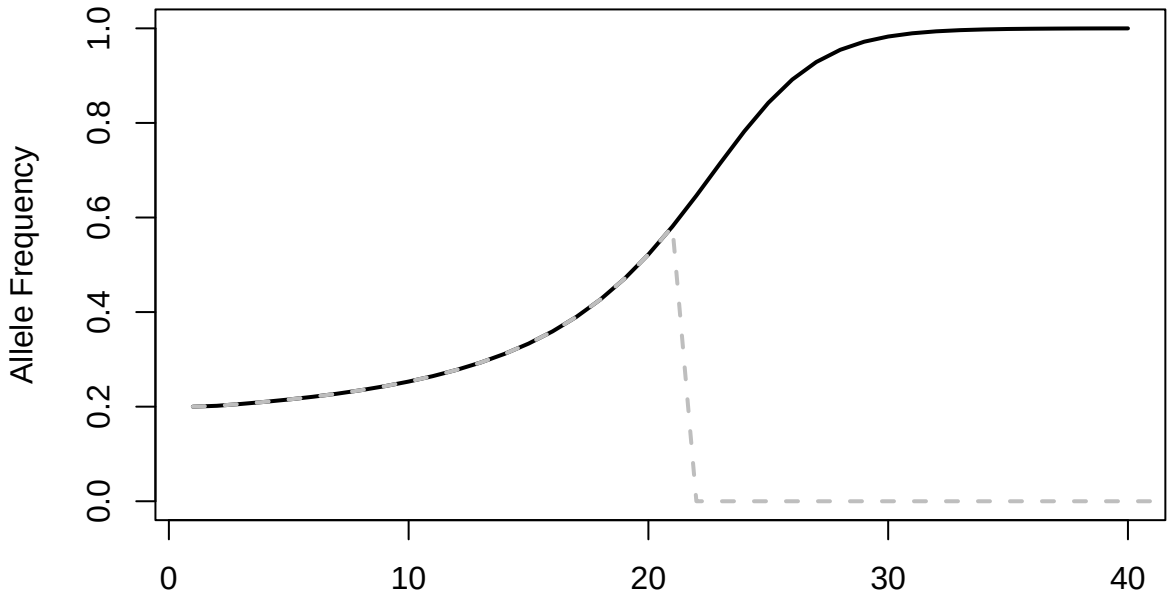
```r
single.locus <- function(G, W, dd, imm=0, gens=100){
        N <- Wbar <- p <- numeric(length=gens) #p is the frequency of the rescue allele
        #1 compute total population size and actual allele frequencies
        N[1] <- sum(G)
        p[1] <- G%*%c(1/N[1], 1/(2*N[1]), 0)
        Wbar[1] <- (G%*%W)/N[1]
    for(i in 1:(gens-1)){
        #2 compute expected allele frequencies in the next generation
        ## selection before random mating
        # p[i+1] <- (G[1]*W[1]+G[2]*W[2]/2)/(N[i]*Wbar[i])
        ## selection after random mating (as in Lewontin & Kojima)
        af <- (G[1]+G[2]/2)/sum(G)
        Wbar[i] <- af^2*W[1]+2*af*(1-af)*W[2]+(1-af)^2*W[3]
        p[i+1] <- (af^2*W[1]+af*(1-af)*W[2])/Wbar[i]
        #3 compute expected population size in the next generation
```

```
        N[i+1] <- Wbar[i]*N[i]/(1+dd*N[i])
        #4 compute new genotype numbers
        G <- N[i+1]*W*c(p[i+1]^2,2*p[i+1]*(1-p[i+1]),(1-p[i+1])^2)
        #5 Add immigrants
        G[1] <- G[1]+imm
        N[i+1] <- N[i+1]+imm
        Wbar[i+1] <- (G%*%W)/N[i+1]
        #6 Return to step 1 or finish loop
        if(N[i+1]<3){break}
    }
    data.frame(N=N, p=p, Wbar=Wbar)
}
```

Now run it for 100 generations with the fitness values in Table 1 of the manuscript with two different starting population sizes:

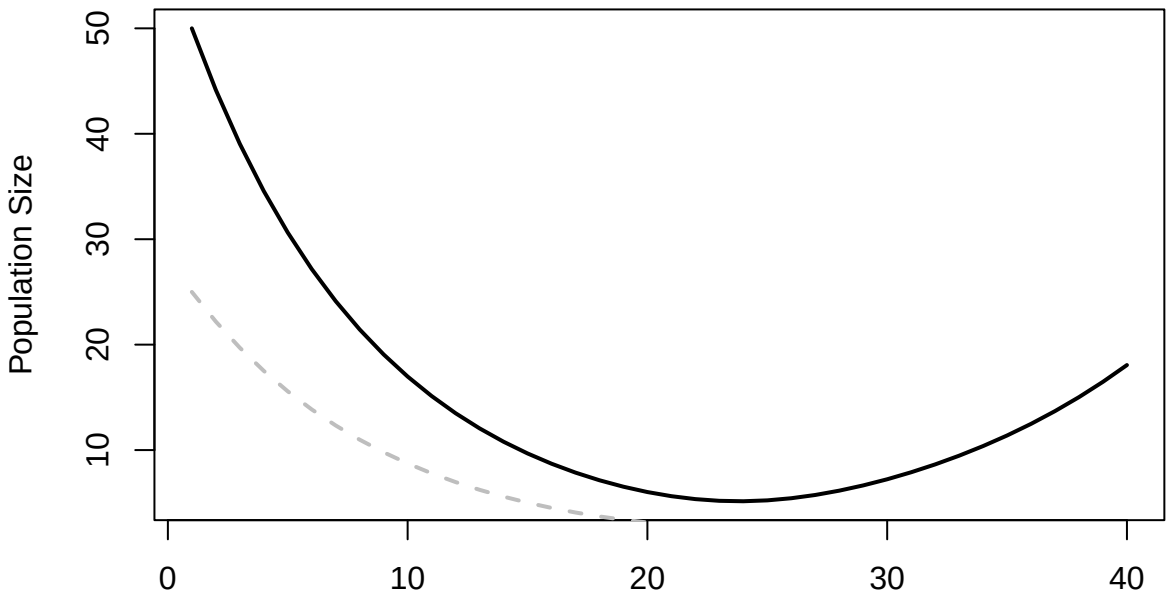
```
gens <- 100
W <- c(1.1, 0.85, 0.9)
imm <- 0

K <- 500
dd <- (max(W)-1)/K
N0 <- 50

G <- c(0.2*N0, 0, 0.8*N0)
N5.20 <- single.locus(G*.5, W, dd, imm, gens)
N10.40 <- single.locus(G, W, dd, imm, gens)

#quartz(title="Figure_2", width=4,height=8)
# par(mfrow=c(3,1), mar=c(3,4,2,1), cex=1)
plot(N10.40$p[1:40], type="l", ylim=c(0,1),xlab="",ylab="Allele Frequency", lwd=2)
lines(N5.20$p, lty=2, col="grey", lwd=2)
#mtext("a", line=0,at=-8, cex=1.2)
title(main="Fig. 1a")
```

Fig. 1a



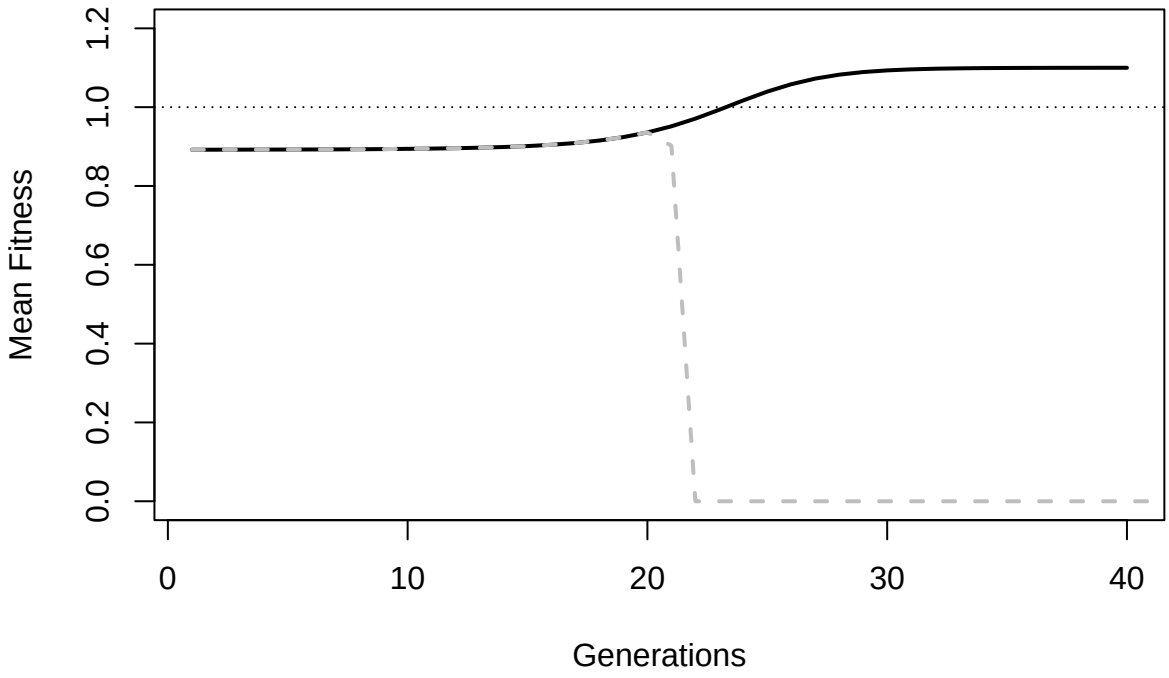

```
#par(mar=c(3.5,4,1.5,1))
plot(N10.40$N[1:40], type="l",xlab="",ylab="Population Size", lwd=2)
lines(N5.20$N, lty=2, col="grey", lwd=2)
#mtext("b", line=0,at=-8, cex=1.2)
title(main="Fig. 1b")
```

Fig. 1b




```
#par(mar=c(4,4,1,1))
plot(N10.40$W[1:40], type="l",xlab="Generations",ylab="Mean Fitness", ylim=c(0,1.2), lwd=2)
lines(N5.20$W, lty=2, col="grey", lwd=2)
#mtext("c", line=0,at=-8, cex=1.2)
title(main="Fig. 1c")
abline(h=1, lty=3)
```

Fig. 1c



Two-locus models

First define the functions to run the computations from Lewontin and Kojima

```
# deterministic two locus model
LK.infinite <- function(G,W,R=0.5,gens){
    Wbar <- p <- r <- numeric(length=gens)
    # distribution of gametotypes before selection
        g11 <- G[1] + G[2]/2 + G[4]/2 + (1-R)*G[5]/2 + R*G[6]/2
        g10 <- G[2]/2 + G[3] + R*G[5]/2 + (1-R)*G[6]/2 + G[7]/2
        g01 <- G[4]/2 + R*G[5]/2 + (1-R)*G[6]/2 + G[8] + G[9]/2
        g00 <- (1-R)*G[5]/2 + R*G[6]/2 + G[7]/2 + G[9]/2 + G[10]
        p[1] <- g11+g10
        r[1] <- g11+g01

    Wbar[1] <- sum(G*W)

    for(i in 2:gens){
    # genotype frequency according to HW (in order of L & K table 4)

        g11 <- round(g11, 10)
        g10 <- round(g10, 10)
        g01 <- round(g01, 10)
        g00 <- round(g00, 10)

        # random union of gametes
```

```
        G <- c(g11*g11, 2*g11*g10, g10*g10, 2*g11*g01, 2*g11*g00, 2*g10*g01,2*g10*g00, g01*g01, 2*g01*g

    # expected frequency after selection
        Wbar[i] <- sum(G*W)
        Gs <- G*W/Wbar[i]

    # new gametes
        g11 <- Gs[1] + Gs[2]/2 + Gs[4]/2 + (1-R)*Gs[5]/2 + R*Gs[6]/2
        g10 <- Gs[2]/2 + Gs[3] + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[7]/2
        g01 <- Gs[4]/2 + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[8] + Gs[9]/2
        g00 <- (1-R)*Gs[5]/2 + R*Gs[6]/2 + Gs[7]/2 + Gs[9]/2 + Gs[10]
    # new allele frequencies
        p[i] <- g11+g10
        r[i] <- g11+g01
    }

    data.frame(p,r,Wbar)
}

#######################################################
# Deterministic model with population regulation
LKBH <- function(G,K,W,R=0.5,gens){
    # Lewontin-Kojima-Beverton-Holt
    dd <- (max(W)-1)/K
    N <- p <- r <- Wbar <- numeric(length=gens)
    N[1] <- sum(G)
    G <- G/N[1]
    # distribution of gametotypes before selection
        g11 <- G[1] + G[2]/2 + G[4]/2 + (1-R)*G[5]/2 + R*G[6]/2
        g10 <- G[2]/2 + G[3] + R*G[5]/2 + (1-R)*G[6]/2 + G[7]/2
        g01 <- G[4]/2 + R*G[5]/2 + (1-R)*G[6]/2 + G[8] + G[9]/2
        g00 <- (1-R)*G[5]/2 + R*G[6]/2 + G[7]/2 + G[9]/2 + G[10]
        p[1] <- g11+g10
        r[1] <- g11+g01

    Wbar[1] <- sum(G*W)

    for(i in 2:gens){
        g11 <- round(g11, 10)
        g10 <- round(g10, 10)
        g01 <- round(g01, 10)
        g00 <- round(g00, 10)

    # random union of gametes
        G <- c(g11*g11, 2*g11*g10, g10*g10, 2*g11*g01, 2*g11*g00, 2*g10*g01,2*g10*g00, g01*g01, 2*g01*g

    # expected frequency after selection
        Wbar[i] <- sum(G*W)
        Gs <- G*W/Wbar[i]
    #expected population size
        N[i] <- Wbar[i]*N[i-1]/(1+dd*N[i-1])

    # new gametes
```

```
        g11 <- Gs[1] + Gs[2]/2 + Gs[4]/2 + (1-R)*Gs[5]/2 + R*Gs[6]/2
        g10 <- Gs[2]/2 + Gs[3] + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[7]/2
        g01 <- Gs[4]/2 + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[8] + Gs[9]/2
        g00 <- (1-R)*Gs[5]/2 + R*Gs[6]/2 + Gs[7]/2 + Gs[9]/2 + Gs[10]
    # new allele frequencies
        p[i] <- g11+g10
        r[i] <- g11+g01
    }

    data.frame(N, p,r,Wbar)

}

#########################################################
# Drift with constant population size
LK.finite <- function(G,W,R=0.5,gens){
    N <- sum(G)
    G <- G/N
    # distribution of gametotypes before selection
        g11 <- G[1] + G[2]/2 + G[4]/2 + (1-R)*G[5]/2 + R*G[6]/2
        g10 <- G[2]/2 + G[3] + R*G[5]/2 + (1-R)*G[6]/2 + G[7]/2
        g01 <- G[4]/2 + R*G[5]/2 + (1-R)*G[6]/2 + G[8] + G[9]/2
        g00 <- (1-R)*G[5]/2 + R*G[6]/2 + G[7]/2 + G[9]/2 + G[10]
        p <- g11+g10
        r <- g11+g01

    Wbar <- sum(G*W)

    for(i in 2:gens){
        # because some weird rounding thing was giving slightly negative values for some that should be
        g11 <- round(g11, 10)
        g10 <- round(g10, 10)
        g01 <- round(g01, 10)
        g00 <- round(g00, 10)

    # genotype frequency according to HW (in order of L & K table 4)
        G <- c(g11*g11, 2*g11*g10, g10*g10, 2*g11*g01, 2*g11*g00, 2*g10*g01,2*g10*g00, g01*g01, 2*g01*g

    # expected frequency after selection
        Ps <- G*W/Wbar[i-1]
    # Wright-Fisher sampling
        Ns <- rmultinom(1,N,Ps)[,1]
        Gs <- Ns/N

    # new gametes
        g11 <- Gs[1] + Gs[2]/2 + Gs[4]/2 + (1-R)*Gs[5]/2 + R*Gs[6]/2
        g10 <- Gs[2]/2 + Gs[3] + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[7]/2
        g01 <- Gs[4]/2 + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[8] + Gs[9]/2
        g00 <- (1-R)*Gs[5]/2 + R*Gs[6]/2 + Gs[7]/2 + Gs[9]/2 + Gs[10]
    # new allele frequencies
        p[i] <- g11+g10
        r[i] <- g11+g01
    Wbar[i] <- sum(W*c(g11*g11, 2*g11*g10, g10*g10, 2*g11*g01, 2*g11*g00, 2*g10*g01,2*g10*g00, g01*g01,
```

```
    }
    data.frame(N,p,r,Wbar)
}

# Beverton-Holt Population Regulation
# with immigration
LK.popi <- function(G, K, W, imm, R=0.5, gens){
    # demographic stochasticity, population regulation, Wright-Fisher sampling
    N <- p <- r <- Wbar <- numeric(length=gens)
    dd <- (max(W)-1)/K
    N[1] <- sum(G)
    G <- G/N[1]
    # distribution of gametotypes before selection
        g11 <- G[1] + G[2]/2 + G[4]/2 + (1-R)*G[5]/2 + R*G[6]/2
        g10 <- G[2]/2 + G[3] + R*G[5]/2 + (1-R)*G[6]/2 + G[7]/2
        g01 <- G[4]/2 + R*G[5]/2 + (1-R)*G[6]/2 + G[8] + G[9]/2
        g00 <- (1-R)*G[5]/2 + R*G[6]/2 + G[7]/2 + G[9]/2 + G[10]
    p[1] <- g11+g10
    r[1] <- g11+g01

    Wbar[1] <- sum(G*W)

    for(i in 2:gens){
        g11 <- round(g11, 10)
        g10 <- round(g10, 10)
        g01 <- round(g01, 10)
        g00 <- round(g00, 10)
    # random mating
    # new genotype frequency according to HW (in order of L & K table 4)
        G <- c(g11*g11, 2*g11*g10, g10*g10, 2*g11*g01, 2*g11*g00, 2*g10*g01,2*g10*g00, g01*g01, 2*g01*g

    # expected frequency after selection
        Wbar[i] <- sum(W*G)
        Ps <- G*W/Wbar[i]
    #expected population size
        N[i] <- Wbar[i]*N[i-1]/(1+dd*N[i-1])
    #Draw random population size
        N[i] <- rpois(1,N[i])
        if(N[i]<2) break
    # Wright-Fisher sampling
        Ns <- rmultinom(1,N[i],Ps)[,1]
    # add immigrants
        Ns[10] <- Ns[10]+imm
        N[i] <- N[i]+imm
        Gs <- Ns/N[i]

    # new gametes
        g11 <- Gs[1] + Gs[2]/2 + Gs[4]/2 + (1-R)*Gs[5]/2 + R*Gs[6]/2
        g10 <- Gs[2]/2 + Gs[3] + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[7]/2
        g01 <- Gs[4]/2 + R*Gs[5]/2 + (1-R)*Gs[6]/2 + Gs[8] + Gs[9]/2
        g00 <- (1-R)*Gs[5]/2 + R*Gs[6]/2 + Gs[7]/2 + Gs[9]/2 + Gs[10]
    # new allele frequencies
        p[i] <- g11+g10
```

```
        r[i] <- g11+g01

    }

    data.frame(N,p,r,Wbar)
}
```

Note: For the two locus simulations, I saved the basic functions above as a script to read in with the 'source' command: source("NewTwoLocusModels.R")

Neutral hitchhiker

```
N <- 50
K <- 500
p <- 0.80 # frequency of the neutral native allele
r <- 0.80 # frequency of the deleterious native allele
LD <- p*(1-r) # initial admixture = maximum correlation
G <- c(0.8,rep(0,8),0.2)
# multilocus fitness function (double-heterozygote twice)
W <- c(0.9, .85, 1.1,
        0.9, .85, .85, 1.1,
        0.9, .85, 1.1)
wmat <- matrix(W[-5],nrow=3)
colnames(wmat) <- c("AA","AB","BB")
rownames(wmat) <- c("DD","DR","RR")
t(wmat)
```

```
##     DD   DR  RR
## AA 0.9 0.85 1.1
## AB 0.9 0.85 1.1
## BB 0.9 0.85 1.1
```

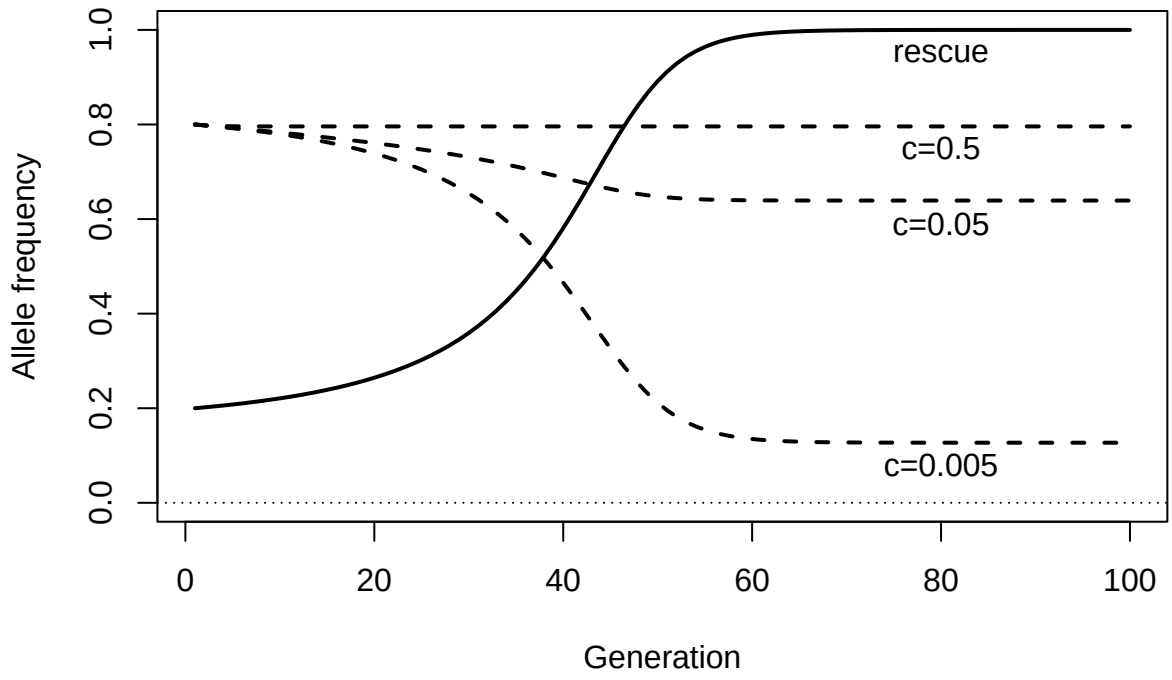
```
hitch0.50 <- LKBH(G, K, W, R=0.5, 100)
hitch0.05 <- LKBH(G, K, W, R=0.05, 100)
hitch0.005 <- LKBH(G, K, W, R=0.005, 100)

# stronger selection, keeping phat<0.2: Wrr=(Wdd-Wdr)/Phat -Wdd+2Wdr
Phat <- 0.2
Wdd <- 0.85
Wdr <- 0.72
(Wrr <- (Wdd-Wdr)/Phat-Wdd+2*Wdr+0.06)
```

```
## [1] 1.3
```

```
W <- c(Wdd, Wdr, Wrr,
        Wdd, Wdr, Wdr, Wrr,
        Wdd, Wdr, Wrr)
hitchS.50 <- LKBH(G, K, W, R=0.5, 100)
hitchS.05 <- LKBH(G, K, W, R=0.05, 100)
hitchS.005 <- LKBH(G, K, W, R=0.005, 100)

#quartz("hitchhiking effects", width=4, height=6.5)
#par(mfrow=c(2,1), mar=c(4,4,2,1))
plot(hitch0.005$p, ylim=c(0,1), xlab="Generation", ylab="Allele frequency", type="l", lwd=2, lty=2)
lines(hitch0.05$p, lwd=2, lty=2)
```

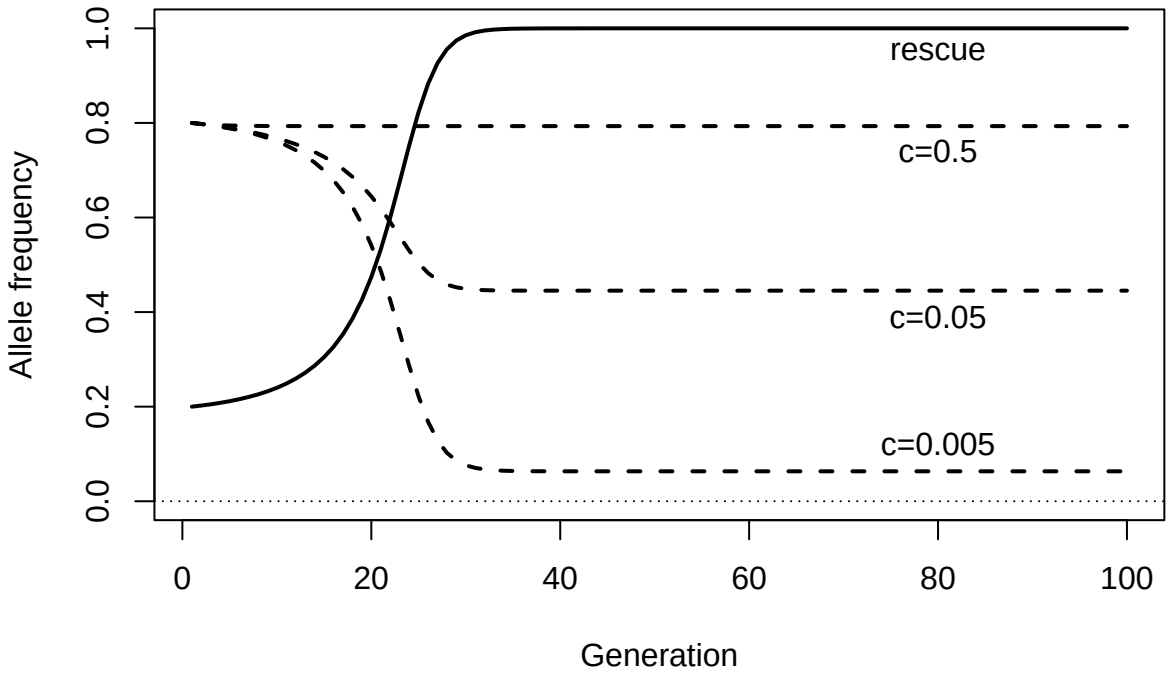
```
lines(hitch0.50$p, lwd=2, lty=2)
lines(1-hitch0.50$r, lwd=2)
text(80,.95,"rescue")
text(80,.75,"c=0.5")
text(80,.59,"c=0.05")
text(80,.08,"c=0.005")
abline(h=0,lty=3)
#mtext("a", at=-23, cex=1.2)
title(main="Fig. 3a")
```

Fig. 3a




```
plot(hitchS.005$p, ylim=c(0,1), xlab="Generation", ylab="Allele frequency", type="l", lwd=2, lty=2)
lines(hitchS.05$p, lwd=2, lty=2)
lines(hitchS.50$p, lwd=2, lty=2)
lines(1-hitchS.50$r, lwd=2)
text(80,.95,"rescue")
text(80,.74,"c=0.5")
text(80,.39,"c=0.05")
text(80,.12,"c=0.005")
abline(h=0,lty=3)
#mtext("b", at=-23, cex=1.2)
title(main="Fig. 3b")
```

Fig. 3b



With local adaptation in conflict

This chunk will create the right-hand side of figure 4, showing the fitness functions and deterministic trajectories of the rescue allele (solid line) and native local advantage allele (dashed line):

```
N <- 50
K <- 500
sa <- 0.15 # additive effect of immigrant allele at the local adaptation locus
sr <- 0.10 # additive effect of immigrant allele at the rescue locus
Wn <- 0.90 # native fitness

# multilocus fitness function (double-heterozygote twice)
WA <- Wn + c(0, sr, 2*sr,
             -sa, -sa+sr, -sa+sr, -sa+2*sr,
             -2*sa, -2*sa+sr, -2*sa+2*sr)
wmat <- matrix(WA[-5],nrow=3)
colnames(wmat) <- c("AA","AB","BB")
rownames(wmat) <- c("DD","DR","RR")
t(wmat)
```

```
##      DD   DR   RR
## AA 0.90 1.00 1.10
## AB 0.75 0.85 0.95
## BB 0.60 0.70 0.80
```

```
# Dobzhansky-Muller
sa <- 0.10
```

```
h0 <- 0.05
h1 <- 0.10
h2 <- 0.15
WD <- Wn + c(0, sr, 2*sr,
             -sa-h1, -sa+sr-h0, -sa+sr-h0, -sa+2*sr,
             -2*sa-h2, -2*sa+sr-h1, -2*sa+2*sr)
wmat <- matrix(WD[-5],nrow=3)
colnames(wmat) <- c("AA","AB","BB")
rownames(wmat) <- c("DD","DR","RR")
t(wmat)
```

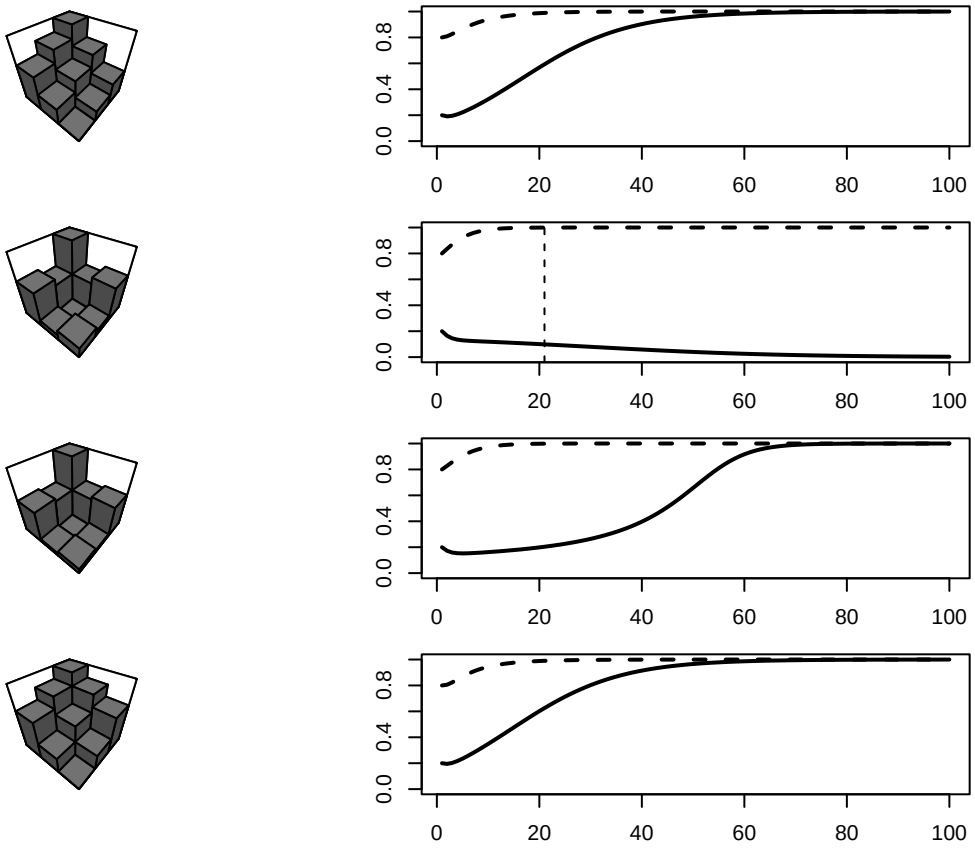
```
##      DD   DR  RR
## AA 0.90 1.00 1.1
## AB 0.70 0.85 1.0
## BB 0.55 0.70 0.9
```

```
# two loci with heterozygote dysfunction
sa <- 0.10
d1 <- 0.15
d2 <- 0.15
WE <- Wn + c(0, sr-d1, 2*sr,
             -sa-d2, -sa+sr-d1-d2, -sa+sr-d1-d2, -sa+2*sr-d2,
             -2*sa, -2*sa+sr-d1, -2*sa+2*sr)
wmat <- matrix(WE[-5],nrow=3)
colnames(wmat) <- c("AA","AB","BB")
rownames(wmat) <- c("DD","DR","RR")
t(wmat)
```

```
##      DD   DR   RR
## AA 0.90 0.85 1.10
## AB 0.65 0.60 0.85
## BB 0.70 0.65 0.90
```

```
d1 <- 0.12
d2 <- 0.12
WE2 <- Wn + c(0, sr-d1, 2*sr,
              -sa-d2, -sa+sr-d1-d2, -sa+sr-d1-d2, -sa+2*sr-d2,
              -2*sa, -2*sa+sr-d1, -2*sa+2*sr)

# unlinked (R=0.5)
add1 <- LKBH(N*G, K, WA, 0.5, 100)
het1 <- LKBH(N*G, K, WE, 0.5, 100)
het2 <- LKBH(N*G, K, WE2, 0.5, 100)
bdm1 <- LKBH(N*G, K, WD, 0.5, 100)

# quartz("Figure_4_part")
par(mfrow=c(4,2), mar=c(2,2,1,1))

hist3D(z=t(matrix(WA[-5], nrow=3)), colvar=FALSE, border="black", col="grey", lighting=TRUE, axes=FALSE

plot(add1$p, type="l", ylim=c(0,1), lwd=2, lty=2); lines(1-add1$r, lwd=2)

hist3D(z=t(matrix(WE[-5], nrow=3)), colvar=FALSE, border="black", col="grey", lighting=TRUE, axes=FALSE

plot(het1$p, type="l", ylim=c(0,1), lwd=2, lty=2); lines(1-het1$r, lwd=2)
```

```
abline(v=which(het1$N<3)[1], lty=2)

hist3D(z=t(matrix(WE2[-5], nrow=3)), colvar=FALSE, border="black", col="grey", lighting=TRUE, axes=FALSE

plot(het2$p, type="l", ylim=c(0,1), lwd=2, lty=2); lines(1-het2$r, lwd=2)

hist3D(z=t(matrix(WD[-5], nrow=3)), colvar=FALSE, border="black", col="grey", lighting=TRUE, axes=FALSE)

plot(bdm1$p, type="l", ylim=c(0,1), lwd=2, lty=2); lines(1-bdm1$r, lwd=2)
```



Stochastic model

To recreate my stochastic outcomes exactly, use

```
set.seed(42)
```

stochastic two locus model

neutral, unlinked hitchhiker withOUT immigration

```
N <- 50
K <- 500
G <- round(N*c(0.8,0,0,0,0,0,0,0,0,0.2))
```

```
# multilocus fitness function (double-heterozygote twice)
W <- c(0.9, .85, 1.1,
        0.9, .85, .85, 1.1,
        0.9, .85, 1.1)

gens <- 100

LK.infinite(G/50,W,gens=100)[100,]
```

```
##            p           r Wbar
## 100 0.795928 3.56691e-07  1.1
```

Just confirming that rescue is expected: deleterious allele dissappears (r ~0). To clarify: p is the frequency of native allele at the 'second' locus, and r is the frequency of the native (deleterios allele) at the 'rescue' locus. Now run the stochastic model with the same parameters:

```
reps <- 10000
imm <- 0
N2 <- P2 <- R2 <- matrix(nrow=gens, ncol=reps)
for(i in 1:reps){
    Xs <- LK.popi(G, K, W, imm, R=0.5, gens=gens)
    N2[,i] <- Xs$N
    P2[,i] <- Xs$p
    R2[,i] <- Xs$r
}

sum(N2[gens,]>2,na.rm=TRUE)
```

```
## [1] 2044
```

20.44% (2044/10000) success rate

```
mean(P2[gens,N2[gens,]!=0], na.rm=TRUE)
```

```
## [1] 0.7569408
```

0.7569 average native neutral allele frequency GIVEN population survival

```
mean(P2[gens,N2[gens,]!=0]==0, na.rm=TRUE)
```

```
## [1] 0.08414873
```

0.0841 fixation probability of nonnative neutral alleles (the command finds the proportion of simulations where the native allele freqeuncy went to zero)

neutral, unlinked hitchhiker with immigration

```
imm <- 1
N2i <- P2i <- R2i <- matrix(nrow=gens, ncol=reps)
for(i in 1:reps){
    Xs <- LK.popi(G, K, W, imm, R=0.5, gens=gens)
    N2i[,i] <- Xs$N
    P2i[,i] <- Xs$p
    R2i[,i] <- Xs$r
}
```

```r
sum(N2i[gens,]>2,na.rm=TRUE)
```

```
## [1] 9648
```

96.48% success rate

```r
mean(P2i[gens,N2i[gens,]>2], na.rm=TRUE)
```

```
## [1] 0.1803377
```

0.1803 . . . i.e., the average introduced AF is 81.97%

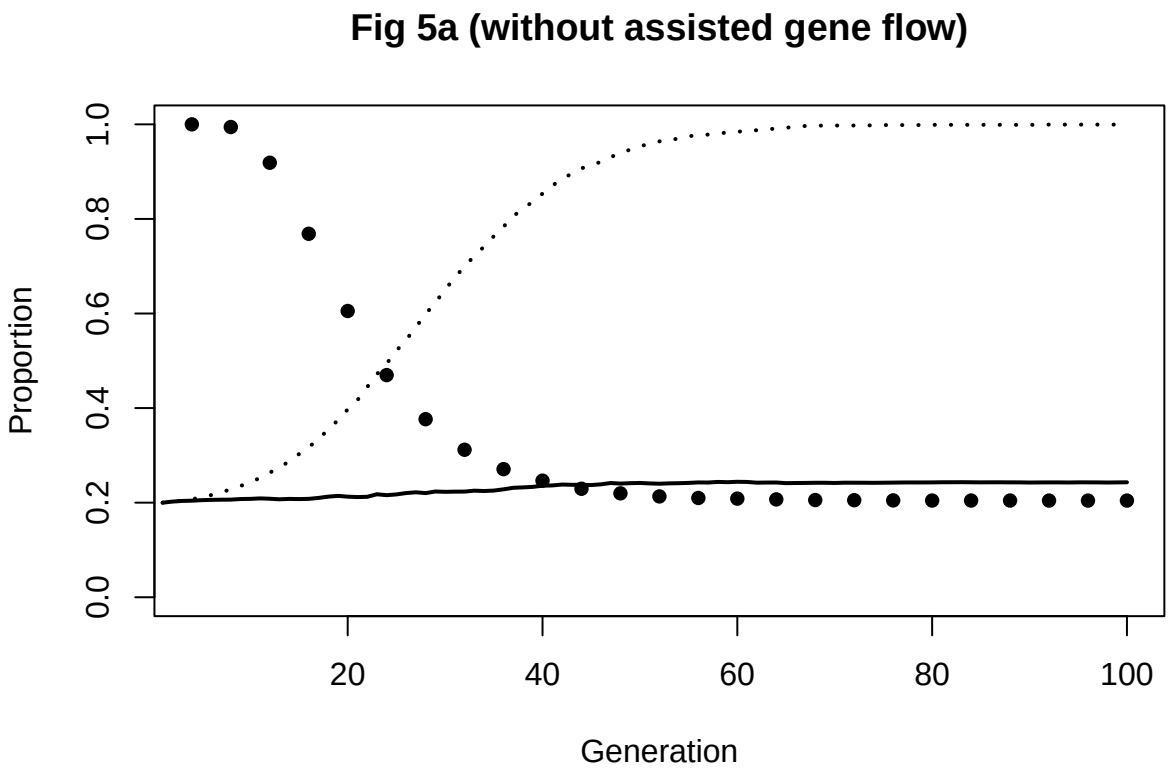
```r
mean(P2i[gens,N2i[gens,]>2]==0, na.rm=TRUE)
```

```
## [1] 0.331364
```

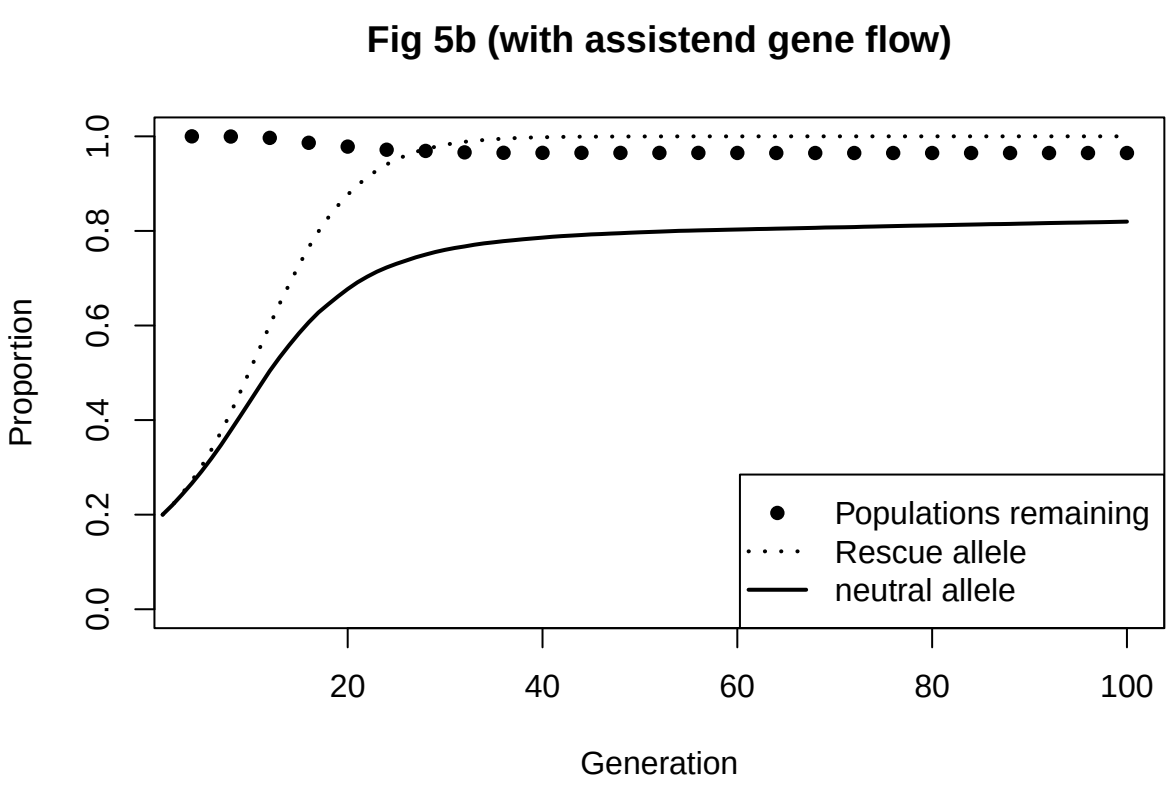
0.3313 fixation prob (proportion of simulations where native neutral allele was lost)

Now make Figure 5

```r
Pext0 <- rowMeans(N2>2)
Pexti <- rowMeans(N2i>2)
p2 <- replace(P2,N2<3,NA)
r2 <- replace(R2,N2<3,NA)
Pneu0 <- rowMeans(p2,na.rm=TRUE)
Pres0 <- rowMeans(r2,na.rm=TRUE)
p2i <- replace(P2i,N2i<3,NA)
r2i <- replace(R2i,N2i<3,NA)
Pneui <- rowMeans(p2i,na.rm=TRUE)
Presi <- rowMeans(r2i,na.rm=TRUE)
# quartz("Figure_5", width=4,height=6.5)

# Fig 5
# par(mfrow=c(2,1), mar=c(4,5,3,1))
spar <- 4*(1:25)
x <- 1:100
plot(x[spar],Pext0[spar], xlab="Generation", ylab="Proportion", pch=16, ylim=c(0,1))
lines(1:100,1-Pneu0, lwd=2)
lines(1:100,1-Pres0,lwd=2, lty=3)
#mtext("a: Without assisted gene flow", line=0.75, at=10)
title(main="Fig 5a (without assisted gene flow)")
```




```
plot(x[spar],Pexti[spar], xlab="Generation", ylab="Proportion", pch=16, ylim=c(0,1))
lines(1:100,1-Pneui, lwd=2)
lines(1:100,1-Presi,lwd=2, lty=3)
#mtext("b: With assisted gene flow", line=0.75, at=7)
legend("bottomright", c("Populations remaining", "Rescue allele", "neutral allele"), lty=c(0,3,1), lwd=

title(main="Fig 5b (with assistend gene flow)")
```



repeat with larger population (for comparison in Fig.6)

```
N <- 100
K <- 500
G <- round(N*c(0.8,0,0,0,0,0,0,0,0,0.2))

# multilocus fitness function (double-heterozygote twice)
W <- c(0.9, .85, 1.1,
       0.9, .85, .85, 1.1,
       0.9, .85, 1.1)

gens <- 100

reps <- 10000
imm <- 0
N2 <- P2 <- R2 <- matrix(nrow=gens, ncol=reps)
for(i in 1:reps){
    Xs <- LK.popi(G, K, W, imm, R=0.5, gens=gens)
    N2[,i] <- Xs$N
    P2[,i] <- Xs$p
    R2[,i] <- Xs$r
}

# success rate
sum(N2[gens,]>2,na.rm=TRUE)
```

```
## [1] 2491
```

```r
# average non-native neutral allele frequency GIVEN persistence
1-mean(P2[gens,N2[gens,]!=0], na.rm=TRUE)
```

```
## [1] 0.2331452
```

```r
# Fixation probability of non-native neutral allele
mean(P2[gens,N2[gens,]!=0]==0, na.rm=TRUE)
```

```
## [1] 0.07544141
```

```r
# neutral, unlinked hitchhiker with immigration
imm <- 1
N2i <- P2i <- R2i <- matrix(nrow=gens, ncol=reps)
for(i in 1:reps){
    Xs <- LK.popi(G, K, W, imm, R=0.5, gens=gens)
    N2i[,i] <- Xs$N
    P2i[,i] <- Xs$p
    R2i[,i] <- Xs$r
}

# success rate
sum(N2i[gens,]>2,na.rm=TRUE)
```

```
## [1] 9768
```

```r
# average non-native neutral allele frequency GIVEN persistence
1-mean(P2i[gens,N2i[gens,]>2], na.rm=TRUE)
```

```
## [1] 0.7786604
```

```r
# Fixation probability of non-native neutral allele
mean(P2i[gens,N2i[gens,]>2]==0, na.rm=TRUE)
```

```
## [1] 0.2623874
```

```r
Pext0.1 <- rowMeans(N2>2)
Pexti.1 <- rowMeans(N2i>2)
p2.1 <- replace(P2,N2<3,NA)
r2.1 <- replace(R2,N2<3,NA)
Pneu0.1 <- rowMeans(p2.1,na.rm=TRUE)
Pres0.1 <- rowMeans(r2.1,na.rm=TRUE)
p2i.1 <- replace(P2i,N2i<3,NA)
r2i.1 <- replace(R2i,N2i<3,NA)
Pneui.1 <- rowMeans(p2i.1,na.rm=TRUE)
Presi.1 <- rowMeans(r2i.1,na.rm=TRUE)
```

With 1 immigrant per generation, the expected frequency of introduced alleles at unlinked neutral loci goes to 0.779, and they are fixed 26% of the time (or 26% of loci) ... so, would it be better to increase the initial introduction and avoid the continual gene flow?

```r
# vector of different numbers of individuals to be translocated
Trans <- seq(from=10, to=60, by=5)
# initial vectors to keep track of outcomes at the end of each simulation
Succ <- numeric() # how often the population is extant at the end
Neut <- numeric() # neutral non-native allele frequency
Rfre <- numeric() # rescue allele frequency
Rinc <- numeric() # is the rescue allele frequency greater than it started
```

```
for(j in 1:length(Trans)){
N2 <- P2 <- R2 <- matrix(nrow=gens, ncol=reps)
G <- c(40,rep(0,8),Trans[j])
for(i in 1:reps){
    Xs <- LK.popi(G, K, W, imm=0, R=0.5, gens=gens)
    N2[,i] <- Xs$N
    P2[,i] <- Xs$p
    R2[,i] <- Xs$r
}

Succ[j] <- sum(N2[gens,]>2,na.rm=TRUE)
Neut[j] <- 1-mean(P2[gens,N2[gens,]>2], na.rm=TRUE)
Rfre[j] <- 1-mean(R2[gens,N2[gens,]>2], na.rm=TRUE)
Rinc[j] <- mean(R2[gens,N2[gens,]>2]<(Trans[j]/sum(G)), na.rm=TRUE)

}

(simout <- data.frame(Trans,Succ,Neut,Rfre,Rinc))
```

```
##    Trans Succ      Neut      Rfre      Rinc
## 1     10 2025 0.2347028 1.0000000 1.0000000
## 2     15 3736 0.3403564 0.9997323 0.9997323
## 3     20 5388 0.3884738 1.0000000 1.0000000
## 4     25 6929 0.4475128 1.0000000 1.0000000
## 5     30 7999 0.4940513 1.0000000 1.0000000
## 6     35 8782 0.5294551 0.9998861 0.9998861
## 7     40 9336 0.5683618 1.0000000 1.0000000
## 8     45 9607 0.5951595 1.0000000 1.0000000
## 9     50 9791 0.6202137 1.0000000 1.0000000
## 10    55 9896 0.6475673 1.0000000 1.0000000
## 11    60 9955 0.6726314 1.0000000 1.0000000
```

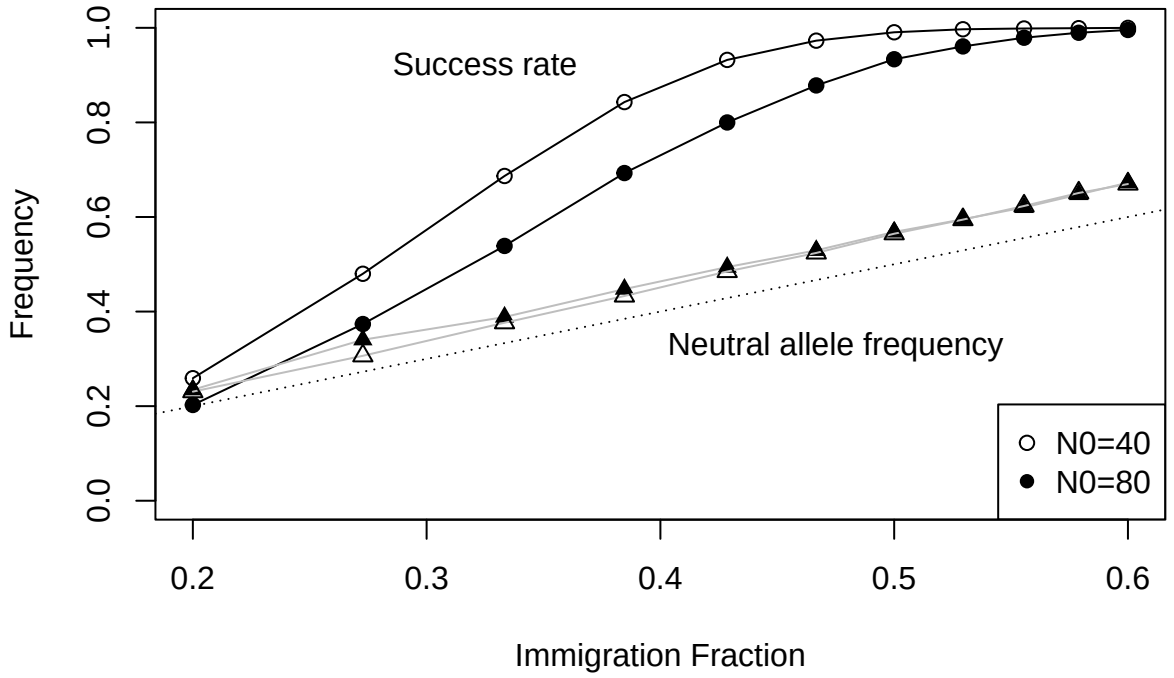
Rinc is the fraction of simulations where the population was extant at the end and the rescue allele was increasing. That is, if this is less than one, the population is extant after the specified number of generations but probably going extinct.

For example 0.9997323*3736 = 3735, aka one simulation was likely heading toward failure but hadn't gotten there yet because the agreement of Rfre and Rinc means the frequency of the rescue allele was zero ...

add a different scenario (double the population size)...

```
for(j in 1:length(Trans)){
N2 <- P2 <- R2 <- matrix(nrow=gens, ncol=reps)
G <- c(40,rep(0,8),Trans[j])*2 # doubling here
for(i in 1:reps){
    Xs <- LK.popi(G, K=500, W, imm=0, R=0.5, gens=gens)
    N2[,i] <- Xs$N
    P2[,i] <- Xs$p
    R2[,i] <- Xs$r
}
Succ[j] <- sum(N2[gens,]>2,na.rm=TRUE)
Neut[j] <- 1-mean(P2[gens,N2[gens,]>2], na.rm=TRUE)
Rfre[j] <- 1-mean(R2[gens,N2[gens,]>2], na.rm=TRUE)
Rinc[j] <- mean(R2[gens,N2[gens,]>2]<(Trans[j]/sum(G)), na.rm=TRUE)
}
```

```r
#(data.frame(IF,Succ,Neut,simout))

#quartz("Figure_6",width=4.5,height=4.5)
IF <- Trans/(Trans+40) # immigrant fraction
Psuc <- simout$Succ/reps
plot(IF,Psuc, ylim=c(0,1), type="l", col="black", xlab="Immigration Fraction", ylab="Frequency")
points(IF,Psuc,pch=21,bg="black")
lines(IF,simout$Neut, col="grey")
points(IF, simout$Neut, pch=17)
text(0.325,0.925,"Success rate", col="black")
text(0.475,0.325, "Neutral allele frequency")

lines(IF,Succ/reps, col="black")
points(IF,Succ/reps, col="black")
lines(IF,Neut, col="grey")
points(IF,Neut, col="black", pch=2)
abline(0,1,lty=3)
legend("bottomright",legend=c("N0=40","N0=80"),pch=c(1,16))
```




```r
data.frame(IF,simout[,-1],Succ,Neut,Rfre,Rinc)
```

```
##           IF Succ      Neut      Rfre      Rinc Succ.1    Neut.1 Rfre.1 Rinc.1
## 1  0.2000000 2025 0.2347028 1.0000000 1.0000000   2591 0.2306510      1      1
## 2  0.2727273 3736 0.3403564 0.9997323 0.9997323   4799 0.3064577      1      1
## 3  0.3333333 5388 0.3884738 1.0000000 1.0000000   6867 0.3763777      1      1
## 4  0.3846154 6929 0.4475128 1.0000000 1.0000000   8430 0.4329700      1      1
```

```
## 5  0.4285714 7999 0.4940513 1.0000000 1.0000000   9322 0.4847525      1      1
## 6  0.4666667 8782 0.5294551 0.9998861 0.9998861   9729 0.5239691      1      1
## 7  0.5000000 9336 0.5683618 1.0000000 1.0000000   9906 0.5646198      1      1
## 8  0.5294118 9607 0.5951595 1.0000000 1.0000000   9970 0.5944820      1      1
## 9  0.5555556 9791 0.6202137 1.0000000 1.0000000   9986 0.6232608      1      1
## 10 0.5789474 9896 0.6475673 1.0000000 1.0000000   9993 0.6508507      1      1
## 11 0.6000000 9955 0.6726314 1.0000000 1.0000000   9999 0.6697314      1      1
```

This last data frame compares success rates at different immigrant fractions for the small and large population sizes (columns with ".1").