

Table 1: Summary of choices made in previous models	
	Lachlan and Servedio [2]
$\sigma$ trait(s)	allele A / a: song predisposition
	song
How $\sigma$ trait is learned	
Inherited	
$\varphi$ trait(s)	
How $\varphi$ trait is learned	
Inherited	
Errors	
Population structure	
Mating structure	
$\varphi$ preference	
Selection	

## 1 Question

1. H1 Song learning can decelerate speciation by allowing for genetically diverse birds to mate with each other, maintaining gene flow between subpopulations that might either begin to diverge
2. H2 Song learning can accelerate speciation by increasing standing genetic variation, which would allow for quicker divergence once new selection pressures arise [2]
3. H3 Song learning can accelerate speciation because culturally inherited traits can evolve more quickly than genetically inherited ones [1]

## 2 Goals

## 3 Model

**3.1 Continuous traits** Each male has a song and each female has preference for a particular song. She will mate with males with songs other than her preferred song, but the probability of her doing so decreases as the potential mate's song gets less similar to her preferred song. Each female mates once and chooses a male according to her preferences and the distribution of songs present in the population. Each male, therefore, may breed multiple times or not at all. We assume that generations are non-overlapping, so once the adults breed they die and we can shift our focus to the new generation. To begin with,

we assume that males acquire their songs from their fathers at birth and females acquire their preferences from their mothers at birth. Before the new generation mates, each male has a small probability of “mutating” its song. This can be interpreted as a learning error or as innovation. After mutation, the new males and females mate.

Mathematically, each male has a song  $x \in \mathbb{R}$  and each female has a preference  $y \in \mathbb{R}$ .

```
Tsteps = 1000 #how many generations
step = 0.01 #step size of trait space
int_step = step #step to use for integration function
mut_prob = 0.01 #probability a male changes song to one on either side
mut_delta = 0 #how to implement mutations of different sizes?
alpha = 0.5 #if preference function is a step fx, strength of preference
sigma2 = 1 #variance of female preference function
fmix_sigma2 = 0.4 #variance of female distribution(s)
mmix_sigma2 = 0.1 #variance of male distribution(s)

mrange = seq(-10,10,by=step) #range of male songs
Nm = length(mrange)
mmin = -1
mmax = 1
m0 = which(mrange==mmin)
m1 = which(mrange==mmax)
mrange_orig = seq(mmin,mmax,by=step)
frange = seq(-10,10,by=step) #range of female preferences
Nf = length(frange)
fmin = -1
fmax = 1
f0 = which(frange==fmin)
f1 = which(frange==fmax)
frange_orig = seq(fmin,fmax,by=step)
```

## PEAKS

```
mut_prob = 0.01
Tsteps = 100
sigma2 = 0.1
fmix_sigma2 = 1

m_init = array(0, dim = c(Nm,1))
m_init[m0] = 0.6
m_init[m1] = 0.4
m_init[,1] = m_init[,1]/int(m_init[,1])

p = .4
```

```

f_init = p*dnorm(frange,-1,fmix_sigma2)+(1-p)*dnorm(frange,1,fmix_sigma2)

Pm = matrix(0,Nm,Tsteps+1) #probability of male songs over time
Pm[,1] = m_init

Pf = matrix(0,Nf,Tsteps+1) #probability of female preferences over time
Pf[,1] = f_init

# t = 1
for(t in 1:Tsteps){
  Pm_adults = Pm[,t]
  Pf_adults = Pf[,t]
  pxy = matrix(0,Nm,Nf) #probability of a (x,y) pair
  ### should I round Pm_adults?!? how?!? is that why I'm getting bumps?!?
  for(j in 1:Nf){
    y = frange[j]
    # weight = 1/sqrt(2*pi*sigma2)*exp(-(mrange-y)^2/(2*sigma2))
    weight = dnorm(mrange,mean=y,sd=sqrt(sigma2)) #female preference function
    # weight = matrix(0,Nf,1)
    # weight[c(f0,x1)] = 1
    # weight[j] = 1+alpha
    z = int(weight*Pm_adults) #normalization factor
    if(z!=0){
      pxy[,j] = Pf_adults[j]*weight*Pm_adults/z
    }
  }
  Pm_beforemut = matrix(0,Nm)
  for(i in 1:Nm){
    Pm_beforemut[i] = int(pxy[i,]) #probability of males being born
  }
  Pm_aftermut = matrix(0,Nm)
  Pm_aftermut = (1-mut_prob)*Pm_beforemut + mut_prob/2*c(Pm_beforemut[2:Nm],0) + mut_prob/2*c(Pm_beforemut[1],Pm_beforemut[2:Nm])
  Pm[,t+1] = Pm_aftermut
  Pf[,t+1] = Pf_adults
}

v = Pm[,1]
Tsteps = 1
m_init = v
m_init = m_init/int(m_init)

p = .4
f_init = p*dnorm(frange,-1,fmix_sigma2)+(1-p)*dnorm(frange,1,fmix_sigma2)
# f_init = .3*dnorm(frange,-1,fmix_sigma2)+.3*dnorm(frange,0,fmix_sigma2)+.4*dnorm(frange,1,fmix_sigma2)

```

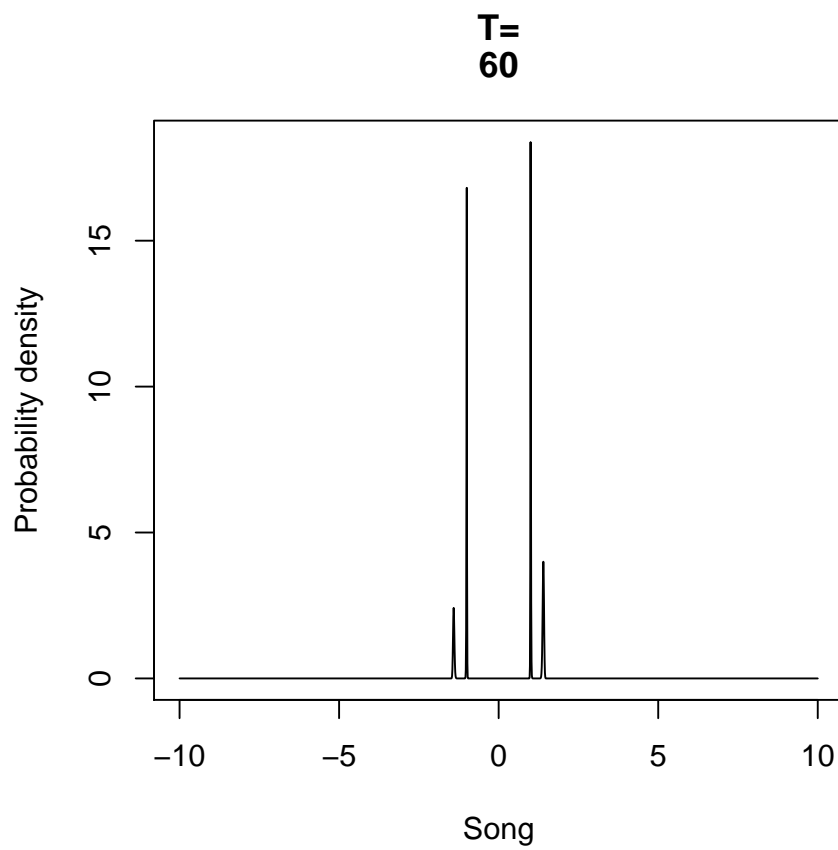


Figure 1: Probability density of male songs. Notice the discrete peaks.

```

Pm2 = matrix(0,Nm,Tsteps+1)
Pm2[,1] = m_init

Pf2 = matrix(0,Nf,Tsteps+1)
Pf2[,1] = f_init

# t = 1
for(t in 1:Tsteps){
  Pm_adults = Pm2[,t]
  Pf_adults = Pf2[,t]
  pxy = matrix(0,Nm,Nf)

  for(j in 1:Nf){
    y = frange[j]
    # weight = 1/sqrt(2*pi*sigma2)*exp(-(mrange-y)^2/(2*sigma2))
    weight = dnorm(mrange,mean=y,sd=sqrt(sigma2))
    # weight = matrix(0,Nf,1)
    # weight[c(f0,x1)] = 1
    # weight[j] = 1+alpha
    z = int(weight*Pm_adults)
    if(z!=0){
      pxy[,j] = Pf_adults[j]*weight*Pm_adults/z
    }
  }
  Pm_beforemut = matrix(0,Nm)
  for(i in 1:Nm){
    Pm_beforemut[i] = int(pxy[i,])
  }
  Pm_aftermut = matrix(0,Nm)
  Pm_aftermut = (1-mut_prob)*Pm_beforemut + mut_prob/2*c(Pm_beforemut[2:Nm],0) + mut_prob/2*c(Pm_beforemut[1],Pm_beforemut[2:Nm])
  Pm2[,t+1] = Pm_aftermut
  Pf2[,t+1] = Pf_adults
}

# break down the mating and preference probabilities
c = 1e-10 #recognition cutoff
d = sqrt(-2*sigma2*log(sqrt(2*pi*sigma2)*c)) #distance / difference at which a female can recognize a male
recognized = array(0,dim=c(Nm,1))

for(i in 1:Nm){
  x = mrange[i]
  y1 = x - d
  y2 = x + d
  w1 = which(frange>=x-d)
  w2 = which(frange<=x+d)

```

```

        recognized[i] = int(Pf_adults[intersect(w1,w2)]*dnorm(x-frange[intersect(w1,w2)],mea
    }

preferences = array(0,dim=c(Nm,Nf))
female_tots = matrix(0,Nf)

for(j in 1:Nf){
    y = frange[j]
    # weight = 1/sqrt(2*pi*sigma2)*exp(-(mrange-y)^2/(2*sigma2))
    weight = dnorm(mrange,mean=y,sd=sqrt(sigma2))
    # weight = matrix(0,Nf,1)
    # weight[c(f0,x1)] = 1
    # weight[j] = 1+alpha
    z = int(weight*Pm_adults)
    # if(z!=0){
        # pxy[,j] = Pf_adults[j]*weight*Pm_adults/z
    # }
    preferences[,j] = weight #preference given by each female to each male
    female_tots[j] = z #total preferences given by each female
}

plot(frange, female_tots, type='l', xlab = 'Preferences',ylab = 'Available mates')

```

## References

- [1] D.E. Irwin. Culture in songbirds and its contribution toward the evolution of new species. In E. Slingerland and M. Collard, editors, *Creating Consilience: Integrating the Sciences and the Humanities*, pages 163–178. Oxford University Press, 2012.
- [2] R. Lachlan and M. Servedio. Song learning accelerates allopatric speciation. *Evolution*, 58(9):2049–2063, 2004.

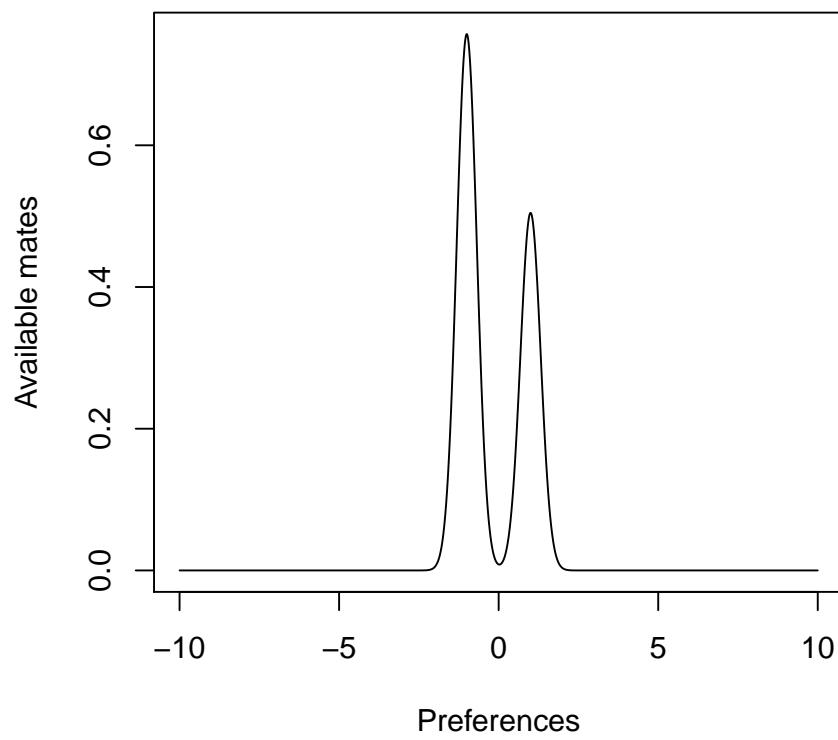


Figure 2: Available mates as a function of female preference.