

Contemporary Evolution Estimates of Selection Loberg Lake Time Series Data – EDA locus

We used empirical estimates of changes in allele frequency over time to estimate EDA locus selection s and dominance h from Loberg Lake contemporary evolution time series data. Below we describe the methods used for modeling from two different perspectives - modeling both selection *favoring* the beneficial allele and selection *against* the deleterious allele. These approaches represent two sides of the same coin and both are used throughout the literature, but importantly the s coefficient for selection against the marine allele is, by nature of the equation bound by 0 and 1, whereas when modeling selection favoring the freshwater allele, s may reach values > 1 . Finally, in addition to non-overlapping generations, we also estimated s and h allowing for overlapping generations (with an estimated mixed contribution from 1- and 2-year old fish).

Modeling the purging of a deleterious allele - *Selection Against Marine Allele* Calculation of selection and dominance coefficients of the marine allele

Selection and dominance coefficients for the EDA marine allele were estimated by modeling the empirically observed allele frequency in the Loberg Lake time series. This was done using maximum likelihood estimates of the parameters s and h in the standard recursive population genetic formula modeling selection against the deleterious q (marine) allele:

$$p' = \frac{p^2 + \frac{2pq(1 - hs)}{2}}{p^2 + 2pq(1 - hs) + q^2(1 - s)}$$

where p' is the allele frequency of the freshwater allele in the next generation, s is the selection coefficient, and h is the dominance coefficient (Falconer & Mackay, 1996). Under this model the baseline fitness of individuals homozygous for the freshwater genotype is set to 1, while the relative fitness of individuals homozygous for the marine allele is reduced by the selection coefficient s , and relative fitness of heterozygotes is reduced by the selection coefficient s multiplied by the dominance of the q allele h . From this we can calculate the fitness of the genotypes as follows:

$$\begin{aligned}\omega_{AA(\text{freshwater homozygotes})} &= 1 \\ \omega_{AB(\text{heterozygotes})} &= 1 - hs \\ \omega_{BB(\text{marine homozygotes})} &= 1 - s\end{aligned}$$

Under this model s is bounded by 0 and 1 ($0 < s < 1$). Strong selection of $s = 1$, $h \geq 1$ would completely purge the deleterious allele from the population within one generation.

Modeling of the change in allele frequency over time requires a third parameter estimate, representing the allele frequency of the freshwater allele at the start of the time series (t_0 , or in this case p_{1983}). We performed two different models firstly estimating s , h and p_{1983} based on a generation time of 1 year with parameter s bound by 0 and 1 and an unbounded dominance coefficient h that is free to vary above 1 and below 0 (allowing for over- and under-dominance respectively), and secondly, a model that allows for both one and two year old individuals to contribute to the breeding population (where the t_1 parameter represents their proportion of alleles from 1-year old fish and $1 - t_1$ the proportion of alleles from 2-year old fish). Estimates for all parameters were obtained using optimization routines (optim package in R) to find the maximum likelihood estimates of the function. The maximum likelihood function assumed the observed allele frequencies were sampled from a binomial distribution around the predicted frequency. Standard errors were calculated from the Fisher Information matrix.

Modeling the spread of a beneficial allele - Selection for Freshwater Allele

Calculation of selection and dominance coefficients of the freshwater allele

Selection and dominance coefficients for the EDA freshwater allele were estimated by modeling the empirically observed allele frequency in the Loberg Lake time series. This was done using maximum likelihood estimates of the parameters s and h in the standard recursive population genetic formula modeling selection acting on the beneficial p (freshwater) allele:

$$p' = \frac{p^2(1+s) + \frac{2pq(1+hs)}{2}}{p^2(1+s) + 2pq(1+hs) + q^2}$$

where p' is the allele frequency of the freshwater allele in the next generation, s is the selection coefficient, and h is the dominance coefficient (Maynard-Smith, 1998). Under this model the baseline fitness of individuals homozygous for the ancestral marine genotype is set to 1, while the relative fitness of individuals homozygous for the derived freshwater allele is increased by the selection coefficient s , and the relative fitness of heterozygotes increased by the selection coefficient s multiplied by the dominance of the p allele h .

$$\begin{aligned}\omega_{AA(\text{freshwater homozygotes})} &= 1 + s \\ \omega_{AB(\text{heterozygotes})} &= 1 + hs \\ \omega_{BB(\text{marine homozygotes})} &= 1\end{aligned}$$

Under this model, s is unbounded. Strong selection $s \gg 1$ and $h \geq 1$ would result in complete fixation of the beneficial freshwater allele within one generation.

Modeling of the change in allele frequency over time requires a third parameter estimate, representing the allele frequency of the freshwater allele at the start of the time series (t_0 , or in this case p_{1983}). We performed two different models firstly estimating s , h and p_{1983} based on a generation time of 1 year, and secondly a model that allows for both one and two year old individuals to contribute to the breeding population (where the t_1 parameter represents their proportion of alleles from 1-year old fish and $1 - t_1$ the proportion of alleles from 2-year old fish). Estimates for all parameters were obtained using optimization routines (optim package in R) to find the maximum likelihood estimates of the function. The maximum likelihood function assumed the observed allele frequencies were sampled from a binomial distribution around the predicted frequency. Standard errors were calculated from the Fisher Information matrix.

Table 1. Estimates of selection and dominance at the EDA locus from Loberg Lake contemporary evolution time series. Models 1 and 2 estimate parameters s and h modeling selection against the marine allele, while models 3 and 4 estimate parameters s and h modeling selection for the freshwater allele. Models 1 and 3 assume a 1-year generation time, while models 2 and 4 allow both 1- and 2-year old fish to breed and contribute to the next generation. In these latter models, the estimated proportion of 1-year old breeders is shown as parameter t_1 .

Model	Type [Selection Against Deleterious q / Selection for Beneficial p]	Generation Time	s parameter bounds	h parameter bounds	EDA									
					s	h_{MARINE}	h_{FRESH}	p_{1983}	t_1	standard error s	standard error h	standard error p_{1983}	standard error t_1	-log likelihood
1	Selection Against q	1	0-1	unbounded	0.490	0.229	na	0.010	na	0.048	0.065	0.010	na	457.094
2**	Selection Against q	1&2	0-1	unbounded	0.503	0.218	na	0.008	0.999	0.005	0.011	0.006	0.009	457.070
3	Selection For p	1	unbounded	unbounded	1.048	na	0.794	0.006	na	0.131	0.048	0.004	na	457.061
4**	Selection For p	1&2	unbounded	unbounded	1.278	na	0.751	0.008	0.788	0.024	0.010	0.005	0.012	457.181

** standard errors of parameter estimates calculated by bootstrapping allele counts 500 times

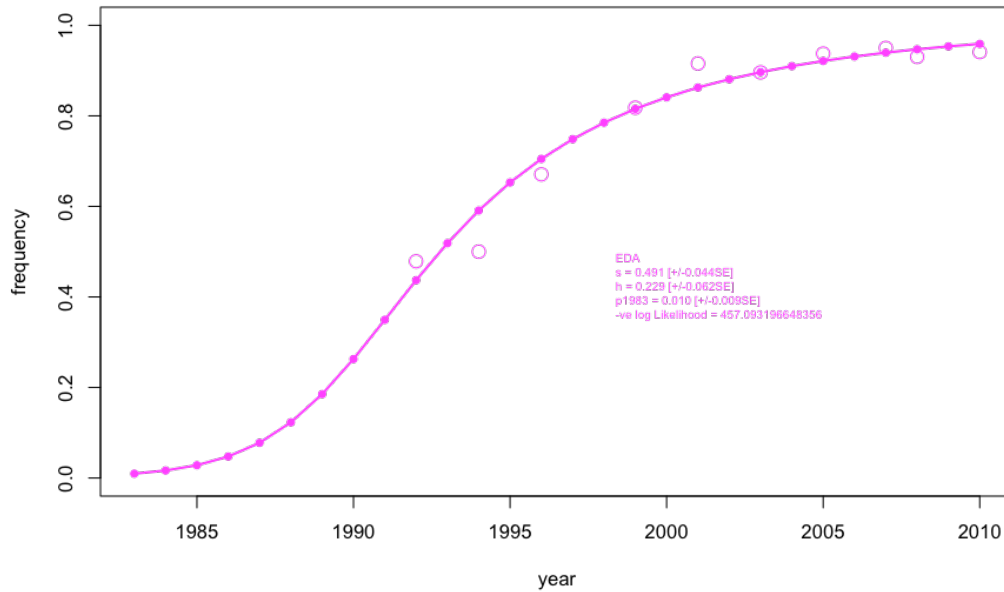


Figure 1. Observed (open circles) and estimated (solid points) freshwater EDA allele frequencies over time in Loberg Lake using model 1 modeling selection against the marine allele with 1-year generation time, $0 < s < 1$. Parameter estimates, standard errors and model likelihood are shown in the plot.

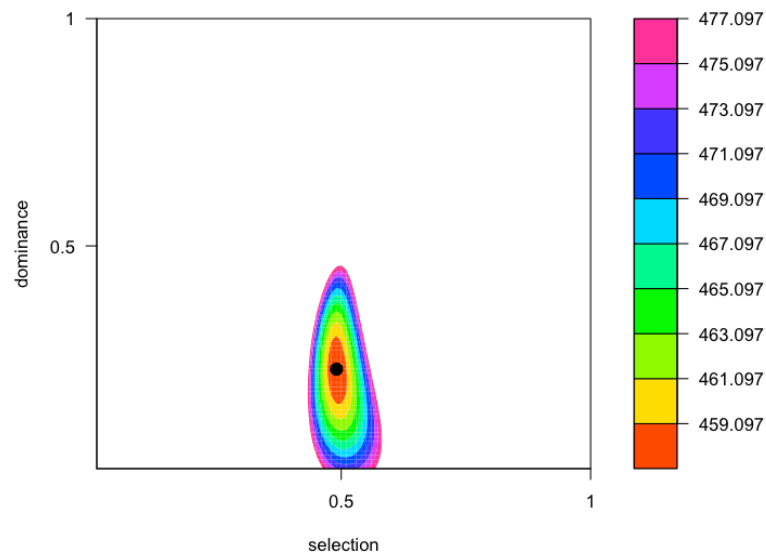


Figure 2 Likelihood surface of s and h parameter estimates under model 1 for selection against the marine EDA allele with a 1-year generation time in the Loberg Lake contemporary evolution time series. Contours are coloured according to the negative log likelihood with steps of unit 2. The boundaries of the red contour are approximately equivalent to $p=0.05$). Estimates from the best model are shown as black point.

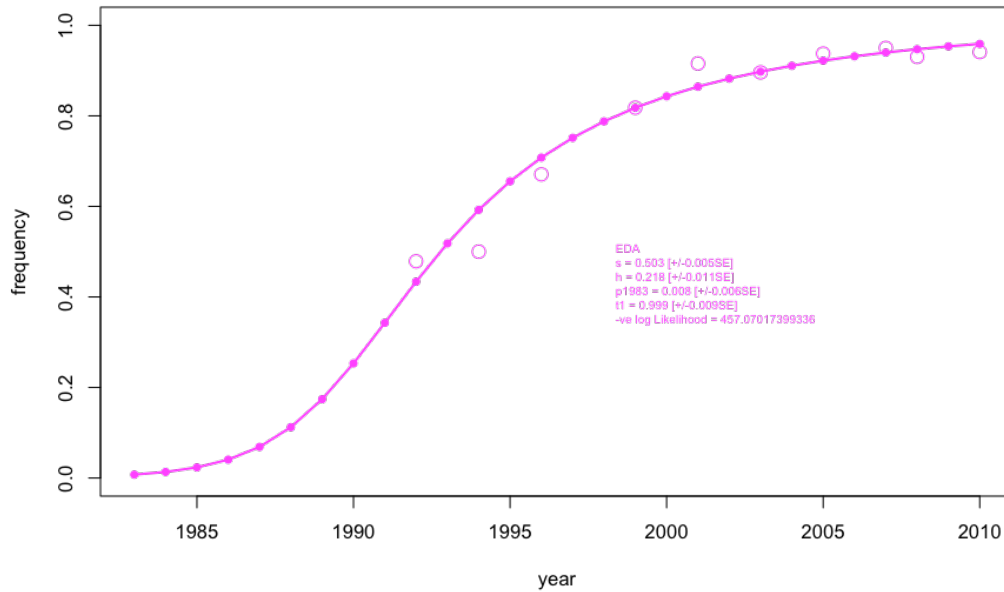


Figure 3. Observed (open circles) and estimated (solid points) freshwater EDA allele frequencies over time in Loberg Lake using model 2 modeling selection against the marine allele with 1- and 2-year old breeders contributing to subsequent generations, $0 < s < 1$. Parameter estimates, standard errors and model likelihood are shown in the plot.

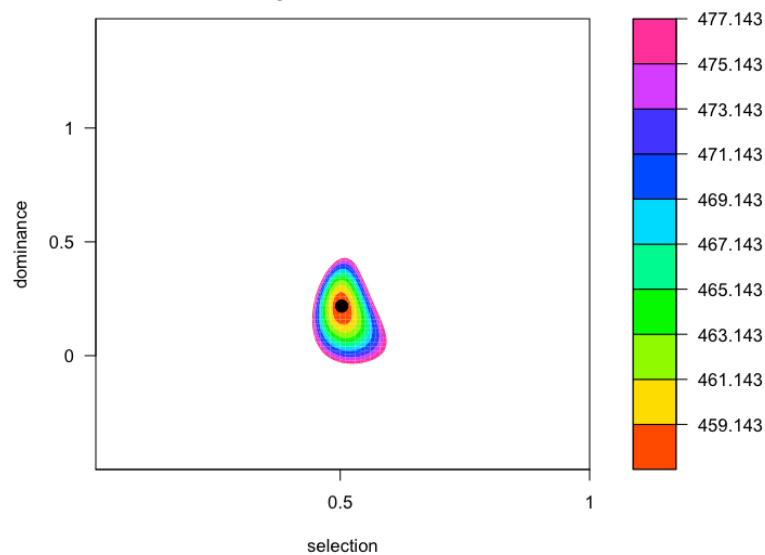


Figure 4 Likelihood surface of s and h parameter estimates under model 2 for selection against the marine EDA allele in the Loberg Lake time series with 1- and 2-year old breeders contributing to subsequent generations, $0 < s < 1$. Contours are coloured according to the negative log likelihood with steps of unit 2. The boundaries of the red contour are approximately equivalent to $p=0.05$). Estimates from the best model are shown as black point.

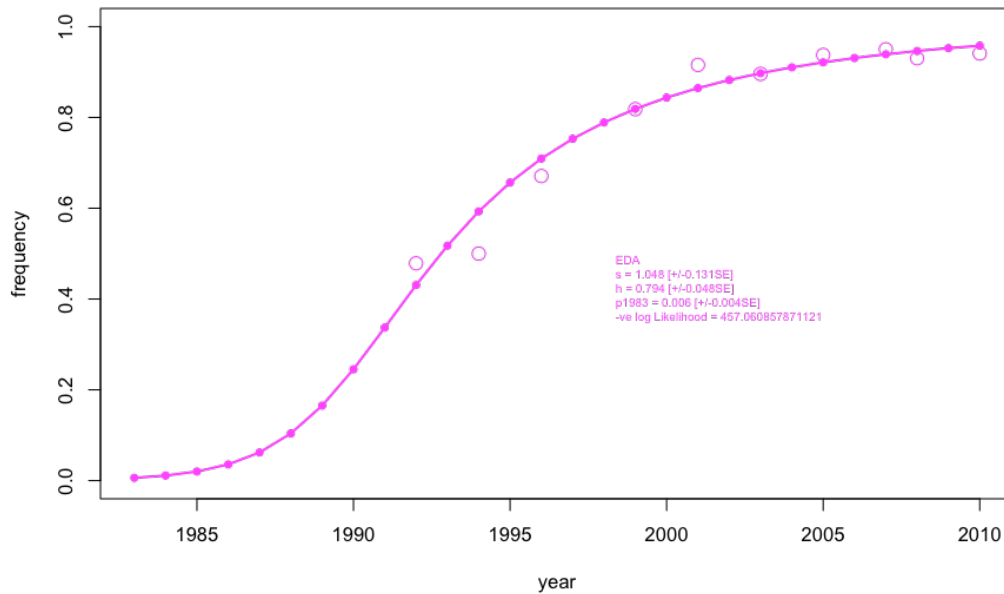


Figure 5. Observed (open circles) and estimated (solid points) freshwater EDA allele frequencies over time in Loberg Lake using model 3 modeling selection favoring the freshwater allele with 1-year generation time, $0 < s < 1$. Parameter estimates, standard errors and model likelihood are shown in the plot.

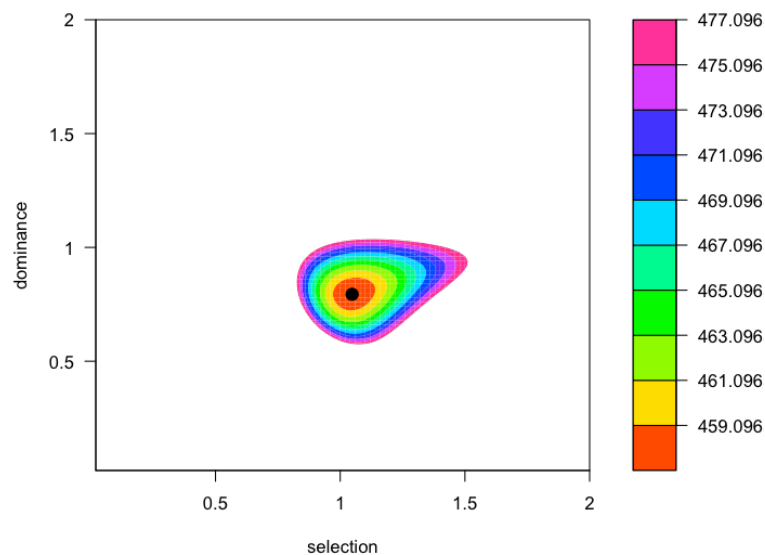


Figure 6 Likelihood surface of s and h parameter estimates under model 3 for selection favoring the freshwater EDA allele in the Loberg Lake time series with 1-year old breeders contributing to subsequent generations, $0 < s < 1$. Contours are coloured according to the negative log likelihood with steps of unit 2. The boundaries of the red contour are approximately equivalent to $p=0.05$). Estimates from the best model are shown as black point.

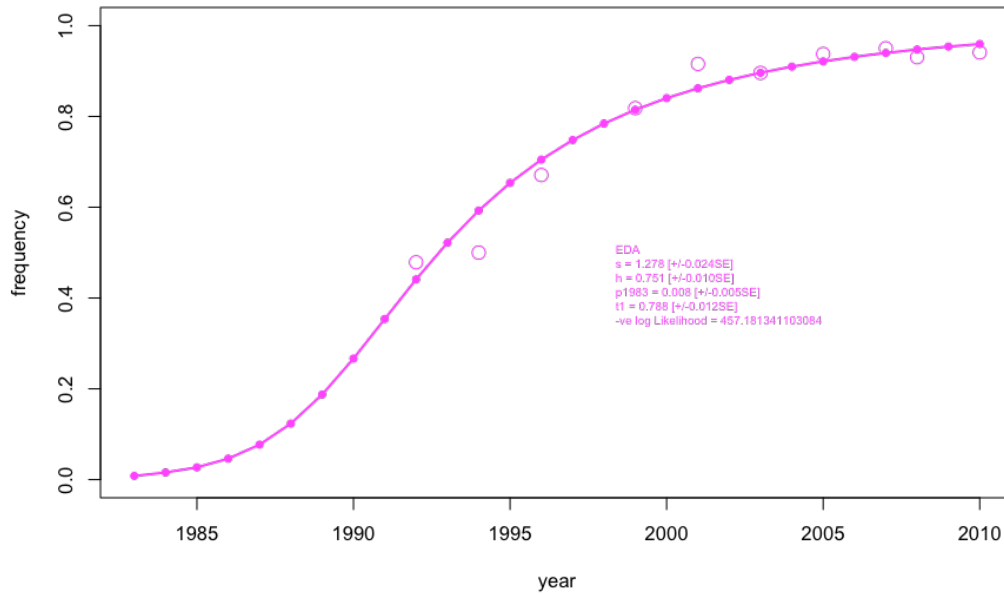


Figure 7. Observed (open circles) and estimated (solid points) freshwater EDA allele frequencies over time in Loberg Lake using model 4 modeling selection favoring the freshwater allele with 1- and 2-year old fish breeding and contributing to the subsequent generations, $0 < s < 1$. Parameter estimates, standard errors and model likelihood are shown in the plot.

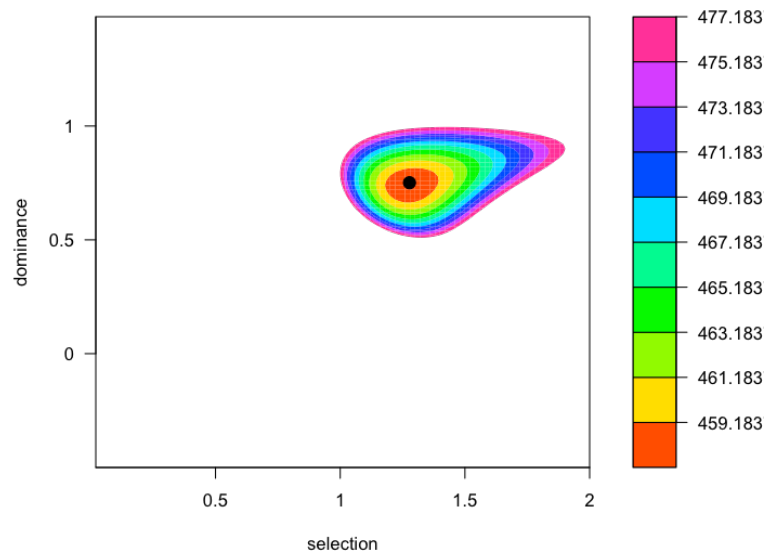


Figure 8 Likelihood surface of s and h parameter estimates under model 4 for selection favoring the freshwater EDA allele in the Loberg Lake time series with 1-year old breeders contributing to subsequent generations, $0 < s < 1$. Contours are coloured according to the negative log likelihood with steps of unit 2. The boundaries of the red contour are approximately equivalent to $p=0.05$). Estimates from the best model are shown as black point.

R Code for contemporary evolution estimates of selection from Loberg Lake Time Series

```
# raw genotype counts
year  EDAFF  EDAFM  EDAMM  EDAFcount  EDATotcount
1984  NA     NA     NA     NA         NA
1985  NA     NA     NA     NA         NA
1986  NA     NA     NA     NA         NA
1987  NA     NA     NA     NA         NA
1988  NA     NA     NA     NA         NA
1989  NA     NA     NA     NA         NA
1990  NA     NA     NA     NA         NA
1991  NA     NA     NA     NA         NA
1992  8      29     10     45         94
1993  NA     NA     NA     NA         NA
1994  14     12     14     40         80
1995  NA     NA     NA     NA         NA
1996  18     19     4      55         82
1997  NA     NA     NA     NA         NA
1998  NA     NA     NA     NA         NA
1999  35     20     0      90         110
2000  NA     NA     NA     NA         NA
2001  70     12     1      152        166
2002  NA     NA     NA     NA         NA
2003  40     6      2      86         96
2004  NA     NA     NA     NA         NA
2005  77     11     0      165        176
2006  NA     NA     NA     NA         NA
2007  54     6      0      114        120
2008  68     11     0      147        158
2009  NA     NA     NA     NA         NA
2010  82     11     0      175        186

data<-read.table(file="LobergContemporaryEvolutionDataSet.txt",header=T,sep="\t",na.string="NA")
```

```
# Binomial Likelihood function
binom.lik<-function(k,N,p) {logl <- sum(k*log(p)+(N-k)*log(1-p)) return(-logl)}

# EDA Model 1 function – selection against deleterious allele q (marine allele), one-year generation
time, 0<s<1,
estimate_selection_against_m1 <- function(parmselagm1) {

  p <- vector(mode="numeric",length=length(data[,1])+1)
  q <- vector(mode="numeric",length=length(data[,1])+1)
  p[1] <- parmselagm1[3]
  q[1] <- 1-p[1]
  sumloglik <- 0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency
      p[t] = ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
        (parmselagm1[2]*parmselagm1[1])))/2) / ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
        (parmselagm1[2]*parmselagm1[1])))) + (q[t-1]*q[t-1])*(1-parmselagm1[1]))
      q[t] <- 1-p[t]
    } else {
      #for all other subsequent years
      #Model 1: 1-year generation time - the frequency is dependent on frequency from
      one year prior.
      pbreeders <- p[t-1]
      qbreeders <- (1-pbreeders)
      p[t] <- ( (pbreeders*pbreeders) + ((2*pbreeders*qbreeders)*(1-
        (parmselagm1[2]*parmselagm1[1])))/2 ) / ( (pbreeders*pbreeders) + (
        (2*pbreeders*qbreeders)*(1-(parmselagm1[2]*parmselagm1[1])) ) +
        (qbreeders*qbreeders)*(1-parmselagm1[1]))
      q[t] <- 1-p[t]
    }
    # check if we have observed data in this year. In years we have observed data,
    evaluate the fit of the predicted frequency p(t) to the observed data.
    if (is.na(data[t-1,datapcounts])==F) {
      # we have data.
      # the binomial likelihood function can't log0... so make sure p[t] never quite
      reaches fixation.
      if(p[t]==1) {
        p[t] = p[t]-0.0000001
      }
    }
  }
}
```

```

    }
    myloglik <- binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
    sumloglik <- myloglik+sumloglik
  }
}
return(sumloglik)
}

#priors (s, h, p1983)
parmselagm1 <- c(0.2, 0.5, 0.035)
# parameter bounds s, h, p1983
mllowerbounds <- c(0.01, -1.5, 0.001)
mlupperbounds <- c(1, 2.5, 0.999)

#initial parameters
datapcounts<-5
datatotals<-6

# now run model to estimate h and s
edam1<-optim(parmselagm1, fn=estimate_selection_against_m1, hessian=T, lower=mllowerbounds,
upper=mlupperbounds, method="L-BFGS-B")

#estimate standard errors
edaOI_m1<-solve(edam1$hessian)
se_edam1<-sqrt(diag(edaOI_m1))

edam1$par
[1] 0.490473001 0.228562735 0.009719429
edam1$value
[1] 457.094
se_edam1
[1] 0.047672960 0.065022643 0.009520792

# model 1 : plot the results on the time series
# we need to return estimated p for best fit model 1
# so, we modify the function to return p instead of sumloglik

estimate_selection_against_m1_returnp <- function(parmselagm1) {
  p <- vector(mode="numeric",length=length(data[,1])+1)
  q <- vector(mode="numeric",length=length(data[,1])+1)
  p[1] <- parmselagm1[3]
  q[1] <- 1-p[1]
  sumloglik <- 0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency
      p[t] = ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
        (parmselagm1[2]*parmselagm1[1])))/2) / ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
        (parmselagm1[2]*parmselagm1[1])))) + (q[t-1]*q[t-1])*(1-parmselagm1[1]))
      q[t] <- 1-p[t]
    } else {
      #for all other subsequent years
      #Model 1: 1-year generation time - the frequency is dependent on frequency from
      one year prior.
      pbreeders <- p[t-1]
      qbreeders <- (1-pbreeders)
      p[t] <- ( (pbreeders*pbreeders) + ((2*pbreeders*qbreeders)*(1-
        (parmselagm1[2]*parmselagm1[1])))/2 ) / ( (pbreeders*pbreeders) + (
        (2*pbreeders*qbreeders)*(1-(parmselagm1[2]*parmselagm1[1])) ) +
        (qbreeders*qbreeders)*(1-parmselagm1[1]))
      q[t] <- 1-p[t]
    }
    # check if we have observed data in this year. In years we have observed data,
    evaluate the fit of the predicted frequency p(t) to the observed data.
    if (is.na(data[t-1,datapcounts])==F) {
      # we have data.
      # the binomial likelihood function can't log0... so make sure p[t] never quite
      reaches fixation.
      if(p[t]==1) {
        p[t] = p[t]-0.0000001
      }
      myloglik <- binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
      sumloglik <- myloglik+sumloglik
    }
  }
  return(p)
}

```



```

# now run this function to return p for best fit parameters of model
eda_estimatedpml<-estimate_selection_against_ml_returnp(edaml$par)

# model 1: plot the EDA times series with best fit model

pdf("Figure1_Model1_timeseries.pdf",width=9,height=6)
plot(c(1983,data[,1]), eda_estimatedpml, type="n", xlab="year", ylab="frequency", ylim=c(0,1),
main="EDA (pink) selection against q model 1 1983 onwards, 0<s<1", cex.main=0.6)
points(c(1983,data[,1]), eda_estimatedpml, col="magenta", pch=19, cex=0.8)
lines(c(1983,data[,1]), eda_estimatedpml, col="magenta", pch=19, lwd=2)
# observed data as open circles
points(data[,1], data[,5]/data[,6], pch=21, col="magenta", cex=1.5)
text(1998, 0.4, labels=sprintf("EDA\\ns = %.3f [+/-%.3fSE]\\nh = %.3f [+/-%.3fSE]\\np1983 = %.3f [+/-
%.3fSE]\\n-ve log Likelihood =
%.3f",edaml$par[1],se_edaml[1],edaml$par[2],se_edaml[2],edaml$par[3],se_edaml[3],edaml$value), cex=0.6,
pos=4, col="magenta")
dev.off()

# model 1: plot the s and h parameter likelihood surfaces
# to do this we need to calculate likelihood of models for all values of s and h ranging from 0 to 1 in
steps of 100. We fix the starting allele frequency in the first generation to be that of the best fit
model, and use this in all models.

# First, we need to use a function with p1983 set to value from best model.
estimate_selection_against_ml_fixedp<-function(parmselagml) {
  p <- vector(mode="numeric",length=length(data[,1])+1)
  q <- vector(mode="numeric",length=length(data[,1])+1)
  # set starting allele frequency to fixedp. (best estimate from fitted model)
  p[1] <- fixedp
  q[1] <- 1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t] = ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
      (parmselagml[2]*parmselagml[1])))/2) / ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
      (parmselagml[2]*parmselagml[1])))) + (q[t-1]*q[t-1])*(1-parmselagml[1]))
      q[t] <- 1-p[t]
    } else {
      #for all other subsequent years
      #Model 1: 1 year generation time - the frequency is dependent on frequency from
      one year prior.
      pbreeders<-p[t-1]
      qbreeders<-(1-pbreeders)
      p[t] <- ( (pbreeders*pbreeders) + ((2*pbreeders*qbreeders)*(1-
      (parmselagml[2]*parmselagml[1])))/2) / ( (pbreeders*pbreeders) + (
      (2*pbreeders*qbreeders)*(1-(parmselagml[2]*parmselagml[1])) ) +
      (qbreeders*qbreeders)*(1-parmselagml[1]))
      q[t] <- 1-p[t]
    }
    # check if we have observed data in this year
    # in years we have observed data, evaluate the fit of the predicted frequency pt to the
    observed data.

    if (is.na(data[t-1,datapcounts])==F) {
      # we have data
      # the binomial likelihood function can't log0... so make sure p[t] never quite
      reaches fixation.
      if(p[t]==1) {
        p[t] = p[t]-0.0000001
      }
      myloglik <- binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
      sumloglik <- myloglik+sumloglik
    }
  }
  return(sumloglik)
}

# now run with EDA fixed p. We calculate likelihood of model for all values of s and h ranging from 0
to 1 in steps of 100

datapcounts <- 5
datatotals <- 6
fixedp <- edaml$par[3]

```

```

myhmin <- (-0.5)
myhmax <- 1.5
mysmax <- 1
points2 <- 100
hcounter <- 0
allspacem4 <- matrix(nrow=points2, ncol=points2)
for (hh2 in seq(myhmin, myhmax, by = ((myhmax-myhmin)/points2))) {
  hparm2 <- hh2
  hcounter <- hcounter+1
  for (ss2 in seq(1, points2, by = 1)) {
    sparm2 <- mymax*(ss2/points2)
    parms2 <- c(sparm2, hparm2)
    mylikelihood2 <- estimate_selection_against_m1_fixedp(parms2)
    allspacem4[ss2,hcounter] <- mylikelihood2
  }
}
x <- seq(0+(mymax/points2), mymax, by = mymax/points2)
y <- seq(-0.5, 1.48, by=2/points2)
colnum <- 10

#now plot likelihood surface
pdf("Figure2_Model1_likelihood.pdf", width = 7, height = 6)
filled.contour(x, y, allspacem4, plot.axes = c(points(x = edaml$par[1], y = edaml$par[2], cex=1.5, pch
= 19), axis(side = 1, labels = seq(0, mymax, by = 0.5), at = seq(0, mymax, by = 0.5)), axis(side = 2,
labels = seq(0, mymax, by=0.5), at = seq(0, mymax, by = 0.5))), col = c(rainbow(colnum, start = 0.05,
end=0.90)), levels = c(seq(min(allspacem4, na.rm = T), (min(allspacem4, na.rm = T) + (2*(colnum))), by
= 2)), key.axes = axis(4, seq(min(allspacem4, na.rm = T) + 2, min(allspacem4, na.rm = T) +
2*(colnum+1), by = 2)), xlab = "selection", ylab = "dominance", main = sprintf("EDA selection Against q
modell 1 year generation time 0 < s < 1\ns = %.3f[+/-SE:%.3f], h = %.3f[+/-SE:%.3f], p1983 = %.3f[+/-
SE:%.3f],\n -ve log Likelihood = %s", edaml$par[1], se_edaml[1], edaml$par[2], se_edaml[2],
edaml$par[3], se_edaml[3], edaml$value), cex.main = 0.6)
dev.off()

```

model 2: selection against q (marine allele) with mixed breeding contribution from 1 and 2 year old fish

```

#set priors (s, h, p1983, t1) - where t1=proportion of 1 year breeders and 1-t1=proportion of 2-year
breeders.
parmselagm2<-c(0.2,0.5,0.035,0.8)

# parameter bounds s,h,p1983, t1
m2lowerbounds<-c(0.01, -0.5, 0.001,0.001)
m2upperbounds<-c(1, 1.5, 0.999,0.999)

estimate_selection_against_m2<-function(parmselagm2) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-parmselagm2[3]
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t]=((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
      (parmselagm2[2]*parmselagm2[1])))/2) / ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
      (parmselagm2[2]*parmselagm2[1])))) + (q[t-1]*q[t-1])*(1-parmselagm2[1]))
      q[t]<-1-p[t]
    } else {
      #for all other subsequent years
      #Model 2: breeders come from previous 2 years with proportion t1
      pbreeders<-parmselagm2[4]*p[t-1]+(1-parmselagm2[4])*p[t-2]
      qbreeders<-(1-pbreeders)
      p[t]<-((pbreeders*pbreeders) + ((2*pbreeders*qbreeders)*(1-
      (parmselagm2[2]*parmselagm2[1])))/2) / ((pbreeders*pbreeders) + (
      (2*pbreeders*qbreeders)*(1-(parmselagm2[2]*parmselagm2[1])))) +
      (qbreeders*qbreeders)*(1-parmselagm2[1]))
      q[t]<-1-p[t]
    }
  }
  # check if we have observed data in this year
  # in years we have observed data, evaluate the fit of the predicted frequency pt to the
  observed data.
  if (is.na(data[t-1,datapoints])==F) {
    # we have data
    # the binomial likelihood function can't log0... so make sure p[t] never quite
    reaches fixation.
  }
}

```

```

        if(p[t]==1) {
            p[t]=p[t]-0.0000001
        }
        myloglik<-binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
        sumloglik<-myloglik+sumloglik
    }
    }
    return(sumloglik)
}

#run model2 for EDA
datapcounts<-5
datatotals<-6
edam2<-optim(parmselagm2,fn=estimate_selection_against_m2,hessian=T,lower=m2lowerbounds,
upper=m2upperbounds,method="L-BFGS-B")
#get standard errors
edaOI_m2<-solve(edam2$hessian)
se_edam2<-sqrt(diag(edaOI_m2))

# parameter estimates
edam2$par
[1] 0.503078396 0.217804037 0.007524298 0.999000000
# -loglikelihood
edam2$value
[1] 457.0702
# standard errors
# nb hessian matrix cannot be solved so need to bootstrap to get standard errors or 95% CI
se_edam2
[1] NaN NaN NaN NaN

#so bootstrap to get standard errors
data2<-data
m2lowerbounds<-c(0.01, -0.5, 0.001,0.001)
m2upperbounds<-c(1, 1.5, 0.999,0.999)
datapcounts<-5
datatotals<-6
myrepetitions<-1000
bootsample<-data.frame(matrix(NA,nrow=length(data[,1]),ncol=myrepetitions))
for (j in 1:myrepetitions) {for (i in 1:length(data[,1])) {if(is.na(data[i,datatotals])==T)
{bootsample[i,j]<-NA} else {bootsample[i,j]<-
sum(sample(c(rep(1,data[i,datatotals]),rep(0,data[i,datatotals]-
data[i,datapcounts])),data[i,datatotals],replace=T))}}}

bootPars<-data.frame(matrix(NA,nrow=myrepetitions,ncol=length(parmselagm2)+1))

for (k in 1:myrepetitions) {
    print(k)
    data[,datapcounts] <- bootsample[,k]
    #parmselagm2 <- c(0.2,0.5,0.035,0.5)
    m2lowerbounds <- c(0.01, -0.5, 0.001,0.001)
    m2upperbounds <- c(1, 1.5, 0.999,0.999)
    tmp<-optim(parmselagm2, fn=estimate_selection_against_m2, hessian=T, lower=m2lowerbounds,
upper=m2upperbounds, method="L-BFGS-B")
    bootPars[k,] <- c(tmp$par,tmp$value)
}
data<-data2
edabootm2Pars <- bootPars[which(is.na(bootPars[,1])==F),]
edabootm2Pars <- edabootm2Pars[1:500,]
colnames(edabootm2Pars) <- c("s","h","p1983","tlcontribution","Negloglikelihood")

myresults <- data.frame(matrix(NA,nrow=5,ncol=length(parmselagm2)+1))
myresults[1,] <- sapply(edabootm2Pars, mean, na.rm = TRUE)
myresults[2,] <- sapply(edabootm2Pars, function(x) {sd(x[which(is.na(x))==F])})
myresults[3,] <- sapply(edabootm2Pars, function(x)
{sd(x[which(is.na(x))==F])/sqrt(length(x[which(is.na(x))==F])})})
myresults[4,] <- sapply(edabootm2Pars, function(x) {quantile(x,c(0.025),na.rm=T)})
myresults[5,] <- sapply(edabootm2Pars, function(x) {quantile(x,c(0.975),na.rm=T)})
rownames(myresults) <- c("mean","sd","se","lower2.5CI","upper97.5CI")
colnames(myresults) <- c("s","h","p1983","tlcontribution","Negloglikelihood")
edabootm2Results <- myresults

edabootm2Results

```

	s	h	p1983	tlcontribution	Negloglikelihood
mean	0.527210461	0.28681478	0.099096925	0.70437865	420.7326862
sd	0.122340604	0.24908288	0.124533129	0.20856858	21.6886409
se	0.005471238	0.01113933	0.005569291	0.00932747	0.9699455

```

# plot model 2 timeseries and likelihood surface.

```

```

# first get estimated p for best model
# modify function to return p instead of sumloglikelihood.

estimate_selection_against_m2_returnp<-function(parmselagm2) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-parmselagm2[3]
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t] = ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1]*(1-
        (parmselagm2[2]*parmselagm2[1]))/2)) / ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1]*(1-
        (parmselagm2[2]*parmselagm2[1])) + (q[t-1]*q[t-1]*(1-parmselagm2[1]))
        q[t]<-1-p[t]
    } else {
      #for all other subsequent years
      #Model 2: breeders come from previous 2 years with proportion t1(1yr) and 1-
      t1(2yr)
      pbreeders<-parmselagm2[4]*p[t-1]+(1-parmselagm2[4])*p[t-2]
      qbreeders<-(1-pbreeders)
      p[t]<-((pbreeders*pbreeders) + ((2*pbreeders*qbreeders)*(1-
        (parmselagm2[2]*parmselagm2[1]))/2) / ((pbreeders*pbreeders) + (
        (2*pbreeders*qbreeders)*(1-(parmselagm2[2]*parmselagm2[1])) ) +
        (qbreeders*qbreeders)*(1-parmselagm2[1]))
      q[t]<-1-p[t]
    }
    # check if we have observed data in this year
    # in years we have observed data, evaluate the fit of the predicted frequency pt to the
    observed data.
    if (is.na(data[t-1,datapoints])==F) {
      # we have data
      # the binomial likelihood function can't log0... so make sure p[t] never quite
      reaches fixation.
      if(p[t]==1) {
        p[t]=p[t]-0.0000001
      }
      myloglik<-binom.lik(data[t-1,datapoints],data[t-1,datatotals],p[t])
      sumloglik<-myloglik+sumloglik
    }
  }
  return(p)
}

# get values of p for best model2 of eda
datapoints<-5
datatotals<-6
eda_estimatedpm2<-estimate_selection_against_m2_returnp(eda2$par)

# plot model 2 time series
pdf("Figure3_Model2_timeseries.pdf",width=9,height=6)
plot(c(1983,data[,1]),eda_estimatedpm2,type="n",xlab="year",ylab="frequency",ylim=c(0,1),main="EDA
(pink) selection against q (marine allele), model 2\n 0 < s < 1, 1&2 year old breeders",cex.main=0.6)
points(c(1983,data[,1]),eda_estimatedpm2,col="magenta",pch=19,cex=0.8)
lines(c(1983,data[,1]),eda_estimatedpm2,col="magenta",pch=19,lwd=2)
points(data[,1],data[,5]/data[,6],pch=21,col="magenta",cex=1.5)
text(1998,0.4,labels=sprintf("EDA\ns = %.3f [+/-%.3fSE]\nh = %.3f [+/-%.3fSE]\np1983 = %.3f [+/-
%.3fSE]\nt1 = %.3f [+/-%.3fSE]\n-ve log Likelihood =
%s",eda2$par[1],edabootm2Results[3,1],eda2$par[2],edabootm2Results[3,2],eda2$par[3],edabootm2Results
[3,3],eda2$par[4],edabootm2Results[3,4],eda2$value),cex=0.6,pos=4,col="magenta")
dev.off()

# now create the likelihood surfaces for fixed values of p and t1
# modify the function to take fixed starting p in 1983, and fixed t1:
estimate_selection_against_m2_fixedp<-function(parmselagm2) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-fixedp
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.

```

```

p[t]=((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1]*(1-
(parmselagm2[2]*parmselagm2[1]))/2)) / ((p[t-1]*p[t-1])+((2*p[t-1]*q[t-1])*(1-
(parmselagm2[2]*parmselagm2[1])) + (q[t-1]*q[t-1])*(1-parmselagm2[1]))
q[t]<-1-p[t]
} else {
  #for all other subsequent years
  #Model 2: breeders come from previous 2 years with proportion t1(1yr) and 1-
  t1(2yr). Here for the likelihood surface of s and h we keep this proportion
  fixed to estimate from best model
  #pbreeders<-p[t-1]
  pbreeders<-fixedt1*p[t-1]+(1-fixedt1)*p[t-2]
  qbreeders<-(1-pbreeders)
  p[t]<-( (pbreeders*pbreeders) + ((2*pbreeders*qbreeders)*(1-
  (parmselagm2[2]*parmselagm2[1]))/2 ) / ( (pbreeders*pbreeders) + (
  (2*pbreeders*qbreeders)*(1-(parmselagm2[2]*parmselagm2[1])) ) +
  (qbreeders*qbreeders)*(1-parmselagm2[1]))
  q[t]<-1-p[t]
}
# check if we have observed data in this year
# in years we have observed data, evaluate the fit of the predicted frequency pt to the
observed data.
if (is.na(data[t-1,datapcounts])==F) {
  # we have data
  # the binomial likelihood function can't log0. so make sure p[t] never quite
  reaches fixation.
  if(p[t]==1) {
    p[t]=p[t]-0.0000001
  }
  myloglik<-binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
  sumloglik<-myloglik+sumloglik
}
return(sumloglik)
}

# run for EDA
datapcounts<-5
datatotals<-6
# EDA fixed p
fixedp<-edam2$par[3]
fixedt1<-edam2$par[4]

#now calculate likelihood of model for all values of s and h ranging from 0 to 1 in steps of 100
myhmin<-(-0.5)
myhmax<-1.5
mysmax<-1
points2<-100
hcounter<-0
allspacem4<-matrix(nrow=points2,ncol=points2)
for (hh2 in seq(myhmin,myhmax,by=((myhmax-myhmin)/(points2-1)))) {
  hparm2<-hh2
  hcounter<-hcounter+1
  for (ss2 in seq(1,points2,by=1)) {
    sparm2<-mysmax*(ss2/points2)
    parms2<-c(sparm2,hparm2)
    mylikelihood2<-estimate_selection_against_m2_fixedp(parms2)
    allspacem4[ss2,hcounter]<-mylikelihood2
  }
}

# now plot likelihood surface
x<-seq(0+(mysmax/points2),mysmax,by=mysmax/points2)
y<-seq(-0.5,1.48,by=2/points2)
colnum<-10
pdf("Figure4_Model2_likelihood.pdf",width=7,height=6)
filled.contour(x,y,allspacem4,plot.axes=c(points(x=edam2$par[1],y=edam2$par[2],cex=1.5,pch=19),axis(side=1,labels=seq(0,mysmax,by=0.5),at=seq(0,mysmax,by=0.5)),axis(side=2,labels=seq(0,mymax,by=0.5),at=seq(0,mymax,by=0.5))),col=c(rainbow(colnum,start=0.05,end=0.90)),levels=c(seq(min(allspacem4,na.rm=T),(min(allspacem4,na.rm=T)+(2*(colnum))),by=2)),key.axes=axis(4,seq(min(allspacem4,na.rm=T)+2,min(allspacem4,na.rm=T)+2*(colnum+1),by=2)),xlab="selection",ylab="dominance",main=sprintf("EDA selection against q
model2\n 0 < s < 1, 1 and 2 year breeders\ns=%.3f[+/-SE:%.3f], h=%.3f[+/-SE:%.3f], p1983=%.3f[+/-SE:%.3f], t1=%.3f[+/-SE:%.3f]\n -ve log Likelihood =
%s",edam2$par[1],edabootm2Results[3,1],edam2$par[2],edabootm2Results[3,2],edam2$par[3],edabootm2Results[3,3],edam2$par[4],edabootm2Results[3,4],edam2$value),cex.main=0.6)
dev.off()

```

```

# Model 3 - Selection Favoring a Beneficial Allele (p, freshwater allele), 1 year generation time, s
and h unbounded
# priors (s,h,p1983)
Parmselform3<-c(0.2,0.5,0.035)

# Model 3 function
estimate_selection_for_m3<-function(parmselform3) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-parmselform3[3]
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t]=((p[t-1]*p[t-1])*(1+parmselform3[1])+((2*p[t-1]*q[t-
      1])*(1+(parmselform3[2]*parmselform3[1])))/2) / ((p[t-1]*p[t-
      1])*(1+parmselform3[1])+((2*p[t-1]*q[t-
      1])*(1+(parmselform3[2]*parmselform3[1])))) + (q[t-1]*q[t-1]))
      q[t]<-1-p[t]
    } else {
      #for all other subsequent years
      #Model 3: 1 year generation time - the frequency is dependent on frequency from
      one year prior.
      pbreeders<-p[t-1]
      qbreeders<-(1-pbreeders)
      p[t]<-((pbreeders*pbreeders)*(1+parmselform3[1]) +
      ((2*pbreeders*qbreeders)*(1+(parmselform3[2]*parmselform3[1])))/2) / (
      (pbreeders*pbreeders)*(1+parmselform3[1]) +
      ((2*pbreeders*qbreeders)*(1+(parmselform3[2]*parmselform3[1])))) +
      (qbreeders*qbreeders))
      q[t]<-1-p[t]
    }
    # check if we have observed data in this year
    # in years we have observed data, evaluate the fit of the predicted frequency p(t) to
    the observed data.
    if (is.na(data[t-1,datapcounts])==F) {
      # we have data
      # the binomial likelihood function can't log0... so make sure p[t] never quite
      reaches fixation.
      if(p[t]==1) {
        p[t]=p[t]-0.0000001
      }
      myloglik<-binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
      sumloglik<-myloglik+sumloglik
    }
  }
  return(sumloglik)
}

# Apply model 3
# priors (s,h,p1983)
parmselform1<-c(0.2,0.5,0.035)
datapcounts<-5
datatotals<-6

edam3<-optim(parmselform3,fn=estimate_selection_for_m3,hessian=T)
edaOI_m3<-solve(edam3$hessian)
se_edam3<-sqrt(diag(edaOI_m3))

edam3$par
1.047649196 0.793908384 0.006143567
Edam3$value
457.0609
se_edam3
0.13139984 0.04808353 0.00397080

# prepare to plot model 3 timeseries
# we need to modify the function to return estimated p from best parameters.
# Model 3 function to. return p
estimate_selection_for_m3_returnp<-function(parmselform3) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-parmselform3[3]
  q[1]<-1-p[1]
  sumloglik<-0

```

```

for (t in 2:(length(data[,1])+1)) {
  if (t==2) {
    #the frequency in the first year after founding is dependent solely 1983
    founding frequency.
    p[t]=((p[t-1]*p[t-1])*(1+parmselform3[1])+((2*p[t-1]*q[t-
    1])*(1+(parmselform3[2]*parmselform3[1])))/2) / ((p[t-1]*p[t-
    1])*(1+parmselform3[1])+((2*p[t-1]*q[t-
    1])*(1+(parmselform3[2]*parmselform3[1])))) + (q[t-1]*q[t-1]))
    q[t]<-1-p[t]
  } else {
    #for all other subsequent years
    #Model 3: 1 year generation time - the frequency is dependent on frequency from
    one year prior.
    pbreeders<-p[t-1]
    qbreeders<-(1-pbreeders)
    p[t]<-((pbreeders*pbreeders)*(1+parmselform3[1]) +
    ((2*pbreeders*qbreeders)*(1+(parmselform3[2]*parmselform3[1])))/2) / (
    (pbreeders*pbreeders)*(1+parmselform3[1]) +
    ((2*pbreeders*qbreeders)*(1+(parmselform3[2]*parmselform3[1])))) +
    (qbreeders*qbreeders))
    q[t]<-1-p[t]
  }
  # check if we have observed data in this year
  # in years we have observed data, evaluate the fit of the predicted frequency p(t) to
  the observed data.
  if (is.na(data[t-1,datapcounts])==F) {
    # we have data
    # the binomial likelihood function can't log0... so make sure p[t] never quite
    reaches fixation.
    if(p[t]==1) {
      p[t]=p[t]-0.0000001
    }
    myloglik<-binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
    sumloglik<-myloglik+sumloglik
  }
}
return(p)
}

# get estimated p using best parameter estimates
datapcounts<-5
datatotals<-6
eda_estimatedp3selfor<-estimate_selection_for_m3_returnp(edom3$par)

# Plot Model 3 time series
pdf("Figure5_Model3_timeseries.pdf",width=9,height=6)
plot(c(1983,data[,1]),eda_estimatedp3selfor,type="n",xlab="year",ylab="frequency",ylim=c(0,1),main="ED
A (pink) selection For p model 3",cex.main=0.6)
points(c(1983,data[,1]),eda_estimatedp3selfor,col="magenta",pch=19,cex=0.8)
lines(c(1983,data[,1]),eda_estimatedp3selfor,col="magenta",pch=19,lwd=2)
points(data[,1],data[,5]/data[,6],pch=21,col="magenta",cex=1.5)
text(1998,0.4,labels=sprintf("EDA\\ns = %.3f [+/-%.3fSE]\\nh = %.3f [+/-%.3fSE]\\np1983 = %.3f [+/-
%.3fSE]\\n-ve log Likelihood =
%s",edom3$par[1],se_edom3[1],edom3$par[2],se_edom3[2],edom3$par[3],se_edom3[3],edom3$value),cex=0.6,pos
=4,col="magenta")
dev.off()

# plot the likelihood surfaces
# we need to estimate likelihood for values of h and s at fixed p1983 frequency from best fit model.
modify the function to fix p1983 with best value

# Model 3 function return p
estimate_selection_for_m3_fixedp<-function(parmselform3) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-fixedp
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t]=((p[t-1]*p[t-1])*(1+parmselform3[1])+((2*p[t-1]*q[t-
      1])*(1+(parmselform3[2]*parmselform3[1])))/2) / ((p[t-1]*p[t-
      1])*(1+parmselform3[1])+((2*p[t-1]*q[t-
      1])*(1+(parmselform3[2]*parmselform3[1])))) + (q[t-1]*q[t-1]))
      q[t]<-1-p[t]
    } else {

```

```

# for all other subsequent years
# Model 3: 1 year generation time - the frequency is dependent on frequency from
one year prior.
pbreeders<-p[t-1]
qbreeders<-(1-pbreeders)
p[t]<-((pbreeders*pbreeders)*(1+parmselform3[1]) +
((2*pbreeders*qbreeders)*(1+(parmselform3[2]*parmselform3[1])))/2 ) / (
(pbreeders*pbreeders)*(1+parmselform3[1]) +
((2*pbreeders*qbreeders)*(1+(parmselform3[2]*parmselform3[1])) +
(qbreeders*qbreeders))
q[t]<-1-p[t]
}
# check if we have observed data in this year
# in years we have observed data, evaluate the fit of the predicted frequency p(t) to
the observed data.
if (is.na(data[t-1,datapoints])==F) {
  # we have data
  # the binomial likelihood function can't log0... so make sure p[t] never quite
reaches fixation.
  if(p[t]==1) {
    p[t]=p[t]-0.0000001
  }
  myloglik<-binom.lik(data[t-1,datapoints],data[t-1,datatotals],p[t])
  sumloglik<-myloglik+sumloglik
}
}
return(sumloglik)
}

# run for EDA
datapoints<-5
datatotals<-6

# EDA fixed p
fixedp<-edam3$par[3]

# calculate likelihood of model for all values of s and h ranging from 0 to 2 in steps of 100
mymax<-2
points2<-100
allspacem4<-matrix(nrow=points2,ncol=points2)
for (ss2 in seq(1,points2,by=1)) {
  sparm2<-mymax*(ss2/points2)
  for (hh2 in seq(1,points2,by=1)) {
    hparm2<-mymax*(hh2/points2)
    parms2<-c(sparm2,hparm2)
    mylikelihood2<-estimate_selection_for_m3_fixedp(parms2)
    allspacem4[ss2,hh2]<-mylikelihood2
  }
}

# plot model 3 likelihood surface
x<-seq(0+(mymax/points2),mymax,by=mymax/points2)
y<-seq(0+(mymax/points2),mymax,by=mymax/points2)
colnum<-10
pdf(("Figure6_Model3_likelihoood.pdf",width=7,height=6)
filled.contour(x,y,allspacem4,plot.axes=c(points(x=edam3$par[1],y=edam3$par[2],cex=1.5,pch=19),axis(sid
e=1,labels=seq(0,mymax,by=0.5),at=seq(0,mymax,by=0.5)),axis(side=2,labels=seq(0,mymax,by=0.5),at=seq(0,
mymax,by=0.5))),col=c(rainbow(colnum,start=0.05,end=0.90)),levels=c(seq(min(allspacem4),(min(allspacem4
)+(2*(colnum))),by=2)),key.axes=axis(4,seq(min(allspacem4)+2,min(allspacem4)+2*(colnum+1),by=2)),xlab="
selection",ylab="dominance",main=sprintf("EDA selection for p (freshwater allele) model 3, 1 year
generation time \ns=%3f+/-SE:%3f, h=%3f+/-SE:%3f, p1983=%3f+/-SE:%3f,\n -ve log Likelihood
=
%s",edam3$par[1],se_edam3[1],edam3$par[2],se_edam3[2],edam3$par[3],se_edam3[3],edam3$value),cex.main=0.
6)
dev.off()

```

```

# Model 4 - selection favors freshwater allele, 1 & 2 year old fish breed and contribute to subsequent
generations
# priors (s,h,p1983,t1)
parmselform4<-c(0.2,0.5,0.035,0.5)

# set parameter limits
m4lowerbounds<-c(0.001, -0.5, 0.001,0.001)
m4upperbounds<-c(2, 1.5, 0.999,0.999)

```



```

# Model 4 function - 1 & 2 year old fish breed and contribute to subsequent generations
estimate_selection_for_m4<-function(parmselform4) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-parmselform4[3]
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t]=((p[t-1]*p[t-1])*(1+parmselform4[1])+((2*p[t-1]*q[t-1])*(1+(parmselform4[2]*parmselform4[1])))/2) / ((p[t-1]*p[t-1])*(1+parmselform4[1])+((2*p[t-1]*q[t-1])*(1+(parmselform4[2]*parmselform4[1])))) + (q[t-1]*q[t-1]))
      q[t]<-1-p[t]
    } else {
      #for all other subsequent years
      #Model 4: breeders come from previous 2 years with proportion t1(1yr) and 1-t1(2yr)
      pbreeders<-(p[t-1]*parmselform4[4])+(p[t-2]*(1-parmselform4[4]))
      qbreeders<-(1-pbreeders)
      p[t]<-((pbreeders*pbreeders)*(1+parmselform4[1]) + ((2*pbreeders*qbreeders)*(1+(parmselform4[2]*parmselform4[1])))/2) / ((pbreeders*pbreeders)*(1+parmselform4[1]) + ((2*pbreeders*qbreeders)*(1+(parmselform4[2]*parmselform4[1])))) + (qbreeders*qbreeders))
      q[t]<-1-p[t]
    }
    # check if we have observed data in this year
    # in years we have observed data, evaluate the fit of the predicted frequency p(t) to the observed data.
    if (is.na(data[t-1,datapcounts])==F) {
      # we have data
      # the binomial likelihood function can't log0... so make sure p[t] never quite reaches fixation.
      if(p[t]==1) {
        p[t]=p[t]-0.0000001
      }
      myloglik<-binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
      sumloglik<-myloglik+sumloglik
    }
  }
  return(sumloglik)
}

# Run model 4 selection for p for EDA
parmselform4<-c(0.2,0.5,0.035,0.5)

#set parameter limits
m4lowerbounds<-c(0.001, -0.5, 0.001,0.001)
m4upperbounds<-c(2, 1.5, 0.999,0.999)
datapcounts<-5
datatotals<-6
edam4<-optim(parmselform4,fn=estimate_selection_for_m4,hessian=T,lower=m4lowerbounds,upper=m4upperbounds,method="L-BFGS-B")
edaOI_m4<-solve(edam4$hessian)
se_edam4<-sqrt(diag(edaOI_m4))

#eda parameters s, h, p1983,t1
edam4$par
[1] 1.27767733 0.75099130 0.00813214 0.78846583
#-ve log likelihood
edam4$value
[1] 457.1813
#standard errors - sqrt of negative diagonals of Fisher Information Matrix is not possible - will need to bootstrap this.

# Bootstrapping selection for Model 4
#save original data as data2
data2<-data

datapcounts<-5
datatotals<-6
myrepetitions<-1000
bootsample<-data.frame(matrix(NA,nrow=length(data[,1]),ncol=myrepetitions))
for (j in 1:myrepetitions) {
  for (i in 1:length(data[,1])) {

```

```

        if(is.na(data[i,datatotals])==T) {
          bootsample[i,j]<-NA
        } else {
          bootsample[i,j]<-
            sum(sample(c(rep(1,data[i,datatotals]),rep(0,data[i,datatotals]-
              data[i,datapcounts])),data[i,datatotals],replace=T))
        }
      }
    }

#priors
parmsel4form4<-c(0.2,0.5,0.035,0.5)
bootPars<-data.frame(matrix(NA,nrow=myrepetitions,ncol=length(parmsel4form4)+1))

for (k in 1:myrepetitions) {
  print(k)
  data[,datapcounts]<-bootsample[,k]
  parmsel4form4<-c(0.2,0.5,0.035,0.5)
  m4lowerbounds<-c(0.001, -0.5, 0.001,0.001)
  m4upperbounds<-c(2, 1.5, 0.999,0.999)
  tmp<-optim(parmsel4form4,fn=estimate_selection_for_m4,hessian=T,lower=m4lowerbounds,
    upper=m4upperbounds,method="L-BFGS-B")
  bootPars[k,]<-c(tmp$par,tmp$value)
}

data<-data2
edabootm4Pars<-bootPars[which(is.na(bootPars[,1])==F),]
edabootm4Pars<-edabootm4Pars[1:500,]
colnames(edabootm4Pars)<-c("s","h","p1983","tlcontribution","Negloglikelihood")

myresults<-data.frame(matrix(NA,nrow=5,ncol=length(parmsel4form4)+1))
myresults[1,]<-sapply(edabootm4Pars, mean, na.rm = TRUE)
myresults[2,]<-sapply(edabootm4Pars, function(x) {sd(x[which(is.na(x)==F)])})
myresults[3,]<-sapply(edabootm4Pars, function(x)
  {sd(x[which(is.na(x)==F)])/sqrt(length(x[which(is.na(x)==F)])})})
myresults[4,]<-sapply(edabootm4Pars, function(x) {quantile(x,c(0.025),na.rm=T)})
myresults[5,]<-sapply(edabootm4Pars, function(x) {quantile(x,c(0.975),na.rm=T)})
rownames(myresults)<-c("mean","sd","se","lower2.5CI","upper97.5CI")
colnames(myresults)<-c("s","h","p1983","tlcontribution","Negloglikelihood")
edabootm4Results<-myresults

edabootm4Results

> edabootm4Results
      s          h      p1983 tlcontribution Negloglikelihood
mean 1.36749480 0.744341676 0.084061289    0.6827928      422.13432
sd    0.52719265 0.216245091 0.118284302    0.2654392      21.12995
se    0.02357677 0.009670774 0.005289835    0.0118708      0.94496

# prepare to plot model 4 time series
# we need the estimated p freq from best model
# modify the function to return p instead of -log likelihood.
# Model 4 - 1 & 2 year old fish breed, selection favors fw allele
# priors (s,h,p1983,tl)
parmsel4form4<-c(0.2,0.5,0.035,0.5)

#set parameter limits
m4lowerbounds<-c(0.001, -0.5, 0.001,0.001)
m4upperbounds<-c(2, 1.5, 0.999,0.999)

# Model 4 function to return p
estimate_selection_for_m4_returnp<-function(parmsel4form4) {
  p<-vector(mode="numeric",length=length(data[,1])+1)
  q<-vector(mode="numeric",length=length(data[,1])+1)
  p[1]<-parmsel4form4[3]
  q[1]<-1-p[1]
  sumloglik<-0
  for (t in 2:(length(data[,1])+1)) {
    if (t==2) {
      #the frequency in the first year after founding is dependent solely 1983
      founding frequency.
      p[t]=((p[t-1]*p[t-1])*(1+parmsel4form4[1])+((2*p[t-1]*q[t-1])
        *(1+(parmsel4form4[2]*parmsel4form4[1])))/2) / ((p[t-1]*p[t-1])
        *(1+parmsel4form4[1])+((2*p[t-1]*q[t-1])
        *(1+(parmsel4form4[2]*parmsel4form4[1])))) + (q[t-1]*q[t-1]))
      q[t]<-1-p[t]
    } else {
      #for all other subsequent years

```

```

#Model 4: breeders come from previous 2 years with proportion t1(1yr) and 1-
t1(2yr)
pbreeders<-(p[t-1]*parmsel4[4])+(p[t-2]*(1-parmsel4[4]))
qbreeders<-(1-pbreeders)
p[t]<-((pbreeders*pbreeders)*(1+parmsel4[1]) +
((2*pbreeders*qbreeders)*(1+(parmsel4[2]*parmsel4[1])))/2 ) / (
(pbreeders*pbreeders)*(1+parmsel4[1]) +
((2*pbreeders*qbreeders)*(1+(parmsel4[2]*parmsel4[1])))) +
(qbreeders*qbreeders))
q[t]<-1-p[t]
}
# check if we have observed data in this year
# in years we have observed data, evaluate the fit of the predicted frequency p(t) to
the observed data.
if (is.na(data[t-1,datapoints])==F) {
  # we have data
  # the binomial likelihood function can't log0... so make sure p[t] never quite
  reaches fixation.
  if(p[t]==1) {
    p[t]=p[t]-0.0000001
  }
  myloglik<-binom.lik(data[t-1,datapoints],data[t-1,datatotals],p[t])
  sumloglik<-myloglik+sumloglik
}
}
return(p)
}

# get values of p for best model4
datapoints<-5
datatotals<-6
eda_estimatedpm4<-estimate_selection_for_m4_returnp(edam4$par)

# plot model 4 time series
pdf("Figure7_Model4_timeseries.pdf",width=9,height=6)
plot(c(1983,data[,1]),eda_estimatedpm4,type="n",xlab="year",ylab="frequency",ylim=c(0,1),main="EDA
(pink) selection favoring freshwater allele p model 4 ",cex.main=0.6)
points(c(1983,data[,1]),eda_estimatedpm4,col="magenta",pch=19,cex=0.8)
lines(c(1983,data[,1]),eda_estimatedpm4,col="magenta",pch=19,lwd=2)
points(data[,1],data[,5]/data[,6],pch=21,col="magenta",cex=1.5)
text(1998,0.4,labels=sprintf("EDA\\ns = %.3f [+/-%.3fSE]\\nh = %.3f [+/-%.3fSE]\\np1983 = %.3f [+/-
%.3fSE]\\ntl = %.3f [+/-%.3fSE]\\n-ve log Likelihood =
%s",edam4$par[1],edabootm4Results[3,1],edam4$par[2],edabootm4Results[3,2],edam4$par[3],edabootm4Results
[3,3],edam4$par[4],edabootm4Results[3,4],edam4$value),cex=0.6,pos=4,col="magenta")
dev.off()

# finally plot the model 4 likelihood surfaces.
# modify function to explore s and h for the best fixedp and fixed t1 value
estimate_selection_for_m4_fixedp<-function(parmsel4) {
p<-vector(mode="numeric",length=length(data[,1])+1)
q<-vector(mode="numeric",length=length(data[,1])+1)
p[1]<-fixedp
q[1]<-1-p[1]
sumloglik<-0
for (t in 2:(length(data[,1])+1)) {
  if (t==2) {
    #the frequency in the first year after founding is dependent solely 1983
    founding frequency.
    p[t]=((p[t-1]*p[t-1])*(1+parmsel4[1])+(2*p[t-1]*q[t-
1])*(1+(parmsel4[2]*parmsel4[1]))/2 ) / ((p[t-1]*p[t-
1])*(1+parmsel4[1])+(2*p[t-1]*q[t-
1])*(1+(parmsel4[2]*parmsel4[1])))) + (q[t-1]*q[t-1]))
    q[t]<-1-p[t]
  } else {
    #for all other subsequent years
    #Model 4: breeders come from previous 2 years with proportion t1(1yr) and 1-
    t1(2yr)
    pbreeders<-(p[t-1]*fixedt1)+(p[t-2]*(1-fixedt1))
    qbreeders<-(1-pbreeders)
    p[t]<-((pbreeders*pbreeders)*(1+parmsel4[1]) +
((2*pbreeders*qbreeders)*(1+(parmsel4[2]*parmsel4[1])))/2 ) / (
(pbreeders*pbreeders)*(1+parmsel4[1]) +
((2*pbreeders*qbreeders)*(1+(parmsel4[2]*parmsel4[1])))) +
(qbreeders*qbreeders))
    q[t]<-1-p[t]
  }
}
# check if we have observed data in this year

```

```

# in years we have observed data, evaluate the fit of the predicted frequency p(t) to
the observed data.
if (is.na(data[t-1,datapcounts])==F) {
  # we have data
  # the binomial likelihood function can't log0... so make sure p[t] never quite
  reaches fixation.
  if(p[t]==1) {
    p[t]=p[t]-0.0000001
  }
  myloglik<-binom.lik(data[t-1,datapcounts],data[t-1,datatotals],p[t])
  sumloglik<-myloglik+sumloglik
}
}
return(sumloglik)
}

# run for EDA
datapcounts<-5
datatotals<-6
# EDA fixed p
fixedp<-edam4$par[3]
fixedt1<-edam4$par[4]

#now calculate likelihood of model for all values of s and h ranging from 0 to 1 in steps of 100
myhmin<-(-0.5)
myhmax<-1.5
mysmax<-1
points2<-100
hcounter<-0
allspacem4<-matrix(nrow=points2,ncol=points2)
for (hh2 in seq(myhmin,myhmax,by=((myhmax-myhmin)/(points2-1)))) {
  hparm4<-hh2
  hcounter<-hcounter+1
  for (ss2 in seq(1,points2,by=1)) {
    sparm4<-mysmax*(ss2/points2)
    parms2<-c(sparm4,hparm4)
    mylikelihood2<-estimate_selection_for_m4_fixedp(parms2)
    allspacem4[ss2,hcounter]<-mylikelihood2
  }
}

# plot likelihood surface, model4
x<-seq(0+(mysmax/points2),mysmax,by=mysmax/points2)
y<-seq(-0.5,1.48,by=2/points2)
colnum<-10
pdf(("Figure8_Model4_likelihood.pdf",width=7,height=6)
filled.contour(x,y,allspacem4,plot.axes=c(points(x=edam4$par[1],y=edam4$par[2],cex=1.5,pch=19),axis(sid
e=1,labels=seq(0,mysmax,by=0.5),at=seq(0,mysmax,by=0.5)),axis(side=2,labels=seq(0,mysmax,by=0.5),at=seq(0,
mysmax,by=0.5))),col=c(rainbow(colnum,start=0.05,end=0.90)),levels=c(seq(min(allspacem4,na.rm=T),(min(al
lspacem4,na.rm=T)+(2*(colnum))),by=2)),key.axes=axis(4,seq(min(allspacem4,na.rm=T)+2,min(allspacem4,na.
rm=T)+2*(colnum+1),by=2)),xlab="selection",ylab="dominance",main=sprintf("EDA selection For p model4 1
and 2 year breeders unbounded \ns=%.3f[+/-SE:%.3f], h=%.3f[+/-SE:%.3f], p1983=%.3f[+/-SE:%.3f],
t1=%.3f[+/-SE:%.3f]\n -ve log Likelihood =
%s",edam4$par[1],edabootm4Results[3,1],edam4$par[2],edabootm4Results[3,2],edam4$par[3],edabootm4Results
[3,3],edam4$par[4],edabootm2Results[3,4],edam2$value),cex.main=0.6)
dev.off()

```