

Test of Biomass as Covariate in Aster Models for Lifetime Fitness

Mason W. Kulbaba & Ruth G. Shaw

Last Updated June 28, 2020

Introduction

The following code tests, on the advice of Dr. Daniel Bolnick and Dr. Lynda Delph, the effect of biomass on female and male fitness, as determined with aster models.

Preliminaries

Set working directory and load data.

Examine data

```
fin<- read.csv("data/aster.dat.csv")
```

```
names(fin)
```

```
## [1] "Treat"      "plotID"     "Den"        "Gen"        "plantID"
## [6] "familyID"   "surv"       "flw"        "frt"        "frt.2"
## [11] "seeds"      "aborted"    "rel.seeds"  "seed.wt"    "mass.a"
## [16] "mass.b"
```

```
head(fin)
```

```
##   Treat plotID Den Gen plantID familyID surv flw frt frt.2 seeds aborted
## 1 HDHG      5  H  HG      1         4    1  1  3    2    33      14
## 2 HDHG      5  H  HG      2         6    1  1 21    6    69      22
## 3 HDHG      5  H  HG      3        11    1  1 16    7    73      19
## 4 HDHG      5  H  HG      4        14    1  1 18    8   107      23
## 5 HDHG      5  H  HG      5         6    1  1  1    1    16       4
## 6 HDHG      5  H  HG      6        14    1  1  0    0     0       0
##   rel.seeds seed.wt mass.a mass.b
## 1 0.5714286  6.792   5.8   0.8
## 2 1.1948052  6.995   1.9   0.2
## 3 1.2640693  7.413  20.8   1.3
## 4 1.8528139  8.554  18.2   1.1
## 5 0.2770563  7.771  65.8   6.1
## 6 0.0000000  0.000   6.6   1.2
```

Make sure Den (density treatment), Gen (N_e treatment), plotID, plantID, and familyID are all classified as factors. Otherwise

```
fin$Den<- as.factor(fin$Den)
fin$Gen<- as.factor(fin$Gen)
fin$plotID<- as.factor(fin$plotID)
fin$plantID<- as.factor(fin$plantID)
fin$familyID<- as.factor(fin$familyID)
```

Setting up for aster analyses

Load aster package (make sure you have most current version)

```
library(aster)
```

Begin by naming variables that will be used in the graphical model of the aster analyses:

flw - total number of flowers produced **frr** - total number of fruits produced **frr.2** - subsetting number of fruits collected **seeds** - total number of seeds collected from subsetting fruits

```
vars<- c( "flw", "frr", "frr.2","seeds")
```

Reshape the data so that all response variables are located in a single vector, in a new data set called “redata”

```
redata <- reshape(fin, varying = list(vars), direction = "long",timevar = "varb", times = as.factor(varb))
```

Designate the terminal fitness variable “seeds” (make it numeric), and then add it to the reshaped data

```
fit <- grepl("seeds", as.character(redata$varb))
fit<- as.numeric(fit)
redata$fit <- fit
```

Check

```
with(redata, sort(unique(as.character(varb)[fit == 0])))
```

```
## [1] "flw" "frr" "frr.2"
```

```
with(redata, sort(unique(as.character(varb)[fit == 1])))
```

```
## [1] "seeds"
```

Add a variable “root” to redata, where value is 1. This is the “starting point” of the aster graphical model (i.e. a seed planted)

```
redata<- data.frame(redata, root=1)
```

Set up the graphical model and designate the statistical distribution for each node. This graphical model has four nodes (in object **pred**) described earlier. Statistical family for each node is described by object **fam**.

```
pred<- c(0,1,2,3)
fam<- c(1,2,2,2)
```

Show distribution family for each node

```
sapply(fam.default(), as.character)[fam]
```

```
## [1] "bernoulli" "poisson" "poisson" "poisson"
```

Main Aster Analyses for Female Fitness (Seeds set)

First aster analysis with only fitness data. Note, **aster** reads the **redata** version of the data.

```
aout<- aster(resp~varb, pred, fam, varb, id, root, data=redata)
```

```
aout1<- aster(resp~varb + fit:(familyID), pred, fam, varb, id, root, data=redata)
```

```
aout2<- aster(resp~varb + fit:(familyID + Gen), pred, fam, varb, id, root, data=redata)
```

```
aout3<- aster(resp~varb + fit:(familyID + Gen + Den), pred, fam, varb, id, root, data=redata)
```

```

aout4<- aster(resp~varb + fit:(familyID + Gen + Den + Gen*Den), pred, fam, varb, id, root, data=redata)
aout5<- aster(resp~varb + fit:(familyID + Gen + Den + Gen*Den + mass.a) , pred, fam, varb, id, root, da

anova(aout, aout1, aout2, aout3, aout4, aout5)

## Analysis of Deviance Table
##
## Model 1: resp ~ varb
## Model 2: resp ~ varb + fit:(familyID)
## Model 3: resp ~ varb + fit:(familyID + Gen)
## Model 4: resp ~ varb + fit:(familyID + Gen + Den)
## Model 5: resp ~ varb + fit:(familyID + Gen + Den + Gen * Den)
## Model 6: resp ~ varb + fit:(familyID + Gen + Den + Gen * Den + mass.a)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1         4    132606
## 2        15    132996 11   389.52 < 2.2e-16 ***
## 3        16    133274  1   277.97 < 2.2e-16 ***
## 4        18    133816  2   542.55 < 2.2e-16 ***
## 5        20    133935  2   118.54 < 2.2e-16 ***
## 6        21    134028  1    93.14 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Interpretation

Sequential addition of familyID, Gen (N_e), Den (density), Gen * Den (Gen by Den interaction), and finally mass.a (above ground biomass) consistently explained more variation in total lifetime fitness. Next, move on to specific models to estimate fitness across treatmntns.

First, isolate high (HG) and low (LG) data from the main redata file, and drop unused levels. Therefore, don't have to do "reshape" data step. HG/hg = High N_e, LG/lg = Low N_e

```

redataHG<- subset(redata, Gen=="HG")
redataLG<- subset(redata, Gen=="LG")

redataHG<- droplevels(redataHG)
redataLG<- droplevels(redataLG)

```

Perform aster analysis on HG and LG data with just fitness data (no predictors), then add Den and perform a likelihood ratio test.

```

aouthG<- aster(resp~varb, pred, fam, varb, id, root, data=redataHG)
aouthG2<- aster(resp~varb + fit:(familyID), pred, fam, varb, id, root, data=redataHG)
aouthG3<- aster(resp~varb + fit:(familyID + Den), pred, fam, varb, id, root, data=redataHG)
aouthG4<- aster(resp~varb + fit:(familyID + Den + mass.a), pred, fam, varb, id, root, data=redataHG)

anova(aouthG, aouthG2, aouthG3, aouthG4)

## Analysis of Deviance Table
##
## Model 1: resp ~ varb
## Model 2: resp ~ varb + fit:(familyID)
## Model 3: resp ~ varb + fit:(familyID + Den)
## Model 4: resp ~ varb + fit:(familyID + Den + mass.a)

```

```

##      Model Df Model Dev Df Deviance P(>|Chi|)
## 1         4      89573
## 2        13     90086  9   512.72 < 2.2e-16 ***
## 3        15     90599  2   513.25 < 2.2e-16 ***
## 4        16     90785  1   185.98 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(aoutHG4, show.graph=T, info.tol = 1e-9)

##
## Call:
## aster.formula(formula = resp ~ varb + fit:(familyID + Den + mass.a),
##      pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##      data = redataHG)
##
##
## Graphical Model:
## variable predecessor family
## flw      root      bernoulli
## frt      flw      poisson
## frt.2    frt      poisson
## seeds    frt.2    poisson
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.623e+01  8.223e-01 -44.060 < 2e-16 ***
## varbfrt      4.078e+01  8.334e-01  48.933 < 2e-16 ***
## varbfrt.2    2.529e+01  8.208e-01  30.808 < 2e-16 ***
## varbseeds    3.863e+01  8.226e-01  46.960 < 2e-16 ***
## fit:familyID1 5.225e-02  8.080e-03   6.466 1.01e-10 ***
## fit:familyID2 6.449e-02  7.678e-03   8.400 < 2e-16 ***
## fit:familyID3 2.464e-02  1.174e-02   2.099 0.03585 *
## fit:familyID4 -7.718e-02  1.915e-02  -4.030 5.57e-05 ***
## fit:familyID6 -4.465e-02  1.684e-02  -2.651 0.00802 **
## fit:familyID7 6.854e-02  8.245e-03   8.313 < 2e-16 ***
## fit:familyID8 1.093e-01  8.229e-03  13.287 < 2e-16 ***
## fit:familyID9 2.414e-02  9.139e-03   2.641 0.00827 **
## fit:familyID11 6.367e-02  7.656e-03   8.317 < 2e-16 ***
## fit:DenL      -1.268e-01  6.418e-03 -19.755 < 2e-16 ***
## fit:DenM      -3.788e-02  3.688e-03 -10.273 < 2e-16 ***
## fit:mass.a    -2.000e-03  1.692e-04 -11.819 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Original predictor variables dropped (aliased)
##      fit:familyID14

```

Perform same sequence of analyses on LG dataset

```

aoutLG<- aster(resp~varb, pred, fam, varb, id, root, data=redataLG)
aoutLG2<- aster(resp~varb + fit:(familyID), pred, fam, varb, id, root, data=redataLG)
aoutLG3<- aster(resp~varb + fit:(familyID + Den), pred, fam, varb, id, root, data=redataLG)
aoutLG4<- aster(resp~varb + fit:(familyID + Den + mass.a), pred, fam, varb, id, root, data=redataLG)

anova(aoutLG, aoutLG2, aoutLG3, aoutLG4)

```

```
## Analysis of Deviance Table
##
## Model 1: resp ~ varb
## Model 2: resp ~ varb + fit:(familyID)
## Model 3: resp ~ varb + fit:(familyID + Den)
## Model 4: resp ~ varb + fit:(familyID + Den + mass.a)
##   Model Df Model Dev Df Deviance P(>|Chi|)
## 1      4      43589
## 2     13     43633  9   44.179 1.308e-06 ***
## 3     15     43725  2   92.203 < 2.2e-16 ***
## 4     16     43727  1    1.989  0.1585
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(aoutLG4, show.graph = T, info.tol = 1e-8)

##
## Call:
## aster.formula(formula = resp ~ varb + fit:(familyID + Den + mass.a),
##   pred = pred, fam = fam, varvar = varb, idvar = id, root = root,
##   data = redataLG)
##
##
## Graphical Model:
##   variable predecessor family
##   flw      root      bernoulli
##   frt      flw      poisson
##   frt.2    frt      poisson
##   seeds    frt.2    poisson
##
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.898e+01  8.207e-01 -47.487 < 2e-16 ***
## varbfrt      4.351e+01  8.334e-01  52.203 < 2e-16 ***
## varbfrt.2    2.984e+01  8.228e-01  36.265 < 2e-16 ***
## varbseeds    4.116e+01  8.204e-01  50.174 < 2e-16 ***
## fit:familyID1 6.275e-03  8.761e-03   0.716  0.4738
## fit:familyID2 2.985e-02  1.449e-02   2.061  0.0393 *
## fit:familyID3 -2.516e-02  1.388e-02  -1.813  0.0698 .
## fit:familyID7 5.232e-03  8.770e-03   0.597  0.5508
## fit:familyID8 -2.565e-02  1.397e-02  -1.836  0.0663 .
## fit:familyID9 1.646e-02  8.517e-03   1.932  0.0533 .
## fit:familyID10 -3.130e-02  1.381e-02  -2.267  0.0234 *
## fit:familyID11 1.015e-01  1.323e-02   7.670 1.72e-14 ***
## fit:familyID13 -7.711e-02  1.759e-02  -4.384 1.17e-05 ***
## fit:DenL      -7.115e-02  1.024e-02  -6.947 3.73e-12 ***
## fit:DenM      -4.016e-03  7.178e-03  -0.559  0.5758
## fit:mass.a     3.282e-04  2.252e-04   1.457  0.1451
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Original predictor variables dropped (aliased)
##   fit:familyID14
```

Interpretation

Interestingly, the addition of `mass.a` explains significantly more variation in only the high N_e treatments. Now generate estimates of fitness (female function - seeds set) for both high and low N_e .

```
pout.HG<- predict(aoutHG4, se.fit=TRUE, info.tol = 1e-9)
```

```
pout.LG<- predict(aoutLG4, se.fit=TRUE, info.tol = 1e-8)
```

Playing with predicted values

```
f<- pout.HG$fit
f<- as.data.frame(f)
dim(f)
```

```
## [1] 432 1
```

```
se<- pout.HG$se.fit
se<- as.data.frame(se)
dim(se)
```

```
## [1] 432 1
```

```
flw.se<- as.data.frame(se$se[1:108])
ftr.se<- as.data.frame(se$se[109: 216])
ftr2.se<- as.data.frame(se$se[217:324])
seeds.se<- as.data.frame(se$se[325:432])
```

```
nodes.se<- cbind(flw.se, ftr.se, ftr2.se, seeds.se)
```

```
colnames(nodes.se)<- c("flw.se", "ftr.se", "ftr2.se", "seeds.se")
```

```
hg<- subset(fin, Gen=="HG")
```

```
ex.hg<- hg[c("Treat", "familyID", "plotID", "Den")]
```

```
flw.n<- as.data.frame(f$f[1:108])
ftr.n<- as.data.frame(f$f[109: 216])
ftr2.n<- as.data.frame(f$f[217:324])
seeds.n<- as.data.frame(f$f[325:432])
```

```
nodes<- cbind(flw.n, ftr.n, ftr2.n, seeds.n)
```

```
colnames(nodes)<- vars
```

```
fit.hg<- cbind(ex.hg, nodes, nodes.se)
```

```
female.fit<- aggregate(fit.hg$seeds, by=list(fit.hg$Den), mean)
```

```
female.se<- aggregate(fit.hg$seeds.se, by=list(fit.hg$Den), mean)
```

```
colnames(female.fit)<- c("Density", "Female.Fitness")
```

```
colnames(female.se)<- c("Density", "SE")
```

```
female.fit
```

```
## Density Female.Fitness
## 1 H 418.02778
```

```
## 2      L      57.83333
## 3      M      180.58333
```

```
female.se
```

```
##      Density      SE
## 1      H 39.029146
## 2      L  6.429881
## 3      M 16.527467
```

Try the old way until the second ‘predict’ statement

```
hg<- subset(fin, Gen=="HG")
hg<- droplevels(hg)
```

```
fred.hg <- data.frame( familyID=hg$familyID, Den=hg$Den, mass.a=hg$mass.a, flw=1, frt=1,frt.2=1, seeds=
```

```
lg<-subset(fin, Gen=="LG")
```

```
fred.lg <- data.frame( familyID=lg$familyID, Den=lg$Den, mass.a=lg$mass.a, flw=1, frt=1,frt.2=1, seeds=
```

Reshape the design matrix just as the actual data

```
renewdata.hg <- reshape(fred.hg, varying = list(vars),
                        direction = "long", timevar = "varb",
                        times = as.factor(vars), v.names = "resp")
```

```
renewdata.lg <- reshape(fred.lg, varying = list(vars),
                        direction = "long", timevar = "varb",
                        times = as.factor(vars), v.names = "resp")
```

Make character string from “varb” of renewdata without actual values (i.e., the layers of varb in renewdata), and add it to each renewdata object

```
layer<- gsub("[0-9]", "", as.character(renewdata.hg$varb))
```

```
renewdata.hg<- data.frame(renewdata.hg, layer= layer)
renewdata.lg<- data.frame(renewdata.lg, layer= layer)
```

Add “seeds” in new layer column of renewdata as numeric, called fit Note: only need one fit object as it is the same for both High and Low N_e data, and add to each renew data file

```
fit<- as.numeric(layer=="seeds")
```

```
renewdata.hg<- data.frame(renewdata.hg, fit = fit)
renewdata.lg<- data.frame(renewdata.lg, fit = fit)
```

Rerun prediction of aster analyses, with the reshaped design matrices

```
pout.hg<- predict(aoutHG4, newdata= renewdata.hg, varvar= varb,
                 idvar = id, root = root, se.fit = TRUE, info.tol = 1e-9)

pout.lg<- predict(aoutLG4, newdata= renewdata.lg, varvar= varb,
                 idvar = id, root = root, se.fit = TRUE, info.tol = 1e-8)
```

Now isolate nodes and se for HG analysis

```

f<- pout.hg$fit
f<- as.data.frame(f)
dim(f)

## [1] 432 1

se<- pout.hg$se.fit
se<- as.data.frame(se)
dim(se)

## [1] 432 1

flw.se<- as.data.frame(se$se[1:108])
frt.se<- as.data.frame(se$se[109: 216])
frt2.se<- as.data.frame(se$se[217:324])
seeds.se<- as.data.frame(se$se[325:432])

nodes.se<- cbind(flw.se, frt.se, frt2.se, seeds.se)

colnames(nodes.se)<- c("flw.se", "frt.se", "frt2.se", "seeds.se")

ex.hg<- hg[c("Treat", "familyID", "plotID", "Den", "mass.a")]

flw.n<- as.data.frame(f$f[1:108])
frt.n<- as.data.frame(f$f[109: 216])
frt2.n<- as.data.frame(f$f[217:324])
seeds.n<- as.data.frame(f$f[325:432])

nodes<- cbind(flw.n, frt.n, frt2.n, seeds.n)

colnames(nodes)<- vars

fit.hg<- cbind(ex.hg, nodes, nodes.se)

female.fit<- aggregate(fit.hg$seeds, by=list(fit.hg$Den), mean)

female.se<- aggregate(fit.hg$seeds.se, by=list(fit.hg$Den), mean)

colnames(female.fit)<- c("Density", "Female.Fitness")
colnames(female.se)<- c("Density", "SE")

female.fit

## Density Female.Fitness
## 1 H 418.02778
## 2 L 57.83333
## 3 M 180.58333
female.se

## Density SE
## 1 H 39.029146
## 2 L 6.429881
## 3 M 16.527467

```


Now isolate nodes and se for LG analysis

```
f<- pout.lg$fit
f<- as.data.frame(f)
dim(f)

## [1] 432 1

se<- pout.lg$se.fit
se<- as.data.frame(se)
dim(se)

## [1] 432 1

flw.se<- as.data.frame(se$se[1:108])
frr.se<- as.data.frame(se$se[109: 216])
frr2.se<- as.data.frame(se$se[217:324])
seeds.se<- as.data.frame(se$se[325:432])

nodes.se<- cbind(flw.se, frr.se, frr2.se, seeds.se)

colnames(nodes.se)<- c("flw.se", "frr.se", "frr2.se", "seeds.se")

ex.lg<- lg[c("Treat", "familyID", "plotID", "Den", "mass.a")]

flw.n<- as.data.frame(f$f[1:108])
frr.n<- as.data.frame(f$f[109: 216])
frr2.n<- as.data.frame(f$f[217:324])
seeds.n<- as.data.frame(f$f[325:432])

nodes<- cbind(flw.n, frr.n, frr2.n, seeds.n)

colnames(nodes)<- vars

fit.lg<- cbind(ex.lg, nodes, nodes.se)

female.fit<- aggregate(fit.lg$seeds, by=list(fit.lg$Den), mean)

female.se<- aggregate(fit.lg$seeds.se, by=list(fit.lg$Den), mean)

colnames(female.fit)<- c("Density", "Female.Fitness")
colnames(female.se)<- c("Density", "SE")

female.fit

## Density Female.Fitness
## 1 H 122.91666
## 2 L 67.86112
## 3 M 102.75000
female.se

## Density SE
## 1 H 14.72894
## 2 L 12.05069
## 3 M 12.60316
```