Homework 7

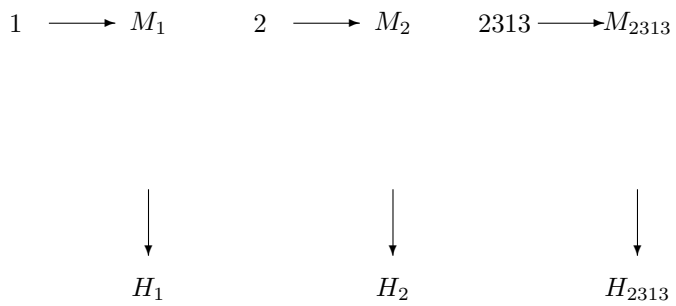
Solution Set

Problem 1 [80 points]: We are going to analyze fitness for a population of yellow monkey flowers, *Mimulus guttatus*. The dataset comes from Lowry and Willis (2010) and the study in which the data is obtained investigates the role of chromosomal inversions in adaptation and speciation. Phenotypic traits and covariates are recorded for 2313 *M. guttatus*. Complete the following:

part a [10 points]: These plants either survive or die. Conditional on survival these plants then reproduce. Reproduction is taken to be Darwinian fitness All surviving plants have non-zero reproduction. Write out the graphical structure corresponding to a life history of *M. guttatus*. Include the distributions specifying the relationships between nodes in the graph.

These plants either survive or die implying the mortality status is a Bernoulli random variable. All surviving plants, given survived, have non-zero reproduction implying the number of off springs follow zero-truncated Poisson distribution.

The figure below shows the graphical model for a few individuals. Arrows go from parent nodes to child nodes. Nodes are labeled by their associated variables. The only root node is associated with the constant variable j. M_j is the mortality status in the particular year(1 implies survival). H_j is the number of off springs. The M_j is a Bernoulli random variable and the H_j is a zero-truncated Poisson conditional on its parent variables (M_j) being one, and zero otherwise.



part b [10 points]: Load in the `Mguttatus` data set (the `Mguttatus.rda` file located in the same directory as this assignment is the relevant data set) and copy the code:

```
data = Mguttatus
redata = Mguttatus.redata
vars = quantities$vars
pred = quantities$pred
group = quantities$group
code = quantities$code
fam = quantities$fam
nnode = length(vars)
n = nrow(redata) / nnode
families = quantities$families
root = redata$root
fit = redata$fit
```

```
varvar = quantities$varvar
idvar = quantities$idvar
```

```
library(aster)
load("Mguttatus.rda")
data = Mguttatus
redata = Mguttatus.redata
vars = quantities$vars
pred = quantities$pred
group = quantities$group
code = quantities$code
fam = quantities$fam
nnode = length(vars)
n = nrow(redata) / nnode
families = quantities$families
root = redata$root
fit = redata$fit
varvar = quantities$varvar
idvar = quantities$idvar
```

part c [20 points]: Fit the main effects only aster model including terms for genetic background (`gen_bac`), site (`site`), inversion (`inversion`), and type (`type`). Note that your call to `aster` will require you to specify the following arguments: `varvar = varvar, idvar = idvar, data = redata, root = root`, i.e. the hard work has already been done for you.

```
aout = aster(resp ~ varb + fit : (gen_bac + site + inversion + type),
             pred, fam, varb, id, root, data = redata)
summary(aout)
```

```
##
## Call:
## aster.formula(formula = resp ~ varb + fit:(gen_bac + site + inversion +
##     type), pred = pred, fam = fam, varvar = varb, idvar = id,
##     root = root, data = redata)
##
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -10.605380   0.118662 -89.374  < 2e-16 ***
## varbflws          13.110165   0.127665 102.692  < 2e-16 ***
## fit:gen_bacCoast  -0.092482   0.015091  -6.128 8.88e-10 ***
## fit:siteFraser    -0.102580   0.008118 -12.637  < 2e-16 ***
## fit:inversionP    -0.079404   0.015020  -5.286 1.25e-07 ***
## fit:typeL         -0.007532   0.019570  -0.385   0.7003
## fit:typeLN        -0.040742   0.019752  -2.063   0.0391 *
## fit:typeS          0.036587   0.015636   2.340   0.0193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Original predictor variables dropped (aliased)
##      fit:gen_bacInland
##      fit:typeSI1
##      fit:typeSN1
```

All variables except fit:typeL are significant.

part d [20 points]: Obtain estimates of expected Darwinian fitness for all unique factor-level combinations of the variables used to fit the main effects only aster model. The code below will come in handy for creating the unique factor-level combinations:

```
test = data$redata
a = levels(test$gen_bac)
b = levels(test$site)
c = levels(test$inversion)
d = levels(test$type)
fred = expand.grid(a = a, b = b, c = c, d = d)
colnames(fred) = c("gen_bac","site","inversion","type")
fred$sur_flw = 1
fred$flws = 1
```

```
test = data$redata
a = levels(test$gen_bac)
b = levels(test$site)
c = levels(test$inversion)
d = levels(test$type)
fred = expand.grid(a = a, b = b, c = c, d = d)
colnames(fred) = c("gen_bac","site","inversion","type")
fred$sur_flw = 1
fred$flws = 1
renewdata = reshape(fred, varying = list(vars),
                    direction = "long", timevar = "varb",
                    times = as.factor(vars), v.names = "resp")
layer=as.character(renewdata$varb)
fit = as.numeric(layer == "flws")
renewdata = data.frame(renewdata, fit = fit)
renewdata$root=1

pout = predict(aout, newdata = renewdata, varvar = varb,
               idvar = id, root = root, se.fit = TRUE)
predout=pout$fit
predout.se = matrix(pout$se.fit, ncol = nnode)
result = cbind(predout[1:48], predout.se[1:48],predout[49:96], predout.se[49:96])
rownames(result) = unique(as.character(renewdata$pop))
colnames(result) = c("sur_flw estimates", "sur_flw std. err.","flws estimates", "flws std. err.")
result
```

```
##       sur_flw estimates sur_flw std. err. flws estimates flws std. err.
##  [1,]        0.50322639        0.02786252      5.6158978      0.3618717
##  [2,]        0.74917363        0.04186672      9.1706185      0.6755315
##  [3,]        0.25442996        0.02055059      2.5626312      0.2327964
##  [4,]        0.47521466        0.05043862      5.2500168      0.6510677
##  [5,]        0.30175265        0.02605362      3.1105016      0.3035162
##  [6,]        0.53986795        0.02846405      6.1041074      0.3781169
##  [7,]        0.13658198        0.01422659      1.2707138      0.1481741
##  [8,]        0.28038814        0.02207805      2.8612440      0.2539342
##  [9,]        0.48229782        0.04845805      5.3419565      0.6268565
## [10,]        0.73152083        0.02231544      8.8873395      0.3447297
## [11,]        0.24036030        0.03272830      2.4027620      0.3689399
## [12,]        0.45459407        0.02577096      4.9845269      0.3244786
```

```
## [13,]       0.28570853       0.05903694      2.9230243      0.6854428
## [14,]       0.51873241       0.04910372      5.8211287      0.6492575
## [15,]       0.12855569       0.02966499      1.1870725      0.3075554
## [16,]       0.26520014       0.03519703      2.6859567      0.4028628
## [17,]       0.39349968       0.04513993      4.2160895      0.5556472
## [18,]       0.64691721       0.02566811      7.6027788      0.3667307
## [19,]       0.18584600       0.02660990      1.7971583      0.2892014
## [20,]       0.36812294       0.02448879      3.9045750      0.2950417
## [21,]       0.22261408       0.04858058      2.2031573      0.5424949
## [22,]       0.42761624       0.04690578      4.6419241      0.5884120
## [23,]       0.09837526       0.02285287      0.8787477      0.2292266
## [24,]       0.20588091       0.02897085      2.0170915      0.3193802
## [25,]       0.60558029       0.05137523      7.0099520      0.7217066
## [26,]       0.82496670       0.05341727     10.4746959      0.9812265
## [27,]       0.33185803       0.04385439      3.4670019      0.5224289
## [28,]       0.57748689       0.08151125      6.6175966      1.1252660
## [29,]       0.38821105       0.02752736      4.1508040      0.3358660
## [30,]       0.64133211       0.05034946      7.5215162      0.7274211
## [31,]       0.18283162       0.01699646      1.7643456      0.1838005
## [32,]       0.36302286       0.04629693      3.8424990      0.5611582
## [33,]       0.50322639       0.02786252      5.6158978      0.3618717
## [34,]       0.74917363       0.04186672      9.1706185      0.6755315
## [35,]       0.25442996       0.02055059      2.5626312      0.2327964
## [36,]       0.47521466       0.05043862      5.2500168      0.6510677
## [37,]       0.30175265       0.02605362      3.1105016      0.3035162
## [38,]       0.53986795       0.02846405      6.1041074      0.3781169
## [39,]       0.13658198       0.01422659      1.2707138      0.1481741
## [40,]       0.28038814       0.02207805      2.8612440      0.2539342
## [41,]       0.50322639       0.02786252      5.6158978      0.3618717
## [42,]       0.74917363       0.04186672      9.1706185      0.6755315
## [43,]       0.25442996       0.02055059      2.5626312      0.2327964
## [44,]       0.47521466       0.05043862      5.2500168      0.6510677
## [45,]       0.30175265       0.02605362      3.1105016      0.3035162
## [46,]       0.53986795       0.02846405      6.1041074      0.3781169
## [47,]       0.13658198       0.01422659      1.2707138      0.1481741
## [48,]       0.28038814       0.02207805      2.8612440      0.2539342
```

part e [10 points]: Report which 5 combinations of phenotypic traits are estimated to have the highest expected Darwinian fitness, and report which 5 combinations of phenotypic traits are estimated to have the lowest expected Darwinian fitness.

```
fitness=pout$fit[49:96]
highest=tail(order(fitness), 5)
lowest=head(order(fitness), 5)

#highest expected Darwinian fitness
high_comb=data.frame(fred[highest,])
high_comb
```

```
##    gen_bac site inversion type sur_flw flws
## 10  Inland Boon         A    L       1    1
## 2   Inland Boon         A  IL1       1    1
## 34  Inland Boon         A  SI1       1    1
```

```
## 42  Inland Boon         A  SN1       1    1
## 26  Inland Boon         A    S       1    1
```

```
#lowest expected Darwinian fitness
low_comb=data.frame(fred[lowest,])
low_comb
```

```
##    gen_bac   site inversion type sur_flw flws
## 23   Coast Fraser         P   LN       1    1
## 15   Coast Fraser         P    L       1    1
## 7    Coast Fraser         P  IL1       1    1
## 39   Coast Fraser         P  SI1       1    1
## 47   Coast Fraser         P  SN1       1    1
```

part f [10 points]: Read through Lowry and Willis (2010) and briefly explain in your own words reasons for why these researchers chose to study *M. guttatus* in their analysis.

The yellow monkeyflower (Mimulus guttatus) is a good genetic model system to explore whether inversions play a role in habitat-mediated adaptation and ecological reproductive isolation. Widespread inland annual (Mediterranean habitats) and coastal perennial ecotypes of M. guttatus have been shown to be locally adapted to their contrasting environments, leading to their reproductive isolation as a result of robust ecological prezygotic reproductive isolating barriers. A significant amount of the local adaptation and reproductive isolation in this system results from an adaptive flowering time shift, which is reinforced by both temporal isolation and selection against immigrants between habitats. In inland annual habitats, transplanted late-flowering coastal perennial plants fail to flower before the onset of the hot yearly summer drought. On the other hand, in a coastal ecosystem, early flowering inland annual plants are at a disadvantage. A complex genetic architecture, comprising a few large-effect quantitative trait loci (QTL), controls this life-history transition involving growth and reproduction. One of those large-effect loci appears to be situated in an area of linkage group eight with an abnormally high number of completely linked markers, suggesting the possible participation of a chromosomal rearrangement.

Hence, Lowry and Willis chose to study Mimulus guttatus (M. guttatus), a species, which is found in diverse habitats and exhibits considerable genetic variation.

Problem 2 [20 points]: In the Aster model notes it was stated that the system of equations

$$\phi_j = \theta_j - \sum_{G \in p^{-1}(\{j\})} c_G(\theta_G), \qquad j \in J,$$

determine an invertible change of parameters. Show this. There are linked references in the Aster model notes that may be helpful.

We have

$$\phi_j = \theta_j - \sum_{G \in p^{-1}(\{j\})} c_G(\theta_G), \qquad j \in J,$$

$$\implies \theta_j = \phi_j + \sum_{G \in p^{-1}(\{j\})} c_G(\theta_G), \qquad j \in J,$$

Now for terminal nodes, we have $\theta_j = \phi_j$ because terminal node can not be the predecessor of any nodes and so, $p^{-1}(\{j\}) = \phi$. Now once we know θ_j for the terminal nodes, we will move on to the nodes whose successors are terminal nodes. The equation for them would look like :

$$\implies \theta_j = \phi_j + c_T(\theta_T), \qquad j \in J_0,$$

where J_0 are nodes whose successors are terminal nodes and θ_T is terminal node. Since $\theta_T = \phi_T$, we can easily calculate θ_j from here. Next after calculating $\theta_j \ \forall j \in J_0$ we will move on to those nodes whose succesors are J_0 and so on. Hence, in this manner we can calculate $\theta_j \quad \forall \quad j \in J$, proving that $\phi \quad$ to $\quad \theta$ determine an invertible change of parameters.