**Article:**

Brooks, ME, Mugabo, M, Rodgers, GM et al. (2 more authors) (2016) How well can body size represent effects of the environment on demographic rates? Disentangling correlated explanatory variables. Journal of Animal Ecology, 85. pp. 318-328. ISSN 0021-8790

https://doi.org/10.1111/1365-2656.12465

# Appendix S4: Decomposing Explanatory Power and Drawing Euler Diagrams

Mollie Brooks, Marianne Mugabo, Gwendolen M. Rodgers,
Timothy G. Benton, Arpat Ozgul

April 9, 2015

## Contents

**1 Procedures** **1**

**2 Preliminaries** **2**

**3 Growth** **2**

**4 Transition** **4**

**5 Probability of Reproducing** **7**

**6 Egg Counts, Given Reproduction** **10**

**7 Version Info** **12**

## 1 Procedures

Fit models predicting the demographic rate using body size, current environment, and a spline of the day by treatment. Fit models that contain all subsets of predictors. Calculate how much overlap there is in the explanatory power by calculating the difference from what the null deviance explained would be if it were addative. Plot the decomposition using eulerAPE (http://www.eulerdiagrams.org/eulerAPE/).

Among these models without random effects and stage effects, find the best subset of predictors using AIC.

## 2 Preliminaries

```
library(reshape)
library(mgcv)
library(bbmle)
```

```
load("/Users/molliebrooks1/Desktop/postdoc/mites/environ dept vital rates/Individual
```

Specify numbers of knots for the splines.

```
ke=10 #environment
kd=10 #days
kl=5 #length
```

## 3 Growth

Supported predictors based on AICc included stage, body size, current environment, and treatment history.

```
temp=na.omit(subset(Indiv, stage!="egg")[,
        c("l0.s", "l24", "density.s", "supply.s", "day", "trt", "l0")])
```

```
gm_full=gam(l24~s(l0.s, k=kl)+s(density.s, supply.s, k=ke)+trt+s(day, by=trt, k=kd),
        data=temp)
gm_es=gam(l24~s(l0.s, k=kl)+s(density.s, supply.s, k=ke), data=temp)
gm_eh=gam(l24~s(density.s, supply.s, k=ke)+trt+s(day, by=trt, k=kd), data=temp)
gm_sh=gam(l24~s(l0.s, k=kl)+trt+s(day, by=trt, k=kd), data=temp)
gm_s=gam(l24~s(l0.s, k=kl), data=temp)
gm_e=gam(l24~s(density.s, supply.s, k=ke), data=temp)
gm_h=gam(l24~trt+s(day, by=trt, k=kd), data=temp)
AICctab(gm_full, gm_es, gm_eh, gm_sh, gm_s, gm_e, gm_h, nobs=nrow(temp))

##         dAICc   df
## gm_full     0.0 33.7054747521038
## gm_sh       3.5 27.7460671645121
## gm_es      44.9 14.4690816943962
```

```
## gm_s      118.0 5.89627889855371
## gm_h    10180.7 14.0009642367018
## gm_eh   10183.6 13.8932092018417
## gm_e    10194.4 9.16777097284889
```

Extract null deviance explained from each model.

```
(R.sq.full=summary(gm_full)[["dev.expl"]])

## [1] 0.9928

(Rsq_es=summary(gm_es)[["dev.expl"]])

## [1] 0.9925

(Rsq_eh=summary(gm_eh)[["dev.expl"]])

## [1] 0.0383

(Rsq_sh=summary(gm_sh)[["dev.expl"]])

## [1] 0.9928

(Rsq_s=summary(gm_s)[["dev.expl"]])

## [1] 0.9922

(Rsq_e=summary(gm_e)[["dev.expl"]])

## [1] 0.02888

(Rsq_h=summary(gm_h)[["dev.expl"]])

## [1] 0.03972
```

Calculate areas of intersections.

```
#//s | e | h | se | sh | eh | seh
areas=data.frame(t(c(Rsq_s, Rsq_e, Rsq_h,
                        Rsq_s+ Rsq_e-Rsq_es,
                        Rsq_s+ Rsq_h-Rsq_sh,
```

```
                        Rsq_h+ Rsq_e-Rsq_eh,
                        Rsq_s+Rsq_e+Rsq_h-Rsq_es-Rsq_eh-Rsq_sh+R.sq.full)))
names(areas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
areas

##        s       e       h      se      sh      eh     seh
## 1 0.9922 0.02888 0.03972 0.02855 0.03915 0.03029 0.03002

with(areas, c(se/e,sh/h))

## [1] 0.9886 0.9856
```

Calculate non-overlapping subareas.

```
subareas=with(areas, t(c(
                        s-se-sh+ seh, e-eh-se+ seh, h-eh-sh+ seh,
                        se-seh,
                        sh-seh,
                        eh-seh,
                        seh)))
colnames(subareas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
subareas

##           s         e         h        se       sh       eh     seh
## [1,] 0.9545 5.452e-05 0.0002956 -0.001471 0.009129 0.000276 0.03002
```

Not all subareas are positive. Set the negative one to 1e-5 (basically 0). Plot the Euler diagrams.

```
subareas[4]=1e-5
write.table(subareas,
                        file="growth.els", sep=" | ", eol="",
                        row.names=FALSE, col.names=FALSE)
system("java -jar eulerAPE_3.0.0.jar -i growth.els -l no")
```

# 4 Transition

Supported predictors based on AIC included stage, body size, current environment, and treatent history. Subset data to only include the juvenile survivors.

```
tempT=transform(subset(Indiv, (!stage %in% c("male","female") & (s24==1))),
        t.init.24=as.numeric(t24 %in% c("next stage", "quiescent")),
        t.done.24=as.numeric(t24 == "next stage"),
        stage=stage[, drop=TRUE]
        )
```

```
temp=na.omit(tempT[,c("t.init.24","l0.s","trt","day","density.s", "supply.s")])
```

```
t24m_full=gam(t.init.24~s(l0.s, k=kl)+trt+s(day, by=trt, k=kd)+
        s(density.s, supply.s, k=ke), family=binomial, data=temp)
t24m_es=gam(t.init.24~s(l0.s, k=kl)+s(density.s, supply.s, k=ke),
        family=binomial, data=temp)
t24m_eh=gam(t.init.24~s(density.s, supply.s, k=ke)+trt+s(day, by=trt, k=kd),
        family=binomial, data=temp)
t24m_sh=gam(t.init.24~s(l0.s, k=kl)+trt+s(day, by=trt, k=kd), family=binomial, data=t
t24m_s=gam(t.init.24~s(l0.s, k=kl), family=binomial, data=temp)
t24m_e=gam(t.init.24~s(density.s, supply.s, k=ke), family=binomial, data=temp)
t24m_h=gam(t.init.24~trt+s(day, by=trt, k=kd), family=binomial, data=temp)

AICctab(t24m_full, t24m_es, t24m_eh, t24m_sh,
        t24m_s, t24m_e, t24m_h, nobs=nrow(temp))

##            dAICc df
## t24m_es     0.0 12.7619496130163
## t24m_sh     3.4 19.5221057878456
## t24m_full   4.0 19.993975699977
## t24m_s     71.6 4.59845505425022
## t24m_e    117.8 9.20833060777449
## t24m_h    121.0 16.8994159810501
## t24m_eh   125.8 16.11825755538
```

Extract null deviance explained from each model.

```
(R.sq.full=summary(t24m_full)[["dev.expl"]])

## [1] 0.2806

(Rsq_es=summary(t24m_es)[["dev.expl"]])
```

```
## [1] 0.2686

(Rsq_eh=summary(t24m_eh)[["dev.expl"]])

## [1] 0.1384

(Rsq_sh=summary(t24m_sh)[["dev.expl"]])

## [1] 0.2802

(Rsq_s=summary(t24m_s)[["dev.expl"]])

## [1] 0.1719

(Rsq_e=summary(t24m_e)[["dev.expl"]])

## [1] 0.1316

(Rsq_h=summary(t24m_h)[["dev.expl"]])

## [1] 0.1454
```

Calculate areas of intersections.

```
#//s | e | h | se | sh | eh | seh
areas=data.frame(t(c(Rsq_s, Rsq_e, Rsq_h,
                        Rsq_s+ Rsq_e-Rsq_es,
                        Rsq_s+ Rsq_h-Rsq_sh,
                        Rsq_h+ Rsq_e-Rsq_eh,
                        Rsq_s+Rsq_e+Rsq_h-Rsq_es-Rsq_eh-Rsq_sh+R.sq.full)))
names(areas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
areas

##        s      e      h     se      sh     eh     seh
## 1 0.1719 0.1316 0.1454 0.0349 0.03712 0.1385 0.04222

with(areas, c(se/e,sh/h))

## [1] 0.2653 0.2553
```

Calculate non-overlapping subareas.

```
subareas=with(areas, t(c(
                        s-se-sh+ seh, e-eh-se+ seh, h-eh-sh+ seh,
                        se-seh,
                        sh-seh,
                        eh-seh,
                        seh)))
colnames(subareas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
subareas

##            s         e       h        se        sh     eh     seh
## [1,] 0.1421 0.0003602 0.01198 -0.007312 -0.005095 0.0963 0.04222
```

Not all subareas are positive. Set the negative ones to 1e-5 (basically 0). Plot the Euler diagrams.

```
subareas[4:5]=1e-5
write.table(subareas,
                        file="trans.els", sep=" | ", eol="",
                        row.names=FALSE, col.names=FALSE)
system("java -jar eulerAPE_3.0.0.jar -i trans.els -l no")
```

## 5 Probability of Reproducing

Supported predictors based on AIC included body size, current environmen, and past environmentt. Remove ones that were dead at 24hrs. Use females only.

```
Indivtmp=subset(Indiv, stage=="female" & !is.na(s24))
```

```
temp=subset(Indivtmp, !is.na(l0.s*as.numeric(trt)*f24) & (s24==1))[
        ,c("f24","l0.s","trt","day","density.s", "supply.s")]
```

```
fbm_full=gam(as.numeric(f24>0)~s(l0.s, k=kl)+s(density.s, supply.s, k=ke)+
        trt+s(day, by=trt, k=kd), family=binomial, data=temp)
fbm_es=gam(as.numeric(f24>0)~s(l0.s, k=kl)+s(density.s, supply.s, k=ke),
        family=binomial, data=temp)
```

```
fbm_eh=gam(as.numeric(f24>0)~s(density.s, supply.s, k=ke)+trt+s(day, by=trt, k=kd),
        family=binomial, data=temp)
fbm_sh=gam(as.numeric(f24>0)~s(l0.s, k=kl)+trt+s(day, by=trt, k=kd),
        family=binomial, data=temp)
fbm_s=gam(as.numeric(f24>0)~s(l0.s, k=kl), family=binomial, data=temp)
fbm_e=gam(as.numeric(f24>0)~s(density.s, supply.s, k=ke), family=binomial, data=temp)
fbm_h=gam(as.numeric(f24>0)~trt+s(day, by=trt, k=kd), family=binomial, data=temp)
AICctab(fbm_full, fbm_es, fbm_eh, fbm_sh, fbm_s, fbm_e, fbm_h, nobs=nrow(temp))
```

```
##           dAICc df
## fbm_sh      0.0 20.2495173856494
## fbm_full    2.6 24.0978468187126
## fbm_h      18.1 17.497202388451
## fbm_eh     19.4 21.5804417096759
## fbm_es     22.4 12.7470404335229
## fbm_e      41.1 9.40038156380709
## fbm_s     153.2 4.85877433485277
```

Extract null deviance explained from each model.

```
(R.sq.full=summary(fbm_full)[["dev.expl"]])
```

```
## [1] 0.4603
```

```
(Rsq_es=summary(fbm_es)[["dev.expl"]])
```

```
## [1] 0.395
```

```
(Rsq_eh=summary(fbm_eh)[["dev.expl"]])
```

```
## [1] 0.4274
```

```
(Rsq_sh=summary(fbm_sh)[["dev.expl"]])
```

```
## [1] 0.4519
```

```
(Rsq_s=summary(fbm_s)[["dev.expl"]])
```

```
## [1] 0.1765
```

```
(Rsq_e=summary(fbm_e)[["dev.expl"]])
```

```
## [1] 0.3569
```

```
(Rsq_h=summary(fbm_h)[["dev.expl"]])
```

```
## [1] 0.4162
```

Calculate areas of intersections.

```
#//s | e | h | se | sh | eh | seh
areas=data.frame(t(c(Rsq_s, Rsq_e, Rsq_h,
                        Rsq_s+ Rsq_e-Rsq_es,
                        Rsq_s+ Rsq_h-Rsq_sh,
                        Rsq_h+ Rsq_e-Rsq_eh,
                        Rsq_s+Rsq_e+Rsq_h-Rsq_es-Rsq_eh-Rsq_sh+R.sq.full)))
names(areas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
areas

##        s      e      h     se     sh     eh    seh
## 1 0.1765 0.3569 0.4162 0.1384 0.1408 0.3457 0.1356

with(areas, c(se/e,sh/h))

## [1] 0.3877 0.3383
```

Calculate non-overlapping subareas.

```
subareas=with(areas, t(c(
                        s-se-sh+ seh, e-eh-se+ seh, h-eh-sh+ seh,
                        se-seh,
                        sh-seh,
                        eh-seh,
                        seh)))
colnames(subareas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
subareas

##            s        e       h       se       sh   eh    seh
## [1,] 0.03295 0.008474 0.06534 0.002708 0.005181 0.21 0.1356
```

All subareas are positive.
Plot the Euler diagrams.

```
write.table(subareas,
                        file="reprod.els", sep=" | ", eol="",
                        row.names=FALSE, col.names=FALSE)
system("java -jar eulerAPE_3.0.0.jar -i reprod.els -l no")
```

# 6 Egg Counts, Given Reproduction

Supported predictors based on AIC included body size and current environment.

```
temp=na.omit(subset(Indivtmp, (f24>0) & (s24==1))[
        ,c("f24","l0.s","trt","day","density.s", "supply.s")])
```

```
fcm_full=gam(f24-1~s(l0.s, k=kl)+s(density.s, supply.s, k=ke)+
        trt+s(day, by=trt, k=kd), family=poisson, data=temp)
fcm_es=gam(f24-1~s(l0.s, k=kl)+s(density.s, supply.s, k=ke),
        family=poisson, data=temp)
fcm_eh=gam(f24-1~s(density.s, supply.s, k=ke)+trt+s(day, by=trt, k=kd),
        family=poisson, data=temp)
fcm_sh=gam(f24-1~s(l0.s, k=kl)+trt+s(day, by=trt, k=kd),
        family=poisson, data=temp)
fcm_s=gam(f24-1~s(l0.s, k=kl), family=poisson, data=temp)
fcm_e=gam(f24-1~s(density.s, supply.s, k=ke), family=poisson, data=temp)
fcm_h=gam(f24-1~trt+s(day, by=trt, k=kd), family=poisson, data=temp)
AICctab(fcm_full, fcm_es, fcm_eh, fcm_sh, fcm_s, fcm_e, fcm_h, nobs=nrow(temp))

##          dAICc df
## fcm_full   0.0 30.6293823170902
## fcm_sh     9.2 29.149939843577
## fcm_es    45.8 13.3082508139272
## fcm_eh   109.9 32.5509132183938
## fcm_h    137.5 26.4667867263547
## fcm_e    264.8 9.74831903782391
## fcm_s    513.5 2.00063858206843
```

Extract null deviance explained from each model.

```
(R.sq.full=summary(fcm_full)[["dev.expl"]])

## [1] 0.6669

(Rsq_es=summary(fcm_es)[["dev.expl"]])

## [1] 0.6211
```

```
(Rsq_eh=summary(fcm_eh)[["dev.expl"]])

## [1] 0.6099

(Rsq_sh=summary(fcm_sh)[["dev.expl"]])

## [1] 0.6601

(Rsq_s=summary(fcm_s)[["dev.expl"]])

## [1] 0.3552

(Rsq_e=summary(fcm_e)[["dev.expl"]])

## [1] 0.4984

(Rsq_h=summary(fcm_h)[["dev.expl"]])

## [1] 0.5872
```

Calculate areas of intersections.

```
#//s | e | h | se | sh | eh | seh
#//s | e | h | se | sh | eh | seh
areas=data.frame(t(c(Rsq_s, Rsq_e, Rsq_h,
                        Rsq_s+ Rsq_e-Rsq_es,
                        Rsq_s+ Rsq_h-Rsq_sh,
                        Rsq_h+ Rsq_e-Rsq_eh,
                        Rsq_s+Rsq_e+Rsq_h-Rsq_es-Rsq_eh-Rsq_sh+R.sq.full)))
names(areas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
areas

##        s      e      h     se     sh     eh    seh
## 1 0.3552 0.4984 0.5872 0.2325 0.2823 0.4758 0.2167

with(areas, c(se/e,sh/h))

## [1] 0.4665 0.4807
```

Calculate non-overlapping subareas.

```
subareas=with(areas, t(c(
                        s-se-sh+ seh, e-eh-se+ seh, h-eh-sh+ seh,
                        se-seh,
                        sh-seh,
                        eh-seh,
                        seh)))
colnames(subareas)=c('s', 'e', 'h', 'se', 'sh', 'eh', 'seh')
subareas

##              s        e       h      se      sh     eh    seh
## [1,] 0.05703 0.006826 0.04583 0.01582 0.06561 0.2591 0.2167
```

All subareas are positive.

Plot the Euler diagrams.

```
write.table(subareas,
                file="egg_counts.els", sep=" | ", eol="",
                row.names=FALSE, col.names=FALSE)
system("java -jar eulerAPE_3.0.0.jar -i egg_counts.els -l no")
```

# 7 Version Info

```
sessionInfo()

## R version 3.1.0 (2014-04-10)
## Platform: x86_64-apple-darwin13.1.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] reshape_0.8.5 gamm4_0.2-2   mgcv_1.8-1    nlme_3.1-117  lme4_1.1-7
## [6] Rcpp_0.11.2   Matrix_1.1-3  bbmle_1.0.17  knitr_1.6
```

```
##
## loaded via a namespace (and not attached):
##  [1] codetools_0.2-8   digest_0.6.4      evaluate_0.5.5
##  [4] formatR_0.10      grid_3.1.0        highr_0.3
##  [7] lattice_0.20-29   MASS_7.3-31       minqa_1.2.3
## [10] nloptr_1.0.0      numDeriv_2012.9-1 plyr_1.8.1
## [13] splines_3.1.0     stringr_0.6.2     tools_3.1.0
```