

## An estimator of the Opportunity for Selection that is independent of mean fitness

### Supporting Information

#### SI 1. Underdispersed variance in reproductive success.

This paper has focused on overdispersed variance in reproductive success ( $s_{\bar{k}_1}^2 > \bar{k}_1$ , using the Poisson approximation), which reduces effective size and provides opportunities for natural selection to operate. Underdispersion ( $s_{\bar{k}_1}^2 < \bar{k}_1$ ) is less common but nevertheless occurs in many taxa, at least with respect to seasonal reproduction. For example, the number of eggs or early-juvenile offspring can be constrained to a relatively small range of values, and this limits how large  $s_{\bar{k}_1}^2$  can be. At the extreme, females of many large animal species, and some birds, can only produce 0 or 1 offspring per season, in which case it is impossible for the variance in offspring number to be as large as the mean. This can be shown as follows. Of the  $N$  females, assume that  $XN$  produce exactly 1 offspring that appears in a sample, and the rest produce none. Mean offspring number in the sample is  $\bar{k}_1 X$  and  $s_{\bar{k}_1}^2 = \text{mean of the squares} - \text{the square of the mean} = SS/N - (\bar{k}_1)^2 = XN/N - X^2 = X - X^2$ , which leads to  $\hat{\varphi}_1 = 1 - X$ .

The original Wright-Fisher model of reproduction incorporated an extreme version of this underdispersion scenario: each of  $N$  potential parents were imagined to contribute equally to an infinitely large pool of gametes, which then united at random to form the next generation of  $N$  individuals. Under this scenario, using the current notation,  $s_{\bar{k}_1}^2$  and  $\hat{\varphi}_1$  were both 0 while  $\bar{k}_1$  was infinitely large, and substituting these values into Eq. 1A produces

$$E(\hat{\varphi}_2) = \frac{N-1}{N} + \frac{\bar{k}_2}{\infty} \left(0 - \frac{N-1}{N}\right) = \frac{N-1}{N}, \quad (\text{A1})$$

which is simply  $E(\hat{\varphi}_{drift})$  (i.e., RS is neither overdispersed nor underdispersed). Note that because the initial gamete pool (and hence  $\bar{k}_1$ ) is assumed to be infinitely large, this result holds regardless what  $\bar{k}_2$  is.

All real populations are of course finite in size. Also, we can generalize the idea of an initial gamete pool to production of diploid offspring ranging from zygotes to adults. If we still assume equal contributions to the initial pool of offspring ( $s_{\bar{k}_1}^2 = 0$ ), but in finite amounts by each parent, we have

$$E(\hat{\varphi}_2) = \frac{N-1}{N} + \frac{\bar{k}_2}{X} \left(0 - \frac{N-1}{N}\right) = \frac{N-1}{N} \left(1 - \frac{\bar{k}_2}{X}\right), \quad (\text{A2})$$

where  $X = \bar{k}_1$  could be relatively large but is finite. Equation A2 shows that  $E(\hat{\varphi}_2)$  is always less than the random expectation, so variance in RS is underdispersed, and the degree of underdispersion depends on the ratio  $\bar{k}_2/X$ .

This model is still not very realistic, as it assumes 0 variance among parents in contributions to the initial gamete pool. Overdispersion can be generated by having unequal parental contributions to the initial gamete pool. Waples et al. (2018) used a variation of this approach to provide a minimum estimate of  $\hat{\varphi}_2$  in southern bluefin tuna (*Thunnus maccoyii*), based on empirical data for variation in female size at specific ages (and hence assumed variance in age-

specific egg production). For the current project, in modeling underdispersion I relaxed that restrictive assumption that  $s_{k_1}^2 = 0$  by having two separate processes for producing offspring: 1) a directed process, whereby each parent contributes equally to the fraction  $Y$  of all offspring; and 2) a random process, whereby parents for the remaining fraction  $(1-Y)$  of all offspring are chosen randomly from the  $N$  candidates. As in the main text, in the simulations 100 parents produced a total of 10,000 initial offspring (so  $\bar{k}_1 = 100$ ), and random subsampling produced  $\bar{k}_2$  values as small as 0.1. I considered two values of  $Y$ : 0.5 (moderate underdispersion) and 0.9 (strong underdispersion).

Results (Figure S1) show that initial Index of Variability for the two scenarios is simply the fraction of offspring that were randomly assigned to parents (i.e.,  $\hat{\phi}_1 = 1-Y$ ). As the sample size of offspring declines,  $\hat{\phi}_1$  for both scenarios rapidly approach 1, in agreement with predictions from Equation 1A. It is apparent that sparse sampling of offspring will generally have a poor chance of detecting underdispersed variance in reproductive success, even when underdispersion is pronounced.

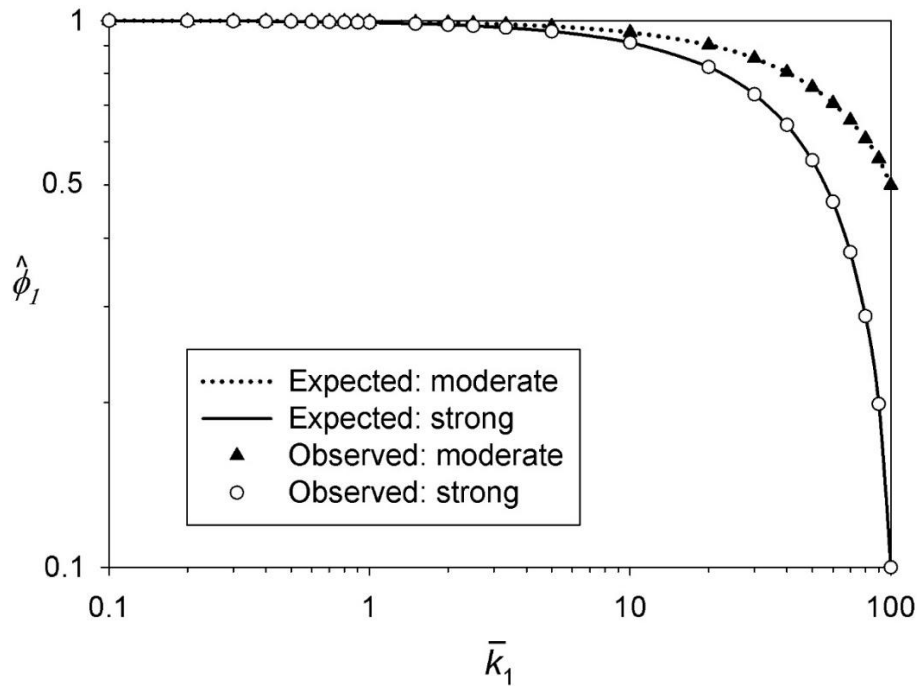


Figure S1. Relationship between the estimated Index of Variability ( $\hat{\phi}_1$ ) and sample  $\bar{k}_1$  when variance in reproductive success is underdispersed. Observed values (symbols) are from simulations that initially generated 10,000 offspring from 100 parents and then randomly subsampled offspring to produce smaller sample  $\bar{k}_1$  values. Expected values (lines) were obtained from Equation 1A using the empirical value of raw  $\hat{\phi}_1$  for maximum  $\bar{k}_1 = 100$ . Moderate and strong underdispersion scenarios are described in the Appendix text.

SI 2. Poisson approximation to the binomial variance.

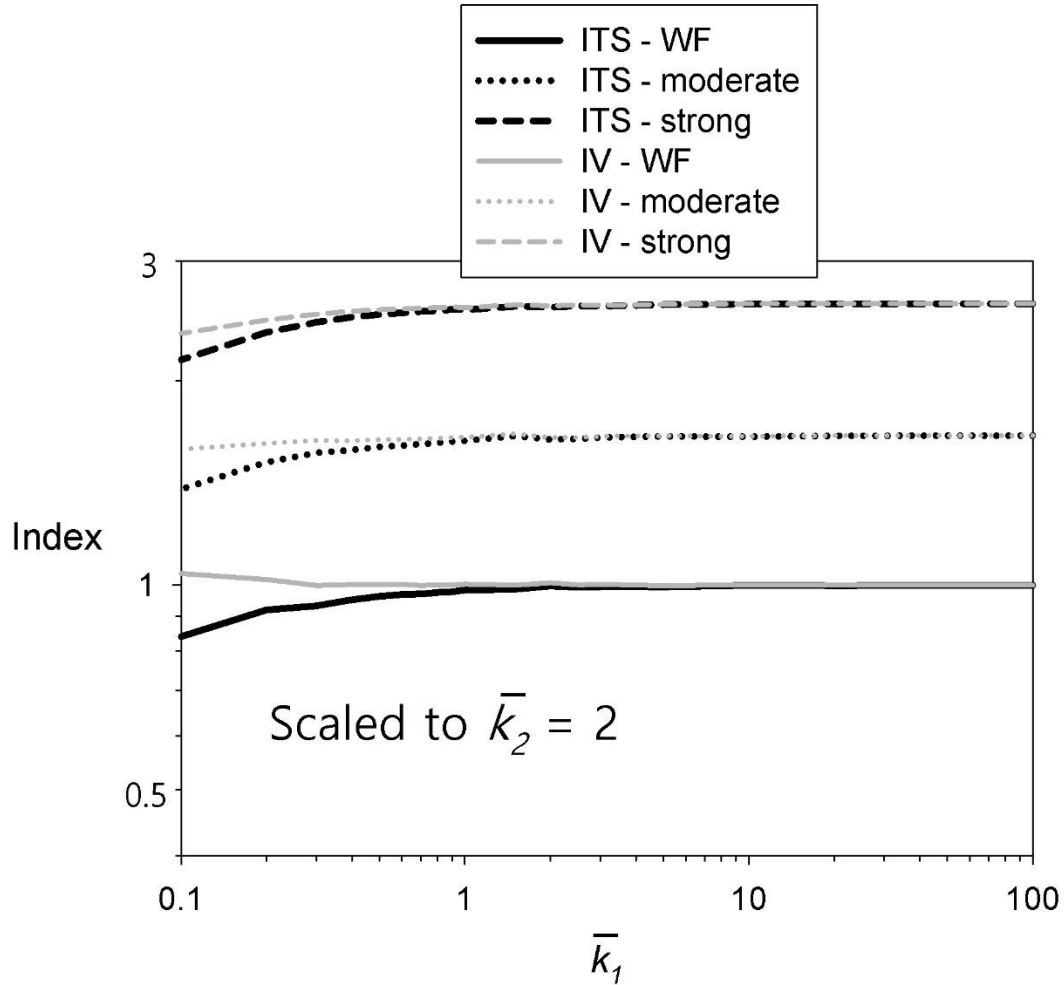


Figure S2. Comparison of results for rescaling empirical data for  $\hat{\varphi}_1$  from Figure 2A to  $\bar{k}_2 = 2$ , using Equation 1 (black) and Equation 1A (gray). The two equations differ in that 1 uses the Poisson approximation that  $E(\hat{\varphi}_{drift}) \approx 1$ , whereas 1A uses the exact value  $E(\hat{\varphi}_{drift}) = (N-1)/N$ , with  $N$  being the number of parents. The difference between  $(N-1)/N$  and 1 is small (0.99 vs 1 for the  $N = 100$  parents used in simulating the data), but that difference becomes magnified when  $\bar{k}_1$  is small, in which case the term  $\bar{k}_2/\bar{k}_1 = 2/\bar{k}_1$  becomes large.

### SI 3. R code used in the simulations, conducted with version 3.6.1.

```
#install.packages("matrixStats")
#install.packages("EnvStats")
library(matrixStats)
library(EnvStats)

a=as.numeric(Sys.time())
set.seed(a)

NReps1 = 1000
NReps2 = 100 ## extra reps for smaller sample sizes

### this version considers only female parents of each offspring
NPParents = 100
NOffspring = 100*NPParents ## this, compared to NPParents, determines initial mean offspring number = kbar1
Parents = 1:NPParents

A = rep(1,NPParents) ## equal weights = Wright-Fisher model
B = Parents ## moderately unequal weights for parents
C = Parents^2 ## more strongly unequal weights for parents
weights = C

kbar1 = NOffspring/NPParents
kbar2 = c(kbar1/2,kbar1/5,kbar1/10,kbar1/20,kbar1/30,kbar1/40,kbar1/50,1.5,1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1)
PartOff = kbar2*NPParents
BigOff = matrix(NA,NReps1*NReps2,NOffspring)
LittleOffA = list()
LittleOffB = list()
for (j in 1:9) {
  LittleOffA[[j]] = matrix(NA,NReps1,PartOff[j])
  LittleOffB[[j]] = matrix(NA,NReps1*NReps2,PartOff[j+9])
}

ITSA = matrix(NA,NReps1,10) ## Crow's Index of Total Selection = Var/mean^2
IVA = ITSA ## Crow's Index of Variability = Var/mean
ITSB = matrix(NA,NReps1*NReps2,9)
IVB = ITSB

for (k in 1:NReps1) {
  Offspring = sample(Parents,NOffspring,replace=T,prob=weights)
  BigOff[k,] = Offspring
  for (j in 1:9) {
    ## take a random subsample drawn from all offspring; equivalent to allowing random mortality
    LittleOffA[[j]][k,] = sample(Offspring,PartOff[j])
  } ## end for j
}
for (j in 1:9) {
  jj = j+9
  for (q in 1:NReps2) {
    kq = (k-1)*NReps2+q
    LittleOffB[[j]][kq,] = sample(Offspring,PartOff[jj])
  }
}
```

```

} # end for q
} ## end for j
} ## end for k

for (k in 1:NReps1) {

  X = BigOff[k,]
  T1 = as.vector(table(X))
  NZero = NParents - length(T1)
  addzero = rep(0,NZero)
  T1 = c(T1,addzero) ## add in parents with zero offspring
  ITSA[k,1] = var(T1)/mean(T1)^2
  IVA[k,1] = var(T1)/mean(T1)

for (j in 1:9) {
  jj = j+1
  Y = LittleOffA[[j]][k,]
  T2 = as.vector(table(Y))
  NZero2 = NParents - length(T2)
  addzero2 = rep(0,NZero2)
  T2 = c(T2,addzero2) ## add in parents with zero offspring
  ITSA[k,jj] = var(T2)/mean(T2)^2
  IVA[k,jj] = var(T2)/mean(T2)
} ## end for j

for (j in 1:9) {
  for (q in 1:NReps2) {
    kq = (k-1)*NReps2+q
    Y = LittleOffB[[j]][kq,]
    T2 = as.vector(table(Y))
    NZero2 = NParents - length(T2)
    addzero2 = rep(0,NZero2)
    T2 = c(T2,addzero2) ## add in parents with zero offspring
    ITSB[kq,j] = var(T2)/mean(T2)^2
    IVB[kq,j] = var(T2)/mean(T2)
  } # end for q
} ## end for j

} ## end for k

## get geometric means
R1 = 1:10
R2 = 1:10
R3 = 1:9
R4 = 1:9
for (i in 1:10) {
R1[i] = geoMean(ITSA[,i])
R2[i] = geoMean(IVA[,i])
} ## end for i

```

```
for (i in 1:9) {  
  R3[i] = geoMean(ITSB[,i])  
  R4[i] = geoMean(IVB[,i])  
} ## end for i  
  
X = c(R1,R3)  
Y = c(R2,R4)  
Z = rbind(X,Y)  
colnames(Z) = c(kbar1,kbar2)  
rownames(Z) = c("I", "Phi")  
Z
```