

**A Correction of Bias and Necessary Sampling Effort for  
Estimation of Relative Reproductive Success**

Prepared under the  
**Collaborative Systemwide Monitoring and Evaluation Project (CSMEP)**

Administered by the  
**COLUMBIA BASIN FISH & WILDLIFE FOUNDATION**

851 SW Sixth Avenue, Suite 260

Portland, Oregon 97204

Prepared by

WEST, Inc.  
2003 Central Avenue  
Cheyenne, Wyoming 82001  
Tel: 307-634-1756  
<http://www.west-inc.com/>

Dr. Lyman McDonald, Project Supervisor

[lmcdonald@west-inc.com](mailto:lmcdonald@west-inc.com)

James D. Griswold, Statistician

[jgriswold@west-inc.com](mailto:jgriswold@west-inc.com)

25 July 2008

## Introduction

Current methods for estimation of Relative Reproductive Success are biased. A basic and apparently unsolved problem exists in the estimation of Relative Reproductive Success (RRS) (Chris Beasley, Quantitative Consultants, and Michael Ford, NOAA Fisheries, personal communications). RRS is defined as the ratio of offspring per hatchery parent spawning naturally to offspring per wild (natural spawned) parent spawning naturally (Hinrichsen 2003, Araki et al. 2008). Table 1 presents the form of a simple hypothetical data set for which one may wish to estimate the RRS. The problem stems from the fact that, e.g., a hatchery male (#2 in Table 1) may be known to have been on the spawning ground, because he was released above a weir or his carcass was seen on the spawning ground, however no offspring are observed in samples of par, smolt, or returning adults. If the zero offspring for male #2 is left in the data set, the results are biased negatively for the fitness of hatchery males, because offspring could have been produced but none were observed in the samples of par, smolt, or returning adults. If the zero for male #2 is removed from the data set, the results are biased positively for the fitness of hatchery males, because the average fitness of hatchery males with observed offspring is being estimated and there are likely some hatchery males that did fail to produce offspring.

Table 1. Matrix of data of interest for RRS studies.

sample	Sex of Parent M(0) F(1)	offspring	origin	age	weight
1	M	4	H	3	14
2	M	0	H	4	16
3	M	6	W	4	18
4	M	2	W	4	22
5	F	12	H	4	24
6	F	0	H	5	28
7	F	16	W	4	25
8	F	22	W	5	28
...					

As it is likely that data will contain substantial zero offspring cases for a given parent (male, female, hatchery or wild) it is crucial to explicitly model this portion of the population to avoid biases in the estimation of fitness.

## A correction for bias in estimation of RRS

This method, conceived by Dr. Brian Manly, WEST, Inc., depends on the assumption or verification that an appropriate model is available for the probability distribution of the 'true' number of offspring for a given parent. These discrete probabilities are denoted  $P_0, P_1, P_2, \dots, P_n$ , for the probabilities of 0, 1, 2, ...  $n$  offspring for a given parent. We chose to first implement the Poisson distribution with inflated zeroes to model the number of offspring (Johnson, Kotz, and Kemp, 1992). The Poisson model by itself is appropriate for counts greater than zero for this type of data in most cases, however the zero inflated Poisson allows for an excess of zeros followed by the standard Poisson for positive counts. Other models such as the negative binomial are candidates, and tests for the goodness of fit of the basic model to be used are necessary parts of the analysis. At this point in time, only the zero inflated Poisson model is programmed into the R computer software programs in the Appendix 1. Other models that can be considered and programmed, with tests for which is appropriate, are the Poisson, negative binomial, and the zero inflated negative binomial.

Let  $Q$  be the sampling probability of the offspring (assumed known or estimated). For example,  $Q$  may be equal to 1.0 if all returning adults are observed at a weir leading to the spawning ground, or  $Q$  may be estimated as the proportion of smolts sampled in screw traps during out migration. The probability of observing 0 offspring for a given parent is the probability of 0 offspring, plus the probability of 1 offspring times the probability of not observing the offspring in the sample, plus the probability of 2 offspring times the probability of not observing either of the two in the sample, ..., etc., ... The sum is continued until the terms are sufficiently small. In symbols, this is written as

$$(1) \text{ Prob}(0) = P_0 + P_1(1-Q) + P_2(1-Q)^2 + P_3(1-Q)^3 + \dots$$

Similarly, the probability of observing 1 offspring is

$$\text{Prob}(1) = P_1Q + 2P_2Q(1-Q) + 3P_3Q(1-Q)^2 + \dots, \text{ and the probability of } i \text{ offspring is}$$

$$\text{Prob}(i) = P_iQ^i + P_{i+1}\binom{i+1}{i}Q^i(1-Q) + P_{i+2}\binom{i+2}{i}Q^i(1-Q)^2 + \dots, i = 0, 1, 2, 3, \dots,$$

where  $\binom{i+n}{i}$  denotes the number of ways  $i+n$  items can be taken  $i$  at a time.

In the zero- inflated Poisson model, the  $P_i$  are written as

$$P_i = 1 - \pi + \pi e^{-\lambda} \text{ when } i = 0 \text{ and } P_i = \pi \frac{e^{-\lambda} \lambda^i}{i!}, \text{ when } i > 0, \text{ where } 1 - \pi \text{ is the probability of zero}$$

offspring and  $\pi$  is the probability of observing at least one offspring when excess zeros come from a different generating process than non-zero counts and  $\lambda$  is the mean of the non-zero counts.

The values of  $i$  are the observed number of offspring corrected for the fact that only Q(100)% are observed in the samples. The number of offspring observed,  $o$ , for a given parent is expanded by multiplication by the factor  $\frac{1}{Q}$  to estimate  $i$ , i.e.,  $i = o/Q$ , for a given parent. The likelihood (equation 2) of the data is then written as the product of the probabilities of the observed numbers, the  $\text{Pr ob}(i)$ , and is then maximized to estimate the zero inflation Poisson parameters, ( $\pi$ , and  $\lambda$ ) that give  $P_0, P_1, P_2, \dots, P_n$  for an adult parent. The function 'optim' in the statistical software program R is used to numerically maximize the likelihood.

$$(2) L(P_0, P_1, \dots, P_n) = \prod_{i=1}^n \text{Pr ob}(i).$$

Having obtained the maximum likelihood estimates of  $\pi$  and  $\lambda$  and the probabilities of occurrence of 0, 1, 2, ... offspring,  $P_0, P_1, P_2, \dots, P_n$ , the expected number of offspring can then be calculated as the sum of the probabilities of occurrence times the number of offspring. For example, the probabilities would be estimated for a hatchery ( $H$ ) female ( $F$ ) parent and the expected number of offspring estimated by:

(3)

$$E(\text{number of offspring} | H, F) = 0 \cdot \text{Pr ob}_{hf}(0) + 1 \cdot \text{Pr ob}_{hf}(1) + 2 \cdot \text{Pr ob}_{hf}(2) + \dots + i \cdot \text{Pr ob}_{hf}(i) + \dots$$

For a wild ( $W$ ) female ( $F$ ) parent the probabilities would be re-estimated and the expected number of offspring would be:

$$E(\text{number of offspring} | W, F) = 0 \cdot \text{Pr ob}_{wf}(0) + 1 \cdot \text{Pr ob}_{wf}(1) + 2 \cdot \text{Pr ob}_{wf}(2) + \dots + i \cdot \text{Pr ob}_{wf}(i) + \dots$$

An estimate of RRS for hatchery females relative to wild females is then calculated as the ratio,

$$(4) RRS_{hf/wf} = \frac{E(\text{number of offspring} | H, F)}{E(\text{number of offspring} | W, F)}.$$

Confidence intervals and standard errors for estimated RRS can be obtained by re-sampling procedures implemented in the R program, 'manly.main' via the R boot and boot.ci functions as shown in Appendix 1a. If sufficient data are available, this program can be used to estimate RRS for other subsets of the data. That is other attributes of the parent influencing RRS such as age and weight might be considered. An example might be the RRS of age 4 hatchery males and age 4 wild males confined to a certain weight class.

### **Determination of Adequate Sampling Effort**

Three sources of sampling variation can be identified in relative reproductive success (RRS) studies which are measured at the juvenile life stage; adult enumeration, juvenile enumeration and genotyping error (Galbreath et al 2008). A critical need exists for information on precision and confidence intervals for RRS or statistical power in RRS experiments. In this paper, we consider the effect of different levels of smolt sampling intensity on acceptable levels of confidence in the assessment of significance of estimated RRS different from 1. For this stage of the work it has been assumed that adult enumeration is complete with 100 percent of adult escapement genetically sampled. It is also assumed that the assignment of offspring to their respective parents is without error.

Studies without a sufficient sample size, i.e.,  $Q$  is too small, will result in a failure to detect a significant effect when it exists; however, there is often a high cost associated with field collection and genetic evaluation of large samples of parr, smolt and adults (Murdoch et al. 2007). This consideration makes sample size (power) a crucial step in designing RRS studies. One approach is to develop simulation models which would assist investigators in choosing adult and juvenile sample sizes that could be expected to attain a desired level of precision in the estimation of RRS. The models should be based on real data to reflect, actual expected data distributions, sampling rates, and hatchery – wild escapement conditions similar to those expected for the particular location or region of a proposed study (Berejikian et al. 2004). However users could create their own hypothetical input data to reflect alternative smolt abundance, fitness levels and adult covariate scenarios.

This report describes a method for examining alternative sample sizes, i.e., the proportion of parr, smolt, or returning adults,  $Q$ , to sample for parentage analysis. The objective is to meet precision and power requirements in RRS studies in which the number of offspring per hatchery parent is to be compared to the number of offspring per wild parent. The overall objective is to provide a flexible tool for researchers in RRS studies which would allow them to design experiments that would attain chosen levels of statistical power at optimal sampling rates of parr and smolt for genetic assignment. A proposal for future work is given at the conclusion of this report that would extend the boundaries and assumptions of current work to include adult enumeration variability, genotyping error and full development of

computer programs as well as a Monte Carlo study of potential fitness estimation bias and the effectiveness of the Manly program to correct for bias.

#### Determination of Q to meet precision requirements in RRS studies.

We examine alternative sample sizes to meet precision requirements in RRS studies using the bootstrap percentile method or power simulation algorithm (Beran, 1986), but without correction for estimation of non-zero offspring for cases in which the sampled data show zero offspring, i.e., without correcting the zeros. As we understand the current state of knowledge, researchers are analyzing their data without correcting the zeros. Also, our professional judgment is that recommendations for sample sizes to meet precision using this method will not vary much from recommendations that may be made when correcting the zeros. Future work would include determination of adequate sample sizes and sampling rates (Q ) when correcting for zeros using the Manly model.

When there are adults which have not been sampled, the case of incomplete adult enumeration, the effect is similar to that of reducing Q, decreased power to detect RRS different from 1 . This is because progeny of the missing adults cannot be assigned to their respective parents. Thus the total number of assigned offspring is reduced and precision to estimate fitness is lower. If adults are sampled in an unbiased manner, e.g., genetic information is randomly collected on 50% of the adults, then Q will effectively be reduced by the sampling fraction. That is, if 50% of the adults are sampled then it will only be possible to identify the male parent (or the female parent) of approximately 50% of the sampled juveniles. In the case of random collection of genetic samples from the parents, this section can be used to help determine the proportion of juveniles for which parents can be determined, i.e., the effective value of Q. In the case of an experiment in which incomplete adult enumeration is expected, the sampling fraction, Q, could be increased to help offset the effects of the missing adults for which offspring will be unassigned. Use of the programs then could assist fisheries investigators in the design of an RRS experiment where errors in adult enumeration are expected by allowing them to set sampling rates sufficient to offset the lowered power due to missing parent fish.

The case of incomplete adult enumeration may or may not result in unbiased estimates of fitness and RRS. Certainly RRS would be more likely to be biased if adults were missing disproportionately by origin of adult or level of fitness.

The most general approach uses the superpopulation non-parametric bootstrap that takes into account the potential finite sampling nature of smolt collections for subsequent parentage assignment (Davison and Hinkley, 1997). The use of infinite population methods in finite sampling problems results in an overestimation of variance (Thompson, 2002; Davison

and Hinkley, 1997). Thus offering a technique for handling the finite sampling case for studies in which  $Q$  is greater than 0.1 seems appropriate. If  $Q < 0.1$ , this step can be skipped and there will be a savings in required computer time. When  $Q > 0.1$  there is an advantage of shorter confidence intervals. The finite nature of the sampling is mimicked by creating a bootstrap super-population, concatenating  $\frac{1}{Q}$  copies of bootstrap re-samples from the original data.

Adjustments are available for the case when  $\frac{1}{Q}$  is not an integer (Davison and Hinkley, 1997).

The bootstrap superpopulation is then sampled without replacement at the original sample size to achieve one bootstrap sample. RRS is computed from the re-sample datasets as well as a bootstrap standard error. The superpopulation bootstrap program plots standard errors against user selected alternative  $Q$  values (Figure 1). Suppose it was of interest to know the standard error of RRS at a value of  $Q$  equal to .30 when the original input data was collected under a sampling scheme that allowed a  $Q$  of only .10. The user would input these two values of  $Q$  when running the program along with the name of the dataset. The first step is to create a facsimile of the original population. In the case of  $Q = .10$ ,  $1/.1$  or 10 copies of the original dataset are sampled with replacement from the original data and then concatenated to create the simulated version the original population that was sampled at  $Q = 0.10$ . A bootstrap sample for  $Q = 0.3$  would then consist of re-sampling without replacement a sample  $.3/.1$  or three times the size of the original data set from the simulated population data. Compared to the case where  $Q = .10$ , the  $Q = 0.3$  bootstrap estimate having been made on samples three times larger than the  $Q = 0.1$  case, will produce smaller standard errors for RRS.

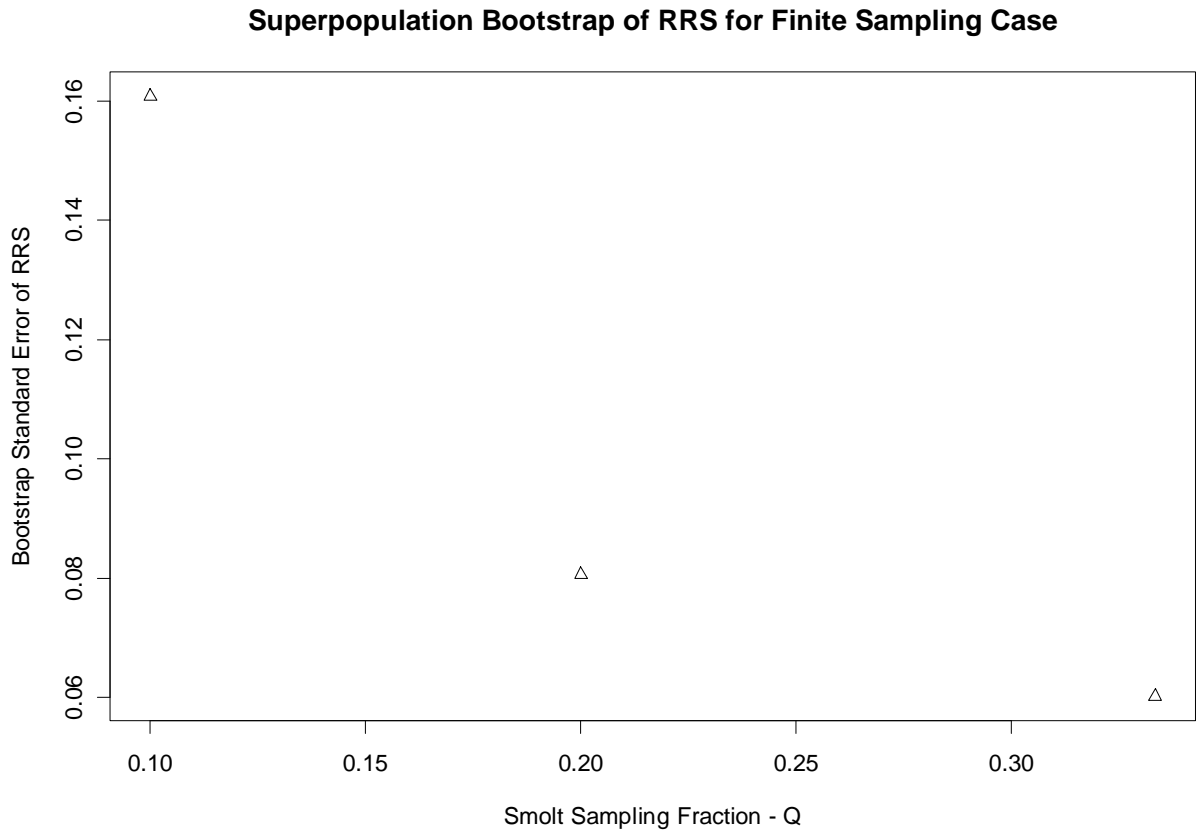


Figure 1. Bootstrap standard error of estimated RRS for  $Q = 0.1, 0.2$  and  $0.4$  using the bootstrap super-population method which accounts for the finite sampling case where  $Q$  is the smolt detection probability. Data for hatchery offspring per parent and wild offspring per parent were generated from Poisson distributions with means of 15 and 20 respectively.

When these data are sampled at rates of less than 10% the difference in variance estimated using finite population methods from that when using methods which assume an infinite population becomes negligible. For example, when applying a finite population correction adjustment for a sampling fraction of  $f = n / N$ , a confidence interval for the mean offspring count is shortened by a factor  $(1 - f)^{1/2}$ . For  $Q = 0.07$  this amounts to only .96. Thus for samples in which  $f$  is less than 0.1 the data may be considered to have been sampled from an infinite population without appreciable bias in variance estimation. Non parametric bootstrap methods which assume that the data are sampled from an infinite population include the percentile bootstrap or power simulation method.

The power simulation method employs a bootstrap null hypothesis distribution of RRS, the critical percentiles of which are then compared to the bootstrap alternative distribution to give a value of the power of rejecting a null hypothesis of  $RRS = 1$ , or equivalently that the



difference between hatchery fitness and wild fitness equals 0. The effect size is the value of RRS computed from the original data without correcting the zeros and a significance level is chosen by the user. The bootstrap percentile method has the advantage of simulating the null distribution of RRS and providing statistical power without a distributional assumption. Using the R software program in appendix 1c., it is possible for the user to vary the sampling fraction,  $Q$ , to find associated power, Figure 2.

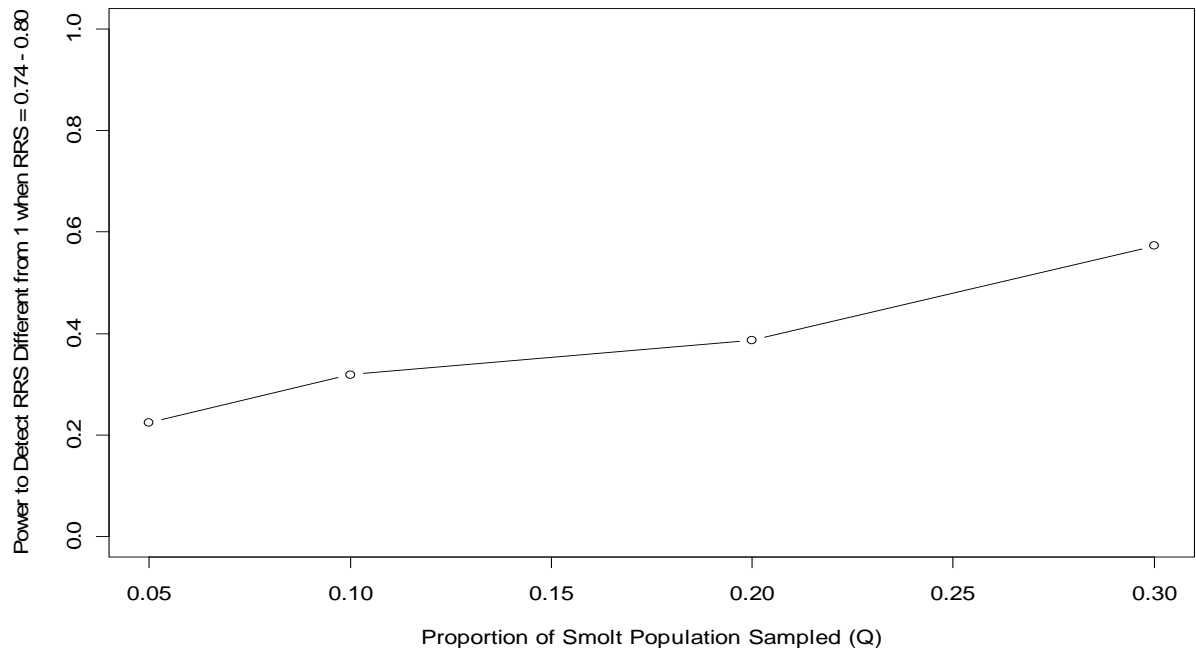


Figure 2. Power to detect RRS different from 1 and proportion of smolt population sampled. Data was generated from zero-inflated Poisson distributions with the probability of a zero in the data equal to  $\pi = 0.3$ . Sample sizes for hatchery and wild offspring counts was 75 and 150 cases respectively. Data sets for  $Q = .3, .2, .1$  and  $.05$  were obtained by sampling from zero inflated Poisson distributions with  $\pi = 0.3$  and hatchery and wild means of offspring equal to 30, 22.5 The R package VGAM was used to generate the four datasets.

Another use of the software is to investigate the effects of the proportion of zeros in the data on the power to reject the null hypothesis that the  $RRS = 1.0$ . Figure 3 contains an illustration of the type of power curves that can be developed using the R software programs in the Appendix.

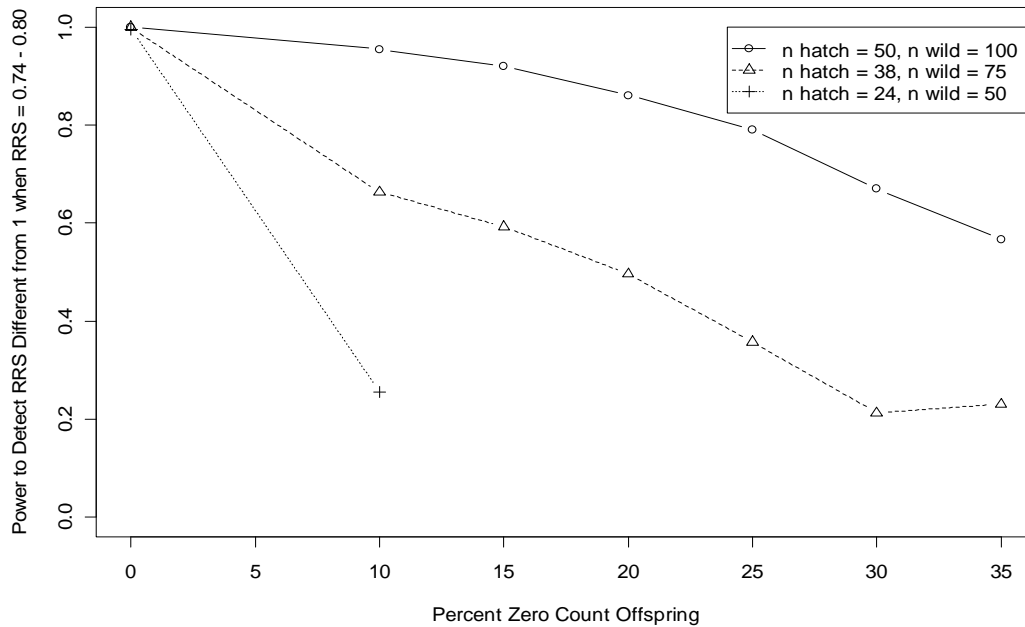
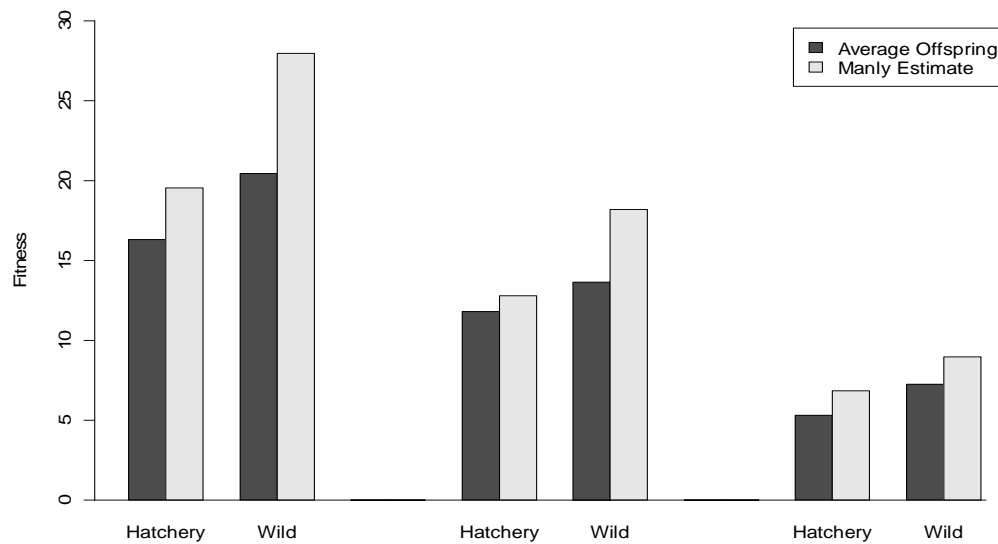


Figure 3. Power to detect RRS of 0.74-0.80 different from 1.0 for three sample sizes of hatchery smolt and wild smolt: 50, 100; 38, 75; and 25, 50 for Poisson simulated data having hatchery mean = 15 and wild offspring mean = 20 with frequencies of zero count offspring varying from 0.0 to 35 percent.

Figure 4. shows expected fitness estimated by the manly.main R program and fitness using average offspring per parent for three data sets have 30 percent zero cases.



## Algorithms

The R program, 'manly.main' calls two functions, 'manly.RRS.w' and 'manly.RRS.h' which compute the likelihood of the data as functions of the zero-inflated Poisson (zip) probabilities of occurrence of number of offspring per parent. The zip probabilities are functions of two parameters,  $\pi$ , the probability of a zero offspring and  $\lambda$ , the mean. The likelihood is then maximized via the R function 'Optim' in the main program, and maximum likelihood estimates of the parameters are returned; these are used in the estimate of hatchery fitness, wild fitness and RRS within the main program. The bootstrap simulation step is handled by the built-in R function 'boot' for which the user specifies the data, the above program, manly.main, the sampling fraction, Q, initial values for the parameters of the overall zip probability distribution function as well as the number of bootstrap replicates as arguments. Output of the boot function consists of the estimate of RRS, bootstrap standard error of RRS and an estimate of bias. Bootstrap confidence intervals can be obtained by using the built-in R function 'boot.ci' with the boot function output as an argument. The user may specify the type of confidence interval (percentile or bias corrected) as well as the confidence level as additional arguments to 'boot.ci.'

*Brian Manly Model Based Maximum Likelihood Bootstrap Algorithm*

- 1 expand the maximum value of both hatchery and wild offspring per parent in the data by a factor of  $1/Q$  where  $Q$  is the estimate of the smolt sampling fraction to obtain an estimate of the maximum offspring per parent in the smolt population.
- 2 apply the zero-inflated Poisson (zip) probability function to the sequence 1:max(offspring) for each origin offspring maximum computed in 1 and index the sequence by the letter j.  
multiply the vector of probabilities of 1 through max offspring in 2 for each origin by a matrix of binomial probabilities, one matrix for each origin type. Create the two hatchery and wild binomial probability matrices as follows:
- 3
  - a. the rows of the matrix are indexed by increasing  $i$  where  $i$  is a unique number of offspring per parent observed in the data. The row entries consist of  $i - 1$  zeros followed by the sequence of binomial probabilities of  $i$  offspring where the number of binomial trials is  $i$  through max offspring and where  $p$  the probability of a sampled offspring equals  $Q$  and  $1-p$  the probability of not sampling an offspring equals  $1-Q$ .
- 4 compute an expression for the likelihood by writing the product of the entries of the vectors obtained in step 3. Do this for each origin type.
- 5 maximize the two likelihoods, one for each origin type, by separate calls to the R function Optim.
- 6 apply the zip parameters estimated in step 5 to the zip probability function to estimate the probabilities of the observed numbers of offspring for each origin type.
- 7 estimate fitness for each origin type by summing the products of the observed offspring numbers by their respective estimated probabilities.
- 8 estimate RRS as the ratio of estimated hatchery fitness to estimated wild fitness.
- 9 repeat steps 1-8,  $r = 1$  through  $R$  times for  $R$  bootstrap datasets obtained by re-sampling with replacement from the original data.
- 10 compute the bootstrap mean fitness and bootstrap standard error as the empirical mean and standard error of the bootstrap estimates of RRS.
- 11 compute the bootstrap estimate of bias by subtracting the original estimate of RRS from the mean bootstrap estimate of RRS.
- 12 Compute the  $1 - \alpha$  percent bootstrap percentile confidence interval by choosing the  $\alpha/2 \times 100$  percentile and  $1 - \alpha/2 \times 100$  percentile of the sorted bootstrap RRS estimates where  $1 - \alpha \times 100$  is the confidence level of interest. Divide the confidence interval by 2 to obtain a half-width confidence interval.
- 13 Note: Steps 10 – 12 are carried out by the R built-in functions boot and boot.ci.

*Finite Sampling Superpopulation Bootstrap Algorithm (Davison and Hinkley 1997)*

Description: The R program, 'RRS.superpop.boot.multq' in Appendix 1b. gives bootstrap estimates of standard error of RRS when input is a dataset formatted as in Appendix 1a. and a value of Q, the probability of observing a given offspring. This approach takes into consideration the finite-sampling effects of estimated variances when Q ranges from 0.1-0.5 (Davison and Hinkley 1997, pgs. 92-94)

For  $r = 1, \dots, R$ , and origin = Hatchery, Wild

- 1 generate a replicate superpopulation  $Y^* = (Y_1^*, \dots, Y_N^*)$  by sampling  $N$  times with replacement from  $y_1, \dots, y_n$  and concatenating these  $N$  samples of size  $n$ . Here  $N \cong n / Q$ .
- 2 generate a bootstrap sample  $Y^* = (Y_1^*, \dots, Y_n^*)$  by sampling  $n$  times without replacement from  $Y^* = (Y_1^*, \dots, Y_N^*)$ , and set  $RRS_r^* = rrs(Y_1^*, \dots, Y_n^*)$ .
- 3 calculate the empirical standard error and half-width confidence intervals of the  $R$  bootstrap RRS statistics.
- 4 repeat 1-3 for different datasets and variable  $Q$  or choose alternate values of  $Q$  with the original  $Q$ .
- 5 plot bootstrap standard errors and half-width confidence intervals against  $Q$ .

*Calculating Statistical Power with the Percentile Bootstrap (Beran 1986, Hall and Wilson 1991) for  $Q < 0.1$ .*

Description: The R program, 'RRS.power' in Appendix 1c, uses input data formatted as in Appendix 1a. , a significance level and the number of bootstrap replicates required. Output is the power of detecting a difference in fitness between hatchery origin adults and wild origin adults equal to that of the sample data (effect size).

- 1 center the hatchery and wild offspring counts on their respective means.
- 2 sample  $n_w$  times with replacement from the entire centered offspring vector and sample  $n_w$  times with replacement from the adult wild fish sample ids. Compute the fitness of the above bootstrap sample using the bootstrapped sample ids.
- 3 repeat 2, this time sampling  $n_h$  times with replacement from the entire centered offspring vector and sampling  $n_h$  times with replacement from the adult hatchery fish sample ids. Compute the fitness of the above bootstrap sample using the bootstrapped sample ids.

- 4 subtract the above hatchery fitness from the wild fitness. This difference is one bootstrap sample comprising the empirical null distribution (choose  $R = 500, 1000, \dots$ ). The empirical null distribution will be centered on 0.
- 5 calculate the critical scores that correspond to the 2.5<sup>th</sup> and 97.5<sup>th</sup> critical alpha regions under the empirical null distribution: round  $((.05/2) \times (\# \text{bootstrap samples}))$  for lower percentile; and round  $((1 - (.05/2)) \times (\# \text{bootstrap samples}))$ . Locate the scores that correspond to those percentiles.
- 6 generate the bootstrap alternative distribution:
  - A.) re-sample with replacement from wild portion of the data matrix offspring vector with replacement to generate a bootstrap sample of wild offspring with length equal to the original wild offspring sample size.
  - B.) re-sample with replacement from the hatchery portion of the data matrix offspring vector with replacement to generate a bootstrap sample of hatchery offspring with length equal to the original hatchery offspring sample size.
  - C.) calculate fitness for both wild bootstrap sample and hatchery bootstrap sample using respective adult sample ids.
  - D.) subtract the hatchery fitness from the wild origin fitness. This is one bootstrap difference representing the difference in fitness under the empirical alternate distribution. This difference is centered on the population difference under the alternate hypothesis.
- 7 Calculate the empirical power of the statistical test
  - A.) Using the upper and lower critical scores for the empirical null hypothesis calculated in step 5.), calculate the number of difference values in the empirical alternative sampling distribution that are as or more extreme than the critical scores under the null distribution.
  - B.) Take the count in step A.) and divide by the total number of bootstrap samples. This is the empirical power for the statistical test that tests no difference between hatchery fitness and wild fitness (or RRS different from 1) versus the alternative hypothesis that hatchery fitness is different from wild fitness for the value of  $Q$  given for the original data set at the specified value of  $\alpha$  for the effect size equal to the original sample fitness of hatchery fish minus fitness of wild fish.

Notes: It should be possible and perhaps recommended that the finite-sampling super-population re-sampling approach be applied in Method 2 if  $Q$  is greater than 0.1, otherwise variances may be overestimated (Davison and Hinkley 1997).

## Conclusion

Copies of this report and the computer software programs in the appendices are available for download on the West, Inc. web site <http://www.west-inc.com>.

The Manly method has been presented for point estimation of RRS while correcting for excess zeros in the data with measures of precision in the form of confidence intervals and standard errors for RRS. We also provide sample size guidance for establishing required levels of precision in RRS studies where it is expected that offspring sampling will be incomplete. For this stage of the work it has been assumed that adult enumeration is complete with 100 percent of escapement genetically sampled. It is also assumed that the assignment of offspring to their respective parents is without error. The basic method 'corrects the zeros' in studies of RRS. Point estimates of RRS for specific subsets of the data with standard errors and confidence intervals can be obtained using the R program 'manly.main' in conjunction with built-in R functions for bootstrap estimates. The programs along with instructions for use and an example are given in Appendix 1. Potential models for the numbers of observed offspring that have been programmed are the zero-inflated Poisson, and zero-inflated negative binomial. The zero-inflated Poisson program is available for use while the zero-inflated negative binomial requires further work.

Applying non-zero inflated models where the data are distributed according to a zero-inflated model when using traditional asymptotic likelihood ratio sample size (power) estimation approaches (Self, 1992), often results in an underestimate of sample size (Williamson et al. 2007). The Brian Manly model based likelihood bootstrap method models the excess zeros in the data as well as accounting for the probability of non-zero offspring for parent/offspring pairs having zero offspring in the original dataset. The bootstrap percentile method and the super-population finite sampling methods do not require a parametric assumption but may be biased when data have excess zeros.

The super-population bootstrap is a non-parametric approach for estimating RRS and has the advantage of taking into account the finite-sampling nature of offspring collections when the sampling fraction is greater than 0.1. This method creates a simulated population equal in size to the original population. Bootstrap samples are drawn without replacement from the simulated population data to mimic finite sampling from the real population of offspring. The method is simplified to be exchanged for the percentile bootstrap method when the sampling fraction  $Q$  is small, e.g., below 0.1. The advantage of the bootstrap percentile method is its easy implementation and intuitive approach as well as not having to adopt a parametric model for the data.

Appendix 1 displays the R computer programs for estimating RRS and an estimate of its precision by the Manly method as well as by the other two non parametric bootstrap methods. Sample sizes to achieve required precision in RRS studies in which parr, smolt, or adult stage offspring are assigned to a single parent fish are among the output. Instructions for running the programs are included in Appendix 1 as well as sample output.

### Future work.

#### *Sample sizes for precision and power using the Manly method*

The preferred approach for the study of power is unfortunately still under development. It would utilize the model based maximum likelihood method conceived by Brian Manly, WEST, Inc., and discussed above. The method involves an underlying parametric model in which a likelihood is constructed relating probabilities of observing a given number of offspring to the probability of occurrence of numbers of offspring/parent at values estimated from bootstrap re-samples of RRS data. The estimated probabilities are then used to estimate expected fitness for each origin group. The standard model for counts of the observed number of offspring per parent is the Poisson. However, if the data are over-dispersed such that the variance of the count variable is greater than the mean, then a negative binomial distribution may be used as it employs an additional parameter to describe the variance. If in addition the data are zero-inflated, a zero-inflated Poisson or zero-inflated negative binomial distribution may be appropriate. A useful R program would accommodate the above alternate distributions. A simulation is run by re-sampling data where sampling effort and juvenile capture probability,  $Q$  are set by the user.

The Manly method has the greatest potential of the models considered in this report, because it allows estimation of non-zero offspring for cases in which the sampled data show zero offspring and avoids bias if cases with zero observed offspring are dropped from the data set. As it is likely that data will contain substantial zero offspring cases for a given parent it may be crucial to explicitly model this portion of the population to avoid negative bias in the estimation of fitness. The zero-inflated Poisson distribution assumes that a subpopulation generates the zero counts; this distribution accounts for excess zeros by estimating a separate parameter for the probability of zero values. Sample size (power) calculations are especially important for zero-inflated models because a larger sample size is required to detect a significant effect with these models than with the standard Poisson or negative-binomial models (Williamson et al.



2007). Another objective is to include variability in the estimate of the sampling fraction,  $Q$  into the estimation of RRS within the Brian Manly model based program.

The theory for this method is well understood, however the R computer software program requires some additional work to identify and implement the appropriate alternative distributions where necessary. Output would consist of standard errors or half-width confidence intervals of RRS plotted against alternative values of  $Q$  to provide guidance in sample size selection. The program would employ score tests to determine which distribution is the most suitable for the data (van den Broek, 1995; Ridout et al. 2001).

#### *Incomplete adult enumeration, its Relationship to $Q$ , and effect on power*

When there are adults which have not been sampled, the case of incomplete adult enumeration, the effect is similar to that of reducing  $Q$ , decreased power to detect RRS different from 1. This is because progeny of the missing adults cannot be assigned to their respective parents. Thus the total number of assigned offspring is reduced and precision to estimate fitness is lower. If adults are sampled in an unbiased manner, e.g., genetic information is randomly collected on 50% of the adults, then  $Q$  will effectively be reduced by the sampling fraction. That is, if 50% of the adults are sampled then it will only be possible to identify the male parent (or the female parent) of approximately 50% of the sampled juveniles. In the case of an experiment in which incomplete adult enumeration is expected, the sampling fraction,  $Q$ , could be increased to help offset the effects of the missing adults for which offspring will be unassigned. Use of the computer software provided in this report then could assist fisheries investigators in the design of an RRS experiment where errors in adult enumeration are expected by allowing them to set sampling rates sufficient to offset the lowered power due to incomplete collection of genetic information from the parents.

The case of incomplete adult enumeration may or may not result in unbiased estimates of fitness and RRS. Certainly RRS would be more likely to be biased if adults were missing disproportionately by origin of adult or level of fitness.

#### *Incomplete adult enumeration and errors in assignment of offspring to parents*

It is also necessary to extend the algorithms to experiments in which there is incomplete adult enumeration and assignment of offspring to parents is not without error. Guidelines are needed for the numbers or percentage of adult spawners to sample to provide acceptable power and precision for estimates of RRS. Similarly the effect of errors in assignment of offspring to parents on precision of estimates of RRS should be studied

One way in which the effect of incomplete adult enumeration could be included into the analysis of RRS precision would be to allow the user to randomly eliminate cases (rows) from

the data. The point estimate of RRS and bootstrap confidence interval obtained with the full data could be compared to the estimate and bootstrap confidence interval obtained from the data with the randomly missing adult/offspring cases to obtain an estimate of the effect of unsampled adults which may have escaped collection at the weir.

#### *Modeling the effect of assignment error*

To simulate assignment error the program could be configured to allow the user to miss-match random cases of assignment of offspring to parent. Estimates from the mismatch computer run could then be compared to estimates from computer output from the original error-free data to investigate effects of assignment error on power and estimates of fitness and RRS.

#### *Modeling the effect of covariates such as age and weight on RRS*

It would also be of benefit to include covariates measured on parents in order to assess the effect of adult morphological and behavioral characteristics on fitness, RRS and estimates of their standard errors. This could be achieved by including these covariates in the zero-inflated Poisson probability function which is used in the Brian Manly model based bootstrap. Null model output would then be compared to covariate models to assess the effect of the covariate.

#### *Comparison of RRS between studies or years*

Extension of the programs and algorithms to enable statistical comparison of RRS estimates between studies or studies between years will be an important objective. Here the statistic to be bootstrapped would be the difference in RRS between the two studies or years.

#### *Programming of additional distributions and tests of goodness of fit*

Future work should also entail configuring the Brian Manly model based bootstrap to use distributions other than the zero-inflated Poisson distribution where appropriate. The zero-inflated negative binomial has been programmed but not yet tested. Other distributions would include the Poisson and negative binomial when excess zero data is not present. Score tests and or goodness of fit tests within the program would help to insure selection of the correct distribution to apply to the data.

#### *Sensitivity of current estimation methods to the presence of zeros*

Finally it is desirable to estimate and study the magnitude of the negative bias when estimating fitness for populations in which data contain excess zeros. The level of this bias in relation to sample size and data distributional properties should be explored. Monte Carlo methods can be applied to run sensitivity analyses to establish where bias exists and under which sampling

conditions it attains unacceptable levels when the goal is accurate and precise estimation of RRS.

## References Cited

Araki, H., B.A. Berejikian, M.J. Ford and M.S. Blouin. 2008. Fitness of hatchery-reared salmonids in the wild. To be published in *Evolutionary Applications*.

Beran, R. (1986). Simulated Power Functions. *The Annals of Statistics*, 14(1), 151-173.

Berejikian, B. A. and M. J. Ford. 2004. Review of Relative Fitness of Hatchery and Natural Salmon. NOAA Technical Memorandum NMFS-NWFSC-61. Seattle, Washington.

van den Broek, J. (1995). A Score test for zero inflation in a Poisson distribution. *Biometrics* **51**, 738-743.

Davison, A.C. and D.V. Hinkley. 1997. *Bootstrap Methods and their Application*, Cambridge University Press.

Galbreath, P. F., C. A. Beasley, B. A. Berejikian, R. W. Carmichael, D. E. Fast, M. J. Ford, J. A. Hesse, L. L. McDonald, A. R. Murdoch, C. M. Peven, D. A. Venditti. 2008. Recommendations for Broad Scale Monitoring to Evaluate the Effects of Hatchery Supplementation on the Fitness of Natural Salmon and Steelhead Populations. Final Draft Report of the Ad Hoc Supplementation Monitoring and Evaluation Workgroup.

Hall P, S.R. Wilson. (1991). Two guidelines for bootstrap hypothesis testing. *Biometrics* **47**, 757-762.

Hinrichsen, R.A. (2003). The power of experiments for estimating relative reproductive success of hatchery-born spawners. *Can. J. Fish. Aquat. Sci.* 60: 864-872.

Johnson, N., S. Kotz, and A.W. Kemp. 1992. *Univariate Discrete Distributions*, 2<sup>nd</sup> Edition, Wiley: New York.

Murdoch, A.R., T.N. Pearsons, T.W. Maitland, C.L. Deason, M.J. Ford and K. Williamson. 2007. Monitoring the reproductive success of naturally spawning hatchery and natural spring Chinook salmon in the Wenatchee River. BPA Project No. 2003-039-00, Contract No. 00020391; Performance Period February 1, 2006-January 31, 2007.

Ridout, M., J. Hinde, G.B. Demetrio. (2001). A score test for testing a zero-inflated Poisson regression model against zero-inflated negative binomial alternatives. *Biometrics* **57**, 219-223.

R Statistical Computing Download. <http://cran.r-project.org/bin/windows/base/old/>

Self, S.G., R.H. Mauritsen and J. Ohara. 1992. Power Calculations for Likelihood Ratio Tests in Generalized Linear Models. *Biometrics*, 48(1), 31-39.

Thompson, S.K. 1992. Sampling, 2<sup>nd</sup> Edition, Wiley: New York.

Williamson, J.M., Hung-Mo Lin, R.H. Lyles, and A.W. Hightower. 2007. *Journal of Data Science* 5, 519-534.

## Appendix 1 – R Programs, Instructions for use, data description, and sample output,

The program R can be downloaded from the following site:

<http://cran.r-project.org/bin/windows/base/old/>

### Appendix 1a. Brian Manly Model Based Maximum Likelihood Bootstrap

Program Name: manly.main

(Current program uses a zero-inflated Poisson distribution to model offspring counts.)

#### **Instructions for use**

1. Download R version 2.6.2 by going to the website above.
2. Open package boot by typing 'library(boot)' at the R prompt.
3. Set up an RRS dataset in EXCEL which should look like the following example header and first few lines:

sample	origin	offspring
1	h	15
2	h	14
3	h	19
.	.	.
.	.	.
51	w	23
52	w	19
53	w	13
.	.	.
.	.	.

Column 'sample' are numbers to identify an adult parent to its respective assigned number of sampled offspring in the 'offspring' column.

Save the dataset as a .csv file in a folder and directory of your choice.

4. Set the R directory to the same one where the data file is stored by clicking on 'file' then 'change directory' and selecting the appropriate directory and folder; click on 'OK'.
5. Create a dataframe which the R program can use for input by typing the following at the R prompt:  

```
mydata<- read.csv(file = "mydata.csv", header = TRUE, sep = ",")
```
6. Highlight and paste the three R program, manly.main below, into R at the R prompt.
7. As an example, to obtain bootstrap estimates of RRS, type the following into the R prompt:

```
out.boot <- boot( data=mydata, type='i', statistic=manly.main, q=.3, R=10, par=c(.7,80) )
```

**notes on the arguments of the built-in R boot function:**

Type 'boot.out' to see bootstrap estimates of RRS. Type 'summary(boot.out)' for a summary of program run. Type boot.out\$t to see R bootstrap estimates of RRS. 'help(boot)' for more information on the function boot.

'data' is the dataframe you just created.

'type= i' is the re-sample index for the data.

'statistic' is the R function or R program in which the statistic to be bootstrapped is computed.

'par' consists of initial values for the parameters of the zip distribution (ie. Percent 0's in your data set and maximum single offspring count for your data set's offspring counts expanded by a factor of 1/Q.

'q' is the smolt sampling fraction.

'R' is the user selected number of bootstrap re-samples.

**Notes:** Currently the program takes a considerable amount of time, say several hours, to run R = 50 bootstrap replicates.

The current version of the code has a limit of max offspring of the expanded data of 160.

It may take more than one trial set of initial values to find one that allows convergence.

Type help(boot) for more information on how to use the R boot function.

The zero-inflated negative binomial distribution has been programmed but not tested.

8. To obtain bootstrap confidence intervals after having run the **boot** function , use the R built-in function **boot.ci** by entering the following code into the R prompt:

```
boot.ci.out<- boot.ci( boot.out, type="perc", conf=c(.80,.95) )
```

The arguments of the **boot.ci** function are the output of the **boot** function in step 7.

'type' is the type of confidence interval, percentile in this case and 'conf' in which I have selected an 80% and 95% confidence interval to be output are the confidence levels.

After running the boot.ci function, type 'boot.ci.out' to see the results.

For more information on the R **boot.ci** function type help(boot.ci) at the R prompt.

### Sample Output from Program manly.main

```
> boot.out
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

```
boot(data = dataset, statistic = manly.main, R = 10, stype = "i",
```

```
  par = c(0.7, 80), q = 0.3)
```

Bootstrap Statistics :

original	bias	std. error
----------	------	------------

t1*	0.6828665	-0.03847215 0.05328027
-----	-----------	------------------------

```
> boot.ci.out
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 10 bootstrap replicates

CALL :

```
boot.ci(boot.out = out, conf = c(0.8, 0.95), type = "perc")
```

Intervals :

Level	Percentile
-------	------------

80%	( 0.5563, 0.7237 )
-----	--------------------

95%	( 0.5519, 0.7275 )
-----	--------------------

**Notes:** The sample simulated data for the above program was generated from a zip distribution with parameters 0.30 probability of a zero offspring count and lambda (mean) for hatchery and wild offspring counts of 22.5 and 30 respectively. The sample sizes were 75 for hatchery and 150 for wild origin fish.

### **R Computer Program: manly.main for the Brian Manly Maximum Likelihood based Bootstrap**

```
manly.main<-function(dataset,index,par,q)

{

  # This program calls two functions, manly.RRS.h and manly.RRS.w, which compute the likelihood for
  # probabilities of observing offspring/parent.

  # numbers in RRS data.

  # Likelihoods for hatchery and wild observed probabilities are maximized wrt the parameters of a zero-
  # inflated Poisson (zip) distribution.

  # Output is an estimate of RRS.

  # par - initial values for zero-inflated Poisson, c(theta, lambda), where theta is the proportion of zeros
  # in 'dataset' and lambda is the mean offspring of the dataset/q.

  # q - is the capture probability of a smolt or the sampling fraction.

  dataset<- dataset[index,]          # Set up bootstrap sample index.

  attach(as.data.frame(dataset))      # Attach to name data columns in program.

  n_h<- unique(dataset[origin=="h",]$offspring) # Compute and sort unique numbers of observed
  offspring/parent

  n_w<- unique(dataset[origin=="w",]$offspring)

  n_h<- sort(n_h)

  n_w<- sort(n_w)

  len_n_h<- length(n_h)

  len_n_w<- length(n_w)

  #=====
  #=====

  manly.h<- function(par,...)

  {
```



# This function writes the negative loglikelihood of zero-inflated Poisson distribution data (mixture type) of each of hatch and wild offspring count data

# for the Brian Manly method of estimating simulated se of RRS when offspring data is zero-inflated Poisson distributed.

# Initial values for zero-inflated Poisson distribution of the hatchery offspring.

```
#par<- rep(NA,2)
```

```
theta_h<- par[1]
```

```
lambda_h<- par[2]
```

# The test smolt sampling fraction for the input dataset.

```
Q<- q
```

# dataset used for example.

```
#dat<-dataset
```

```
attach(as.data.frame(dataset))
```

# Distinct values of numbers of offspring for each origin type.

```
n_h<- unique(dataset[origin=="h",]$offspring)
```

```
mean_h<- mean(dataset[origin=="h",]$offspring)
```

```
n_h<- sort(n_h)
```

```
len_n_h<- length(n_h)
```

#write number of offspring as zero inflated poisson probabilities where 1-theta is the proportion of extra zeros.

```

offsprg_h<- rep(NA,len_n_h)

offsprg_h<- round(n_h/Q,0)

max_offsprg_h<- max(offsprg_h)

# Set up vectors of probabilities of occurrence of 0,1,... offspring using parameters of zero-inflated
Poisson dist. which are to be estimated.

p_h<- rep(NA,max_offsprg_h+1)

p_h[2:(max_offsprg_h+1)]<- theta_h*(ppois(1:max_offsprg_h,lambda_h)-ppois(0:(max_offsprg_h-
1),lambda_h))

p_h[1]<- (1 - theta_h) + theta_h*ppois(0,lambda_h)

p_h<- as.vector(p_h)

#print(length(p))

# Set up, m, a matrix, the rows of which are observed probabilities expressed as a function of p and Q.

m_h<- rep(NA,len_n_h*(max_offsprg_h+1))

dim(m_h)<- c(len_n_h,(max_offsprg_h+1))

# compute binomial terms containing Q for observed probabilities
for (j in 2:len_n_h)    {

m_h[j, 1:n_h[j]]<- rep( 0, n_h[j] )

m_h[j,(n_h[j]+1):(max_offsprg_h+1)]<- dbinom( (n_h[j]+1),(n_h[j]+1):(max_offsprg_h+1),Q)

```

```
}
```

```
m_h[1,]<- (1-Q)^(0:max_offsprg_h)
```

```
# Multiply the rows of m by the probabilities of occurrence to get the observed probabilities of 0,1,..
offspring per parent.
```

```
prob_h<-m_h*%p_h
```

```
#print(sum(prob_h))
```

```
# Write the negative loglikelihood as the product of the observed probabilities of 0,1,..
offspring/parent.
```

```
print(prob_h)
```

```
-log(prod(prob_h))
```

```
}
```

```
#=====
=====
```

```
manly.w<- function(par,...)
```

```
{
```

```
# This function writes the negative loglikelihood of zero-inflated Poisson distribution data (mixture type)
for wild offspring count data
```

```
# for the Brian Manly method of estimating simulated se of RRS when offspring data is zero-inflated
Poisson distributed.
```

```
# Initial values for zero-inflated Poisson distribution of the hatchery offspring.
```

```

#par<- rep(NA,2)

theta_w<- par[1]

lambda_w<- par[2]


# The smolt test sampling fraction for the input dataset.

Q<- q


# dataset used for example.

#dat<-dataset

attach(as.data.frame(dataset))


# Distinct values of numbers of offspring for each origin type.

n_w<- unique(dataset[origin=="w",]$offspring)

mean_w<- mean(dataset[origin=="w",]$offspring)


n_w<- sort(n_w)


len_n_w<- length(n_w)


#write number of offspring as zero inflated poisson probabilities where 1-theta is the proportion of
extra zeros.


offsprg_w<- rep(NA,len_n_w)

offsprg_w<- round(n_w/Q,0)

max_offsprg_w<- max(offsprg_w)

```

```
# Set up vectors of probabilities of occurrence of 0,1,... offspring using parameters of zero-inflated
Poisson dist. which are to be estimated.
```

```
p_w<- rep(NA,max_offsprg_w+1)
```

```
p_w[2:(max_offsprg_w+1)]<- theta_w*(ppois(1:max_offsprg_w,lambda_w)-ppois(0:(max_offsprg_w-
1),lambda_w))
```

```
p_w[1]<- (1 - theta_w) + theta_w*ppois(0,lambda_w)
```

```
p_w<- as.vector(p_w)
```

```
#print(length(p))
```

```
# Set up, m, a matrix, the rows of which are observed probabilities expressed as a function of p and Q.
```

```
m_w<- rep(NA,len_n_w*(max_offsprg_w+1))
```

```
dim(m_w)<- c(len_n_w,(max_offsprg_w+1))
```

```
# compute binomial terms containing Q for observed probabilities
```

```
for (j in 2:len_n_w)    {
```

```
  m_w[j, 1:n_w[j]]<- rep( 0, n_w[j] )
```

```
  m_w[j,(n_w[j]+1):(max_offsprg_w+1)]<- dbinom( (n_w[j]+1),(n_w[j]+1):(max_offsprg_w+1),Q)
```

```
}
```

```

m_w[1,]<- (1-Q)^(0:max_offsprg_w)

# Multiply the rows of m by the probabilities of occurrence to get the observed probabilities of 0,1,..
offspring per parent.

prob_w<- m_w%*%p_w

#print(prob_w)

# Write the negative loglikelihood as the product of the observed probabilities of 0,1,..
offspring/parent.

#print(prob_h)

-log(prod(prob_w))

}

#=====
=====

# Maximize the likelihood for hatchery offspring by calling the function manly.h.

out.h<- optim(par,manly.h,factr=1e3,maxit=250, method = "L-BFGS-B",
  lower=c(.1, 10), upper=c(.99, 500))

out<-optim(par,manly.w,factr=1e3,maxit=250, method = "L-BFGS-B",
  lower=c(.1, 10), upper=c(.99, 500))

# Simulated Annealing

#out.h <- optim(par,manly.h, method="SANN",
#  control=list(maxit=200, temp=20))

#out <- optim(par,manly.w, method="SANN",
#  control=list(maxit=200, temp=20))

```

# Maximize the likelihood for wild offspring by calling the function manly.RRS.w.

```
thet_h<- out.h$par[1]          # name estimated zip 0 probability parameter estimated by  
optim above.
```

```
lam_h<- out.h$par[2]          # name estimated zip mean parameter estimated by optim  
above.
```

```
lam_h<-lam_h*q                # Scale zip mean parameter back to that of data.
```

```
pr_h<- rep(NA,length(n_h))
```

```
pr_h[2:len_n_h]<- thet_h*(ppois(n_h[2:len_n_h],lam_h)-ppois(n_h[1:(len_n_h-1)],lam_h)) # Compute  
probabilites of occurrence of 0, 1, 2, ... offspring.
```

```
pr_h[1]<- (1-thet_h) + thet_h*ppois(0,lam_h)          # of offspring/parent.
```

```
fitness_h<- t(pr_h)%*%n_h
```

```
sample numbers                # Compute hatchery fitness by summing products of  
respective probabilities.      # of offspring/parent with their
```

```
probabilities.                # of offspring/parent with their respective estimated
```

```
thet_w<- out$par[1]
```

```
lam_w<- out$par[2]
```

```
lam_w<-lam_w*q
```

```
pr_w<- rep(NA,length(n_w))
```

```

pr_w[2:len_n_w]<-thet_w*(ppois(n_w[2:len_n_w],lam_w)-ppois(n_w[1:(len_n_w-1)],lam_w))
pr_w[1]<- (1-thet_w ) + thet_w*ppois(0,lam_w)
fitness_w<- t(pr_w)%**n_w

RRS<- fitness_h/fitness_w                                # Compute RRS as ratio of hatchery to wild
fitness.

return(RRS)

}

```

#### **Appendix 1b. Finite-sampling superpopulation bootstrap**

**Program Name** - RRS.superpop.boot.multq

**Output** – The output is a plot of bootstrap estimated standard errors of RRS versus user selected values of Q.

**Instructions for Use** – See comment lines in the RRS.boot.multi.q program below.



1. Download R using the website in Appendix 1a.
2. Format an RRS dataset, set R directory and create a dataframe by following steps 3-5 in Appendix 1a.
3. Copy the RRS.boot.multi.q program into R at the R prompt.
4. Type **RRS.boot.multi.q( )** at the R prompt and type in the following arguments separated by commas within the parentheses the arguments for the program:
  - a. The name of your dataframe.
  - b. The smolt sampling fraction
  - c. A list of sampling fractions as in “c(.1, .2, .3333,..)”
  - d. The number of bootstrap replicates.
  - e. An example would be **RRS.boot.multi.q(mydata, .2, c(.1, .2, .3333), 99 )**

```
RRS.superpop.boot.multq<- function(data,q,fraction,iter)
```

```
{
```

```
  #This function implements a finite sampling superpopulation bootstrap for RRS, Davison and Hinckley (1997) pgs.92-97.
```

```
  #The function plots bootstrap se's of RRS versus user input values of q.
```

```
  #user's values of q are in fraction.
```

```
  #q is the sampling fraction of original data.
```

```
  #iter is the number of bootstrap samples
```

```
  #store bootstrap RRS means and RRS standard errors
```

```
  RRS_values<- rep(NA,length(fraction))
```

```
  sd_RRS_multi_q<- rep(NA,length(fraction))
```

```
  #bootstrap RRS and it se for user's q values
```

```
  for ( k in 1:length(fraction) )
```

```
  {
```

```
    Q<- fraction[k]
```

```
    R<- iter
```

```
    dat<- data
```

```
    attach(dat)
```

```

fitness_h<- rep(NA,R)

fitness_w<- rep(NA,R)

# concatenate 1/Q copies of the original data

for ( i in 1: round(1/Q,0) )

{

  dat_concat<- rbind(dat,dat)

}

# sample with replacement from expanded (1/Q times) data set a new data set of same dim

# sample without replacement from above data set with length equal to original data

# compute fitness by origin from above data set

for ( r in 1:R ) {

  index<- sample(1:dim(dat_concat)[1],dim(dat_concat)[1], replace=TRUE )

  super_pop<- dat_concat[index,]

  index_2<- sample(1:dim(super_pop)[1],dim(dat)[1]*(Q/q),replace=FALSE)

  boot_samp<- super_pop[index_2,]

  attach(boot_samp)

  # sum no. of offspring by adult id and compute fitness

  boot_samp_adultID<- aggregate(boot_samp$offspring,
list(boot_samp$sample,boot_samp$origin),sum)

  #boot_samp_adultID<- boot_samp

  names(boot_samp_adultID)<- c("sample","origin","offspring")

  attach(boot_samp_adultID)

  fitness_h[r]<- mean(as.numeric(boot_samp_adultID[origin=="h",]$offspring),na.rm=T)

  fitness_w[r]<- mean(as.numeric(boot_samp_adultID[origin=="w",]$offspring),na.rm=T)

  #print(fitness_h[r])

}

```

```

RRS<- fitness_h/fitness_w
sd_RRS_multi_q[k]<- sd(RRS )
RRS_values[k]<- mean(RRS)
}

#print(length(sd_RRS_multi_q))

Par(cex=1.5)

plot(fraction,sd_RRS_multi_q, xlab="Smolt Sampling Fraction - Q",ylab="Bootstrap Standard Error of
RRS",main="Superpopulation Bootstrap of RRS for Finite Sampling Case")

attach(dat)

sample_RRS<- mean(dat[origin=="h",]$offspring)/mean(dat[origin=="w",]$offspring)

cat("Sample RRS =",sample_RRS)

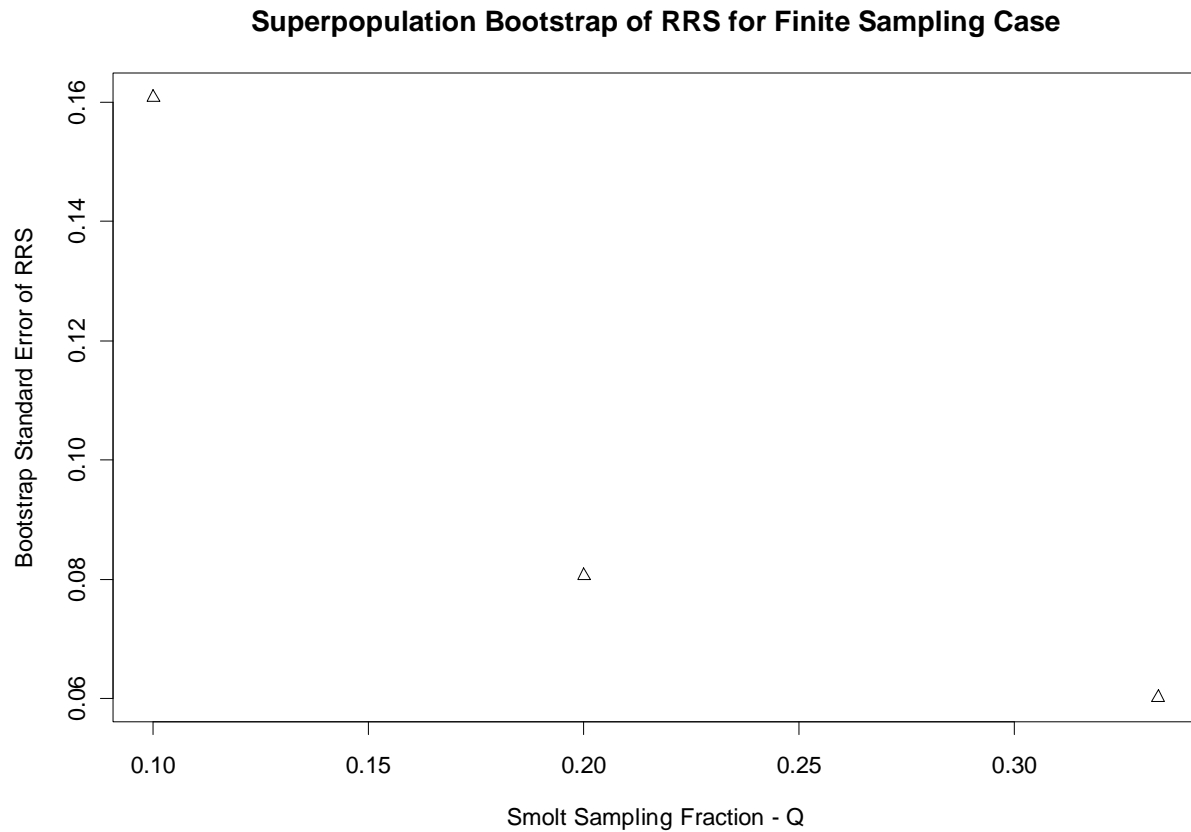
}

```

### Example Run

```
> RRS.superpop.boot.multq(zip_wh_.3_30_22.5_q.3,.333,c(.1,.2,.3333),10)
```

> Sample RRS = 0.7904855



**Sample Data Set for this Example:**

Data used for this example was simulated from a zero-inflated Poisson distribution with probability of zero equal to 0.3 and hatchery and wild offspring mean offspring count of 22.5 and 30 respectively. Sample sizes for the hatchery and wild were 75 and 150 respectively.

**Appendix 1c. Bootstrap Percentile Method (power simulation algorithm)**

This R function outputs the power of a test of the difference in fitness between hatchery parents and wild parents (or RRS different from 1). Output includes the power of the test, p-value of the test, a bootstrap estimate of bias, and the bootstrap estimate of the standard error of estimated RRS.

**Program Name: RRS.power**

**Instructions for use:**

1. Download R from the website given in Appendix 1a. and follow steps 3. – 5. In Appendix 1a for setting an R directory and creating an RRS dataset and dataframe.
2. Type **RRS.power( )** at the R prompt.
3. In the parentheses enter the arguments for RRS.power:
  - a. Your dataframe ('data') formatted as in Appendix 1a. Example - mydata
  - b. A level of significance ('alpha') for the estimated power. Example – 0.05.
  - c. The number of bootstrap replicates ('iter'). Example – 499.

**The R program – RRS.power**

**RRS.power<- function(data,alpha,iter)**

```
{  
  
  dat<- data  
  
  attach(dat)  
  
  n_h<- length(dat[origin=="h",]$offspring)  
  
  n_w<- length(dat[origin=="w",]$offspring)  
  
  n_h.total<- sum(dat[origin=="h",]$offspring)  
  
  n_w.total<- sum(dat[origin=="w",]$offspring)  
  
  #Center hatchery data  
  
  offspr_h_center<- dat[origin=="h",]$offspring - mean(dat[origin=="h",]$offspring)  
  
  #Center wild data  
  
  offspr_w_center<- dat[origin=="w",]$offspring - mean(dat[origin=="w",]$offspring)  
  
  #stack data  
  
  centered<- c(offspr_h_center,offspr_w_center)  
  
  dat<-cbind(dat,centered)  
  
  names(dat)<- c("sample","origin","offspring","cent_offspring")
```

```

attach(dat)

h0offspr_h<- matrix(sample(cent_offspring, n_h*iter,replace=T),nrow=iter)

h0offspr_w<- matrix(sample(cent_offspring, n_w*iter,replace=T),nrow=iter)

h0offspr_mean_h<-apply(h0offspr_h,1,mean)

h0offspr_mean_w<-apply(h0offspr_w,1,mean)

h0offspr_mean<- sort(h0offspr_mean_w - h0offspr_mean_h)

#### Calculate critical cutoffs

critup<- quantile(h0offspr_mean,.975)

critlow<- quantile(h0offspr_mean,.025)

#### Calculate two-sided mean difference test probability for observed difference

diff.empirical<- mean(dat[origin=="w",]$offspring)-mean(dat[origin=="h",]$offspring)

RRS.empirical<- (mean(dat[origin=="h",]$offspring))/(mean(dat[origin=="w",]$offspring))

count<-length(h0offspr_mean[abs(h0offspr_mean)>=abs(diff.empirical)])

pvalue.empirical<-count/iter

# Sampling from H1: Sample with replacement from centered offspring by origin

#fitness_h<- rep(NA,iter)

#fitness_w<- rep(NA,iter)

#for (r in 1:iter ) {

#index<- sample(1:dim(dat)[1],dim(dat)[1],replace=T)

# boot_samp<- dat[index,]

# sum no. of offspring by adult id and compute fitness

#boot_samp_adultID<- aggregate(boot_samp$cent_offspring,
list(boot_samp$sample,boot_samp$origin),sum)

# boot_samp_adultID<- boot_samp

# names(boot_samp_adultID)<- c("sample","origin","offspring","cent_offspring")

# attach(boot_samp_adultID)

```

```

# fitness_h[r]<- mean(as.numeric(boot_samp_adultID[origin=="h"],$cent_offspring),na.rm=T)

# fitness_w[r]<- mean(as.numeric(boot_samp_adultID[origin=="w"],$cent_offspring),na.rm=T)

# }

h1offspr_h<- matrix(sample(dat[origin=="h"],$offspring, n_h*iter,replace=T),nrow=iter)

h1offspr_w<- matrix(sample(dat[origin=="w"],$offspring, n_w*iter,replace=T),nrow=iter)

h1offspr_mean_h<-apply(h1offspr_h,1,mean)

h1offspr_mean_w<-apply(h1offspr_w,1,mean)

#### Sort and subtract fitness data vectors

# and accumulate into a fitness difference vector

h1bvec<- sort(h1offspr_mean_w - h1offspr_mean_h)

RRS.boot.mean<- mean(h1offspr_mean_h/h1offspr_mean_w)

RRS.boot.se<- sd(h1offspr_mean_h/h1offspr_mean_w)

boot.bias<-RRS.empirical-RRS.boot.mean

### Calculate the upper and lower cutoff percentiles for

# the lower and upper alpha criterion

effectlow<-round((alpha/2)* iter)

effectup<-round((1-alpha/2)* iter)

#### Calculate Empirical Power

countup<-length(h1bvec[h1bvec>=critup])

countlow<-length(h1bvec[h1bvec<=critlow])

power.twotail<-(countup+countlow)/iter

### Calculate fitness differences that correspond to the

# upper and lower alpha criterion cutoffs

h1.ci<-list(ci=c(h1bvec[effectlow], h1bvec[effectup]))

###Display Results

```

```

cat("The number of hatchery parents and wild parents assigned offspring are respectively ", n_h)
cat(" and ", n_w,"\n")
cat("The number of hatchery offspring and wild offspring assigned parents are respectively ", n_h.total)
cat(" and ", n_w.total,"\n")
cat ("The power of the test is ", power.twotail,"\n")
cat ("for a difference in fitness between hatchery and wild parents of ",diff.empirical,"\n")
cat("The empirical p-value is ", pvalue.empirical,"\n")
print("The bootstrap percentile confidence interval for the difference in fitness is: ")
print(h1.ci)
cat("The observed RRS is ", RRS.empirical)
cat(" and the bootstrap mean RRS is ",RRS.boot.mean,"\n")
cat("The bootstrap estimate of bias is ", boot.bias,"\n" )
cat("The bootstrap estimate of the standard error of RRS is ",RRS.boot.se,"\n")
}

```

### **Sample Output for Program RRS.power**

```
>RRS.power(zip_wh_.3_30_22.5_q.3, .05, 499)
```

The number of hatchery parents and wild parents assigned offspring are respectively 75 and 150

The number of hatchery offspring and wild offspring assigned parents are respectively 1213 and 3069

The power of the test is 0.5831663

for a difference in fitness between hatchery and wild parents of 4.286667

The empirical p-value is 0.03406814

The bootstrap percentile confidence interval for the difference in fitness is

```
[1] 0.4733333 7.5600000
```

The observed RRS is 0.7904855 and the bootstrap mean RRS is 0.796873



The bootstrap estimate of bias is -0.006387487

The bootstrap estimate of the standard error of RRS is 0.08282443

**Sample Data Set for this Example:**

Data used for this example was simulated from a zero-inflated Poisson distribution with probability of zero equal to 0.3 and hatchery and wild offspring mean offspring count of 22.5 and 30 respectively. Sample sizes for the hatchery and wild were 75 and 150 respectively.

