Testing the Fair Raffle

By Bryan D. Neff and Lindi M. Wahl
21 April 2004

This program uses the maximum likelihood method developed by Neff and Wahl (2004) to examine mechanisms of sperm competition. The model calculates the relative competitiveness of two males' sperm ("loadings" = r) as well as the economy of scale (non-linear returns to sperm number = t) and compares these to a defined null hypothesis for r and t – e.g., for the fair raffle r = 1 and t = 1. The likelihood method also can be generalized to other functions relating sperm and offspring number to P_2. Users with other functions may request the C++ code from Neff (bneff@uwo.ca).

The program is run by double clicking the "Neff&Wahl_Raffle.exe" file (from windows explorer, for example). All files must be in the same directory. The program requires a single input file ("raffledata.txt") with the following structure (the file can be created in Microsoft Excel and saved as 'tab delimited' or in a text editor with tabs or a space used to separate values within a row and a carriage return to separate rows):

Title: raffledata.txt
Structure:

Test parameter for *r*	Test parameter for *t*	Number of broods	
Sperm number male 1, brood 1	Sperm number male 2, brood 1	Offspring number male 1, brood 1	Offspring number male 2, brood 1
Sperm number male 1, brood 2	Sperm number male 2, brood 2	Offspring number male 1, brood 2	Offspring number male 2, brood 2
Sperm number male 1, brood 3	Sperm number male 2, brood 3	Offspring number male 1, brood 3	Offspring number male 2, brood 3
…	…	…	…
…	…	…	…

Importantly, offspring numbers must be integers, while sperm numbers may be a real number; i.e., decimals are allowed. Decimal values should be used when, for example, sperm number is calculated to the nearest 100,000 and the numbers are in the order of millions. For example, 1.2 might represent 1.2 million sperm and it should be entered over the number 1200000. There is a test file (raffledata.txt) with mock data that can be downloaded for inspection and trial. It should contain the following data:

1	1	10	
48.5	85.1	4	5
27.3	78.5	2	7
49.3	79.0	5	3
78.7	55.5	12	2
86.2	33.2	4	1
98.1	94.6	6	9
3.1	78.1	0	8
41.1	27.9	5	5
52.0	10.2	2	0
32.9	53.3	3	3

In this sample data, the first line denotes a null hypothesis of $r = 1$ and $t = 1$ and 10 broods. Each subsequent line represents the data for broods 1 through 10. As an example, 48.5 and 85.1 represent 48.5 million sperm for male 1 and 85.1 million sperm for male 2, and the 4 and 5 represent the number off offspring sired by male 1 and male 2, respectively.

The program generates two output files. The first titled "checkdata.txt" should be equivalent to your data input file raffledata.txt, with the exception that the output values in checkdata.txt are presented to 4 decimal places only. If the checkdata.txt file is not equivalent to the input data file then the data has not been correctly read by the program. Ensure that the file is constructed as described (e.g., text, tab delimited file). If problems persist please contact Neff. The second output file is the results of the analysis ("results.txt"). The first two p-values correspond to the test of *r* or *t* to the null hypothesis of *r* and *t* equal to the inputted values in the raffledata.txt file. So, for example, in the sample file above the null hypothesis is $r = 1$ and $t = 1$. The third p-value corresponds to the test of the estimated *t* to the null hypothesis of *r* equal to the inputted value and $t = 0$. This latter test corresponds to the "sperm independent model" (see Neff and Wahl 2004). The output data are also displayed to the screen, but on completion of the program the execution window sometimes automatically closes. In this case, simply consult the results.txt file using a text editor or Microsoft Excel.

As the program runs there is a counter displayed which goes from 100 down to 0. The program may take some time to execute; e.g., for the sample data file above it takes about 15 minutes on a 1 GHz PC processor. The results from running the program with the sample data should be:

```
Results for inputted null hypothesis of r= 1.0000 and t= 1.0000
r=7.4723e-001          p=  0.1156
t=1.2609e+000          p=  0.2809

Results for null hypothesis of r= 1.0000 and t=0
t=1.2609e+000          p=  0.0000
```

So, based on these data $r \approx 0.75$ and $t \approx 1.26$ and neither of these parameters were significantly different from $r = 1$ and $t = 1$. Thus, the null hypothesis of a fair raffle is accepted. Conversely, the null hypothesis of $r = 1$ and $t = 0$ is rejected with a very small p-value ($p < 0.0001$).

Because of the nature of using a randomization routine to generate the p-values, when the program is run multiple times with the same data slightly different p-values may result. This variation typically occurs in the third digit after the decimal. Thus, a p-value of 0.035 on the first run might appear as 0.038 in a subsequent run. The estimated *r* and *t* should not vary across runs. In the event that a p-value fluctuates around an important value such as 0.050, the p-values from multiple runs can be averaged to provide a more robust estimate.