

To the Graduate Council:

I am submitting herewith a thesis written by Xu Cui entitled “Time series analysis of bat ultrasound signals”. I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Ecology and Evolutionary Biology.

Louis Gross
Major Professor

We have read this thesis
and recommend its acceptance:

Gary McCracken

James Drake

Acceptance for the Council:

Anne Mayhew
Vice Provost and Dean of Graduate
Studies

(Original signatures are on file with official student records.)

Time series analysis of bat ultrasound signals

A Thesis Presented for the Master of Science Degree

The University of Tennessee, Knoxville

Xu Cui
December, 2002

DEDICATION

This thesis is dedicated to my father

Junjie Cui

ACKNOWLEDGEMENTS

I would like to extend deepest thanks to my major professor, Dr. Louis J. Gross, for his advice, support and guidance over the past two years. I would also like to thank the other members of my committee, Dr. Gary McCracken and Dr. James Drake, for their suggestions regarding this work.

A special thanks belongs to my family and friends who have supported me throughout my life.

ABSTRACT

The general objectives of this study are to develop new methods to analyze ultrasound signals recorded during field observations of bats, and relate these to bat biology. In particular, I investigate methods to determine whether or not sound pulses produced by multiple bats are independent of each other when the bats are searching for food in the same environment. Four models are proposed to explain the time series of recorded signals and it is found that the Poisson model, which assumes pulses are independent, fits the data best. A computer program, BatCount was developed as part of this study, to provide an automated method to count bat pulses and to allow comparisons of signals recorded during different observations. This program also included methods to compare the different models developed to the signals as part of the data analysis.

TABLE OF CONTENTS

Chapter		Page
1	INTRODUCTION	1
2	THE BATCOUNT PROGRAM	4
	2.1 The functions of the BatCount program	4
	2.2 Development tools	4
	2.3 How to use the program	5
	2.3.1 To open data files	5
	2.3.2 To set counting parameters	5
	2.3.3 To count number of pulses and feeding buzzes	5
	2.3.4 To obtain the information rate using Strong's method	5
	2.3.5 To create artificial data file according to models and to obtain basic statistics	6
	2.3.6 To calculate the information rate using Gross's method	6
	2.3.7 To calculate the information rate of individual bats of the artificial file	6
	2.4 Inside the program --- methods and algorithms	6
	2.4.1 Wave format	6
	2.4.2 Algorithms	7
	2.4.2.1 Reducing noise	7
	2.4.2.2 Counting pulses and feeding buzzes	8
	2.4.2.3 Obtaining the interval distributions and the pulse duration distributions	8
	2.4.2.4 Calculating the information flow using Strong's method	8
	2.4.2.5 Calculating the information flow using Gross's method	9
3	DATA ANALYSIS AND MODELS	10
	3.1 Data analysis	10
	3.1.1 The data	10
	3.1.2 Data analysis	10
	3.2 Models and data fitting	13
	3.2.1 Models	13
	3.2.2 The information rate	14
	3.2.3 The interval distribution	16
	3.2.4 Conclusion	17
	REFERENCES	20
	APPENDIX	22

I The format of WAVE file	23
II Figures	26
III Tables	60
IV User's guide of the BatCount program	65
VITA	69

LIST OF FIGURES

Figures		Page
2-1	The appearance of the BatCount program.	27
2-2	The file open dialog.	28
2-3	The parameter setting dialog.	29
2-4	The dialog for setting the output parameters.	30
2-5	The dialog for setting the parameters for calculating the information flow using Strong's method.	31
2-6	The dialog for creating the artificial data file and obtaining basic statistics.	32
2-7	The dialog for obtaining the interval statistics for calculating information flow.	33
2-8	The dialog for calculating the information rate using Gross's method.	34
2-9	The dialog for obtaining the interval statistics of a single bat for calculating information flow.	35
2-10	The original data.	36
2-11	The absolute time series	37
2-12	The time series after the treatment of the method of moving average.	38
2-13	(Above) Noise level (the cyan line) detected. (Below) Threshold (cyan) set by users after noise is removed (black).	39
2-14	Noise-free time series.	40
2-15	The final time series.	41
3-1	The call rate.	42
3-2	The distribution of pulse duration	43
3-3	The distribution of intervals for all the data files	44
3-4	The interval distribution for 10 data files	45
3-5	The relationship between the mean and the standard deviation of the interval distributions for the 10 data files (see Figure 3-4)	46
3-6	The relationship between the call rate and pulse duration	47
3-7	The duration and interval	48
3-8	The call rate and the scaled call rate	49
3-9	The information Rate and the scaled call rate.	50
3-10	The upper (green line) and lower (red line) limit of the information rate	51
3-11	The call rate and the number of bats.	52
3-12	The information rate and the number of bats.	53
3-13	The information rate and the call rate.	54
3-14	The information rate and the call rate	55
3-15	The interval distribution for the Poisson model, with $p=0.05$.	56
3-16	The interval distribution of the periodic model	57
3-17	The interval distribution for the Poisson model with refractory period	58
3-18	The interval distribution of the semi-periodic model.	59

LIST OF TABLES

Tables		Page
3-1	The properties of the 40 data files	61
3-2	Moments of the interval distributions for the 10 data files indicated in table 3-1	62
3-3	The call rate and the information rate for the models.	63
3-4	Comparison of the model distribution with the data distribution.	64

CHAPTER 1 INTRODUCTION

About two hundred years ago, Spallanzani and Jurine found that hearing but not vision was critical for orientation by bats [1]. A hundred years later Maxim speculated that bats might detect obstacles by emitting low-frequency sounds and somehow detecting their reflection from obstacles [2]. Hartridge then suggested that high-frequency sounds were more physically suitable [3]. Hartridge's proposal was proved correct after equipment capable of detecting ultrasound were developed [4].

In fact, all the nearly 760 species of Microchiroptera, including the Mexican free-tailed bat (*Tadarida brasiliensis mexicana*) [5, 6, 7] the species focused upon here, have evolved sophisticated echolocation systems that they use for orientation and obtaining food [8]. When flying, they emit signals of high frequency (mostly ultrasonic) at varying intervals and analyze the returning echoes to detect, characterize, and locate the reflected objects [9]. For an individual bat, the procedure during a typical capture of a food insect, such as the corn earworm [10], includes three phases: searching, approaching and capturing [11, 9]. In searching phases, bats emit pulses at a low rate. When they find an object, they produce pulses more and more frequently to locate and approach it [12, 13, 9, 14]. The series of pulses produced in a search phase is called a feeding buzz.

The properties of such sound pulses are environment dependent [8, 15, 9]. Bats of species that search for food in open space (such as *Tadarida brasiliensis*) produce pulses at a relatively low rate and the pulses are usually constant-frequency (CF) ones [12, 13, 9]. CF means constant sound frequency during the pulse. Most species of Microchiroptera do their hunting in moderately open environments through which obstacles to flight such as trees or buildings are scattered (this includes species such as *Eptesicus fuscus*). These bats produce pulses at high rate and the pulses produced are frequency-modulated (FM). [12, 13] FM means varying sound frequency occurs during the pulse. This is understandable because FM pulses allow bats to gain more precise environment information than CF pulses do [8].

Little is known about how the properties of sound pulses from different bats in the same area affect each other. Do bats modify their pulses in response to the pulses produced by other bats? If so, how do bats partition time and frequency space to avoid signal interference? Do the bats shorten their pulses to accommodate more pulses when the bat number is high? The above questions are of great interest in behavioral ecology. If bat signals are independent, then how can many bats fly in the same area without any collision? If not, then how can they manipulate their signals to minimize interference and avoid colliding with each other and obstacles? Also, it is of interest whether the mechanism used by bats to avoid colliding might be applied to ship systems employing sonar.

An objective of this study is to develop methods that can address some of these issues. As a preview of the main results of this study, I point out in Chapter 3 that bats do shorten their pulse duration when the call rate is high. However, at the same time, bats seem independent of each other with respect to the pulse interval. This thesis consists of three major parts. (1) A computer program, BatCount, is developed to analyze the data and do the simulations based on the several models developed here (Chapter 2). (2) The data are analyzed with respect to pulse duration, the call rate and the interval distribution (Chapter 3). (3) Four models are proposed to explain the data and they are compared (Chapter 3). Finally, I summarize the results of this study and point out some possible future approaches (Chapter 3).

A widely used program for bat sound analysis is BatSound written by Pettersson Elektronik[16]. This program allows users to visualize and hear the sounds from an audio file. It provides a display of the oscillogram as well as the spectrogram. The oscillogram shows the amplitude of the sound over time and the spectrogram shows the sound frequency. The program can also show the power spectrum. However, it doesn't provide an automated method to count the number of pulses. Although it does interval and pulse duration analysis, this is not well developed as seen from the analysis results it generates. A new program called BatCount is described below which was developed to fulfill a variety of functions not well developed in BatSound. The details of the program are described in Chapter 2.

The data used for analysis were collected by Gary McCracken and his coworkers. They set up audio devices above experimental corn and cotton fields near the Frio cave, located in the South Central region of Texas, which holds a maternity colony of approximately 10 million Mexican free-tailed bats in the summer months. The audio devices recorded the ultrasound pulses produced by bats in a digital format that is easily transformed to standard audio computer files for further analysis. The data collected on July 9, 2001 were used in this study because of the low level of noise. Higher levels of noise would decrease the accuracy of the procedures used here. However, the data analysis techniques and models developed in this study should be applicable to many recordings of bats aside from those used during the method development.

Several aspects of the data used in this study should be kept in mind:

- (1) The sound frequency recorded is not the real frequency of sounds produced by the bats. Sound is sampled 11025 times per second during the recording. With this sampling rate, the maximum frequency observable is $11025/2 = 5512.5\text{Hz}$, while the pulses produced by bats are ultrasound and usually fall above 20KHz, which is above the hearing range of most humans. The audio devices lower the frequency by a factor 10 so that the sound becomes audible to humans.
- (2) Pulses produced by one individual bat are almost indistinguishable from ones produced by other bats because they are of the same species.

(3) Not all pulses produced by bats were recorded. The recorder was fixed and bats were flying. Thus some bats were sometimes beyond the range of the recorder. How many pulses were not recorded is unknown.

(4) The number of bats was not observed. The high speed of bats makes it almost impossible for field workers to count them.

Analysis of the data reveals some general patterns of the interval distribution (Chapter 3). The interval distributions for all the long data files have a skewed shape with a single peak near the short-interval end. These general patterns suggest there might be a mechanism underlying the process of bat calls. To reveal the mechanism, four models, the periodic model, the semi-periodic model, the Poisson model and the Poisson model with refractory period, are proposed below to explain the data time series. The goodness of the models is tested in two respects: the information rate and the interval distribution. It is found that the Poisson model is the best model among the four models.

CHAPTER 2 THE BATCOUNT PROGRAM

2.1 The functions of the BatCount program

The level of foraging activity of a bat is manifested by the rate of its calls, which is defined as the number of pulses divided by the time period within which the pulses are made. Additionally, bats produce feeding buzzes which are very rapid (an order of magnitude smaller intervals between pulses than other calls [12]). Counting the number of pulses as well as feeding buzzes is essential for the field research because the bat foraging activity is directly related to the number of pulses and feeding buzzes. Counting calls and estimating the call rate has been done manually previously. There are several disadvantages of hand-counting. First, when the file is large, for example, longer than one minute, the pulse number is so large that hand-counting is very time-consuming. Since the data files are usually several-hours long, hand-counting is not feasible. Second, to distinguish a peak in the time series produced from recording a bat pulse instead of noise is subjective and thus two people may count differently. Third, it is almost impossible to obtain statistics such as the interval distribution from counting large files because the work is too burdensome. The advantage of hand-counting is that it is potentially more accurate. The counter can use frequency information besides sound intensity to distinguish a pulse from its background noise.

The BatCount program has as its goal dealing with the above problems of hand-counting. This program has been updated several times since it was first developed, to increase its functionality. Several components have been included: 1) to count the number of pulses during any given time period from data files; 2) to count the number of feeding buzzes during any given time period; 3) to obtain statistics for pulse duration and the interval between adjacent pulses; 4) to calculate the information flow rate using Strong's and Gross' methods respectively; 5) to simulate the time series of bat calls and to create artificial data files (wave format) according to a user model; 6) to calculate the information flow using Gross's method and obtain statistics of pulse duration and interval of the model.

2.2 Development tools

The BatCount program was originally written in Borland Turbo C++ 3.0 and was a DOS-like program. To enhance usability, it was rewritten in Microsoft Visual C++ 6.0 and became an interface-friendly Windows program (Figure 2-1, All figures and tables are located in the appendix.) which employs numerous classes and functions provided by Microsoft as well as some standard C++ functions. The program currently is not configured to work on other operating systems, though the underlying C++ code could be modified to allow its use under other operating systems.

2.3 How to use the program

The BatCount program is easy to use and the below sections provide a summary of its functionality. A basic user's guide is provided as an appendix to this thesis.

2.3.1 To open data files

To open data files for processing, click File|Open (X|Y means click menu X and then sub-menu Y.). Then the file-opening dialog will pop up (Figure 2-2).

If just one file will be processed, select "Read from a single WAVE file" and enter the file's name. If more than one file will be processed sequentially, then select "Read from a file list" and enter the file name of the list, which is a text file. In such a list, data file names should be saved in the format of the following example:

```
bat_8.wav  
s29.wav  
0541.wav  
jul15top_0145.wav  
end
```

An oscillogram will be displayed if "Show Oscillogram" is selected. Click "OK" when ready. The file(s) will not be processed until Analyze|Count|Run is clicked.

2.3.2 To set counting parameters

Click Analyze|Count|Parameters to set the parameters for the counting portions of the program (Figure 2-3). Values of the parameters chosen critically affect the results and thus should be set with care. The default values were chosen to work well for the data used here, in which the noise level is low and relatively consistent over time. For data sets with higher noise levels, calibrating the parameters using several files for which counts of pulses have been made manually is strongly suggested. The BatSound program, which visualizes the data and shows the frequency information for each pulse, is an appropriate tool to aid the calibration process for the parameters in BatCount.

Click Analyze|Count|Output/SaveOptions to set how frequently you want to save the results (Figure 2-4).

2.3.3 To count the number of pulses and feeding buzzes

Click Analyze|Count|Run to begin to process the file(s). Processing a two-hour-data file requires about 15 minutes for an Intel® Celeron™ processor at 633MHz. The result will be saved in files named by the user.

2.3.4 To obtain the information rate using Strong's method

Click Analyze|InformationFlow|Run to calculate the information rate using Strong's method. Click Analyze|InformationFlow|ParameterSetting to set the parameters (Figure

2-5).

For details about Strong's method and the algorithm, please refer to section 2.4.2.4.

2.3.5 To create an artificial data file according to various models and to obtain basic statistics

Click Model|CreateWaveFile and then a dialog will appear (Figure 2-6). Here the users can set the number of bats, the model type, and the bat properties. Click OK to begin creating the artificial file and obtaining basic statistics of the model.

2.3.6 To calculate the information rate using Gross's method

First, a file containing distances between every adjacent two pulses should be created. Such a file is automatically created when you create the artificial data file (2.3.5). For real data, the file can be created by clicking Model|GetInfoDatafromRealFile (Figure 2-7).

After the file is created, then click Model|CalculateInfoRatefromFile (Figure 2-8). A dialog will pop up. Fill in the file's name and click "Get Information Rate" to calculate the information rate using Gross's method.

For detailed information about Gross's method, please refer to 2.4.2.5.

2.3.7 To calculate the information rate of individual bats from the artificial file

Click Model|GetIndividualInfoRate to create the file(s) containing distances between every adjacent two pulses of a single bat (Figure 2-9). Then refer to the second step of section 2.3.6 to calculate the information rate using Gross's method.

2.4 Inside the program --- methods and algorithms

2.4.1 Wave format

The sound of bat calls is sampled 11025 times per second and the intensity of every sample is stored sequentially in a wave format file. Thus the data can be regarded as a discrete time series with time unit 1/11025 seconds. The detail of the wave format is shown in appendix I. For the objectives of BatCount, high intensity is calculated based upon the absolute value of the data in the wave file. For example, 32767, or 0x7FFF represents the highest intensity, 0 represents silence and -32768 represents negative highest intensity. See Figure 2-10 for an example of the data.

2.4.2 Algorithms

2.4.2.1 Reducing noise

First, those sound intensities whose values are negative are replaced by its absolute value (Figure 2-10 and 2-11). This process modifies the frequency content of the pulses, but since the objective of BatCount is to count pulses and feeding buzzes the loss of frequency content is not of importance

Second, the original time series is smoothed by the method of moving averages (Figure 2-12). Every sample intensity is set to the average of its previous M samples' intensities. Let $f(t)$ be the absolute intensity of a sample at time t , and $m(t)$ be the intensity of sample at time t after transformation, then

$$m(t) = \frac{1}{M} \sum_{i=1}^M f(t-i) \quad (\text{eq-1})$$

where M is the averaging period, a parameter set by users. The moving average method can efficiently smooth the original curve, $f(t)$. For simplicity, $m(t)$ is set to $f(t)$ where $t \leq M$, so the initial portion of the time series is not smoothed..

In C++, the M continuous sample intensities are stored in an array A . The following is the implementation of the moving average.

```
for(i=1; i<=FileLength, i++)          // loop until the end of file is reached
{
    A[i%M] = fgetc(sourcefile);         // read the original time series (saved in
    fputc(average(A), targetfile);      // calculate the average of all the elements of
                                        // the array and save the average to the target
                                        // file
}
```

Third, the noise level is automatically detected (Figure 2-13, the cyan line). In some data, the background noise level varies greatly over time and the program calculates an estimate of the noise level at each time. The noise level here is calculated as the average intensity of the moving average intensity over a user-set period N . i.e.

$$n(t) = \frac{1}{N} \sum_{i=1}^N m\left(\left[\frac{t}{N}\right]N + i\right). \quad (\text{eq-2})$$

where $\left[\frac{t}{N}\right]$ is the rounded integer of $\frac{t}{N}$. For example $\left[\frac{t}{N}\right]=0$ if $t=5$ and $N=20$,

$\left[\frac{t}{N}\right]=1$ if $t=21$ and $N=20$. Fourth, the noise is subtracted from the moving average time series (Figure 2-13). Let $d(t)$ be the intensity of the sample at time t after subtracting the noise, then $d(t) = \max[m(t)-n(t), 0]$. In this time series, the noise level is very low and consistent over time.

Fifth, a threshold set by users is subtracted (Figure 2-13, 2-14). The time series $d(t)$ is not noise free, though the noise level is quite low and consistent over time. Let $u(t)$ be the intensity at sample t after subtracting the threshold set by the user, then $u(t) = \max[d(t) - n_0, 0]$, where n_0 is the noise level set by the user, which is assumed constant over time. The time series $u(t)$ is essentially noise free.

2.4.2.2 Counting pulses and feeding buzzes

The program reads through the modified time series $u(t)$. At the right tail of a peak where $u(t)=0$ and $u(t-1)\neq 0$, the program will calculate (1) the length of the current peak(**pp**), (2) the interval between the current peak and the previous one(**pi**), and (3) the number of intervals which are less than a certain critical value(**fn**) (which is user set). If both **pp** and **pi** are larger than the user-set values, then the current peak is regarded as a bat pulse. If **fn** is larger than the user-set value, the number of feeding buzzes will increase by one. The result is saved automatically every time, with the time period of data storage set by the user.

As seen from the above method, a feeding buzz is a series of pulses with short intervals. The pulses in a feeding buzzes are counted individually instead of just once.

Another time series, $final(t)$, is also created at the same time (Figure 2-15). $final(t)$ is an all-or-none series without noise. In this series, 0 represents silence and 1's, if they are together, represents a pulse.

2.4.2.3 Obtaining the interval distributions and the pulse duration distributions

When counting the pulses and feeding buzzes, the program also saves the interval, **pi**, to result files. The interval and pulse duration distributions can also be obtained by reading through the time series $final(t)$.

2.4.2.4 Calculating the information flow using Strong's method

Strong's method[17], which was originally used to quantify the information flow of neural spikes, is applied here to quantify the information flow of bat calls. Before explaining the method, a couple of concepts need to be interpreted.

Letter length (dt) is the time period during which all values in $final(t)$ will be reduced into one value. The letter length could be regarded as the resolution time period within which values can't be distinguished. Note that with the sampling rate of recording being 11025 per second, $dt = 1$ corresponds to $1/11025 = 0.00009$ second.

Word length (L) is the number of letters in a word.

First, the 0-or-1 time series $final(t)$ is transformed to another time series $bin(t)$ according

to the letter length. If there exists a value within the letter length which is 1, then the reduced value in $bin(t)$ is set to 1; otherwise it is set to 0. The letters are then grouped into words. In the following example, the letter length $dt = 5$ and the word length, $L = 3$.

$final(t)$ 000000001110000000000000000011111111111000000000000001111110
 $bin(t)$ 0 1 1 0 0 1 1 1 0 0 1 1

Second, the frequency of word occurrence is counted. In the example above, the word 011 appears twice, 001 once and 110 once. Thus the occurrence frequency of the above three words, p_1 , p_2 and p_3 , are 0.5, 0.25 and 0.25 respectively.

Third, the total word entropy is calculated as

$$H(dt, L) = -\sum_{i=1}^n p_i \log_2 p_i \quad (eq-3)$$

where n is the number of different words. For the above example,

$$H(5, 3) = -(0.5 \log_2 0.5 + 0.25 \log_2 0.25 + 0.25 \log_2 0.25) = 1.5$$

Then the entropy rate is defined as

$$R(dt) = \lim_{L \rightarrow +\infty} \frac{H(dt, L)}{Ldt} \quad (eq-4)$$

In the C++ code developed for this project, the most difficult part of calculating the entropy rate is to count the frequency of word occurrence. A binary tree is created to save the words and the occurrence of each. Operations on trees, for example, adding, sorting and removing, is about 10 times faster than in the list data structure.

2.4.2.5 Calculating the information flow using Gross's method

A simple but reasonable method to study the information context of bat calls was suggested by Professor Gross. Suppose a pulse is associated with a certain amount of information. Call this I_0 . The value of I_0 can't be obtained from this method itself. For convenience, we assume $I_0 = 1$. There is a reduction (R) of this information if this pulse is too close to other pulses. Assume $R = R(d_1, d_2, \dots)$, where d_i is the time distance of the pulse to other pulses. Once the above function is determined, the information rate of a given call series can be calculated.

The form of the R function is critical in this problem. The simplest function, $R = \exp(-d)$, is employed here. Thus the remaining information of a pulse is

$$I = \prod_i (1 - e^{-d_i}) \quad (eq-5)$$

where d_i is the distance of other pulses to the focal pulse.

Theoretically the interference from all the other pulses in the time series should be considered. However, since $\exp(-50)$ is very small, only the 100 adjacent pulses, 50 on the left and 50 on the right, of the focal one are taken into account here.

CHAPTER 3 DATA ANALYSIS AND MODELS

3.1 Data Analysis

3.1.1 The data

Data recorded on July 17, 2001 were used for the analysis presented here because these data have less noise compared to other available data. This includes a total of 36-minutes of data, gathered intermittently from 3:30 to 6:25 and 21:40 to midnight. These include 40 separate files and 20896 bat calls in total. The length of the 40 data files ranges from 2.4 to 181 seconds, with 30 seconds the average length. No data are available from 6:30 to 21:40.

The data are time series of bat calls (pulses) and are saved in standard audio format (WAVE format). The analysis of the data was mainly done by the BatCount program, Microsoft Excel[18] and the statistics software JMP IN [19].

Some basic properties of the 40 data files, including the file name, file length, time at which the data were recorded and the number of calls contained, are listed in Table 3-1.

3.1.2 Data analysis

(1) The call rate (R)

The intensity of bat activity is manifested by the call rate. The call rate is defined as the number of bat calls divided by the time period within which the calls were made. The call rate is directly related to the number of bats. Some models are proposed later to study the relationship between the bat number and the call rate. However, since the number of bats was not observed, the relationship can't be obtained from the data and the models can't be tested in this respect.

In fact, even if the number of bats were known, the relationship would still be unclear due to the incompleteness of data gathering. The audio recorder was fixed and thus only a portion of all the bat calls are recorded. How many calls are not recorded is unknown.

The overall call rate of the 40 files is 16.4 calls per second. The call rates for each of the 40 individual data files are calculated and plotted against the time of day in Figure 3-1. It is seen that bat activity is much higher around 6:00 in the morning than at other times. The highest call rate is around 30 calls per second. The call rate is much lower in the night, ranging from 2 to 6 calls per second.

(2) The distribution of the call duration

The pulse duration varies for many reasons. The biological reason, which is what we are interested in, is that the bats tend to shorten their pulse duration when the call rate is high. Measurement error provides another reason. When a pulse is weak or the background noise is high, the duration of this pulse is underestimated compared to its “true” duration.

The distribution of the durations of the total 20896 pulses is plotted in Figure 3-2. The distribution is highly skewed, with shorter pulses more frequent. The mean is 5.13 milliseconds and the standard deviation is 5.20 milliseconds. The longest and shortest pulses are 76.1 and 1.0 milliseconds, respectively, while 99.5% of pulses are shorter than 22.9 milliseconds and 50% are pulses are shorter than 2.8 milliseconds.

(3) The distribution of the intervals

The distribution of the intervals between every two adjacent calls is plotted in Figure 3-3. The distribution is also highly skewed, with shorter intervals more frequent. The mean is 55.4 milliseconds, about 10 times the mean duration. The standard deviation is 78.4 milliseconds. The longest and shortest intervals are 2174 and 1.0 milliseconds, respectively, with 99.5% of the intervals shorter than 477 milliseconds and 50% of intervals shorter than 35.6 milliseconds.

Interval distributions for the 10 individual files indicated in Table 3-1 are also plotted (Figure 3-4). Major moments of the distributions are listed in Table 3-2.

General patterns are easily found for the 10 interval distributions in Figure 3-4. (1) The shapes of the distribution are highly skewed; (2) There is a peak close to the shortest interval; (3) Each of the distributions has a long tail.

It should be noted that there is a strong correlation between the mean and the standard deviation of the interval distribution (Figure 3-5). The correlation is 0.975 and the fitted equation is:

$$Deviation = 10.5 + 0.712 Mean \quad (eq\ 3-1)$$

The general patterns and the strong correlation between the mean and standard deviation suggest there is a common mechanism underlying the bat call processes. Several models are proposed later to evaluate alternative explanations for these patterns.

(4) The pulse duration, the call rate and the interval

The relationship between the call rate and the duration of pulses is very important because it can help to determine if bats shorten their call durations when the call rate is high. For each of the 40 data files, an average of the durations of the calls and a call rate were calculated. The scatter plot of the logarithm of the averaged duration and the call rate is shown in Figure 3-6. The correlation between the two variables is -0.877 ($p < 0.001$).

The best-fit line has equation

$$\ln(\text{Duration}) = 17.4 - 0.71 \text{ Call Rate} \quad (\text{eq 3-2})$$

This strongly supports the conclusion that bats shorten their pulse duration when the call rate is high.

The relationship between the duration of a single pulse and the distance to the following pulse was also studied (Figure 3-7). It turns out that there is a weak correlation (-0.349, $p < 0.0001$) between the two variables. The negative correlation also indicates that when the call rate is high, bats may shorten their pulse duration.

(5) Scaled call time series and call rate (R_s)

The scaled time series is the original time series compressed by a factor that is the average pulse duration. Such a transformation is necessary to ensure the comparability between the data time series and the time series produced by the models proposed later. The model time series are discrete and the pulse duration is fixed to one time unit.

The scaled call rate (R_s) is the product of the original call rate (R) and the duration average. A scatter plot of R_s and R is shown in Figure 3-8. The correlation between the two variables is 0.765 ($p < 0.0001$)

(6) Information rate

Suppose a pulse is associated with a certain amount of information I_0 (without loss of generality, I_0 is assigned to be 1). There is a reduction of this information due to the interference of other pulses and the remaining information J is.

$$J = J(d_1, d_2, \dots, d_{n-1}), \quad (\text{eq 3-3})$$

where d_i is the distance of the focal pulse to pulse i and n is the total number of pulses in the time series.

Once the above function J is determined, we can calculate the information rate of a given call series. A simple function is chosen for J :

$$J = \prod_{k=1}^{n-1} (1 - e^{-d_k}) \quad (\text{eq 3-4})$$

And the information rate I is the sum of the information contained in all the pulses divided by the time period over which the pulses are produced.

$$I = \frac{\sum J}{T} \quad (\text{eq 3-5})$$

The above method to calculate the information rate is called Gross's method. The information rate is calculated for each of the 40 scaled data time series. The information rate is plotted against the scaled call rate in Figure 3-9. It is clearly seen that the information rate increases almost linearly with increasing scaled call rate. Further discussion of the information rate follows later.

3.2 Models and data fitting

3.2.1. Models

Four models are proposed to evaluate alternative possible underlying mechanisms of bat calls. The models are evaluated using two criteria: 1) Whether or not the interval distribution of the model fits the distribution from the data; 2) Whether the information rates of the models are consistent with that from real data.

The four models proposed are: Periodic call model, Semi-Periodic call model, Poisson model and Poisson model with refractory period. All the models are time-discrete models with time unit $dt = 1$.

(1) Periodic call Model

This model assumes that every bat produces pulses periodically with period T_i . Each pulse occupies one time unit. Since all the bats are of the same species, it is reasonable to assume that every bat has the same call rate. In fact, those bats which produce pulses with higher rate would have a disadvantage with respect to energy use. So we assume $T_i = T$ for every bat. Also, we assume that the position of the initial call of a bat is randomly assigned.

It was found that the interval between two adjacent calls is about 20 times the duration of a call in some data files in which there appeared to be only one bat producing calls. So it is assumed $T = 20$ in the simulation. The initial call begins at a time chosen as a random number following the uniform distribution in the range $[0, T]$.

(2) Semi-Periodic call model

This model also assumes that every bat has an intrinsic call period T . The intervals between two adjacent calls is modified to be $T + d$, where d is a random number following the normal distribution $N(0, s)$. The variance s is assumed to be the same for all bats. The duration of every pulse is one time unit. Also, we assume that the bats evenly partition the time space initially, that is, the initial call of bat i is at time $(i-1)T/n$, where n is the total number of bats.

To keep this model comparable, set $T = 20$ and s is set to 2 in the simulation.

(3) Poisson model

At each time unit dt , the probability to make a call is p_i for bat i . Assume $p_i = p$ since any differences between bats is ignored, implying they are all of the same species with no differences in age, size or sex which would affect call probability.

Here $p = 1/T = 0.05$, where T is the call period mentioned in the periodic call model.

Note that the Poisson model is a memoryless model and bats are independent.

(4) Poisson model with refractory period

This model is almost the same as the Poisson model but with one additional assumption: if a bat produced a pulse at time t_0 , then it can't produce the next pulse before $t_0 + t_r$, where t_r is the refractory period for all the bats. t_r is set to 10 in the simulation.

3.2.2. Information rate of the models

3.2.2.1 The upper and lower limit of the information rate, given a call rate

The information rate is determined by the distribution of calls. It is maximized when the calls are evenly distributed and minimized when they are cluttered together. Note that since all the models are discrete, the upper limit of the call rate is 1.

(1) The upper limit of the information rate

Let x be the distance between two adjacent calls for an evenly distributed call time series. Then the information contained in a single pulse is

$$J(x) = \prod_{k=1}^{\infty} (1 - e^{-kx})^2 \quad (\text{eq 3-6})$$

Thus the information rate is

$$I(x) = \frac{\prod_{k=1}^{\infty} (1 - e^{-kx})^2}{x} \quad (\text{eq 3-7})$$

And the call rate R is

$$R(x) = \frac{1}{x} \quad (\text{eq 3-8})$$

So

$$I(R) = R \prod_{k=1}^{\infty} (1 - e^{-\frac{k}{R}})^2 \quad (\text{eq 3-9})$$

$I(R)$ is plotted against R in Figure 3-10 (the green curve). Note that there is an optimal call rate ($R = 0.563$) which maximizes the information rate (0.362).

(2) The lower limit of the information rate

Given a time period T (long enough) and a certain number of calls N (large enough), the call rate R is N/T . If all the calls are placed together, the information rate will be minimized. Note that if more than one pulse is produced simultaneously, they are not distinguishable and so they may be regarded as one pulse. Thus the minimum distance between two pulses is 1, instead of 0.

The information contained in one pulse is given in eq 3-6 where $x=1$.
 $J_s(1) = 0.2544$

Thus the information rate is

$$I(R) = \frac{NJ_s(1)}{T} = J_s(1)R = 0.2544R \quad (\text{eq 3-10})$$

$I(R)$ is plotted against R in Figure 3-10 (the red line).

3.2.2.2 The information rate of the models

For each of the 4 models, 8 simulations were run for 1, 2, 4, 8, 16, 32, 64 and 128 bats respectively. The results, the information rates and the call rates, are listed in Table 3-3.

(1) The number of bats and the call rate

The call rate for each of the four models increases linearly when the number of bats is small (Figure 3-11). It soon levels off as the bat number increases further. The differences among the four models, especially that between the periodic model, the semi-periodic model and the Poisson model, is not great. The Poisson model with refractory period has the lowest call rate of all the models at any bat number.

(2) The information rate and the number of bats

The information rate for each of the four models increases quickly and reaches its peak (0.31) when the bat number is around 25 (Figure 3-12). It soon decreases a little bit and levels off around 0.25. Also, the differences between the four models are quite small.

(3) The information rate and the call rate

The information rate is plotted against the call rate in Figure 3-13. The upper limit and lower limit of the information rate, calculated in section 3.2.2.1, are also plotted. The information rates of the Poisson model, Poisson model with refractory period, and the semi-periodic model are particularly close. They all increase with decreasing rate, reach their peaks around $R = 0.64$ and decrease until reaching the lower limit. The information rate of the periodic model is a little bit different, due to the randomness of the initial calls.

3.2.2.3 Comparison of the information rate of the models and that from data

The number of bats was not observed when the data were collected and so it is not possible to use the relationship between the call rate and the bat number, and the relationship between the information rate and the bat number to evaluate the models. The relationship between the information rate and the call rate is employed instead in the evaluations.

However, the call rates (scaled) of the data collected are usually very small (<0.15) (See

Figure 3-14). So the data can be compared with the models only over a small range (from 0 to 0.15).

In Figure 3-14, the information rate curve for the Poisson model, Poisson model with refractory period and the semi-periodic model are not theoretically derived but statistically fitted with a quadratic function. For each of the above three models, 4 points have been calculated corresponding to the four different numbers of bats (1, 2, 4 and 8, see Figure 3-13 and Table 3-3). The fits are very good, with all the three R^2 larger than 0.999. The information rate of the periodic model could be derived theoretically for a small number of bats (See section 3.2.2.1).

It is seen from Figure 3-14 that the information rates of the data and of the models are within the limits theoretically calculated. None of the models fit the data well, because the data information rate is always smaller than that of the models. Define the distance (D) between the data and the model to be

$$D = \sum_i [I_d(R_i) - I_m(R_i)] \quad (\text{eq 3-11})$$

where $I_d(R)$ and $I_m(R)$ are the information rate of the data and the model at call rate R . i is the data point. Then the distances between the data and the four models are:

Periodic model: 0.486

Semi-Periodic model: 0.334

Poisson model: 0.197

Poisson model with refractory period: 0.296

The Poisson model is thus the closest to the data under this distance measure, which might indicate that the Poisson model is the closest of these 4 models to the real bat call process.

The consistent deviation of the models from the data indicates that there are some factors not included in the models which are very important. One possibility is that the variance of the pulse duration is important to include.

3.2.3. The interval distribution of the models

3.2.3.1 The interval distribution of the models

For the Poisson model, the interval distribution can be derived analytically. For the other three models, the interval distributions are obtained through simulation.

(1) The Poisson Model

Let p be the probability a call is made at any time unit, then the frequency of the interval with length i is

$$f_1(i) = p(1-p)^i \quad (\text{eq 3-12})$$

For N independent bats, the frequency of the interval with length i is

$$f_N(i) = [1 - (1-p)^N](1-p)^{iN} \quad (\text{eq 3-13})$$

$f_N(i)$ is a decreasing function, which means the most frequently occurring interval is the interval with length 0. See Figure 3-15.

(2) The periodic model

The interval distribution of the periodic call model is shown in Figure 3-16. It is apparent that it doesn't fit the data so the periodic model is not analyzed in the section 3.2.3.2.

(3) The Poisson model with refractory period

The interval distribution of the periodic call model is shown in Figure 3-17. Compared to the data interval distribution, the distributions for this model also have a long tail. But they don't have a peak.

(4) The semi-periodic model

The interval distribution of the periodic call model is shown in Figure 3-18. The distributions have peaks when the bat number is small, but they don't have long tails.

3.2.3.2 Comparison of the interval distribution of the models and that of the data

The ten files indicated in Table 3-1 were chosen for the interval distribution comparison because of their larger sample sizes. The intervals used here are the scaled interval (original interval divided by the average pulse duration). Since the numbers of bats calling in the data files are unknown, the call rate (scaled) is used as the base of comparison. The comparison of data (say, the data recorded at 6:00) with a model (say, the Periodic model) includes three steps. First, the number of bats N (not necessary integer) is calculated from the call rate of the data R_d based on the relationship between the call rate and the number of bats (Figure 3-11). Second, if N is a fraction, say, 1.2, then a new model interval distribution is generated from the distributions of 1 bat and 2 bats by averaging the two distributions. For this case, $f_{1.2}(x) = 0.8 * f_1(x) + 0.2 * f_2(x)$. Third, the goodness of the fit of the model to the data is tested by chi-square statistics.

The result is shown in Table 3-4. It is seen that the Poisson model fits the data best (fitting 5 data files), then the Poisson model with refractory period (fitting 4 data files) and the semi-periodic model has the poorest fit.

3.2.4 Conclusion and future work

As the Poisson model is the closest to the data with respect to information rate and interval distribution, the following conclusion is reasonable based on the properties of the Poisson model: in reality, bats are nearly independent and bat pulses are nearly memoryless. This means that the occurrence of a pulse almost doesn't influence the probability of the occurrence of another call.

This conclusion is very surprising because it is expected that bats would interfere with each other at least to some extent. In fact, the strong correlation between the pulse duration and the call rate does support such interference. However, if duration is not taken into consideration (the models developed here do not do this), the above conclusion is valid.

The conclusion also sheds lights on the understanding of how bats partition the sonic space. The sonic space includes several components such as time (pulse interval), frequency of a pulse, pulse duration and pulse intensity. It is expected that when multiple bats are flying and foraging together, they try to keep as far away as possible in the sonic space to minimize interference. As our models only take pulse interval into account, we can only address the sonic partition problem with respect to the pulse interval. From the point of view of these intervals, bats don't endeavor to keep away from other bats in the sonic space because they are nearly independent.

However, the consistent deviation of the information rate of the data from that of the models and the peaks in the data interval distributions (which are not explained by the Poisson model) indicates the Poisson mechanism is not at all a perfect description of the call process. Other models which take into account the variance of the pulse duration may do a better job.

The effect of feeding buzzes should also be taken into consideration in future work. A feeding buzz is a series of short-duration pulses with short intervals and so it will affect the pulse duration distribution and the interval distribution to a certain extent. In future work, either the effect of feeding buzzes might be removed by truncating the short-duration pulses or newly-proposed models might be developed to include feeding buzzes. Either approach may give a more accurate understanding of the mechanisms underlying bat calls.

To improve our understanding of the relationship between calls among groups of bats, ideally more accurate data collecting should be a high priority. First, new methods should be developed which allow the number of bats studied to be estimated as the calls are recorded. Second, an audio recorder device could be attached to each bat studied. This kind of recorder would make it possible to sift from the data just the sound produced by the bat to which it is attached. Third, another audio recorder could be fixed to record all the pulses produced by all bats studied, just as the data files discussed here were collected. By comparing the data from each bat and the data from all the bats, it would be possible to elucidate the effects of interactions between bats on the complete sequence of calls from a population.

Further improvements could be obtained by applying more limitations to the experiments. Bats could be constrained to flying in a large enclosure and though this would no doubt modify the sonic responses compared to flying in the open, a variety of further information about bat responses to the presence of other bats could be obtained. The number of bats could be controlled allowing the relationship between the calling behavior

and the bat number to be investigated. The environment dependence of bat calling behavior could also be further analyzed by varying the number of obstacles (such as string) and studying how bats change their pulse properties.

The above suggestions would allow more accurate models to be proposed and tested. For example, new individual-based models could be developed which explicitly take into account the interactions between individual bats and their calls.

REFERENCES

- [1] Griffin, D. Return to the magic well: echolocation behavior of bats and responses of insect prey, *Bioscience*, 51(7), 555-556, 2001
- [2] Maxim H. The sixth sense of the bat. Sir Hiram Maxim's contention. The possible prevention of sea collisions. *Scientific American (Supplement, 7 September)*: 148–150, 1912
- [3] Hartridge H, The avoidance of objects by bats in their flight. *Journal of Physiology* 54: 54–57, 1920
- [4] Pierce, G. and Griffin, D. Experimental determination of supersonic notes emitted by bats. *Journal of Mammalogy* 19: 454–455, 1938
- [5] Simmons, J. et al. Echolocation by free-tailed bats (*Tadarida*), *J. comp. Physiol.* 125:291-299, 1978
- [6] McCracken, G.F. Why are we losing our Mexican free-tailed bats? *Bats* 3:1-4, 1986
- [7] Cagle, F.R. A Texas colony of bats, *Tadarida mexicana*. *J.Mamm.* 31: 400-402, 1950
- [8] Kunz, H. and Racey, P., *Bat Biology and Conservation*, Smithsonian Institution Press, 1998 p 181
- [9] Schnitzler, H.U. and Kalko, E. Echolocation by insect-eating bats, *Bioscience*, 51(7), 557-569
- [10] Lee, Y-F.. Resource use and foraging activity of Mexican free-tailed bats, *Tadarida brasiliensis mexicana* (Molossidae). Ph.D. dissertation. University of Tennessee, 130 p. 1999
- [11] Kalko, E. and Schnitzler, H. How echolocating bats approach and acquire food. *Bat Biology and Conservation*, Smithsonian Institution Press, 197-204, 1998
- [12] Simmons, J. et al Echolocation and pursuit of prey by bats, *Science*, 203:16-21, 1979
- [13] Schnitzler, H. and Kalko, E. How echolocating bats search and find food. *Bat Biology and Conservation*, Smithsonian institution press, 183-196, 1998
- [14] Miller, L. and Surlykke, A. How some insects detect and avoid being eaten by bats: tactics and countertactics of prey and predator, *Bioscience*, 51(7): 570-581, 2001
- [15] Fenton, M. The foraging behaviour and ecology of animal-eating bats, *Can. J. Zool.* 68:411-422, 1990
- [16] <http://www.batsound.com/>
- [17] Strong, S. et al., Entropy and information in neural spike trains, *Phys Rev Lett* 80: 197-200, 1998
- [18] Microsoft Excel 2002 (10.2614.2625)
- [19] SAS Institute Inc. JMPIN 4.0.4 (Academic)

APPENDIX

APPENDIX I THE WAVE FORMAT¹

¹ See more at <http://www.technology.niagarac.on.ca/courses/comp630/WavFileFormat.html>

The WAV file consists of three "chunks" of information: the RIFF chunk which identifies the file as a WAV file, the FORMAT chunk which identifies parameters such as sample rate and the DATA chunk which contains the actual data (samples).

Each Chunk breaks down as follows:

RIFF Chunk (12 bytes in length total)

Byte Number	
0 – 3	"RIFF" (ASCII Characters)
4 – 7	Total Length Of Package To Follow (Binary)
8 - 11	"WAVE" (ASCII Characters)

FORMAT Chunk (24 bytes in length total)

Byte Number	
0 – 3	"fmt " (ASCII Characters) ("fmt space")
4 – 7	Length Of FORMAT Chunk (Binary, always 0x10)
8 – 9	Always 0x01
10 - 11	Channel Numbers (Always 0x01=Mono, 0x02=Stereo)
12 - 15	Sample Rate (Binary, in Hz)
16 - 19	Bytes Per Second
20 - 21	Bytes Per Sample: 1=8 bit Mono, 2=8 bit Stereo or 16 bit Mono, 4=16 bit Stereo
22 - 23	Bits Per Sample

DATA Chunk

Byte Number	
0 - 3	"data" (ASCII Characters)
4 - 7	Length Of Data To Follow
8 - end	Data (Samples)

The easiest approach to this file format might be to look at an actual WAV file to see how data are stored. In this case, we examine DING.WAV which is standard with all Windows packages. DING.WAV is an 8-bit, mono, 22.050 KHz WAV file of 11,598 bytes in length.

Let's begin by looking at the header of the file (using DEBUG).

```

246E:0100  52 49 46 46 46 2D 00 00-57 41 56 45 66 6D 74 20  RIFF-..WAVEfmt
246E:0110  10 00 00 00 01 00 01 00-22 56 00 00 22 56 00 00  ..... "V.. "V..
246E:0120  01 00 08 00 64 61 74 61-22 2D 00 00 80 80 80 80  ....data"-.....
246E:0130  80 80 80 80 80 80 80 80-80 80 80 80 80 80 80 80  .....
246E:0140  80 80 80 80 80 80 80 80-80 80 80 80 80 80 80 80  .....

```

As expected, the file begins with the ASCII characters "RIFF" identifying it as a WAV file. The next four bytes tell us the length is 0x2D46 bytes (11590 bytes in decimal) which is

the length of the entire file minus the 8 bytes for the "RIFF" and length (11598 - 11590 = 8 bytes).

The ASCII characters for "WAVE" and "fmt " follow. Next (line 2 above) the value 0x00000010 in the first 4 bytes occurs (length of format chunk: always constant at 0x10). The next four bytes are 0x0001 (always) and 0x0001 (a mono WAV, so only one channel is used).

Since this is an 8-bit WAV, the sample rate and the bytes/second are the same at 0x00005622 or 22,050 in decimal. For a 16-bit stereo WAV the bytes/sec would be 4 times the sample rate. The next 2 bytes show the number of bytes per sample to be 0x0001 (8-bit mono) and the number of bits per sample to be 0x0008.

Finally, the ASCII characters for "data" appear followed by 0x00002D22 (11,554 decimal) which is the number of bytes of data to follow (actual samples). The data values are from 0x00 to 0xFF. In the example above 0x80 would represent "0" or silence on the output since the DAC used to playback samples is a bipolar device (i.e. a value of 0x00 would output a negative voltage and a value of 0xFF would output a positive voltage at the output of the DAC on the sound card).

If it is a 16-bit mono file, then 0x7FFF represents maximum positive voltage, 0x8000 represents maximum negative voltage and 0x0000 represents silence.

Note that there are extension to the basic WAV format which may be supported in newer systems -- for example if you look at DING.WAV in Windows '95 you'll see some extra bytes added after the format chunk before the "data" area -- but the basic format remains the same.

As a final example consider the header for the following WAV file recorded at 44,100 samples per second in 16-bit stereo.

246E:0100	52 49 46 46 2C 48 00 00-57 41 56 45 66 6D 74 20	RIFF,H..WAVEfmt
246E:0110	10 00 00 00 01 00 02 00-44 AC 00 00 10 B1 02 00	D.....
246E:0120	04 00 10 00 64 61 74 61-00 48 00 00 00 00 00 00	data.H.....
246E:0130	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	

Again all the expected structures occur. Note that the sample rate is 0xAC44 (44,100 as an unsigned int in decimal) and the bytes/second is 4 times that figure since this is a 16-bit WAV (* 2) and is stereo (again * 2). The Channel Numbers field is also found to be 0x02 here and the bits per sample value is 0x10 (16 decimal).

APPENDIX II FIGURES

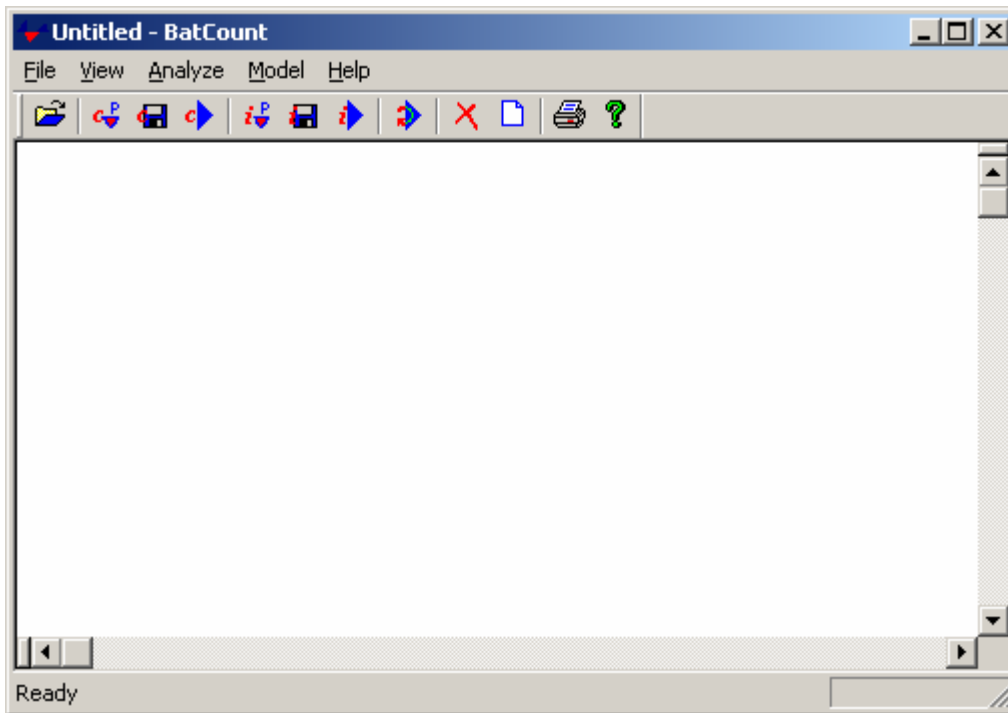


Figure 2-1. The appearance of the BatCount program. The program is an interface-friendly Windows program developed in Microsoft Visual C++ 6.0.

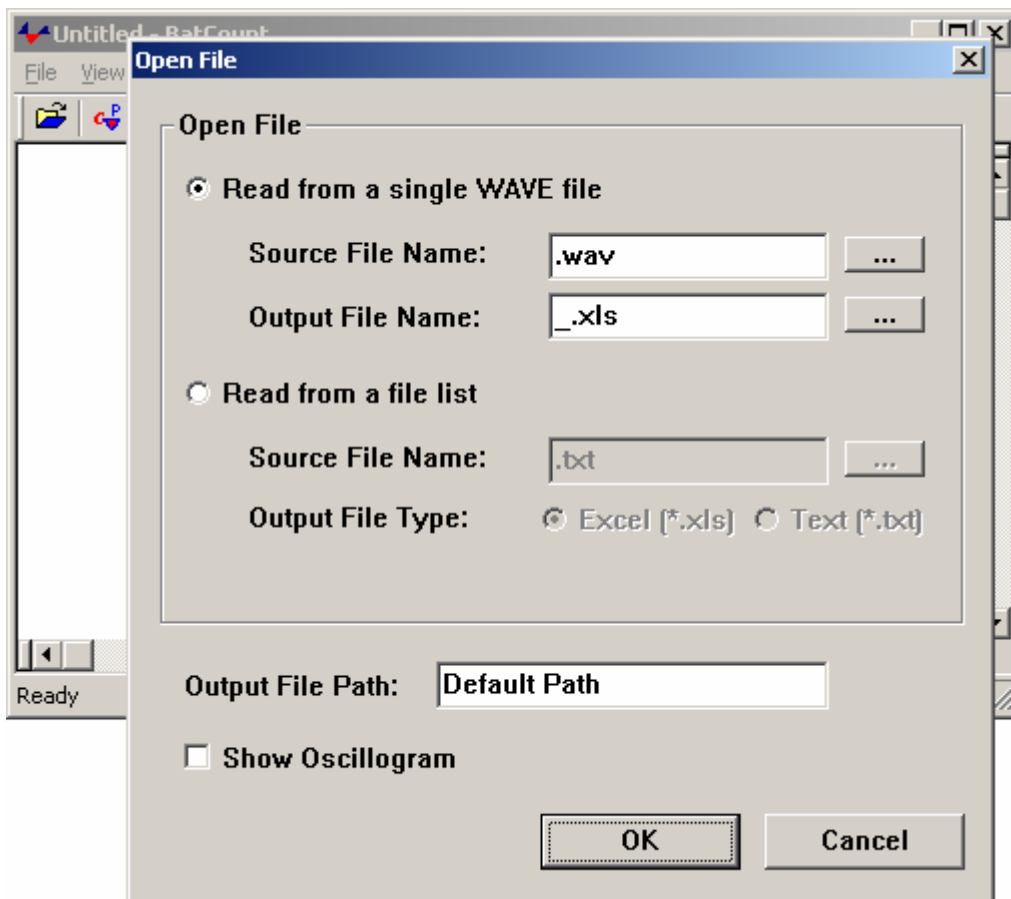


Figure 2-2. The file open dialog.

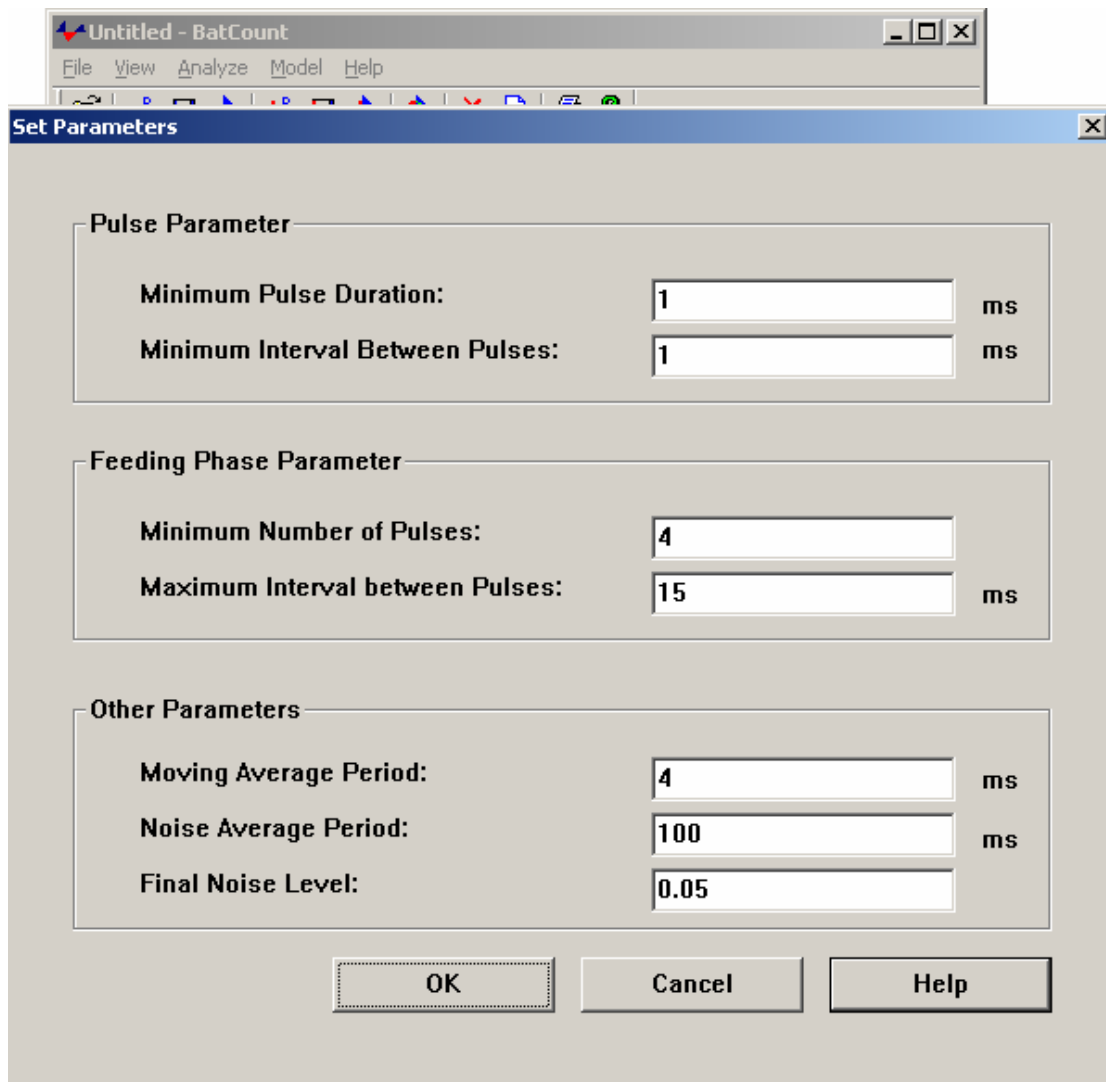


Figure 2-3. The parameter setting dialog.

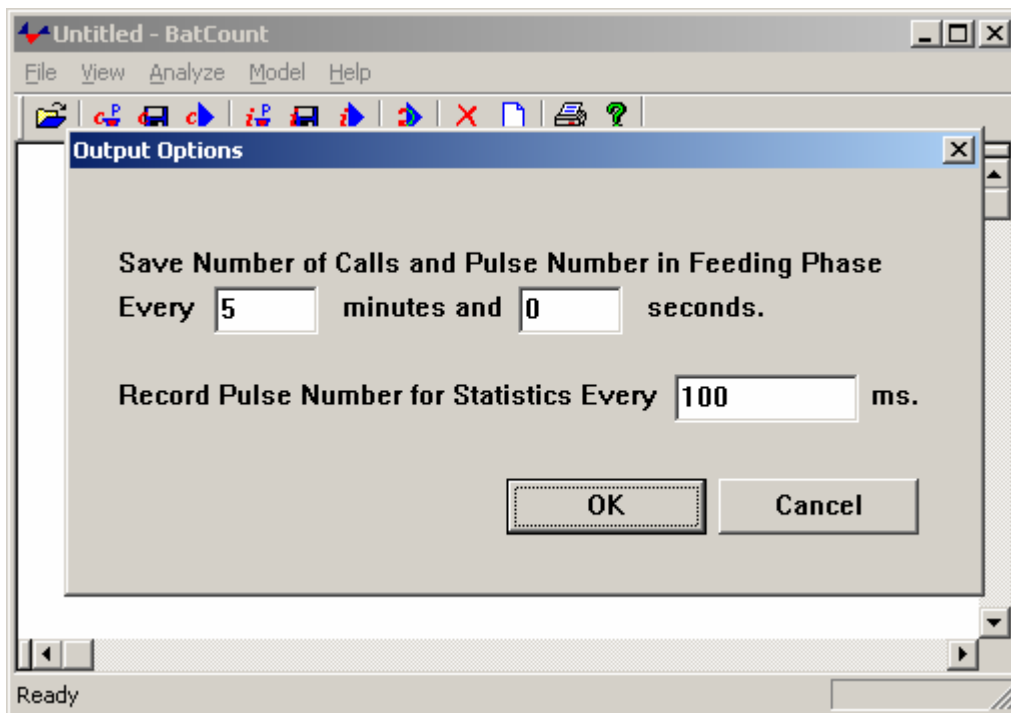


Figure 2-4. The dialog for setting the output parameters.

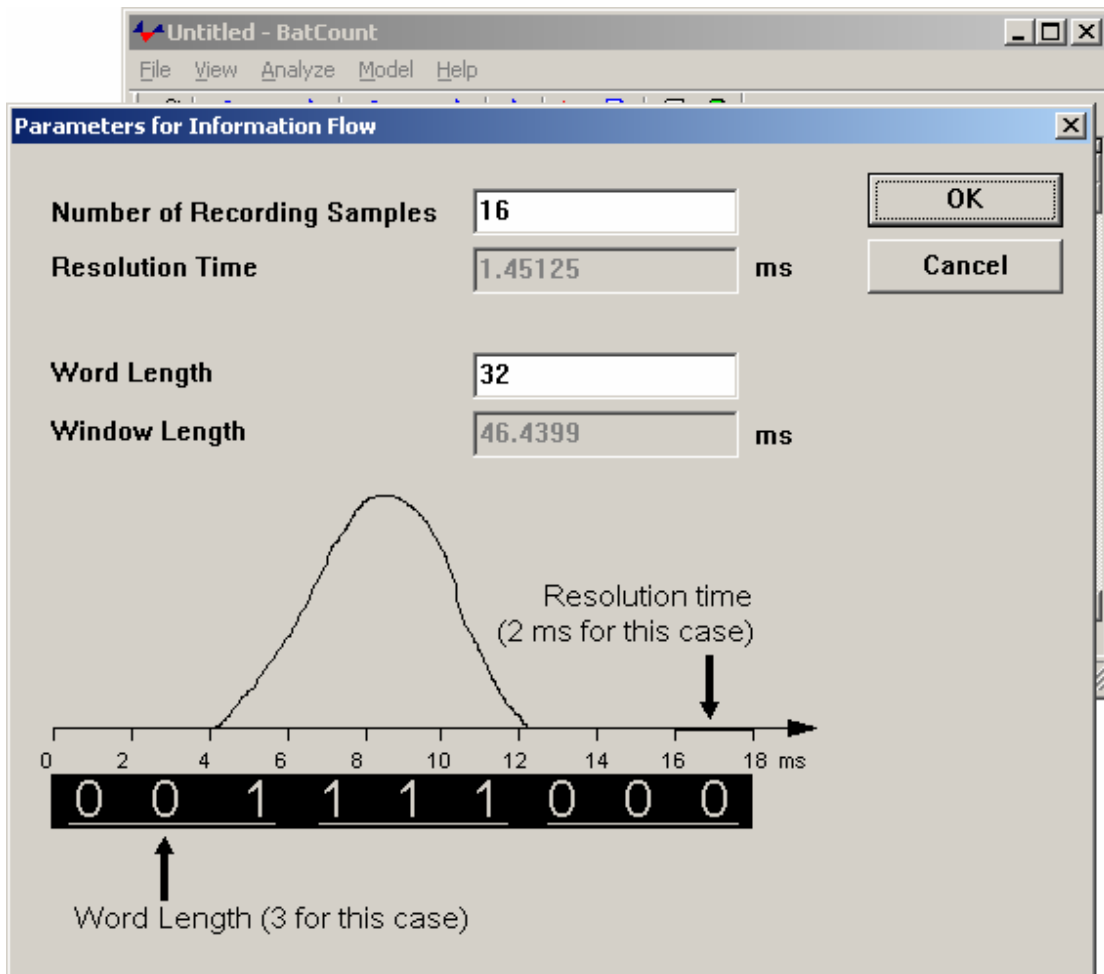


Figure 2-5. The dialog for setting the parameters for calculating the information flow using Strong's method.

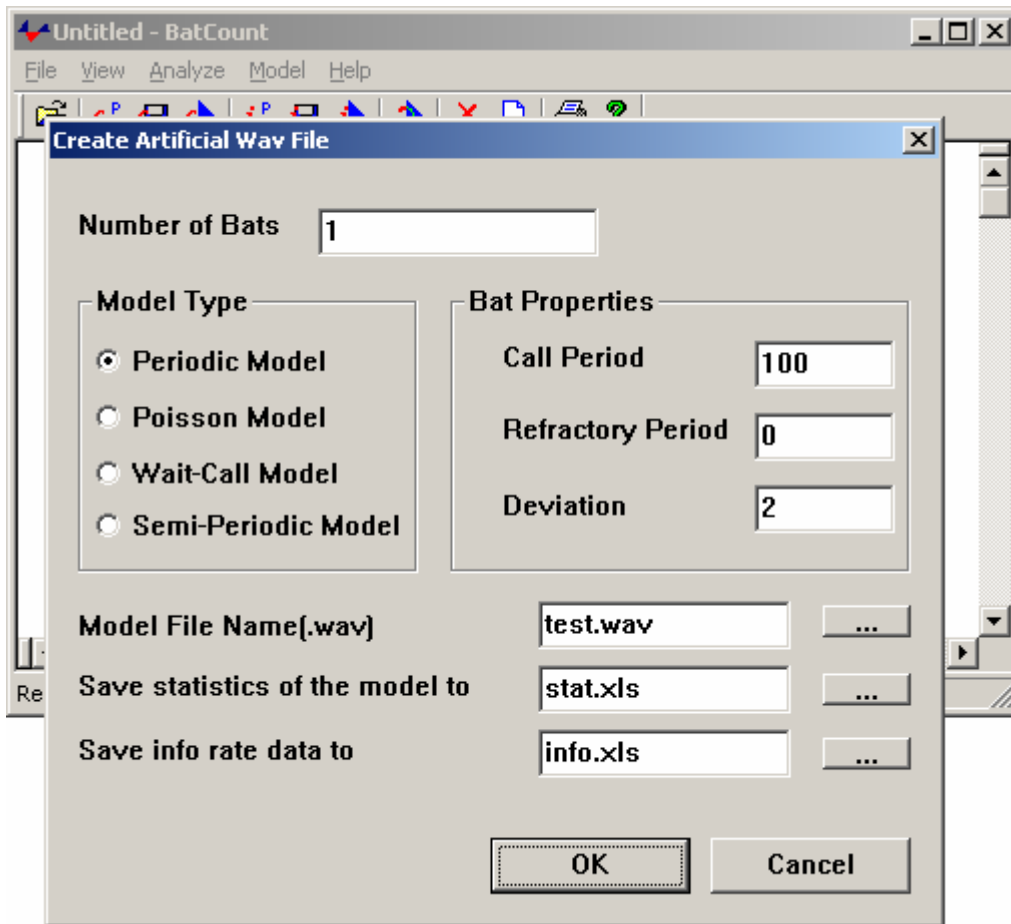


Figure 2-6. The dialog for creating the artificial data file and obtaining basic statistics.

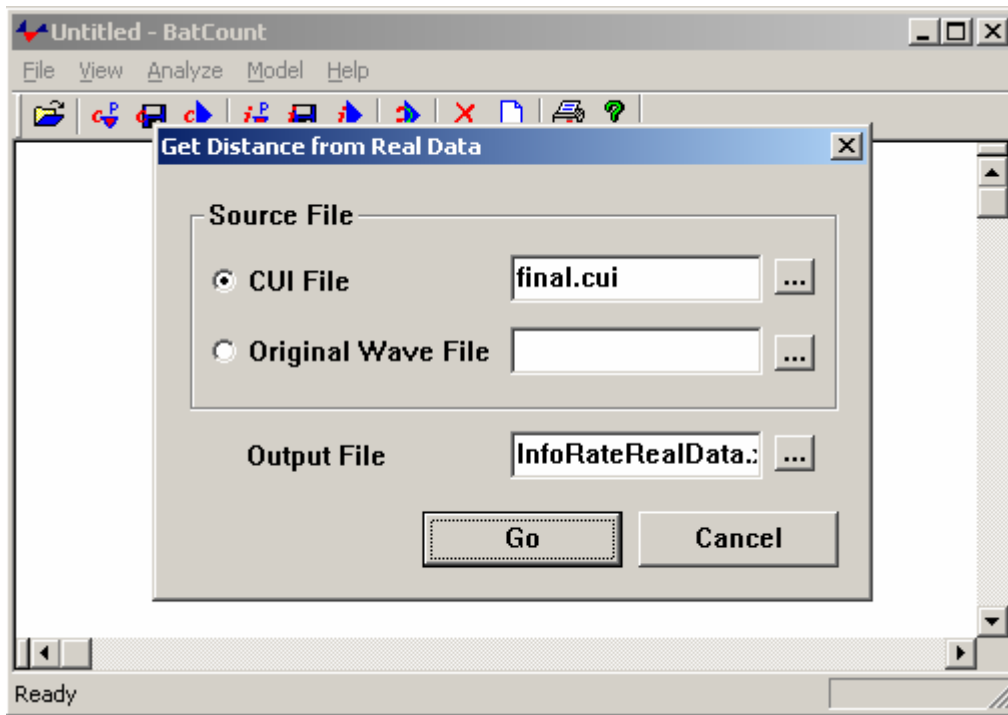


Figure 2-7. The dialog for obtaining the interval statistics for calculating information flow.

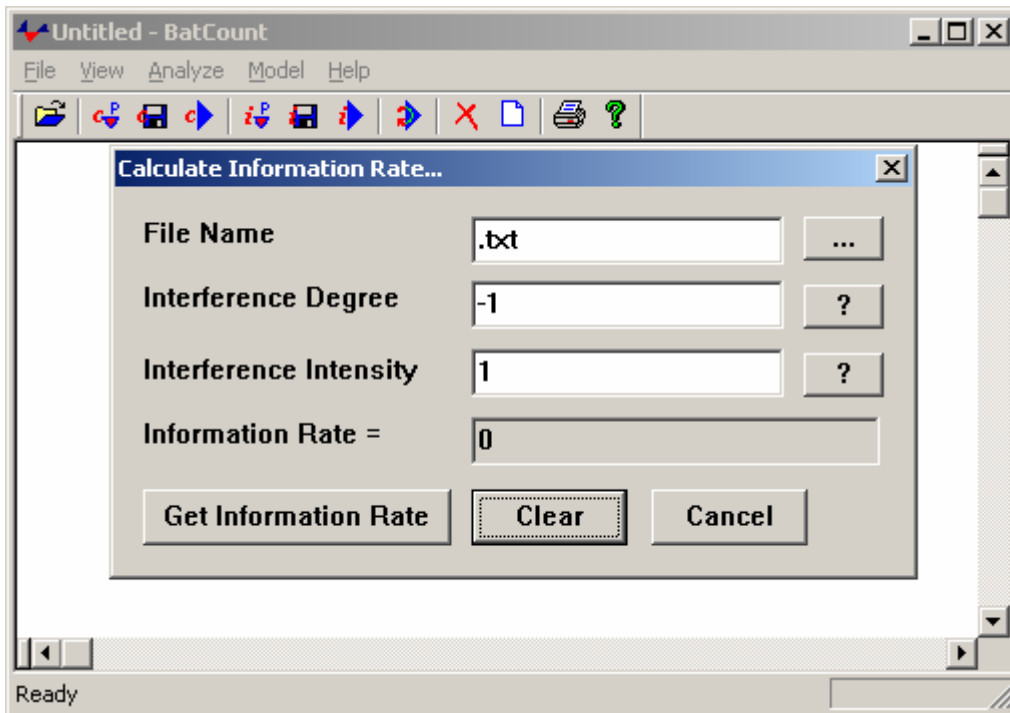


Figure 2-8. The dialog for calculating the information rate using Gross's method.

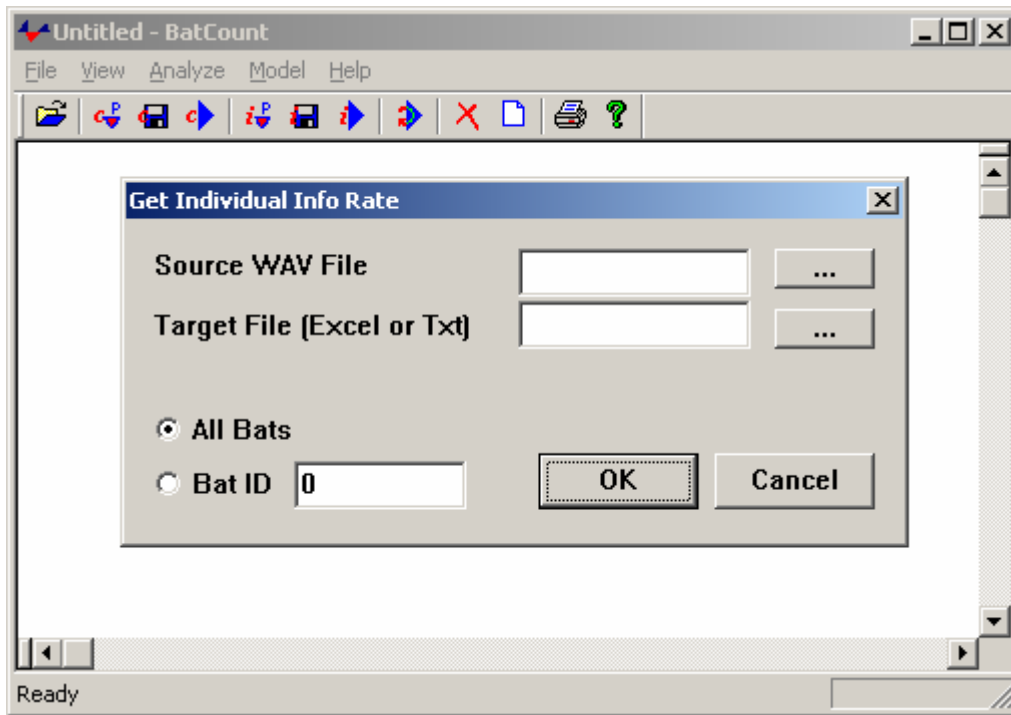


Figure 2-9. The dialog for obtaining the interval statistics of a single bat for calculating information flow.



Figure 2-10. An example original data file.

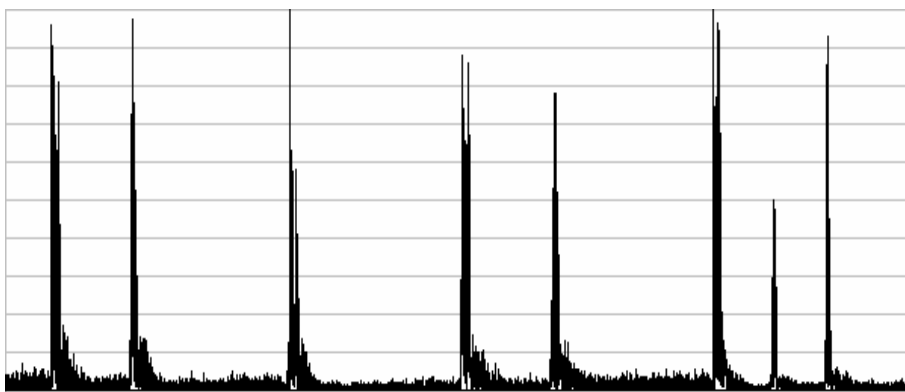


Figure 2-11. The absolute time series. The original intensity is replaced by its absolute value, $f(t)$.

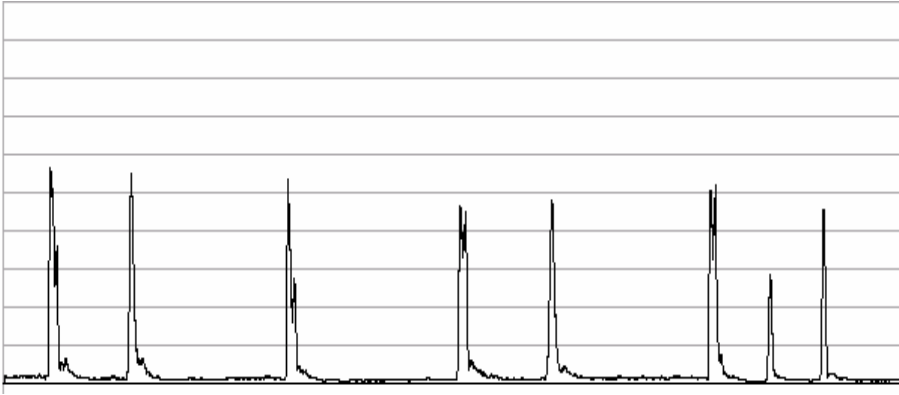


Figure 2-12. The time series after treatment by the method of moving averages.

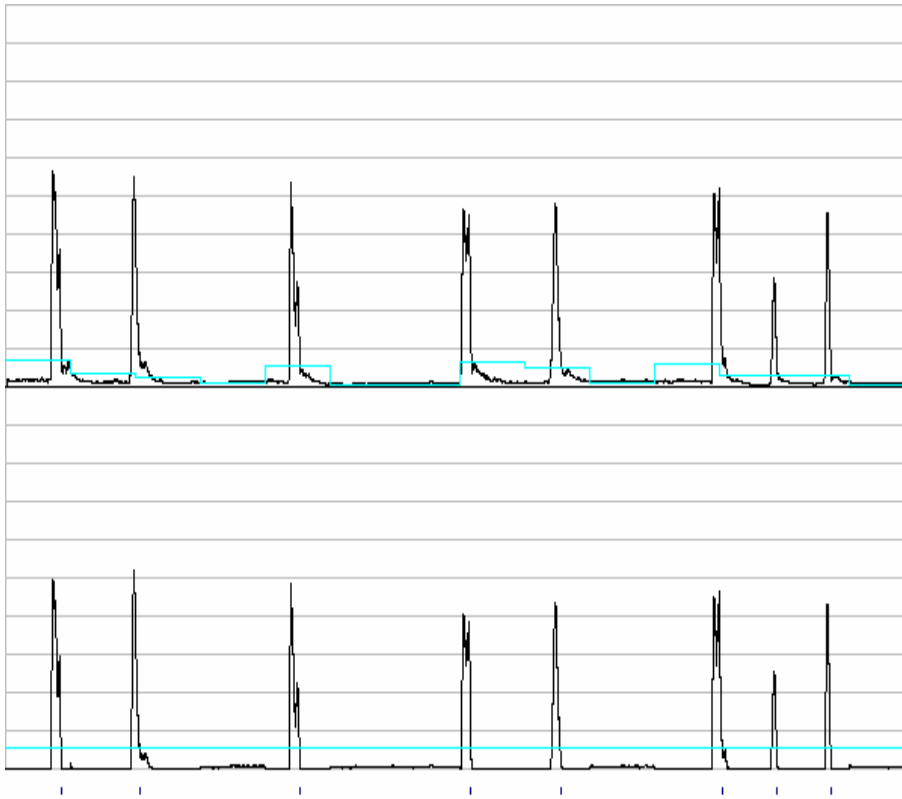


Figure 2-13. (Above) Noise level (the cyan line) detected. (Below) Threshold (cyan) set by users after noise is removed (black).

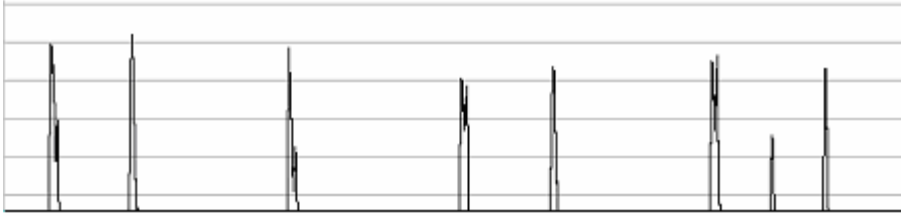


Figure 2-14. Noise-free time series.

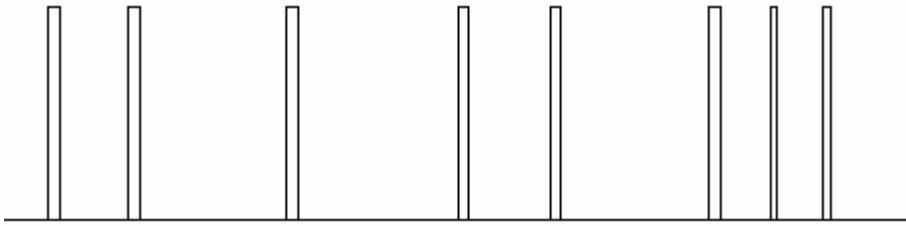


Figure 2-15. The final time series.

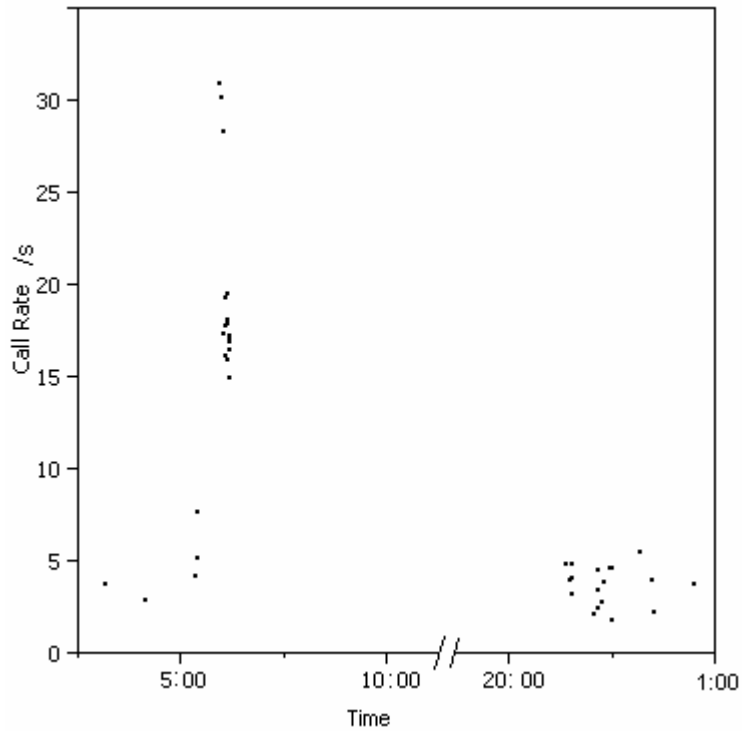


Figure 3-1. The call rate. The call rate reaches its peak (around 30 calls per second) around 6:00 in the morning.

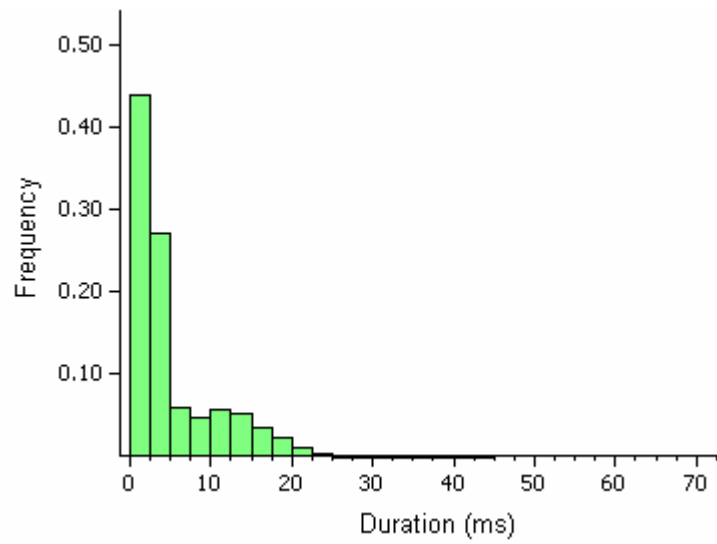


Figure 3-2. The distribution of pulse duration using all the data files.

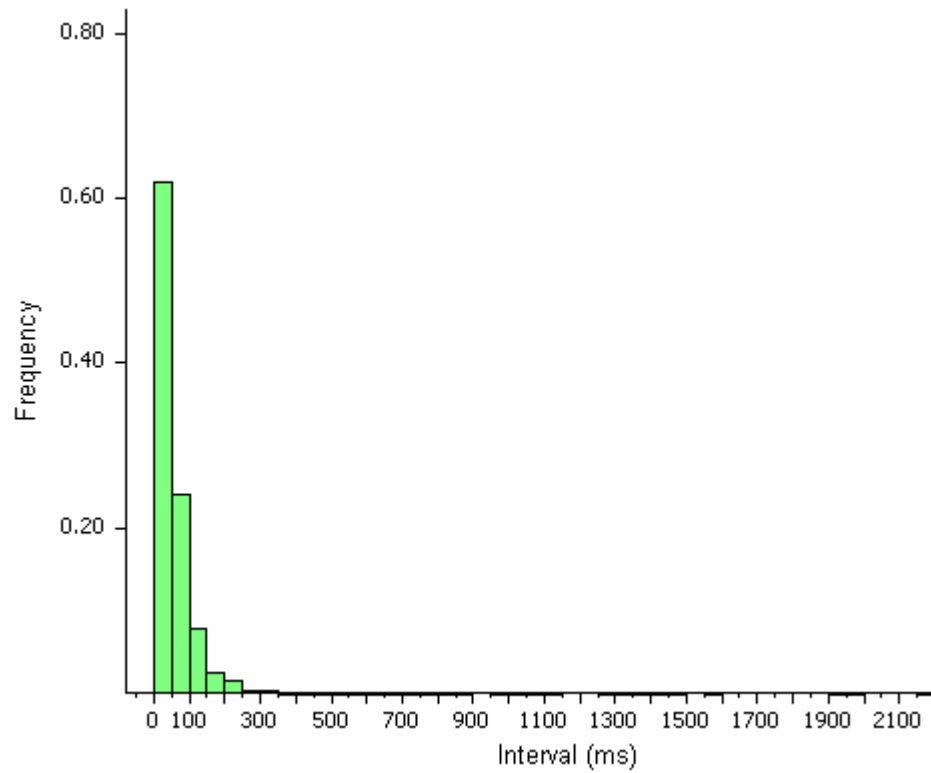


Figure 3-3. The distribution of intervals for all the data files.

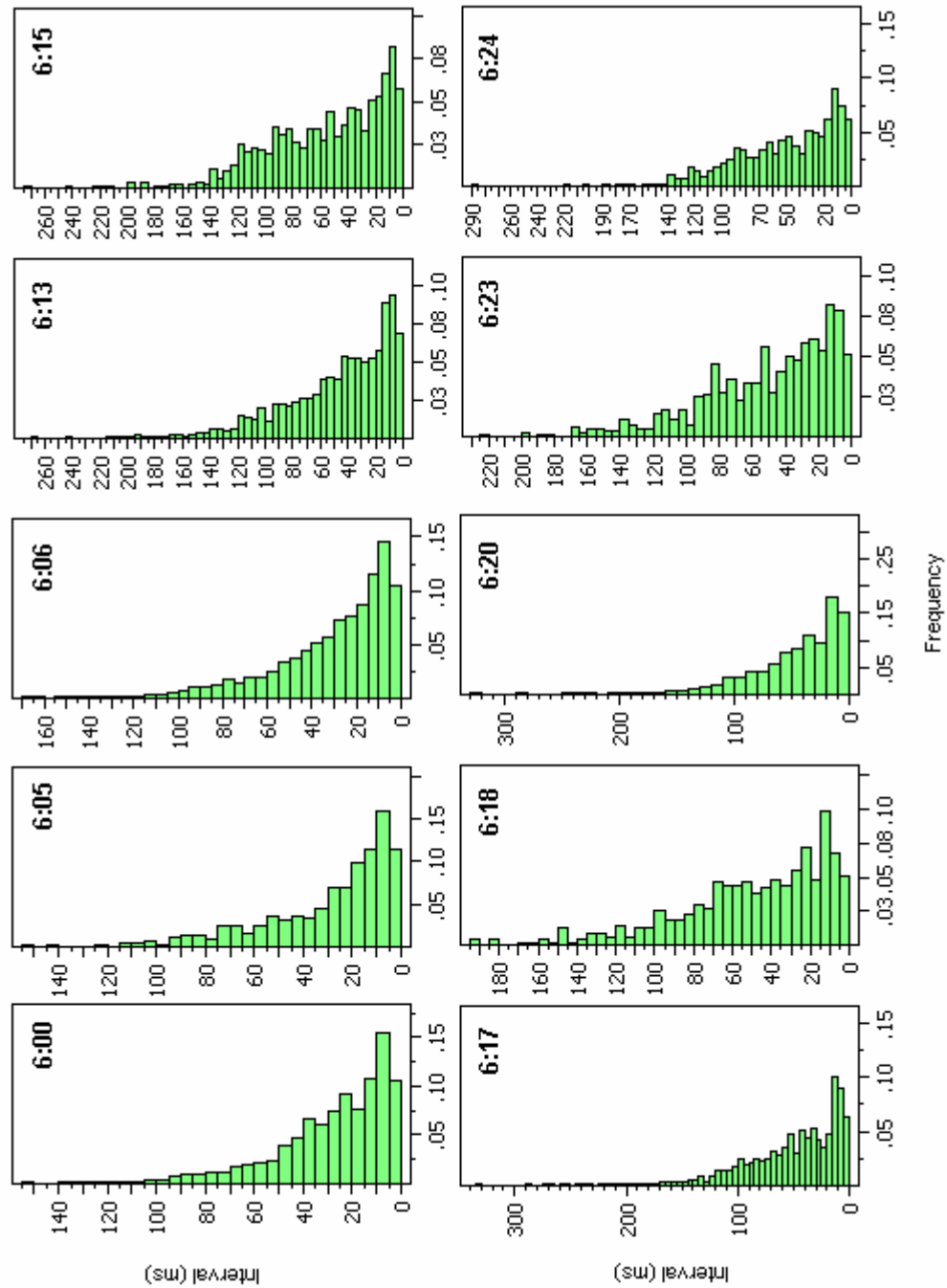


Figure 3-4. The interval distribution for 10 data files. These 10 files were chosen because of their large sample sizes.

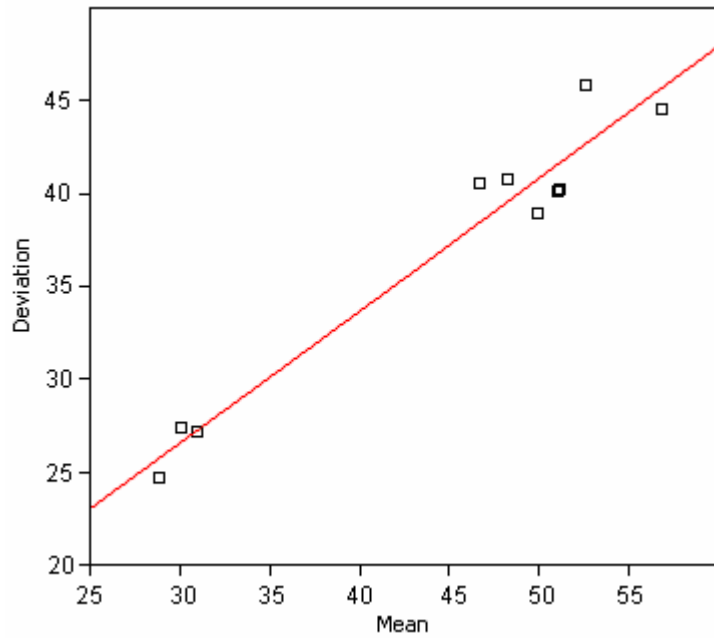


Figure 3-5. The relationship between the mean and the standard deviation of the interval distributions for the 10 data files (see Figure 3-4).

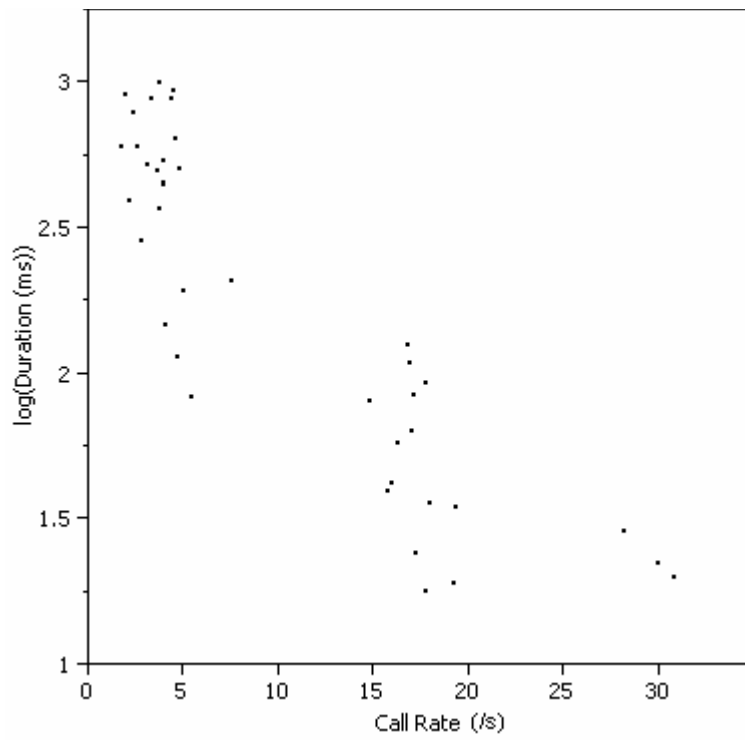


Figure 3-6. The relationship between the call rate and pulse duration.

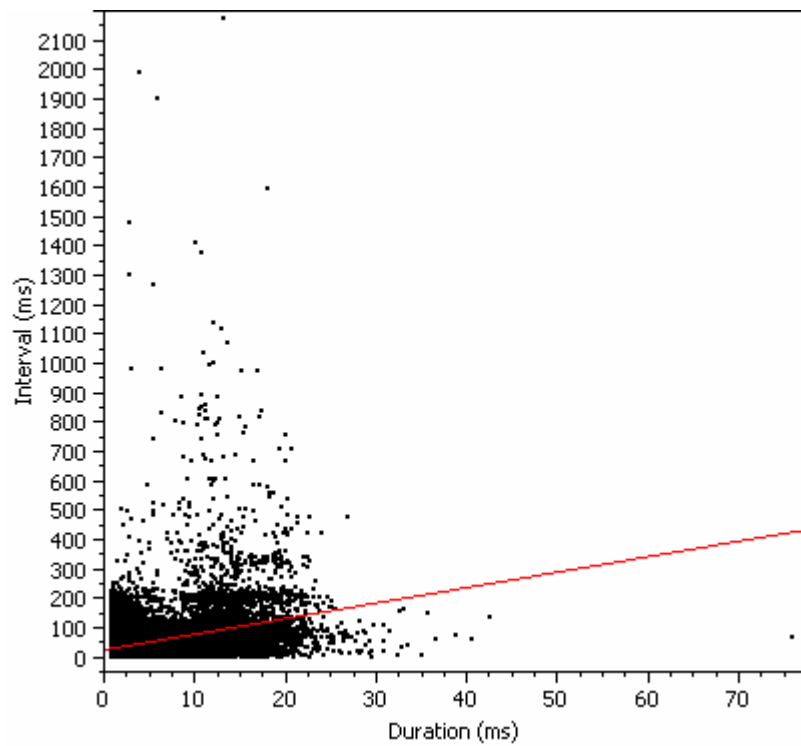


Figure 3-7. The duration and interval. The correlation between the two variables is -0.349.

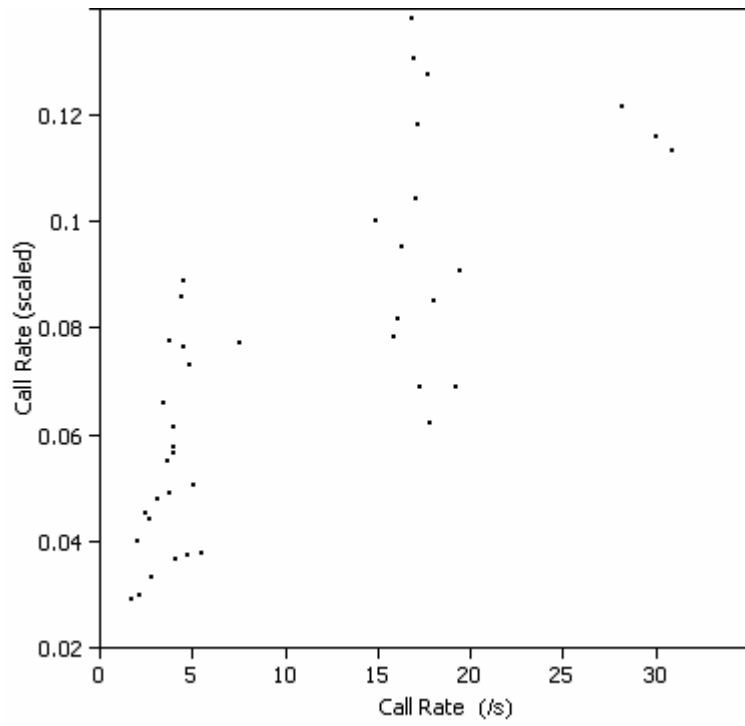


Figure 3-8. The call rate and the scaled call rate.

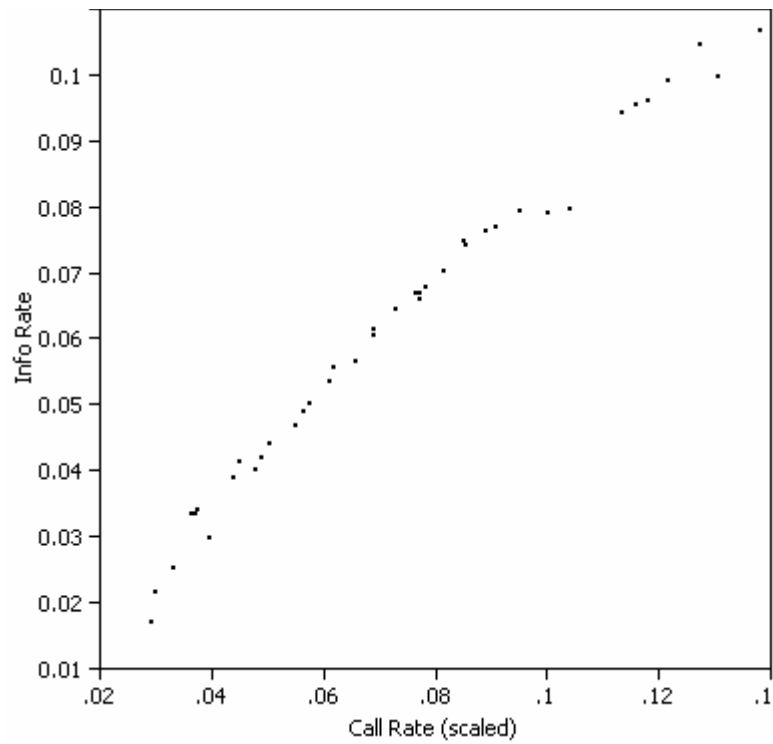


Figure 3-9. The information rate as it relates to the scaled call rate.

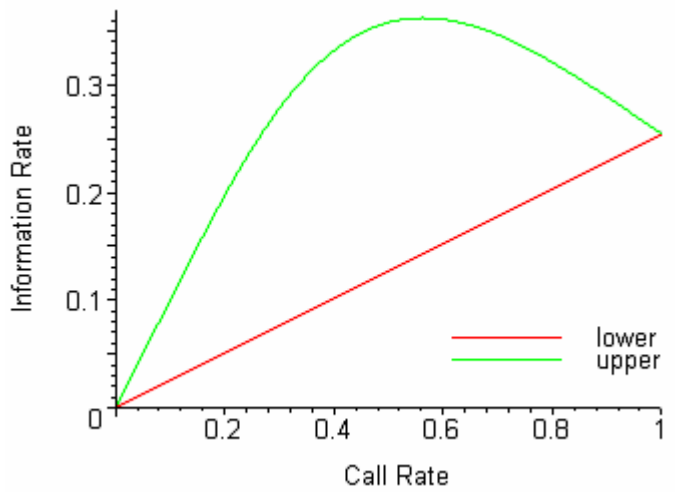


Figure 3-10. The upper (green line) and lower (red line) limit of the information rate.

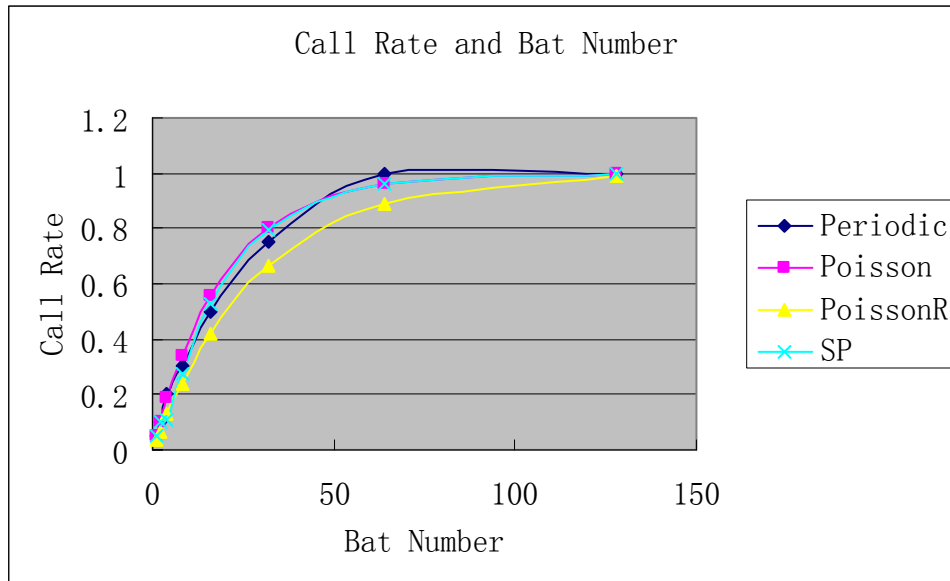


Figure 3-11. The call rate and the number of bats. (PoissonR: Poisson model with refractory period; SP: Semi-periodic model)

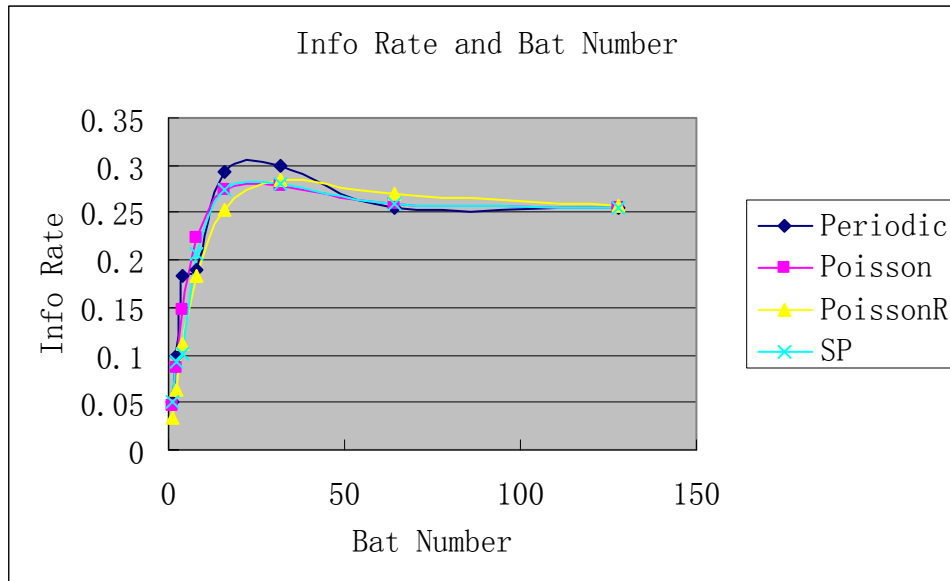


Figure 3-12. The information rate and the number of bats. (PoissonR: Poisson model with refractory period; SP: Semi-periodic model)

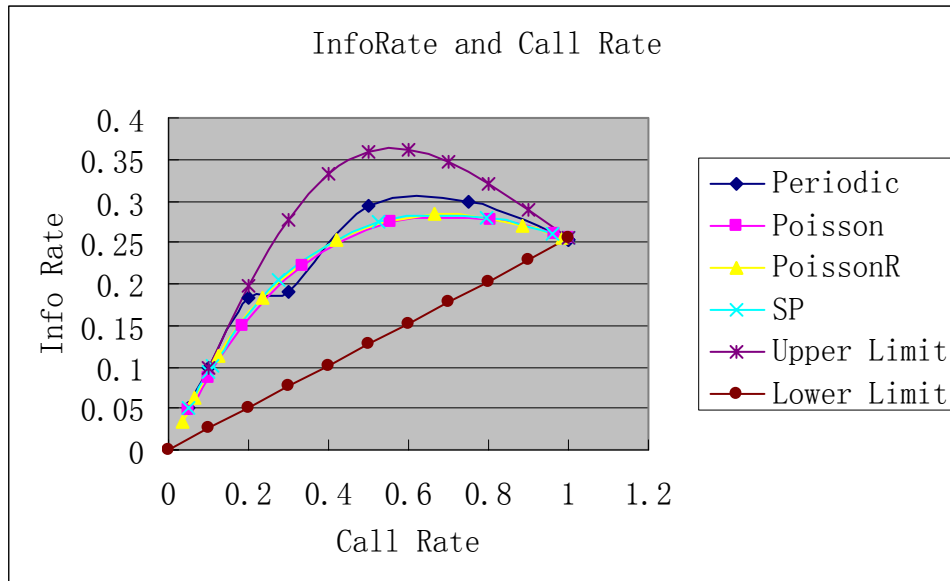


Figure 3-13. The information rate and the call rate. (PoissonR: Poisson model with refractory period; SP: Semi-periodic model)

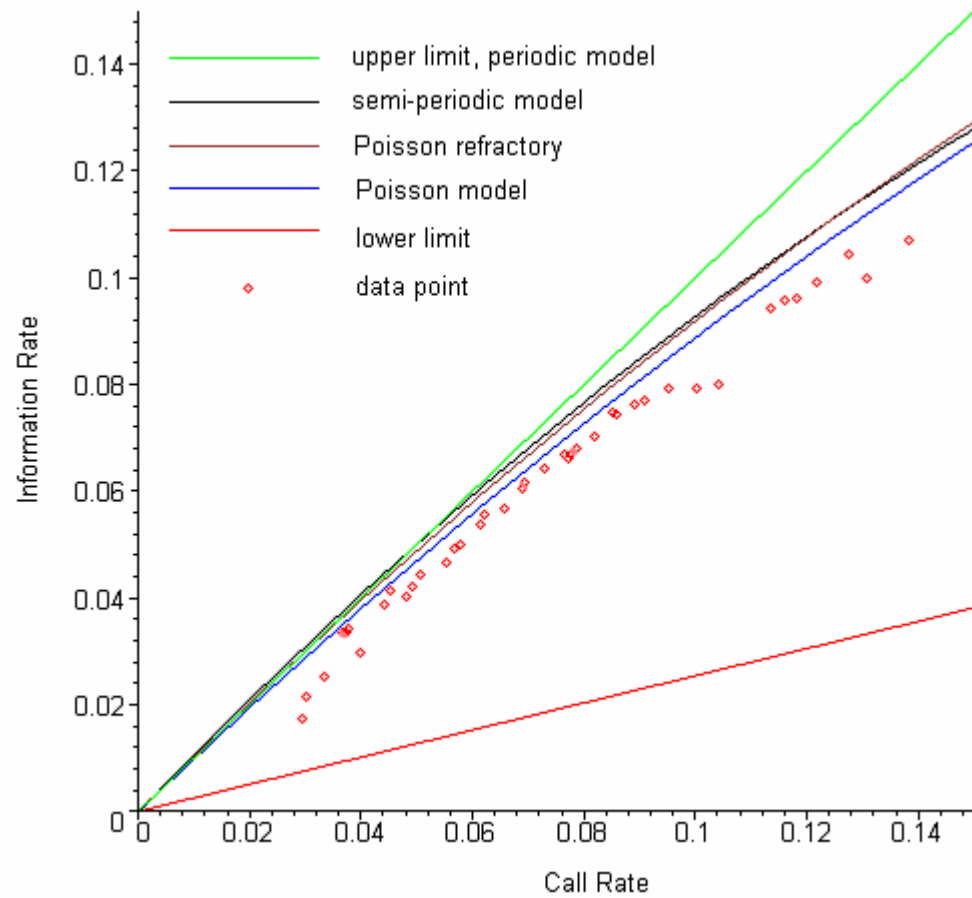


Figure 3-14. The information rate as call rate varies for the models and the data.

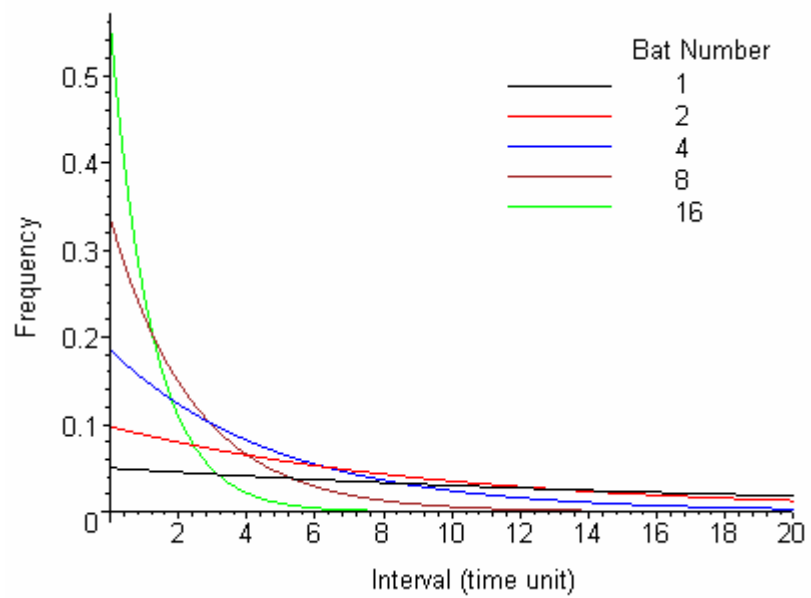


Figure 3-15. The interval distribution for the Poisson model, with $p=0.05$.

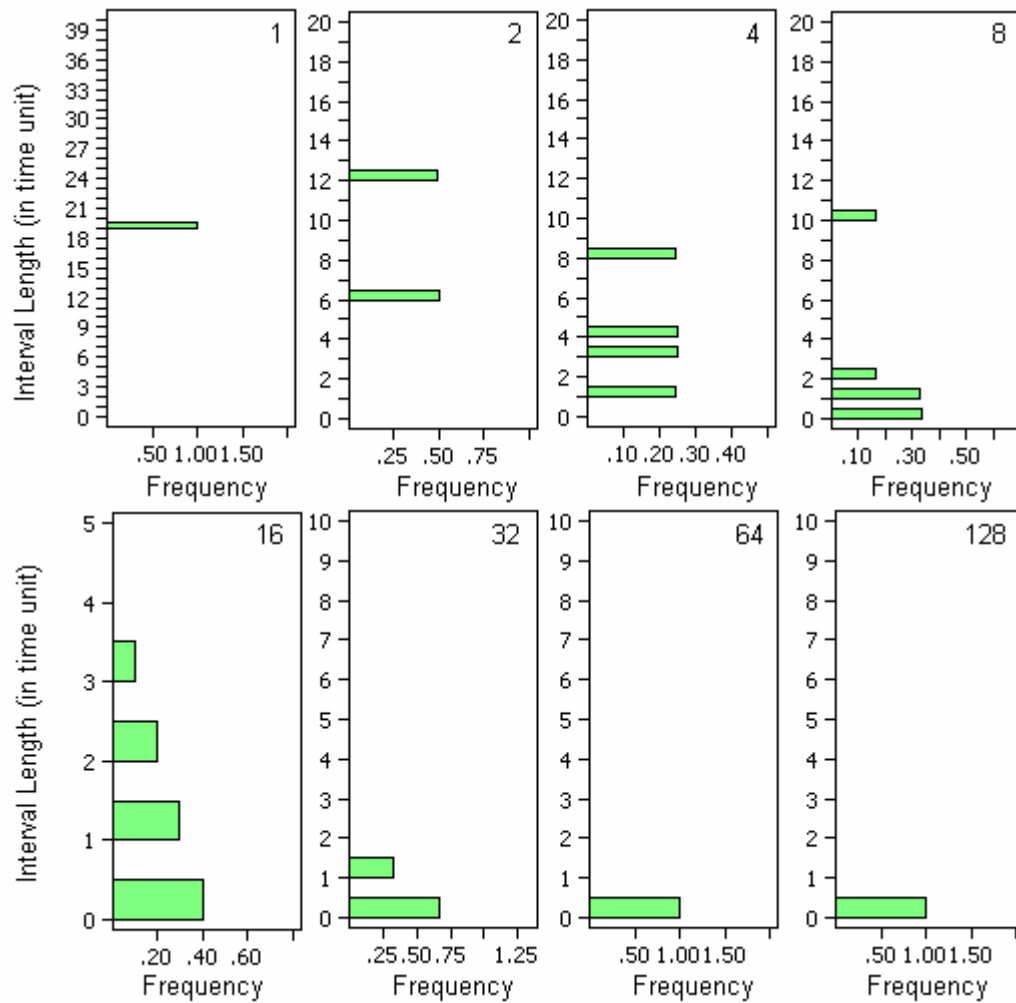


Figure 3-16. The interval distribution of the periodic model. The number at the upper right corner of every panel is the number of bats.

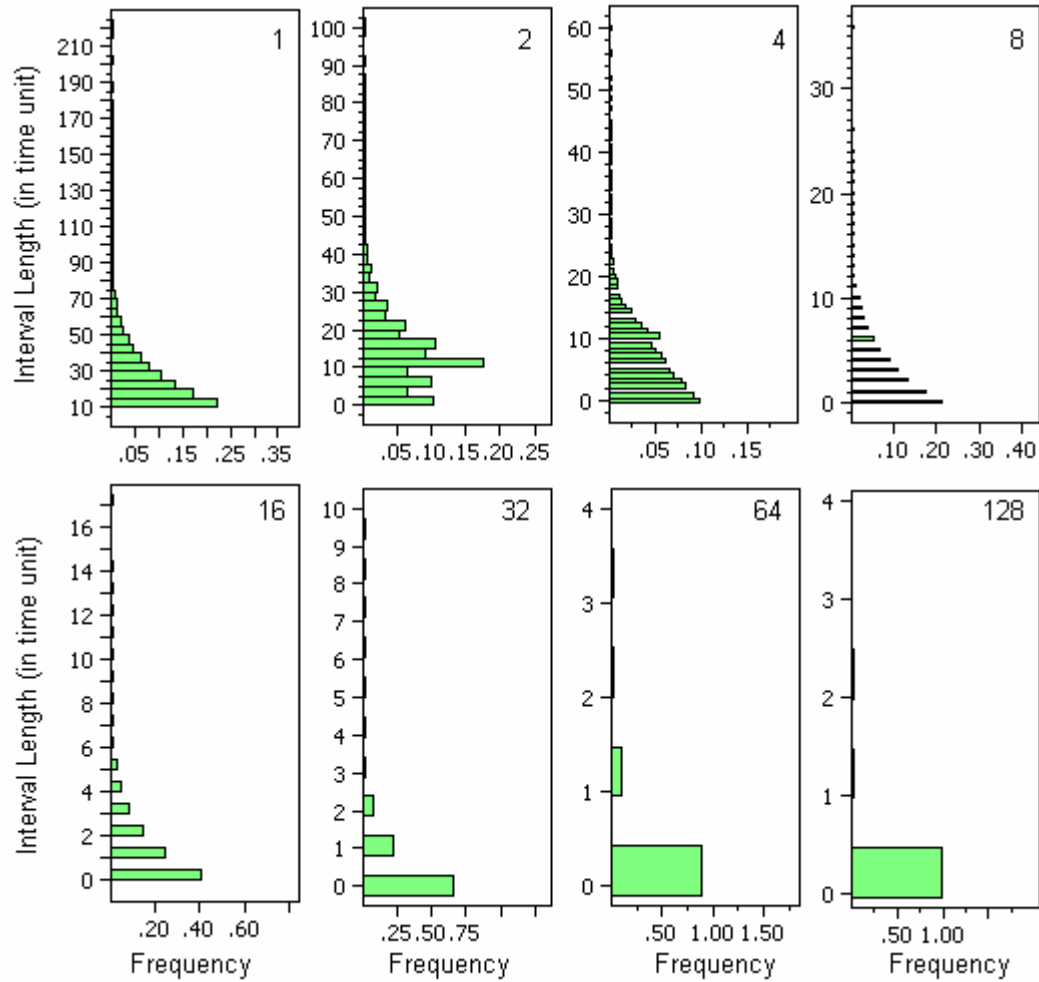


Figure 3-17. The interval distribution for the Poisson model with refractory period.
The number at the upper right corner of every panel is the number of bats.

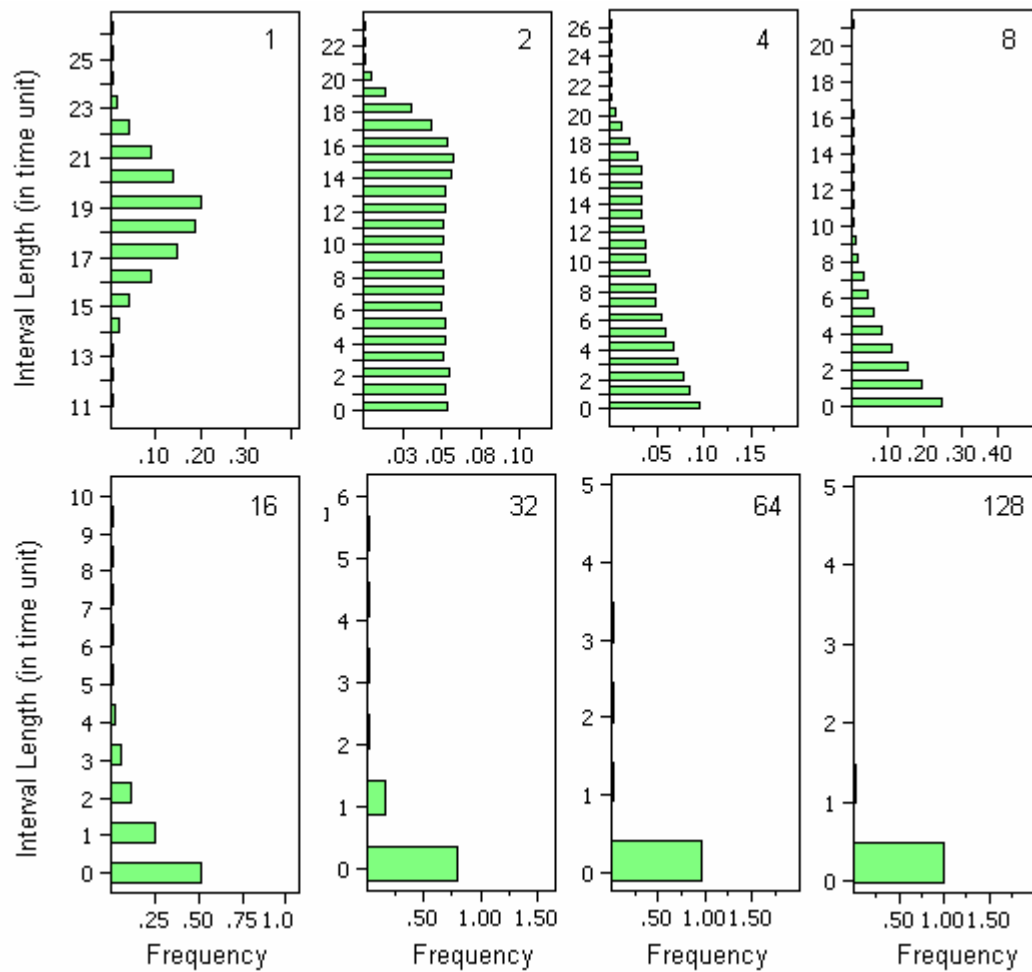


Figure 3-18. The interval distribution of the semi-periodic model.

APPENDIX III TABLES

Table 3-1 The properties of the 40 data files used in this analysis. The interval distributions of the 10 files in bold were analyzed because of their larger sample sizes.

File Name	Time Recorded	File Length (s)	Number of Calls
320	3:20	6.4	24
420	4:20	12.5	36
540	5:40	36.9	155
545	5:45	54.0	279
546	5:46	56.2	428
600	6:00	40.3	1248
605	6:05	13.2	396
606	6:06	159.4	4511
610	6:10	12.8	221
612	6:12	7.8	139
613	6:13	124.0	2397
615	6:15	57.1	920
617	6:17	128.9	2299
618a	6:18	26.2	473
618b	6:19	12.6	200
620	6:20	181.0	3527
621a	6:21	12.0	179
621b	6:21:30	8.3	144
622a	6:22	9.1	156
622b	6:22:30	11.3	185
623	6:23	44.0	745
624	6:24	48.7	830
2140	21:40	7.6	37
2150	21:50	6.9	28
2155a	21:55	4.5	22
2155b	21:55:30	16.9	69
2157	21:57	13.5	43
2210	22:10	12.0	25
2218	22:18	7.2	25
2220	22:20	20.4	51
2222	22:22	5.7	26
2230	22:30	11.3	31
2235	22:35	7.4	29
2251	22:51	4.5	21
2255a	22:55	7.1	13
2255b	22:55:30	8.7	40
2320	23:20	9.9	55
2350	23:50	2.4	10
2355	23:55	7.5	17
2455	24:55	2.6	10

Table 3-2. Moments of the interval distributions for the 10 data files indicated in Table 3-1.

Time	Scaled Call Rate	Sample Size	Mean (ms)	Standard Deviation	Maximum (ms)	Minimum (ms)	Median (ms)
6:00	0.1136	1248	28.85	24.7	150.75	1	22.55
6:05	0.1161	396	30.05	27.35	151.1	1.1	20.4
6:06	0.1219	4511	31	27.15	167.44	1	22.95
6:13	0.06925	2397	48.2	40.75	269.75	1	38.1
6:15	0.08179	920	56.85	44.5	271.45	1	48.3
6:17	0.06225	2299	52.6	45.8	330.8	1	42
6:18	0.08522	473	49.9	38.95	192	1.2	42.1
6:20	0.09101	3527	46.65	40.55	328.7	1	35.9
6:23	0.1384	745	51.1	40.25	224.6	1	41.35
6:24	0.1307	830	51.05	40.15	288.25	1.1	43.65

Table 3-3. The call rate and the information rate for the models.

	Periodic Model		Poisson Model		Poisson Model Refractory		Semi-Periodic Model	
Bat Number	Call Rate	Info Rate	Call Rate	Info Rate	Call Rate	Info Rate	Call Rate	Info Rate
1	0.0500	0.0500	0.0498	0.0470	0.0336	0.0335	0.0513	0.0511
2	0.1000	0.0999	0.0980	0.0873	0.0658	0.0632	0.0999	0.0936
4	0.2000	0.1838	0.1854	0.1483	0.1266	0.1125	0.1116	0.1011
8	0.3000	0.1905	0.3370	0.2227	0.2373	0.1830	0.2772	0.2057
16	0.5000	0.2932	0.5573	0.2746	0.4201	0.2534	0.5252	0.2750
32	0.7500	0.2987	0.8060	0.2779	0.6626	0.2839	0.7968	0.2795
64	1.0000	0.2542	0.9635	0.2599	0.8859	0.2705	0.9619	0.2601
128	1.0000	0.2542	0.9986	0.2544	0.9869	0.2563	0.9984	0.2544

Table 3-4. Comparison of the model distribution with the data distribution.

Data	600	605	606	613	615	617	618	620	623	624	Sum
Poisson	103.80	42.93	207.86	235.32	171.89	309.11	86.08	245.82	69.86	96.87	1569.54
df	100	100	100	100	100	100	100	100	100	100	
$\chi^2_{df,0.05}$	124.34	124.34	124.34	124.34	124.34	124.34	124.34	124.34	124.34	124.34	1243.4
Poisson Ref	104.69	73.60	281.76	390.09	130.47	663.43	62.82	372.07	33.90	44.23	2157.06
df	64	64	64	91	91	100	91	91	51	51	
$\chi^2_{df,0.05}$	83.68	83.68	83.68	114.27	114.27	124.34	114.27	114.27	68.67	68.67	969.8
Semi-Periodic	668.80	503.17	2525.96	13414.24	3047.69	13824.69	990.94	11777.51	101.31	170.04	47024.35
df	22	22	22	27	27	27	27	27	22	22	
$\chi^2_{df,0.05}$	33.92	33.92	33.92	40.11	40.11	40.11	40.11	40.11	33.92	33.92	370.15

Highlighted are the cases where $\chi^2 < \chi^2_{df,0.05}$, which indicates that the model interval distribution fits the data well.

APPENDIX IV USER'S GUIDE

The objective of the BatCount program is to count the number of pulses and feeding buzzes produced by bats in any given time period using recorded data files (WAVE format). The appearance of the program is shown in Figure 2-1.

The basic steps involved in running the program include:

- 1) open a data file
- 2) set parameters for counting
- 3) set saving parameters
- 4) count
- 5) read the result from the output file

For new data, it is strongly recommended that the user calibrate the counting parameters carefully. The calibration procedures involve:

- a) cut at least three segments from the data file and save them as individual files. Count the number of pulses and feeding buzzes by hand in the BatSound program for each segment
- b) run the counting using BatCount with the default parameter values for each of the data segments
- c) compare the results with that obtained by hand-counting
- d) if the difference between the two results is unacceptable, then run the counting again with different parameter values and find the appropriate values.

- 1) open a data file

Click File|Open (X|Y means click menu X and then sub-menu Y.) then the file-opening dialog will pop up (Figure 2-2). Enter the name of the data file and click OK button. An oscillogram will be displayed if “Show Oscillogram” is selected.

- 2) set parameters for counting

Click Analyze|Count|Parameters to set the parameters for the counting portions of the program (Figure 2-3). See the parameter calibration section for detailed description of the parameter setting.

- 3) set saving parameters

Click Analyze|Count|Output/SaveOptions to set how frequently you want to save the results (Figure 2-4). By default, results will be saved every 5 minutes.

- 4) count

Click Analyze|Count|Run to begin counting.

- 5) read the result from the output file

The results are saved in an Excel file. The name of the file is already set by the user in the

file-open dialog (Figure 2-2).

Parameter calibration

Calibrating parameters is strongly recommended for new data. Follow the steps below to calibrate the parameters:

a) Cut at least three segments from the data file and save them as individual files. Each of the three segments should contain at least 50 pulses and a couple of feeding buzzes. Count the number of pulses and feeding buzzes by hand in the BatSound program for each segment. The two numbers serve as the standard when calibrating the parameters.

b) Run the counting with the default parameter values for each of the data segment. Be sure to display the oscillogram.

c) Compare the results with that obtained by hand-counting.

d) If the difference between the two results is unacceptable, then run the counting again with different parameter values and find the appropriate values. To set the parameter values, click Analyze|Count|Parameters (Figure 2-3). There are seven parameters you can set:

Parameter	Meaning	Default value
Minimum Pulse Duration	If the duration of a peak is less than this parameter, then it is not considered as a bat signal pulse.	1 ms
Minimum Interval between Pulses	If two peaks are too close and the distance between them is less than this parameter, they are considered as only one pulse.	1 ms
Minimum Number of Pulses in a Feeding Buzz	A feeding buzz has to include at least a certain number of pulses.	4
Maximum Interval between Pulses in a Feeding Buzz	The interval between every two adjacent pulses in a feeding buzz should be shorter than this parameter.	15 ms
Moving Average Period	The averaging period for the method of moving average. The larger this parameter is, the smoother is the resulting time series. (See Chapter 2 for details) You don't have to change this value.	4 ms
Noise Average Period	The period for calculating the noise level. (See Chapter 2 for details) You don't have to change this value.	100 ms
Final Noise Level	Sound intensity below which is considered as noise.	0.05

The default setting works well for low-noise-level data. For data which contains so much noise that even the user can't distinguish the pulses from the background noise, it is

recommended that the user abandon the data.

When the result obtained is acceptable, run the program for the whole data file.

VITA

Xu Cui was born in Zhengzhou, China on 18 February 1978. He went to elementary school and middle school in Xinmi. He graduated in Xinmi Experimental School in 1996. From there, he went to Peking University and received a BS in biology in 2000.

Xu is currently pursuing his Master's degree in Ecology and Evolutionary Biology at the University of Tennessee at Knoxville.