Electronic Journal «Technical Acoustics»
http://webcenter.ru/~eeaa/ejta/

2004, 20

Susan M. Bertram*, Luke A. Johnson, Jerome Clark, Carmenlita Chief

School of Life Sciences, Arizona State University, Tempe, Arizona, USA 85287-4501

An electronic acoustic recorder for quantifying total signaling time, duration, rate and magnitude in acoustically signaling insects

Received 05.11.2004, published 03.12.2004

APPENDIX II

Summary software written (runs in MatLab 6.1). This software is designed to take the large data matrix produced by the EAR and summarize it into two distinct files. The SS.mat file will contain all the summary statistics. The min.mat file contains the overall summary of the number of seconds each individual signaled each minute.

```
% Cricket_Summary_Statistics.m
% To be used with files that are from 6pm to 10am (18:00 to 10:00)
% File should have no column headings or time headings.
% Before running, replace all a's (representing a 10) with 9's and save as a .cvs file

clear; % Clears out all previously used variables
load EAR_Data_Matrix.txt;
Smtrx= EAR_Data_Matrix;

% DELETE TIME COLUMN AND GET THE SIZE OF THE REST OF THE FILE
Smtrx(:,1) = []; % [] deletes the time column
Time_removed=clock
SizeSmtrx=size(Smtrx);
LgthSmtrx=SizeSmtrx(:,1); % Determines number of rows in Smtrx
WdthSmtrx=SizeSmtrx(:,2); % Determines number of columns in Smtrx
Sized=clock
i = find(Smtrx == 1); % Replaces all values of one with zero
Smtrx(i) = 0;

% Calculate Total Signaling Time (TSCmin)
for i=1:WdthSmtrx;
      calltime(:,i)=nnz(Smtrx(:,i));
end;
TSCsec=calltime';
CMPfctr=round(LgthSmtrx/(60*16)); % Avg number of samples in a minute of data
TSCmin=TSCsec/CMPfctr;

% Calculate Median Values
```

*corresponding author, e-mail: sbertram@asu.edu

```
medSmtrx=Smtrx;
WndwSize=5;
for frlp=1:WdthSmtrx
      if TSCmin(frlp,:) > 1
         for i=1:WdthSmtrx; % Sets up a loop
            medSmtrx(:,frlp) = medfilt1(Smtrx(:,frlp),WndwSize,LgthSmtrx); % Runs a
                  sliding window taking median values
         end; % Ends the loop
      end
end

% Calculate Microphone Score (MicScore) and Sound Pressure Level (SPL)
SUMSarray = sum(medSmtrx); % Sum of all microphone scores for each individual
SUMSarray = SUMSarray';
MicScore=SUMSarray./TSCsec; % Average microphone score for each individual
SPL=(MicScore+12.118)/0.232; % Equation for Gryllus texensis SPL at 4 cm distance

% Calculate Start time (TCPstart), Mean time (TCPmean), Stop time (TCPstop)
for i=1:WdthSmtrx;
      if sum(medSmtrx(:,i))==0;
         TCPstart(i)=0;
         TCPmean(i)=0;
         TCPstop(i)=0;
      else
         [R,C,VA] = find(medSmtrx(:,i) > 0);
         TCPstart(i)=min(R);
         TCPmean(i)=mean(R);
         TCPstop(i)=max(R);
      end
end
numtime=16/LgthSmtrx;

TCPstart=TCPstart'*numtime;
TCPmean=TCPmean'*numtime;
TCPstop=TCPstop'*numtime;

% Compress the second x second file to a minute x minute file
OneSmtrx=medSmtrx;
i = find(OneSmtrx > 0); OneSmtrx(i)=1; % Smtrx is a matrix with 0 for no calling and
         1 for calling

CUMSmtrx=cumsum(OneSmtrx); % Cumulative sum of medSmtrx
CMPfctr=round(LgthSmtrx/(60*16)); % Avg number of samples in a minute of data
Z=fix(LgthSmtrx/CMPfctr);
for i=1:Z;
      MinSmtrx(i,:)=CUMSmtrx(CMPfctr*i,:);
end
MinSmtrx=diff(MinSmtrx); % Compresses the data from second data to minute data
```

```
% Calculate the number of bouts, breaks, avg break length, avg bout length
SizeMinSmtrx=size(MinSmtrx); % Calculates the size of the Smtrx
LgthMinSmtrx=SizeMinSmtrx(:,1); % Determines length of Smtrx
WdthMinSmtrx=SizeMinSmtrx(:,2); % Determines width of Smtrx
NumBouts=zeros(1,WdthMinSmtrx);

for i=1:WdthMinSmtrx;
      for j=2:LgthMinSmtrx;
         if MinSmtrx(j-1,i)==0 && MinSmtrx(j,i)>0
            NumBouts(1,i)=NumBouts(1,i)+1; % Calculating average number of bouts
         end
      end
end
NumBreaks=NumBouts;
for i=1:WdthMinSmtrx;
      if NumBreaks(1,i)>0
         NumBreaks(1,i)=NumBreaks(1,i)-1; % Calculating average number of breaks
      end
end
NumBouts=NumBouts';
NumBreaks=NumBreaks';
AvgBoutLength=TSCmin./NumBouts; % Calculating average bout length
AvgBreakLength=((TCPstop-TCPstart)*60-TSCmin)./NumBreaks; % Calculating
      average break length

% SAVE THE RESULTS (TSC, MicScore, START, MEAN, STOP)
RSLTS=[TSCmin MicScore SPL NumBouts AvgBoutLength NumBreaks
      AvgBreakLength TCPstart TCPmean TCPstop];
savefile = ‘SS.mat';
save(savefile,'RSLTS','-ASCII')
savefile = 'min.mat';
save(savefile,'MinSmtrx','-ASCII')
```