Data Analysis Using the Ebert 1.0 Suite of Programs (15 October 2014)

Timothy Ebert

Introduction

This is a short tutorial on how to use these programs. This tutorial is much like a baking recipe. It is possible to replace some steps using other methods, and sometimes these other methods can be easier depending on the experience of the user. If you can get the same result using a different method, then please use the better approach. This is only a basic introduction. This document is only to help people take their aphid data and produce a statistical analysis suitable for publication in the shortest path. The worked examples presented in other documents adds a little complexity. The psyllid data are provided primarily to help people who use Windaq for data acquisition, while the aphid data is primarily for those using Probe. Additional documents will enable the full potential of this suite of programs.

I assume that you have recorded your insect's waveforms, and have coded the data using one of two programs: Windaq, or Probe. The programs should be a recent version. Older versions might work, or might not. This depends on whether or not there was any change in the format of the output files for Windaq or Probe.

Please pay close attention to highlighted text.

It is recommended that file names be short and repetitive. A name like "Admire 210814 trt2 Machine3" may be very useful for keeping track of where and when each recording was made. When you convert this into a data file for each insect try names like "Control 1" or "Trt 2". Such simple names are easier to type into the file reading program, and are easier to check for typing mistakes.

Finally, sample raw data files along with the expected output are provided, and it is suggested that users load all the files onto their computer and run the analysis with the sample data. One should check a few of the numbers in your output against the output that we provide to make sure everything works properly[1].

PHASE 1: Reading the data

In this part we assume that your data is a collection of files, one file for each insect, with waveform codes and times. The file may contain additional numbers put there by the software. The SAS program for Phase 1 will remove extraneous data and concatenate all the by insect data files into a single data file.

We assume that the data come from either Windaq, or Probe. Windaq files will have no extension, Probe files use the ".ana" extension.

> File extensions are visible in Window's Explorer (Windows 8). Click on "view" tab at the top of the screen. The show/hide pane is towards the right side and it has a box to check labeled "File Name Extensions." With this box checked you should now see file names that look like

[1] This software was made with great care, but it comes without warrantee as to its usefulness or accuracy. The user assumes all responsibility for outcomes and consequences.

name.csv, or name.txt. The “.txt” is the file name extension. It helps the computer figure out what to do with the file.

To make the following discussions more readable, this tutorial will follow a few conventions.

1) File name: We will use a generic name “Filename” to refer to any file. The file name may have an extension which consists of a period and up to four characters at the end of the Filename. The extension identifies the type of file and the program the computer will use to open the file. Unless stated otherwise, the file name in this tutorial will include the path, the name, and the extension. So entering the file name into any of the programs will look something like this: C:\User\Project 3\Filename.csv.
    a. To find the path in Windows Explorer (In Windows 8) go to a file from one of your insect. Single click on the file name to highlight it. There should be a menu bar with “File Home Share View”. Look below this bar and you should see some arrows. To the right of the arrows is a folder icon and some names that end in the name of the folder where your file is located. Single mouse click the folder icon or anywhere in the box around the icon and folder names. The display will change to something like C:\name\name\name and it will be highlighted. Press the control key and while holding the key down press the c key (Ctrl-c). Go to your SAS program, highlight the area where the file name goes and press ctrl-v to paste the text into your program. Type in another \ and then your file name.
2) File Format: This includes the order in columns of numbers, the number of columns, and any codes a program may add to a file. Type in a few numbers into Excel, save the file in the standard Excel format, and then open it using a text program like Wordpad. You will see something like this
    “PK
    !
    ¤SÅÏN
    [Content_Types].xml
    ¢(
    .“ The exact characters depend on the font, but the point is that there are a number of characters present in the file that help the computer but are meaningless to most people. These characters are also part of the file format.

A windaq file will look something like this:

```
"NP                                                              "

 84127.700,  84015.220, 1.2561E+00, 0.0000E+00, 3.0884E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00, 0.0000E+00
```

In this case the insect had only one behavior (np). More typically there would be many pairs of lines like this one in the file. Decimals are marked with a period (US format) rather than a comma (European format), and values are separated by a comma and a space. The first two numbers are written out to three decimal places and the remainder are written in scientific notation.

A Probe file will look something like this:

```
1           0           223,7412089

2           3560,66     86,61461999

8           3600,6163 55,47075516
```

Commas are used rather than periods to mark decimal points. All waveforms have a numerical code. Values are separated by a tab.

These features are important because any change in these features could require a rewrite of this software. Such changes typically occur when software developers write a new version of their software. If this happens the program will fail to work properly, even though you may still get output in SAS.

The files to use will be either FileManipW (Windaq) or FileManipP (Probe). Each file name will include a number that is a date in Month/Day/Year format. The date is the last time I changed the file.

The Windaq version of the file reading program has three modules that need to be modified to get the program to work. The first module reads the first data file and then sets up the SAS dataset where all the data will be accumulated. The second module reads other files, and there should be one copy of this module for each additional insect that you recorded. The third module saves the data to a file. The first and second module have two places that need to be changed: the file name, and the insect number. You will find an infile statement (similar to the one shown) that needs a file name.

```
infile 'C:\Users\Project 3\Water 1' dsd missover;
```

Several lines later you will see the following line:

```
data two; set two; insectno="a01";
```

This line must have a three character insect number. Please use a letter to designate a treatment, and a number to designate the insect within that treatment. The other programs are written to assume this format. If you don't follow this format then you will need to make a few changes as necessary to the other programs.

Next there will be a set of commands that look something like this (Microsoft Word has formatted this to fill three lines, but it appears as only one line in SAS):

```
Data one; infile 'C:\Users\Project 3\Water 2' dsd missover; input waveform$
Var1 Var2 Var3 Var4 Var5 Var6 Var7 Var8 Var9; Data one; Set one; %Manip; data
two; set two; insectno="a02"; proc append base=four data=two; Run;
```

Copy and paste this section of code, once for each insect in your study. Change the file name and insect identifier (insectno="a02";) as appriorate.

Finally, you need a file name to save the data. Near the end of the program is a line similar to this:

```
proc export data=five outfile='C:\Users\Project 3\DataFile.csv' dbms=csv
replace;
```

Change the file name in this statement as appropriate, then run the program.

The Probe version of the software looks slightly different because more steps are involved in reading the file. However, the same changes need to be made in the file: change the infile statements, change the insect numbers, and change the output file name in the “proc export” statement.

The first infile statement looks like this:

```
infile 'C:\Users\Project 3\Aphid1.ana' dsd dlm='09'x truncover;
```

It has a corresponding insect number identifying statement that may need to be changed:

```
insectno="a1";
```

This statement is on a line by itself. The next task is to change the file names in the other read modules, and if necessary copy or delete to make the number of modules match the number of insects in the file. Each module takes several lines:

```
data one;
infile 'C:\Users\Project 3\Aphid3.ana' dsd dlm='09'x truncover;
input @; _infile_=compress(translate(_infile_,'.',','),'"'); input a b c ;

data one; set one; drop c; dur=0;                             insect1="a2";
data one; set one; retain holder1 in0;
if in0 ne insect1 then do; holder1=0; in0=insect1; dur=b; end; else dur=b-
holder1; holder1=b;
data two; set one; insectno=insect1; waveform=a; duration=dur;
data two; set two; drop a b holder1 in0 dur insect1;
data two; set two; retain holder1 in0;
if in0 ne insectno then do; in0=insectno; holder1=0; end; else wave1=holder1;
holder1=waveform;
data two; set two; if wave1 ne "." then output; data two; set two;
waveform=wave1;
data two; set two; drop in0 wave1 holder1; data two; set two; proc append
base=allsets data=two;
proc datasets nolist nodetails; delete one two;
```

The file name that needs to be changed is the second line of this module. It is followed by an input statement, then a blank line. The insect number is towards the right side on the first line after the blank.

If you are not using aphids, or if you have a waveform that is not a standard behavior then you might need to change the following code that converts the numeric codes into letter codes for each behavior.

```
if waveform=1 then SW="NP";
if waveform=2 then SW="C";
if waveform=3 then SW="E1e";
if waveform=4 then SW="E1";
if waveform=5 then SW="E2";
if waveform=6 then SW="F";
if waveform=7 then SW="G";
if waveform=8 then SW="PD";
if waveform=9 then SW="II2";
if waveform=10 then SW="II3";
if waveform=11 then SW="PDL";
```

Finally, change the output file name to whatever is appropriate.

```
proc export data=allsets outfile='C:\Users\Project 3\AhidRaw.csv' dbms=csv
replace;
```

Run the program.

PHASE 2: Checking for Errors

The file for this part is called “Error Checker” followed by a date code. In this part we assume that your data is a single file with the insect number in column 1, the waveform in column 2, and the duration of that waveform in column 3. Columns are separated by commas, hence the “.CSV” extension. Any treatment information should be part of the insect number.

In this part we look for common errors. These can be any of the following:

1) Novel waveforms caused by typographical error. It is suggested that behavioral codes are all upper case. This section will treat np as a separate behavior from nP. A full list of the codes that were used in the file is provided to assist with finding errors.
2) Improper behavioral transitions: These are specific to your insect. For aphids examples would include no E1 before E2, Np-G, and E2-G. There are other impossible transitions depending on how pd is recorded. For non-aphids, there will be other impossible behavioral transitions. This program prints a table of all transitions, making it easier to catch this type of mistake.
3) No repeated behaviors: Np-C-C-E1 is not allowed.
4) All recordings start with the non-probing waveform. This error is not trapped by the program. If this error is present then one doesn’t know when the first probe started, or even if one is looking at the first probe. However, this only affects a few variables. If there are a few insects that do not start with the non-probing behavior, then run the analysis twice. Once with these insects present in the data, and once after deleting them. From the latter analysis get the results for variables that include information about first probe, second probe, or time from start of the first probe. It is easier to always start recording before putting the insect on the plant.
5) There is only one non-probing waveform.
6) All durations are greater than zero.
7) Waveforms are all of the same case. By default SAS will treat g and G as different. So np, NP, Np, and nP are four different behaviors. The SAS programs use a command Compress(upcase(waveform)) to remove any spaces that may have been added “NP” is different from “NP “ and to change all characters to upper case. If you have a set of waveforms where the only difference is capitalization, then you will have to disable this part of the program. It is suggested that you avoid this problem if at all possible.

The process of finding and correcting errors is mostly run the program, find an error, fix it, rerun the program, find the next error, fix it, and so on. If there are only a few errors and a large data set this process is easy. If there are a large number of errors it might be better to find most of them using Excel, and then making a final check using this program. You can also use this program and then use the global replace feature in Excel. If you find nP once and suspect that it might have happened several times, then do a global replace nP for NP and the task is done.

The program consists of a number of macro statements in the beginning. Skip over these until you find the Infile statement. Enter the file name you used in the Proc Export line in the previous program. Run the program.

What you should see in the output screen are two tables.

Frequency Table of Waveform Event Transitions

The FREQ Procedure

waveform	Frequency	Percent	Cumulative Frequency	Cumulative Percent
C	1163	43.77	1163	43.77
D	120	4.52	1283	48.29
E1	201	7.56	1484	55.85
E2	107	4.03	1591	59.88
G	137	5.16	1728	65.04
NP	929	34.96	2657	100.00

Frequency Table of Waveform Event Transitions

The FREQ Procedure

trans1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
C to D	120	4.59	120	4.59
C to G	137	5.24	257	9.84
C to NP	885	33.87	1142	43.70
D to C	11	0.42	1153	44.13
D to E1	109	4.17	1262	48.30
E1 to C	94	3.60	1356	51.89
E1 to E2	107	4.09	1463	55.99
E2 to C	8	0.31	1471	56.30
E2 to E1	92	3.52	1563	59.82
G to C	136	5.20	1699	65.02
NP to C	914	34.98	2613	100.00

If these tables are not at the top of the output log, then there are errors in the data. The most common output with errors will look like this:

Duration by waveform Output

Obs	insectno	waveform	dur	w0	w1	in0	marker1
3	a01	C	742.4	C	C	a01	1

This table may be very long, but go to the end of this table and note the last entry in the table. Find this entry in your data and fix the problem. The error in this data file was at the beginning of the file. Observation 3 for insect a01, and was caused by two consecutive C behaviors (error #3 in the list). So one opens the file using a program like Excel and goes to the third observation. It is suggested that most of the errors be corrected in the “by insect” data files and one reruns the file manipulation program. That way the error is permanently removed from the data. This approach takes a bit more time, but it can save time if there is an error that requires you to reanalyze an insect waveform.

> Consider a hypothetical situation where you correct all the errors in insect 1 to 80, and find that you really need to reexamine and reanalyze the waveform for insect 81. Upon reflection you decide that there was an error and you take the new data. You rerun the file manipulation program to integrate the new results into your data set. If you have changed the by insect files then you are good to go. If you only changed the output file from the file manipulation program then you will have to redo all the corrections for insects 1 through 80.

The following was the result of a negative value for duration. Looking at the Duration column, one can see the negative value (error #6 in the list).

Duration by waveform Output

Obs	insectno	waveform	dur	w0	w1	in0	marker1
4	a01	C	-104.00	G	C	a01	1
5	a01	G	511.04	G	G	a01	1

If there are many errors it will be faster to find most of them using a spreadsheet program like Excel or OpenOffice.

Assuming that there were no repeat behaviors and no negative durations, you should see two tables like these:

Frequency Table of Waveform Event Transitions

The FREQ Procedure

waveform	Frequency	Percent	Cumulative Frequency	Cumulative Percent
C	1163	43.77	1163	43.77
D	120	4.52	1283	48.29
E1	201	7.56	1484	55.85

E2	107	4.03	1591	59.88
G	137	5.16	1728	65.04
NP	929	34.96	2657	100.00

Frequency Table of Waveform Event Transitions

The FREQ Procedure

trans1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
C to D	120	4.59	120	4.59
C to G	137	5.24	257	9.84
C to NP	885	33.87	1142	43.70
D to C	11	0.42	1153	44.13
D to E1	109	4.17	1262	48.30
E1 to C	94	3.60	1356	51.89
E1 to E2	107	4.09	1463	55.99
E2 to C	8	0.31	1471	56.30
E2 to E1	92	3.52	1563	59.82
G to C	136	5.20	1699	65.02
NP to C	914	34.98	2613	100.00

The first table tells you the number of times each behavior occurred in your data. Look carefully at this table and make sure that each behavior should be there. Typographical errors might let an E2 become an E3, or G might become an F or GH depending on how the fingers typed. You may also see some G and some g. It is better if these conditions are all fixed.

The second table is also important because it will show you all the transitions from one behavior to another. So in this case I find that there were 120 times when behavior D came after behavior C. By looking at all the entries I also note that C was the only behavior before D. From my knowledge of this insect I know this to be correct. If I had seen an entry “NP to D” in the above table I would know that I had more errors to correct.

Congratulations, your data set is now as free from errors as we can make it at this stage of the analysis.

PHASE 2a: Data Analysis using the Error Checking Program (This part is optional)

In many cases there are either applied treatments (e.g. two host plants, or three insecticides), or observational treatments (e.g. gender or age). It may be useful to have the output from the two tables for each of the treatments. To do this look for the following code in the program:

```
Data one; set one; *if substr(insectno,1,1)="e" then output;
```

The * character in SAS indicates a comment. The code is not executed when the program is run. If each insect was identified as we suggested then delete the * from this line and place the letter for the treatment inside the quotation marks (currently the treatment is e). Run the program, save the results, change the letter, and repeat for each treatment.

PHASE 3: Data Analysis

The program we will use in this part is called "Ebert 1 0". The only required change is the infile statement at the beginning of the program. The output is usually put into a file. The statement is about 28 lines below the infile statement. It looks like this:

```
ODS HTML file='C:\Users\Project 3\AnalysisP3';
```

If your data conforms to the default, and if there are no other issues, then run the program and you are done. Please look at some of the other options that are described below, just to make sure that the default is the correct strategy.

PHASE 4: Optional Features

The analysis of EPG data can be complex. So there are a number of options for dealing with issues. We will deal with these in the order in which they appear in the program. Some of these features are present in case you have used other programs to get to this point. Topic headings are underlined.

Numeric waveform Codes: at the top of the program. The Probe software uses numeric codes for waveforms. These are changed to letter codes that are needed for this program to work. If your data has numeric codes for the different waveforms, please check this table to insure that your data are analyzed properly.

```
data one; set one;
if waveform='1' then waveform='NP';
if waveform='2' then waveform='C';
if waveform='3' then waveform='E1E';
if waveform='4' then waveform='E1';
if waveform='5' then waveform='E2';
if waveform='6' then waveform='F';
if waveform='7' then waveform='G';
if waveform='8' then waveform='PD';
if waveform='9' then waveform='II2';
if waveform='10' then waveform='II3';
if waveform='11' then waveform="PDL";
```

A separate column for treatment codes: at the top of the program. If your data has a separate column for treatment codes please make sure that the treatment codes are (preferably) one letter. You will have to modify the infile statement to read the data, and you will need to delete the asterisk from the following line:

```
*insectno=compress(trt||insectno);
```

If you use treatment codes that are not a single letter, then you should run the program and check the SAS dataset (Ebert), and the Log file. Make sure that there is only a single line in Ebert for each insect, and make sure that there are no unexpected errors in the SAS Log.

Typical runs will generate a few errors in the SAS Log because the data are not suitable for all analyses. The usual cause is missing data. For example, if you don't have an F behavior, SAS will still try to calculate mean duration of F, and then give you an error when it can't find data.

Transformations: towards the bottom of the program. Scroll down through the program and towards the bottom of the program you will see large banners. They are:

"Activate this section if treatments are fatal"

"Transformations"

"Data Analyses"

Consider your choices carefully. I typically run the program twice. The first time I turn off everything in the "Transformations" section. I change the output file name, activate transformations, and rerun the program. I then bring up the output in Excel and take means and standard deviations of the untransformed values and combine that with the statistical analysis for the transformed data.

Data Analyses: towards the bottom of the program. This part is only a default. A place holder to indicate where data analysis should take place in this program. The type of analysis is really your choice. The restricted maximum likelihood method used here with an LSD mean comparison procedure is good. It might be better to use a different mean comparison procedure. More importantly, one can include linear regression and multivariate techniques to better understand the data. However, there are too many choices and this tutorial is not a broad overview of statistical methods.

Sarria: towards the bottom of the program. It may be useful to print out our results and compare them to the Sarira output. To get the "by insect" results, activate this code by removing the asterisk.

```
*proc export data=Ebert outfile='C:\Users\Ebert1.txt' dbms=csv
replace;
```

Note the replace command will delete the old file. If you want to save the output from previous runs you will need to do one of the following: rename the file, copy and paste the contents to a new file, or change the name of the output file. There is no warning that the old file is being deleted.

Run the program once you have made appropriate changes to file names, decided on handling missing values, and applied any transformations.

It is suggested to work through the worked example using the data provided. This way you can work through a data set where the answer is known before working on a data set where you have to rely on the program to give you the correct answer.