

Introduction.....	1
Installation.....	1
Java.....	2
Cygwin.....	2
WFDB Software Package.....	2
MATLAB.....	2
JMatLink.....	2
MySQL.....	2
Puka.....	3
Configuration.....	3
Raw Data File Format.....	4
Reading Data into Puka.....	5
Loading Data Without a Database.....	5
Loading Data Using a Database.....	6
Indicating the Clip and Onset Point.....	7
Respiration Analysis.....	8
Step 1: Checking the Signal.....	8
Step 2: Identifying the Breaths.....	9
Step 3: Checking the Breaths for Validity.....	10
Step 4: Locating Pauses.....	12
Step 5: Statistical Calculations.....	13
EKG Analysis.....	14
Step 1: Checking the Signal.....	14
Step 2: Using the WFDB Software Package to Identify the EKG R Waves.....	14
Step 3: Calculation of HRV Statistics.....	15
Step 4: Peak-Valley RSA Calculation.....	16
Saving Analysis Results.....	16
Appendix A.....	17
Appendix B.....	19
References.....	23

Introduction

Puka calculates descriptive statistics such as heart rate variability (HRV), peak-valley respiratory sinus arrhythmia (RSA), and respiratory variables from EKG and strain gauge respiration data. *Puka* is a Hawaiian word meaning *appear*, *emerge*, or *an opening*, and is also the name of a small round shell with a center hole found on some Hawaiian beaches. It was selected to reflect the exploratory nature of the experiments which generate the data to be analyzed with this program as well as the new knowledge that may be gained. Its icon is of the ocean and sky with red hibiscus flowers in the foreground, ©2000 Saho (Saho's IconRoom, <http://saho.vis.ne.jp/>).

Installation

Puka, written in Java, uses MATLAB for signal processing and statistical calculations and the WFDB Software Package for EKG peak detection. Puka can also read and write data from a MySQL database. The use of these separate elements, while allowing the use of multiple analysis programs from one package, necessitates a complex installation process. Five programs (Java, Cygwin, the WFDB Software Package, MATLAB, and JMatLink) must be installed before puka. If the database will be used MySQL

must also be installed. Puka has been installed on Windows 2000 and Windows XP; its compatibility with other operating systems is unknown.

Java

Puka requires the Java version 1.4 or higher runtime environment (J2SE JRE), which can be downloaded as a self-installing executable from <http://java.sun.com/> by following the download links on the main page. Java is available at no cost under Sun's Community Source License Program (SCSL, <http://www.sun.com/software/java2/license.html>). The directory where `java.exe` is located (usually `\j2sdk1.4.2_04\bin\`) must be added to the computer's system path.

Cygwin

Cygwin, a Linux emulator for Windows, is available free of charge from <http://www.cygwin.com> under the GNU General Public License (GPL). Cygwin simplifies the configuration and use of the WFDB Software Package. During the Cygwin setup process ensure that the C++ compilers are installed (this option is not selected by default). The directory where `cygwin1.dll` is located (usually `\cygwin\bin\`) must be added to the computer's system path.

WFDB Software Package

Puka uses programs from the WFDB Software Package for EKG peak detection. The `ecgpuwave` application (documentation at <http://www.physionet.org/physiotools/wag/ecgpuw-1.htm>) is used to identify the EKG waveforms, from which other programs extract the list of RR interval times and location of each R peak. The WFDB Software Package (Goldberger, Amaral et al. 2000), available under the GNU General Public License (GPL), is a sophisticated set of programs to analyze and process EKG waveforms maintained as part of PhysioNet by the Research Resource for Complex Physiologic Signals (sponsored by NIH/NCRR). Documentation, installation directions, and source code for the WFDB Software Library can be downloaded from <http://www.physionet.org/physiotools/wfdb.shtml>.

It is recommended to download the WFDB Software Package as source code and install it using Cygwin following the directions provided in the package, although it can also be downloaded in binary form. It is helpful to install the WFDB Software Package as a subdirectory of cygwin (for example, if Cygwin was installed at `c:\cygwin`, install the WFDB Software Package at `c:\cygwin\wfd\wfd-10.3.2`).

MATLAB

MATLAB, which must be purchased from The MathWorks, Inc. (<http://www.mathworks.com/>) is used by puka for graphing and computations. The path to `matlab.bat` (such as `\matlabR13p5p1\bin\`) and `matlab.exe` (such as `\matlabR13p5p1\bin\win32\`) should also be added to the windows system path.

JMatLink

JMatLink, which enables Java programs to communicate with MATLAB, is available at no charge from <http://www.held-mueller.de/JMatLink/>. Follow the installation and testing directions included with the program.

MySQL

The installation of MySQL is not required by puka, but is helpful. MySQL is an open-source database program, licensed under the GNU General Public License (GPL), and available free of charge from <http://www.mysql.com/downloads/index.html>. Puka is designed to interact with a specific database; the SQL commands to create the database are included in Appendix A.

Puka

After Java, Cygwin, the WFDB Software Package, MATLAB, and JMatLink have been installed puka can be installed. It is strongly recommended to test each program separately before attempting to use puka. If MySQL was installed, create a database and start the MySQL server before running puka.

To install puka:

1. Copy the puka directory and all of its contents to a convenient location, such as `c:\programs\puka`. Do not uncompress or otherwise alter any of the files in either the `puka` or `puka\MATLABScripts` directories.
2. Copy `convertECG.exe` (provided with the puka installation files) to the WFDB Software Package top directory. For example, if the WFDB Software Package was installed to `c:\cygwin\wfdb\wfdb-10.3.2`, install `convertECG.exe` in `c:\cygwin\wfdb`.
3. Move the shortcut to puka from the main puka directory (such as `c:\programs\puka`) to the desktop or other convenient location.
4. If Windows has been previously configured to recognize self-executing jars the program will start when `puka.jar` or its shortcut is double-clicked. If not, the "Executable Jar File" file type needs to be added to Windows by following Microsoft's instructions or the ones at http://vietunicode.sourceforge.net/howto/file_association.html. Alternatively, puka can be run from a command prompt by typing `java -jar puka.jar` at a command window in its directory.

Configuration

After puka has been installed it must be configured so that it can find all of the necessary programs. Information about the EKG equipment needs to be specified, as do details about the data that will be analyzed. Puka assumes that the data was collected in discrete segments called *clips* and that the EKG and respiration data were collected at the same sampling rate. The length (in number of sampling units) and name of each clip needs to be entered during the configuration process but can be altered later if desired. All of the configuration information is stored in a text file called `preferences.txt` which must be in the same directory as `puka.jar`. This file can be altered by a text editor or using puka's preferences tool.

1. Start puka by double-clicking on its shortcut or at the command prompt.
2. Puka will search for a file named `preferences.txt` and attempt to read the information from that file. If it can't find `preferences.txt`, or the file is invalid, a message box appears and the program exits. If this happens check that a text file named `preferences.txt` is in the same directory as `puka.jar`. If necessary, create a blank text file and name it `preferences.txt`.
3. When puka starts a MATLAB environment should be loaded and be visible as a running program in the Windows Taskbar and Windows Task Manager Applications tab. If clicked in the taskbar the MATLAB window should appear, as shown in Figure 1. This window indicates that puka, JMatLink, and MATLAB are communicating properly. Puka does not find MATLAB by reading `preferences.txt` but rather through JMatlink and the system path, so if MATLAB is not started or if an error message is displayed check the installation of JMatlink.
4. After MATLAB has been loaded the main puka window should appear, as shown in Figure 2. Click the OK button on this dialog. Before any data can be analyzed puka must be configured.
5. From the top menu select **Tools** then **Preferences**, which will cause the **Program Preferences** window to appear. Fill in the boxes on the **paths** tabs with the paths to the directories in which the indicated programs were installed. The **EKG** tab contains settings related to the raw EKG signal, and the **data columns** tab indicates which column in the raw text file holds which data. Finally, the **clips** tab contains settings to use to indicate the length and name of each clip for which data was collected.
6. When all settings are filled in, click **OK**. An updated preference file (`preferences.txt`) will be written to the same directory as `puka.jar` so that the preferences will not need to be reset each time.

7. Close puka by clicking `File` then `Exit` from its top menu. Puka and the MATLAB window will close. Now, puka can be restarted, properly configured and ready to analyze data.

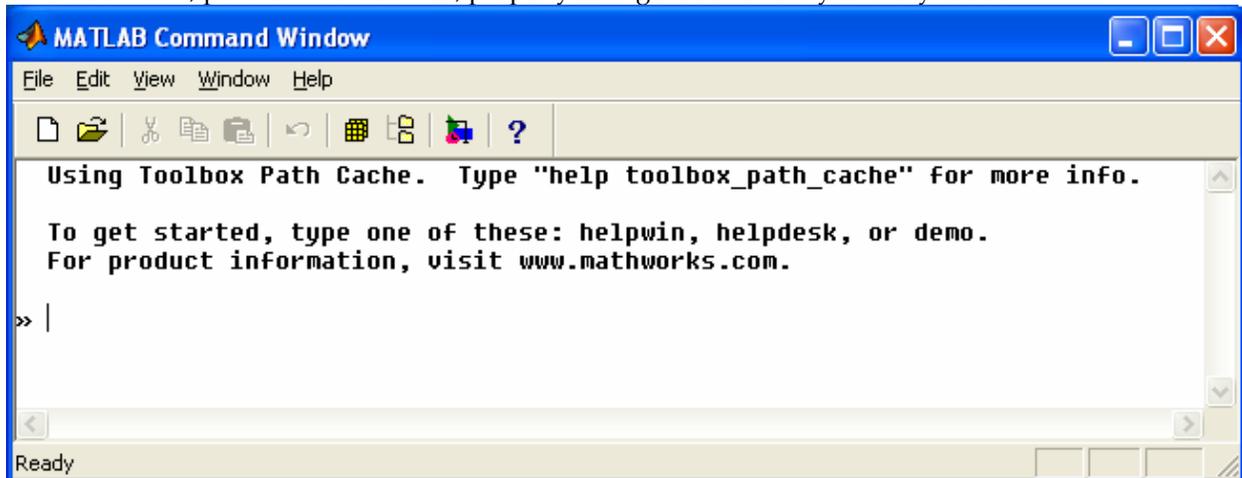


Figure 1. MATLAB command window, as it appears when puka is started.

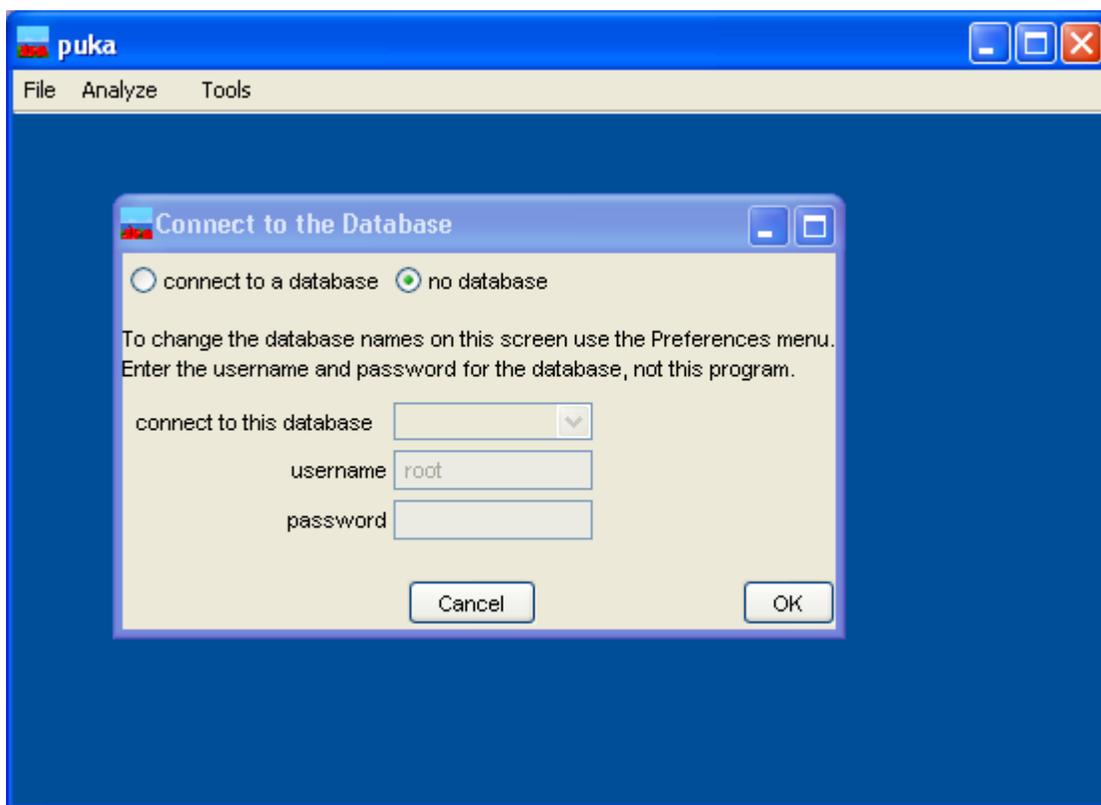


Figure 2. Appearance of the puka application at startup.

Raw Data File Format

Puka was designed to analyze data collected using equipment and software from BIOPAC Systems, Inc. but should work with data collected with other equipment. The respiration signal was collected with a TSD201 Respiratory Effort Transducer, a single strain gauge recorder, wrapped around the chest. EKGs

were recorded using a standard Lead II configuration, creating a single channel of data. BIOPAC's *Acknowledge* software was used for data collection during each experiment, converted to plain (ascii) text file for analysis by puka.

The raw data should be saved as a plain text file with tab-delimited columns, as shown in Figure 3. No column header or other text information can be included, but columns of numbers that will not be analyzed by puka can be present. Currently, puka will read data from onset trigger, EKG, and respiration channels. The column in which each data type appears is indicated in the preferences. Each row of the file is the value of each channel at a time step.

Each data file is analyzed in its entirety, so each clip must be in a separate data file. For example, consider an experiment in which five subjects are each asked to complete three paced breathing trials. This experiment will result in 15 data files, one for each subject at each paced respiration rate. The length (in sampling units) and name of three clips, one for each paced rate, should be added to puka using the Program Preferences dialog.

-0.100098	-8.99323	5.81818	-0.00152588	4.98962	0.0140381	0	-0.758812
-0.0854492	-8.98865	5.8136	-0.00213623	4.99023	0.109253	0	-0.857601
-0.0756836	-8.98499	5.81512	-0.00183105	4.99084	-0.0402832	0	0.847163
-0.065918	-8.98041	5.81207	-0.00244141	4.99023	0.0158691	0	0.203558
-0.0558472	-8.97583	5.81512	-0.00152588	4.99023	-0.126953	0	-0.506027
-0.050354	-8.97156	5.8197	-0.00183105	4.99054	0.0146484	0	-0.368633
-0.0436401	-8.96729	5.81512	-0.00213623	4.99023	0.139771	0	0.959319
-0.0415039	-8.96301	5.81818	-0.00152588	4.99084	-0.057373	0	-0.951903
-0.0396729	-8.95813	5.8136	-0.00213623	4.99054	0.0415039	0	0.275399
-0.0390625	-8.95294	5.81665	-0.00183105	4.99023	-0.0341797	0	0.747673

Figure 3. Sample excerpt from a raw data file for input into puka. Each column is a different channel of data (EKG, respiration, onset trigger, skin conductance, etc.) and each row is the value of the channel at a different time step (i.e. if data is collected at 1000 Hz there will be 1000 rows in the data file for each second of the recording).

Reading Data into Puka

Puka can read data either from a raw text file or from a zipped archive of raw text files. Additionally, puka can use the subject database to retrieve subject information, data storage locations, and to save analysis results.

Loading Data Without a Database

Each time puka is started the Connect to the Database dialog is displayed, as shown in Figure 2. If no database will be used select the no database option and click OK. This dialog can be displayed at any time by selecting Tools then Connect to the Database from the top menu.

To load the data file into puka select File then Load Data from the top menu. The Load File dialog will be displayed, as shown in Figure 4. Click the Select File button, which opens a standard file open dialog. Select the text data or zipped archive containing the data you wish to analyze. If the file is a text file select the single raw text file button. If the file is a zipped archive click the file within zipped archive radio button then the Show files within zip button which will display the names of the files contained within the zipped archive in the data file name list box. Select the data file to be opened from the zipped archive in this box.

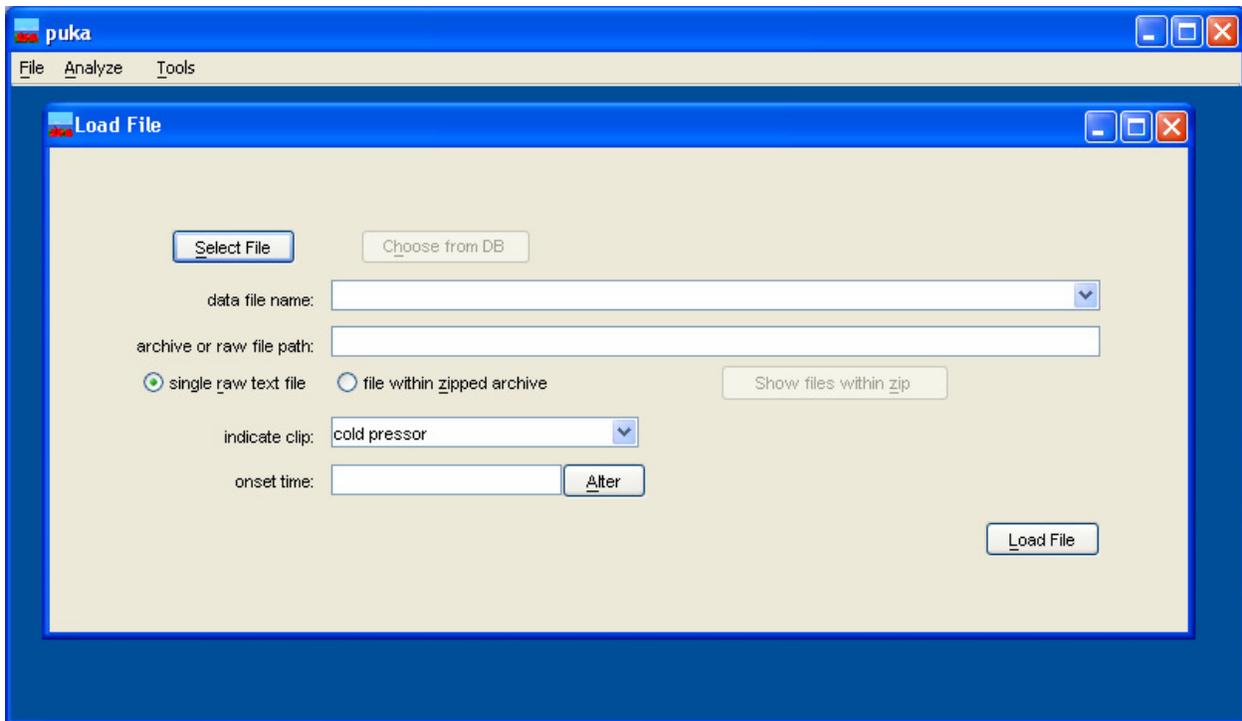


Figure 4. The Load File dialog when puka is not using a database.

Loading Data Using a Database

If a database has previously been created and filled with information (using SubDB) the data can be read using puka. The name of the database, as well as its username and password, is indicated on the Connect to the Database dialog. The database names appearing on the Connect to the Database dialog are entered on the database tab of the Program Preferences dialog. When using a database it is necessary to indicate which subject and session the data corresponds to as well as the data file itself.

The subject and session are selected on the Load Data from Database dialog (shown in Figure 5), which appears when the Choose from DB button on the Load File dialog is clicked. The subject for whom to analyze data is selected from the Select subject list, which contains the name of all subjects entered in the database. When a subject has been selected the sessions for that subject will be listed in the Select session list. When a session has been selected the path to that data file will be displayed in the Path to file or archive box. Click OK to close the dialog.

The Load Data from Database window will close after clicking the OK button, and the Load File window will be filled with information describing the file selected from the database. As shown in Figure 6, the file within zipped archive radio button will be selected, and the data file name drop-down list will contain the names of all of the data files contained in the archive, from which the file to be opened can be selected.

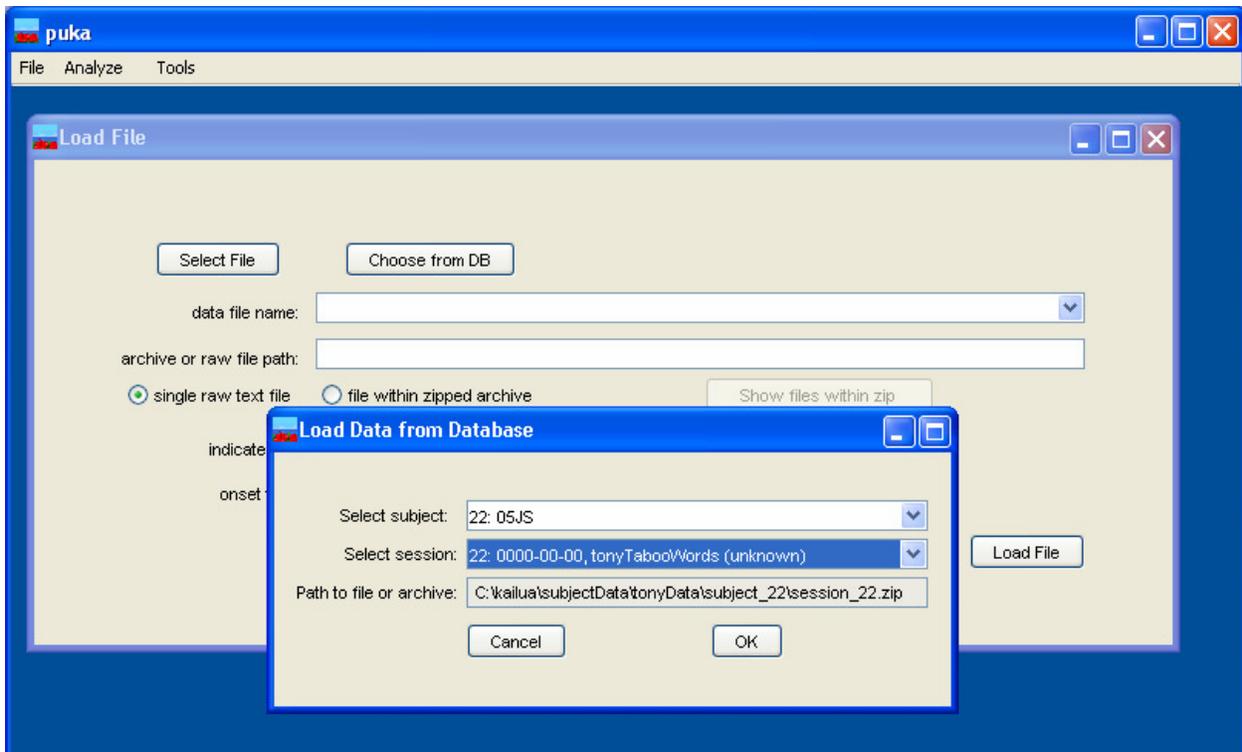


Figure 5. Loading data from a subject database with the Load Data from Database dialog.

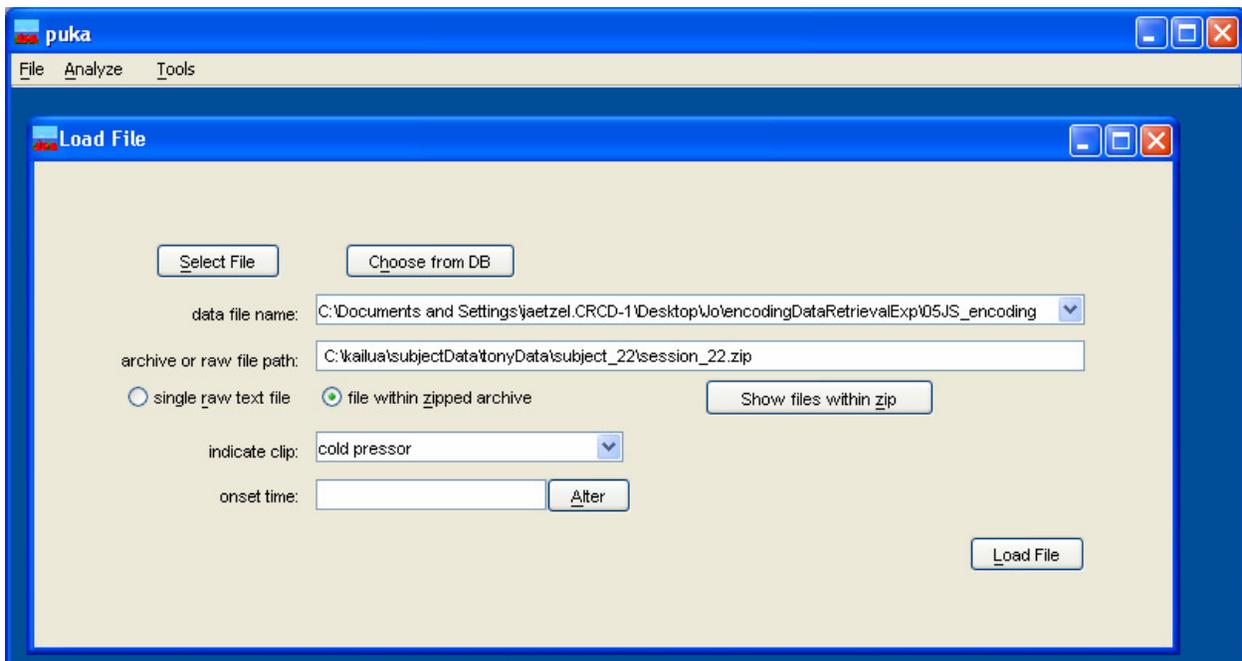


Figure 6. The Load File window after selecting a file from the database.

Indicating the Clip and Onset Point

After the file has been identified (either directly or using the database) the clip needs to be selected. Puka can not automatically detect which clip is present in a data file. Instead, the names of all of the clips

entered on the `clips` tab of the `Program Preferences` are shown in the `indicate clip` box. Select the name of the clip and puka will read the length of the clip as entered into the preferences.

After the clip name has been selected click the `Load File` button to read the file into MATLAB and find the onset time. If the data file contains an onset trigger channel puka will use MATLAB to detect the first trigger point and use this as the stimulus onset time. As shown in Figure 7, puka will display a MATLAB window of the trigger channel data and the value that it chose as the onset time. If the signal contains multiple triggers the first trigger is used. The trigger point chosen by puka is indicated by a small circle on the zero line, as shown in Figure 7. The onset time chosen by puka is displayed in the `onset time` box on the `Load File` window, from which it can be changed if required.

If puka is not able to detect an onset trigger point, either because the onset trigger channel is missing or because it is invalid, then an error message box will be displayed and the onset time must be entered in the `onset time` box. This time needs to be in number of samples, and must be between 1 and the number of sample units in the clip. For example, if the data was recorded at 1000 Hz for two seconds, the onset time must be between 1 and 1999. After entering the onset time click the `Alter` button to save the value.

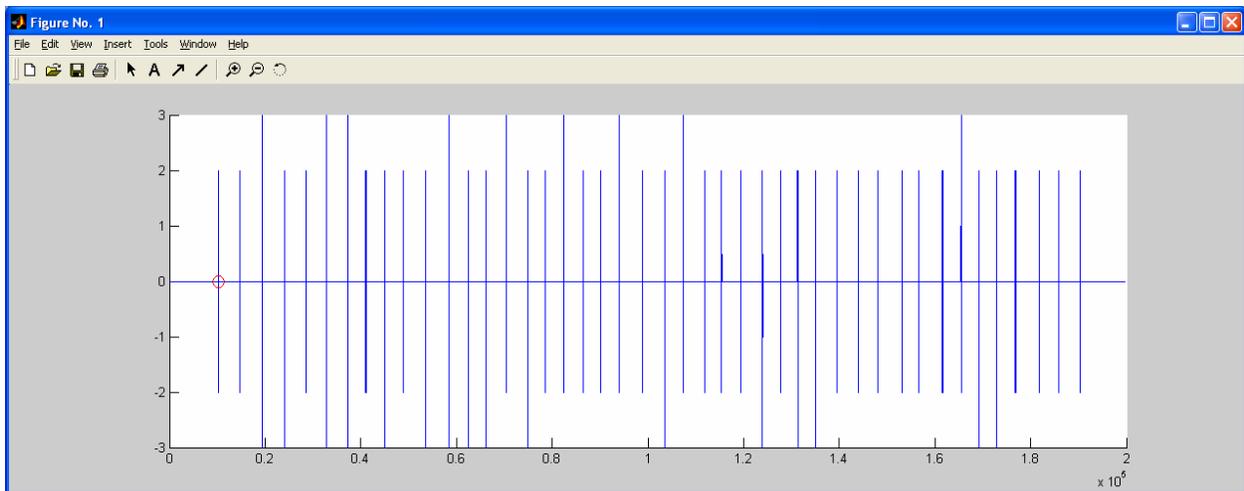


Figure 7. Onset trigger channel showing detected trigger location (circle).

Respiration Analysis

Step 1: Checking the Signal

After a data file has been loaded it can be analyzed. To analyze the respiration signal, select `Analyze`, then `Respiration` from the top menu. A window entitled `Respiration Analysis` will appear, as will a MATLAB window displaying a graph of the respiration channel (Figure 8). This graph is a plot of the entire respiration column from the data file without any processing. The respiration plot should be examined to determine if it is a valid recording that should be analyzed, or has so many artifacts or inconsistencies that it should be discarded.

A short comment describing the appearance of the respiration channel data should be entered in the `Comment` box on the `Respiration Analysis` window. If the signal is too flawed to be analyzed a descriptive comment should be entered and the `Invalid Signal` button clicked. This will close the `Respiration Analysis` window and store the comment and "invalid" inside puka for later writing to the output files and/or database.

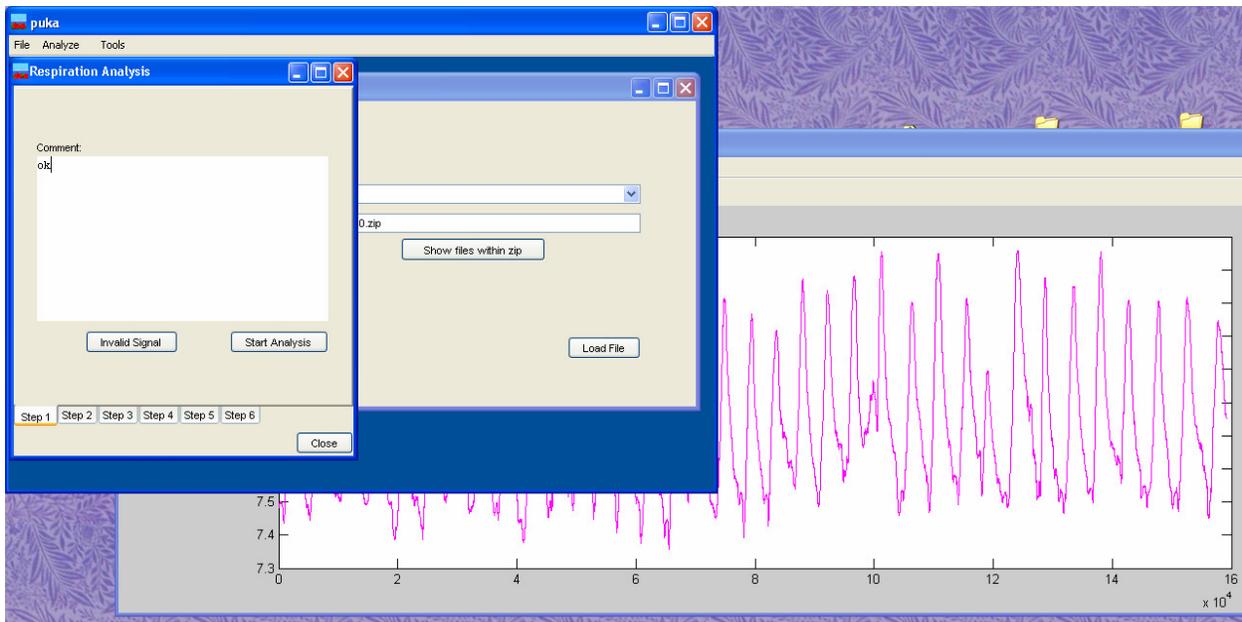


Figure 8. Appearance of puka when starting respiration analysis.

Step 2: Identifying the Breaths

If the signal is valid click the `Start Analysis` button on the `Step 1` tab to change the `Respiration Analysis` window to the `Step 2` tab (Figure 9). `Step 2` consists of applying a peak-detection algorithm to the respiration signal to identify the breaths. The peaks and troughs found by the algorithm are circled for visual verification. Only breaths occurring after the onset trigger (as set when on the `Load Data` dialog) and before the end of the clip (as determined by the length of the selected clip) are marked.

It is important to make sure that all of the breaths are marked. It is often possible to adjust the threshold so that all the breaths, but only the breaths, have been marked. In some cases this will not be possible, most often due to a respiration signal that contains shifts in magnitude (some short breaths and some tall breaths). If a perfect detection is not possible the threshold should be set so that all breaths are identified even if noise is also marked. These extra (invalid) breaths will be eliminated in the next step of the analysis.

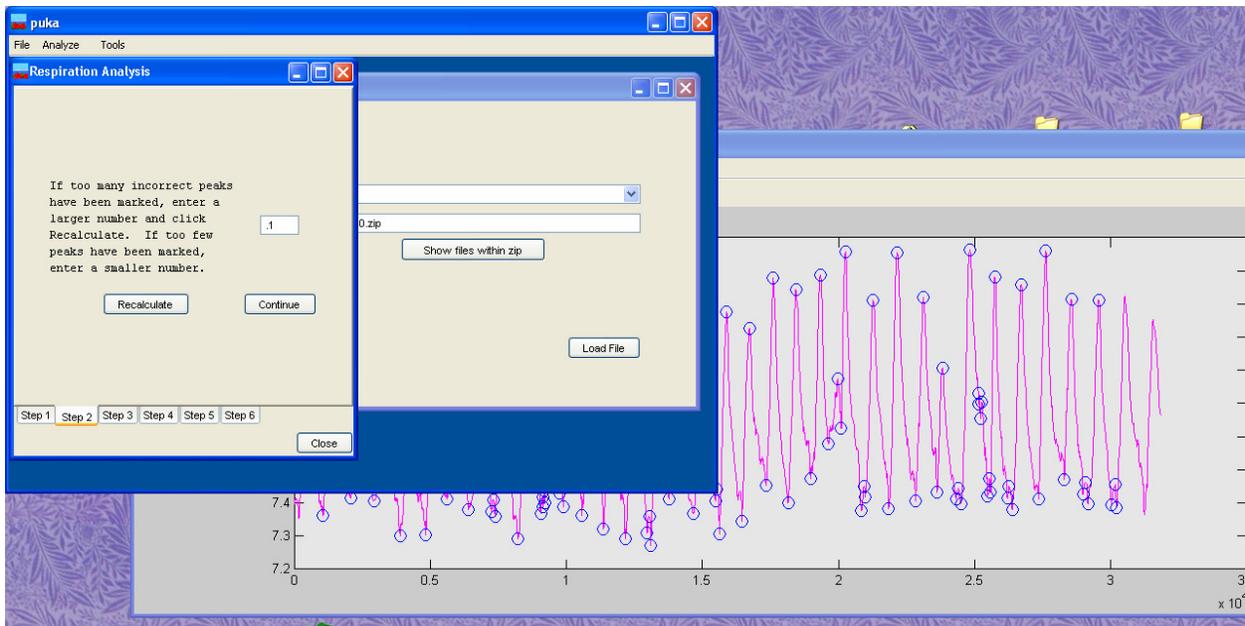


Figure 9. Step 2 of respiration analysis, in which the peak detection algorithm is used to identify the breaths. Too many points have been marked as “breaths,” indicating that the threshold needs to be increased.

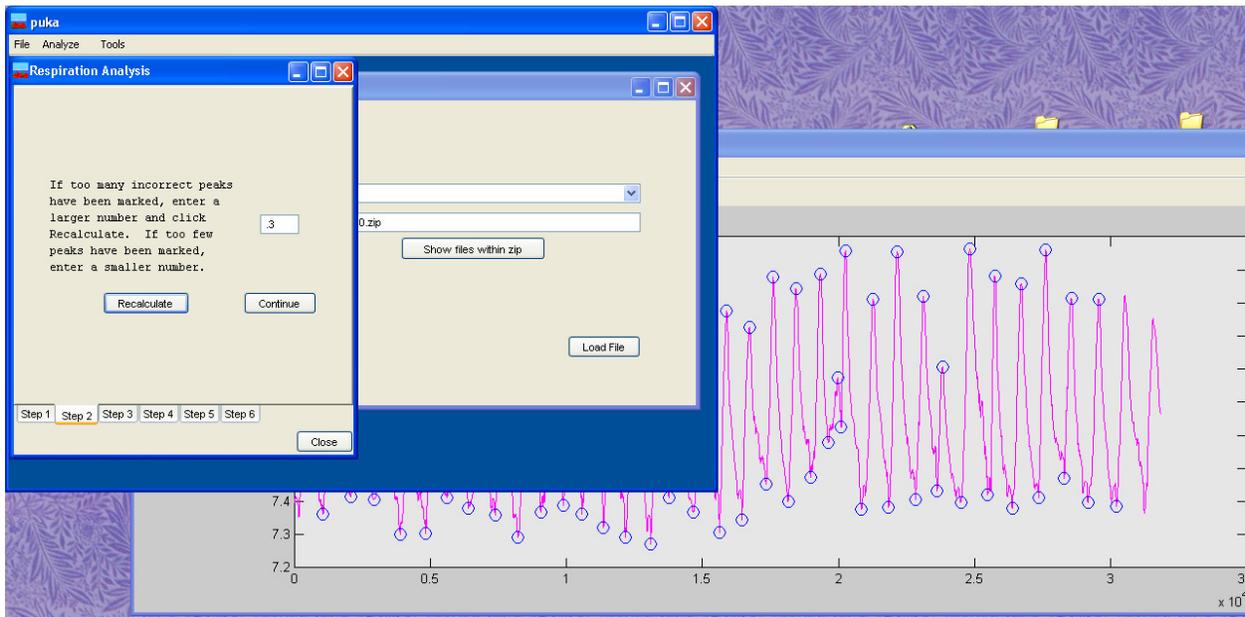


Figure 10. Respiration signal with the peaks and troughs correctly identified. The threshold was raised from 0.1 to 0.3.

Step 3: Checking the Breaths for Validity

After all breaths have been identified and marked with circles at the end of Step 2 they are checked for validity so that points incorrectly marked as breaths are eliminated. It is not always possible to adjust the threshold at Step 2 to a value such that all of the breaths, but no other points, are marked as breaths.

In these cases there will be extra circles on the respiration plot, or “invalid breaths.” For example, the signal in Figure 11 has two segments marked as breaths.

When the Continue button on the Step 2 tab is clicked the plot changes so that numbers and colors distinguish the points, as seen in Figure 12. The peaks and troughs marked by blue X will be included in the analysis, while those marked in red will be ignored. Puka attempts to identify which points should be included and which are invalid, producing the initial graph. If some peaks or troughs were improperly classified check or uncheck the boxes on the Step 3 tab to produce a correct classification of all breaths.

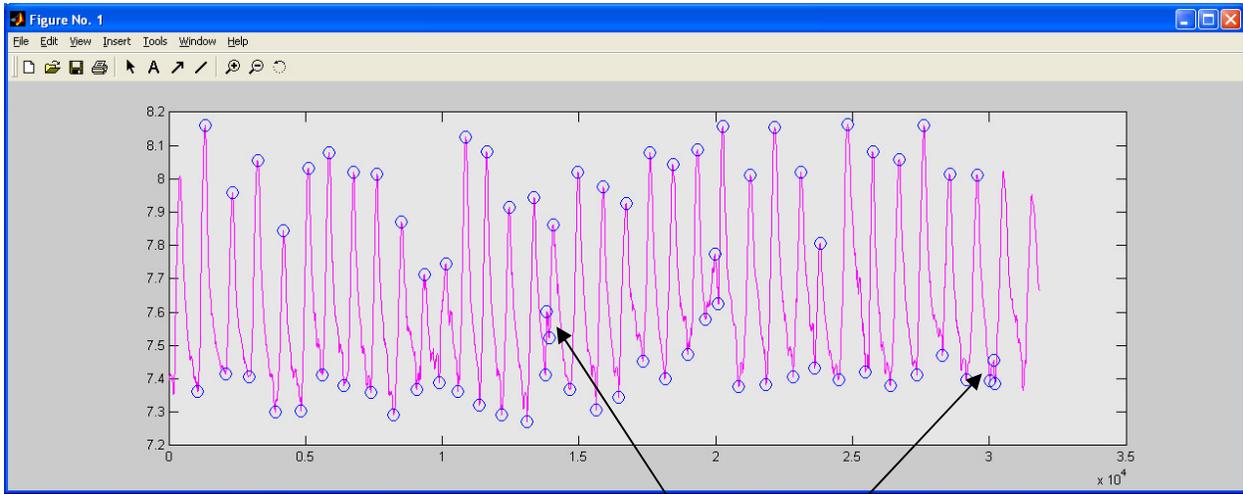


Figure 11. Respiration signal with invalid breaths, as indicated.

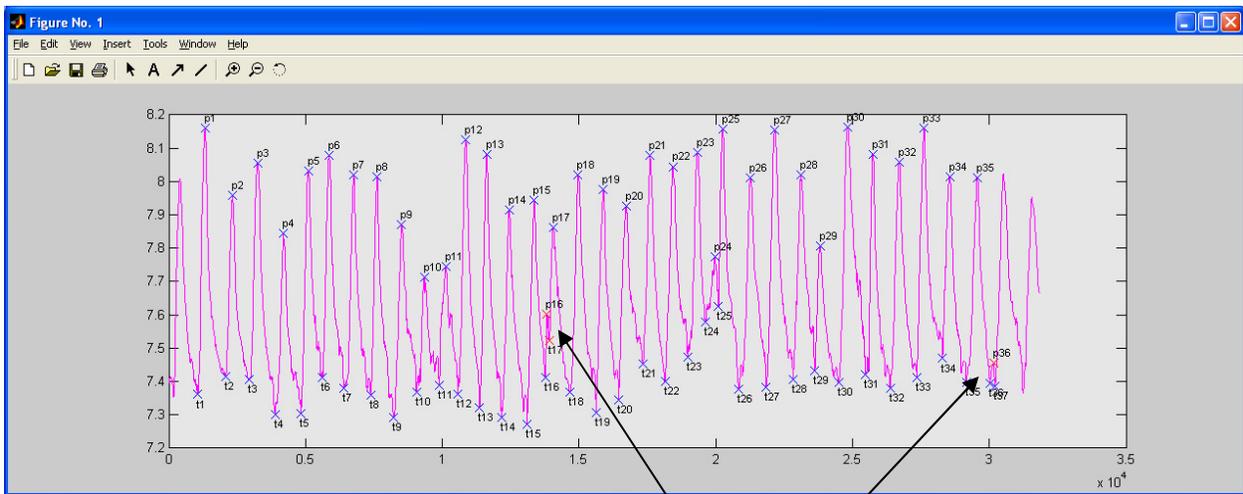


Figure 12. Respiration signal at the start of Step 3. p16, p36, and t17 were correctly marked (as invalid) for exclusion, but t36 was not.

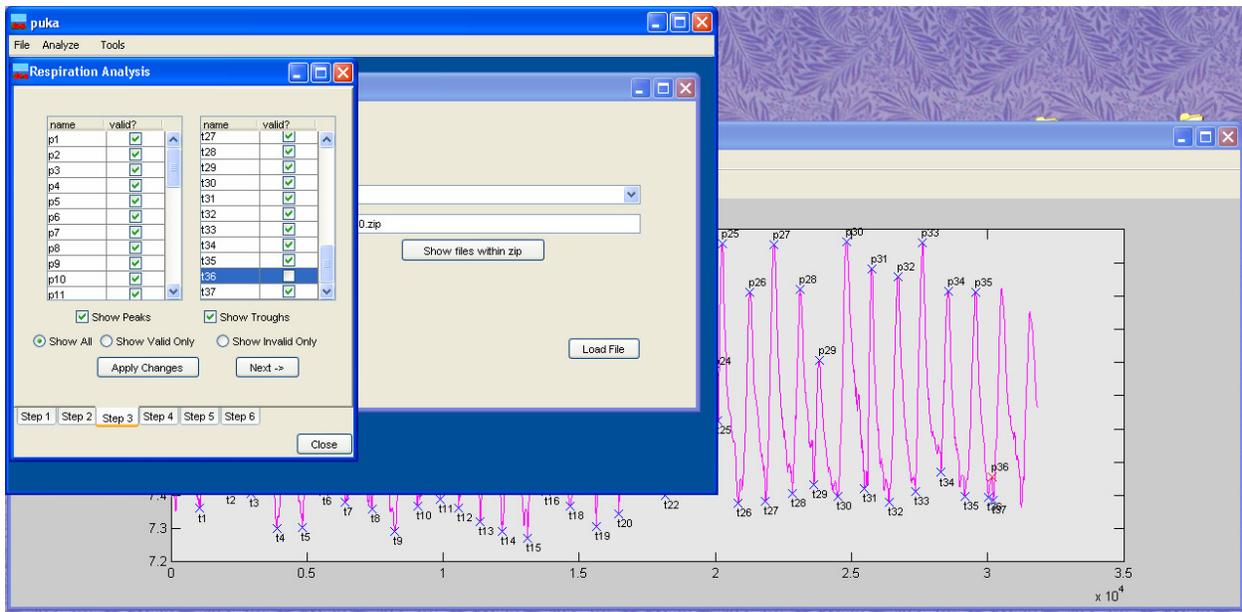


Figure 13. Appearance of puka during Step 3 of the respiration analysis. Trough t36 is unchecked so that it will not be included as a breath.

Step 4: Locating Pauses

During Step 4 puka identifies pauses around each peak and trough marked as valid in Step 3. The Step 4 tab is shown in Figure 14. The plot of the respiration signal changes to show circles at the pauses and X at the peaks and troughs. If no pause is present at a particular peak or trough the circles will be superimposed on the X.

When pauses are present it is generally helpful to move the peak and trough markers to the center of the surrounding pauses to increase the calculation accuracy. This is done by clicking the *Recenter Markers* button, which will move the markers and update the plot.

The Step 4 tab of the Respiration Analysis window is also used to indicate whether the pauses should be included in the statistical calculations. By default, the *Use pauses in calculations* option is selected, but the *no pauses in calculations* option can be selected if desired. If the pauses are not included statistics such as inspiration and expiration time will be based on the peaks and troughs only, but the pause locations will be saved.

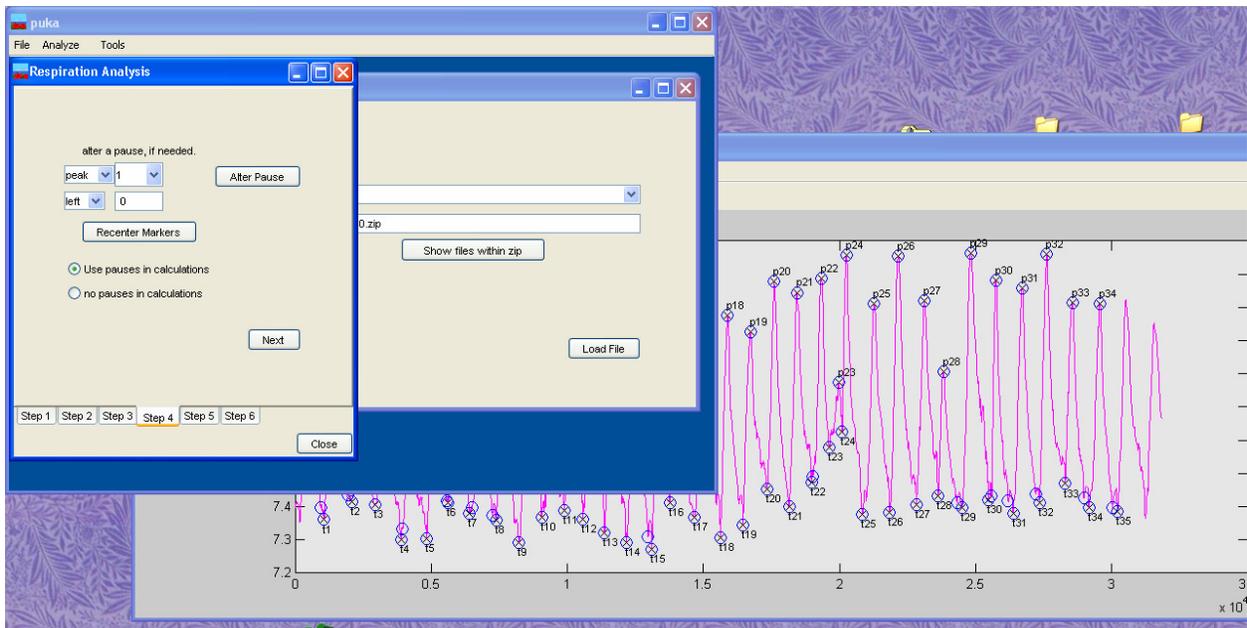


Figure 14. Appearance at Step 4.

Step 5: Statistical Calculations

Finally, results of basic statistical calculations are shown on the Step 5 tab, as shown in Figure 15. These calculations are based on the entire clip, onset trigger to end (as determined by clip length, not recording end). These values are saved with the peak, trough, and pause locations, but should be examined to ensure that they are reasonable, confirming that the analysis was correct.

If the EKG will not be analyzed the respiration results can be saved by selecting File then Save Results from the top menu. Otherwise, the Respiration Analysis window should be left open while the EKG analysis is performed.

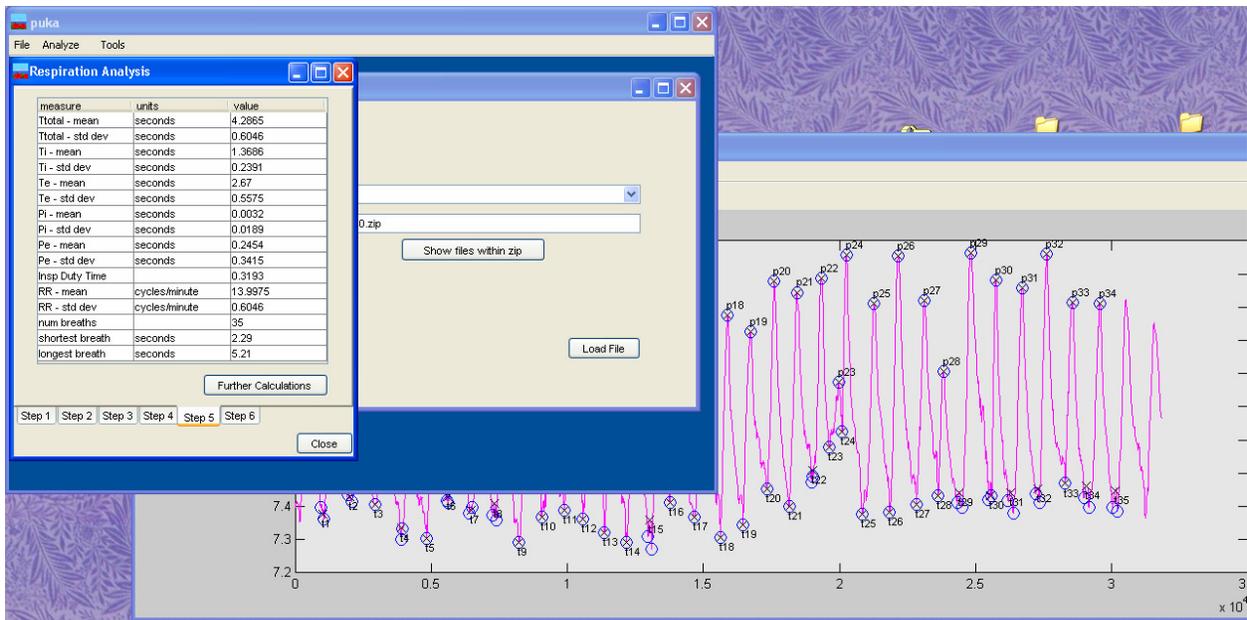


Figure 15. Appearance of puka at Step 5 of respiration signal analysis.

EKG Analysis

Step 1: Checking the Signal

To start the analysis of the EKG signal select *Analyze* then *HRV & RSA* from the top *puka* menus, which will cause the *RSA* window and a plot to appear, as shown in Figure 16. If the data file contains a respiration channel and the peak-valley RSA is to be calculated, the respiration channel must be analyzed before the EKG. The EKG analysis description will continue the previous example, so the *Respiration Analysis* window is visible in the background of Figure 16.

When *HRV & RSA* is selected from the main *puka* menu the raw EKG channel data will be graphed, as shown in Figure 16. The signal should be visually checked for validity before proceeding. In this example the signal is fine, so "ok" is entered into the *Comment* box on the *RSA* window. If the signal had excessive noise or other artifacts and so should not be analyzed, the problem should be described in the *Comment* box and the *Invalid Signal* button clicked, stopping the EKG analysis.

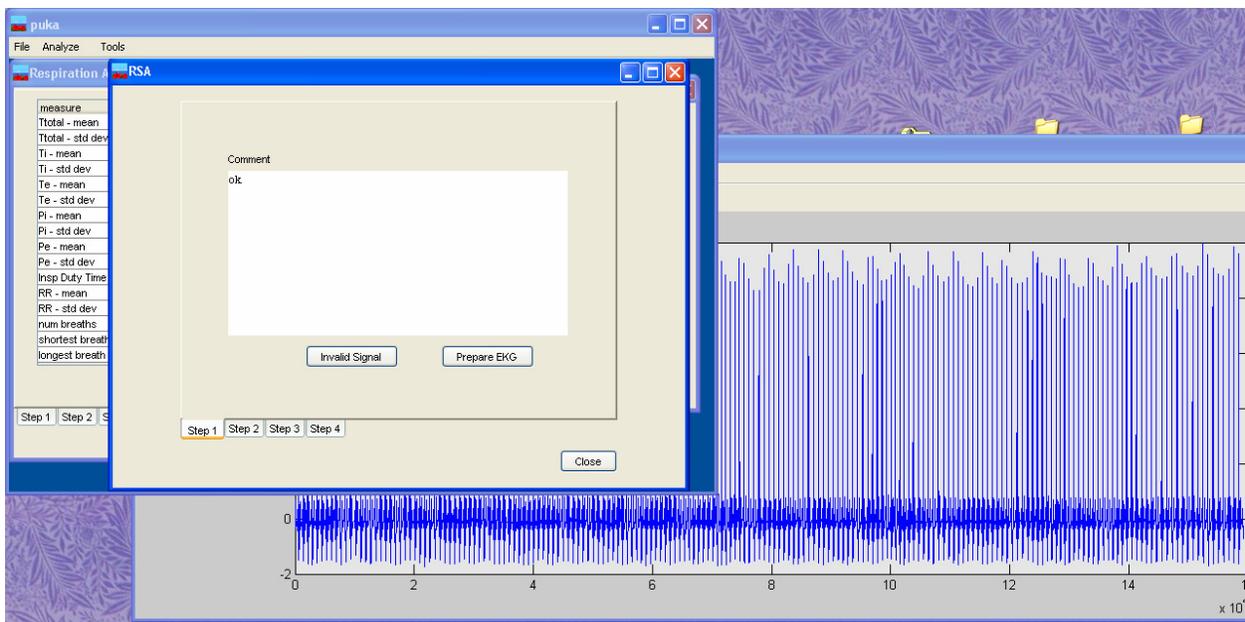


Figure 16. Appearance of *puka* at Step 1 of the EKG analysis.

Step 2: Using the WFDB Software Package to Identify the EKG R Waves

When the *Prepare EKG* button on the *Step 1* tab is clicked *puka* sends the EKG signal to the WFDB Software Package and the screen changes to match Figure 17. Several programs convert the raw data to the binary WFDB Software Package format, according to the EKG characteristics set in the *Preferences*. After the WFDB Software Package has detected the R waves in the EKG MATLAB is used to plot the signal with the peaks marked, as shown in Figure 17.

As with the respiration analysis, the WFDB Software Package only identifies peaks during the clip, which is why the marked peaks end before the end of the signal in Figure 17. The marked peaks should be checked to ensure that they were detected accurately before clicking the *Calculate HRV* button to continue the analysis. There is no way in *puka* to override the R peaks detected by the WFDB Software Package. If errors are present the EKG information entered into the preferences and the WFDB Software Package should be checked.

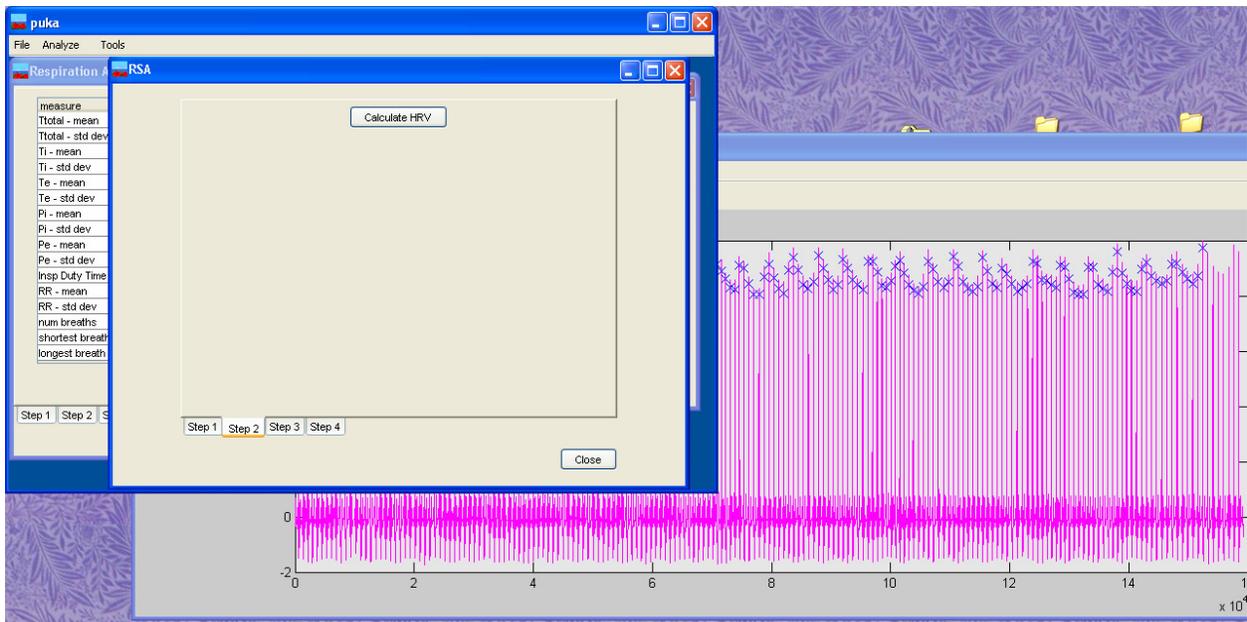


Figure 17. Step 2 of the EKG analysis; EKG signal with the R waves found using the WFDB Software Package marked by x.

Step 3: Calculation of HRV Statistics

When the Calculate HRV button is clicked puka uses the R wave times found by the WFDB Software Package to compute several HRV summary statistics, which are displayed on the Step 3 tab, as shown in Figure 18. The WFDB Software Package values used to calculate each statistic are listed in Table 1.

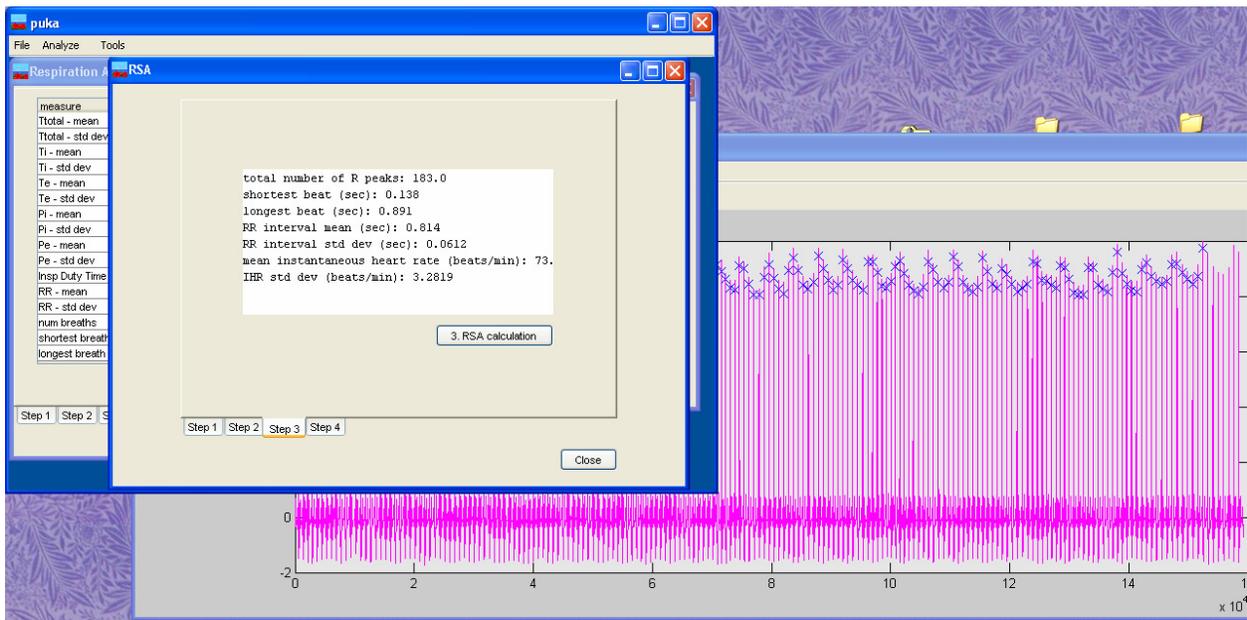


Figure 18. Appearance of puka at Step 3.

Table 1. Source of derived HRV measures.

name	description
heartRateMean	mean of the vector from <code>i_{hr}</code> , calculated in MATLAB
heartRateStdDev	standard deviation of the vector from <code>i_{hr}</code> , calculated in MATLAB
rrMean	mean of the vector from <code>ann2_{rr}</code> , calculated in MATLAB
rrStdDev	standard deviation of the vector from <code>ann2_{rr}</code> , calculated in MATLAB

Step 4: Peak-Valley RSA Calculation

If the respiration signal was analyzed prior to the EKG signal the peak-valley RSA can be calculated, otherwise EKG analysis ends at Step 3. If the respiration analysis results are available, clicking the RSA Calculation button on the Step 3 tab causes Step 4 tab to appear, as shown in Figure 19. The Step 4 tab displays summary statistics and details about the RSA calculation as well as a plot illustrating the results for each breath.

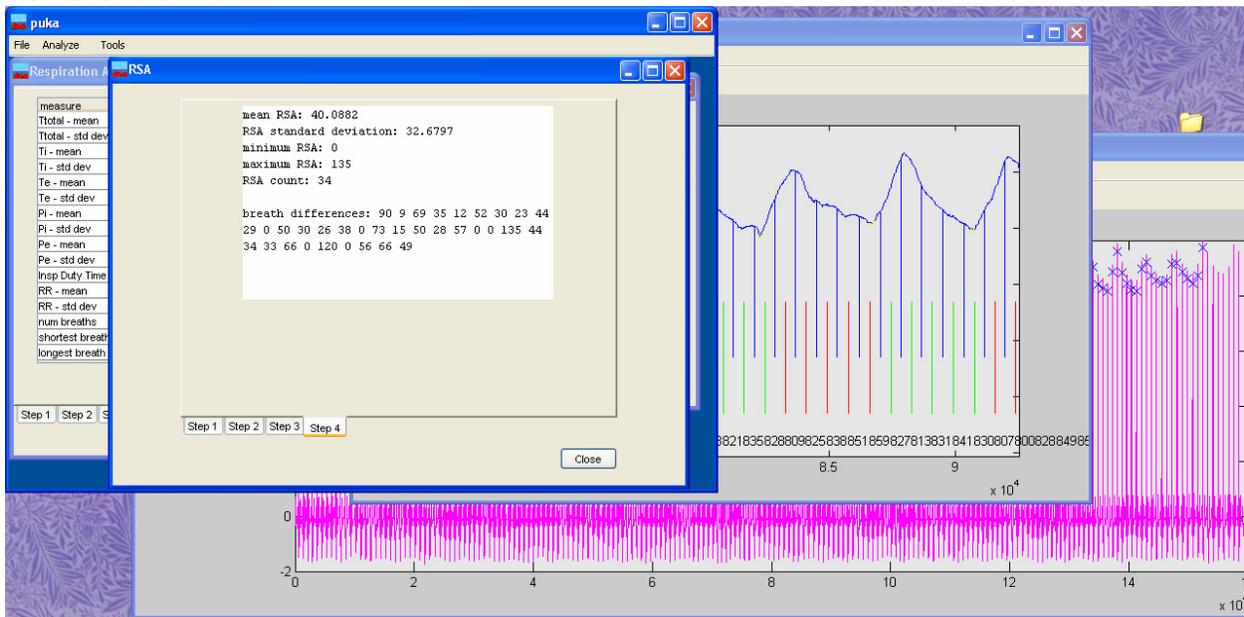


Figure 19. Appearance of puka at analysis Step 4.

Saving Analysis Results

After the respiration and/or EKG signals have been analyzed the results can be saved. The `Select save location` window appears after selecting `File` then `Save Results` from the top menu. This window, shown in Figure 20, is used to save the results to either the database or an xml file.

Puka fills the `Select save location` window with most details of the analysis automatically. Puka checks if respiration and/or EKG data has been analyzed, and then checks the boxes (under `Check to save data type`) if the analysis results are present. The boxes can be unchecked, however, if it is not desired to save all of the analysis results. Also, puka inserts the clip name from the `Load Data` dialog into the `Clip ID` box. If the data was loaded out of a database, the database box will be checked and the subject and session names will be entered in the `Select subject` and `Select session` boxes.

In the Figure 20 example the database was not used and data will not be written into the database, so the database box is left unchecked. Instead, the analysis results will be saved as an xml file so the `XML document` box is checked. Clicking the `Select file` button causes a file save dialog box to appear, in which the name and location of the xml file to create can be specified. Clicking the `Save` button will

then write the analysis results to either the xml document or database, as specified. The xml document created in this example is attached as Appendix B. It can be seen that all derived measures, as well as all located peaks, troughs, and pauses, are saved in the data file.

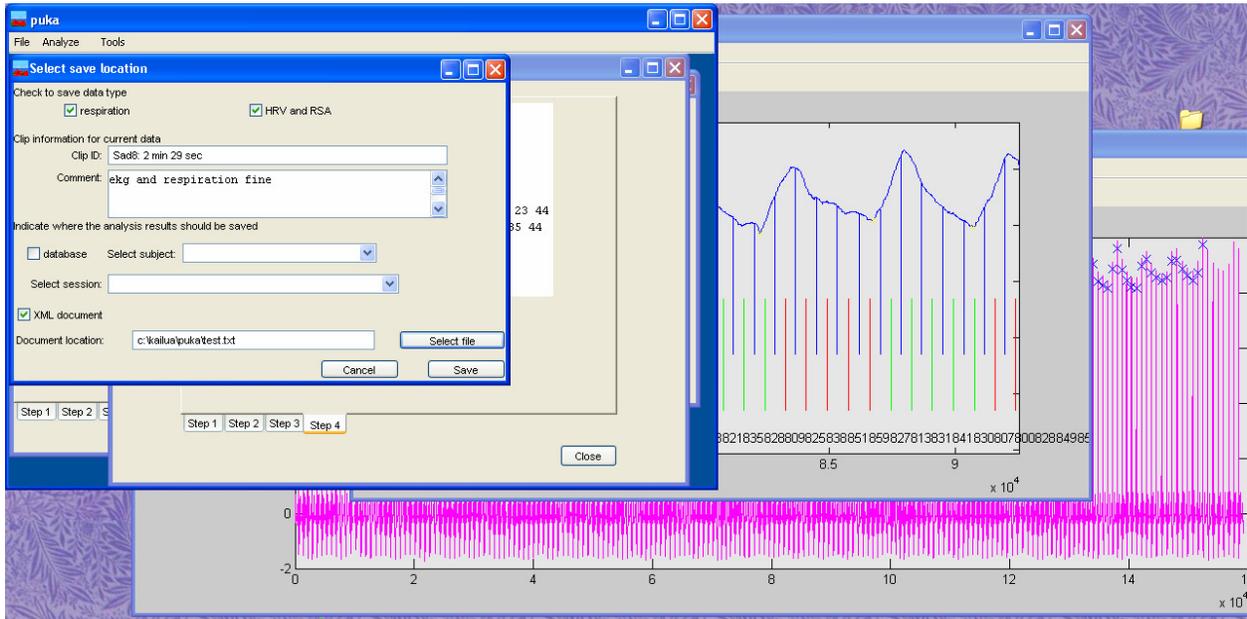


Figure 20. Appearance of puka when saving data.

Appendix A.

SQL commands to create the database.

```
--
-- Table structure for table `subject`
--

CREATE TABLE subject (
  subID smallint(5) unsigned NOT NULL auto_increment,
  PPG varchar(20) default NULL,
  firstName varchar(100) default NULL,
  middleInitial char(1) default NULL,
  lastName varchar(100) default NULL,
  gender char(1) default NULL,
  birth date default NULL,
  education varchar(50) default NULL,
  brainLesion tinyint(1) default NULL,
  ethnicity varchar(50) default NULL,
  hearing varchar(50) default NULL,
  sight varchar(50) default NULL,
  handedness varchar(50) default NULL,
  subjectType varchar(50),
  subjectSubType varchar(50),
  comment blob,
  PRIMARY KEY (subID)
) TYPE=MyISAM;

--
-- Table structure for table `sessionData`
--

CREATE TABLE sessionData (
```

```

subID smallint(5) unsigned NOT NULL default '0',
sessionID smallint(5) unsigned NOT NULL auto_increment,
expDate date NOT NULL default '0000-00-00',
experimenter varchar(100) NOT NULL default '',
comment blob,
expType varchar(100) NOT NULL default '',
age smallint(5) unsigned default '0',
timeStarted TIME,
dataFile varchar(250),
PRIMARY KEY (sessionID)
) TYPE=MyISAM;

```

```

--
-- Table structure for table `heartmeasures`
--

```

```

CREATE TABLE heartmeasures (
  subID smallint(5) unsigned default NULL,
  sessionID smallint(5) unsigned default NULL,
  clipID varchar(50) default NULL,
  dataComment blob,
  heartRateMean double default NULL,
  heartRateStdDev double default NULL,
  RSAMean double default NULL,
  RSAStdDev double default NULL,
  RRMean double default NULL,
  RRStdDev double default NULL,
  comment blob,
  isValid tinyint(1) default NULL,
  RPeakList text,
  numRPeaks double default NULL,
  shortestBeat double default NULL,
  longestBeat double default NULL,
  RSAMin double default NULL,
  RSAMax double default NULL,
  RSACount double default NULL,
  RSAList text
) TYPE=MyISAM;

```

```

--
-- Table structure for table `respmeasures`
--

```

```

CREATE TABLE respmeasures (
  subID smallint(5) unsigned default NULL,
  sessionID smallint(5) unsigned default NULL,
  clipID varchar(50) default NULL,
  dataComment blob,
  totalBreathMean double default NULL,
  totalBreathStdDev double default NULL,
  inspTimeMean double default NULL,
  inspTimeStdDev double default NULL,
  expTimeMean double default NULL,
  expTimeStdDev double default NULL,
  postInspPauseMean double default NULL,
  postInspPauseStdDev double default NULL,
  postExpPauseMean double default NULL,
  postExpPauseStdDev double default NULL,
  inspDutyTimeMean double default NULL,
  inspDutyTimeStdDev double default NULL,
  respRateMean double default NULL,
  respRateStdDev double default NULL,
  comment blob,
  isValid tinyint(1) default NULL,
  peakPauseList text,
  troughPauseList text,
  peakList text,
  troughList text,
  numBreaths double default NULL,
  shortestBreath double default NULL,
  longestBreath double default NULL
) TYPE=MyISAM;

```

```

--
-- Table structure for table `comboboxes`
--

CREATE TABLE comboboxes (
  boxID varchar(20) NOT NULL default '',
  boxChoice varchar(50) NOT NULL default ''
) TYPE=MyISAM;

--
-- Dumping data for table `comboboxes`
--

INSERT INTO comboboxes VALUES ('Education','unknown');
INSERT INTO comboboxes VALUES ('Experimenter','unknown');
INSERT INTO comboboxes VALUES ('Ethnicity','white');
INSERT INTO comboboxes VALUES ('Ethnicity','unknown');
INSERT INTO comboboxes VALUES ('Hearing','unknown');
INSERT INTO comboboxes VALUES ('Sight','unknown');
INSERT INTO comboboxes VALUES ('Handedness','unknown');
INSERT INTO comboboxes VALUES ('ExperimentType','unknown');
INSERT INTO comboboxes VALUES ('ExperimentType','Erica Sound Clips');
INSERT INTO comboboxes VALUES ('subjectSubType','not entered');
INSERT INTO comboboxes VALUES ('SubjectType','1 normal control');
INSERT INTO comboboxes VALUES ('SubjectType','2 b. amygdala');
INSERT INTO comboboxes VALUES ('SubjectType','3 left amygdala');
INSERT INTO comboboxes VALUES ('SubjectType','4 right amygdala');
INSERT INTO comboboxes VALUES ('SubjectType','5 VM prefrontal');
INSERT INTO comboboxes VALUES ('SubjectType','6 right hemisphere');
INSERT INTO comboboxes VALUES ('SubjectType','7 BD control');
INSERT INTO comboboxes VALUES ('SubjectType','8 excluded');

--
-- Table structure for table `calcekg`
--

CREATE TABLE calcekg (
  subID smallint(5) unsigned NOT NULL default '0',
  subType varchar(50) default NULL,
  gender char(1) default NULL,
  age smallint(5) unsigned default NULL,
  clip varchar(50) default NULL,
  RPeakList blob,
  RSAList blob
) TYPE=MyISAM;

--
-- Table structure for table `calcresp`
--

CREATE TABLE calcresp (
  subID smallint(5) unsigned NOT NULL default '0',
  subType varchar(50) default NULL,
  gender char(1) default NULL,
  age smallint(5) unsigned default NULL,
  clip varchar(50) default NULL,
  peakList blob,
  troughList blob,
  peakPauseList blob,
  troughPauseList blob
) TYPE=MyISAM;

```

Appendix B.

XML document containing the results of the example analysis.

```

<record subID="" sessionID="" clipID="Sad8: 2 min 29 sec">
<dataComment>ekg and respiration fine</dataComment>
<respMeasures>
<totalBreathMean>4.28647058823529381044181718607433140277862548828125</totalBreathMean>
<totalBreathStdDev>0.60462927460948756586134322787984274327754974365234375</totalBreathStdDev>
<inspTimeMean>1.368636363636363650897465049638412892818450927734375</inspTimeMean>

```

<inspTimeStdDev>0.2391269038321778561684283204158418811857700347900390625</inspTimeStdDev>
<expTimeMean>2.6699999999999999289457264239899814128875732421875</expTimeMean>
<expTimeStdDev>0.5574789794243366625181579365744255483150482177734375</expTimeStdDev>
<postInspPauseMean>0.0032352941176470588653668603029700534534640610218048095703125</postInspPauseMean>
<postInspPauseStdDev>0.0188648443656759708109138529152914998121559619903564453125</postInspPauseStdDev>
<postExpPauseMean>0.2454285714285714126159376746727502904832363128662109375</postExpPauseMean>
<postExpPauseStdDev>0.341519897423465501606898442332749255001544952392578125</postExpPauseStdDev>
<inspDutyTimeMean>0.319292139176865441552166657857014797627925872802734375</inspDutyTimeMean>
<inspDutyTimeStdDev>0</inspDutyTimeStdDev>
<respRateMean>13.997529847673941816310616559349000453948974609375</respRateMean>
<respRateStdDev>0.60462927460948756586134322787984274327754974365234375</respRateStdDev>
<respComment>ok</respComment>
<respIsValid>true</respIsValid>
<numBreaths>35</numBreaths>
<shortestBreath>2.29000000000000003552713678800500929355621337890625</shortestBreath>
<longestBreath>5.20999999999999996447286321199499070644378662109375</longestBreath>
<peak>6660</peak> <trough>94855</trough>
<peak>11650</peak> <trough>98155</trough>
<peak>16220</peak> <trough>100445</trough>
<peak>20980</peak> <trough>104250</trough>
<peak>25530</peak> <trough>109215</trough>
<peak>29345</peak> <trough>114110</trough>
<peak>33790</peak> <trough>118070</trough>
<peak>38085</peak> <trough>122440</trough>
<peak>42575</peak> <trough>127385</trough>
<peak>46760</peak> <trough>131895</trough>
<peak>50625</peak> <trough>136720</trough>
<peak>54290</peak> <trough>141465</trough>
<peak>58130</peak> <trough>145870</trough>
<peak>62250</peak> <trough>151080</trough>
<peak>66800</peak> <peakPause>1332</peakPause>
<peak>70355</peak> <peakPause>1332</peakPause>
<peak>74840</peak> <peakPause>2330</peakPause>
<peak>79450</peak> <peakPause>2330</peakPause>
<peak>83640</peak> <peakPause>3244</peakPause>
<peak>88005</peak> <peakPause>3244</peakPause>
<peak>92140</peak> <peakPause>4196</peakPause>
<peak>96665</peak> <peakPause>4196</peakPause>
<peak>99835</peak> <peakPause>5106</peakPause>
<peak>101275</peak> <peakPause>5106</peakPause>
<peak>106385</peak> <peakPause>5869</peakPause>
<peak>110800</peak> <peakPause>5869</peakPause>
<peak>115530</peak> <peakPause>6758</peakPause>
<peak>119115</peak> <peakPause>6758</peakPause>
<peak>124100</peak> <peakPause>7617</peakPause>
<peak>128730</peak> <peakPause>7617</peakPause>
<peak>133530</peak> <peakPause>8515</peakPause>
<peak>138065</peak> <peakPause>8515</peakPause>
<peak>142750</peak> <peakPause>9352</peakPause>
<peak>147800</peak> <peakPause>9352</peakPause>
<trough>5340</trough> <peakPause>10125</peakPause>
<trough>10355</trough> <peakPause>10125</peakPause>
<trough>14665</trough> <peakPause>10858</peakPause>
<trough>19470</trough> <peakPause>10858</peakPause>
<trough>24170</trough> <peakPause>11626</peakPause>
<trough>28140</trough> <peakPause>11626</peakPause>
<trough>32065</trough> <peakPause>12450</peakPause>
<trough>36920</trough> <peakPause>12450</peakPause>
<trough>41140</trough> <peakPause>13360</peakPause>
<trough>45300</trough> <peakPause>13360</peakPause>
<trough>49460</trough> <peakPause>14049</peakPause>
<trough>52880</trough> <peakPause>14071</peakPause>
<trough>56830</trough> <peakPause>14968</peakPause>
<trough>60840</trough> <peakPause>14968</peakPause>
<trough>65530</trough> <peakPause>15890</peakPause>
<trough>68825</trough> <peakPause>15890</peakPause>
<trough>73395</trough> <peakPause>16728</peakPause>
<trough>78085</trough> <peakPause>16728</peakPause>
<trough>82250</trough> <peakPause>17601</peakPause>
<trough>86750</trough> <peakPause>17601</peakPause>
<trough>90710</trough> <peakPause>18428</peakPause>

<peakPause>18428</peakPause>
<peakPause>19333</peakPause>
<peakPause>19333</peakPause>
<peakPause>19967</peakPause>
<peakPause>19967</peakPause>
<peakPause>20255</peakPause>
<peakPause>20255</peakPause>
<peakPause>21277</peakPause>
<peakPause>21277</peakPause>
<peakPause>22160</peakPause>
<peakPause>22160</peakPause>
<peakPause>23106</peakPause>
<peakPause>23106</peakPause>
<peakPause>23823</peakPause>
<peakPause>23823</peakPause>
<peakPause>24820</peakPause>
<peakPause>24820</peakPause>
<peakPause>25746</peakPause>
<peakPause>25746</peakPause>
<peakPause>26706</peakPause>
<peakPause>26706</peakPause>
<peakPause>27613</peakPause>
<peakPause>27613</peakPause>
<peakPause>28550</peakPause>
<peakPause>28550</peakPause>
<peakPause>29560</peakPause>
<peakPause>29560</peakPause>
<troughPause>958</troughPause>
<troughPause>1068</troughPause>
<troughPause>1928</troughPause>
<troughPause>2071</troughPause>
<troughPause>2933</troughPause>
<troughPause>2933</troughPause>
<troughPause>3894</troughPause>
<troughPause>3946</troughPause>
<troughPause>4834</troughPause>
<troughPause>4834</troughPause>
<troughPause>5589</troughPause>
<troughPause>5628</troughPause>
<troughPause>6413</troughPause>
<troughPause>6504</troughPause>
<troughPause>7246</troughPause>
<troughPause>7384</troughPause>
<troughPause>8228</troughPause>
<troughPause>8228</troughPause>
<troughPause>9060</troughPause>
<troughPause>9060</troughPause>
<troughPause>9892</troughPause>
<troughPause>9892</troughPause>
<troughPause>10576</troughPause>
<troughPause>10576</troughPause>
<troughPause>11366</troughPause>
<troughPause>11366</troughPause>
<troughPause>12168</troughPause>
<troughPause>12168</troughPause>
<troughPause>12971</troughPause>
<troughPause>13106</troughPause>
<troughPause>13765</troughPause>
<troughPause>13765</troughPause>
<troughPause>14679</troughPause>
<troughPause>14679</troughPause>
<troughPause>15617</troughPause>
<troughPause>15617</troughPause>
<troughPause>16450</troughPause>
<troughPause>16450</troughPause>
<troughPause>17350</troughPause>
<troughPause>17350</troughPause>
<troughPause>18142</troughPause>
<troughPause>18142</troughPause>
<troughPause>18971</troughPause>
<troughPause>19027</troughPause>
<troughPause>19631</troughPause>
<troughPause>19631</troughPause>

<troughPause>20089</troughPause>
<troughPause>20089</troughPause>
<troughPause>20850</troughPause>
<troughPause>20850</troughPause>
<troughPause>21843</troughPause>
<troughPause>21843</troughPause>
<troughPause>22822</troughPause>
<troughPause>22822</troughPause>
<troughPause>23614</troughPause>
<troughPause>23614</troughPause>
<troughPause>24316</troughPause>
<troughPause>24488</troughPause>
<troughPause>25477</troughPause>
<troughPause>25578</troughPause>
<troughPause>26195</troughPause>
<troughPause>26379</troughPause>
<troughPause>27210</troughPause>
<troughPause>27344</troughPause>
<troughPause>28293</troughPause>
<troughPause>28293</troughPause>
<troughPause>28990</troughPause>
<troughPause>29174</troughPause>
<troughPause>30037</troughPause>
<troughPause>30216</troughPause>
</respMeasures>

<heartMeasures>
<NumRPeaks>183</NumRPeaks>
<ShortestBeat>0.1380000000000000115463194561
0162802040576934814453125</ShortestBeat>
<LongestBeat>0.89100000000000001421085471520
20037174224853515625</LongestBeat>
<heartRateMean>73.50497071823203043550165602
93734073638916015625</heartRateMean>
<heartRateStdDev>3.2819459634447278872926290
2052141726016998291015625</heartRateStdDev>
<rrMean>0.8140054644808742967398984546889550
983905792236328125</rrMean>
<rrStdDev>0.06116151354574336784786225962307
07131326198577880859375</rrStdDev>
<hmComment>ok</hmComment>
<hmIsValid>true</hmIsValid>
<rsaMean>40.08823529411764496899195364676415
920257568359375</rsaMean>
<rsaStdDev>32.679682917362015359685756266117
095947265625</rsaStdDev>
<RSAMin>0</RSAMin>
<RSAMax>135</RSAMax>
<RSACount>34</RSACount>
<RPeak>3637</RPeak>
<RPeak>4436</RPeak>
<RPeak>5242</RPeak>
<RPeak>6048</RPeak>
<RPeak>6828</RPeak>
<RPeak>7616</RPeak>
<RPeak>8432</RPeak>
<RPeak>9277</RPeak>
<RPeak>10147</RPeak>
<RPeak>11004</RPeak>
<RPeak>11831</RPeak>
<RPeak>12657</RPeak>
<RPeak>13492</RPeak>
<RPeak>14325</RPeak>
<RPeak>15160</RPeak>
<RPeak>15962</RPeak>
<RPeak>16747</RPeak>
<RPeak>17563</RPeak>
<RPeak>18399</RPeak>
<RPeak>19244</RPeak>
<RPeak>20098</RPeak>
<RPeak>20930</RPeak>
<RPeak>21747</RPeak>
<RPeak>22584</RPeak>

<RPeak>23427</RPeak>
<RPeak>24279</RPeak>
<RPeak>25104</RPeak>
<RPeak>25922</RPeak>
<RPeak>26744</RPeak>
<RPeak>27568</RPeak>
<RPeak>28398</RPeak>
<RPeak>29206</RPeak>
<RPeak>30015</RPeak>
<RPeak>30849</RPeak>
<RPeak>31705</RPeak>
<RPeak>32565</RPeak>
<RPeak>33398</RPeak>
<RPeak>34212</RPeak>
<RPeak>35037</RPeak>
<RPeak>35881</RPeak>
<RPeak>36723</RPeak>
<RPeak>37549</RPeak>
<RPeak>38351</RPeak>
<RPeak>39158</RPeak>
<RPeak>39969</RPeak>
<RPeak>40791</RPeak>
<RPeak>41616</RPeak>
<RPeak>42429</RPeak>
<RPeak>43244</RPeak>
<RPeak>44078</RPeak>
<RPeak>44930</RPeak>
<RPeak>45787</RPeak>
<RPeak>46626</RPeak>
<RPeak>47452</RPeak>
<RPeak>48296</RPeak>
<RPeak>49151</RPeak>
<RPeak>50004</RPeak>
<RPeak>50841</RPeak>
<RPeak>51684</RPeak>
<RPeak>52519</RPeak>
<RPeak>53347</RPeak>
<RPeak>54134</RPeak>
<RPeak>54902</RPeak>
<RPeak>55685</RPeak>
<RPeak>56492</RPeak>
<RPeak>57310</RPeak>
<RPeak>58098</RPeak>
<RPeak>58889</RPeak>
<RPeak>59694</RPeak>
<RPeak>60505</RPeak>
<RPeak>61323</RPeak>
<RPeak>62115</RPeak>
<RPeak>62899</RPeak>
<RPeak>63696</RPeak>
<RPeak>64499</RPeak>
<RPeak>65309</RPeak>
<RPeak>66109</RPeak>
<RPeak>66890</RPeak>
<RPeak>67680</RPeak>
<RPeak>68490</RPeak>
<RPeak>69309</RPeak>
<RPeak>70110</RPeak>
<RPeak>70849</RPeak>
<RPeak>71581</RPeak>
<RPeak>72300</RPeak>
<RPeak>73045</RPeak>
<RPeak>73826</RPeak>
<RPeak>74606</RPeak>
<RPeak>75387</RPeak>
<RPeak>76198</RPeak>
<RPeak>77036</RPeak>
<RPeak>77889</RPeak>
<RPeak>78731</RPeak>
<RPeak>79551</RPeak>
<RPeak>80374</RPeak>
<RPeak>81195</RPeak>
<RPeak>82030</RPeak>
<RPeak>82858</RPeak>
<RPeak>83667</RPeak>
<RPeak>84492</RPeak>
<RPeak>85330</RPeak>
<RPeak>86181</RPeak>
<RPeak>87040</RPeak>
<RPeak>87867</RPeak>
<RPeak>88680</RPeak>
<RPeak>89511</RPeak>
<RPeak>90352</RPeak>
<RPeak>91182</RPeak>
<RPeak>91989</RPeak>
<RPeak>92789</RPeak>
<RPeak>93617</RPeak>
<RPeak>94466</RPeak>
<RPeak>95323</RPeak>
<RPeak>96164</RPeak>
<RPeak>96960</RPeak>
<RPeak>97763</RPeak>
<RPeak>98568</RPeak>
<RPeak>99350</RPeak>
<RPeak>100099</RPeak>
<RPeak>100819</RPeak>
<RPeak>101524</RPeak>
<RPeak>102242</RPeak>
<RPeak>103005</RPeak>
<RPeak>103801</RPeak>
<RPeak>104641</RPeak>
<RPeak>105497</RPeak>
<RPeak>106326</RPeak>
<RPeak>107156</RPeak>
<RPeak>108005</RPeak>
<RPeak>108873</RPeak>
<RPeak>109746</RPeak>
<RPeak>110562</RPeak>
<RPeak>111350</RPeak>
<RPeak>112147</RPeak>
<RPeak>112948</RPeak>
<RPeak>113766</RPeak>
<RPeak>114588</RPeak>
<RPeak>115395</RPeak>
<RPeak>116198</RPeak>
<RPeak>117023</RPeak>
<RPeak>117859</RPeak>
<RPeak>118695</RPeak>
<RPeak>119519</RPeak>
<RPeak>120366</RPeak>
<RPeak>121233</RPeak>
<RPeak>122123</RPeak>
<RPeak>123007</RPeak>
<RPeak>123836</RPeak>
<RPeak>124643</RPeak>
<RPeak>125451</RPeak>
<RPeak>126252</RPeak>
<RPeak>127032</RPeak>
<RPeak>127778</RPeak>
<RPeak>128495</RPeak>
<RPeak>129210</RPeak>
<RPeak>129965</RPeak>
<RPeak>130748</RPeak>
<RPeak>131552</RPeak>
<RPeak>132387</RPeak>
<RPeak>133204</RPeak>
<RPeak>134011</RPeak>
<RPeak>134841</RPeak>
<RPeak>135680</RPeak>
<RPeak>136500</RPeak>
<RPeak>137298</RPeak>
<RPeak>138076</RPeak>
<RPeak>138848</RPeak>
<RPeak>139634</RPeak>
<RPeak>140446</RPeak>
<RPeak>141274</RPeak>

```

<RPeak>142102</RPeak>
<RPeak>142922</RPeak>
<RPeak>143764</RPeak>
<RPeak>144629</RPeak>
<RPeak>145515</RPeak>
<RPeak>146393</RPeak>
<RPeak>147254</RPeak>
<RPeak>148096</RPeak>
<RPeak>148956</RPeak>
<RPeak>149835</RPeak>
<RPeak>150722</RPeak>
<RPeak>151613</RPeak>
<RPeak>152462</RPeak>
<RSA>90</RSA>
<RSA>9</RSA>
<RSA>69</RSA>
<RSA>35</RSA>
<RSA>12</RSA>
<RSA>52</RSA>
<RSA>30</RSA>
<RSA>23</RSA>
<RSA>44</RSA>
<RSA>29</RSA>
<RSA>0</RSA>
<RSA>50</RSA>
<RSA>30</RSA>
<RSA>26</RSA>
<RSA>38</RSA>
<RSA>0</RSA>
<RSA>73</RSA>
<RSA>15</RSA>
<RSA>50</RSA>
<RSA>28</RSA>
<RSA>57</RSA>
<RSA>0</RSA>
<RSA>0</RSA>
<RSA>135</RSA>
<RSA>44</RSA>
<RSA>34</RSA>
<RSA>33</RSA>
<RSA>66</RSA>
<RSA>0</RSA>
<RSA>120</RSA>
<RSA>0</RSA>
<RSA>56</RSA>
<RSA>66</RSA>
<RSA>49</RSA>
</heartMeasures>
</record>

```

References

- BIOPAC Systems, Inc. Available at <http://biopac.com/index.html>.
- Goldberger, A. L., L. A. Amaral, et al. (2000). "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." *Circulation* **101**(23): E215-20.