Research Project 2012

Danfei Xu

Developing a Technique for Classifying Pain-Related SEP Waveforms

I. Introduction

Somatosensory evoked potentials (SEPs) can be obtained by measuring voltage resulting from ionic current fluctuations within neurons in the brain, which reflect the activity of the nervous system in the entire human body.

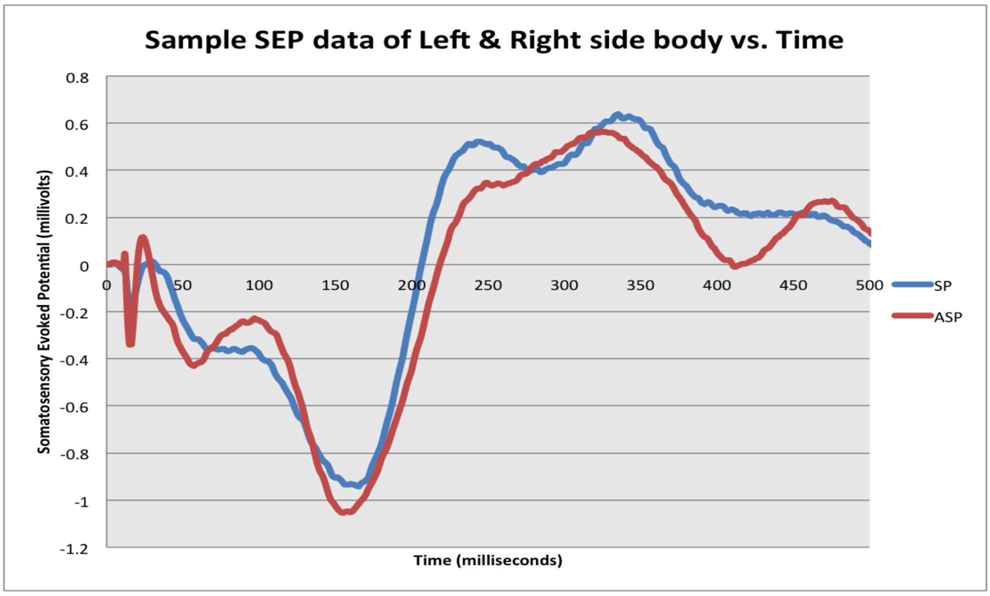


Figure 1.1: Sample SEP data.

Chronic pain, a very common neuron activity, can be distinctively depicted by SEP waveform. Figure 1.1 shows the sample SEP data obtained from a participant with chronic pain. We can clearly see the difference between the waveform representing asymptomatic side and symptomatic of the participant's body. With sufficient amount of SEP data, we could possibly quantify these differences using mathematical tools to determine a correlation between SEP waves and whether the individual is experiencing chronic pain.

The way we devised is to design a computer program to help us process the SEP data. We use **decision trees** as the core algorithm of our program. The idea is that if we train the program with large amount of data that quantify the difference between asymptomatic and symptomatic SEP data and corresponding asymptomatic or symptomatic information, the program can automatically predict whether an individual is in pain or not based only on SEP data. Data for training the decision tree is currently being collected and is not yet available.

II. Methods

With **decision tree algorithm**, we would be able to classify the **SEP waveform** based on its **mathematical features**.

Mathematical Features:

For now, we implemented 5 mathematical features that would potentially be part of the correlation.

(1) Difference of Maximum Amplitude (millivolts)

Variable name used in the program:
- Amplitude
- AmplitudeD

Definitions:
- **Amplitude**: voltage difference between the trough and the peak of the signal.
- **AmplitudeD**: absolute value of the difference between the maximum amplitude of two signals

Formulas:

Amplitude = (Maximum Voltage) – (Minimum Voltage)

AmplitudeD = $|(Amplitude_{sp}) - (Amplitude_{asp})|$

(2) Difference of Latency (milliseconds)

Variable name used in the program:
- Latency
- LatencyD

Definitions:
- **Latency**: absolute value of the time difference between where the trough occurs and where the peak occurs
- **LatencyD**: absolute value of the difference between the latency values of two signals.

Formulas:

Latency = |(time where the peak occurs) – (time where the trough occurs)|

LatencyD = $|Latency_{sp} - Latency_{asp}|$

(3) Difference of Base Line (millivolts)

Variable name used in the program:

- BaseLine
- BaseLineD

Definitions:

- **BaseLine**: a value in millivolts obtained by averaging the last 100 milliseconds of the signal.
- **BaseLineD**: absolute value of the difference between the base line value of two signals

Formulas:

BaseLine = $\sum_{i=n-100}^{n} i$, where i is the value of i^{th} data point on a single waveform (in millivolts), n is the number of data points.

BaseLineD = $|BaseLine_{sp} – BaseLine_{asp}|$

(4) Difference of Scaled Area (millivolts)

Variable name used in the program:

- KArea

Definitions:

- **KArea**: the scaled area between two curves.

Formulas:

Linespace: L(x) = 0.01*x (0 <= x <= 200)

Area: A = $\sum_{i=0}^{n} S(i)$ S(i) is the i^{th} data point on one waveform curve.

KArea(x)= $\min(|A_{sp}*L(x) – A_{asp}|)$

*x is the value where KArea minimum occurs.

(5) Difference of Shifted Area (millivolts)

Variable name used in the program:

- HArea

Definitions:

- **HArea**: the area of between shifted curves.

Formulas:

Linespace: L(x) = 0.01*x (0 <= x <= 200)

Area: A = $\sum_{i=0}^{n} S(i)$ S(i) is the ith data point on one waveform curve.

HArea(x)= min(|A_{sp} - L(x) - A_{asp}|)

).

*x is the value where HArea minimum occurs.

Sample computed data:

AmplitudeD	BaseLineD	LatencyD	KArea	HArea	Prediction
0.275486	1.754764	23	132.1711	187.922431	ASP
0.838722	4.542731	42	166.369264	515.778682	ASP
0.091652	0.936752	3	24.155364	16.374453	SP
1.966599	2.881989	103	29.087663	340.950952	SP

III. How to use the program

Preparation before running the program:

1. The data analysis program is written in Java™, so the computer needs to have Java Virtual Machine (Java VM) before running the program.

2. Prepare the SEP data file:

- File format:

All the SEP data need to be saved as .CSV file format. You can easily convert .xlsx or .xls file to CSV file by using "Save as" in Excel.

- Naming data files:

Naming files is very critical process in preparing files. Following is the naming system for the data files:

The file names used for the pain study will start off with two letters that identify the experimental condition ("c" for control or "e" for experimental) and whether pain is present or not ("s" for pain, "a" and "p" for no pain; the "p" will only be used for the control group that has no pain, so the control group will have "a" and "p" as designators that no pain is present and the experimental group will have "s" and "a" as designators for pain or no pain).

These letters will be followed by a number designating location of test (01 = ankles, 02 = calf, 03 = shins, 04 = knees, 05 = leg/quadriceps, 06 = leg/hamstrings, 07 = wrists, 08 = forearms, 09 = elbows, 10 = arm/biceps, 11 = arm/triceps, and 12 = shoulders; additional numbers may be used).

Number will be followed by a dash then a three digit participant number (the first participant is 001, the second 002, etc.).

EXAMPLES

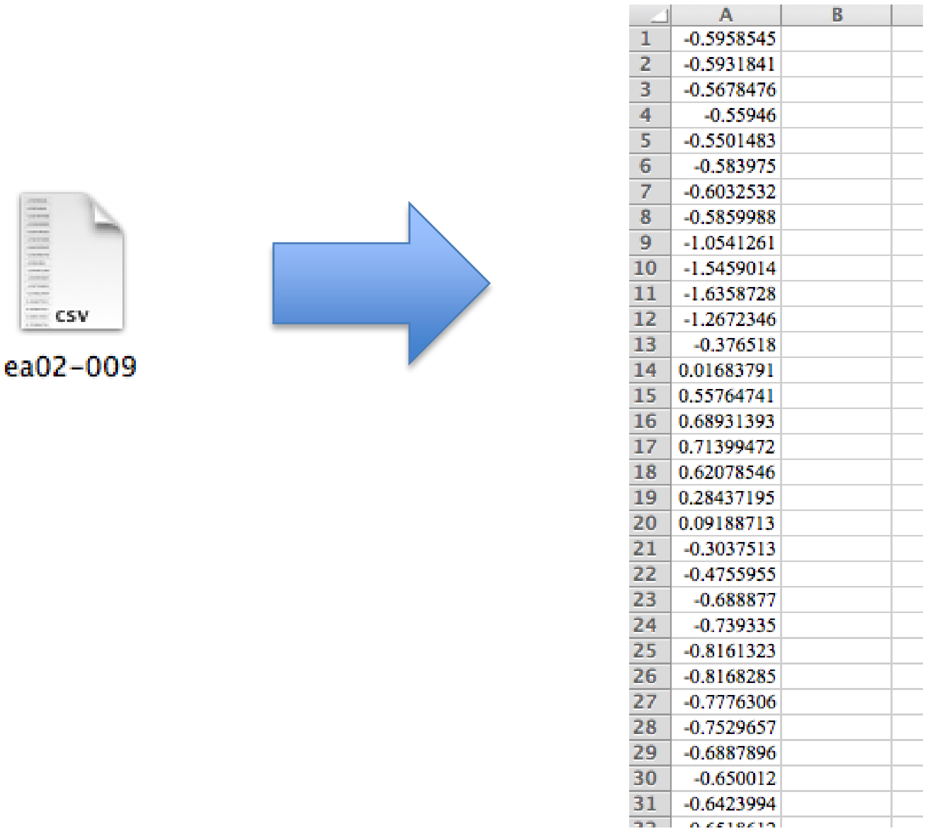
1. For an injured athlete with a strained right calf muscle that happens to be the fourth participant in the study.
File name for the strained right calf: es02-004
File name for the left calf: ea02-004

2. For a non-injured student from the psychology participant pool that is the 16 participant in the study and has the shoulders tested.
File name for right shoulder: ca12-016
File name for the left shoulder: cp12-016

- Contents of data files:

The datasheet typically consists of one column of data. The unit of each data point should be in millivolts. You do not need to specify the time information in the file.

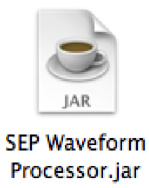
Sample input data:



	A	B
1	-0.5958545	
2	-0.5931841	
3	-0.5678476	
4	-0.55946	
5	-0.5501483	
6	-0.583975	
7	-0.6032532	
8	-0.5859988	
9	-1.0541261	
10	-1.5459014	
11	-1.6358728	
12	-1.2672346	
13	-0.376518	
14	0.01683791	
15	0.55764741	
16	0.68931393	
17	0.71399472	
18	0.62078546	
19	0.28437195	
20	0.09188713	
21	-0.3037513	
22	-0.4755955	
23	-0.688877	
24	-0.739335	
25	-0.8161323	
26	-0.8168285	
27	-0.7776306	
28	-0.7529657	
29	-0.6887896	
30	-0.650012	
31	-0.6423994	

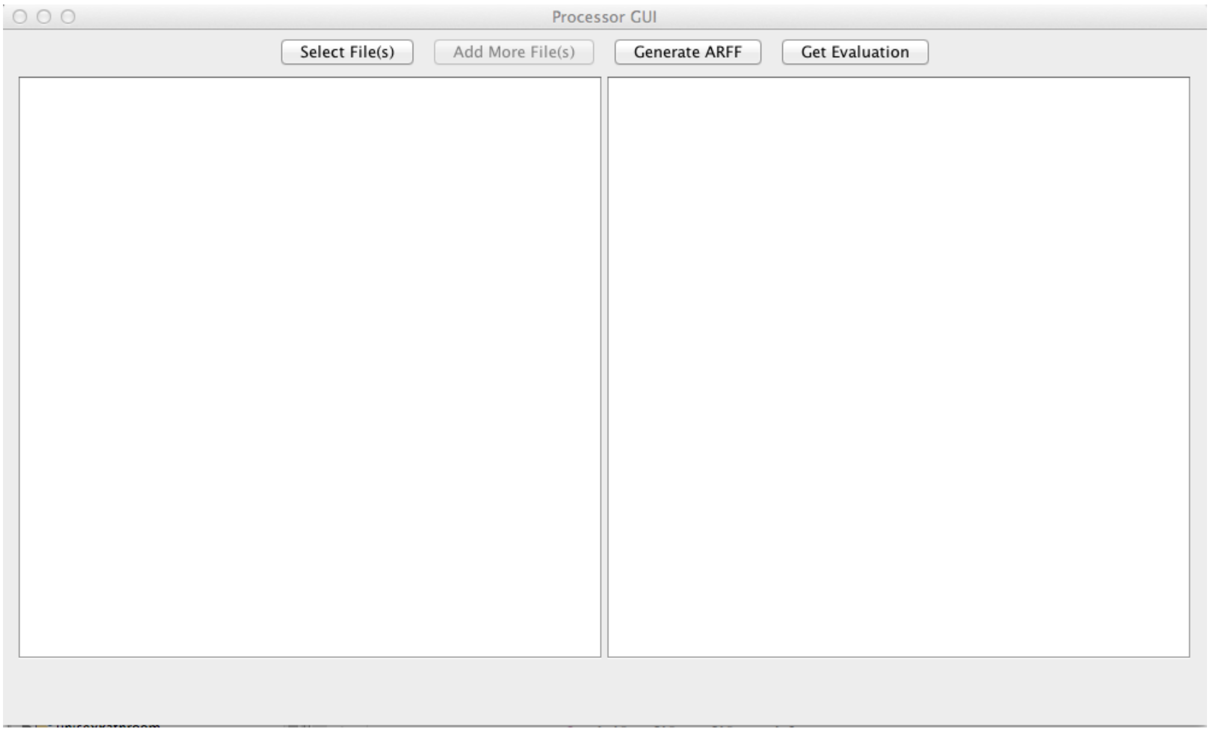
Using the program:

- Launch the program by double clicking the icon:



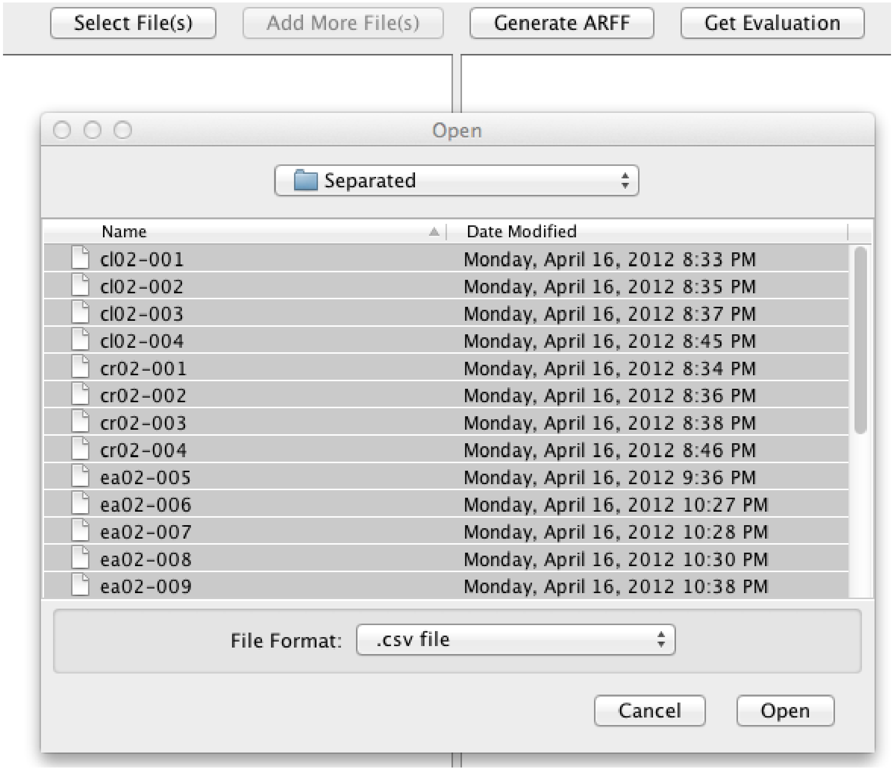
-

- The user interface of the program.

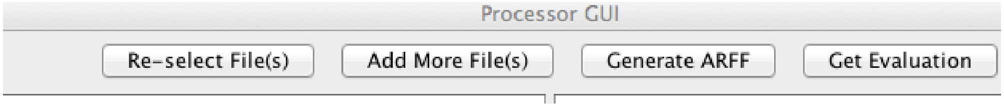


- **Import files**

Click "Select File(s)" button. A window will show up, allowing you select .CSV files you prepared for input.



After selecting files, the buttons on the top of the window will change to:



You can abandon the files you just selected and re-select files by clicking "Re-select Files(s)"

OR-

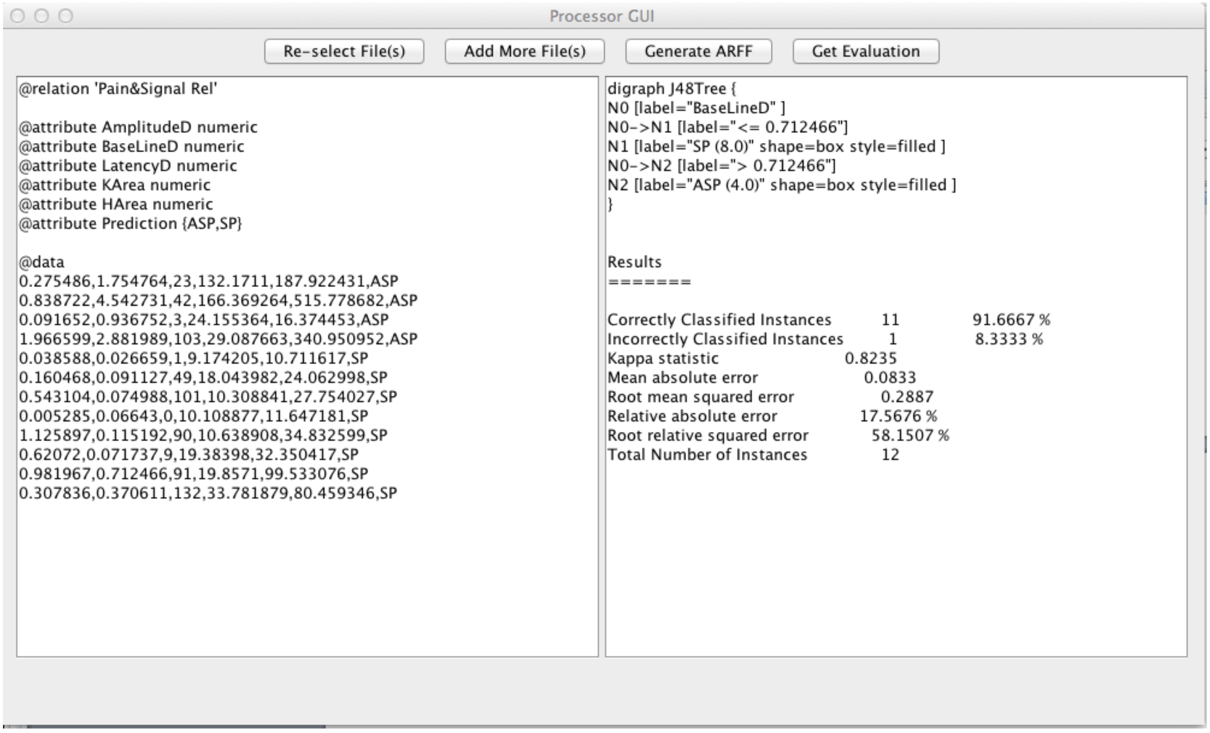
Add more files to the files you have selected by clicking "Add More File(s)".

- Process files

After successfully selecting all the files you want the program to process, click the "Generate ARFF" button.

The program will generate the input data for Weka data mining program on the left side of the window.

We have already implemented the entire Weka data mining library in our program. You can simply click "Get Evaluation" to view the data evaluation output on the right side of the window.

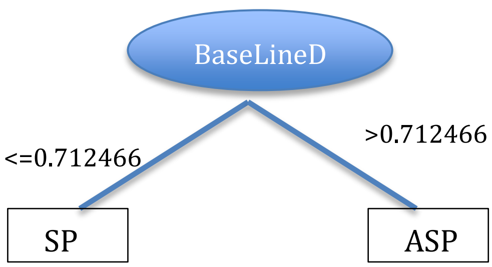


Interpret the output decision tree and its evaluation:

Decision Tree:

```
digraph J48Tree {
N0 [label="BaseLineD" ]
N0->N1 [label="<= 0.712466"]
N1 [label="SP (8.0)" shape=box
style=filled ]
N0->N2 [label="> 0.712466"]
N2 [label="ASP (4.0)" shape=box
style=filled ]
}
```

This is a sample J48 decision tree. It has two leaves "SP" and "ASP".



Textual explanation of the tree would be "In each instances, if the value of BaseLineD attribute is greater than or equal to 0.712466, then the instance corresponds to SP (Symptomatic) prediction attribute. If BaseLineD is smaller than 0.712466, then the instance corresponds to ASP (Asymptomatic) prediction value”.

Evaluation Results:

Results
=======

Correctly Classified Instances	11	91.6667 %
Incorrectly Classified Instances	1	8.3333 %
Kappa statistic	0.8235	
Mean absolute error	0.0833	
Root mean squared error	0.2887	
Relative absolute error	17.5676 %	
Root relative squared error	58.1507 %	
Total Number of Instances	12	

This result means, “Using this decision tree, 91.6667% of instances (11/12) can be correctly classified.”