PLOS ONE Surface EMG Onset Algorithms

Matthew S. Tenan

March 10th, 2017

This file will show to how to extract the data and run some of the analysis from our manuscript

We encourage authors to use our data with their own algorithms!

This helps advance the field more rapidly. You can access the data using the code below. The data is in R's list format and has already been bandpass filtered 10-1000 Hz per the details in our manuscript. Next, then we'll do a simple plot of one of the trials.

This code reads all of our intramuscular EMG data into your local working directory:

URL1 <- "http://github.com/TenanATC/EMG/blob/master/sEMG_PLOS.rds?raw=true"

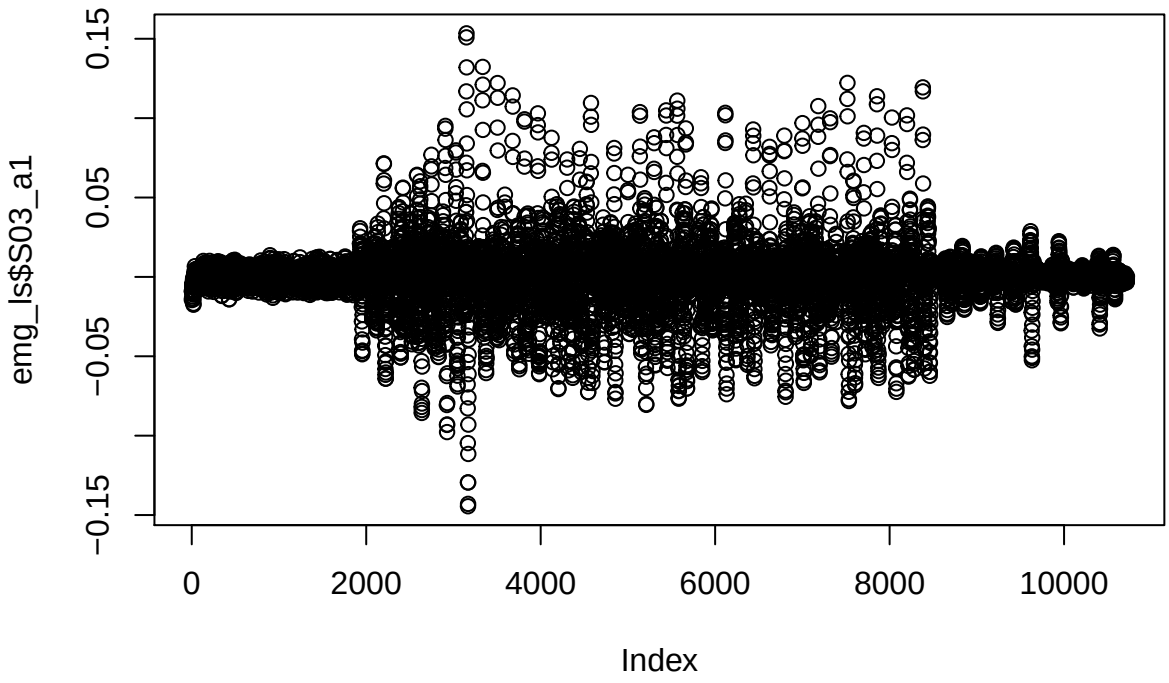
download.file(URL1, destfile = "~/fwEMG_JAB.rds", mode = 'wb', quiet = T)

URL2 <- "http://raw.githubusercontent.com/TenanATC/EMG/master/VisualOnset_PLOS.csv"

download.file(URL2, destfile = "~/VisualOnset_JAB.csv", quiet = T)

Next, then we'll open it in the workspace do a simple plot of one of the trials.

```
emg_ls <- readRDS('sEMG_PLOS.rds')
vis <- read.csv('VisualOnset_PLOS.csv')
plot(emg_ls$S03_a1)
```



Note that all data, including visual onset timing, is in sample number. It is not yet in seconds, but this is an easy conversion since we know the sampling rate was 2048 Hz.

Next, we'll load the packages we'll use for the rest of these examples. This is done using the library() function. If you have never installed these packages before, you may need to use this line of code prior to the library functions: install.packages(c('ggplot2', 'dplyr', 'bcp', 'xts', 'signal', 'seewave'))

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(bcp)
```

```
## Loading required package: grid
```

```
library(xts)
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
library(signal)
```

```
##
## Attaching package: 'signal'

## The following object is masked from 'package:dplyr':
##
##     filter

## The following objects are masked from 'package:stats':
##
##     filter, poly
```

```
library(seewave)
```

```
##
## Attaching package: 'seewave'

## The following object is masked from 'package:signal':
##
##     unwrap
```

Now on to the fun stuff

Now we'll do an example of the Bayesian Changepoint analysis algorithm which performed well in our systematic analysis (rectified EMG, p0 =0 and posterior probability threshold onset at 95%) and show the differences in onset time.

```
#extract a trial and rectify EMG
sEMGrect <- abs(emg_ls$S03_a1)
#run BCP algorithm and extract the first point where posterior probability of onset is 95%
result_bcp <- bcp(sEMGrect, p0 = 0)
onset_bcp <- which(result_bcp$posterior.prob >= 0.95)[1]
#Now we extract the visually determined EMG onset
onset_visual <- vis$value[match('S03_a1', vis$sbj)]

print(onset_bcp)
```

```
## [1] 1958
```

```
print(onset_visual)
```

```
## [1] 1929.167
```

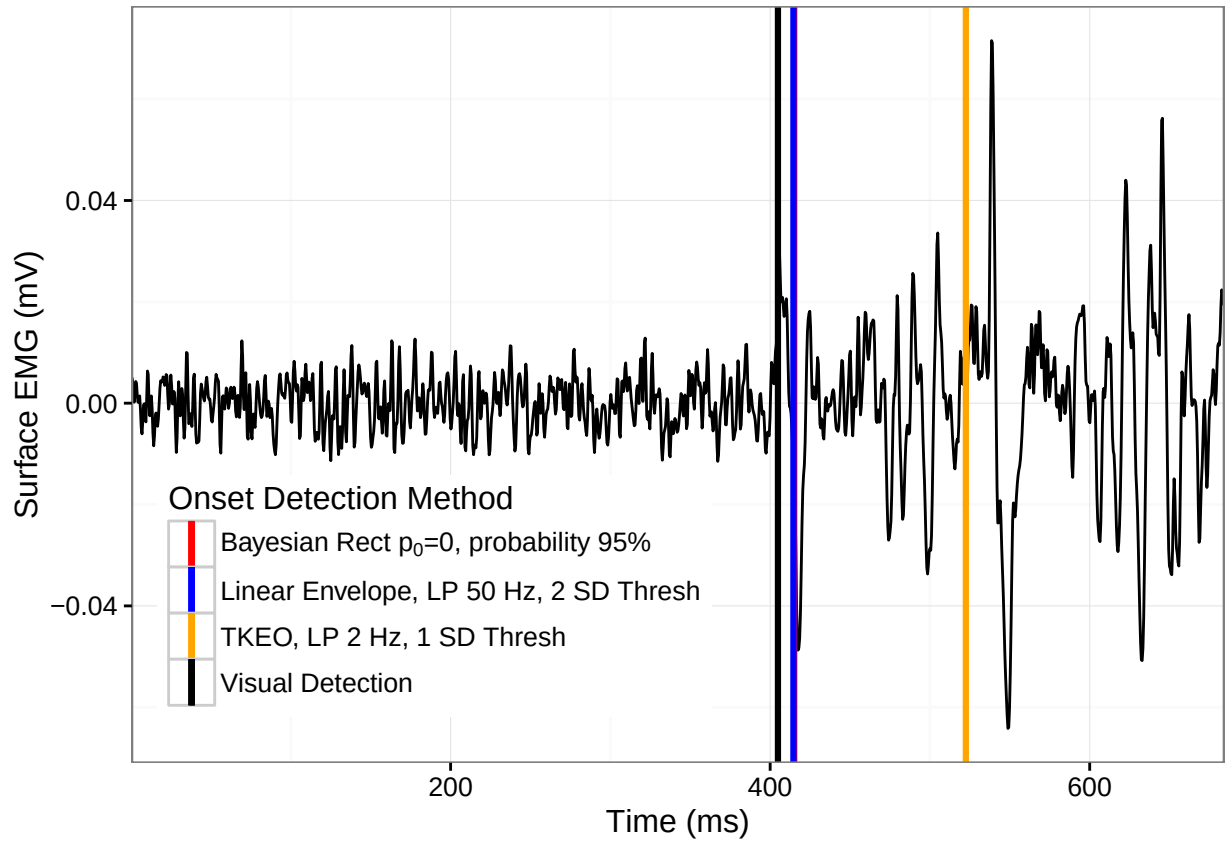
So the difference between the two measures for this trial is 40.833, corresponding to 0.0199 seconds.

Let's plot this data in comparision with other algorithm types we tested. We'll actually be re-creating the Figure 5 from our manuscript.

Comments are added, but hopefully future versions of this document will explain the analyses in a more stepwise fashion.

```
#Build Function to do linear envelopes and get detection point
# x = time series
# a = low-pass filter (in Hz)
# N = Nyquist Frequency (half sampling rate (samples/sec))
# d = detection threshold in standard deviations

lin_env<- function(x, a, N, d){
  lowrate <- a/N #lowpass filter rate for butter

  envelope <- butter(2, lowrate, type='low')
  dat_env <- filtfilt(envelope, x)

  threshold <- sd(dat_env, na.rm=T)*d

  idx_on <- which(dat_env > threshold)[1]
  idx_on
}



##Figure 1 analysis
#extract a trial and rectify EMG
sEMG1 <- emg_ls$S03_a1
sEMGrect1 <- abs(sEMG1)


#BCP and Visual
result_bcp <- bcp(sEMGrect1, p0 = 0)
onset_bcp1 <- ((which(result_bcp$posterior.prob >= 0.95)[1])-1100)/2.048
#Now we extract the visually determined EMG onset
onset_visual1 <- ((vis$value[match('S03_a1', vis$sbj)])-1100)/2.048

#Now check best Linear Envelope and TKEO methods
#Teager-Kaiser Energy Operator according to Slonik et al. 2008
sEMG_filt_TKEO <- TKEO(sEMG1, 2048, plot=F)
#Replace NAs from TKEO with zeros
sEMG_filt_TKEO[is.na(sEMG_filt_TKEO)] <-0
#Full-Wave Rectify
sEMG_filtTKEO_abs <- abs(sEMG_filt_TKEO[,2])
sEMG_filt_abs <- sEMGrect1

#now get onsets for linear envelope & TKEO!
onset_linearenv1 <- (lin_env(sEMG_filt_abs,a=50, N=1024, d=2)-1100)/2.048
onset_tkeo1 <- (lin_env(sEMG_filtTKEO_abs, a=2, N=1024, d=1)-1100)/2.048


##NOW CREATE THE FIGURES
#Original data needs to be in dataframes we'll convert to milliseconds
#Cut data to make it clearer for publication
```

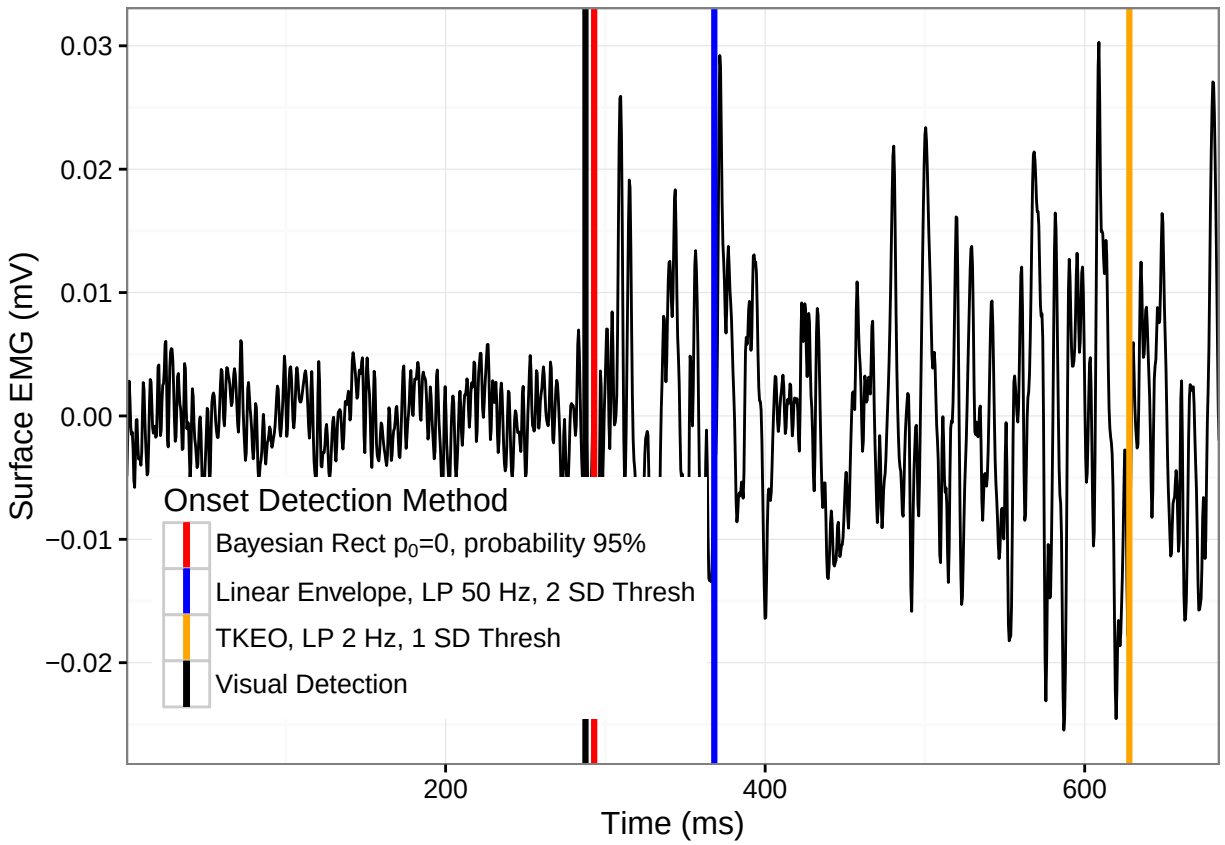
```
sEMG1_cut <- sEMG1[1100:2500]
sample1 <- seq.int(from =1, to = length(sEMG1_cut))/2.048
emg_plot1 <- as.data.frame(cbind(sEMG1_cut, sample1))
colnames(emg_plot1) <- c('sEMG', 'Time')

ggplot(emg_plot1, aes(x=Time, y=sEMG)) +
            geom_line() +
            theme_bw() +
            geom_vline(aes(xintercept = onset_bcp1, color = "bcp"), size= 1.1) +
            geom_vline(aes(xintercept = onset_visual1, color = 'vis'), size= 1.1) +
            geom_vline(aes(xintercept = onset_linearenv1, color = 'linenv'), size= 1.1) +
            geom_vline(aes(xintercept = onset_tkeo1, color = 'tkeodat'), size= 1.1) +
            scale_x_continuous(expand = c(0,0)) +
            scale_color_manual(name = "Onset Detection Method", values = c(bcp = "red", vis = "black",
                               labels = c(expression('Bayesian Rect p'[0]*'=0, probability 95%'), 'Linea
                                          'TKEO, LP 2 Hz, 1 SD Thresh', "Visual Detection")) +
            theme(legend.justification = c(0.05,0.05), legend.position = c(0.03,0.05), legend.text.alig
            xlab('Time (ms)') + ylab('Surface EMG (mV)')
```



Great! Now let's recreate our Figure 6 from the manuscript (minus some formatting)...

```
##Figure 2 analysis
#extract a trial and rectify EMG
sEMG2 <- emg_ls$S16_l3
sEMGrect2 <- abs(sEMG2)
```

```
#BCP and Visual
result_bcp <- bcp(sEMGrect2, p0 = 0)
onset_bcp2 <- ((which(result_bcp$posterior.prob >= 0.95)[1])-1100)/2.048
#Now we extract the visually determined EMG onset
onset_visual2 <- ((vis$value[match('S16_13', vis$sbj)])-1100)/2.048

#Now check best Linear Envelope and TKEO methods
#Teager-Kaiser Energy Operator according to Slonik et al. 2008
sEMG_filt_TKEO <- TKEO(sEMG2, 2048, plot=F)
#Replace NAs from TKEO with zeros
sEMG_filt_TKEO[is.na(sEMG_filt_TKEO)] <-0
#Full-Wave Rectify
sEMG_filtTKEO_abs <- abs(sEMG_filt_TKEO[,2])
sEMG_filt_abs <- sEMGrect2

#now get onsets for linear envelope & TKEO!
onset_linearenv2 <- ((lin_env(sEMG_filt_abs,a=50, N=1024, d=2))-1100)/2.048
onset_tkeo2 <- ((lin_env(sEMG_filtTKEO_abs, a=2, N=1024, d=1))-1100)/2.048


##NOW CREATE THE FIGURES
#Original data needs to be in dataframes we'll convert to milliseconds
#Cut data to make it clearer for publication
sEMG2_cut <- sEMG2[1100:2500]
sample2 <- seq.int(from =1, to = length(sEMG2_cut))/2.048
emg_plot2 <- as.data.frame(cbind(sEMG2_cut, sample2))
colnames(emg_plot2) <- c('sEMG', 'Time')

ggplot(emg_plot2, aes(x=Time, y=sEMG)) +
            geom_line() +
            theme_bw() +
            geom_vline(aes(xintercept = onset_bcp2, color = "bcp"), size= 1.1) +
            geom_vline(aes(xintercept = onset_visual2, color = 'vis'), size= 1.1) +
            geom_vline(aes(xintercept = onset_linearenv2, color = 'linenv'), size= 1.1) +
            geom_vline(aes(xintercept = onset_tkeo2, color = 'tkeodat'), size= 1.1) +
            scale_x_continuous(expand = c(0,0)) +
            scale_color_manual(name = "Onset Detection Method", values = c(bcp = "red", vis = "black",
                    labels = c(expression('Bayesian Rect p'[0]*'=0, probability 95%'), 'Linear Envelop
                               'TKEO, LP 2 Hz, 1 SD Thresh', "Visual Detection")) +
            theme(legend.justification = c(0.05,0.05), legend.position = c(0.03,0.05), legend.text.alig
            xlab('Time (ms)') + ylab('Surface EMG (mV)')
```



We're still in the process of making this documentation more user-friendly, please be patient.

Hopefully you are enticed to play around more with these algorithms and our publically available data.