Introduction to Package PML in Circadian Analysis Using Actigraphy

Xinyue Li

2019-08-20

Introduction

This vignette gives examples on how to apply penalized multi-band learning to accelerometer data to analyze circadian rhythms. Specifically, it uses L1/L2 combined penalty to select important harmonics sequentially. To conduct statistical tests to pick significant harmonics, functions for Fisher's harmonic test are also included. In addition, one can use functions on trelliscope visualization to plot individual mean activities compared to the population mean activity, and individual day activities compared to the individual mean activity and the population mean activity.

Penalized Multi-band Learning

We use a synthetic dataset to implement the method. 'lis3' is a data list consisting of three matrices, each giving activity data for one individual. Each column of the matrix is one-day observation, and here the physical activity (PA) is measured every one minute, so the matrix is 1440 by 'nday'. 'lis3' is also a named list, the name of which is the individual ID.

To convert the data to the required format as the input of the function for penalized multi-band learning, we use function 'form':

```
suppressMessages(library(PML))
data(lis3)
pa3 <- form(lis3)
pa3
```

```
## # A tibble: 13 x 3
##       ID ID_Nday activity
##    <dbl>   <int> <list>
##  1     1       1 <dbl [1,440]>
##  2     1       2 <dbl [1,440]>
##  3     1       3 <dbl [1,440]>
##  4     1       4 <dbl [1,440]>
##  5     3       1 <dbl [1,440]>
##  6     3       2 <dbl [1,440]>
##  7     3       3 <dbl [1,440]>
##  8    15       1 <dbl [1,440]>
##  9    15       2 <dbl [1,440]>
## 10    15       3 <dbl [1,440]>
## 11    15       4 <dbl [1,440]>
## 12    15       5 <dbl [1,440]>
## 13    15       6 <dbl [1,440]>
```

The resulting dataset 'pa3' is in tbl_df format. It has 13 observations for 3 individuals, and the variables are "ID", "ID_Nday" (the ith day observation for an individual), and activity. The activity variable is an embedded list with each element consisting of a vector of one-day observation.

Further, 'pa3' serves as part of the input for the function `bandSelect`. It uses combined L1/L2 penalty to select important harmonics sequentially based on Fast Fourier Transform (FFT) results. X_k is the vector of the kth harmonic FFT results. λ is the tuning parameter for penalty, and it ranges from 0 (no penalty) to $2\max(||X_k||^2)$. θ_i's are the estimated coefficents for different harmonics given a specified penalty.

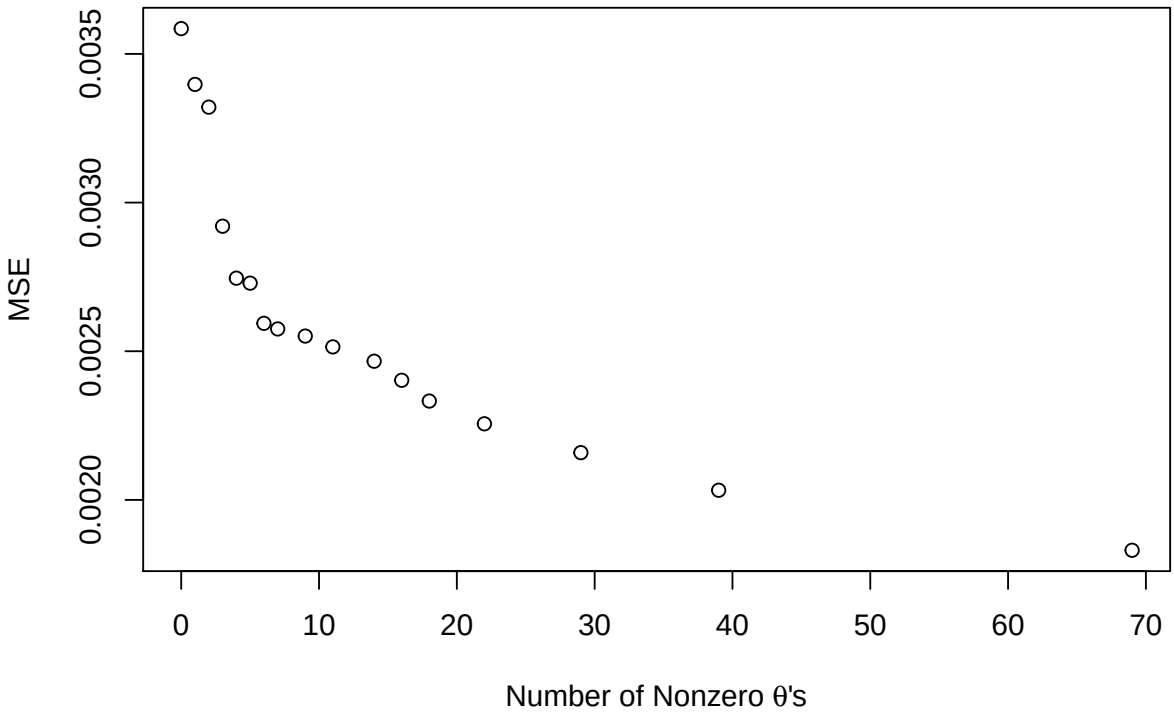
`alpha`: controls the balance between L1 and L2 penalties. Because here we want to pick a few important harmonics, we set α=1 to use L1 penalty only.

`Nlength`: In the analysis we need to use individual observations of equal lengths (equal number of day observations), because different lengths of inputs into FFT will result in different sets of harmonics and expect different magnitudes of signals (i.e. for certain harmonics, the signals are expected to be stronger in observations of longer time). Therefore, we use the 'Nlength' argument to specify the length of observation for an individual to be entered into the analysis. Here we choose 3 days, namely 1440*3 observations per individual. We then show the top 3 important harmonics in the population ('Ntop=3'). Because the sample data only have 3 individuals, we do not perform 5-fold cross-validation ('cross=FALSE').

`plot=TRUE`: We want to see the effect of the first a few harmonics compared to other harmonics and all harmonics, so we plot Mean Squared Error (MSE) against the number of nonzero θ's as λ decreases. Note that it only plots the points whether the number of nonzero θ's increases; between plotted points, the values of the already nonzero θ's increase.

```
data(pa3)
re <- bandSelect(df=pa3,Nlength=1440*3,Nlambda=100,alpha=1,Ntop=3,
                 cross=FALSE,Ncross=NULL,plot=TRUE)
```

```
## [1] "With a threshold of  4320  observations per individual, "
## [1] "3 individuals are used for penalized multi-band learning."
```



The top three frequencies are 1440, 720, 480, corresponding to 1-day, 1/2-day, and 1/3-day periodicities. The convolution of the three frequencies can give two-peak day activity patterns, with one activity peak in the morning, one peak in the afternoon and evening, and interims of one short nap in the afternoon and one long sleep at night. The output 'xscore' and 'xprop' gives FFT results (signal magnitudes and proportions standardized by individuals respectively) in 3 by 2159 matrices, with row standing for 3 individuals and column standing for 2159 frequencies. We may want to see the top frequencies with mean signal proportions:

```
freq <- data.frame(Frequency=re$freq,Proportion=colMeans(re$xprop))
rownames(freq) <- NULL
head(freq[order(freq$Proportion,decreasing = TRUE),],5)
```

Frequency	Proportion
1440	0.0107955
720	0.0086717
480	0.0082258
360	0.0059348
2160	0.0045538

As is shown in the table, the top three proportions are quite dominant.

Fisher's Harmonic Test

To conduct statistical tests on FFT results, we can apply the tests developed by R.A. Fisher (1929). Because we are testing multiple frequencies simultaneously, it is better to incorporate procedures to correct for it. Here we use the Bonferroni correction. The function `test.harmonic` takes the observation as the input and gives the significant frequencies (the first insignificant frequency will also be given).

```
## no significant frequencies
ob <- do.call("c",pa3$activity[1:4])
re <- test.harmonic(ob,p=0.05/(length(ob)-1)/2)
re$sig;head(re$fft)
```

```
##   frequency    prop (g)      p-value  p-threshold
## 1      1440 0.006679843 1.208083e-05 4.341031e-06
```

```
##      freq        prop
## [1,] 1440 0.006679843
## [2,]  720 0.006534117
## [3,]  360 0.005587945
## [4,]  480 0.004471479
## [5,]   90 0.003526596
## [6,]  288 0.003511198
```

For the first individual, no significant frequency is detected.

```
## 3 significant frequencies
ob2 <- do.call("c",pa3$activity[11:13])
re2 <- test.harmonic(ob2,p=0.05/(length(ob2)-1)/2)
re2$sig;head(re2$fft)
```

```
##   frequency    prop (g)      p-value  p-threshold
## 1      1440 0.009868019 1.096319e-06 5.788377e-06
## 2       720 0.009248061 7.417411e-12 5.788377e-06
## 3       480 0.008836504 1.106203e-16 5.788377e-06
## 4      2160 0.004189404 1.103045e-04 5.788377e-06
```

```
##           freq        prop
## [1,] 1440.0000 0.009868019
## [2,]  720.0000 0.009248061
## [3,]  480.0000 0.008836504
## [4,] 2160.0000 0.004189404
## [5,]  617.1429 0.004080741
## [6,]  432.0000 0.003522150
```

For the third individual, 3 significant frequencies are detected.

To conduct tests on the population level, we can use the mean FFT proportions from previous results (dataset freq) as input:

```
re3 <- test.harmonic(freq,p=5.79e-06,fft=TRUE)
print(re3$sig,digits=3,row.names=FALSE)
```

```
##  frequency prop (g)  p-value p-threshold
##       1440  0.01080 1.45e-07    5.79e-06
##        720  0.00867 9.34e-11    5.79e-06
##        480  0.00823 6.40e-15    5.79e-06
##        360  0.00593 2.74e-11    5.79e-06
##       2160  0.00455 9.36e-08    5.79e-06
##        240  0.00429 4.51e-08    5.79e-06
##        288  0.00365 1.27e-05    5.79e-06
```

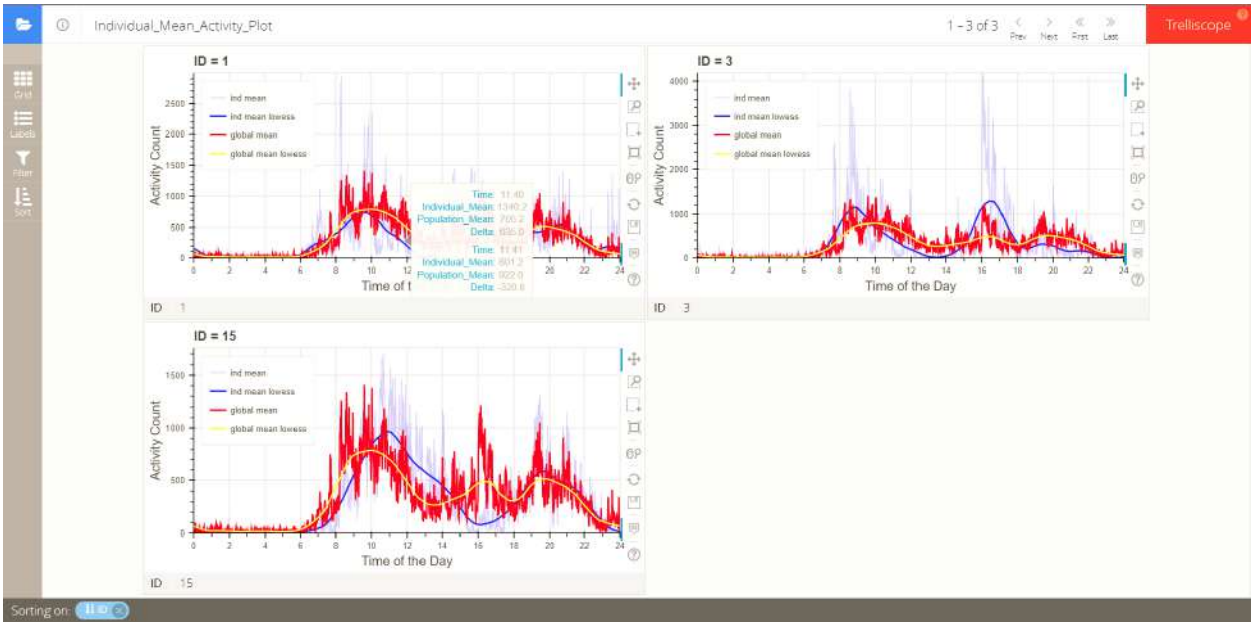
The harmonic tests pick out 6 significant frequencies.

Trelliscope Visualization

It is often of interest to see the actual daily activity patterns and compare different individuals with different characteristics. Trelliscope makes it easy to see the plots and interactively sort and filter them based on covariates.

Here we can use the function `tre` to generate trelliscope panels based on activity data. The required data format is the same as `lis3`, with each element as the activty matrix, the column of which is one-day activity observation. We can either see the individual mean plots or all the daily activity plots. If we are only interested in each individual compared to the population as a whole, we can set `plot.ind=TRUE`, and the default is TRUE.

```
#### directly display trelliscope panels, no dataset is returned
tre(lis3,plot.ind=TRUE,plot.tre=TRUE)
```



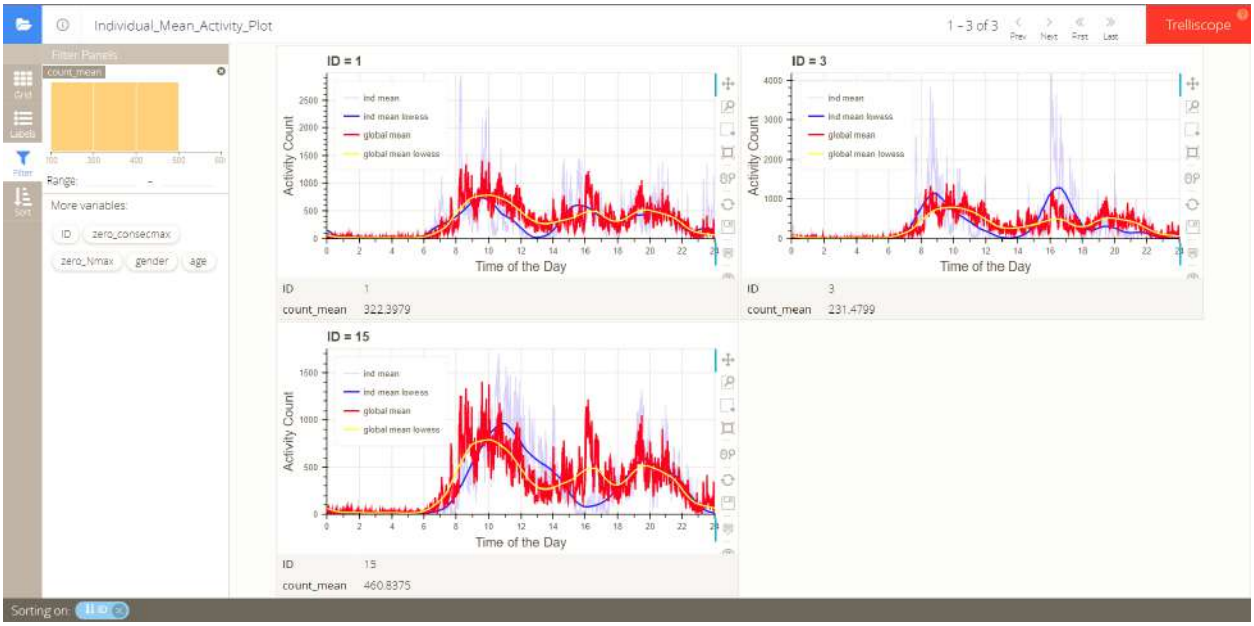
The variables that can be used to filter plots include ID, mean activity counts, total number of zero counts per day, and maximal number of consecutive zero counts per day.

We can always incorporate more variables for filtering, labelling, and sorting, using the `varlis` argument. The covariate dataset needs to have the first column as "ID", so that it can be merged to the activity data.

```
#### return a dataset with trelliscope panels
data(var3)
tre.ind <- suppressWarnings(tre(lis3,plot.ind=TRUE,varlis=var3))
```

```
## [1] "Generating trelliscope individual plots... It may take some time."
## [1] "Total time: 0 mins"
```

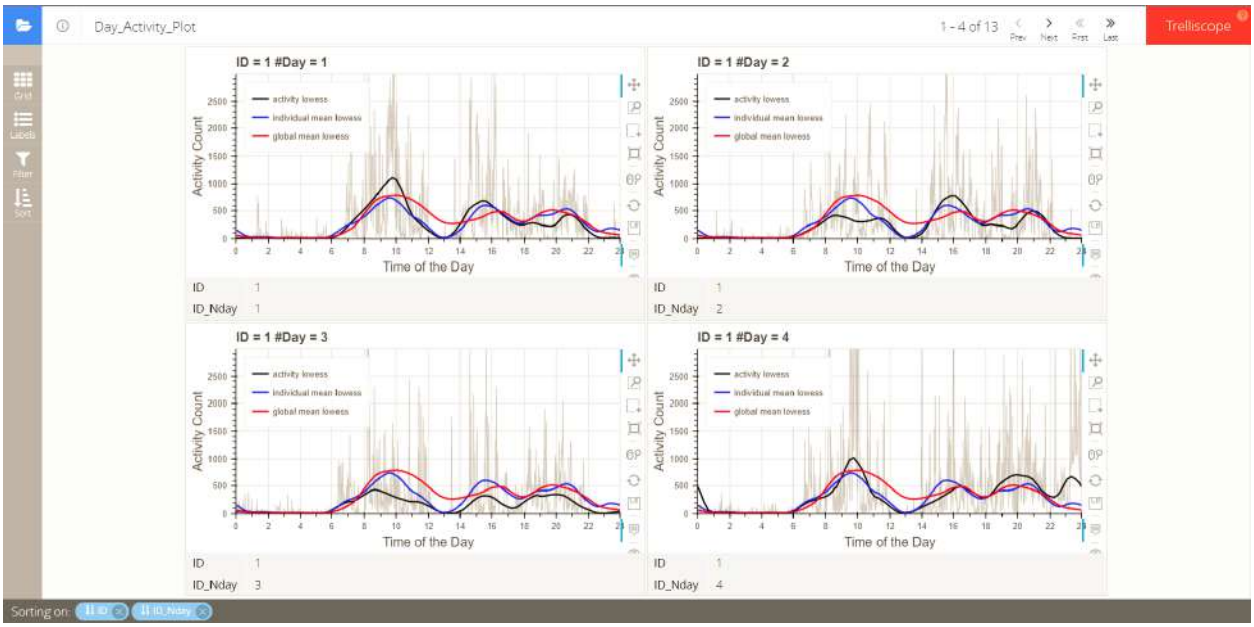
```
## prepare for visualization; no lists are allowed
tre.ind$activity_ind <- tre.ind$activity_all <- NULL
trelliscopejs::trelliscope(tre.ind,name = "Day Activity Plot", nrow = 2, ncol = 2,path=getwd())
```



If we merge covariates with the activity data using `varlis=var3`, in the trelliscope panels we can filter by the covariates: age and gender.

On the other hand, if we are interested in daily activity plots in addition to individual mean activity plots, we can set `plot.ind=FALSE`.

```
tre(lis3,varlis=var3,plot.ind=FALSE,plot.tre=TRUE)
```



It plots all the daily activity plots for each individual. The plots can also be filtered or sorted based on variables in the merged covariate dataset.