

# Time Series Clustering on Fluorescence Data from In Vivo Imaging

Shan Xu

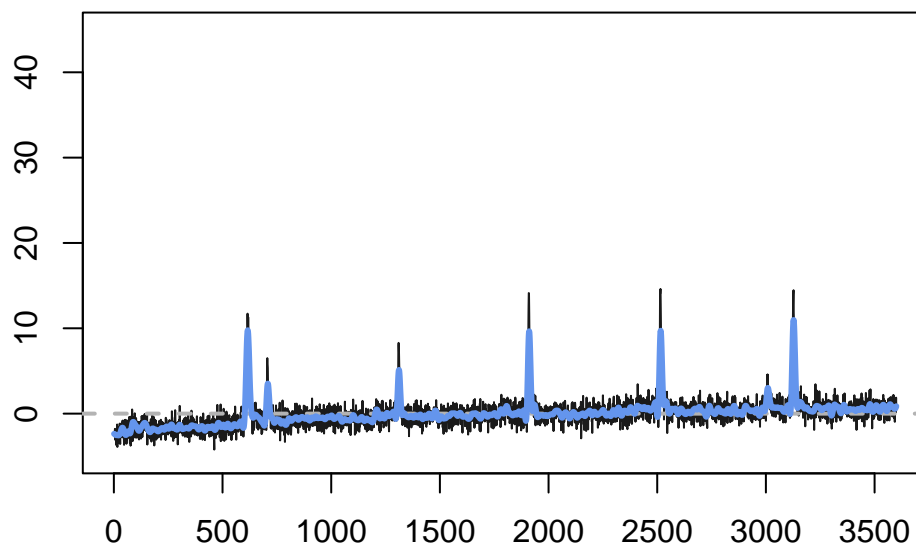
3/30/2022

```
Data = read.csv("AllDeidentified081921.csv")
Data = Data[1:4799, ]
Data = Data[-c(1)]
DataT = t(Data)
```

## Data preprocess

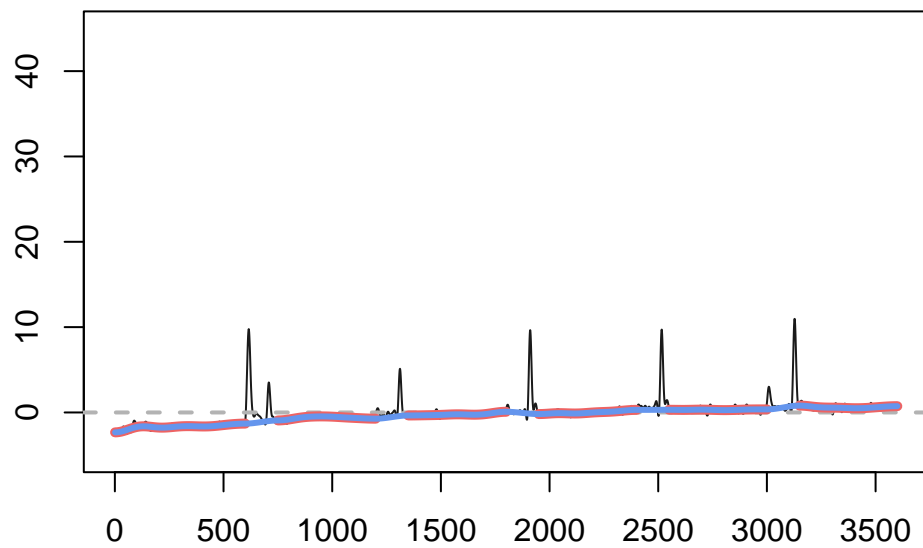
Remove high frequency noise

```
Data.LowP = apply(DataT, 1, function(x) pass.filt(x, W = 0.05,
  type = "low", method = "Butterworth"))
par(mar = c(2, 2, 1, 1))
plot(Data$X0682.0373[1:3600], type = "l", col = "grey10", ylim = c(-5,
  45), xlab = "", ylab = "") #,axes=FALSE)
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)
lines(Data.LowP[1:3600, "X0682.0373"], type = "l", col = "cornflowerblue",
  lwd = 3)
```

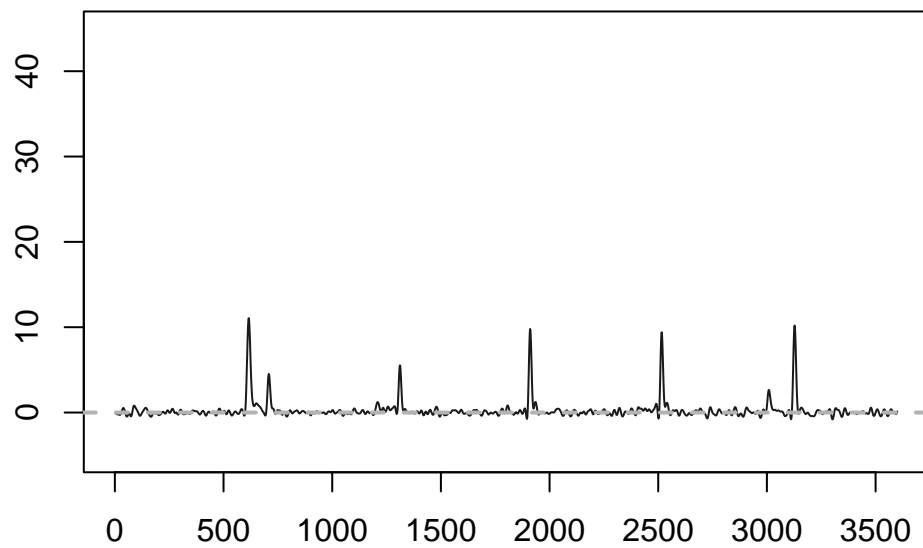


## Remove baseline drift

```
Data.NoInd <- Data.LowP
Data.trend <- Data.LowP
Data.NoInd[c(500:750, 1200:1350, 1800:1950, 2400:2550, 3000:3150,
            3600:3800, 4200:4400), ] = NaN
for (i in colnames(Data.LowP)) {
  for (j in 1:8) {
    if (j == 1) {
      Data.NoInd[1:(j * 600), i] = pass.filt(t(Data.LowP[1:(j *
        600), i]), W = 0.005, type = "low", method = "Butterworth")
    }
    if (j > 1 & j < 7) {
      Data.NoInd[((j - 1) * 600 + 150):(j * 600), i] = pass.filt(t(Data.LowP[((j -
        1) * 600 + 150):(j * 600), i]), W = 0.005, type = "low",
        method = "Butterworth")
    }
    if (j == 7) {
      Data.NoInd[((j - 1) * 600 + 200):(j * 600), i] = pass.filt(t(Data.LowP[((j -
        1) * 600 + 200):(j * 600), i]), W = 0.005, type = "low",
        method = "Butterworth")
    }
    if (j == 8) {
      Data.NoInd[((j - 1) * 600 + 200):(nrow(Data)), i] = pass.filt(t(Data.LowP[((j -
        1) * 600 + 200):(nrow(Data)), i]), W = 0.005,
        type = "low", method = "Butterworth")
    }
  }
  Data.trend[, i] = spline(Data.NoInd[, i], method = "natural",
    n = nrow(Data))$y
}
par(mar = c(2, 2, 1, 1))
plot(Data.LowP[1:3600, "X0682.0373"], type = "l", col = "grey10",
     ylim = c(-5, 45), xlab = "", ylab = "") #, axes=FALSE)
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)
lines(Data.NoInd[1:3600, "X0682.0373"], type = "l", col = "indianred2",
      lwd = 5)
lines(Data.trend[1:3600, "X0682.0373"], col = "cornflowerblue",
      lwd = 3)
```



```
Data.OMean <- Data.LowP - Data.trend
DataT.OMean = t(Data.OMean)
par(mar = c(2, 2, 1, 1))
plot(Data.OMean[1:3600, "X0682.0373"], type = "l", col = "grey10",
      ylim = c(-5, 45), xlab = "", ylab = "") #, axes=FALSE)
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)
```



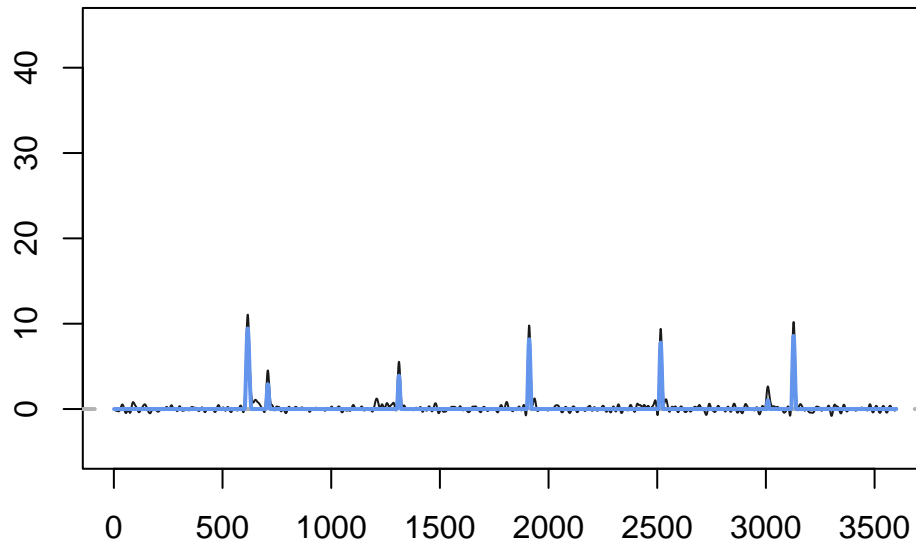
## Response detection

```
Data.Rsp <- Data.OMean
basetime <- 580
baseparts <- 1
baseinter <- basetime/baseparts
for (i in 1:ncol(Data.OMean)) {
```

```

FreFlted <- rep(0, nrow(Data.OMean))
maxsum <- 0
for (ibase in 1:baseparts) {
  tempmax <- max(Data.OMean[(ibase * baseinter - baseinter +
    1):(ibase * baseinter), i])
  maxsum <- maxsum + tempmax
}
QHigh <- maxsum * 2/baseparts
AmpFlted = FreFlted
for (j in 1:nrow(Data.OMean)) {
  if (Data.OMean[j, i] > QHigh) {
    AmpFlted[j] = Data.OMean[j, i] - QHigh
  }
}
Data.Rsp[, i] <- AmpFlted
}
par(mar = c(2, 2, 1, 1))
plot(Data.OMean[1:3600, "X0682.0373"], type = "l", col = "grey10",
  ylim = c(-5, 45), xlab = "", ylab = "") #, axes=FALSE)
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)
lines(Data.Rsp[1:3600, "X0682.0373"], col = "cornflowerblue",
  lwd = 2)

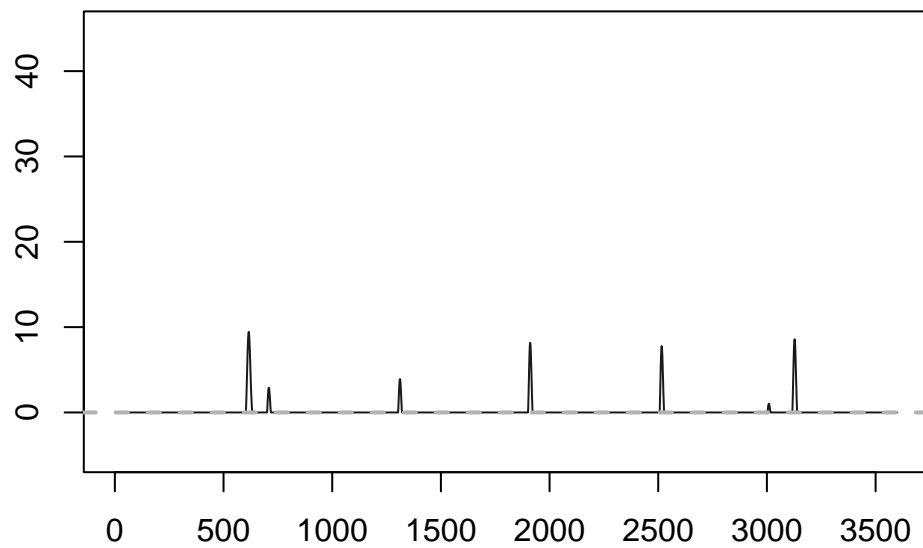
```



```

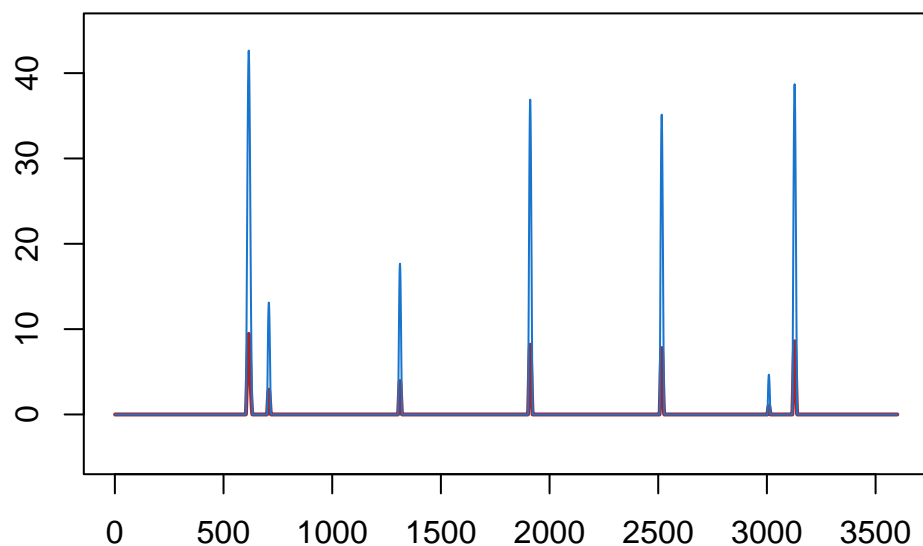
par(mar = c(2, 2, 1, 1))
plot(Data.Rsp[1:3600, "X0682.0373"], type = "l", col = "grey10",
  ylim = c(-5, 45), xlab = "", ylab = "") #, axes=FALSE)
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)

```



## Normalization

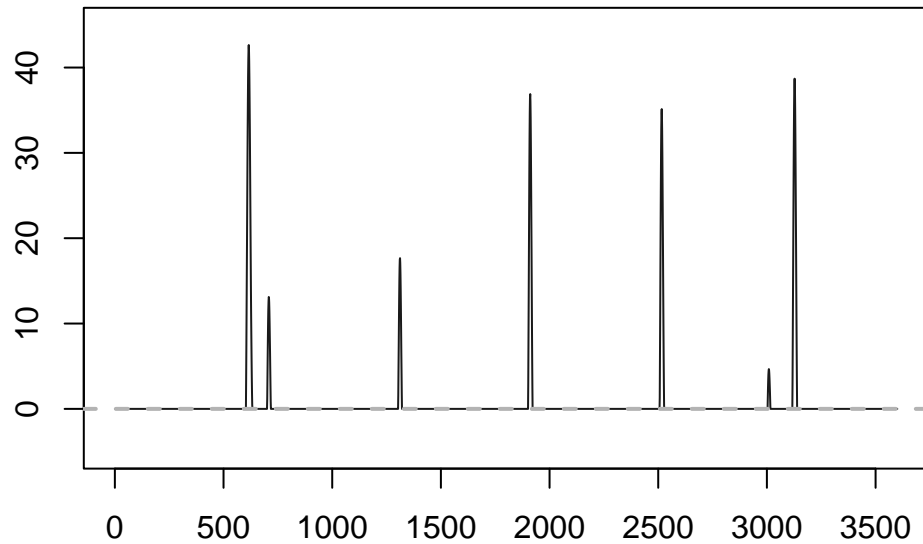
```
DataN <- Data.Rsp[1:3600, ] # only responses to indentation stimuli (brushing excluded)
MaxForAll <- max(DataN)
for (i in colnames(DataN)) {
  DataN[, i] = Data.Rsp[1:3600, i] * max(Data.Rsp[1:3600, ])/max(Data.Rsp[1:3600,
    i])
}
DataN[is.na(DataN)] <- 0
DataNT = t(DataN)
par(mar = c(2, 2, 1, 1))
plot(Data.Rsp[1:3600, "X0682.0373"], type = "l", col = "firebrick",
  ylim = range(-5:45), lwd = 2, xlab = "", ylab = "")
lines(DataN[, "X0682.0373"], type = "l", col = "dodgerblue3")
```



```

par(mar = c(2, 2, 1, 1))
plot(DataN[, "X0682.0373"], type = "l", col = "grey10", ylim = range(-5:45),
     xlab = "", ylab = "")
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)

```

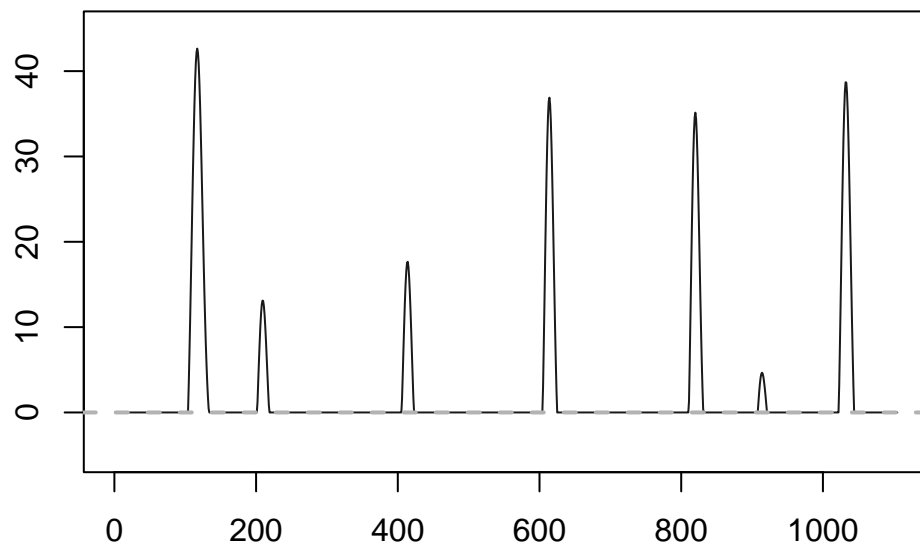


Remove long intervals

```

Inden = DataN[1:3600, ]
IndenT <- t(Inden)
Inden.NoIntv <- DataN[c(500:800, 1200:1400, 1800:2000, 2400:2600,
                       3000:3200), ]
IndenT.NoIntv = t(Inden.NoIntv)
par(mar = c(2, 2, 1, 1))
plot(Inden.NoIntv[, "X0682.0373"], type = "l", col = "grey10",
     ylim = range(-5:45), xlab = "", ylab = "")
abline(h = 0, col = "grey70", lty = "dashed", lwd = 2)

```



## Hierarchical clustering

register distance metric: FourierDistance

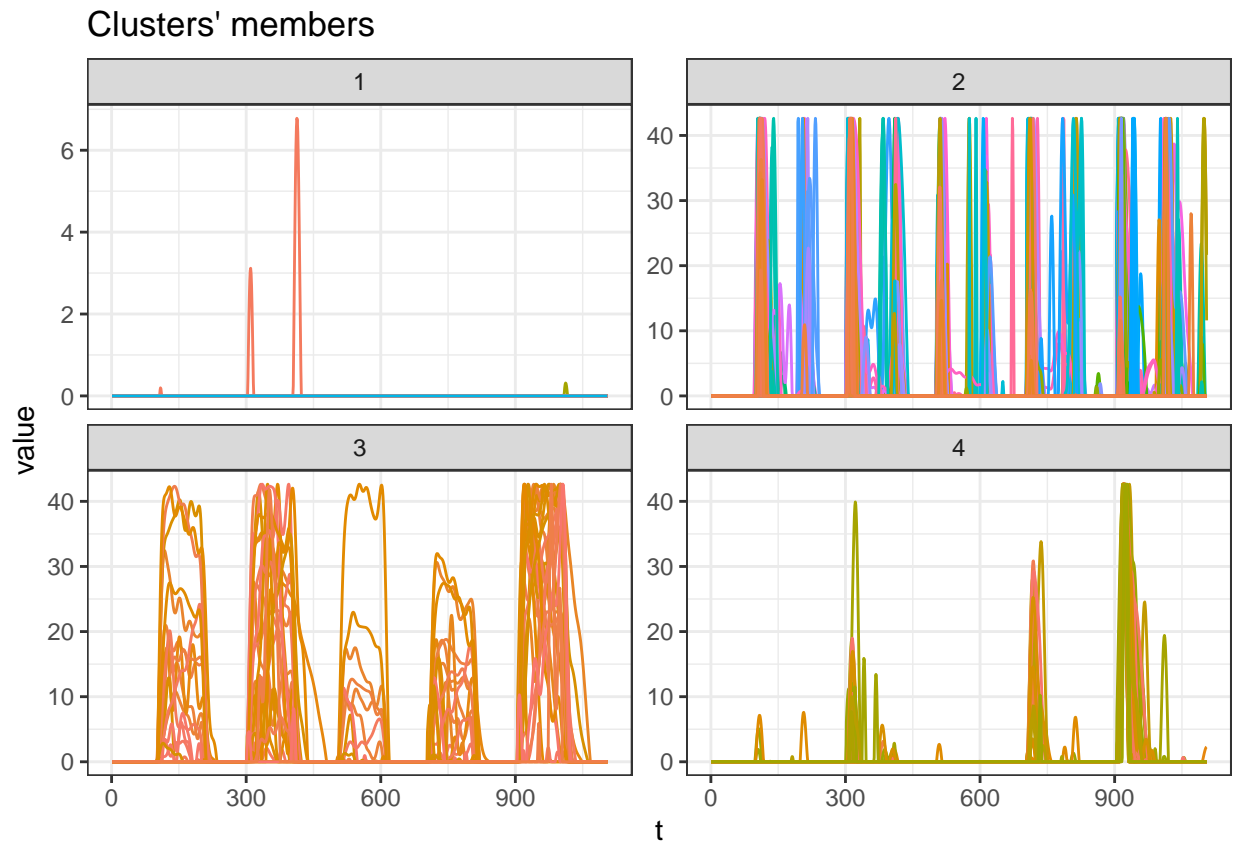
```
proxy::pr_DB$set_entry(FUN = FourierDistance, names = c("Fourier"),
  loop = TRUE, type = "metric", distance = TRUE, description = "Customized distance")
```

1st layer

```
hc <- tsclust(IndenT.NoIntv, type = "hierarchical", k = 4, trace = TRUE,
  distance = "Fourier", control = hierarchical_control(method = "ward.D"))
```

```
##
## Calculating distance matrix...
## Performing hierarchical clustering...
## Extracting centroids...
##
## Elapsed time is 42.02 seconds.
```

```
plot(hc, type = "series")
```



```
hc@clusinfo
```

```
##   size   av_dist
## 1  264   1.962624
## 2  267 3309.412219
## 3   28 6814.248582
## 4   57 1540.252361
```

## 2nd layer

data before response detection

```
Inden = Data.OMean[1:3600, ]
IndenT <- t(Inden)
Inden.NoIntv <- Data.OMean[c(500:800, 1200:1400, 1800:2000, 2400:2600,
  3000:3200), ]
IndenT.NoIntv = t(Inden.NoIntv)
```

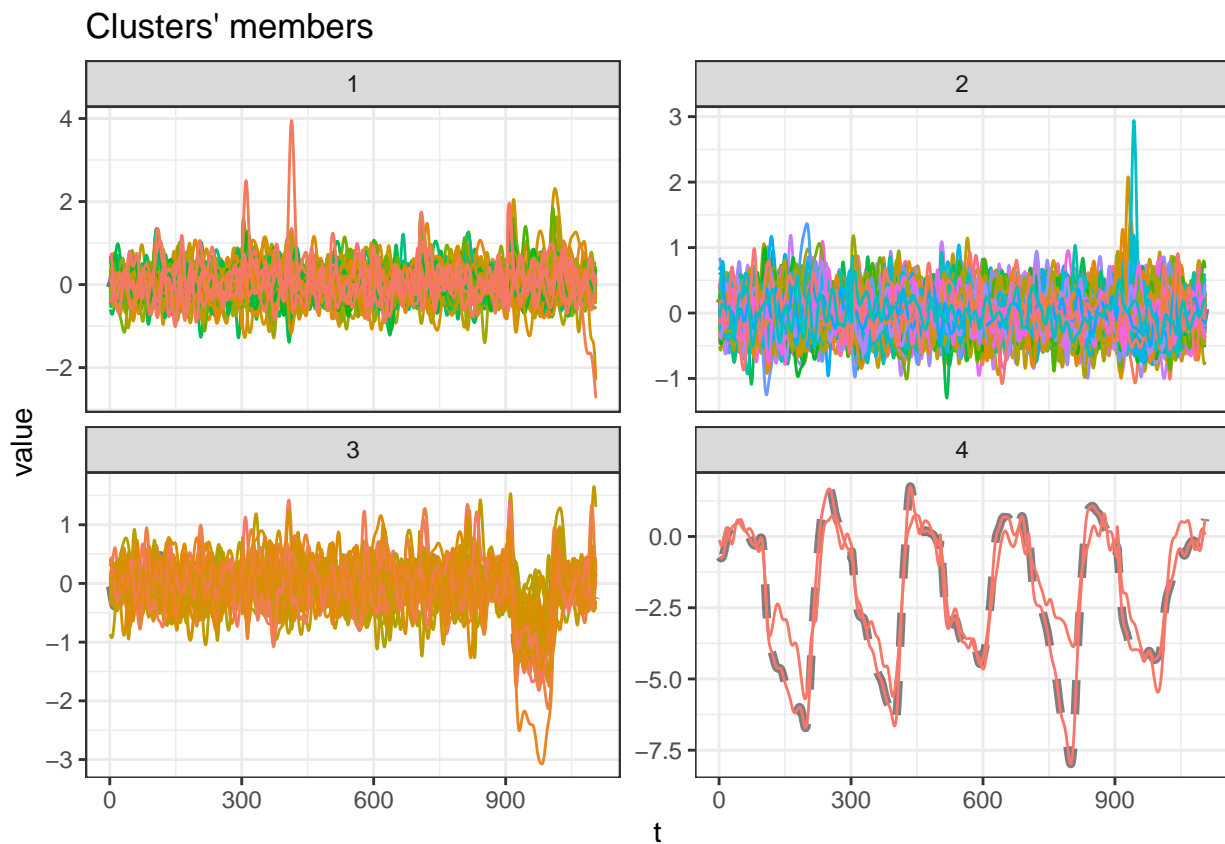
- Within cluster 1



```
group1_idx <- c(which(hc@cluster == 1))
IndenT.NoIntv.group1 <- IndenT.NoIntv[group1_idx, ]
hc.group1 <- tsclust(IndenT.NoIntv.group1, type = "hierarchical",
  k = 4, trace = TRUE, distance = "Fourier", control = hierarchical_control(method = "ward.D"))
```

```
##
## Calculating distance matrix...
## Performing hierarchical clustering...
## Extracting centroids...
##
## Elapsed time is 8.11 seconds.
```

```
plot(hc.group1, type = "sc")
```



```
hc.group1@clusinfo
```

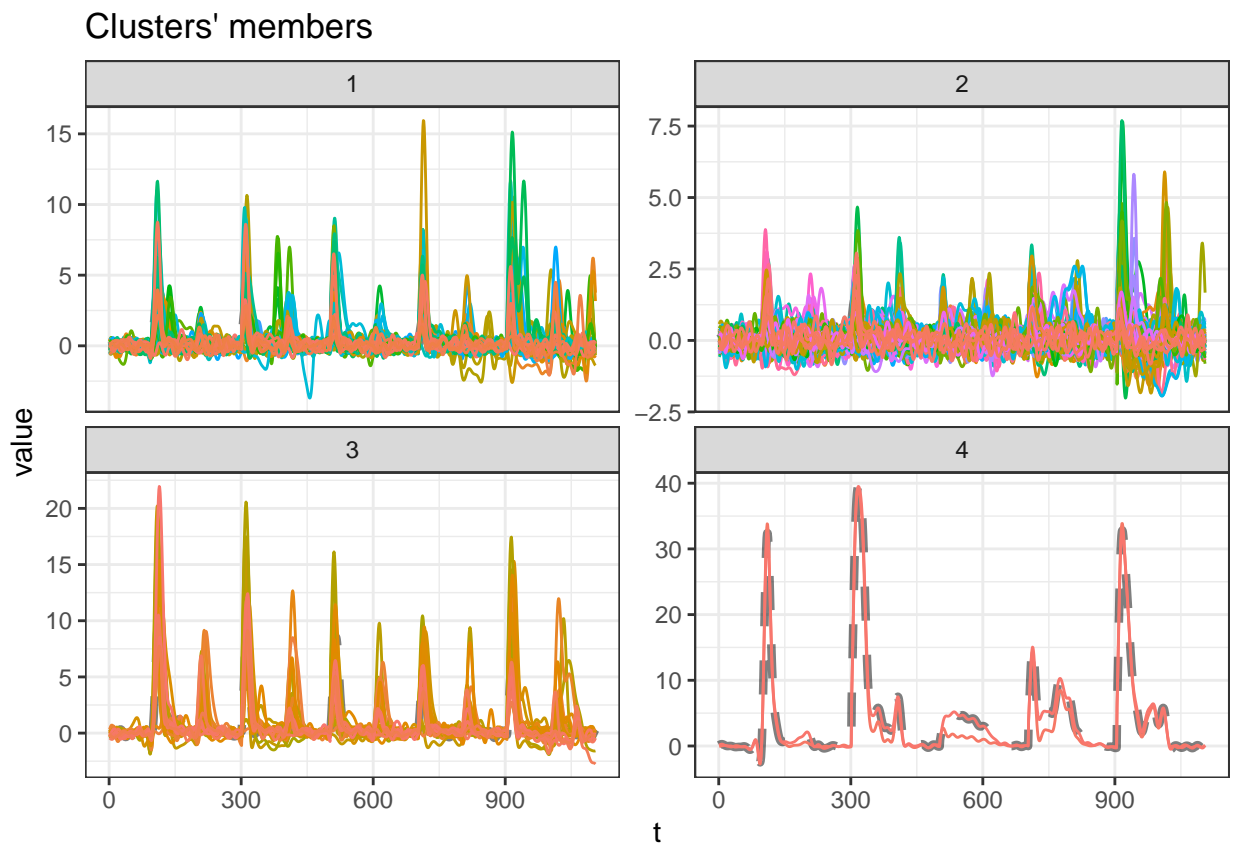
```
##   size av_dist
## 1   68 278.1642
## 2  163 221.3448
## 3   31 292.4714
## 4    2 456.3941
```

- Within cluster 2

```
group2_idx <- c(which(hc@cluster == 2))
IndenT.NoIntv.group2 <- IndenT.NoIntv[group2_idx, ]
hc.group2 <- tsclust(IndenT.NoIntv.group2, type = "hierarchical",
  k = 4, trace = TRUE, distance = "Fourier", control = hierarchical_control(method = "ward.D"))
```

```
##
## Calculating distance matrix...
## Performing hierarchical clustering...
## Extracting centroids...
##
## Elapsed time is 8.19 seconds.
```

```
plot(hc.group2, type = "sc")
```



```
hc.group2@clusinfo
```

```
##   size  av_dist
## 1   94  570.4530
## 2  143  334.2092
## 3   28 1087.8611
## 4    2  580.2068
```

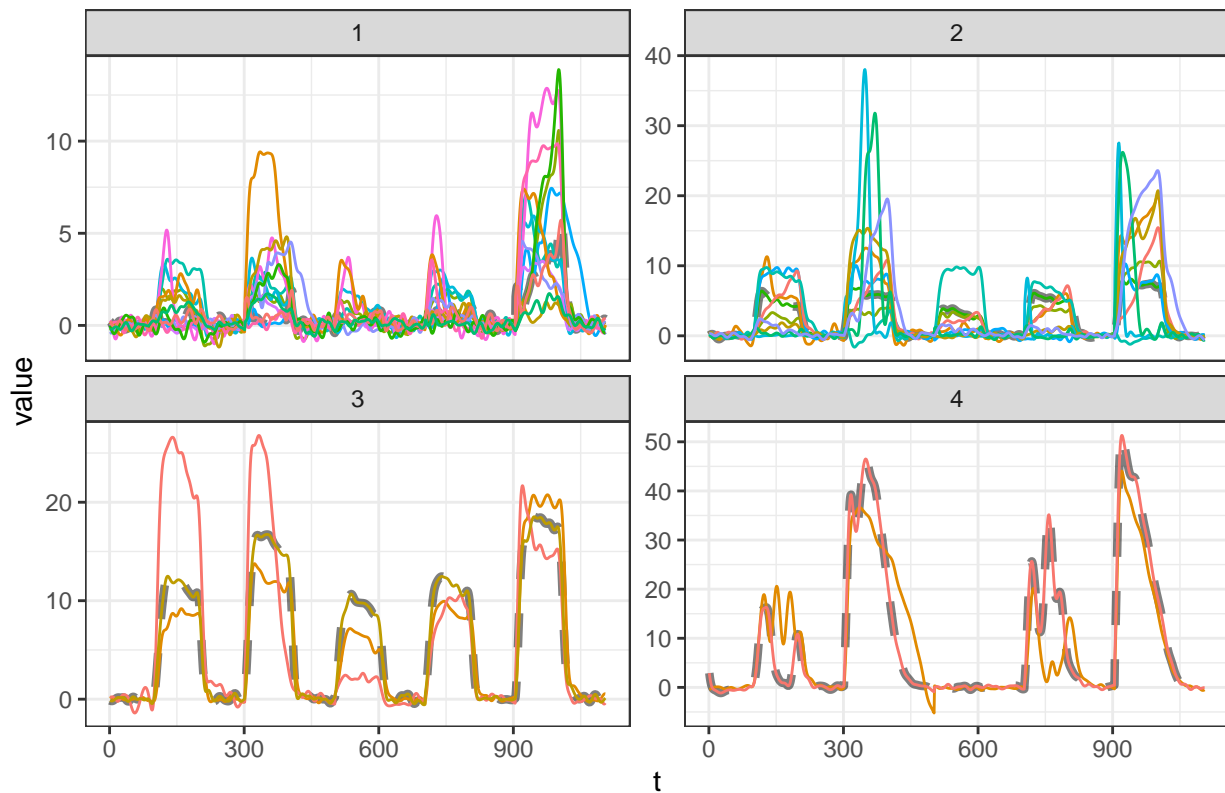
- Within cluster 3

```
group3_idx <- c(which(hc@cluster == 3))
IndenT.NoIntv.group3 <- IndenT.NoIntv[group3_idx, ]
hc.group3 <- tsclust(IndenT.NoIntv.group3, type = "hierarchical",
  k = 4, trace = TRUE, distance = "Fourier", control = hierarchical_control(method = "ward.D"))
```

```
##
## Calculating distance matrix...
## Performing hierarchical clustering...
## Extracting centroids...
##
## Elapsed time is 0.11 seconds.
```

```
plot(hc.group3, type = "sc")
```

### Clusters' members



```
hc.group3@clusinfo
```

```
##   size av_dist
## 1   13 1130.755
## 2   10 2595.421
## 3    3 1942.093
## 4    2 2690.347
```

- Within cluster 4

```
group4_idx <- c(which(hc@cluster == 4))
IndenT.NoIntv.group4 <- IndenT.NoIntv[group4_idx, ]
hc.group4 <- tsclust(IndenT.NoIntv.group4, type = "hierarchical",
  k = 4, trace = TRUE, distance = "Fourier", control = hierarchical_control(method = "ward.D"))
```

```
##
## Calculating distance matrix...
## Performing hierarchical clustering...
## Extracting centroids...
##
## Elapsed time is 0.44 seconds.
```

```
plot(hc.group4, type = "sc")
```



```
hc.group4@clusinfo
```

```
##   size  av_dist
## 1    6 1023.7348
## 2   27  403.2497
## 3    2 1432.9657
## 4   22  713.5191
```