# 8281_Application

*Wanting Jin, Yiling Xiong, Xinyue Lu*

*4/21/2019*

```
library("dtwclust")
data("uciCT")
```

# Character Trajectories Data Set

20 characters: A B C D E G H L M N O P Q R S U V W Y Z
5 samples for each

```
length(CharTraj)
```

```
## [1] 100
```

```
names(CharTraj)
```

```
##   [1] "A.V1" "A.V2" "A.V3" "A.V4" "A.V5" "B.V1" "B.V2" "B.V3" "B.V4" "B.V5"
##  [11] "C.V1" "C.V2" "C.V3" "C.V4" "C.V5" "D.V1" "D.V2" "D.V3" "D.V4" "D.V5"
##  [21] "E.V1" "E.V2" "E.V3" "E.V4" "E.V5" "G.V1" "G.V2" "G.V3" "G.V4" "G.V5"
##  [31] "H.V1" "H.V2" "H.V3" "H.V4" "H.V5" "L.V1" "L.V2" "L.V3" "L.V4" "L.V5"
##  [41] "M.V1" "M.V2" "M.V3" "M.V4" "M.V5" "N.V1" "N.V2" "N.V3" "N.V4" "N.V5"
##  [51] "O.V1" "O.V2" "O.V3" "O.V4" "O.V5" "P.V1" "P.V2" "P.V3" "P.V4" "P.V5"
##  [61] "Q.V1" "Q.V2" "Q.V3" "Q.V4" "Q.V5" "R.V1" "R.V2" "R.V3" "R.V4" "R.V5"
##  [71] "S.V1" "S.V2" "S.V3" "S.V4" "S.V5" "U.V1" "U.V2" "U.V3" "U.V4" "U.V5"
##  [81] "V.V1" "V.V2" "V.V3" "V.V4" "V.V5" "W.V1" "W.V2" "W.V3" "W.V4" "W.V5"
##  [91] "Y.V1" "Y.V2" "Y.V3" "Y.V4" "Y.V5" "Z.V1" "Z.V2" "Z.V3" "Z.V4" "Z.V5"
```

# 1-nearest-neighbor classifier

```
train <- CharTraj[-66L]
test <- CharTraj[66L]
d <- proxy::dist(train, test, method = "dtw", window.size = 10L)
cbind(test_data = names(CharTraj)[66], nearest_neighbor = rownames(d)[which.min(d)])
```

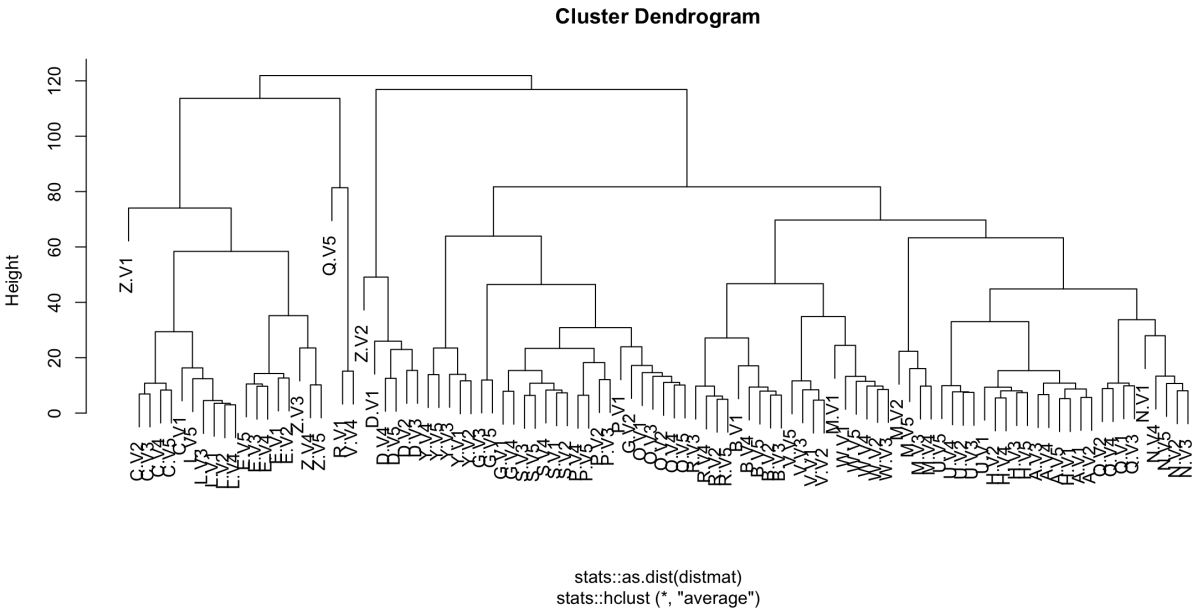
```
##      test_data nearest_neighbor
## [1,] "R.V1"    "R.V2"
```

# Hierarchical Clustering

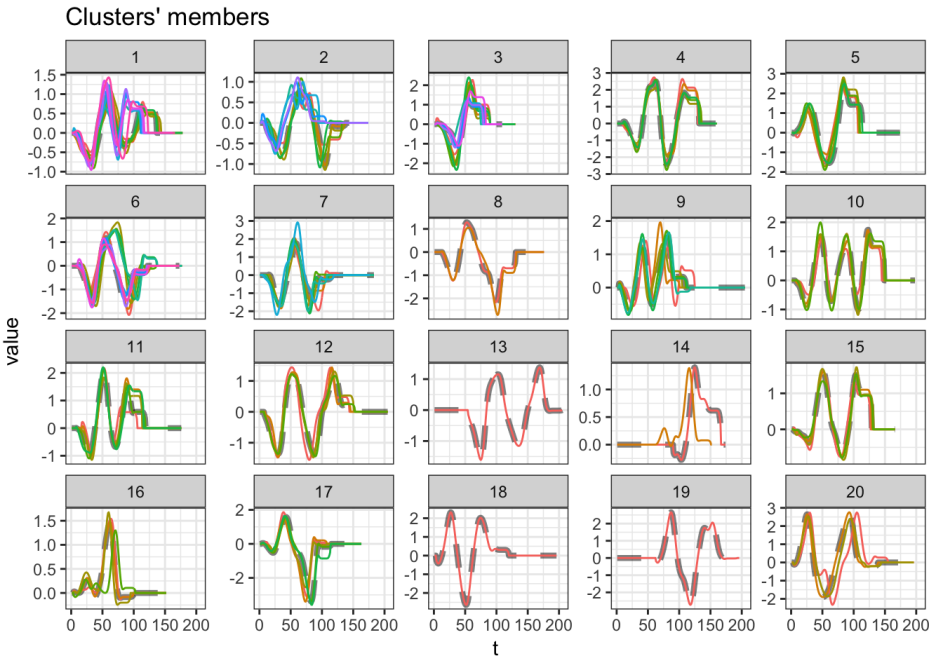
```
hc_dtw <- tsclust(CharTraj, type = 'hierarchical', k = 20L, distance = 'dtw',
                  control = hierarchical_control(method = "average"),
                  args = tsclust_args(dist = list(window.size = 18L)))
```

```
## Warning in tsclust(CharTraj, type = "hierarchical", k = 20L, distance =
## "dtw", : Distance matrix is not symmetric, and hierarchical clustering
## assumes it is (it ignores the upper triangular).
```

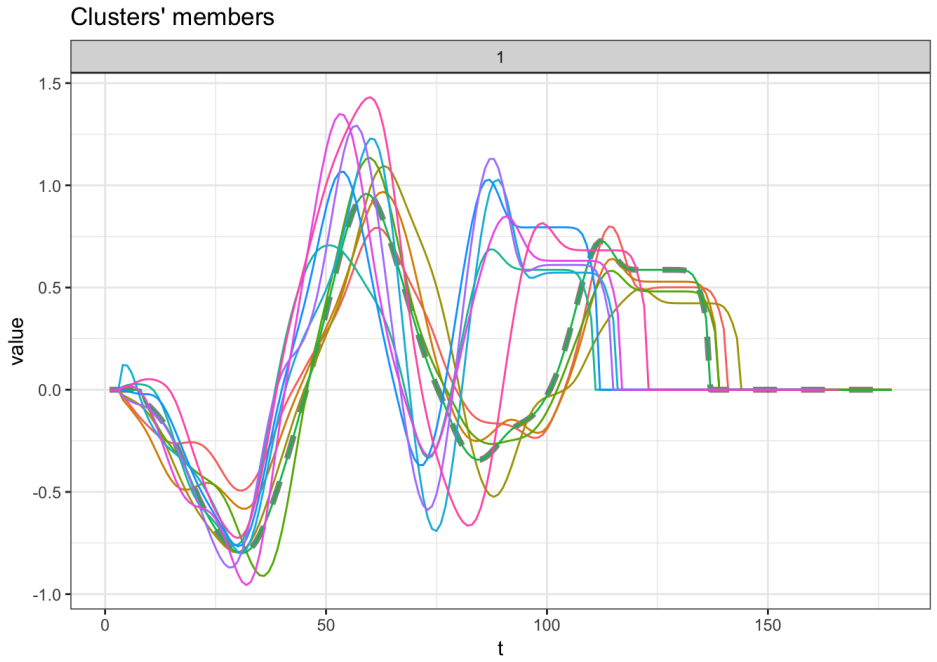
```
##plot the dendrogram
plot(hc_dtw, type = "dendrogram")
```




```
##plot the series and centroids
plot(hc_dtw, type = "sc")
```




```
##zoom in the first cluster
plot(hc_dtw, type = "sc", clus = 1L)
```




```
##check the ground labels and the resulted clusters
matrix(hc_dtw@cluster, ncol = 5, byrow = T)
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    1    1    1    1    1
##  [2,]    2    2    2    2    2
##  [3,]    3    3    3    3    3
##  [4,]    4    4    4    4    4
##  [5,]    5    5    5    5    5
##  [6,]    6    7    8    6    8
##  [7,]    1    1    1    1    1
##  [8,]    3    3    3    3    3
##  [9,]    9   10   10   10   10
## [10,]   11   11   11   11   11
## [11,]    7    7    7    7    7
## [12,]    7    6    6    6    6
## [13,]   12   12   12   12   13
## [14,]   14    2    2    2    2
## [15,]    6    6    6    6    6
## [16,]    1   15   15   15   15
## [17,]   16   16   16   14   16
## [18,]    9    9    9    9    9
## [19,]   17   17   17   17   17
## [20,]   18   19   20   20   20
```

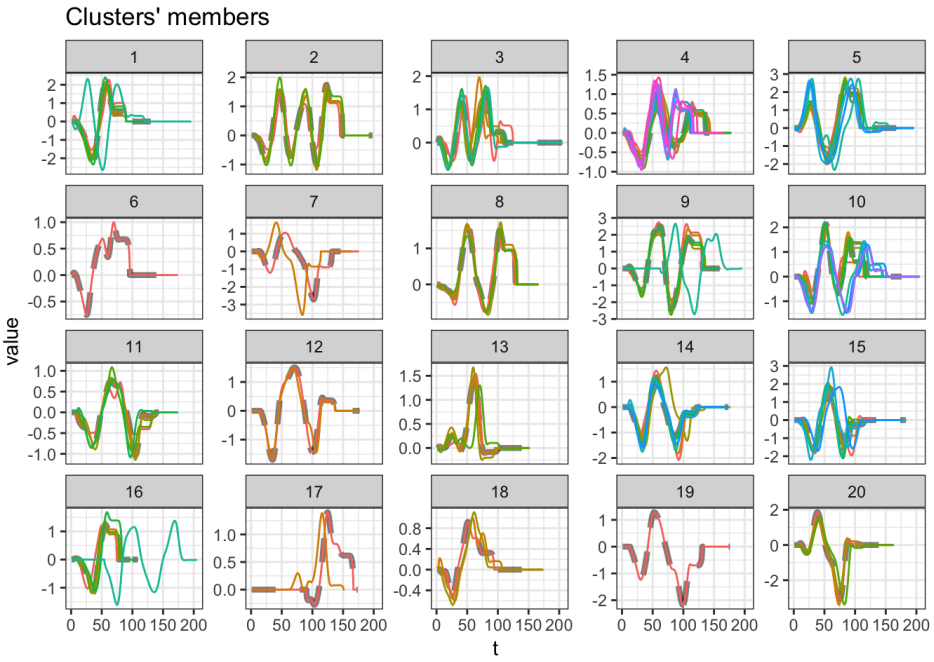
The $1^{st}$ character (A) and the $7^{th}$ character (H) are clustered into one. Let's apply the nearest-neighbor classifier again to see which has the closest DTW distance with the $7^{th}$ character

```
d2 <- proxy::dist(CharTraj[-c(31:35)], CharTraj[31:35], method = "dtw", window.size = 10L)
which.min(apply(matrix(apply(d2, 1, mean), ncol = 5, byrow = T), 1, mean))
```

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

# Partitional Clustering

```
pc_dtw <- tsclust(CharTraj, type = 'partitional', k = 20L, distance = 'dtw', centroid = "pam",
                  seed = 8,
                  args = tsclust_args(dist = list(window.size = 18L)))
##plot the series and centroids
plot(pc_dtw, type = "sc")
```




```
##check the ground labels and the resulted clusters
matrix(pc_dtw@cluster, ncol = 5, byrow = T)
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    4    4    4    4    4
##  [2,]   11   11   11   11   11
##  [3,]    1    1    1    1    1
##  [4,]    9    9    9    9    9
##  [5,]    5    5    5    5    5
##  [6,]   14   15   19   14    7
##  [7,]    4    4    4    4    4
##  [8,]   16   16   16   16   16
##  [9,]    3    2    2    2    2
## [10,]   10   10   10   10   10
## [11,]   15   15   15   15   15
## [12,]   15   15   14   12   12
## [13,]   10   10   10   10   16
## [14,]   17   18    6   18   18
## [15,]   14   14   14   14   14
## [16,]    4    8    8    8    8
## [17,]   13   13   13   17   13
## [18,]    3    3    3    3    3
## [19,]   20   20   20   20    7
## [20,]    1    9    5    5    5
```

```
#evaluate the two algorithms using "VI" values
sapply(list(DTW = pc_dtw, hc_dtw),
       cvi, b = CharTrajLabels, type = "VI")
```

```
##    DTW.VI        VI
## 0.3634873 0.3018201
```