Clustering growth curves

JM Nunes

2023-12-18

Preamble

This source document is intended to be run with **RStudio**.

This document requires some not too old **R** installation (if case of problem see session details at bottom), and a few packages that are not always installed by default, namely: ggplot2, reshape2, ggfortify, factoextra. Install them if necessary.

Finally, the data is supposed to be in a **data** directory in **protein_name.csv** files (one per protein).

Data acquisition

The data is stored in the directory data. The list of files is

```
files_to_read <-
  list.files(path='data', pattern='\\.csv$', full.names=TRUE)
files_to_read <- files_to_read[-c(15,16)]
files_to_read
```

```
##  [1] "data/C2CD3.csv"          "data/CCDC77.csv"
##  [3] "data/Centrin_distal.csv" "data/CEP135_distal.csv"
##  [5] "data/CEP135_prox.csv"    "data/CEP162.csv"
##  [7] "data/CEP290.csv"         "data/CEP295.csv"
##  [9] "data/CEP44.csv"          "data/CEP97.csv"
## [11] "data/CP110.csv"          "data/CPAP_distal.csv"
## [13] "data/CPAP_prox.csv"      "data/FAM161A.csv"
## [15] "data/POC5.csv"           "data/SAS6.csv"
## [17] "data/SFI1.csv"           "data/SPICE.csv"
## [19] "data/STIL.csv"           "data/WDR67.csv"
```

We start by reading a single file

```
read_one_file <- function(filename) {
  temp_df <- read.table(filename, header=TRUE, sep=',')
  names(temp_df) <- c('tubulin', 'length')
  temp_df$protein <- factor(gsub('.*/(.*)\\.csv', '\\1', filename))
  temp_df
}
test1 <- read_one_file(files_to_read[1])
summary(test1)
```

```
##     tubulin           length        protein
##  Min.   : 36.88   Min.   : 47.39   C2CD3:78
##  1st Qu.:172.34   1st Qu.:186.90
##  Median :378.53   Median :364.28
##  Mean   :318.23   Mean   :313.24
##  3rd Qu.:425.38   3rd Qu.:410.65
##  Max.   :616.83   Max.   :582.07
```

The data seems correct, we read all files (Map) and combine them (Reduce) into a single file.

```
growth <- Reduce(rbind, Map(read_one_file, files_to_read))
str(growth)
```

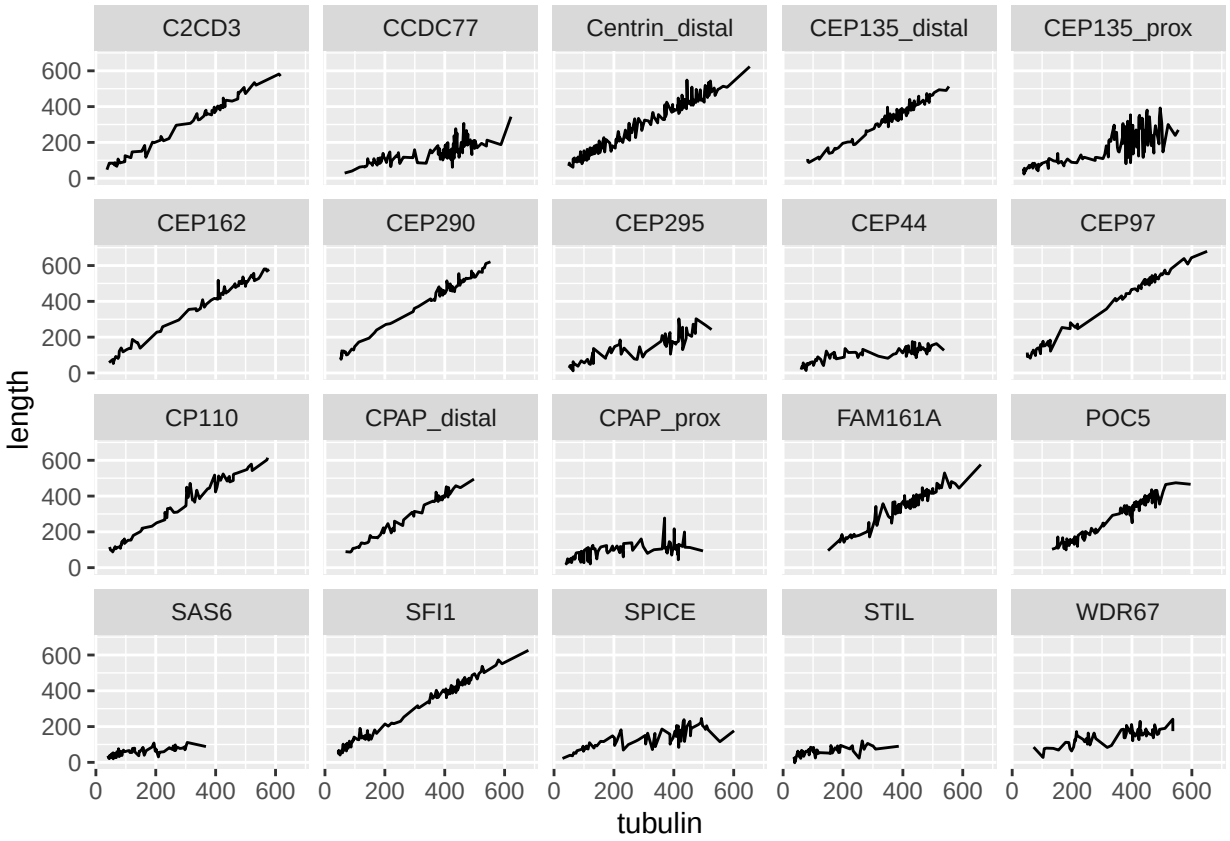
```
## 'data.frame':    1943 obs. of  3 variables:
##  $ tubulin: num  72.8 153.3 166.8 217.2 84.5 ...
##  $ length : num  89.7 150.6 116.4 233.6 87.5 ...
##  $ protein: Factor w/ 20 levels "C2CD3","CCDC77",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(growth)
```

```
##     tubulin           length                  protein
##  Min.   : 28.32   Min.   : -2.393   Centrin_distal: 187
##  1st Qu.:152.34   1st Qu.:108.447   SFI1          : 146
##  Median :361.13   Median :187.369   POC5          : 139
##  Mean   :305.72   Mean   :242.697   CEP135_prox   : 133
##  3rd Qu.:434.77   3rd Qu.:382.833   FAM161A       : 123
##  Max.   :679.44   Max.   :679.516   CCDC77        : 119
##                                     (Other)       :1096
```

We look at the distribution of protein lengths.

```
library(ggplot2)
ggplot(data=growth, aes(tubulin, length)) + geom_line() +
  facet_wrap(~protein)
```



Smoothing

To reduce the irregularities inherent to theis kind of data and facilitate comparability, we look for a smoothed version of each protein's growth curve. For that we calculate the loess fit of each protein and then apply this fit to a grid of evenly spaced tubulin lengths.

```
growth_by_prot <- split(growth, growth$protein)

make_loess_for_prot <- function(prot) loess(data=prot, length ~ tubulin)

growth_loess <- Map(make_loess_for_prot, growth_by_prot)

# points spaced by 10, between 30 and 600 ### maybe this needs some justification
create_tub_grid <- function(start=30, stop=600, step=10)
  data.frame(tubulin=seq(start, stop, step))
tub.grid <- create_tub_grid(stop=500)

predict_loess_for_prot.slice <- function(protfit) {
  # range m,M of tubulin lengths for current protein
  m <- min(protfit$x)
  M <- max(protfit$x)
  aux <- tub.grid[ !( tub.grid$tubulin < m | tub.grid$tubulin > M ), ]
  data.frame(tubulin=aux, length=predict(protfit, newdata=aux))
  }
```

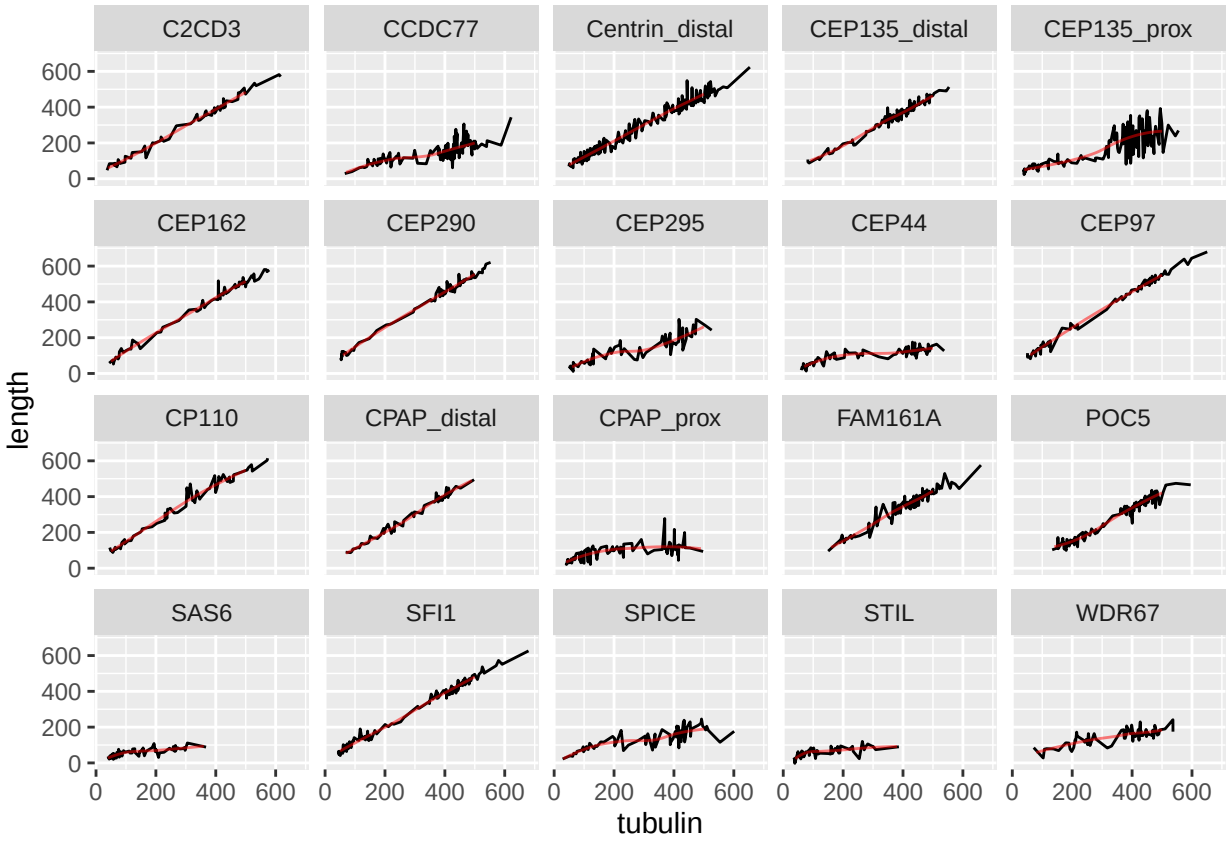
```
predict_loess_for_prot <- predict_loess_for_prot.slice

growth_smoothed <- Map(predict_loess_for_prot, growth_loess)
```

Graphical check

```
growth_smoothed_df <-
  Reduce(rbind,
    Map(cbind, growth_smoothed, protein=names(growth_smoothed)))

ggplot(data=growth, aes(tubulin, length)) + geom_line() +
  geom_line(data=growth_smoothed_df, colour='red', aes(tubulin, length), alpha=0.5) +
  facet_wrap(~protein)
```



Current data

Averaging distance

We calculate the manhattan distance between the smoothed curves of two proteins, that is, the sum of the absolute differences of their lengths on the smoothed fit at the points of the grid where both the curves are defined.

```r
str(growth_smoothed_df)
```

```
## 'data.frame':    863 obs. of  3 variables:
##  $ tubulin: num  40 50 60 70 80 90 100 110 120 130 ...
##  $ length : num  60 67.9 76 84.1 92.3 ...
##  $ protein: chr  "C2CD3" "C2CD3" "C2CD3" "C2CD3" ...
```

```r
summary(growth_smoothed_df)
```

```
##     tubulin          length         protein
##  Min.   : 30.0   Min.   : 23.18   Length:863
##  1st Qu.:170.0   1st Qu.:110.90   Class :character
##  Median :280.0   Median :169.77   Mode  :character
##  Mean   :275.8   Mean   :213.93
##  3rd Qu.:380.0   3rd Qu.:315.02
##  Max.   :500.0   Max.   :554.43
```

```r
prot_lengths_wide <- reshape2::dcast(growth_smoothed_df,  protein ~  tubulin, value.var='length')
# growth_smoothed_df[duplicated(growth_smoothed_df),]   ### this is ok now
summary(prot_lengths_wide)
```

```
##    protein                30              40              50
##  Length:20          Min.   :23.18   Min.   :24.04   Min.   :34.43
##  Class :character   1st Qu.:23.18   1st Qu.:28.75   1st Qu.:40.29
##  Mode  :character   Median :23.18   Median :32.84   Median :60.17
##                     Mean   :23.18   Mean   :37.95   Mean   :58.04
##                     3rd Qu.:23.18   3rd Qu.:46.15   3rd Qu.:68.13
##                     Max.   :23.18   Max.   :60.03   Max.   :95.10
##                     NA's   :19      NA's   :14      NA's   :10
##        60              70              80              90
##  Min.   : 36.09   Min.   : 35.69   Min.   : 42.08   Min.   : 48.27
##  1st Qu.: 47.85   1st Qu.: 51.43   1st Qu.: 58.52   1st Qu.: 62.51
##  Median : 75.95   Median : 71.23   Median : 62.57   Median : 85.00
##  Mean   : 69.48   Mean   : 73.09   Mean   : 79.72   Mean   : 88.02
##  3rd Qu.: 86.17   3rd Qu.: 91.85   3rd Qu.:101.94   3rd Qu.:110.19
##  Max.   :105.83   Max.   :116.58   Max.   :127.39   Max.   :138.27
##  NA's   :7        NA's   :4        NA's   :3        NA's   :2
##       100             110             120             130
##  Min.   : 54.27   Min.   : 60.05   Min.   : 62.85   Min.   : 63.98
##  1st Qu.: 66.45   1st Qu.: 71.70   1st Qu.: 75.26   1st Qu.: 78.91
##  Median : 91.72   Median : 97.99   Median :104.16   Median :110.27
##  Mean   : 95.23   Mean   :102.19   Mean   :109.02   Mean   :115.80
##  3rd Qu.:119.04   3rd Qu.:127.91   3rd Qu.:136.79   3rd Qu.:145.70
##  Max.   :149.18   Max.   :160.12   Max.   :171.15   Max.   :182.29
##  NA's   :2        NA's   :2        NA's   :2        NA's   :2
##       140             150             160             170
##  Min.   : 64.95   Min.   : 66.20   Min.   : 67.09   Min.   : 67.60
##  1st Qu.: 83.24   1st Qu.: 87.47   1st Qu.: 92.04   1st Qu.: 95.83
##  Median :120.30   Median :124.17   Median :122.50   Median :129.16
##  Mean   :122.37   Mean   :128.80   Mean   :134.19   Mean   :140.54
##  3rd Qu.:152.23   3rd Qu.:161.10   3rd Qu.:167.39   3rd Qu.:176.04
```

```
##  Max.   :193.52   Max.   :204.80   Max.   :216.13   Max.   :227.48
##  NA's   :1        NA's   :1
##       180              190              200              210
##  Min.   : 67.99   Min.   : 68.55   Min.   : 69.53   Min.   : 70.77
##  1st Qu.: 99.45   1st Qu.:102.47   1st Qu.:104.98   1st Qu.:108.13
##  Median :136.12   Median :143.36   Median :150.89   Median :158.70
##  Mean   :146.86   Mean   :153.18   Mean   :159.52   Mean   :165.86
##  3rd Qu.:184.68   3rd Qu.:193.85   3rd Qu.:203.06   3rd Qu.:212.39
##  Max.   :238.82   Max.   :250.15   Max.   :261.42   Max.   :272.63
##
##       220              230              240              250
##  Min.   : 72.12   Min.   : 73.53   Min.   : 75.01   Min.   : 76.52
##  1st Qu.:111.86   1st Qu.:116.10   1st Qu.:119.59   1st Qu.:121.58
##  Median :166.76   Median :175.05   Median :183.58   Median :192.50
##  Mean   :172.16   Mean   :178.41   Mean   :184.61   Mean   :190.80
##  3rd Qu.:221.70   3rd Qu.:230.89   3rd Qu.:240.10   3rd Qu.:249.58
##  Max.   :283.75   Max.   :294.77   Max.   :305.75   Max.   :316.78
##
##       260              270              280              290
##  Min.   : 78.05   Min.   : 79.57   Min.   : 81.1   Min.   : 82.62
##  1st Qu.:122.21   1st Qu.:122.93   1st Qu.:123.2   1st Qu.:123.29
##  Median :201.79   Median :211.40   Median :221.3   Median :231.36
##  Mean   :197.00   Mean   :203.22   Mean   :209.5   Mean   :215.85
##  3rd Qu.:259.21   3rd Qu.:269.34   3rd Qu.:279.8   3rd Qu.:290.48
##  Max.   :327.81   Max.   :338.79   Max.   :349.7   Max.   :360.37
##
##       300              310              320              330
##  Min.   : 84.15   Min.   : 85.68   Min.   : 87.23   Min.   : 88.79
##  1st Qu.:123.53   1st Qu.:124.09   1st Qu.:125.14   1st Qu.:126.83
##  Median :241.71   Median :252.30   Median :263.01   Median :273.71
##  Mean   :222.31   Mean   :228.87   Mean   :235.49   Mean   :242.14
##  3rd Qu.:301.35   3rd Qu.:311.44   3rd Qu.:321.21   3rd Qu.:330.97
##  Max.   :370.87   Max.   :381.16   Max.   :391.40   Max.   :401.52
##
##       340              350              360              370
##  Min.   : 89.94   Min.   : 90.6   Min.   : 91.18   Min.   : 91.7
##  1st Qu.:129.35   1st Qu.:132.9   1st Qu.:137.35   1st Qu.:150.0
##  Median :284.27   Median :294.6   Median :304.48   Median :318.6
##  Mean   :248.82   Mean   :255.5   Mean   :262.12   Mean   :277.9
##  3rd Qu.:340.67   3rd Qu.:350.3   3rd Qu.:359.72   3rd Qu.:372.6
##  Max.   :411.45   Max.   :421.1   Max.   :430.47   Max.   :439.6
##                                                    NA's   :1
##       380              390              400              410
##  Min.   : 92.15   Min.   :118.1   Min.   :119.4   Min.   :118.8
##  1st Qu.:154.37   1st Qu.:166.2   1st Qu.:169.8   1st Qu.:172.9
##  Median :327.33   Median :348.8   Median :357.4   Median :366.2
##  Mean   :284.72   Mean   :302.6   Mean   :309.6   Mean   :316.6
##  3rd Qu.:382.11   3rd Qu.:395.7   3rd Qu.:405.2   3rd Qu.:414.6
##  Max.   :448.43   Max.   :457.1   Max.   :465.8   Max.   :474.4
##  NA's   :1        NA's   :2       NA's   :2       NA's   :2
##       420              430              440              450
##  Min.   :118.2   Min.   :117.3   Min.   :116.3   Min.   :115.2
##  1st Qu.:176.0   1st Qu.:180.5   1st Qu.:184.7   1st Qu.:188.7
##  Median :374.7   Median :383.3   Median :392.1   Median :401.0
```

```
##  Mean   :323.4   Mean   :330.1   Mean   :336.8   Mean   :343.4
##  3rd Qu.:423.9   3rd Qu.:433.2   3rd Qu.:442.2   3rd Qu.:451.2
##  Max.   :483.0   Max.   :491.5   Max.   :499.8   Max.   :508.1
##  NA's   :2       NA's   :2       NA's   :2       NA's   :2
##       460             470             480             490
##  Min.   :113.9   Min.   :112.4   Min.   :110.8   Min.   :109.1
##  1st Qu.:193.1   1st Qu.:198.3   1st Qu.:203.7   1st Qu.:209.3
##  Median :410.0   Median :419.1   Median :428.5   Median :438.1
##  Mean   :349.9   Mean   :356.4   Mean   :362.9   Mean   :369.4
##  3rd Qu.:460.1   3rd Qu.:468.9   3rd Qu.:477.5   3rd Qu.:486.0
##  Max.   :516.3   Max.   :524.4   Max.   :534.1   Max.   :544.2
##  NA's   :2       NA's   :2       NA's   :2       NA's   :2
##       500
##  Min.   :141.3
##  1st Qu.:245.6
##  Median :447.3
##  Mean   :384.9
##  3rd Qu.:491.0
##  Max.   :554.4
##  NA's   :4
```

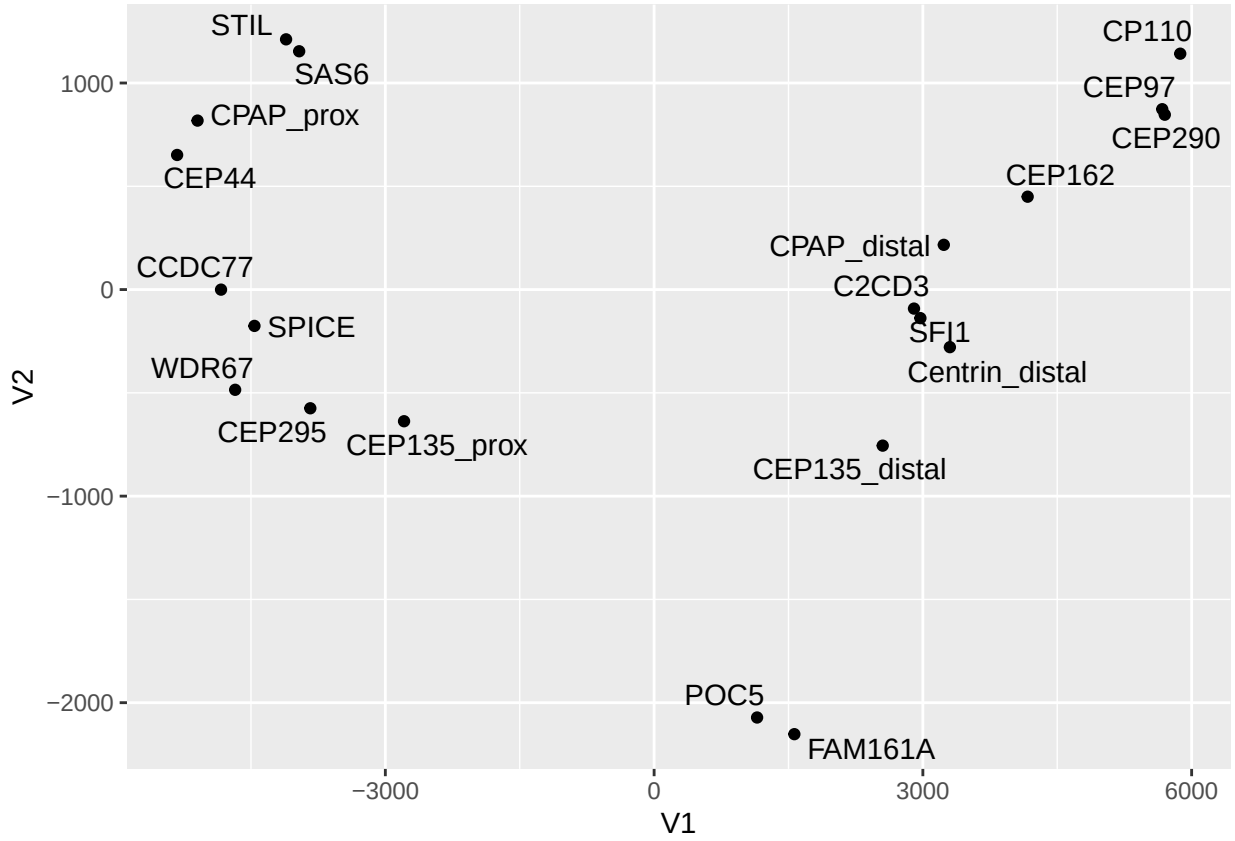
```r
prot_dists <- {
  aux <- dist(prot_lengths_wide, method='manha')
  aux <- as.matrix(aux)
  dimnames(aux) <- list(prot_lengths_wide$protein, prot_lengths_wide$protein)
  as.dist(aux)
}
```

```
## Warning in dist(prot_lengths_wide, method = "manha"): NAs introduced by
## coercion
```

A visual check using NMDS (preserves local distance structure better than PCA)

```r
library(ggfortify)
autoplot(MASS::sammon(prot_dists), label=TRUE, label.repel=TRUE)
```

```
## Initial stress        : 0.00926
## stress after  10 iters: 0.00144, magic = 0.500
## stress after  20 iters: 0.00134, magic = 0.500
```



Normalisation

The distances as calculated above make a proportional resizing for the independent (tubulin grid) lengths not present. According to the R documentation: > ... If some columns are excluded in calculating a Euclidean, Manhattan, Canberra or Minkowski distance, the sum is scaled up proportionally to the number of columns used. If all pairs are excluded when calculating a particular distance, the value is NA.

This affects distances because some proteins are not present all along the tubulin grid lengths. Some normalisation seems required. In order to normalise the values we need the get start and stop points for each protein, and protein pairs thereof. The average the manhattan distances over the common tubulin range. #### curr data

```
prot_limits <- {
  x0 <- aggregate(data=growth_smoothed_df, tubulin ~ protein, \(x) c(min(x), max(x)))
  x1 <- data.frame(protein=x0[[1]], min=x0[[2]][,1], max=x0[[2]][,2])
  x1
  }

prot_pairs_limits <- {
  x0 <- expand.grid(prot_limits$protein, prot_limits$protein, stringsAsFactors=FALSE)
  x0 <- x0[x0$Var1 >= x0$Var2,]
  get_prot_pairs_limits <- function(x,y) {
    x1 <- prot_limits[prot_limits$protein %in% c(x,y),]
    data.frame(Var1=x, Var2=y, min=max(x1$min), max=min(x1$max))
  }
```

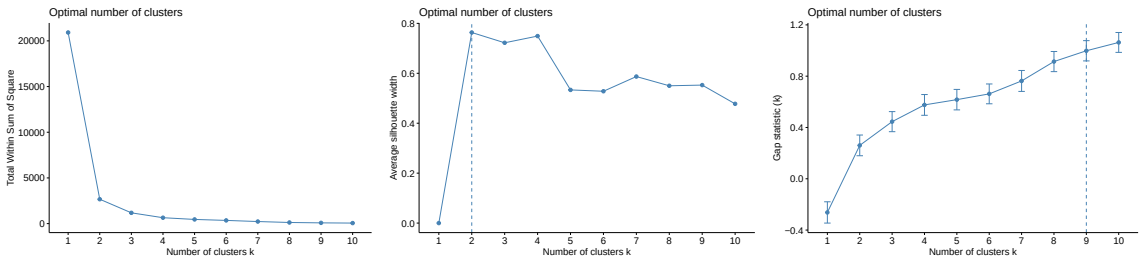
```
  Reduce(rbind, Map(get_prot_pairs_limits, x0$Var1, x0$Var2))
}

prot_pairs_dists <- {
  pair_dist <- function(x, y, m, M) {
    if (x == y) return(0)
    x0 <- growth_smoothed_df
    x0 <- x0[x0$protein %in% c(x, y) & x0$tubulin >= m & x0$tubulin <= M, ]
    x1 <- aggregate(data=x0, length ~ tubulin, \(x) abs(diff(x)))
    sum(x1$length) / (M-m)
  }
  cbind(
    prot_pairs_limits,
    dist=unlist(Map(pair_dist, prot_pairs_limits$Var1, prot_pairs_limits$Var2, prot_pairs_limits$min, p
  )
}
prot_dists_norm <- as.dist(t(as.matrix.data.frame(reshape2::dcast(data=prot_pairs_dists, Var2 ~ Var1, va
```

Cluster analysis

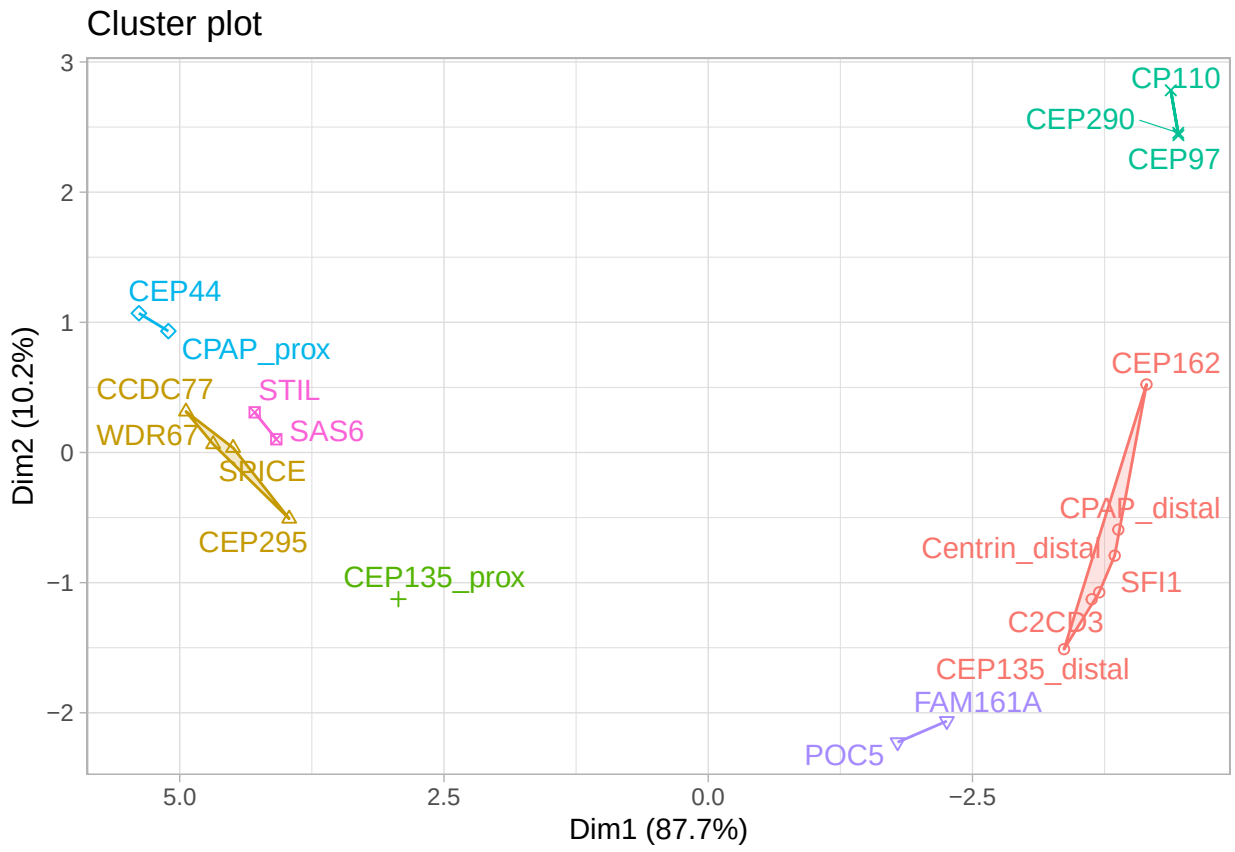
How many clusters to use?

```
library(cluster)
library(factoextra)
prot_dists_df <- as.matrix(prot_dists_norm)
#rownames(prot_dists_df) <-
#  rownames(prot_dists_df) <- levels(growth$protein)
fviz_nbclust(prot_dists_df, pam, 'wss')
fviz_nbclust(prot_dists_df, pam, 'silhouette')
fviz_nbclust(prot_dists_df, pam, 'gap_stat')
```



Combining the results above it seems reasonable to go with 2 or 7 clusters. We'll look at the representations for a final decision.

```
fviz_cluster(pam(prot_dists_df,7), repel=TRUE) +
  scale_x_reverse() +
#  coord_fixed() +
  theme_light() +
  guides(fill='none', shape='none', colour='none')
```



Colophon

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.6.8
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Zurich
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] cluster_2.1.4    factoextra_1.0.7 ggfortify_0.4.16 ggplot2_3.4.4
##
```

```
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4       generics_0.1.3   tidyr_1.3.0      rstatix_0.7.2
##  [5] stringi_1.8.3    digest_0.6.33    magrittr_2.0.3   evaluate_0.23
##  [9] grid_4.3.2       fastmap_1.1.1    plyr_1.8.9       ggrepel_0.9.4
## [13] backports_1.4.1  gridExtra_2.3    purrr_1.0.2      fansi_1.0.6
## [17] scales_1.3.0     abind_1.4-5      cli_3.6.2        rlang_1.1.2
## [21] munsell_0.5.0    withr_2.5.2      yaml_2.3.8       tools_4.3.2
## [25] reshape2_1.4.4   ggsignif_0.6.4   dplyr_1.1.4      colorspace_2.1-0
## [29] ggpubr_0.6.0     broom_1.0.5      vctrs_0.6.5      R6_2.5.1
## [33] lifecycle_1.0.4  stringr_1.5.1    car_3.1-2        MASS_7.3-60
## [37] pkgconfig_2.0.3  pillar_1.9.0     gtable_0.3.4     glue_1.6.2
## [41] Rcpp_1.0.11      xfun_0.41        tibble_3.2.1     tidyselect_1.2.0
## [45] highr_0.10       knitr_1.45       farver_2.1.1     htmltools_0.5.7
## [49] rmarkdown_2.25   carData_3.0-5    labeling_0.4.3   compiler_4.3.2
```