Clustering in Alzheimer’s Disease Neuroimaging Initiative Data - Insights from Longitudinal Metabolomics Data

Yao Chen

yc108@iu.edu

**Introduction**

Alzheimer's Disease (AD) is a progressive neurological disorder that is the most common cause of dementia, particularly affecting individuals aged 65 and above. The disease is characterized by a combination of risk factors, such as age, genetics, and pathophysiological changes, including abnormal protein accumulations and neuronal loss. Its symptoms include memory loss and cognitive decline.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset was obtained by recruiting cohorts of cognitively normal subjects, subjects with mild cognitive impairment (MCI), and subjects with Alzheimer disease with anticipated baseline characteristics. It was designed to characterize cross-sectionally and longitudinally neuroimaging, genetic, clinical, and longitudinal metabolomics measures of enrolled subjects.

Four widely used assessment tools for evaluating cognitive function and the progression of dementia, including Alzheimer's disease, are ADAS11 and ADAS13 (Alzheimer's Disease Assessment Scale-Cognitive Subscale), MMSE (Mini-Mental State Examination), and CDRSB (Clinical Dementia Rating Scale Sum of Boxes). They help in determining the stage of the disease and evaluating the efficacy of treatments.

This report aims to detect underlying patterns in longitudinal metabolomics data through longitudinal clustering to better understand the progression of AD and identify potential targets for therapeutic intervention.

**Methods**

Our analysis began with the ADNI cohort data, initially comprising 1524 patients across 13 time points (baseline to 120 months) and measurements of 781 metabolite identifiers. We focused on variables such as follow-up visit time, metabolomics abundance, demographic information, and cognitive performance metrics like CDRSB, ADAS11, ADAS13, and MMSE. We manually imputed incomplete follow-up times for 37 patients based on the records index. Exclusions were made for patients with follow-up visits greater than or equal to 48 months (n=600) and those with only a single visit (n=65), resulting in a final dataset of 1458 patients with 6 time points (baseline to 36 months) and every patient having at least two follow-ups. The baseline characteristics of the final dataset for clustering are in Table 1.

To control for confounding effects, Generalized Estimating Equations (GEE) were applied to every specific metabolite, considering age as a time-variant factor and gender and years of education as time-invariant factors. The GEE model residuals were then used for longitudinal clustering analysis.

The traditional k-means clustering, which partitions a dataset into 'k' clusters according to the distance between observations in a multidimensional space, was adapted for longitudinal data using the kml3d package in RStudio. However, longitudinal data presents a unique challenge; each subject's data represents a series of measurements over time, forming trajectories.

The implementation through the kml3d package extends the k-means algorithm by treating each individual's data as a temporal trajectory, not as discrete, static data points. In this framework, the centroid of a cluster is conceptualized as a dynamic trajectory that denotes the average path of all included trajectories, effectively capturing the progression pattern within each cluster. Moreover, kml3d can handle missing data points and unbalanced follow-up times, allowing us to apply it to our real-world ADNI data.

Table 1: Baseline characteristics and cognitive performance, at baseline, 30.4% of participants were Cognitively Normal (CN), 51.9% of participants had Mild Cognitive Impairment (MCI), and 17.7% of them had Dementia.

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristics** | **CN 443, 30.4%** | **MCI 757, 51.9%** | **Dementia 258, 17.7%** |
| Age mean (SD) | 74.2 (5.95) | 72.8 (7.43) | 75.2 (7.60) |
| Gender   Female N (%) | 226 (51.0) | 312 (41.2) | 111 (43.0) |
| Male N (%) | 217 (49.0) | 445 (58.8) | 147 (57.0) |
| Years of education mean (SD) | 16.4 (2.67) | 16.0 (2.79) | 15.2 (2.94) |
| CDRSB mean (SD) | 0.04 (0.14) | 1.53 (0.90) | 4.30 (1.62) |
| ADAS11 mean (SD) | 5.79 (2.87) | 10.2 (4.56) | 18.9 (6.45) |
| ADAS13 mean (SD) | 9.00 (4.27) | 29.1 (7.72) | 29.1 (7.72) |
| MMSE mean (SD) | 29.0 (1.14) | 27.6 (1.81) | 23.4 (1.98) |

A flowchart of patients

Description automatically generated

Figure 1: Flowchart of Data Preparation

**Results**

We performed k-means for joint longitudinal data (or joint trajectories) and evaluated 2, 3, 4 clusters with 10 iterations of the algorithm. In Figure 2, the Calinski–Harabasz index (CHI) determined that the optimal number of clusters was two, with 928 (64%) participants in one cluster and 530 (36%) in the other.

Cluster 2 consistently showed a higher percentage of dementia at all time points compared to Cluster 1, which could indicate that individuals in Cluster 2 are more likely to progress to dementia. After the first year of follow-up, the percentages of cognitively normal (CN) individuals increased over time in both clusters. At the same time, the percentage of MCI was inversely proportional to the trend in CN in Cluster 1. It may suggest a transition from CN to MCI, especially in Cluster 1.

Cluster 2 exhibited significantly higher scores on ADAS11 and ADAS13, alongside lower MMSE scores from baseline to 36 months of follow-up. And Cluster 2 showed consistently higher CDRSB scores compared with Cluster 1, indicating more severe cognitive impairment. All four assessments were shown to be statistically significant with p-values less than 0.01 in Chi-sq tests that compared the two clusters.

A graph of a line

Description automatically generated with medium confidence

Figure 2: The Assessment of the Clustering Quality by Calinski–Harabasz Index (CHI), a metric for evaluating clustering algorithms (the higher the better).

A graph of different colored lines

Description automatically generated with medium confidence

Figure 3: Percentages of Dementia (left), MCI(middle), CN(right) by Cluster Over Time.

A screenshot of a graph

Description automatically generated

Figure 4: Pairwise Boxplots for ADAS11(top left), ADAS13(top right), CDRSB(bottom left), MMSE(bottom right), x-axis shows (cluster.month), i.e., 2.12 is the Cluster 2 at 12 Months of follow-up.

**Discussion**

The clustering analysis of the ADNI dataset identified two clusters within the patient population, with Cluster 2 associated with a decline in cognitive function. This clustering result is helpful for the development of targeted therapeutic interventions and underscores the importance of personalized patient management. The significant differences in cognitive scores between clusters reinforce the potential of metabolomics data in distinguishing between different AD progression profiles. These findings offer a foundation for future research to dive into the biological underpinnings that may contribute to these distinct progression trajectories.

**Challenges and Considerations**

The study faced several limitations, including loss to follow-up and unbalanced data due to varying numbers of follow-up visits. Methodological challenges included the need for dimension reduction, where techniques such as Furry’s Common PCA and Longitudinal Functional PCA (LFPCA) proved to be computationally intensive. The use of kml3d for longitudinal multivariate clustering provided a solution for handling missing data and varying time points, although determining the optimal number of clusters was time-consuming. The increasing percentages of cognitively normal people over time might be attributed to bias, as participants who retained in the study could be due to their cognitive conditions.

Additionally, interpreting the significance of clusters required a comparison of individual trajectories, reflecting the complexity of AD progression and the multifactorial nature of its manifestation over time.

References

Genolini, C., Pingault, J. B., Driss, T., Côté, S., Tremblay, R. E., Vitaro, F., Arnaud, C., & Falissard, B. (2013). KmL3D: a non-parametric algorithm for clustering joint trajectories. *Computer methods and programs in biomedicine*, *109*(1), 104–111. <https://doi.org/10.1016/j.cmpb.2012.08.016>

Genolini, C., Alacoque, X., Sentenac, M., & Arnaud, C. (2015). kml and kml3d: R Packages to Cluster Longitudinal Data. *Journal of Statistical Software*, *65*(4), 1–34. <https://doi.org/10.18637/jss.v065.i04>

Petersen, R. C., Aisen, P. S., Beckett, L. A., Donohue, M. C., Gamst, A. C., Harvey, D. J., Jack, C. R., Jr, Jagust, W. J., Shaw, L. M., Toga, A. W., Trojanowski, J. Q., & Weiner, M. W. (2010). Alzheimer's Disease Neuroimaging Initiative (ADNI): clinical characterization. *Neurology*, *74*(3), 201–209. <https://doi.org/10.1212/WNL.0b013e3181cb3e25>

Appendix: R Codes

library(dplyr)

library(tidyr)

#BiocManager::install("pcaMethods")

# BiocManager::install("DMwR")

library(fda)

library(mclust)

library(kml)

library(kml3d)

library(lcmm)

library(geepack)

library(ggplot2)

library(patchwork)

setwd("~/Desktop/IUclasses/statistical\_learning/final\_project")

### load data 1

{

adni\_merge <- read.csv("data/ADNIMERGE.csv", header = T)

adni\_merge$RID <- as.character(adni\_merge$RID)

adni\_merge$time <- gsub("m", "", adni\_merge$VISCODE) # Remove 'm' from all other visit codes

adni\_merge$time <- gsub("bl", 0, adni\_merge$time) # Replace 'bl' with '0'

}

# load data 2

{

load("data/ADNI\_longdata.RData")

adni\_longdata <- cbind(id,x,Y)

names(adni\_longdata)[names(adni\_longdata) == "id"] <- "RID"

names(adni\_longdata)[names(adni\_longdata) == "x"] <- "time"

}

missing\_VIS\_ids <- adni\_longdata[is.na(adni\_longdata$time), 1]

adni\_longdata[adni\_longdata$RID %in% missing\_VIS\_ids,1:3]

id\_to\_time <- c(`101` = 24, `1044` = 24, `1098` = 0, `1116` = 0,

`1201` = 0, `1213` = 0, `1267` = 12, `1283` = 24,

`1300` = 0, `1393` = 36, `21` = 24, `2106` = 0,

`22` = 12, `2379` = 12, `291` = 0, `4001` = 12,

`4075` = 12, `41` = 24, `4104` = 24, `4415` = 36,

`4565` = 24, `4644` = 12, `4652` = 12, `4712` = 12,

`5135` = 24, `514` = 0, `621` = 12, `633` = 24,

`649` = 36, `692` = 24, `695` = 12, `759` = 0,

`783` = 12, `835` = 48, `861` = 12, `913` = 0,

`982` = 24)

adni\_longdata <- adni\_longdata %>%

mutate(time = ifelse(is.na(time) & (as.character(RID) %in% names(id\_to\_time)),

id\_to\_time[as.character(RID)],

time))

adni\_longdata[adni\_longdata$RID %in% missing\_VIS\_ids,1:3]

### GEE to adjust for age and patient gender

{

adni\_longdata <- left\_join(adni\_longdata,

adni\_merge %>% dplyr::select(RID, AGE, PTGENDER, PTEDUCAT),

by = "RID")

adni\_longdata <- adni\_longdata %>% distinct()

adni\_longdata <- adni\_longdata %>% dplyr::select(RID, AGE, PTGENDER, PTEDUCAT, everything())

adni\_longdata$AGE <- adni\_longdata$AGE + as.numeric(adni\_longdata$time)/12

adni\_longdata$time <- as.numeric(adni\_longdata$time )

}

{

adni\_res <- matrix(NA, nrow = nrow(adni\_longdata), ncol = 781)

for (i in 7:ncol(adni\_longdata)) {

response <- names(adni\_longdata)[i]

predictors <- c("AGE", "PTGENDER", "PTEDUCAT")

formula <- as.formula(paste(response, "~", paste(predictors, collapse = "+")))

# Use tryCatch to handle errors

tryCatch({

model <- geeglm(formula,

data = adni\_longdata,

id = RID,

family = gaussian, corstr = "unstructured")

# Store residuals in the matrix

adni\_res[, i-6] <- model$residuals

}, error = function(e){

cat("Error in modeling", response, ": ", e$message, "\n")

})

}

adni\_res\_time <- cbind(adni\_longdata[,c(1,5)], adni\_res)

names(adni\_res\_time)[3:783] <- names(adni\_longdata[7:787])

adni\_res\_time$time <- as.numeric(adni\_res\_time$time)

adni\_res\_time <- adni\_res\_time %>% arrange(RID, time)

save(adni\_res\_time, adni\_res, file = "data/adni\_residuals.RData")

}

load("data/adni\_residuals.RData")

### data pre-processing

{

# Step 1: Segment data

follow\_ups\_under\_48 <- adni\_res\_time[adni\_res\_time$time < 48, ]

follow\_ups\_over\_48 <- adni\_res\_time[adni\_res\_time$time >= 48, ]

# Step 2: Process follow-ups ≥ 48 months

# For each patient, keep only the most recent visit

# follow\_ups\_over\_48 <- follow\_ups\_over\_48 %>%

# group\_by(RID) %>%

# slice(which.min(time)) %>%

# ungroup()

# Step 3: Merge subsets

# adni\_res\_df <- rbind(follow\_ups\_under\_48, follow\_ups\_over\_48)

adni\_res\_df <- follow\_ups\_under\_48

# adni\_res\_df$new\_time <- ifelse(adni\_res\_df$time >= 48, 48, as.numeric(adni\_res\_df$time))

# Step 4: Remove patients with only one visit

adni\_res\_df<- adni\_res\_df %>%

group\_by(RID) %>%

filter(n() > 1) %>%

ungroup()

adni\_res\_df <- adni\_res\_df %>% dplyr::select(RID, time, everything())

# adni\_res\_df <- adni\_res\_df %>% dplyr::select(RID, time, new\_time, everything())

}

### dimension reduction

patient\_visits <- as.data.frame(table(adni\_res\_df$RID))

names(patient\_visits)[1] <- "RID"

table(patient\_visits$Freq)

table(adni\_res\_df$time)

table(adni\_longdata$time)

# library(multigroup)

# results <- FCPCA(as.matrix(adni\_res\_time[,-c(1,2)]),Group=adni\_longdata$time)

# Y <- as.matrix(adni\_res\_time[, -c(1, 2)]) # Remove subject ID and time columns

# subject <- adni\_res\_time$RID

# T <- adni\_res\_time$time

# results <- LFPCA(Y, subject, T)

###### kml3d

patient\_ids <- unique(adni\_res\_df$RID)

time\_points <- sort(unique(adni\_res\_df$time))

# Number of variables (excluding RID and time)

num\_vars <- ncol(adni\_res\_df) - 2

# Initialize the 3D array

adni\_array <- array(NA, dim = c(length(patient\_ids), length(time\_points), num\_vars))

# Fill the array

for (i in 1:length(patient\_ids)) {

for (j in 1:length(time\_points)) {

# Subset for each patient and time point

subset\_data <- adni\_res\_df[adni\_res\_df$RID == patient\_ids[i] &

adni\_res\_df$time ==

time\_points[j], -c(1,2)]

if (nrow(subset\_data) == 1) {

adni\_array[i, j, ] <- as.numeric(subset\_data)

}

}

}

myCLD3d <- clusterLongData3d(traj = adni\_array,

idAll = as.character(patient\_ids),

time = time\_points,

varNames = colnames(adni\_res\_df)[3:ncol(adni\_res\_df)],

maxNA = 4 # Adjust this based on your data's sparsity

)

kml3d(myCLD3d,2:4,10, toPlot="criterion")

#plot(myCLD3d,2,parMean=parMEAN(type="l"))

plot(myCLD3d,2,parTraj=parTRAJ(col="clusters"))

cluster2 <- myCLD3d@c2[[1]]@clusters

cluster2 <- cluster2

# Inspect the result

# subset\_df <- df[df$column\_name == condition, ]

length(patient\_ids)

clustering\_result <- cbind(patient\_ids, cluster2)

colnames(clustering\_result)[1] <- c("RID")

clustering\_subset <- adni\_merge[adni\_merge$RID %in% patient\_ids,]

clustering\_subset <- merge(clustering\_subset, clustering\_result, by="RID")

clustering\_subset1 <- clustering\_subset[clustering\_subset$cluster2==1, ]

clustering\_subset2 <- clustering\_subset[clustering\_subset$cluster2==2,]

prop.table(table(clustering\_subset1$DX\_bl))

prop.table(table(clustering\_subset2$DX\_bl))

prop.table(table(clustering\_subset1$DX))

prop.table(table(clustering\_subset2$DX))

clustering\_subset\_bl <- clustering\_subset[clustering\_subset$time==0,c(1,8:11,19:22, 52,55:57, 95)]

bl\_sum\_by\_dx <- clustering\_subset\_bl %>%

group\_by(DX) %>%

summarise( age\_mean = mean(AGE, na.rm = TRUE),

age\_sd = sd(AGE, na.rm = TRUE),

YrsEDU\_mean = mean(PTEDUCAT, na.rm = TRUE),

YrsEDU\_sd = sd(PTEDUCAT, na.rm = TRUE),

CDRSB\_mean = mean(CDRSB, na.rm = TRUE),

CDRSB\_sd = sd(CDRSB, na.rm = TRUE),

ADAS11\_mean = mean(ADAS11\_bl, na.rm = TRUE),

ADAS11\_sd = sd(ADAS11\_bl, na.rm = TRUE),

ADAS13\_mean = mean(ADAS13\_bl, na.rm = TRUE),

ADAS13\_sd = sd(ADAS13\_bl, na.rm = TRUE),

MMSE\_mean = mean(MMSE\_bl, na.rm = TRUE),

MMSE\_sd = sd(MMSE\_bl, na.rm = TRUE) )

library(ggplot2)

{

plot <- na.omit(clustering\_subset[as.numeric(clustering\_subset$time) < 80, c(1,8,52,95,96)])

plot$time <- as.numeric(plot$time)

plot <- plot %>%

mutate(DX = factor(DX, levels = c("CN", "MCI", "Dementia")))

plot <- na.omit(plot)

plot\_data <- plot %>%

group\_by(cluster2, time) %>%

count(DX) %>%

mutate(total = sum(n),

percentage = n / total \* 100) %>%

ungroup()

names(plot\_data)[1] <- c("Clusters")

plot1 <- ggplot(plot\_data %>% filter(DX == "Dementia"), aes(x = time, y = percentage, fill = Clusters)) +

geom\_bar(stat = "identity", position = position\_dodge()) +

facet\_wrap(~ Clusters) +

labs(title = "Percentage of Dementia by Cluster Over Time",

x = "Time",

y = "Percentage (%)") +

theme\_minimal()+

theme(legend.position = "none")

plot2 <- ggplot(plot\_data %>% filter(DX == "MCI"), aes(x = time, y = percentage, fill = Clusters)) +

geom\_bar(stat = "identity", position = position\_dodge()) +

facet\_wrap(~ Clusters) +

labs(title = "Percentage of MCI by Cluster Over Time",

x = "Time",

y = " ") +

theme\_minimal()+

theme(legend.position = "none")

plot3 <- ggplot(plot\_data %>% filter(DX == "CN"), aes(x = time, y = percentage, fill = Clusters)) +

geom\_bar(stat = "identity", position = position\_dodge()) +

facet\_wrap(~ Clusters) +

labs(title = "Percentage of CN by Cluster Over Time",

x = "Time",

y = " ") +

theme\_minimal()+

theme(legend.position = "none")

combined\_plot <- (plot1 | plot2 | plot3) & theme(legend.position = "none")

}

combined\_plot

chisq\_data <- clustering\_subset[as.numeric(clustering\_subset$time) < 80, c(1,8,19:22, 52,55:57, 95,96)]

chisq\_data$time <- as.numeric(chisq\_data$time)

chisq\_data <- chisq\_data %>%

mutate(DX = factor(DX, levels = c("CN", "MCI", "Dementia")))

chisq\_data <- na.omit(chisq\_data)

chisq\_data <- chisq\_data %>%

group\_by(cluster2, time) %>%

ungroup()

# You can also use chi-square test to check for independence

chisq.test(chisq\_data$cluster2, chisq\_data$DX)

long\_chisq\_data <- chisq\_data %>%

pivot\_longer(cols = c("CDRSB", "MMSE", "ADAS11", "ADAS13"),

names\_to = "Test",

values\_to = "Value")

ggplot(long\_chisq\_data, aes(x = cluster2, y = Value, fill = Test)) +

geom\_boxplot() +

facet\_wrap(~Test, scales = "free") +

labs(title = "Pairwise Boxplots for CDRSB, MMSE, ADAS11, and ADAS13",

x = "Cluster",

y = "Score",

fill = "Test") +

theme\_minimal()

ggplot(long\_chisq\_data, aes(x = interaction(cluster2, time), y = Value, fill = cluster2)) +

geom\_boxplot() + # Creates boxplots

facet\_wrap(~Test, scales = "free\_y") +

labs(title = "Pairwise Boxplots for CDRSB, MMSE, ADAS11, and ADAS13 Over Time",

x = "Cluster and Time Point",

y = "Score",

fill = "Cluster") +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))