UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO E
FACULDADE DE MEDICINA DE RIBEIRÃO PRETO

Trabalho de Reconhecimento de Padrões

Professor: Ricardo Zorzetto Nicoliello Vêncio
Disciplina: IBM1090 - Reconhecimento de Padrões

Jessica Caroline Alves Nunes Temporal **Nº USP: 7547611**
Raíssa Poch **Nº USP: 8058889**
Wilbert Dener Lemos Costa **Nº USP: 7961760**

Ribeirão Preto
Dezembro, 2015

But there is only one surefire method
of proper pattern recognition, and
that is science.

- Michael Shermer

SUMÁRIO

Introdução

Com o objetivo de utilizar as técnicas de reconhecimento de padrões aprendidas durante o segundo semestre do ano 2015, escolhemos um estudo publicado sobre dengue para automatizar e realizar os passos para analizar os dados usando a linguagem R.

A dengue é uma doença infecciosa transmitida por mosquitos, que causa uma doença ciclica em quase 100 milhões de pessoas anualmente (Bhatt et al., 2013). A infecção pode ser de diferentes tipos de serótipos (febre, febre hemorrágica ou síndrome do choque da dengue, que é uma doença com risco de vida (Simmons et al., 2012)).

O vírus da dengue induz a expansão de plasmablastos , o qual produz anticorpos que podem neutralizar o vírus da dengue mas também agravar a doença por infecção secundária com outro serótipo de vírus. Para entender como foram gerados, foi utilizada uma abordagem biológica de sistemas para analisar, em seres humanos, as respostas imunes para a dengue. A analise de transcriptoma total do sangue revela que os genes que codificam mediadores pró-inflamatórios do tipo I e IFN-relacionada a proteínas que são relacionados com alto níveis de vírus da dengue durante o início sintomático da doença. Adicionalmente, monócitos de $CD14^+$ e $CD16^+$ foram incluídos no sangue. Similarmente, modelo de primata não humano, a infecção do virus da dengue impulsionou o numero de monócitos $CD14^+$ e $CD16^+$ no sangue e no gânglios linfáticos. Após a infecção da dengue in vitro, monócitos supra regulados $CD16^+$ e mediadas diferenciação de células B em repouso para plasmablasts, bem como a secreção de IgC e IgM. Esses resultados fornecem uma visão detalhada de respostas inatas para a dengue e destacam um papel para monócitos $CD14^+$ e $CD16^+$ na promoção da diferenciação de plasmablast e respostas de anticorpos anti-dengue.

Outro aspecto característico da infecção por dengue é a expansão maciça de plasmablastos produtoras de anticorpos no sangue, que ocorre dentro de poucos dias de infecção (Balakrishnan et ia , 2011; . Garcia – Bates et al , 2013;. Wrammert et ai, 2012.). No entanto, embora a infecção com dados serótipos podem induzir anticorpos que são reativos transversalmente a outros serótipos, em geral, geralmente imunidade a longo prazo é gerada apenas contra o serótipo original (Green and Rothman, 2006). De fato, em muitos casos, imunidade contra o serótipo heterólogo não é protetora mas pode aumentar a gravidade da doença (Burke et al., 1988; ., Sangkawibha et ak., 1984), possivelmente através de um mecanismo denominado “amplificação dependente de anticorpos ” (Halstead et al., 2010) .

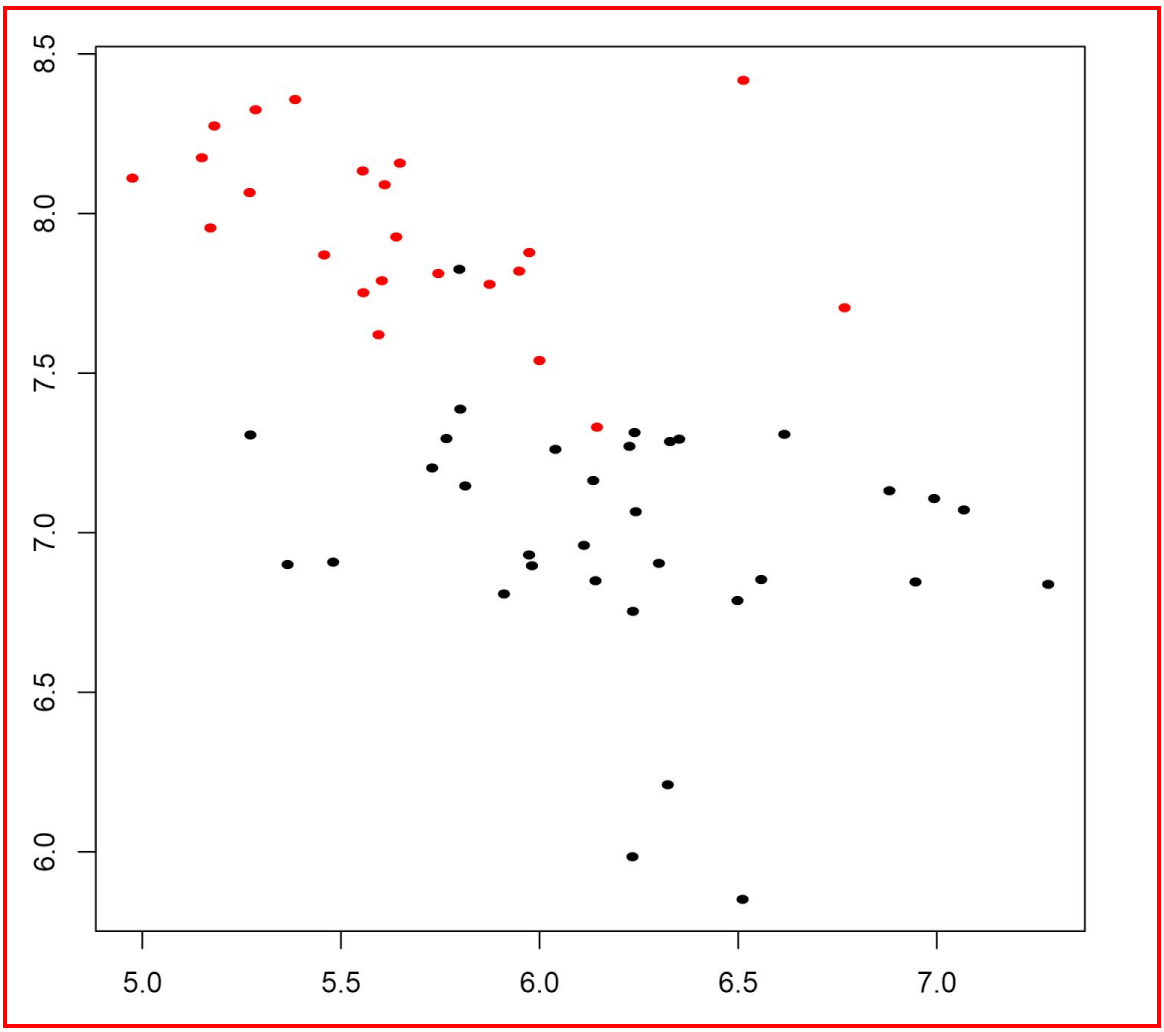
O artigo uma abordagem integrada para obter uma imagem detalhada da resposta inata durante a dengue aguda, nossos perfis de transcrição e analise imunológica de pacientes com dengue clínica, em conjunto com os resultados de um modelo de um primata não humano de infecção com o vírus da dengue e experiências in vitro sugerem um papel distinto de monócitos de $CD14^+$ e $CD16^+$ na mediação da imunidade humoral à infecção do vírus da dengue.

Métodos utilizados e Resultados

Kmeans - Análise de Clusters

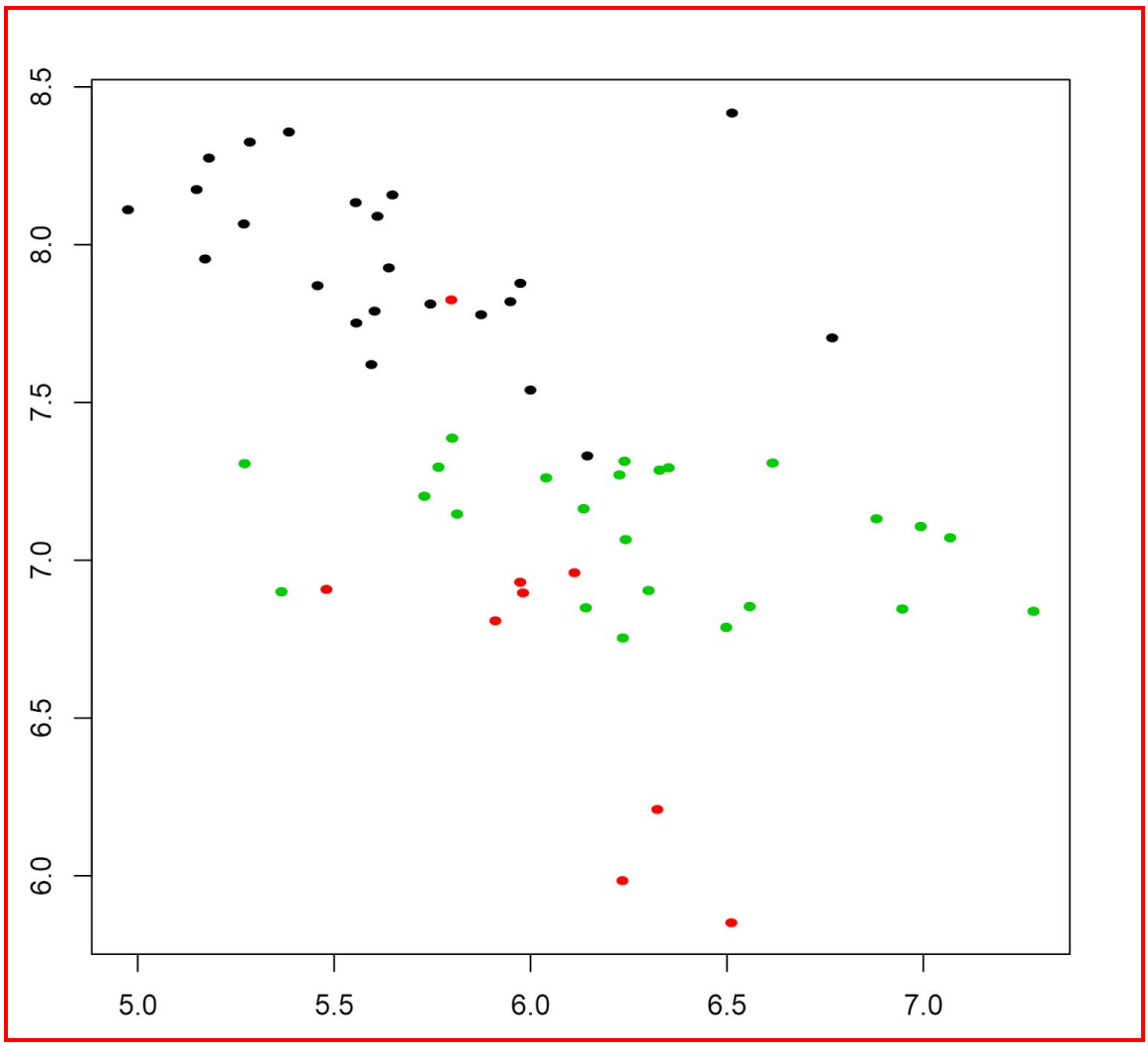
O *kmeans* é um método de análise não supervisionada de particionamento o qual gera os grupos baseado em médias. A ideia consiste em indicar a média de cada grupo dividindo (ou particionando) as amostras e essa média define um centro entre os *k* centros. Logo, o centros com médias muito próximas vão se juntando até sobrarem o k centros com médias mais distantes. Sendo assim, o algoritmo do *kmeans* é bem simples, porém pouco flexível, dado que ele tenta separar ao máximo os centróides de forma recursiva. É um processo custoso, principalmente se quiser analisar muitos centros, tendo em vista que o kmeans trabalha com o quadrado das distancias euclidianas entre os centros.

Kmeans: k=2



Podemos notar que amostras se separam em dois grupos principais quando k=2, porém, ainda encontramos uma outra que acabam por se misturar no outro grupo.

Kmeans: k=3

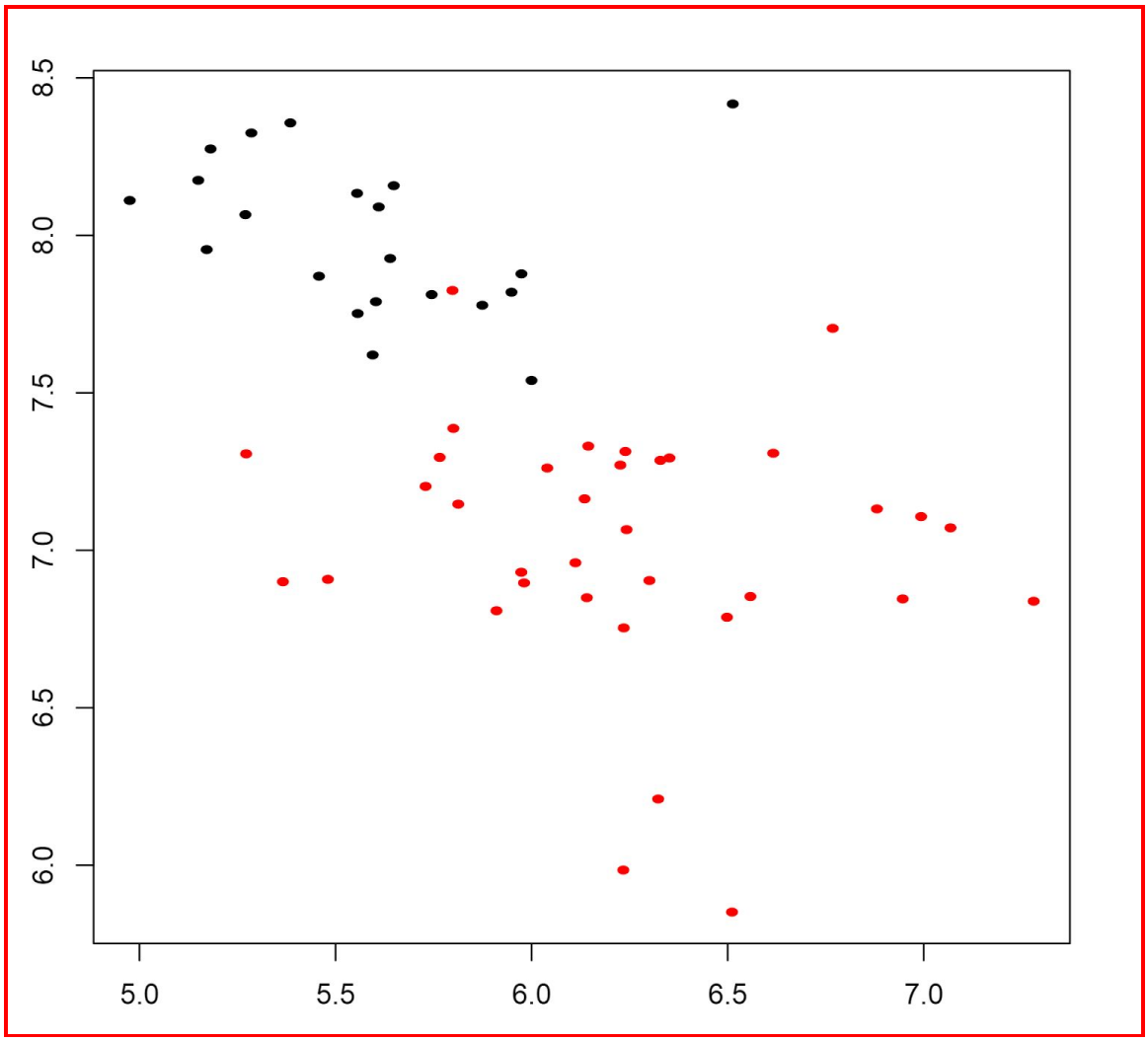


Note que, tanto com 2 clusters quanto 3 clusters, o kmeans mostra que algumas amostras se misturam entre os grupos. Mais a frente mostraremos quais são essas amostras.

PAM - Análise de Clusters

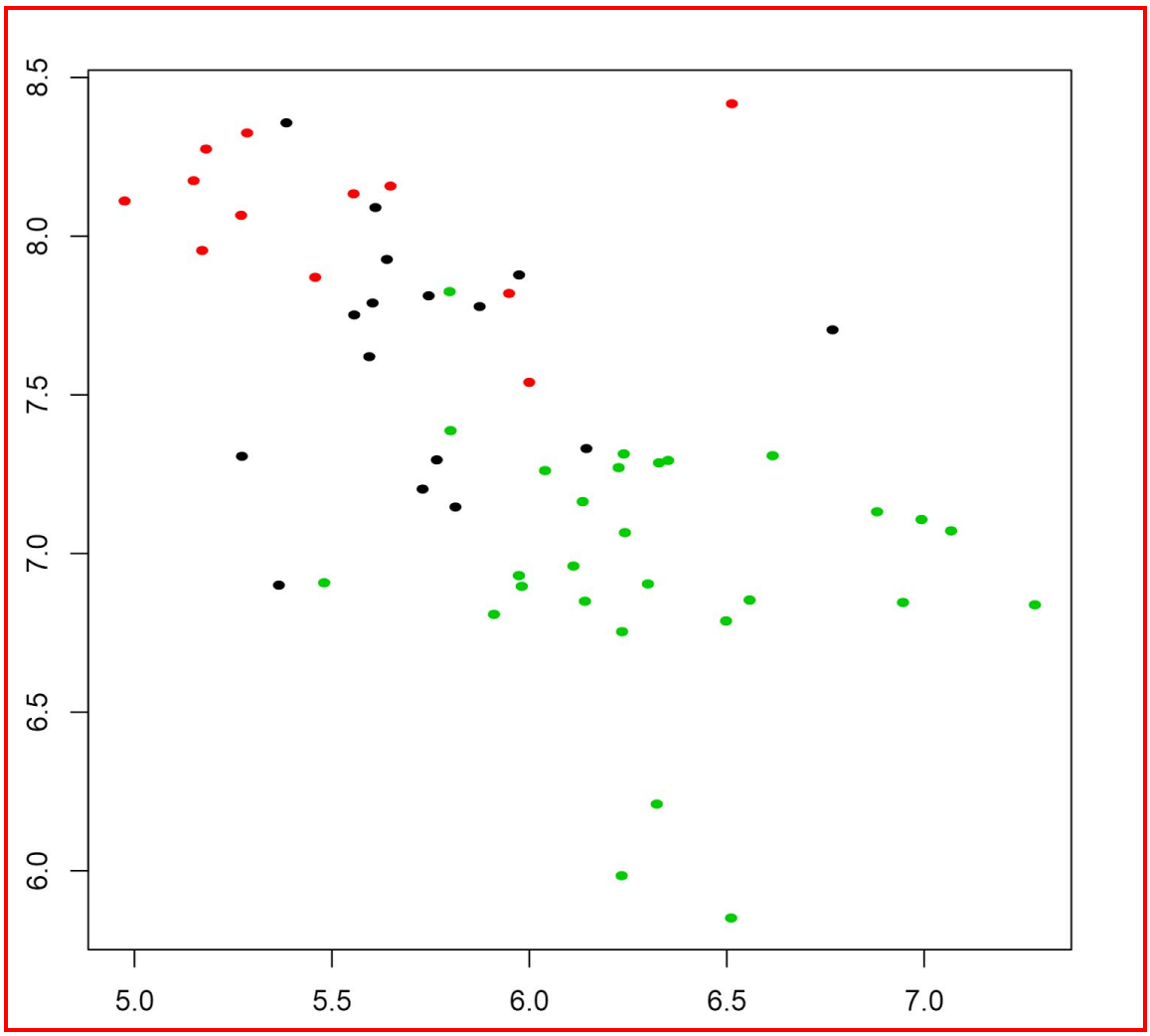
O *pam* (do inglês *partitioning around medoids*), assim como o *kmeans,* é um método não supervisionado de análise de cluster, também categorizado como método de particionamento (como indicado no próprio nome do método). Porém, por ser do tipo que seleciona os *medoids*, ele escolhe um ponto do dataset para definir como os centros, diferentemente do kmeans que procura a média dos pontos como centros. Além disso, o pam é um método mais robusto do que o kmeans em relação à *outliers* e *ruído.* Como vamos explicar mais a frente, assim como o hclust, pam também pode aceitar uma matriz de dissimilaridades.

PAM: k=2



Notamos que assim como no kmeans e mesmo o pam sendo mais robusto, existe certo overlap de amostras entre os dois grupos. Isso se deve ao fato de que o perfil de algumas amostras podem ser mais "próximas" das amostras do outro grupo. Porém, existe uma divisão um pouco mais clara de dois grupos do que aquele apresentado pelo kmeans com a mesma quantidade de centros.

PAM: k=3



Com k = 3 os clusters passam a se misturar mais, o que nos leva a conclusão de que para o nosso conjunto de amostras a separação mais clara é aquela fornecida por apenas dois clusters, mostrando que a separação entre pacientes com dengue com quadros diferentes não pode ser caracterizada pelo perfil transcricional do sangue.

Tabela comparativa de clusters: PAM e Kmeans

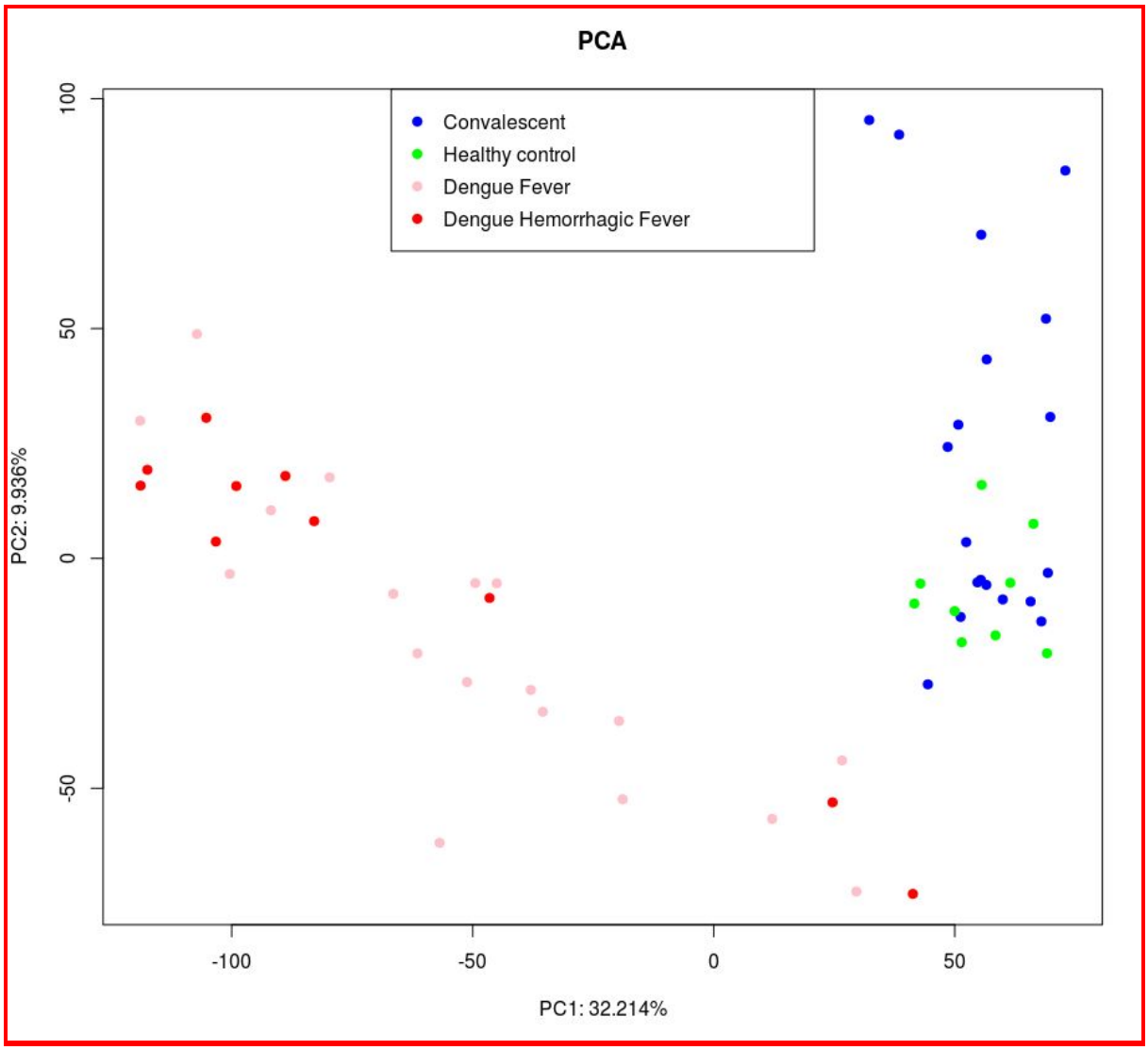
amostra	paciente	Pam	Kmeans
GSM1253028	Dengue Fever Patient	1	2
GSM1253029	Dengue Fever Patient	1	2
GSM1253030	Dengue Fever Patient	1	2
GSM1253031	Dengue Fever Patient	1	2
GSM1253032	Dengue Hemorrhagic Fever Patient	2	1
GSM1253033	Dengue Fever Patient	2	1
GSM1253034	Dengue Hemorrhagic Fever Patient	1	2
GSM1253035	Dengue Fever Patient	1	2
GSM1253036	Dengue Fever Patient	1	2
GSM1253037	Dengue Hemorrhagic Fever Patient	1	2
GSM1253038	Dengue Fever Patient	1	2
GSM1253039	Dengue Hemorrhagic Fever Patient	1	2
GSM1253040	Dengue Hemorrhagic Fever Patient	1	2
GSM1253041	Dengue Hemorrhagic Fever Patient	1	2
GSM1253042	Dengue Fever Patient	1	2
GSM1253043	Dengue Fever Patient	2	2
GSM1253044	Dengue Fever Patient	1	2
GSM1253045	Dengue Fever Patient	2	2
GSM1253046	Dengue Hemorrhagic Fever Patient	2	1
GSM1253047	Dengue Fever Patient	2	1
GSM1253048	Dengue Hemorrhagic Fever Patient	1	2
GSM1253049	Dengue Hemorrhagic Fever Patient	1	2
GSM1253050	Dengue Fever Patient	1	2
GSM1253051	Dengue Fever Patient	1	2
GSM1253052	Dengue Hemorrhagic Fever Patient	1	2
GSM1253053	Dengue Fever Patient	1	2
GSM1253054	Dengue Fever Patient	2	1
GSM1253055	Dengue Fever Patient	1	2
GSM1253056	Convalescent Patient	2	1
GSM1253057	Convalescent Patient	2	1
GSM1253058	Convalescent Patient	2	1
GSM1253059	Convalescent Patient	2	1
GSM1253060	Convalescent Patient	2	1
GSM1253061	Convalescent Patient	2	1
GSM1253062	Convalescent Patient	2	1
GSM1253063	Convalescent Patient	2	1
GSM1253064	Convalescent Patient	2	1
GSM1253065	Convalescent Patient	2	1
GSM1253066	Convalescent Patient	2	1
GSM1253067	Convalescent Patient	2	1
GSM1253068	Convalescent Patient	2	1
GSM1253069	Convalescent Patient	2	1
GSM1253070	Convalescent Patient	2	1
GSM1253071	Convalescent Patient	2	1
GSM1253072	Convalescent Patient	2	1
GSM1253073	Convalescent Patient	2	1
GSM1253074	Convalescent Patient	2	1
GSM1253075	Healthy control	2	1
GSM1253076	Healthy control	2	1
GSM1253077	Healthy control	2	1
GSM1253078	Healthy control	2	1
GSM1253079	Healthy control	2	1
GSM1253080	Healthy control	2	1
GSM1253081	Healthy control	2	1
GSM1253082	Healthy control	2	1
GSM1253083	Healthy control	2	1

Note que, com exceção de dois pacientes que apresentam apenas febre (GSM1253043 e GSM1253045) todos os demais pacientes se separaram em dois clusters diferentes quando k=2 (kmeans e pam colorem de formas contrárias). Isso também se nota olhando os gráficos mostrados acima.

PCA - Análise de componentes principais

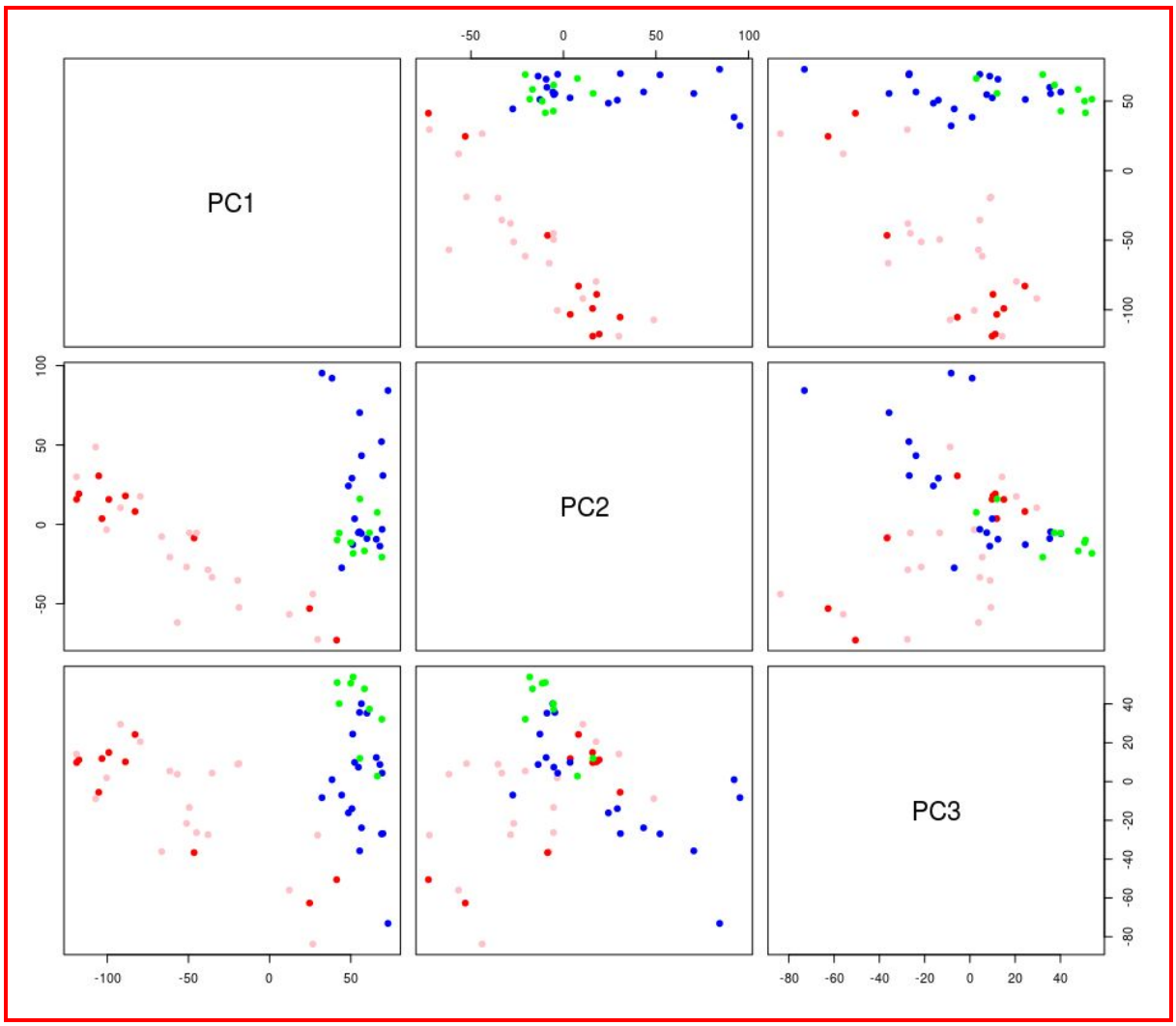
A análise de componentes principais (do inglês *principal componets analysis*), assim como o kmeans e o pam, é um método não-supervisionado que tenta manter a maior variabilidade *inner-group.* Dessa forma, se os grupos forem o suficientemente diferentes, o gráfico deve mostrar claramente uma divisão de grupos como vemos abaixo. É altamente visível que os pacientes convalescentes e saudáveis se misturam formando um grupo e os pacientes com dengue com febre e com dengue hemorrágica se misturam formando um outro grupo.

PC1 vs PC2



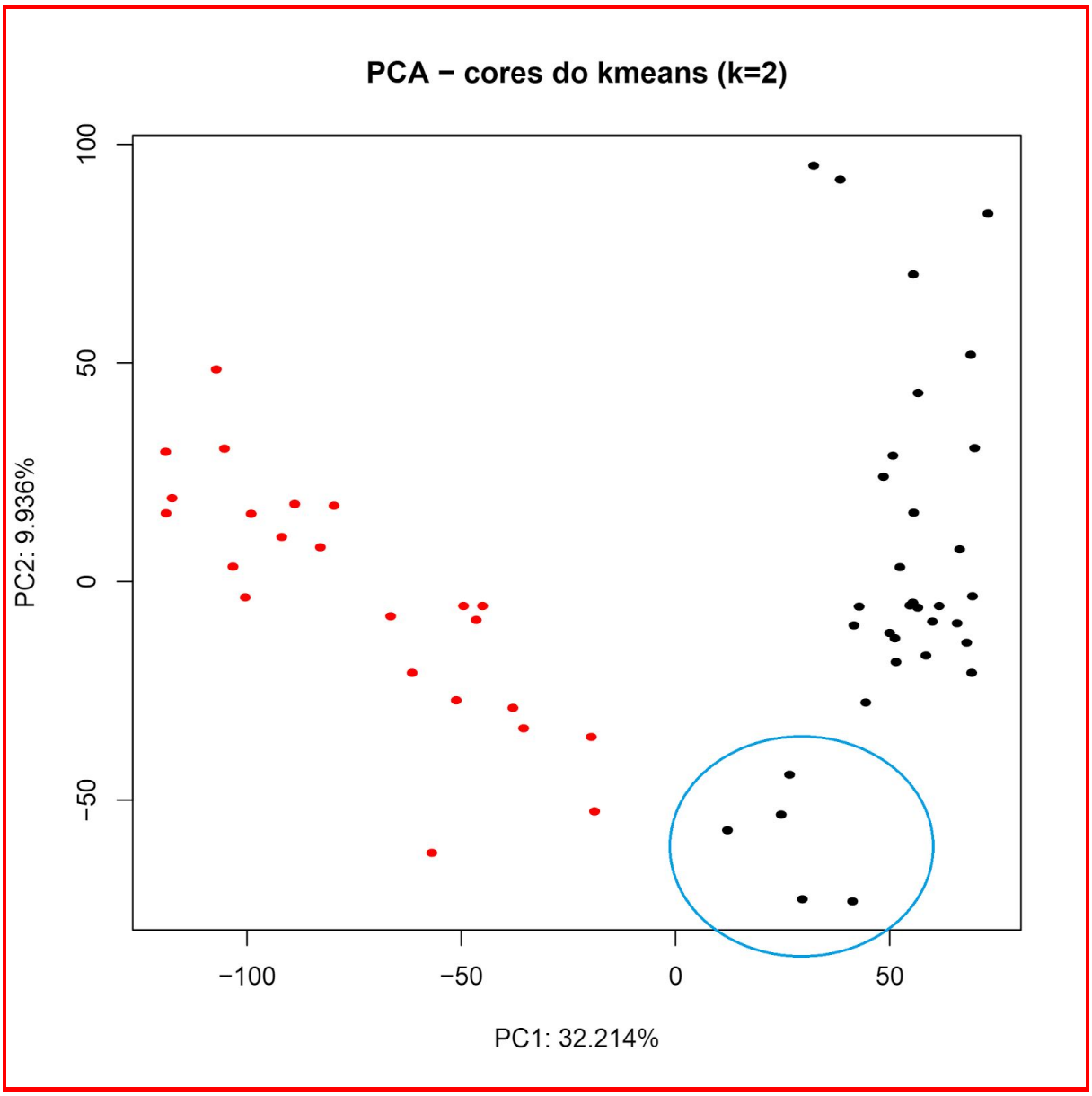
Note que o primeiro componente principal (PC) mostra aproximadamente 32% da variabilidade do nosso grupo de amostras. Apesar de ser um valor relativamente baixo, o PC1 ainda assim apresenta a maior parte da variabilidade das amostras analisadas aqui, enquanto o PC2 apresenta cerca de 10% e o PC3 aproximadamente 7%. A partir disso, os PCs apresentam menos que 5% da variabilidade chegando próximo do 1% cada no décimo terceiro PC.

PCA pareado



Agora podemos conferir se o grupamento de amostras que estamos vendo no PCA é o mesmo que encontramos, tanto no kmeans (k=2) quanto no pam (k=2), apenas colorindo o gráfico do PCA de acordo com as cores dos métodos de análise de cluster como abaixo.

PCA com cores de cluster (k=2)

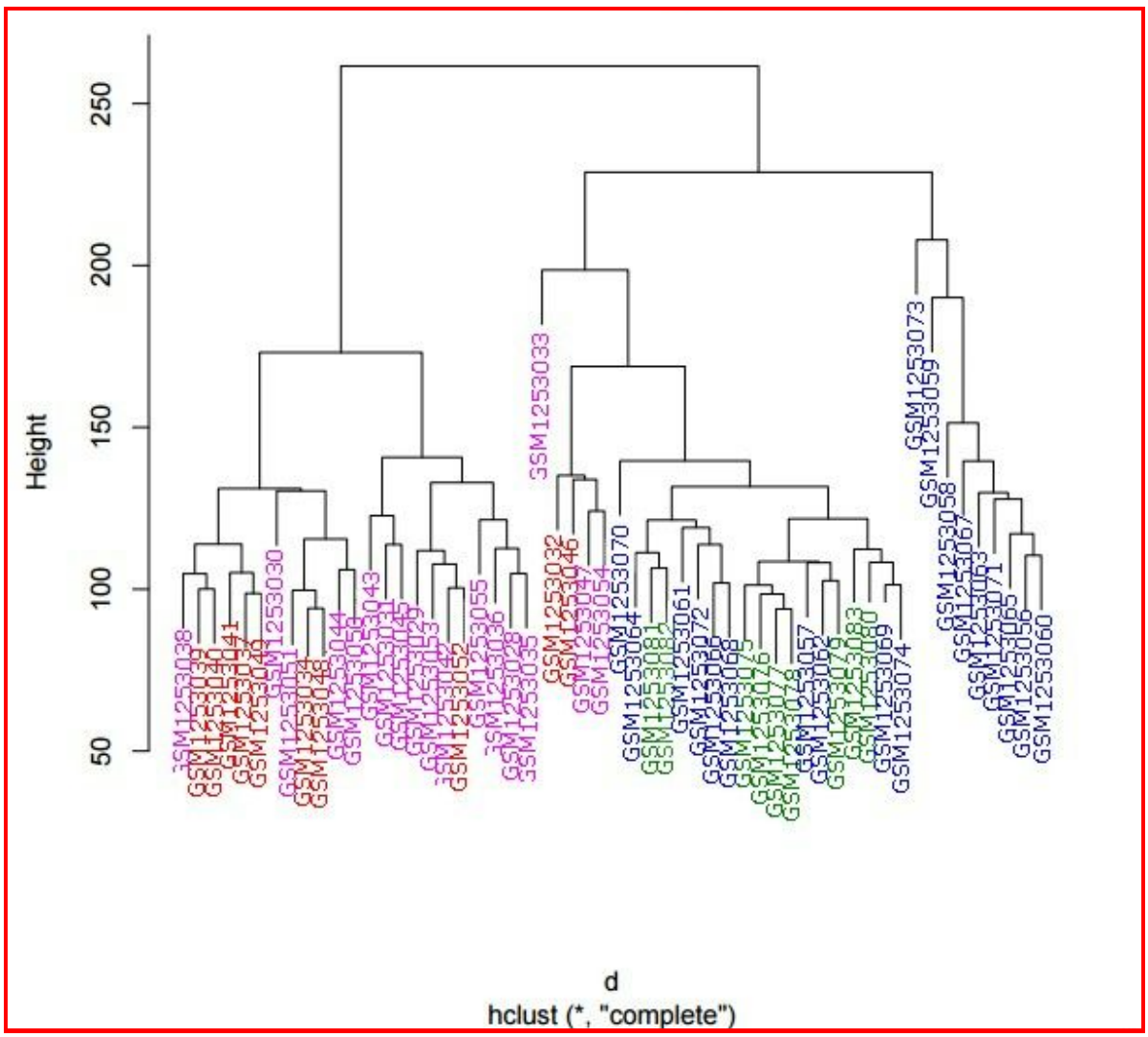


Note que as cinco amostras circuladas em azul no PCA 2D colorido por grupo de amostras poderiam ser confundidas como parte do grupo com características de pacientes com dengue. Enquanto, na verdade, a clusterização junta esses cinco pacientes com o grupo de pacientes convalescentes e controles. Isso vai ser visto claramente se for feito uma clusterização hierárquica que será mostrada mais a frente.

Dendrograma

Dendrograma são diagramas em formato de árvores que ilustram o arranjo de *clusters. N*o caso do nosso trabalho, os dendrogramas são hierárquicos e procuram definir a ordem dos clusters formados pelas amostras. Existem dois tipos de métodos para fazer a analise de cluster hierárquico: os métodos aglomerativos e os métodos divisivos. Basicamente diferem no "sentido" que definem os clusters. Os aglomerativos vão "de baixo para cima" enquanto os divisivos vão "de cima para baixo".

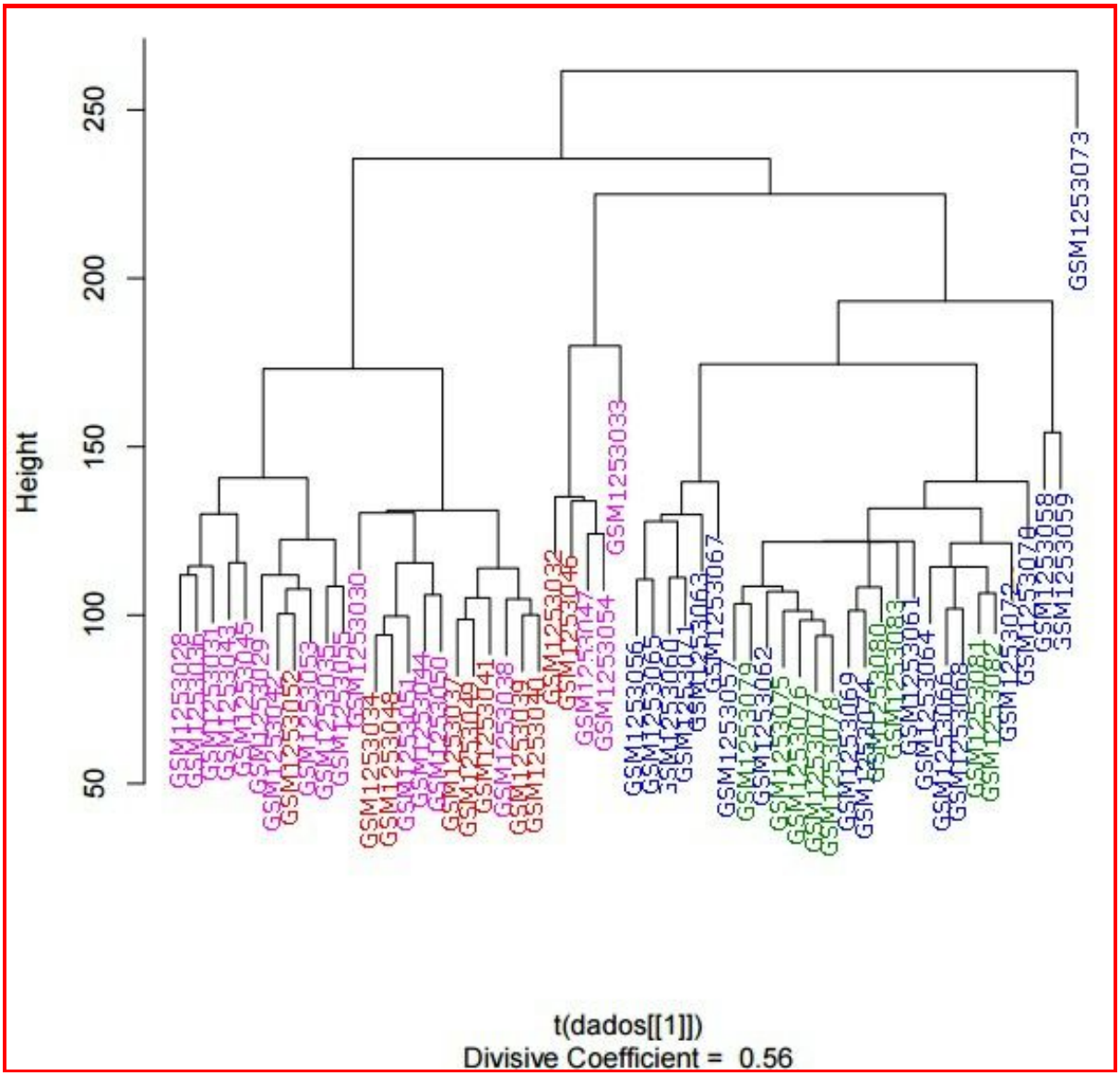
Dendrograma gerado pela função hclust



A função `hclust` é um método de clusterização aglomerativo. Ele começa colocando cada amostra em um *cluster* e vai agrupando as mais parecidas, até chegar ao topo da árvore. É importante ressaltar que `hclust` recebe uma matriz de distancias conhecida como matriz de dissimilaridades, e tais medidas indicam o quanto um par de amostras são parecidas. A matriz de dissimilaridades para esse caso foi calculada usando a função `dist` com parâmetros *default*. Com o método hclust é possível notar claramente a existência de dois clusters: um com as amostras de pacientes convalescentes e controles saudáveis (azul e verde respectivamente), e um outro cluster de pacientes com dengue e quadro febril e com dengue e quadro hemorrágico (rosa e vermelho respectivamente). Como mostramos no PCA com as cores do kmeans,

existem cinco amostras de pacientes com dengue que se aproximam mais do cluster com indivíduos saudáveis, aquelas circuladas em azul.

Dendrograma gerado pela função diana



Ao contrário da maioria dos métodos de clusterização hierárquica, a função DIANA (do inglês *DIvisive ANAlisys clustering*) como o próprio nome indica, é um método divisivo (de cima para baixo) de criação de clusters. Enquanto os métodos aglomerativos funcionam como o hclust, juntando as amostras mais semelhantes (menos distantes), o método diana, coloca todas as amostras em um grande cluster e vai dividindo, como descrito na documentação do método. Então o cluster vai sendo divido até cada cluster conter apenas uma amostra dessa forma, obtemos a árvore mostrada acima. Note que o método diana também separou as cinco amostras de pacientes com dengue que se mostram mais perto das amostras de pacientes convalescentes e de indivíduos normais.

Código

Requerimentos em R

Os requerimentos são os pacotes em R necessários para rodar tanto as funções desenvolvidas pela equipe quanto as funções encontradas em pacotes como MASS, cluster e outros. Eles podem ser encontrados dentro do arquivo `Requirements.R` que é o script em R descrito abaixo para instalar os pacotes necessários caso estes não estejam instalados, e carregá-los no *working space* do R possibilitando assim o uso de suas funções. É importante ressaltar que se estiver usando o Linux é preciso ter instalado duas bibliotecas para o funcionamento de alguns pacotes (XML e RCurl).

```
# Requirements for R

# list of necessary packages
packagesList <- c("XML", "RCurl", "downloader", "R.utils", "cluster", "ggplot")

# checks if the packages are installed if not put the non-istalled in a vector
toInstall <- packagesList[!(packagesList %in% installed.packages()[, "Package"])]

# installs non-istalled packages
if(length(toInstall)){
        install.packages(toInstall)
}

# finally loads packages
for(i in seq_along(packagesList)){
        library(packagesList[i], character.only=T)
}

# end of installation and loading process

# Requirements for Linux

# libcurl4-openssl-dev

# libxml2-dev
```

Funções auxiliares

São funções desenvolvidas pelo grupo para automatizar alguns dos passos necessários para fazer uma análise dos dados vindos do GEO, passos esses que normalmente são repetidos. Os automatizando, o analista pode se concentrar em outros aspectos da análise. Essas funções podem ser encontradas na pasta `funcoes/`. Dentro desta pasta temos uma série de arquivos, um arquivo `Functions.R`, que contém o código descrito abaixo, e um arquivo `.R`, para cada função onde podem ser encontrados comentários pertinentes para o entendimento do funcionamento de cada uma das funções, esses arquivos são nomeados baseados na função que contem.

```
# Função getLinkDownloadMatrix
getLinkDownloadMatrix <- function(gse){
    l <- list("null")
    for (i in seq_along(gse)){
        url <- paste0( "http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=", gse[i])
        parseURL <- XML::htmlParse(url)
        links <- XML::xpathSApply(parseURL, "//a/@href")
        link <- links[grep("matrix/", links)]
        mat <- RCurl::getURL(link)
        mat <- unlist(strsplit(mat, " "))
        mat <- mat[length(mat)]
        mat <- unlist(strsplit(mat, "\n"))
        link <- paste0(link,mat)
        aux <- c(link,mat)
        l[[i]] <- aux
    }
        return(l)
}

# Função downloadMatrix
downloadMatrix <- function(gseList){
        for (i in 1:length(gseList)){
                downloader::download(gseList[[i]][1],gseList[[i]][2])
        }
        return("Todos os downloads foram concluídos")
}

# Função findMatrixBegin
findMatrixBegin <- function(mat){
    x <- readLines(con = mat)
    lineNum <- grep("series_matrix_table_begin",x)
    return(lineNum)
}

# Função readMyData
readMyData <- function(gse){
    tally <- list("null")
    for (i in seq_along(gse)){
        files <- list.files(pattern = gse[i])
        R.utils::gunzip(files, ext="gz")
        files <- list.files(pattern=gse[i])
        print(files)
        x <- findMatrixBegin(files)
        data <- read.table(file = files, header = T, skip = x, fill = T, blank.lines.skip = T)
        data <- na.omit(data)
        rownames(data) <- data$ID_REF
        data <- data[,-1]
        assign(gse[i], data)
```

```
        tally[[i]] <- data
    }
    names(tally) <- gse
    return(tally)
}

# Função doMeta
doMeta <- function(gse){
    l <- list("null")
    for (j in seq_along(gse)){
        x <- readLines(con = paste0("http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=",
                       gse[j]))
        gsmIndex <- grep("GSM", x)
        gsm <- strsplit(x=x[gsmIndex], split='">')
        patient <- strsplit(x=x[gsmIndex + 1], split='">')
        for (k in seq_along(gsmIndex)){
            patient[[k]] <- patient[[k]][2]
            patient[[k]] <- substr(patient[[k]], start=1, stop=nchar(patient[[k]])-5)
            gsm[[k]] <- substr(gsm[[k]][length(gsm[[k]])], start=1, stop=10)
        }
        gsm <- unlist(gsm)
        patient <- unlist(patient)
        l[[j]] <- cbind(gsm,patient)
    }
    names(l) <- gse
    return(l)
}

# Função doColourPalette
doColourPalette <- function(df, type = "none", col = "black"){
    if (type == "none"){
        return("Você esqueceu de informar a categoria, corrija o codigo e rode novamente")
    }
    else {
        if (col != "black" && length(col) == length(type)){
            df$col = "black"
            for (i in 1:length(type)) {
                df$col[grep(type[i], df[,2])] = col[i]
            }
            return(df)
        }
        else if (col != "black" && length(col) != length(type)){
            return("A quantidade de cores não condiz com a quantidade de categorias\nCorrija um dos
vetores e rode novamente! :)")
        }
        else {
                df$col = col
                col = sample(colours(),length(type))
                for (i in 1:length(type)) {
                    df$col[grep(type[i], df[,2])] = col[i]
                }
                return(df)
            }
        }
    }
```

Script principal

É o código que mostra o uso das funções desenvolvidas pela equipe e algumas funções implementadas pelos desenvolvedores da linguagem R para obter as analises e gerar os gráficos mostrados anteriormente nesse trabalho. Este código pode ser encontrado na pasta principal do trabalho com o nome de `script.R` e é uma versão mais simplificada do que o texto encontrado em `mainCode.Rmd`, tendo em vista que não contem comentários explicativos.

```
# script of main code
# note that the study not specified to facilitate change in the code

source("Requirements.R")
source("funcoes/Functions.R")

gse <- c("GSE51808")
link <- getLinkDownloadMatrix(gse)
downloadMatrix(link)
dados <- readMyData(gse)
# dados[[1]][1:6,1:3]

categoria <- c("Convalescent", "Healthy control", "Dengue Fever", "Dengue Hemorrhagic Fever")


metadados <- doMeta(gse)
dfMeta <- as.data.frame(metadados[[1]])
#coloring <- sample(colours(),4)
coloring <- c("blue", "green", "pink", "red")
dfMeta <- doColourPalette(dfMeta, categoria, coloring)
#dfMeta <- doColourPalette(dfMeta, categoria)


t.dados <- t(dados[[1]])

# plots
# to generate the pdfs uncomment the line above and the one under the plot line

# kmeans
km <- kmeans(t.dados, centers=2)
# pdf("kmeans2centers.pdf")
plot(t.dados,col=km$cluster, pch=19, xlab=NA, ylab=NA)
# dev.off()


km3 <- kmeans(t.dados, centers=3)
# pdf("kmeans3centers.pdf")
plot(t.dados,col=km3$cluster, pch=19, xlab=NA, ylab=NA)
# dev.off()

# pam
p <- cluster::pam(t.dados, k=2)
# pdf("pam2centers.pdf")
plot(t.dados,col=p$clustering, pch=19, xlab=NA, ylab=NA)
# dev.off()


p3 <- cluster::pam(t.dados, k=3)
# pdf("pam3centers.pdf")
plot(t.dados,col=p3$cluster, pch=19, xlab=NA, ylab=NA)
# dev.off()

# diana
di <- cluster::diana(t.dados)
# pdf("diana.pdf")
plot(di)
# dev.off()

# hclust
```

```r
d <- dist(t.dados)
hc <- hclust(d)
#pdf("hclust.pdf")
plot(hc)
#dev.off()

# pca
pca <- prcomp(as.matrix(t.dados), cor=T, scale=F)
#pdf("pca-pairs-1to3.pdf")
pairs(pca$x[,1:3], col=dfMeta$col, pch=19)
#dev.off()

# pdf("pca-1e2.pdf")
plot(pca$x, col=dfMeta$col, pch=19, main = "PCA",
    xlab=paste0("PC1: ", summary(pca)$importance[2,1]*100, "%"),
    ylab=paste0("PC2: ", summary(pca)$importance[2,2]*100, "%"))
legend( "top", pch=rep(19,length(coloring)), col=coloring, legend=categoria)
# dev.off()

# pdf("pca-1e2-cluster-k2.pdf")
plot(pca$x, col=km$cluster, pch=19,
    main = "PCA - cores do kmeans (k=2)",
    xlab=paste0("PC1: ", summary(pca)$importance[2,1]*100, "%"),
    ylab=paste0("PC2: ", summary(pca)$importance[2,2]*100, "%"))
# dev.off()

# ploting pca usando o ggplot
dfMeta2 <- dfMeta
dfMeta2$Species <- NA


for (i in 1:length(categoria)) {
    dfMeta2$Species[grep(categoria[i], dfMeta2[,2])] = categoria[i]
}


dataset <- data.frame(species = dfMeta2[,"Species"], pca = pca$x)


prop.pca <- pca$sdev^2/sum(pca$sdev^2)


p2 <- ggplot(dataset) +
geom_point(aes(pca.PC1, pca.PC2, colour = species,
    shape = species), size = 2.5) +
  labs(x = paste("PC1 (", scales::percent(prop.pca[1]), ")", sep=""),
       y = paste("PC2 (", scales::percent(prop.pca[2]), ")", sep=""))

# pdf("pca_with_ggplot.pdf")
plot(p2)
# dev.off()

# gerando apendices
write.table(x=dfMeta, # dado a ser escrito
            append=F, # boleano: o arq ja existe escreve linhas extras(T) ou sobrescreve(F)
            sep="\t", # o tipo do separador, nesse caso \t indica tab
            col.names=T, # boleano para os nomes das colunas
            file="tabelaDeCores.txt", # nome do arquivo
            row.names=F, # boleano para os nomes das linhas
            quote=F) # boleano para presenca de aspas

write.table(x=km$cluster, append=F, sep="\t", col.names=F,
            file="kmeans_k2_clusters.txt", row.names=T, quote=F)

write.table(x=p$clustering, append=F, sep="\t", col.names=F,
            file="pam_k2_clusters.txt", row.names=T, quote=F)
```

Referências

https://cran.r-project.org

github.com/jtemporal/recDePadroes2015

1: Kwissa M, Nakaya HI, Onlamoon N, Wrammert J, Villinger F, Perng GC, Yoksan S, Pattanapanyasat K, Chokephaibulkit K, Ahmed R, Pulendran B. Dengue virus infection induces expansion of a CD14(+)CD16(+) monocyte population that stimulates plasmablast differentiation. Cell Host Microbe. 2014.

https://tgmstat.wordpress.com/2014/01/15/computing-and-visualizing-lda-in-r/

http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

http://www.ncbi.nlm.nih.gov/geo/

Apêndices

A - Acesso ao GitHub

Todos os códigos, arquivos, gráficos e outras informações necessárias podem ser encontradas no *GitHub* através do link https://github.com/jtemporal/recDePadroes2015/. Caso tenha o `git` instalado no seu computador, o repositório pode ser clonado através do terminal pelo seguinte comando:

```
git clone https://github.com/jtemporal/recDePadroes2015.git
```

A estrutura de árvore dos diretórios e subdiretórios é a seguinte:

```
.
└── recDePadroes2015
    ├── apendices
    ├── funcoes
    └── plots

4 directories, 32 files
```

No diretório `apendices/` podem ser encontrados os apendices descritos abaixo; em `funcoes/` são encontrados os arquivos descritos no item *Códigos*, subitem *Funções auxiliares* desse trabalho; em `plots/` podem ser encontrados todos os arquivos *pdf* e *png* que contém os gráficos apresentados e discutidos nesse trabalho. Os demais códigos e scripts podem ser encontrados na pasta `recDePadroes2015/` (e não estão mostrados aqui). Nós recomendamos fortemente que os códigos sejam utilizados a partir da clonagem do repositório, tendo em vista que a versão disponível no GitHub não apresenta bugs.

B - Tabela Series Matrix

Versão não comprimida do arquivo Table Matriz Series baixado do banco de dados GEO.

Nome do arquivo: GSE51808_series_matrix.txt

C - Tabela de cores

Arquivo de texto que contém a definição de cor de cada amostra, foi usada para colorir os gráficos de PCA e os Dendrogramas.

Nome do arquivo: tabelaDeCores.txt

D - Tabela de clusters

Tabela apresentada nos Resultados desse trabalho onde estão descritos os clusters definidos pelas funções pam e kmeans.

Nome do arquivo: tabelaDeClusters.xlsx

E - Videos dos pcas em 3D

Como enviar o vídeo para o youtube demorou mais do que o esperado, os arquivos podem ser encontrados no repositório do GitHub, assim como também o link direto para os vídeos.