HIV and tuberculosis co-infection: peripheral blood mononuclear cell

Trabalho produzido por:

- Antonio Dias
- Emanuel Lima Oliveira
- Joao Sequeira

Introducao

O dataset provem do artigo *HIV and tuberculosis co-infection: peripheral blood mononuclear cell* e pode ser consultado e descarregado a partir desta hiperligacao (http://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS4786).

O estudo consiste na analise de celulas mononucleares do sangue periferico em pacientes infectados com VIH, sendo alguns portadores de tuberculose (co-infecao). Colocando como hipotese de que a co-infecao e uma das causas de morte eminente em individuos previamente portadores do virus VIH, esta analise visa descobrir um certo padrao molecular que possa distinguir pacientes co-infectados de pacientes mono-infectados com VIH a partir de assinaturas moleculares.

Metodologia

> Importacao de bibliotecas usadas no nosso script R

```
source("https://bioconductor.org/biocLite.R")
biocLite("genefilter")
biocLite("illuminaHumanv4.db")
library(Biobase)
library(GEOquery)
library(genefilter)
library(matrixStats)
library(gplots)
library(limma)
library(class)
library(nnet)
library(rpart)
library(illuminaHumanv4.db)
```

Carregamento e pre-tratamento dos dados

De forma a efetuar um preprocessamento dos dados, foi executado o carregamento do dataset para o ambiente (formato soft.gz) e, posteriormente, guardado no diretorio de trabalho R. Seguidamente, o dataset foi sujeito a certos comandos com os objetivos de obter informacoes basicas para uma contextualizacao e a criacao de um *expression set* capacitado para manipulacao de dados.

```
GDS4786=getGEO(filename='GDS4786.soft.gz')

Meta(GDS4786)$title
Meta(GDS4786)$description
Meta(GDS4786)$type
Meta(GDS4786)$sample_count
Meta(GDS4786)$sample_organism
Meta(GDS4786)$sample_type
Meta(GDS4786)$feature_count

eset=GDS2eSet(GDS4786, do.log2=TRUE) #expression set contendo dados
```

Do conjunto de linhas demonstradas previamente, foram obtidos varias informacoes: * titulo do artigo; * breve descricao (summary); * tipos de conjuntos de dados **HIV** e **HIV and TB**, sendo declarados como *infection*; * o numero, organismo e tipo da amostra.

Com base nestas informacoes, verifica-se que **44** amostras de correspondentes aos individuos do genero *Homo sapiens*, metade é atribuido para cada conjunto de dados com uma quantidade de genes (features) para analise de **47323**.

Apos constatar a existencia de um elevado numero de entradas como valor nulo *NA*, tomou-se a decisao de remove-los para obter como resultado uma matriz contendo apenas valores numericos do dataset.

```
exp=exprs(eset)
data=exp[complete.cases(exp),] #matriz de valores numericos
```

Foram reduzidas o numero de *features* para **15529**, sendo respetivamente o profiling de cada individuo em analise.

Apesar da reducao de amostra, tornou-se um pouco impraticavel analisar elevada quantidade de dados devido a possiveis problemas de sobre-ajustamento. Com um reduzido ganho positivo, efetuou-se uma filtragem *flat patterns*. Deste modo, a amostra contem apenas genes cujo desvio padrao de expressao e maior do que duas vezes a mediana dos desvios padroes de todos os genes.

```
sds=rowSds(data)
m=median(sds)

dataf=data[sds >= 3*median(sds), ] #matriz de dados ap?s filtragem
```

Assim, foram reduzidas o numero de *features* para **206**. Desta forma, fica apenas enquadrado o grupo mais significativo dos dados, nao e comprometida a fidelidade e o ganho em facilidade de analise aumenta.

Analise do dataset e Resultados

Expressao diferencial

Para analisar a expressao genetica utilizou-se o package *limma*, e efetuou-se o ajuste em funcao do teorema de **Bayes**, obtendo resultados de ajuste para os genes mais influentes. Do mesmo modo recolheu-se informacao detalhada sobre cada um desses genes, assim como as suas funcoes. Posteriormente, ordenou-se os genes que apresentavam p-values significativos (*rank*).

```
f <- factor(as.character(eset$infection))
design <- model.matrix(~f)
fit <- eBayes(lmFit(eset,design))
diff = topTable(fit, coef=2)
diff
unlist(mget(rownames(fit),illuminaHumanv4SYMBOL))

rankFit = order(fit$p.value)
p20Fit = rank[1:20]
fit$p.value[p20Fit]
g = featureNames(eset[p20Fit])
unlist(mget(g,illuminaHumanv4SYMBOL))
```

Clustering

Para visualmente efetuar uma procura de clusters onde possam indicar *hotspots* de genes, ou seja, apresentarem uma elevada diferenca de valores entre os dois grupos de estudo, foi gerado um *heatmap*. A escolha do metodo *heatmap.2* do qual pertencente a biblioteca *gplots*, permitiu um controlo superior sobre os esquema de apresentacao e cores. Como se pretendia fazer calculos da distancia sob a influencia do coeficiente de correlacao de pearson e em funcao da media de valores, ambas as funcoes foram criadas para uso como argumentos na geracao do grafico. De forma a reduzir o esforco de re-calculo do *heatmap*, sao fornecidos de seguida os codigos para a sua obtencao e imagem resultante.

```
dist.pear = function(x) as.dist(1-cor(t(x)))
hclust.ave = function(x) hclust(x, method="average")
heatmap.2(dataf, col=redgreen(75), labRow = F, distfun=dist.pear, hclustfun=hclust.ave, scale
="row", key=TRUE, symkey=FALSE, density.info="none", trace="none", cexRow=0.5)
```

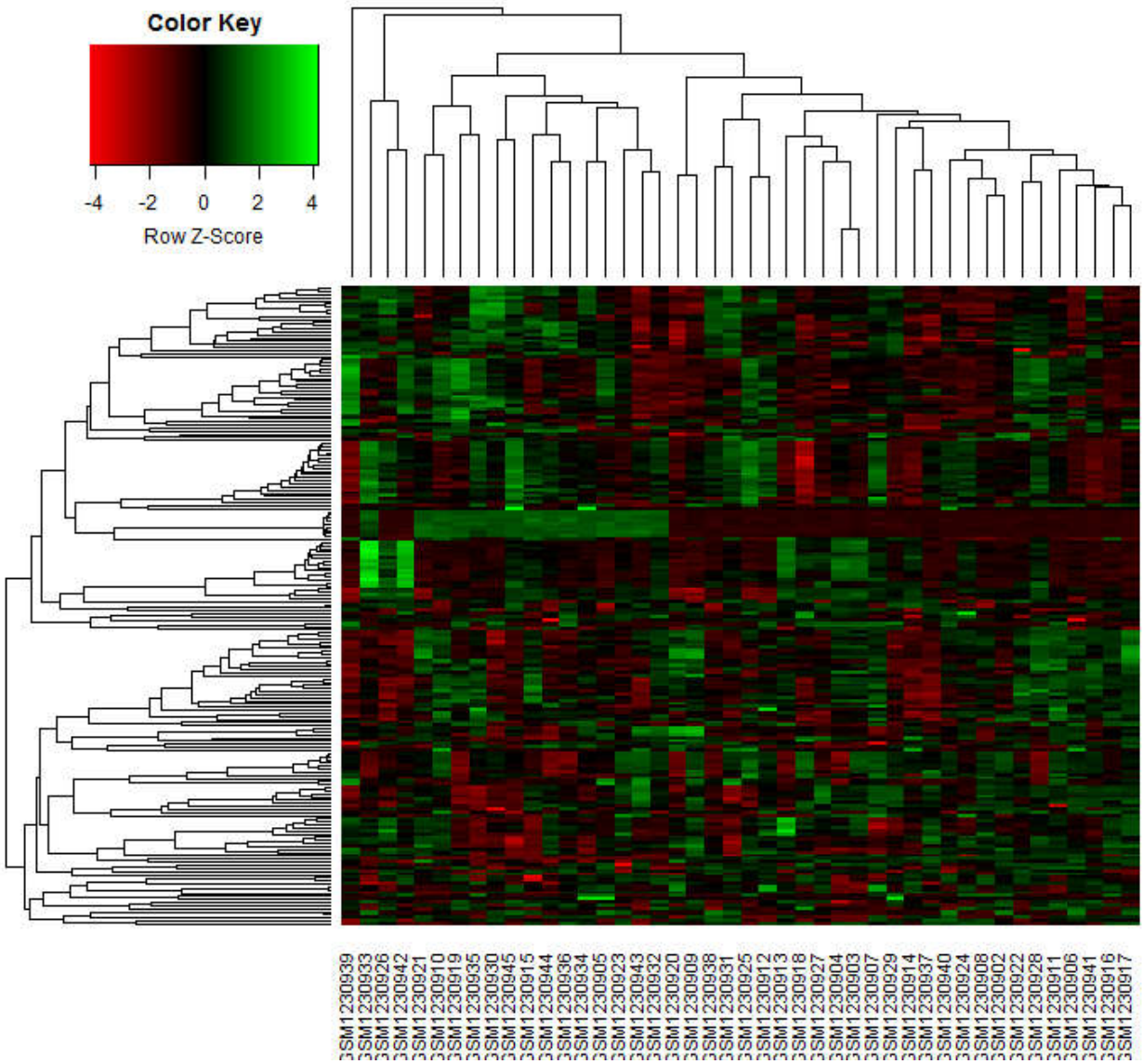

Fig 1: cluster heatmap.2

Modelos de previsao

De forma a trabalhar com modelos preditivos, a matriz de dados foi transposta. As linhas passaram a representar os diferentes individuos e as colunas os diferentes genes. Foi igualmente adicionada a coluna **Infection**, representando a condicao inerente de cada individuo. Neste novo data-frame foram obtidas as amostras de treino e de teste para posteriores estudos preditivos.

```
tdados = t(dataf)
dados = data.frame(tdados,Infection = eset$infection)

set.seed(9001)
inTrain = createDataPartition(y = dados$Infection, p = 0.7, list = F)
trainData = dados[inTrain,]
testData = dados[-inTrain,]
```

K-means

O modelo obtido por k-means previu com sucesso 42% dos casos.

```
knn_prev = knn(trainData[,1:206],testData[,1:206],eset$infection[inTrain])
table(knn_pred, eset$infection[-inTrain])
sum(knn_prev==testData$Infection)/dim(testDataf)[1]
```

Redes neuronais

O modelo obtido por redes neuronais previu com sucesso 50% dos casos. De notar que previu que todos os individuos nao se encontravam infetados com tuberculose.

```
ann = nnet(trainData$Infection[1:length(trainData$Infection)]~.,trainData,size=3)
ann_prev = predict(ann, testData,type="class")
table(ann_prev,eset$infection[-inTrain])
sum(ann_prev==testData$Infection)/dim(testData)[1]
```

Arvores de decisao

O modelo obtido por Arvores de decisao previu com sucesso 67% dos casos. Foi construida uma Arvore de decisao baseada no valor do unico gene “ILMN_2176063” com a decisao de valores de expressao desse gene abaixo de 9.328 para a folha de “HIV” e valores superiores a 9.328 de decisao para a folha de “HIV and TB”.

```
arv = rpart(trainDataf$Infection[1:length(trainDataf$Infection)]~.,trainDataf)
plot(arv,uniform=T,branch=0.4,margin=0.1,compress=T)
text(arv,use.n=T,cex=0.9)
cprevistas = predict(arv, testDataf, type="class")
table(cprevistas,eset$infection[-inTrain])
sum(cprevistas==testDataf$Infection)/dim(testDataf)[1]
```

Resultados e Discussao

Expressao diferencial

Pelo conjunto de resultados obtidos, é possivel verificar varios p-values correspondentes a genes. Em micro-arrays, p-values sao considerados como uma estimativa de frequencia com que os dados sao observados ao acaso isoladamente. Por exemplo, para um p-value de 0.05 e tipicamente complicado indicar significancia, dado que a estimativa de observar dados ao acaso e de 5%. No entanto, para o caso de ~15000 genes determinados no micro-array e sendo que **206** genes foram previamente identificados como significativos atraves de um desvio padrao significativo, poderao conter genes que nao se identificam como significativos por nao apresentarem diferencas entre os dois grupos experimentais (VIH/VIH e TB). Algumas solucoes podem ser utilizadas como restringir o criterio de p-values e/ou utilizar a *correcao de Bonferroni* para ajustar p-values em proporcao com o numero de testes paralelos envolvidos. No entanto, para evitar restantes falsos-positivos, fixam-se os p-values. Assim, foram obtidos em rank os 20 possiveis genes que apresentam maior significancia para o estudo.

```
> unlist(mget(g,illuminaHumanv4SYMBOL))
ILMN_1794187 ILMN_1669966 ILMN_3289262 ILMN_1698307 ILMN_2301955 ILMN_2046024 ILMN_2057826 ILMN_2174296
     "FBXL3"     "NDUFS7"      "PNRC2"       "DBNL"      "THAP1"     "DUSP11"       "PHF3"     "DNAJC2"
ILMN_1772658 ILMN_2208373 ILMN_1766718 ILMN_3235642 ILMN_2415439 ILMN_1770667 ILMN_2229205 ILMN_3291921
      "ICE1"    "TMEM164"     "LYSMD3"      "PRKDC"       "NAE1"       "HECA"      "CENPC"   "C11orf58"
ILMN_3288830 ILMN_2088612 ILMN_1697694 ILMN_1687036
     "EIF4E"       "XPO4"    "ATP6AP1"     "MRPL47"
```

Fig. 2: 20 genes identificados no rank.

Clustering

Como se pode verificar pelo *heatmap* existem 2 clusters correspondentes a um pequeno numero de genes, que demostram uma clara diferenca num reduzido numero de genes (aproximadamente 10), entre individos que se encontram infectados apenas com VIH em comparacao com os individos co-infectados.

Modelos de previsao

K-means e redes neuronais

Ambos os modelos de k-means e redes neuronais nao foram eficientes na previsao da condicao dos individuos de teste.

Arvores de decisao

O modelo de Arvores de decisao foi escolhido como o mais eficaz a prever a condicao dos individuos de teste. No entanto, nao e um bom modelo para estudar este tipo de dados, dado que apenas toma em consideracao uma variavel para decidir entre duas hipoteses, ignorando as varias centenas de outras variaveis. Por sua vez, a variavel escolhida torna-se logicamente como a melhor opcao, pois representa uma sonda que tem como gene alvo o gene FCGR1A, do qual codifica o fragmento Fc de Imunoglobulina G, uma biomolecula muito usada na identificacao de tuberculose.