

Mestrado integrado em Engenharia Informática

Departamento de Engenharia Informática

Unidade Curricular de Computação Natural

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Extração do Conhecimento de Bases de Dados Biológicas:

Efeitos do abuso da cocaína no cérebro

      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Autores

Tiago Araújo - A71346

José Bastos - A74696

José Faria - A67638

# Introdução

A dependência em drogas exerce um peso elevado nos indivíduos afetados e também na sociedade como um todo. A persistência da necessidade de drogas e o risco de recaída, características de dependência, são pensadas estar associadas a alterações de longa duração na expressão genética neural resultando a partir de mudanças na transcrição e regulação de cromatina. Estes mecanismos constituem a “memória molecular” que contribui para o estado de dependência nas drogas.

Dos diferentes tipos de neurónios e circuitos implicados nos efeitos do abuso de drogas, os neurónios responsáveis pela síntese de dopamina no cérebro central, que inervam as regiões frontais do cérebro, são os que desempenham o maior papel. Estas células existem num rácio de 1 para 200,000 neurónios no cérebro humano, contudo a dopamina está criticamente envolvida na mediação entre os efeitos recompensadores do abuso de drogas e as respostas condicionadas a usos de droga anteriores. Para além disto, o abuso de droga pode levar a défices de transmissão de dopamina que contribuem para outras consequências adversas.

Apesar dos desafios associados com o seu uso, o estudo de cérebros humanos *postmortem* constitui um recurso único que pode ser usado para desenvolver novas ideias em relação a perturbações psicológicas complexas como a dependência em drogas. Uma melhor compreensão das mudanças associadas ao uso de cocaína irá permitir o desenvolvimento de biomarcadores e novas terapêuticas para o tratamento de dependência de drogas.

# 1. Dados

Para a análise em questão foram usadas as seguintes bibliotecas:

1. source("http://bioconductor.org/biocLite.R")
2. library(Biobase)
3. library(gtools)
4. library(GEOquery)
5. library(limma)
6. library(AnnotationDbi)
7. library(genefilter)
8. library(gplots)
9. library(e1071)
10. library(caret)
11. library(class)
12. library(nnet)
13. library(rpart)

**Código 1.** Bibliotecas

## 1.1. Carregamento e visualização dos dados

O identificador do estudo em questão é o “GDS5047” para tal foram lidos os dados e de seguida convertidos para o formato de ExpressionSet. Este processo pode ser visto no código abaixo.

1. gds5047 <- getGEO('GDS5047', destdir=".")
2. #Expression set
3. eset <- GDS2eSet(gds5047, do.log2=TRUE)
4. #Matriz de dados de expressao
5. exp=exprs(eset)

**Código 2.** Carregamento dos dados

Este conjunto de dados tem 49576 features (linhas) e 60 samples (colunas).

## 1.2. Meta dados

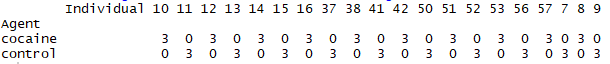
Os meta dados encontram-se organizados em quatro campos:

1. **sample** – indica o código da amostra.
2. **Agent** – indica a condição do individuo do qual a amostra foi recolhida (control, cocaine).
3. **Individual** – Indica o valor da amostra que foi recolhida.
4. **Description**.

Estes dados podem ser estudados com maior detalhe da seguinte maneira:

1. vars=pData(eset)
2. names(vars)
3. levels(vars$agent)
4. levels(vars$individual)
5. #Frequencia absoluta das amostras para as variaveis Agent,Individual
6. ftable(vars$agent, vars$individual, dnn = c("Agent", "Individual"))

**Código 3.** Visualização dos dados



**Tabela 1.** Frequência absoluta das amostras para as variáveis agent e individual.

# 2. Pré-Processamento

## 2.1. Remoção de dados omissos

1. #Remover NA
2. eset\_NA= eset[complete.cases(exp),]

**Código 4.** Remoção de dados omissos

Através da remoção de dados omissos obtivemos um dataset menor com 48761 features e 60 samples, ou seja, foram removidas 815 entradas com dados omissos.

## 2.2. Filtragem dos dados.

Como o ExpressionSet em questão não tinha anotação não foi possível filtrar retirar as sondas que correspondem a genes sem anotação. Foram testados dois filtros nos dados, o filtro de dados pelo rácio do máximo e mínimo e o filtro de dados pela. As funções responsáveis por cada um dos filtros encontram-se de seguida.

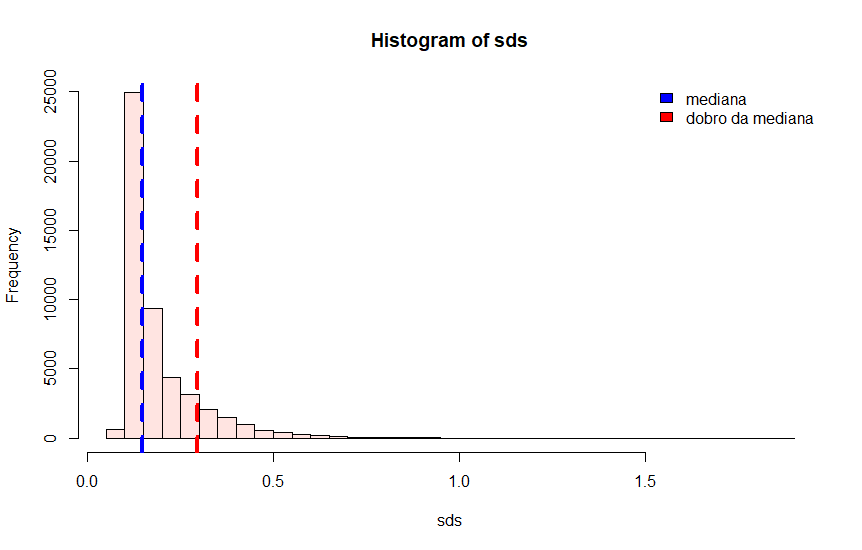
1. filter\_max<-function(dataset){
2. exp2=exprs(dataset)
3. maximos=apply(exp2,1,max)
4. minimos=apply(exp2,1,min)
5. vl=maximos/minimos > 2
6. esetmax=dataset[vl,]
7. return(esetmax)
8. }

**Código 5.** Filtro Rácio entre máximo e mínimo

Este filtro originou um resultado com 0 features e por isso foi descartado.

1. filter\_sd<-function(dataset){
2. exp2=exprs(dataset)
3. sds=rowSds(exp2)
4. m=median(sds)
5. hist(sds,breaks=50,col="mistyrose")
6. abline(v=m,col="blue",lwd=4,lty=2)
7. abline(v=m\*2,col="red",lwd=4,lty=2)
8. esetr = dataset[sds >= 3\*median(sds),]
9. return(esetr)
10. }
11. esetr=filter\_sd(eset\_NA)

**Código 6.** Filtro Mediana



**Figura 1.** Histograma do filtro de medianas

Este filtro originou um dataset com 1930 features e 60 samples o que foi uma redução substancial dos dados de estudo, o que irá tornar o processo de aprendizagem mais rápido na análise de Machine Learning. Para além disto a matriz de dados de expressão foi sujeita a um processo de **normalização** que tem efeitos muito positivos em Machine Learning (impede o escalonamento para números demasiados elevados no treino).

# 3. Análise de Expressão Diferencial e de Enriquecimento

Foram realizadas análises de expressão diferencial entre indivíduos com uso de cocaína e controlo. Estas análises foram realizadas usando a package limma, que usa modelos lineares para avaliar a expressão diferencial de experiências multifatoriais, sendo o método ideal para este tipo de análise.

Na análise de expressão diferencial usou-se o Expression Set completo, visto não ser possível extrair os genes anotados como mencionado anteriormente, com *p-value* *<* 0.05 e *Fold Change* *>* threshold, em que o fold é 1.4 e o threshold o logaritmo associado, para estar de acordo com a análise realizada no artigo. Nos resultados podem ser observados principalmente, o simbolo do gene e a sua função. São também calculados os genes subexpressos e sobreexpressos.

A análise de enriquecimento seria usada a package GOstats que permite a análise funcional dos genes de interesse, associando-os a termos de Gene Ontology(GO). Contudo, como este Expression Set **não inclui a anotação necessária**, esta operação não foi realizada.

## 3.1. Abuso de cocaína vs Controlo

Foi feita a criação do modelo linear:

1. lmfitagent<-function(dataset){
2. design= model.matrix(~dataset$agent)
3. fit=lmFit(dataset,design)
4. fit2=eBayes(fit)
5. diff=topTable(fit2,coef = 2,1000)
6. return(diff)
7. }
8. diff=lmfitagent(eset\_NA)
9. threshold = foldchange2logratio(1.4)
10. genes = diff[which(diff$adj.P.Val<0.05 & (diff$logFC > threshold | diff$logFC < -threshold)),]
11. genes=genes[,c(1,3,16,17,18,23:28)]
12. View(genes)

**Código 7.** Modelo linear

Foram considerados 39 genes diferencialmente expressos entre as duas condições. De seguida foram calculados os genes sobre expressos e sub expressos.

1. #Genes sobreexpressos:
2. genessobre = diff[which(diff$adj.P.Val<0.05 & (diff$logFC > threshold )),]
3. genessobre=genessobre[,c(1,3,16,17,18,23:28)]
4. View(genessobre)

**Código 8.** Genes sobre expressos.

Dos 39 genes mencionados anteriormente, 12 são sobre expressos no abuso de cocaína.

1. #Genes subexpressos:
2. genessub = diff[which(diff$adj.P.Val<0.05 & (diff$logFC < -threshold )),]
3. genessub=genessub[,c(1,3,16,17,18,23:28)]
4. View(genessub)

**Código 9.** Genes sub expressos.

Dos 39 genes mencionados anteriormente, 27 são sub expressos no abuso de cocaína.

Os resultados encontravam-se de acordo com os resultados propostos pela análise feita no artigo.

# 4. Clustering

Foi realizada uma análise de clustering hierárquico, utilizando todos os dados do Expression Set e os genes retirados da análise de expressão diferencial. Pelo cluster é possível perceber se existe um agrupamento de amostras de acordo com os valores de expressão, separando portanto, as amostras provenientes das duas condições. A matriz de distância utilizada foi a correlação de Pearson, indicada quando se verifica um padrao nos dados, e o método de *clustering* foi o *complete.*

Para além disso, foi feita uma outra análise de *clustering*, agrupando tanto as amostras como as sondas, ou seja, aplicando o clustering às colunas e às linhas. Este clustering é apresentado na forma de um heatmap, onde os valores de uma matriz são representados como cores.

Para a construção do *heatmap* foram necessárias as seguintes funções:

1. dist.fun = function(x) {
2. return (as.dist (1 - cor(t (x), method = "pearson")))
3. }
4. clust.fun = function (x) {
5. return (hclust (x))
6. }
7. color.map.tissue <- function(status) { if (status == "cocaine") "turquoise" else "chocolate1" }

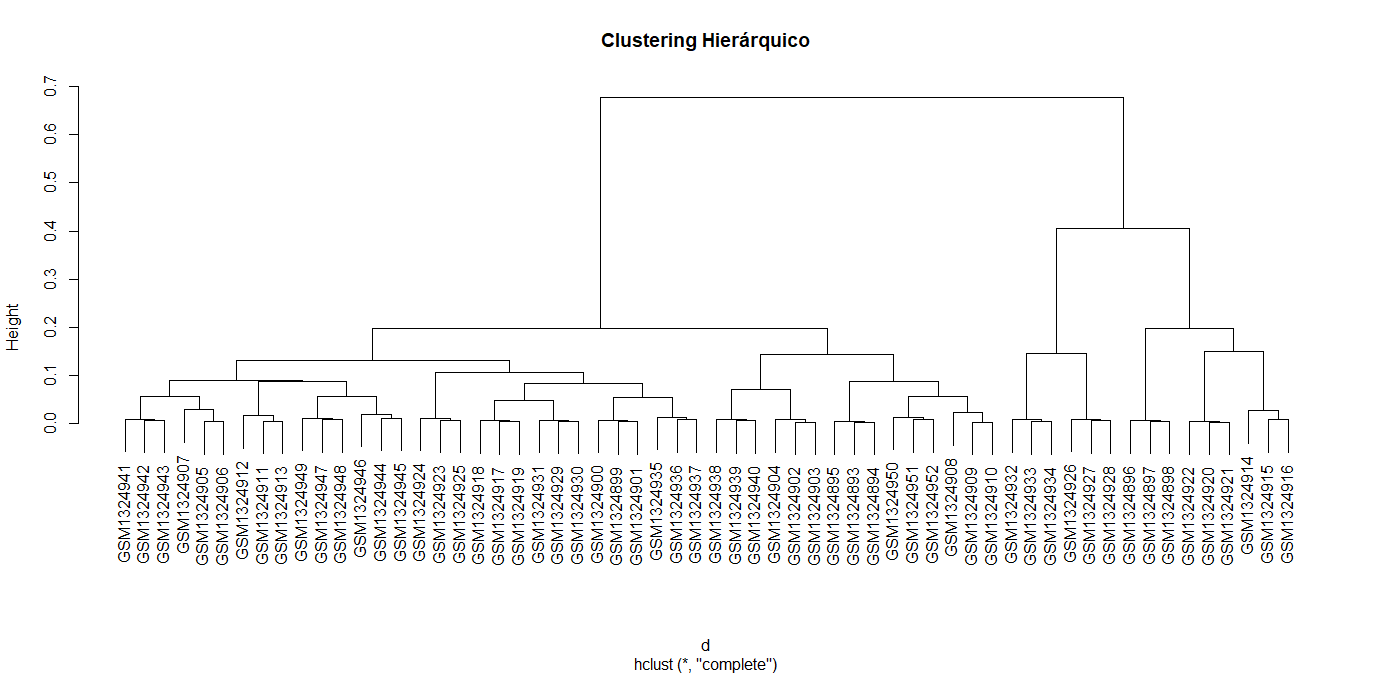
**Código 9.** Funções para o heatmap.

## 4.1. Clustering hierárquico

A função responsável pelo *clustering* hierárquico encontra-se de seguida tal como os seus resultados.

1. clusterhierarch<-function(dataset,res){
2. cl=exprs(dataset[res[,1],])
3. corr=cor(cl,method="pearson")
4. d=as.dist(1-corr)
5. cl.hier <- hclust(d)
6. plot(cl.hier, main = "Clustering Hierárquico")
7. }
9. clusterhierarch(eset\_NA,genes)

**Código 10.** Clustering hierárquico.



**Figura 2.** Resultado do clustering hierárquico.

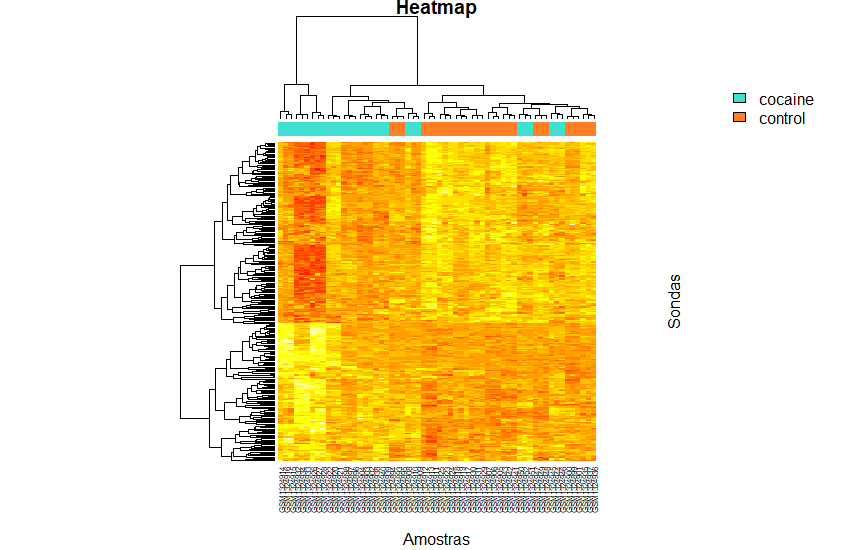
Pelo clustering verifica-se que é possível separar as amostras provenientes de indivíduos com abuso de cocaína das amostras de controlo em dois clusters. Desta forma, conclui-se que os genes correspondentes a amostras de indivíduos com abuso de cocaína apresentam um nível de expressão mais próximo entre si e mais afastado dos genes que provêm de amostras de controlo. Esta divisão era esperada, pois está concordante com os resultados da análise de expressão diferencial.

## 4.1. Heatmap

A função responsável pelo *heatmap* encontra-se de seguida tal como os seus resultados.

1. heatmapTop<-function(dataset,res){
2. cl=dataset[res[,1]]
3. tissuecolorsh <- unlist(lapply(cl$agent, color.map.tissue))
4. heatmap(exprs(cl),ColSideColors = tissuecolorsh, cexRow = 0.5, distfun = dist.fun, hclustfun = clust.fun, labRow = F, margins = c(7,7),
5. ylab = "Sondas", xlab = "Amostras", main = "Heatmap")
6. legend("topright",legend=c("cocaine","control"), bty="n",fill = c("turquoise","chocolate1"))
7. }
9. genes2 = diff[which(diff$adj.P.Val<0.05),]
10. genes2 =genes2[,c(1,3,16,17,18,23:28)]
11. heatmapTop(eset\_NA,genes2)

**Código 11.** Heatmap.



**Figura 3.** Resultado do heatmap.

Analisando o heatmap, verifica-se uma evidente separação das amostras com exceção de algumas amostras, tal como no cluster anterior. Para além disso, observa-se vários agrupamentos dos genes, que parecem estar agrupados consoante os seus níveis de expressão, já que se percebe uma grande porção com uma côr mais intensa (genes sobreexpressos) à esquerda do gráfico pertecente às com abuso de cocaina e uma porção com côr menos intensa (genes subexpressos) abaixo da anterior.

# 4. Análise preditiva

Para realizar a análise preditiva do conjunto de dados, optou-se por usar o package Caret, que permite simplificar o processo de criação de modelos preditivos, ao facilitar várias tarefas como a estimação de parâmetros. Foi então realizada a previsão do estado do individuo (cocaina, controlo). Os métodos de aprendizagem usados nesta análise incluem o método dos K-vizinhos mais próximos, árvores de decisão, máquinas de vetor de suporte (SVMs) e redes neuronais artificiais (ANN), sendo que foi escolhido o método de validação cruzada com 5 folds na seleção de modelos. Para esta análise, foi utilizado o conjunto de dados filtrado e com normalização (esetr). No caso das redes neuronais não foi usado o package Caret sendo usado o package nnet.

## 4.1. Análise dos K vizinhos mais próximos

1. model\_knn = train(t(exprs(esetr)), esetr$agent, method = "knn", trControl=trainControl("cv", number = 5))
2. pred\_knn = predict(model\_knn, t(exprs(esetr)))
3. mk1=confusionMatrix(pred\_knn, esetr$agent)
4. mk1$table; mk1$overal[1]

**Código 12.** KNN.

Este modelo obteve uma precisão de 80% de casos corretos.

## 4.2. Árvores de decisão

1. model\_tree = train(t(exprs(esetr)), esetr$agent, method = "rpart", trControl=trainControl("cv", number = 5))
2. pred\_tree = predict(model\_tree, t(exprs(esetr)))
3. mt2 = confusionMatrix(pred\_tree, esetr$agent)
4. mt2$table; mt2$overal[1]

**Código 13.** Arvore de decisão.

Este modelo obteve uma precisão de 98% de casos corretos.

## 4.3. Support Vector Machines

1. model\_svm = train(t(exprs(esetr)), esetr$agent, method = "svmLinear", trControl=trainControl("cv", number = 5))
2. pred\_svm = predict(model\_tree, t(exprs(esetr)))
3. ms3 = confusionMatrix(pred\_tree, esetr$agent)
4. ms3$table; ms3$overal[1]

**Código 14.** Support Vector Machines.

Este modelo obteve uma precisão de 100% dos casos corretos.

## 4.4. Redes neuronais artificiais

1. train = t(exprs(esetr[,11:50]))
2. test=t(exprs(esetr[,c(1:10,51:60)]))
3. ann = nnet(esetr$agent[11:50]~.,data.frame(train),size=3,MaxNWts=10000)
4. valores.prev.ann = predict(ann,data.frame(test),type="class")
5. valores.reais= esetr$agent[c(1:10,51:60)]
6. table(valores.prev.ann,valores.reais)
7. sum(valores.prev.ann == valores.reais)/length(valores.reais)

**Código 15.** Redes Neuronais Artificiais.

Este modelo obteve resultados bastantes inferiores, 20% dos casos, uma vez que em contraste com os outros modelos não foi treinado e testado com os mesmos dados, ou seja, fez se uma partição dos dados. Fazer a partição dos dados exige cuidado uma vez que os estados estão separados a meio (30 cocaína e 30 controlo) e portanto se apenas fossem escolhidos os últimos para teste não se teria dados de teste relativos à cocaína. O modo como os dados se encontram também dificulta a aprendizagem do modelo.

# 5. Conclusão

O abuso de drogas é considerado um dos problemas mais proeminentes da atualidade