

Metodo implementato per la tesi

Michele Fusaroli

2019

Il seguente metodo è stato sviluppato ricorrendo al linguaggio di programmazione R, versione 3.5.2 (2018-12-20).

Il materiale utilizzato è stato condiviso sulla piattaforma GitHub insieme agli script completi (con anche il codice per la pulizia dei database e l'analisi descrittiva), ai risultati e ai plot per la visualizzazione. Si riportano qui solo le funzioni principali di calcolo delle misure di associazione (ROR) e disegno della Heatmap, e di creazione dei modelli di regressione lineare.

Librerie utilizzate

```
library(tidyverse)
library(cowplot)
library(superheat)
library(RColorBrewer)
```

Calcolo delle misure di associazione

```
Wrangle <- function(df, D) {
  #
  # Argomenti:
  # df : Dataframe con tutte le segnalazioni già pulite
  # D : Lista dei principi attivi
  #
  # Risultato:
  # Wrangle_df : Database contenente i dati riassuntivi
  # delle analisi di disproporzionalità
  Wrangle_df <- data.frame(matrix(ncol = 12, nrow = 0))
  colnames(Wrangle_df) <- c("Drug_Code", "Drug_Name", "AE", "F_EA", "F_nEA",
                           "nF_EA", "nF_nEA", "ROR", "s", "ROR_m", "ROR_M", "IC")
  for (i in 1:length(AE_list)) {
    EA_Name <- AE_list[i]
    print(EA_Name)
    x <- subset(df, str_detect(Reactions, AE_list[i]))
    for (d in D) {
      D_Code <- ATC$Code[ATC$Substance == d]
      y <- subset(df, !str_detect(`Suspect Product Active Ingredients Code`, D_Code))
      if (sum(str_detect(x$`Suspect Product Active Ingredients Code`, D_Code)) == 0) {
        F_nEA <- sum(str_detect(df$`Suspect Product Active Ingredients Code`, D_Code))
        nF_EA <- sum(str_detect(y$Reactions, EA_Name))
        nF_nEA <- sum(!str_detect(y$Reactions, EA_Name))
        new_row <- list(D_Code, d, EA_Name, 0, F_nEA, nF_EA, nF_nEA, NA, NA, NA, NA, NA)
      } else {
        tab <- table(str_detect(df$`Suspect Product Active Ingredients Code`, D_Code),
                     str_detect(df$Reactions, AE_list[i]))
        colnames(tab)[1] <- "nEA" # altri eventi
        colnames(tab)[2] <- "EA" # evento e
```

```

rownames(tab)[1] <- "nF" # altri farmaci
rownames(tab)[2] <- "F"  # farmaco d
t <- as.data.frame.matrix(tab)
F_EA <- t["F","EA"]
F_nEA <- t["F","nEA"]
nF_EA <- t["nF","EA"]
nF_nEA <- t["nF","nEA"]
ROR_m <- NA
ROR_M <- NA
ROR <- NA
IC <- NA
s <- NA
if (F_EA>=3) {
  ROR <- F_EA * nF_nEA / nF_EA / F_nEA
  s <- sqrt(1/F_EA + 1/F_nEA + 1/nF_EA + 1/nF_nEA)
  ROR_m <- exp(log(ROR) - 1.96*s)
  ROR_M <- exp(log(ROR) + 1.96*s)
  if (is.infinite(ROR)) {IC <- "[|Inf|]"} else {
    ROR_m <- round(ROR_m, digits = 1)
    ROR <- round(ROR, digits = 1)
    ROR_M <- round(ROR_M, digits = 1)
    IC <- paste("[", ROR_m,
                "|", ROR, "|",
                ROR_M, "]",
                sep = "")
  }
}
new_row <- list(D_Code, d, EA_Name, F_EA, F_nEA, nF_EA, nF_nEA, ROR, s, ROR_m,
               ROR_M, IC)
}
Wrangle_df[nrow(Wrangle_df)+1,] <- new_row
}
}
return(Wrangle_df)
}

```

Funzione per disegnare la Heatmap

```

Create_Matrix <- function(df, index) {
  m <- matrix(ncol = length(AE_list), nrow = length(D_Code_list))
  rownames(m) <- D_Code_list
  colnames(m) <- AE_list
  for (e in AE_list) {
    x <- subset(df, AE == e)
    for (d in D_Code_list){
      print(d)
      if (!is.na(x$ROR_m[x$Drug_Code == d])) {
        if(x$ROR_m[x$Drug_Code == d] > 1) {
          m[d,e] = x[[index]][x$Drug_Code == d]
        }
      }
    }
  }
}
}

```

```

onlyNAcolums_idx <- m %>%
  is.na() %>%
  apply(MARGIN = 2, FUN = all)
m <- m[,!onlyNAcolums_idx]
m <- m[rowSums(is.na(m)) != ncol(m), ]
m <- m %>%
  as.data.frame()
m <- setNames(cbind(rownames(m), m, row.names = NULL), c("Code", colnames(m)))
m <- m %>%
  left_join(ATC, by = "Code") %>%
  select(Code, "Substance", everything()) %>%
  unite(Drug, c(Code, "Substance"), sep = ": ")
return(m)
}

Print_Heatmap <- function(df) {
  #
  # Argomenti:
  # df : Dataframe con i risultati delle analisi di disproporzionalità
  #
  # Risultato:
  # Heatmap.png : Grafico che facilita la visualizzazione dei ROR
  Heatmatrix <- Create_Matrix(df, "ROR")
  Code_member <- substr(Heatmatrix$Drug, start = 1, stop = 4)
  Heatmatrix <- Heatmatrix %>%
    remove_rownames() %>%
    column_to_rownames(var = "Drug") %>%
    as.matrix()
  IC_matrix <- Create_Matrix(df, "IC")
  IC_matrix <- IC_matrix %>%
    remove_rownames() %>%
    column_to_rownames(var = "Drug") %>%
    as.matrix()
  IC_matrix[is.na(IC_matrix)] <- ""
  Heatmatrix[Heatmatrix[,] == "Inf"] <- 100
  Heatmatrix[Heatmatrix[,] > 100] <- 100
  l <- paste("Visualization/Heatmap_", deparse(substitute((df))), ".png", sep = "")
  png(l, height = 15000, width = 8000)
  superheat(Heatmatrix,
    heat.pal = c("white", "red", "#b35806", "#542788"),
    heat.pal.values = c(0, 0.1, 0.5, 1),
    heat.col.scheme = "red",
    heat.lim = c(1, 100),
    bottom.label.text.angle = 90,
    bottom.label.text.size = 10,
    bottom.label.size = 0.1,
    force.left.label = TRUE,
    left.label.text.size = 4,
    left.label.size = 0.4,
    force.grid.hline = TRUE,
    grid.hline.col = "gray",
    grid.vline.col = "gray",
    heat.na.col = "white",

```

```

    X.text = IC_matrix,
    X.text.size = 3,
    left.label.text.alignment = "right",
    pretty.order.rows = FALSE,
    membership.rows = Code_member,
    left.label = "variable",
    row.title = "Substance",
    row.title.size = 6,
    column.title = "Adverse Event",
    column.title.size = 6)
dev.off()
}

```

Funzione per produrre i modelli di regressione lineare

```

LRM <- function(df, PhD_df, index){
  #
  # Argomenti:
  #   df      : Dataframe con i risultati delle analisi di disproporzionalità
  #   PhD_df  : Database contenente i dati di farmacodinamica
  #   index   : indice farmacodinamico (pChEMBL o Occupancy)
  #
  # Risultato:
  #   LRM.csv : Database contenente i dati riassuntivi
  #             dei modelli di regressione lineare
  #   LRM.pdf : Grafici rappresentanti i modelli di regressione lineare
  i <- deparse(substitute(index))
  Targets_list <- as.list(unique(PhD_df$`Target`))
  Action_list <- as.list(unique(PhD_df$Action))
  LRM_df <- as.data.frame(matrix(nrow=0, ncol=8))
  colnames(LRM_df) <- c("AE", "Target", "Action", "Intercept", "Slope", "SE", "p_value", "Pearson")
  pdf("Visualization/LRM.pdf")
  for (e in AE_list){
    x <- subset(df, df$AE == e)
    x <- subset(x, is.na(x$ROR) == FALSE)
    x <- subset(x, is.infinite(x$ROR) == FALSE)
    for (t in Targets_list) {
      y <- subset(PhD_df, `Target` == t) %>%
        rename(Drug_Name = Substance)
      z1 <- left_join(x,y, by = "Drug_Name")
      z1 <- subset(z1, is.na(z1[[i]]) == FALSE)
      for (m in Action_list){
        z <- subset(z1, z1$Action == m)
        if (dim(z)[1] >= 4){
          Intercept <- round(coefficients(summary(lm(z$ROR~z[[i]])))[1],2)
          Slope <- round(coefficients(summary(lm(z$ROR~z[[i]])))[2],2)
          SE <- round(coefficients(summary(lm(z$ROR~z[[i]])))[4],2)
          p_value <- round(coefficients(summary(lm(z$ROR~z[[i]])))[8],6)
          P <- cor.test(z$ROR, z[[i]], method="pearson")
          Pearson <- round(P$estimate,2)
          LRM_df[nrow(LRM_df)+1,] <- c(e, t, m, Intercept, Slope, SE, p_value, Pearson)
          if (p_value <= 1){
            plot1 <- ggplot(data = z, aes(x=z[[i]], y=z$ROR, main= paste("ROR ~ ", t))) +

```

```

    geom_smooth(method = "lm") +
    geom_point(aes(color = z$Drug_Family)) +
    xlab(index) +
    ylab("ROR")
title <- ggdraw() + draw_label(paste(e, " ~ ", t, "[", m, "]"), fontface = "bold")
results <- ggdraw() + draw_label(paste("Intercept: ", Intercept,
                                     "      Slope: ", Slope,
                                     "      SE: ", SE,
                                     "      p-value: ", p_value,
                                     "      Pearson: ", Pearson))
print(plot_grid(title, plot1, results, ncol=1, rel_heights=c(0.1, 1, 0.1)))
  }
}
}
}
}
dev.off()
LRM_df <- LRM_df %>%
  arrange(p_value) %>%
  mutate(Rank = rank(p_value))
LRM_df <- LRM_df %>%
  mutate(BH20 = (Rank/nrow(LRM_df))*0.20) %>%
  mutate(Sign20 = (p_value <= BH20))
write_csv2(LRM_df, "Results/LRM.csv")
}

```