```
rm(list = ls())

# Load and install require packages
ipak <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}

# Remove unuseful columns where each column has all 1 or all 0
remove_unusuful_col <- function(df){
  unique_eachcol <- apply(df, 2, function(x) length(unique(x)))
  df <- df[, unique_eachcol > 1] # remove columns which has just one
unique values
  df1 <- unique(df) # Get unique dataset
  return(df1)
}

# Merge different data files, We merge all the data and then remove all
the rows with a single NA value (It presents the row which has no
matching between atleast one of the given data list)
merge_all <- function(data_list){
  temp  <- Reduce(function(...) merge(..., all=T), data_list)
  merged_file <- na.omit(temp) # remove NAs
  merged_file <- unique(merged_file) # Keep only unique files
  merged_file <- merged_file[order(merged_file$id), ] # order the data
as known associates
  merged_file <- subset(merged_file, select = -id) # Remove id columns
  final_data  <- remove_unusuful_col(merged_file)
  rownames(final_data)  <- 1:nrow(final_data)

  # refactor drug and disease
  final_data$Disease <- factor(final_data$Disease)
  final_data$Drug    <- factor(final_data$Drug)
  return(final_data)
}

#########################################################################

## Install/Load packages
# Package names
packages <- c("XLConnect", "tidyr", "dplyr", "openxlsx")
ipak(packages)


fileName = "My work.xlsx"
# Read xlsx file sheet by sheet as it is returning error in reading
entire workbook at once
known.associates2 <- read.xlsx(fileName, sheet = 3, startRow = 1,
colNames = TRUE)
drug.substructure <- read.xlsx(fileName, sheet = 4, startRow = 1,
colNames = TRUE)
drug.sideEffect   <- read.xlsx(fileName, sheet = 5, startRow = 1,
colNames = TRUE)
drug.gene         <- read.xlsx(fileName, sheet = 9, startRow = 1,
```

```
colNames = TRUE)
disease.mirna     <- read.xlsx(fileName, sheet = 7, startRow = 1,
colNames = FALSE)
disease.gene <- read.xlsx(fileName, sheet = 8, startRow = 2, colNames =
TRUE)

#######################################
# Preprocessing of the data
# Convert all the names in lower case to maximise matching
known.associates2 <- as.data.frame(sapply(known.associates2, tolower))
drug.substructure <- as.data.frame(sapply(drug.substructure, tolower))
drug.sideEffect   <- as.data.frame(sapply(drug.sideEffect, tolower))
drug.gene         <- as.data.frame(sapply(drug.gene, tolower))
disease.mirna     <-as.data.frame(sapply(disease.mirna, tolower))
disease.gene      <- as.data.frame(sapply(disease.gene, tolower))
known.associates2$id <- 1:nrow(known.associates2)

# Change the columns names so that it is matching
names(drug.substructure)[1] <- "Drug"
names(drug.sideEffect)[1] <- "Drug"
names(drug.gene)[2] <- "Drug"
names(disease.gene)[2] <- "Disease"
names(disease.mirna) <- c("miRNA", "Gene.Symbol", "Disease")

# Remove disease column from disease miRNA
disease.mirna <- disease.mirna[, -3]

# Make 0-1 for drug gene structure
drug.gene <- unique(drug.gene)
drug.gene2 <- drug.gene %>%
  gather(Gene.Symbol,name,starts_with("Gene.Symbol")) %>%
  mutate(present = 1) %>%
  select(-Gene.Symbol) %>%
  spread(name,present,fill = 0)

# Make 0-1 for disease gene structure
disease.gene <- unique(disease.gene)
disease.gene <- disease.gene[, c("Disease", "Gene.Symbol")]
disease.gene2 <- disease.gene %>%
  gather(Gene.Symbol,name,starts_with("Gene.Symbol")) %>%
  mutate(present = 1) %>%
  select(-Gene.Symbol) %>%
  spread(name,present,fill = 0)

# first merge disease gene and disease miRNA
disease.gene.mirna <- merge(disease.gene, disease.mirna, by =
"Gene.Symbol")
disease.gene.mirna <- disease.gene.mirna[, c("Disease", "Gene.Symbol",
"miRNA")]
# mutate it get 0-1
disease.gene.mirna <- unique(disease.gene.mirna)
disease.gene.mirna1 <- disease.gene.mirna %>%
  gather(Gene.Symbol,name,starts_with("Gene.Symbol")) %>%
  mutate(present = 1) %>%
  select(-Gene.Symbol) %>%
  spread(name,present,fill = 0)
disease.gene.mirna2 <- disease.gene.mirna1 %>%
```

```
  gather(miRNA,name,starts_with("miRNA")) %>%
  mutate(present = 1) %>%
  select(-miRNA) %>%
  spread(name,present,fill = 0)
rm(disease.gene.mirna1, disease.gene.mirna)

# Merging drug-disease associations+drug-substructure+drug-side
effect+disease-gene
list.df1 <- list(known.associates2, drug.substructure, drug.sideEffect,
disease.gene2)
merge_file_1 <- merge_all(list.df1)
write.xlsx(merge_file_1, "combine_asso_drugSub_drugse_diseaseGene.xlsx")

# Merging drug-disease associations+drug-substructure+drug-side
effect+disease-miRNA+disease-gene
list.df2 <- list(known.associates2, drug.substructure, drug.sideEffect,
disease.gene.mirna2)
merge_file_2 <- merge_all(list.df2)
write.xlsx(merge_file_2,
"combine_asso_drugSub_drugse_diseaseGeneMIRNA.xlsx")

# Merging drug-disease associations+drug-substructure+drug-side
effect+drug-gene+disease-gene
list.df3 <- list(known.associates2, drug.substructure, drug.sideEffect,
drug.gene2)
merge_file_3 <- merge_all(list.df3)
write.xlsx(merge_file_3,
"combine_asso_drugSub_drugse_drugGene_diseaseGene.xlsx")

# Merging drug-disease associations+drug-substructure+drug-side
effect+drug-gene+disease-miRNA+disease+gene
merge_file_4 <- Reduce(function(...) merge(..., all=T), list.df3)
merge_file_4 <- unique(merge_file_4)
merge_file_4 <- na.omit(merge_file_4)
merge_file_4 <- merge(merge_file_4, disease.gene.mirna2, by = "Disease")
merge_file_4 <- merge_file_4[order(merge_file_4$id),]
merge_file_4 <- subset(merge_file_4, select = -id)
merge_file_4 <- remove_unusuful_col(merge_file_4)
rownames(merge_file_4)  <- 1:nrow(merge_file_4)
merge_file_4$Disease <- factor(merge_file_4$Disease)
merge_file_4$Drug <- factor(merge_file_4$Drug)
write.xlsx(merge_file_4,
"combine_asso_drugSub_drugse_drugGene_diseaseGeneMiRNA.xlsx")

# Plot a frequency chart for merge_file_4 disease and drug
par(mfrow = c(1,2))
barplot(table(merge_file_4$Disease), las = 2, cex.names = 0.4, ylab =
"Disease freq")
barplot(table(merge_file_4$Drug), las = 2, cex.names = 0.4,ylab = "Drug
freq" )

# Pseudoranom data
psudorandom <- function(data){
  m <-  1
  while (m == 1){
    nsamples <- sample(nrow(data), 200) # Extract 200 random sample from
data
```

```
    trainData <- data[-nsamples, ]
    testData <- data[nsamples, ]
    if (length(unique(testData$Disease)) == length(trainData$Disease)){
      m   <- 0
      D <- list("train" = trainData, "test" = testData)
      return(D)
    }
  }
}


# Split train and test dataset
split_data <- function(data){

  # If only one unique disease, keep it in test data
  t <- as.data.frame(table(data$Disease))
  tnew <- t[t$Freq == 1, ]
  temp <- NA
  for (i in 1:nrow(tnew)){
    temp[i] <- which(data$Disease == tnew$Var1[i])
  }
  test <- data[temp, ]
  ndata <- data[-temp, ]
  ndata$Disease <- factor(ndata$Disease)
  ####
  # If two disease, keep it in test data
#   t <- as.data.frame(table(ndata$Disease))
#   tnew <- t[t$Freq == 2, ]
#   foo <- NULL
#   train <- NULL
#   for (i in 1:nrow(tnew)){
#     temp <- which(ndata$Disease == tnew$Var1[i])
#     ind_select <- sample(temp, 1)
#     test <- rbind(test, ndata[ind_select, ])
#     train <- rbind(train,ndata[temp[temp != ind_select], ] )
#     foo <- c(foo, temp)
#   }

  # Split rest data into train and test data with 200 total in testdata
  # ndata <- ndata[-foo, ]
  m <- 1
  while (m == 1){
    sample_ind <- sample(nrow(ndata), 200-nrow(test))
    trainData <- ndata[-sample_ind, ]
    testData <- ndata[sample_ind, ]
    if (length(unique(testData$Disease)) ==
length(unique(trainData$Disease))){
      m   <- 0
    }
  }
  testData <- rbind(testData, test)
  testData$Disease <- factor(testData$Disease)
  trainData$Disease <- factor(trainData$Disease)
  D <- list(train = trainData, test =  testData)
  return(D)
}
```

```
# randomise traindata to obtain rest of the dataset
randomise_disease <- function(data, randnum){
  rand_data <- data
  rand_data$Class <-  rep(1, nrow(data))
  for(i in 1:randnum){
    s_ind <- sample(nrow(data), 1)
    d_ind <- sample(nrow(data), 1)
    if (s_ind != d_ind){
      rand_data[nrow(data)+i, ] <- cbind(data$Disease[d_ind],
data[s_ind, -1] , Class = 0)
    }else
      # temp <- data.frame(Disease= data$Disease[d_ind], Drug =
data$Drug[s_ind], data[s_ind, -c(1,2)], Class = 1)
      rand_data[nrow(data)+i, ] <- cbind(data$Disease[d_ind],
data[s_ind, -1] , Class = 1)
  }
  return(rand_data)
}
D <- split_data(merge_file_4)
# Get test data add class to it
testData <- as.data.frame(D$test)
testData$Class <- rep(1, nrow(testData))

# randmoise train data for disease and add class accordingly
rand_trainData <- randomise_disease(D$train, 1000-nrow(D$train))

write.xlsx(rand_trainData, "rand_disease_trainData.xlsx")
write.xlsx(testData, "test_data.xlsx")
# Combine drug disease for test data
comb_test <- testData
comb_test$Disease <- paste(comb_test$Disease, "-", comb_test$Drug, sep =
"")
comb_test <- comb_test[, -2]
names(comb_test)[1] <- "Disease-Drug"
write.xlsx(comb_test, "combine_disease-drug_testdata.xlsx")


comb_drug_disease <- rand_trainData
comb_drug_disease$Disease <- paste(rand_trainData$Disease, "-"
,rand_trainData$Drug, sep = "")
comb_drug_disease <- comb_drug_disease[, -2]
names(comb_drug_disease)[1] <- "Disease-Drug"
write.xlsx(comb_drug_disease, "rand_combine_disease-
drug_trainData.xlsx")

# randomise train data for drugs and add class accordingly
# randomise traindata to obtain rest of the dataset
randomise_drug <- function(data, randnum){
  rand_data <- data
  rand_data$Class <-  rep(1, nrow(data))
  for(i in 1:randnum){
    s_ind <- sample(nrow(data), 1)
    d_ind <- sample(nrow(data), 1)
    if (s_ind != d_ind){
      rand_data[nrow(data)+i, ] <-
cbind(data$Disease[s_ind],data$Drug[d_ind], data[s_ind, -c(1, 2)] ,
Class = 0)
```

```
    }else
      # temp <- data.frame(Disease= data$Disease[d_ind], Drug =
data$Drug[s_ind], data[s_ind, -c(1,2)], Class = 1)
      rand_data[nrow(data)+i, ] <-
cbind(data$Disease[s_ind],data$Drug[d_ind], data[s_ind, -c(1, 2)] ,
Class = 1)
  }
  return(rand_data)
}

# randmoise train data for disease and add class accordingly
randdrug_trainData <- randomise_drug(D$train, 1000-nrow(D$train))
write.xlsx(randdrug_trainData, "rand_drug_trainData.xlsx")

# Find number of features matching
substructureMatch <- length(intersect(names(merge_file_4),
names(drug.substructure)))
drugseMatch <- length(intersect(names(merge_file_4),
names(drug.sideEffect)))
druggeneMatch <-  length(intersect(names(merge_file_4),
names(drug.gene2)))
diseaseGeneMatch <- length(intersect(names(merge_file_4),
names(disease.gene2)))
diseaseMirnaMatch <- length(intersect(names(merge_file_4),
disease.mirna$miRNA))

x <- data.frame(table(merge_file_4$Disease))
quartz()

png(filename = "disease_freq.png", width = 300, height = 200,units =
"mm", res = 600)
par(mai = c(2, 1,0.5, 1), font.axis = 2, font.lab = 2)
barplot(x$Freq, las = 2, cex.names = 0.8, ylab = "Disease freq",
names.arg = substr(x$Var1, 1, 20), ylim = c(0, 120))
dev.off()
x <- data.frame(table(merge_file_4$Drug))
png(filename = "drug_freq.png", width = 300, height = 200,units = "mm",
res = 600)
par(mai = c(2, 1,0.5, 1), font.axis = 2, font.lab = 2)
barplot(x$Freq, las = 2, cex.names = 0.8, ylab = "Drug freq", names.arg
= substr(x$Var1, 1, 20), ylim = c(0, 120))
dev.off()
```