Basic R tutorial

February 22, 2024

Contents

Problem 3: Learning Bayesian networks from protein data

Preparation to run the code

Setting seed for reproducibility and loading packages.

```r
set.seed(42)
library("GGally")
library("BiDAG")
```

Loading data

```r
data <- read.table("2005_sachs_2_cd3cd28icam2_log_std.csv", sep=",", header=TRUE)
```
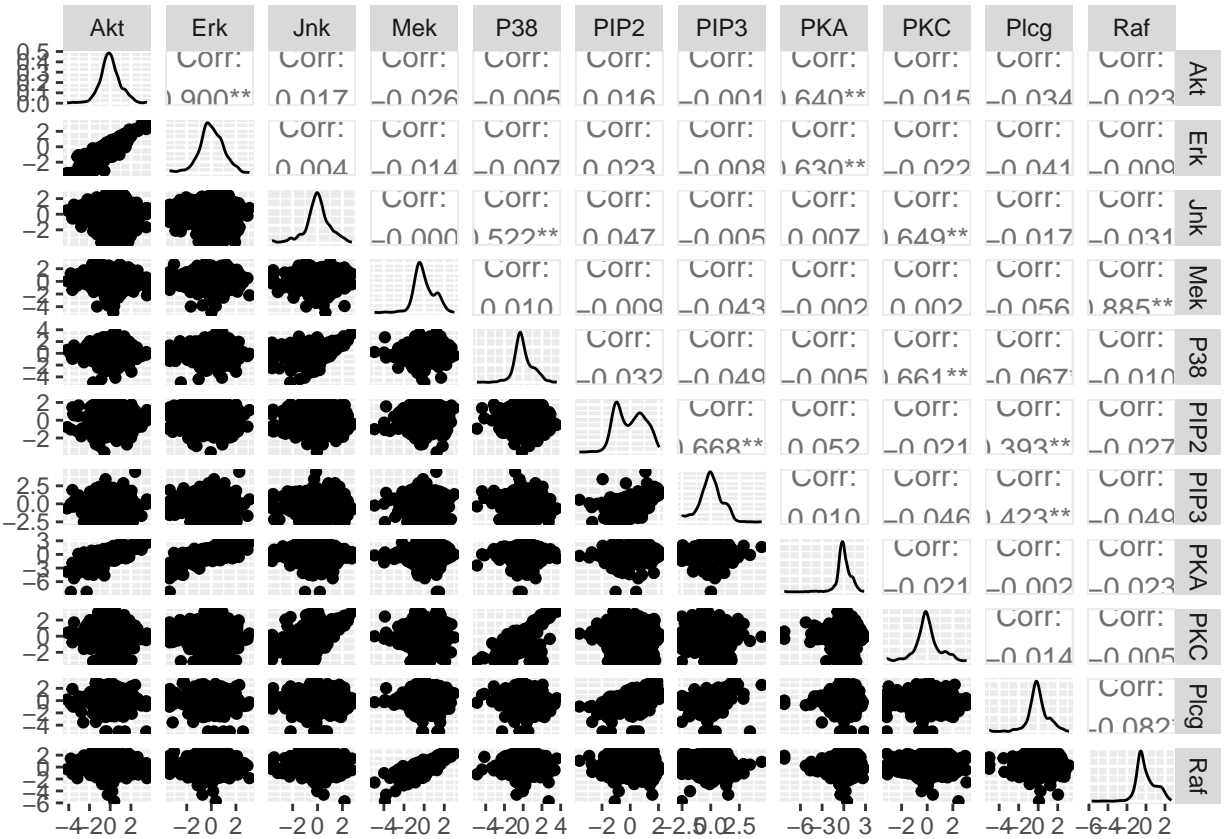
Variables and observations

```r
num_variables <- ncol(data)
num_observations <- nrow(data)
```

Visualisation of transformed data

TODO: must resize the plot

```r
pair_plot <- ggpairs(data, progress=FALSE)
pair_plot
```

Akt | Erk | Jnk | Mek | P38 | PIP2 | PIP3 | PKA | PKC | Plcg | Raf

Corr: 0.900** · Corr: 0.017 · Corr: −0.026 · Corr: −0.005 · Corr: 0.016 · Corr: −0.001 · Corr: 0.640** · Corr: −0.015 · Corr: −0.034 · Corr: −0.023 — Akt

Corr: 0.004 · Corr: −0.014 · Corr: −0.007 · Corr: 0.023 · Corr: −0.008 · Corr: 0.630** · Corr: −0.022 · Corr: −0.041 · Corr: −0.009 — Erk

Corr: −0.000 · Corr: 0.522** · Corr: 0.047 · Corr: −0.005 · Corr: 0.007 · Corr: 0.649** · Corr: −0.017 · Corr: −0.031 — Jnk

Corr: 0.010 · Corr: −0.009 · Corr: −0.043 · Corr: −0.002 · Corr: 0.002 · Corr: −0.056 · Corr: 0.885** — Mek

Corr: −0.032 · Corr: −0.049 · Corr: −0.005 · Corr: 0.661** · Corr: −0.067 · Corr: −0.010 — P38

Corr: 0.668** · Corr: 0.052 · Corr: −0.021 · Corr: 0.393** · Corr: −0.027 — PIP2

Corr: 0.010 · Corr: −0.046 · Corr: 0.423** · Corr: −0.049 — PIP3

Corr: −0.021 · Corr: −0.002 · Corr: −0.023 — PKA

Corr: −0.014 · Corr: −0.005 — PKC

Corr: −0.082 — Plcg

Raf

TODO: execute the following routine with different parameters (10 times for each change of parameter)

TODO: include possibility of parallelization

Defining functions

```r
splitting_data <- function(data) {
    # Shuffling data
    indices <- 1:nrow(data)
    indices <- sample(length(indices))

    # Splitting data
    train_size <- ceiling(nrow(data)*0.8)
    train_indices <- indices[1:train_size]
    test_indices <- indices[(train_size+1):(length(indices))]

    # Checking if there is no overlap
    if(length(unique(c(train_indices, test_indices))) != length(c(train_indices, test_indices))) {
        print("Overlap!")
    }

    train_data <- data[row.names(data) %in% train_indices, ]
    test_data <- data[row.names(data) %in% test_indices, ]

    split_data <- list("train_data"=train_data, "test_data"=test_data)
```

```r
    return(split_data)
}

training_BN <- function(data, bgepar) {
    init_score_par <- scoreparameters("bge", data$train_data, bgepar)
    learnt_BN <- iterativeMCMC(init_score_par)

    return(learnt_BN)
}

testing_BN <- function(data, BN, bgepar) {
    test_score_par <- scoreparameters("bge", data$test_data, bgepar)
    test_score <- scoreagainstDAG(test_score_par, BN$DAG)
    return(mean(test_score))
}
```

Default parameters

```r
bgepar <- list(am=1, aw=NULL, edgepf=1)
split_data <- splitting_data(data)
learnt_BN <- training_BN(split_data, bgepar)
```

```
## maximum parent set size is 3
## core space defined, score table are being computed
## score tables completed, iterative MCMC is running
```

```r
mean_test_score <- testing_BN(split_data, learnt_BN, bgepar)
print(paste0("Average BGe score on testing data: ", mean_test_score))
```

```
## [1] "Average BGe score on testing data: -12.1192090033328"
```

Different parameters

```r
am <- c(10^(-5), 10^(-3), 10^(-1), 10, 10^2)
```

Render this .rmd into a pdf

```r
library(rmarkdown)
render("1/1.Rmd", pdf_document(TRUE), "1.pdf") # TRUE adds table of content
```