

# Basic R tutorial

February 22, 2024

## Contents

<b>Problem 3: Learning Bayesian networks from protein data</b>	<b>1</b>
Preparation to run the code . . . . .	1
Loading data . . . . .	1
Variables and observations . . . . .	1
Visualisation of transformed data . . . . .	1
Splitting data . . . . .	2
Render this .rmd into a pdf . . . . .	3

## Problem 3: Learning Bayesian networks from protein data

### Preparation to run the code

Setting seed for reproducibility and loading packages.

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

### Loading data

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

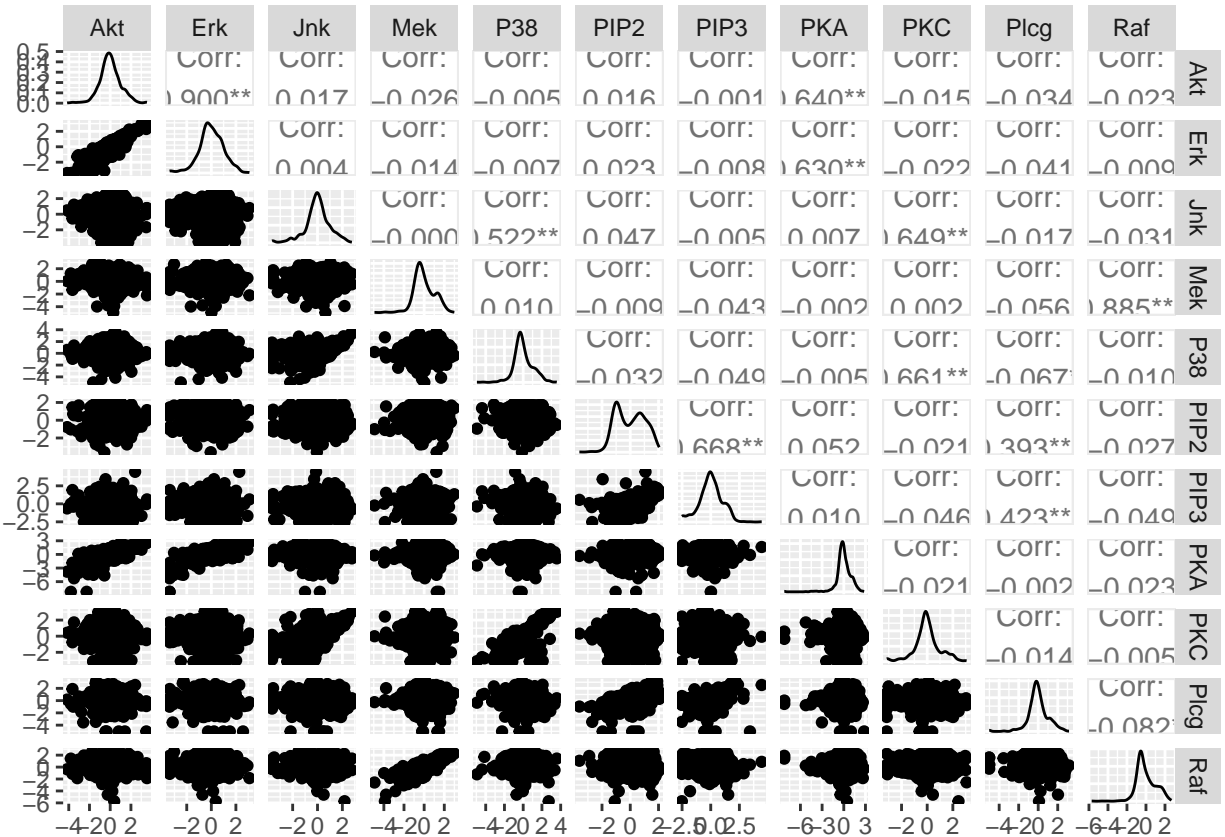
### Variables and observations

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

### Visualisation of transformed data

TODO: must resize the plot

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



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

TODO: include possibility of parallelization

## Splitting data

TODO: shuffling is needed as well

```
train_size <- ceiling(num_observations*0.8)
indices <- 1:num_observations
indices <- sample(length(indices))
train_indices <- indices[1:train_size]
test_indices <- indices[(train_size+1):(length(indices))]

train_data <- data[row.names(data) %in% 1:train_size, ]
test_data <- data[row.names(data) %in% (train_size+1):nrow(data), ]

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

## [1] "No overlap"
init_score_par <- scoreparameters("bge", train_data)

learnt_BN <- iterativeMCMC(init_score_par)

## maximum parent set size is 2
```

```
## core space defined, score table are being computed
## score tables completed, iterative MCMC is running
test_score_par <- scoreparameters("bge", test_data)
test_score <- scoreagainstDAG(test_score_par, learnt_BN$DAG)

mean(test_score)

## [1] -12.87108
```

Render this .rmd into a pdf

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