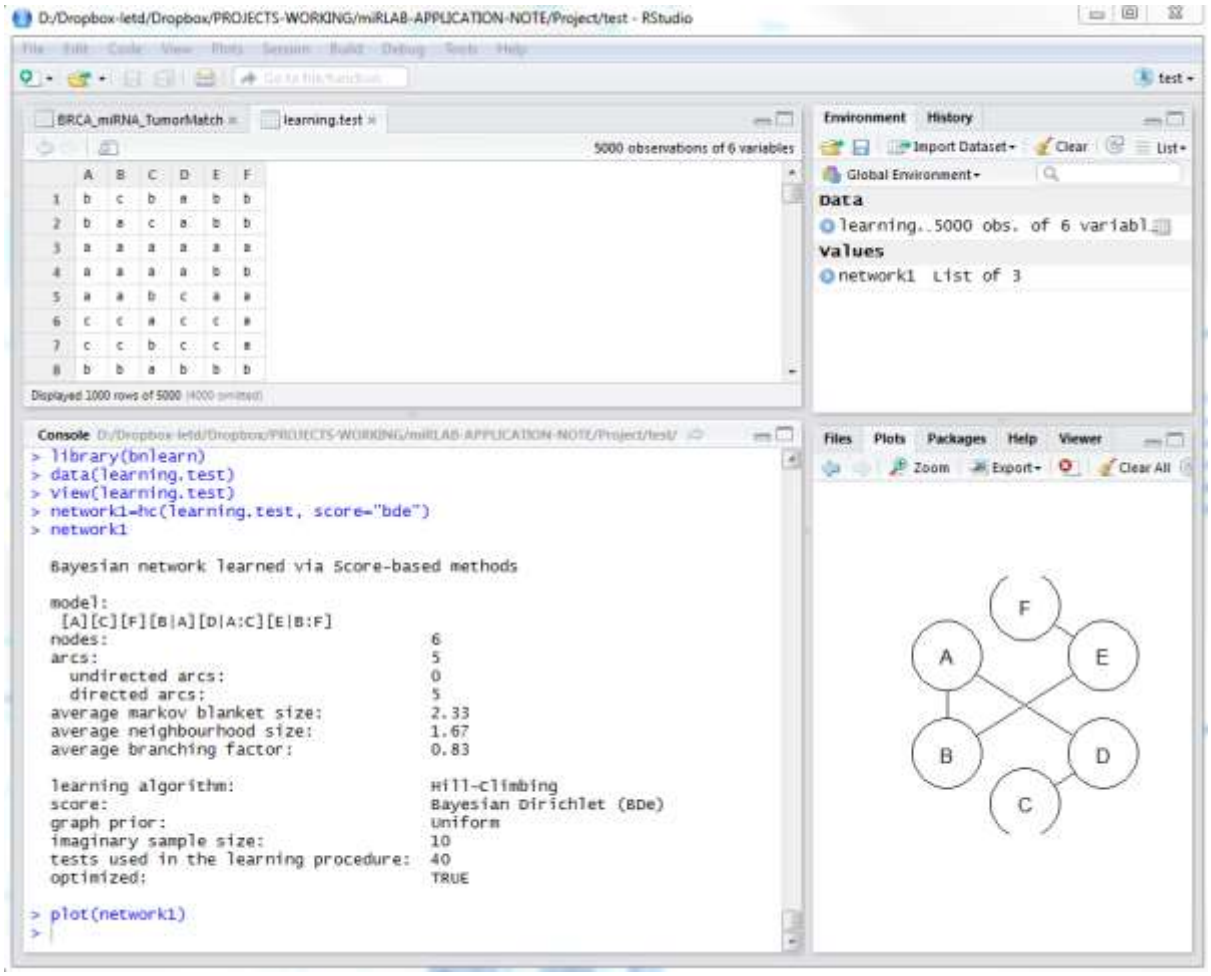
Practical 3: Bayesian Networks

I. Learning Bayesian network structure from data – Search and Score approaches

1. Start R or Rstudio (Recommended).
2. Install *bnlearn* and *pcalg* packages.
   - Select Tools→ install packages→specify the name of the package you want to install.
   - Tick the "Install dependencies" box to install all the dependent packages.
   - Click "Install".
3. Run the following codes to learn the Bayesian network structure for the dataset learning.test using the Hill-Climbing algorithm.

```
> library(bnlearn)
> data(learning.test) #call an example built-in dataset
> View(learning.test)
> network1=hc(learning.test, score="bde")
> network1
```



4. Use the Tabu algorithm (replace hc with tabu) to learn the Bayesian network.
5. Compare the networks in Hill-climbing and Tabu.

II. Learning Bayesian network structure from data – Constraint based approaches

In this section, we use the PC algorithm from the *pcalg* package to learn the Bayesian network structure from data. Please refer to the user manual of pcalg for more details https://cran.r-project.org/web/packages/pcalg/pcalg.pdf

1. *Using numeric data*

```
library(pcalg)
## Load predefined data
data(gmG)
gmG8$x[1:5,]
n <- nrow (gmG8$ x)
V <- colnames(gmG8$ x) # labels aka node names


## estimate CPDAG
pc.fit <- pc(suffStat = list(C = cor(gmG8$x), n = n),
indepTest = gaussCItest, alpha=0.01, labels = V)


if (require(Rgraphviz)) {
      ## show estimated graph
      par(mfrow=c(1,2))
      plot(pc.fit, main = "Estimated graph")
      plot(gmG8$g, main = "True DAG")
}
```

2. *Using discrete data*

```
## Load data
data(gmD)
gmD$x[1:5,]
V <- colnames(gmD$x)


## define sufficient statistics
suffStat <- list(dm = gmD$x, nlev = c(3,2,3,4,2), adaptDF =
FALSE)


## estimate the structure

pc.D <- pc(suffStat, indepTest = disCItest, alpha = 0.01,
labels = V, verbose = TRUE)

#compare the graphs
if (require(Rgraphviz)) {
## show estimated CPDAG
      par(mfrow = c(1,2))
      plot(pc.D, main = "Estimated graph")
      plot(gmD$g, main = "True DAG")
}
```

3. ***Using binary data***

```
## Load binary data
data(gmB)
gmB$x[1:5,]
V <- colnames(gmB$x)

## estimate the structure
pc.B <- pc(suffStat = list(dm = gmB$x, adaptDF = FALSE),
indepTest = binCItest, alpha = 0.01, labels = V, verbose =
TRUE)

pc.B

if (require(Rgraphviz)) {
      ## show estimated CPDAG
      plot(pc.B, main = "Estimated CPDAG")
      plot(gmB$g, main = "True DAG")
}
```