Advanced Data Analytics 2 – Bioinformatics Project

Student name: Minh Tu Tran

Student ID: 110195985

Task 1: Use a causal structure learning algorithm to find the skeleton of the gene regulatory network using the gene expression data.

We apply the PC algorithm to study the skeleton of the gene regulatory network. This algorithm is popular for learning the structure of a causal Bayesian network. There are two steps in this algorithm: learning the skeleton and orientating the edges. The learning skeleton step is implemented by the `skeleton()` function in the pcalg package in R. Figure 1 presents the estimated skeleton while Table 1 shows its node names.

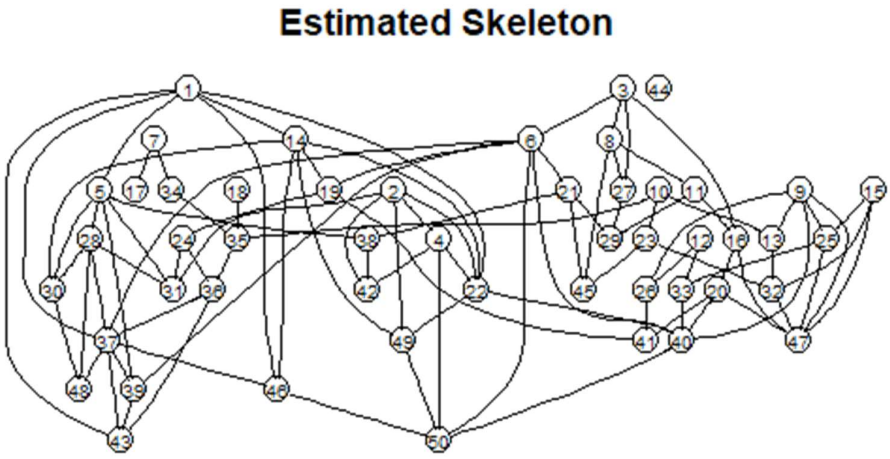


Figure 1. The skeleton of the gene regulatory network

1: FIGF	2: LYVE1	3: CD300LG	4: SCARA5	5: PAMR1
6: SDPR	7: MYOM1	8: BTNL9	9: KCNIP2	10: SLC2A4
11: PDE2A	12: LEP	13: ACVR1C	14: ABCA10	15: AQP7
16: GPR146	17: ATP1A2	18: FXYD1	19: ARHGAP20	20: NPR1
21: ATOH8	22: ABCA9	23: ALDH1L1	24: ADAMTS5	25: RDH5
26: GPAM	27: CA4	28: KLHL29	29: GPIHBP1	30: LOC728264
31: MAMDC2	32: TMEM132C	33: ITIH5	34: HSPB7	35: HSPB6
36: DMD	37: SPRY2	38: IGFBP6	39: CXCL2	40: EBF1
41: KLB	42: CLEC3B	43: TMEM220	44: IBSP	45: HIF3A
46: IGSF10	47: CIDEC	48: C2orf40	49: LEPR	50: ANGPTL1

Table 1. Node names of the skeleton

Below is the code employed to obtain the skeleton.

```
library(bnlearn)
library(pcalg)
library(gRain)

data <- read.csv("BRCA_RNASeqv2_top50.csv")
```

```
#remove class data
data_remove_class <- subset(data, select=-c(class) )

n <- nrow(data_remove_class)

## estimate Skeleton
skel.fit <- pcalg::skeleton(suffStat=list(C=
cor(data_remove_class), n=n), indepTest=gaussCItest, p=
ncol(data_remove_class), alpha=0.01)

if (require(Rgraphviz)) {
  ## show estimated Skeleton
  par(mfrow=c(1,1))
  plot(skel.fit, main="Estimated Skeleton")
}
```

Task 2: Find the top 10 other genes that have strong causal effects on ABCA9 using a causal inference algorithm.

To estimate the effects from the observational data, we employ the IDA (Intervention calculus when the DAG is Absent) method. Firstly, the PC algorithm is employed to learn the CPDAG. After that, the causal effects in all DAGs (from CPDAG) are inferred. If there are multiple effects between 2 variables, the minimum of absolute values of the effects is chosen.

By using the `pc()` function to learn the CPFDAG and `ida()` function to get the effects, coupled with getting the lower bound of the effects, we obtain the list of top 10 other genes that have strong causal effects on ABCA9 as shown in Table 2.

	genes	effects
14	ABCA10	1.7460337
18	FXYD1	1.3264465
28	GPIHBP1	0.9810083
1	FIGF	0.9035947
27	KLHL29	0.8711851
19	ARHGAP20	0.7114694
11	PDE2A	0.6134322
44	HIF3A	0.5996349
24	RDH5	0.5992832
31	TMEM132C	0.5883026

Table 2. Top 10 genes that have strong causal effects on ABCA9

Below is the implemented code to get the top 10 other genes that have strong causal effects on ABCA9 by using the `pc()` and `ida()` functions of the pcalg package.

```
#find the index of ABCA9
grep("ABCA9", colnames(data_remove_class))

suffStat <- list(C = cor(data_remove_class), n =
nrow(data_remove_class))
#get cpdag
pc.fit <- pc(suffStat, indepTest = gaussCItest, alpha = 0.01,labels =
V)
plot(pc.fit@graph)
```

```
#genes names
genes <- rownames(cov(data_remove_class))
genes=genes[-22]

#get effects of other nodes on the node 22
effects <- vector()
for (index in 1:50){
  if(index !=22){
    effect = ida(index, 22, cov(data_remove_class), pc.fit@graph)

    #the effect is the minimum of the absolute possible effects
    effects <- c(effects, min(abs(effect)))
  }
}

#merge gene names and effects
gene_effect = data.frame(genes,effects)

#sort effect from max to min
gene_effect_sort <- gene_effect[order(-effects),]
gene_effect_sort

#get top 10
head(gene_effect_sort,10)
```

Task 3: Use a local causal structure learning algorithm to find genes in the Markov blanket of ABCA9 from data.

The Markov blanket of a node includes its parents, children and children's parents (spouses). To obtain the Markov blanket, we employ a local causal structure learning algorithm called Incremental Association which belongs to the constraint-based structure learning algorithms. Basically, this algorithm consists of two phases: a forward selection to get all variables that belong in the Markov blanket (possibly contains the false positives) and a backward step to remove the false positives. There are 22 genes in the Markov blanket of ABCA9 as shown in Figure 2.

```
 [1] "EBF1"      "ABCA10"    "SCARA5"    "ACVR1C"    "CD300LG"   "LYVE1"     "GPAM"
 [8] "FIGF"      "LEPR"      "LOC728264" "TMEM132C"  "HIF3A"     "LEP"       "ANGPTL1"
[15] "PAMR1"     "CLEC3B"    "GPIHBP1"   "KLB"       "ATOH8"     "RDH5"      "NPR1"
[22] "CIDEC"
```

Figure 2. Genes in the Markov blanket of ABCA9

The code to obtain the Markov blanket of ABCA9 is shown as below.

```
MB.ABCA9=learn.mb(data_remove_class,       'ABCA9',       method='iamb',
alpha=0.01)
MB.ABCA9
```

Data transformation: Discretize the dataset to binary using the average expression of ALL genes as the threshold.

The average of all genes is used as the threshold to discretise the data set.

```
data_new = data_remove_class
theMean = mean(as.matrix(data_remove_class))
for(i in 1:ncol(data_remove_class)){
  data_new[[i]] <- ifelse(data_remove_class[[i]] > theMean, 1, 0)
}
data_new$class = data$class
```

Task 4: Use PC-simple algorithm (pcSelect) to find the parent and children set of the class variable. Evaluate the accuracy of the Naïve Bayes classification on the dataset in the following cases:
a) Use all features (genes) in the dataset
b) Use only the features (genes) in the parent and children set of the class
Compare the accuracy of the models in the two cases using 10-fold cross-validation.

By applying PC-simple algorithm (pcSelect) to find the parent and children set of the class variables, we get 10 variables as shown in Figure 3. They are FIGF, CD300LG, SCARA5, ATP1A2, ARHGAP20, ATOH8, KLHL29, MAMDC2, CXCL2 and TMEM220. In addition, the zMin scores show the influence of variables. The bigger the zMin, the stronger the effect.

```
$G
     FIGF     LYVE1   CD300LG    SCARA5     PAMR1      SDPR     MYOM1     BTNL9    KCNIP2
     TRUE     FALSE      TRUE      TRUE     FALSE     FALSE     FALSE     FALSE     FALSE
   SLC2A4     PDE2A       LEP    ACVR1C    ABCA10      AQP7    GPR146    ATP1A2     FXYD1
    FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE      TRUE     FALSE
 ARHGAP20      NPR1     ATOH8     ABCA9   ALDH1L1   ADAMTS5      RDH5     GPAM       CA4
     TRUE     FALSE      TRUE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
   KLHL29   GPIHBP1 LOC728264    MAMDC2   TMEM132C     ITIH5     HSPB7     HSPB6       DMD
     TRUE     FALSE     FALSE      TRUE     FALSE     FALSE     FALSE     FALSE     FALSE
    SPRY2    IGFBP6     CXCL2      EBF1       KLB    CLEC3B    TMEM220      IBSP     HIF3A
    FALSE     FALSE      TRUE     FALSE     FALSE     FALSE      TRUE     FALSE     FALSE
   IGSF10     CIDEC   C2orf40      LEPR   ANGPTL1
    FALSE     FALSE     FALSE     FALSE     FALSE

$zMin
 [1] 12.4322070  1.7534317  8.4777819  2.7931230  2.3164672  1.2653223  1.5646010  2.4111944
 [9]  0.1763132  2.3741410  2.0819730  2.0447520  2.2876307  1.1887476  1.2882622  1.1976424
[17]  5.0732814  2.3416787 10.7229683  2.1068818  2.6613375  1.8493220  1.9299658  1.7541644
[25]  0.8571422  2.2069921  2.1545528  8.0283830  1.8097174  2.1798232  5.0653144  2.4455780
[33]  0.7277827  2.4673648  0.4879938  2.2280258  1.9153599  1.9897989  7.6465681  1.3002320
[41]  1.9057004  1.9714550  4.3555226  1.7075677  0.8967489  1.5452763  1.2643481  2.5619648
[49]  1.2056382  1.1223886
```

Figure 3. PC-simple algorithm results - the parent and children set of the class variable

Let model 1 and model 2 denote, respectively, Naïve Bayes model using all features and Naïve Bayes model using only the features in the parent and children set of the class

Figures 4 and 5 present the results of two models on the full data set. In general, the two models perform relatively well in the classification by achieving high accuracies. However, the model 2 outperforms the model 1 by having higher values for accuracy 99.1749%, cancer precision 99.9%, and cancer recall 99.2% compared to the respective values in DT1 as 96.5347%, 99.7% and 96.5%.

```
Correctly Classified Instances        1170               96.5347 %
Incorrectly Classified Instances        42                3.4653 %
Kappa statistic                          0.8195
Mean absolute error                      0.0346
Root mean squared error                  0.1813
Relative absolute error                 20.5591 %
Root relative squared error             62.6032 %
Total Number of Instances             1212

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.973    0.035    0.736      0.973    0.838      0.829    0.996     0.945     N
               0.965    0.027    0.997      0.965    0.981      0.829    0.997     1.000     C
Weighted Avg.  0.965    0.028    0.973      0.965    0.967      0.829    0.997     0.995

=== Confusion Matrix ===

    a    b   <-- classified as
  109    3 |    a = N
   39 1061 |    b = C
```

Figure 4. Naïve Bayes classification results on the dataset – using all features

```
Correctly Classified Instances        1202               99.1749 %
Incorrectly Classified Instances        10                0.8251 %
Kappa statistic                          0.9523
Mean absolute error                      0.0085
Root mean squared error                  0.087
Relative absolute error                  5.0582 %
Root relative squared error             30.0558 %
Total Number of Instances             1212

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.991    0.008    0.925      0.991    0.957      0.953    0.999     0.993     N
               0.992    0.009    0.999      0.992    0.995      0.953    0.999     1.000     C
Weighted Avg.  0.992    0.009    0.992      0.992    0.992      0.953    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
  111    1 |    a = N
    9 1091 |    b = C
```

Figure 5. Naïve Bayes classification results on the dataset – using only the features in the parent and children set of the class

Figures 6 and 7 illustrate the models results in the two cases using the 10-fold cross-validation. The results are summarised in Table 3. The model 2 with accuracy, cancer precision and cancer recall as 99.1749%, 99.9% and 99.2%, respectively, surpasses the model 1 which obtains the lower scores at 96.2871%, 99.7% and 96.2% in the same order of metrics. Therefore, the model 2 which uses only the features in the parent and children set of the class is more effective in cancer prediction comparing to the other model which employs all features.

	Accuracy	Precision (Cancer)	Recall (Cancer)
Model 1	96.2871%	99.7%	96.2%
Model 2	99.1749%	99.9%	99.2%

Table 3. The statistics of two models on the 10-fold cross-validation. Model 1: using all features. Model 2: using only the features in the parent and children set of the class

```
Correctly Classified Instances        1167               96.2871 %
Incorrectly Classified Instances        45                3.7129 %
Kappa statistic                          0.8086
Mean absolute error                      0.0358
Root mean squared error                  0.1849
Relative absolute error                 21.2648 %
Root relative squared error             63.8527 %
Total Number of Instances             1212

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.973    0.038    0.722      0.973    0.829      0.820    0.996     0.944     N
                 0.962    0.027    0.997      0.962    0.979      0.820    0.997     1.000     C
Weighted Avg.    0.963    0.028    0.972      0.963    0.965      0.820    0.997     0.995

=== Confusion Matrix ===

    a    b   <-- classified as
  109    3 |    a = N
   42 1058 |    b = C
```

Figure 6.The 10-fold cross-validation results of Naïve Bayes classification – using all features

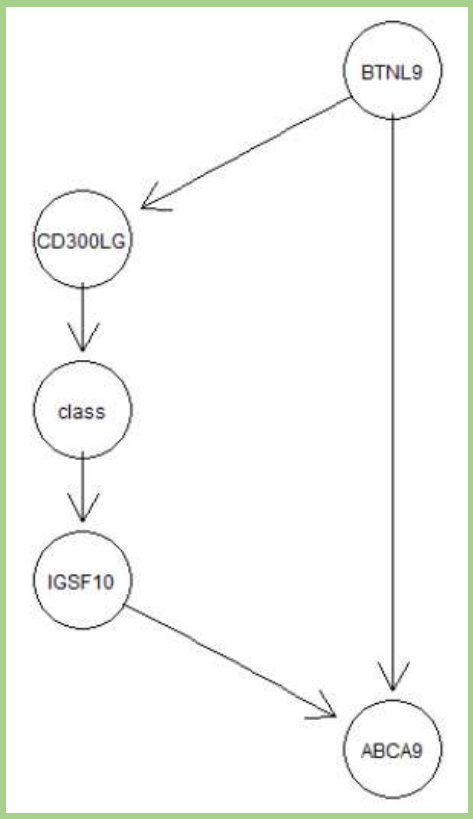
```
Correctly Classified Instances        1202               99.1749 %
Incorrectly Classified Instances        10                0.8251 %
Kappa statistic                          0.9523
Mean absolute error                      0.0088
Root mean squared error                  0.0883
Relative absolute error                  5.2102 %
Root relative squared error             30.5008 %
Total Number of Instances             1212

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.991    0.008    0.925      0.991    0.957      0.953    0.999     0.993     N
                 0.992    0.009    0.999      0.992    0.995      0.953    0.999     1.000     C
Weighted Avg.    0.992    0.009    0.992      0.992    0.992      0.953    0.999     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
  111    1 |    a = N
    9 1091 |    b = C
```

Figure 7. The 10-fold cross validation results of Naïve Bayes classification – using only the features in the parent and children set of the class

Below is the code using PC-simple algorithm (pcSelect) to find the parent and children set of the class variable. The models are built by employing Weka.

```
#get index of the class variable
grep("class", colnames(data_new))
#convert class to numeric to run pcSelect
data_new$class <- as.numeric(data_new$class)
#pcSelect
pcS <- pcSelect(data_new[,51], data_new[,-51], alpha=0.01)
pcS
```

Task 5: Given a Bayesian network as in the below figure



a) Construct the conditional probability tables for the Bayesian network based on data.
b) Estimate the probability of the four genes in the network having high expression levels.
c) Estimate the probability of having cancer when the expression level of CD300LG is high and the expression level of BTNL9 is low.
d) Prove the result in c) mathematically.

a) Construct the conditional probability tables for the Bayesian network based on data.

To create the conditional probability tables, we need to count up the occurrences of each variable value given its parent conditions. For example, we need to count up the occurrences of the CD300LG which has a high expression given the condition that the BTNL9 has a low expression. The conditional probability tables for the Bayesian network based on data are presented in Figure 8.

```
> plist$BTNL9
BTNL9
     high       low
0.1790429 0.8209571
attr(,"class")
[1] "parray" "array"
> plist$CD300LG
       BTNL9
CD300LG     high         low
   high 0.640553 0.008040201
   low  0.359447 0.991959799
attr(,"class")
[1] "parray" "array"
> plist$class
         CD300LG
class         high         low
  normal 0.7414966 0.002816901
  cancer 0.2585034 0.997183099
attr(,"class")
[1] "parray" "array"
> plist$IGSF10
      class
IGSF10 normal     cancer
  high  0.875 0.06454545
  low   0.125 0.93545455
attr(,"class")
[1] "parray" "array"
```

```
> plist$ABCA9
, , IGSF10 = high

      BTNL9
ABCA9        high       low
  high 0.96610169 0.2156863
  low  0.03389831 0.7843137

, , IGSF10 = low

      BTNL9
ABCA9       high        low
  high 0.2626263 0.01694915
  low  0.7373737 0.98305085

attr(,"class")
[1] "parray" "array"
```

Figure 8. Conditional probability tables for the Bayesian network based on data

The code to create conditional probability tables for the Bayesian network based on data is shown as below.

```
hl <- c("high","low")
nc <- c("normal","cancer")

##count the number of instances based on the cause conditions
sum(data_new$BTNL9==1)
sum(data_new$BTNL9==0)

sum(data_new$CD300LG[data_new$BTNL9==1]==1)
sum(data_new$CD300LG[data_new$BTNL9==0]==1)
sum(data_new$CD300LG[data_new$BTNL9==1]==0)
sum(data_new$CD300LG[data_new$BTNL9==0]==0)

sum(data_new$class[data_new$CD300LG==1]=='C')
```

```
sum(data_new$class[data_new$CD300LG==0]=='C')
sum(data_new$class[data_new$CD300LG==1]=='N')
sum(data_new$class[data_new$CD300LG==0]=='N')

sum(data_new$IGSF10[data_new$class=='C']==1)
sum(data_new$IGSF10[data_new$class=='N']==1)
sum(data_new$IGSF10[data_new$class=='C']==0)
sum(data_new$IGSF10[data_new$class=='N']==0)

sum(data_new$ABCA9[data_new$BTNL9==1 & data_new$IGSF10==1]==1)
sum(data_new$ABCA9[data_new$BTNL9==1 & data_new$IGSF10==0]==1)
sum(data_new$ABCA9[data_new$BTNL9==0 & data_new$IGSF10==1]==1)
sum(data_new$ABCA9[data_new$BTNL9==0 & data_new$IGSF10==0]==1)
sum(data_new$ABCA9[data_new$BTNL9==1 & data_new$IGSF10==1]==0)
sum(data_new$ABCA9[data_new$BTNL9==1 & data_new$IGSF10==0]==0)
sum(data_new$ABCA9[data_new$BTNL9==0 & data_new$IGSF10==1]==0)
sum(data_new$ABCA9[data_new$BTNL9==0 & data_new$IGSF10==0]==0)


##put the counts in a proper order to build conditional probability
tables
BTNL <- cptable(~BTNL9, values=c(217,995),levels=hl)
BTNL_CD <- cptable(~CD300LG|BTNL9, values=c(139,78,8,987),levels=hl)
CD_CLASS <- cptable(~class|CD300LG, values=c(109,38,3,1062),levels=nc)
CLASS_IG <- cptable(~IGSF10|class, values=c(98,14,71,1029),levels=hl)
ABCA <- cptable(~ABCA9|BTNL9:IGSF10,values=c(114,4,11,40,
                                             26,73,16,928),levels=hl)
#complie all conditional probability tables
plist <- compileCPT(list(BTNL,BTNL_CD,CD_CLASS,CLASS_IG,ABCA))
plist

plist$BTNL9
plist$CD300LG
plist$class
plist$IGSF10
plist$ABCA9
```

b) Estimate the probability of the four genes in the network having high expression levels.

To estimate the probability of the four genes in the network having high expression levels, we need to calculate the join probability of them having high expression levels. Figure 9 shows the join probabilities of the four genes in the network. Based on this figure, the probability of the four genes in the network having high expression levels is approximately 0.07374.

```
, , IGSF10 = high, ABCA9 = high

      CD300LG
BTNL9          high         low
  high 0.0737360135 0.004155048
  low  0.0009474461 0.011738113

, , IGSF10 = low, ABCA9 = high

      CD300LG
BTNL9          high        low
  high 1.007519e-02 0.01577218
  low  3.742297e-05 0.01288024

, , IGSF10 = high, ABCA9 = low

      CD300LG
BTNL9          high          low
  high 0.002587229 0.0001457912
  low  0.003445258 0.0426840455

, , IGSF10 = low, ABCA9 = low

      CD300LG
BTNL9          high        low
  high 0.028288036 0.04428342
  low  0.002170533 0.74705404
```

Figure 9. Join probabilities of the four genes in the network

Below is the code to get the join probability of the four genes having high expression levels.

```
querygrain(net1, nodes=c("BTNL9","CD300LG","IGSF10","ABCA9"),
type="join")
```

c) Estimate the probability of having cancer when the expression level of CD300LG is high and the expression level of BTNL9 is low.

This is the conditional probability of having cancer given the expression level of CD300LG is high and the expression level of BTNL9 is low. Figure 10 displays the probability of the class variable given the conditions based on CD300LG and BTNL9 gene expression levels. From this figure, the probability of having cancer when the expression level of CD300LG is high and the expression level of BTNL9 is low is 0.2585034.

```
, , class = normal

       BTNL9
CD300LG        high         low
   high 0.741496599 0.741496599
   low  0.002816901 0.002816901

, , class = cancer

       BTNL9
CD300LG      high       low
   high 0.2585034 0.2585034
   low  0.9971831 0.9971831
```

Figure 10. Conditional probabilities of the class variable given CD300LG and BTNL9

Below is the code to get the probability of the class variable given the conditions based on CD300LG and BTNL9.

```
querygrain(net1, nodes=c("class","CD300LG","BTNL9"),
type="conditional")
```

d) Prove the result in c) mathematically.

Let C, D and B' denote, respectively, the class is C, CD300LG is high and BTNL9 is low.

$$\begin{aligned} P(C|D, B') &= P(C|D) \text{ (Markov condition)} \\ &= 0.2585034 \text{ (conditional probability tables)} \end{aligned}$$