Lab 5 - Gr. 14 - Bioinformatics (732A93)

Julius Kittler (julki092), Stefano Toffol (steto820), Saewon Jun (saeju204), Maximilian Pfundstein (maxpf364)

Task 1

Task:

- Install Packages :)

Task 2

Task: The sample data is stored in the workspace as autcon.

- Inspect the dataset and describe it.
- What is the number of features? What is the number of objects in each class?

Description:

- The number of features is 35 (excluding `decision`, the target variable, which is not a feature). All of them are numeric and represent genes.
- The number of rows is 146. Each row represents one male children with autism and healthy ones. There are 82 autistic boys and 64 control observations. This seems like a good enough balance of observations from both classes.

```
## Number of features:   35
```

Table 1: Number of observations by class

Class	Observations
Autism	82
Control	64
Total	146

Task 3

Task: Run `rosetta()` on the default parameters: `autconDefault = rosetta(autcon)`

Use `autconDefault$main` to retrieve the rule table information, assign the result to a separate table. Use `autconDefault$quality` and display the quality statistics of the model:

a. Define what is cross-validation. How many cross-validations are performed in `rosetta` by default?
b. What is the default reduction method? What is it used for?
c. What is the default method of discretization? Describe it shortly. How many discretization bins are calculated?
d. What is the accuracy of the model?
e. How many rules do you obtain? Print the top three most significant rules. Which class get more significant rules? You can assume the rule to be significant if the p-value (PVAL) is lower than 0.05.

a)

Cross-Validation is used to estimate the error of a model under certain conditions (e.g. parameter choices). The supplied data-set is divided randomly (!) into k-folds. The cross-validation iterates `k` times, where each time the `1, 2, ..., k` part is used for validation and all the others parts are used for training.

If we say that `k = 3` then for the first iteration the first fold is used for validation and fold two and three are used for training. In the second iteration the second fold is used for validation and the first and third fold are used for training. So in the last part fold one and two are used for training an fold three is used for validation.

The Rosetta packge is using `10` folds by default.

b)

Default reduction method:

The default reduction method is Johnson. Different reduction methods are: Johnson, Genetic, Holte1R or Manual.

Goal of reduction method:

Reduction methods are used to "reduce" the number of attributes (features). The remaining features must be most informative regarding the target variable. Note that the dimensionality reduction (our main goal) comes at the expense of information loss since variables are removed. (Further note that reduction methods require categorical variables, see next task.)

With terminology from the lecture:

- The reduction method is used for finding a minimal subset which preserves indiscernibility (similarity) between the samples.
- It is basically a way of finding dependencies in the data.

Definition of indiscernibility:

- "Two objects are considered to be indiscernible or equivalent if and only if they have the same values for all attributes in the set. In other words, in terms of the given set of attributes, it is impossible to differentiate the two objects." (http://www2.cs.uregina.ca/~yanzhao/is_complementary.pdf)

c)

Default discretiziation method:

The default is `EqualFrequency`.

Description:

Discretiziation methods are used to transform continuous variables into discrete variables that have a certain number of categories (called bins). Note that reduction methods such as the Johnson method require categorical variables. Since all our features are numerical, they need to be discretized, i.e. transformed to categories (where one category represents a specific range of numerical values).

`EqualFrequency` divides the data into groups where each group has roughly the same size in terms of data-points/samples per bin. A method for finding the best number of bins is creating a histogram and using this the guess a good number.

It seems like the default number of bins is 3 since the default is $discreteParam = 3$ according to the documentation (see `?rosetta`).

For more information http://www.uta.fi/sis/tie/tl/index/Datamining6.pdf.

d)

The accuracy of the model is 82.18%. In other words, 82.18% of the observations in the training data are classified correctly by the fitted model.

```
## Accuracy:  0.821818
```

Definition of accuracy:

The accuracy can be taken from the confusion matrix which shows the four classifications:

- control as control (TP)
- control as autism (FP)
- autism as control (FN)
- autism as autism (TN)

The accuracy is calculated by: (TP+TN)/(TP+TN+FP+FN).

It basically tells us the misclassification rate of our model and the α and β errors made.

e)

Table 2: Number of rules by class

Class	Rules
Autism	108
Control	77
Total	191

Table 3: Top 3 Significant Rules

FEATURES	DECISION	PVAL	CUTS_COND	CUT_1	CUT_2	CUT_3
NCKAP5L,234817_at	control	4.8e-06	value<cut,value<cut	1.90584	1.64213	NaN
MAP7,ATXN8OS	control	4.8e-06	value>cut,value<cut	2.51985	2.22742	NaN
ZSCAN18,NPR2	control	4.8e-06	value<cut,cut<value<cut	2.35647	2.54040	2.59265

Task 4

Task: Export the rules to a text file using the `saveLineByLine()` function.

Task 5

Task: Choose further options: Use the VisuNet tool at http://bioinf.icm.uu.se/~visunet/. Upload your rules.

> File format Choose: "Line by line" Minimum Accuracy Default is 0.7, which means that rules with at least 70% accuracy will be used for displaying a network. Minimum Support Default is 1, which means all rules will be included in the network. You can toggle that and see the effects on the network. Threshold (%) Keep it 100 Show top n nodes Leave it blank Color of nodes Choose: "Level of the gene expression" Is this gene data? Yes. Use the autism_annot.txt file

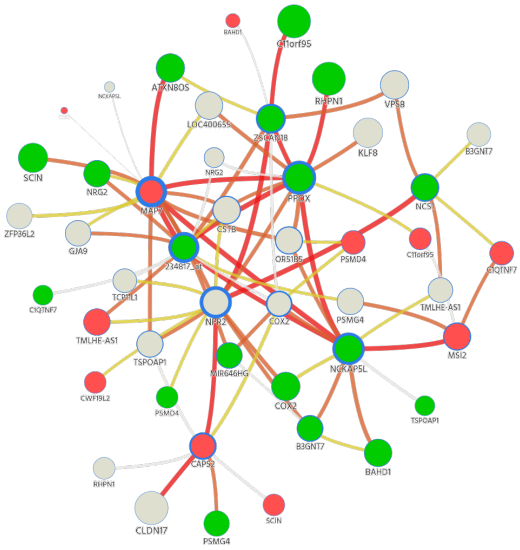

Figure 1: VisuNet - Control

Submit the file to generate a rule-based network. On the left side, there is "Information Bar" where you can select the decision. By clicking on a node, you will be able to see:

a. In the bottom panel: the rules that have the node as one of its conditions.
b. On the right-hand side in the "Selected Node/Edge Info": the name of the node, the number of edges it has, the mean accuracy value and the mean support value. Furthermore, there is information about KEGG Pathways and GO annotations related with the node.

Task 6

Export the networks for the autism and control decisions. Investigate connections present on the networks. Find the strongest connections and the most significant nodes for each decision. Try to interpret these in the context of autism related genes. Hints: Calcium homeostasis is altered in autism disorders (Palmieri et al., 2010). The autism dataset includes a group of genes related to a calcium ion binding (GO:0005509). Take a closer look at SCIN, NCS1 and CAPS2. You may also use the SFARI GENE database containing information about autism-related genes: https://gene.sfari.org/

- Node size: mean support value
- Node color: expression value - green (low), grey (medium), red (high)
- Node borde: number of rules

For control

The most significant node: MAP7

- Name: MAP7=3
- Edges: 16
- Connection: 224.4915
- Mean accuracy: 0.931
- Mean support: 14.875

The strongest connection:

MAP7=3 - NCKAP5L=1 (conn: 38.3052)

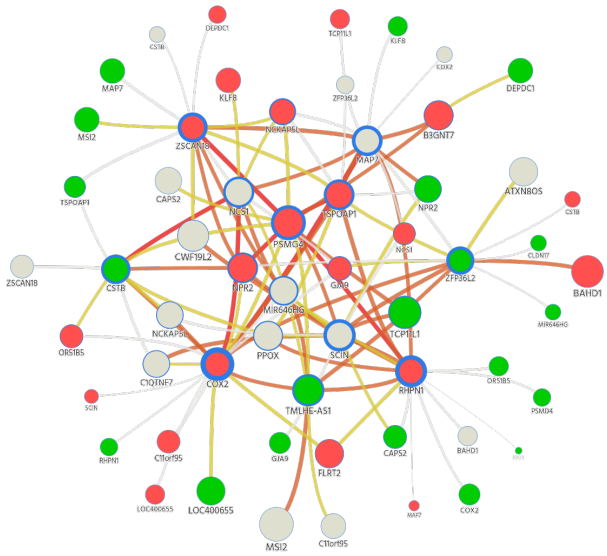

Figure 2: VisuNet - Autism

The most significant node and SCIN, NCS1 and CAPS2

```
##   From    To Support Accuracy
## 1 MAP7  SCIN      16  0.94689
## 2 MAP7  NCS1       -        -
## 3 MAP7 CAPS2       -        -
```

Between SCIN, NCS1 and CAPS2

```
##   From    To Support Accuracy
## 1 NCS1 CAPS2       -        -
## 2 NCS1  SCIN       -        -
## 3 SCIN CAPS2       -        -
```

For autism

The most significant node: COX2

- Name: COX2=3
- Edges: 19
- Connection: 318.58136
- Mean accuracy: 0.994
- Mean support: 16.15

The strongest connection:

PSMG4=3 - RHPN1=3 (conn: 40.09068) COX2=3 - NCS1=2 (conn: 40)

The most significant node and SCIN, NCS1 and CAPS2

```
##   From    To Support Accuracy
## 1 COX2  SCIN       8        1
## 2 COX2  NCS1      20        1
## 3 COX2 CAPS2       -        -
```

Between SCIN, NCS1 and CAPS2

```
##   From    To Support Accuracy
## 1 NCS1 CAPS2      14        1
## 2 NCS1  SCIN      13        1
## 3 SCIN CAPS2      14        1
```

GO:0005509, with the GO term "calcium ion binding" follows the definition - Interacting selectively and non-covalently with calciumions (Ca2+): http://amigo.geneontology.org/amigo/term/GO:0005509.

From SFARI GENE database, we were able to figure out that some genes related to Ca2+-dependent activator protein (with calcium-binding activity) to be associated with autism: https://gene.sfari.org/database/human-gene/CADPS2.

SCIN, NCS1, and CAPS2 are all annotated to calcium ion binding. The tables above indicate that genes related to a calcium ion binding(SCIN, NCS1 and CAPS2) could be interpreted as autism related gene.

Appendix

```
# ------------------------------------------------------------------------------
# Setup
# ------------------------------------------------------------------------------

knitr::opts_chunk$set(fig.width = 7, fig.height = 3, echo = FALSE,
                      warning = FALSE, message = FALSE)

# install.packages("devtools")
# install_github("mategarb/R.ROSETTA")

library(devtools)
library(R.ROSETTA)
library(plyr)
library(dplyr)
library(knitr)


# ------------------------------------------------------------------------------
# Task 2
# ------------------------------------------------------------------------------

cat("Number of features:  ", ncol(autcon)-1, "\n")
# cat("Number of observations:  ", nrow(autcon), "\n")


obs_by_class = as.numeric(table(autcon$decision))
df_table = data.frame(Class = c("Autism", "Control", "Total"),
                      Observations = c(obs_by_class, nrow(autcon)))
knitr::kable(df_table, caption = "Number of observations by class")

# ------------------------------------------------------------------------------
# Task 3
# ------------------------------------------------------------------------------

autconDefault = rosetta(autcon)
rule_table = autconDefault$main
```

```r
quality_statistics = autconDefault$quality

cat("Accuracy: ",  autconDefault$quality$Accuracy.Mean)


# Number of rules ----------------------------------------------------------
rule_table_sub = rule_table[which(rule_table$PVAL < 0.05), ]
rules_by_class = as.numeric(table(rule_table_sub$DECISION))
df_table = data.frame(Class = c("Autism", "Control", "Total"),
                      Rules = c(rules_by_class, nrow(rule_table)))
knitr::kable(df_table, caption = "Number of rules by class")

# Print the top 3 significant rules ----------------------------------------
top3 = head(rule_table[order(rule_table$PVAL),], 3)
top3 = top3 %>% dplyr::select(FEATURES, DECISION, PVAL, CUTS_COND,
                              CUT_1, CUT_2, CUT_3)

knitr::kable(top3, caption = "Top 3 Significant Rules", longtable = TRUE)


# ------------------------------------------------------------------------------
# Task 4
# ------------------------------------------------------------------------------

# Export the rules to a text file using the saveLineByLine() function.
saveLineByLine(autconDefault$main, "outFile.txt")


# ------------------------------------------------------------------------------
# Task 6
# ------------------------------------------------------------------------------

table <- data.frame(From = c("MAP7","MAP7","MAP7")
                    ,To = c("SCIN","NCS1","CAPS2")
                    ,Support = c(16,"-","-")
                    ,Accuracy =c(0.94689,"-","-") )
table
table <- data.frame(From = c("NCS1","NCS1","SCIN")
                    ,To = c("CAPS2","SCIN","CAPS2")
                    ,Support = c("-","-","-")
                    ,Accuracy =c("-","-","-") )
table
table <- data.frame(From = c("COX2","COX2","COX2")
                    ,To = c("SCIN","NCS1","CAPS2")
                    ,Support = c(8,20,"-")
                    ,Accuracy =c(1,1,"-"))
table
table <- data.frame(From = c("NCS1","NCS1","SCIN")
                    ,To = c("CAPS2","SCIN","CAPS2")
                    ,Support = c(14,13,14)
                    ,Accuracy =c(1,1,1))
table
```