Lab 3

Knowledge-Based Systems in Bioinformatics

Rasmus Hammar

Tasks: Discretization

A = (U,A ∪ {d})

Table 1. Decision system A discretized using equal frequency with three levels.

index	Cuts feature_1	decision
x_6	[91, 105)	No
x_7	[91, 105)	No
x_1	[91, 105)	No
x_9	[45, 91)	No
x_8	[45, 91)	Yes
x_2	[45, 91)	No
x_4	[21, 45)	Yes
x_3	[21, 45)	Yes
x_5	[21, 45)	Yes

$POS_{feature\_1}(\{d\}) = \{\{x_1, x_6, x_7\}, \{x_3, x_4, x_5\}\} => \gamma = \frac{6}{9}$

Table 2. Decision system A discretized using naive method.

index	Cuts feature_1	decision
x_6	[77.5, 105)	No
x_7	[77.5, 105)	No
x_1	[77.5, 105)	No
x_9	[77.5, 105)	No
x_8	[61.5, 77.5)	Yes
x_2	[45, 61.5)	No
x_4	[21, 45)	Yes

index	Cuts feature_1	decision
x_3	[21, 45)	Yes
x_5	[21, 45)	Yes

$POS_{feature\_1}(\{d\}) = \{\{x_1, x_6, x_7, x_9\}, \{x_8\}, \{x_2\}, \{x_3, x_4, x_5\}\} => \gamma = \frac{9}{9} = 1$

Comparing $\gamma = \frac{6}{9}$ and $\gamma = \frac{9}{9} = 1$, the naive method would be better, but the equal frequency method is probably good enough and more generalizable, and is therefore the better option.

B = (U,B ∪ {d})

Table 3. Decision system A discretized using equal frequency with three levels.

index	Cuts width [mm]	decision
x_8	[0.67, 0.97)	A
x_1	[0.67, 0.97)	B
x_4	[0.67, 0.97)	A
x_3	[0.29, 0.67)	A
x_2	[0.29, 0.67)	A
x_9	[0.29, 0.67)	B
x_7	[0.04, 0.29)	B
x_6	[0.04, 0.29)	B
x_5	[0.04, 0.29)	A
x_{10}	[0.04, 0.29)	B

$POS_{width}(\{d\}) = \emptyset => \gamma = 0$

Table 4. Decision system A discretized using naive method.

index	Cuts width [mm]	decision
x_8	[0.89, 0.97)	A
x_1	[0.75, 0.89)	B
x_4	[0.39, 0.75)	A
x_3	[0.39, 0.75)	A
x_2	[0.39, 0.75)	A
x_9	[0.17, 0.39)	B
x_7	[0.17, 0.39)	B
x_6	[0.17, 0.39)	B

index	Cuts width [mm]	decision
x_5	[0.08, 0.17)	A
x_{10}	[0.04, 0.08)	B

$POS_{width}(\{d\}) = \{\{x_8\}, \{x_1\}, \{x_4, x_3, x_2\}, \{x_9, x_7, x_6\}, \{x_5\}, \{x_{10}\}\} => \gamma = 1$

Comparing $\gamma = 0$ and $\gamma = 1$, the naive method would be better. Equal frequency method is probably good enough and more generalizable, and could probably be reduced to [0.04, 0.29) and [0.29, 0.97).

Tasks: R.ROSETTA

1

```
rlang::is_installed("R.ROSETTA")
```

```
[1] TRUE
```

2

```
library(R.ROSETTA)
autcon <- autcon
str(autcon)
```

```
'data.frame':   146 obs. of  36 variables:
 $ MAP7     : num  2.36 2.52 2.6 2.37 2.03 ...
 $ COX2     : num  3.76 3.74 3.76 3.76 3.76 ...
 $ NCKAP5L  : num  1.97 2.02 1.89 2.07 1.97 ...
 $ ZSCAN18  : num  2.42 2.32 2.49 2.38 2.33 ...
 $ RHPN1    : num  2.63 2.51 2.67 2.7 2.65 ...
 $ PPOX     : num  2.05 2.07 2.03 2.17 2.09 ...
 $ NPR2     : num  2.61 2.58 2.54 2.54 2.55 ...
 $ NCS1     : num  2.29 2.4 2.28 2.47 2.42 ...
 $ PSMG4    : num  2.34 2.42 2.46 2.51 2.45 ...
 $ SCIN     : num  1.66 1.67 1.63 1.7 1.53 ...
 $ CSTB     : num  2.39 2.39 2.26 2.46 2.41 ...
 $ TSPOAP1  : num  2.67 2.63 2.64 2.56 2.63 ...
 $ TCP11L1  : num  2.26 2.45 2.28 2.24 2.3 ...
 $ 234817_at: num  1.61 1.63 1.56 1.75 1.67 ...
 $ TMLHE-AS1: num  1.94 2.01 1.89 2.19 1.83 ...
 $ PSMD4    : num  3.34 3.33 3.34 3.4 3.33 ...
 $ ZFP36L2  : num  3.16 3.05 3.05 2.97 2.8 ...
 $ B3GNT7   : num  2.51 2.55 2.57 2.35 2.54 ...
 $ MSI2     : num  2.26 2.19 2.41 1.72 1.85 ...
```

```
$ CAPS2    : num  1.25 1.14 1.31 1.26 1.33 ...
$ MIR646HG : num  1.59 1.63 1.65 1.6 1.75 ...
$ CLDN17   : num  2.33 2.41 2.41 2.41 2.44 ...
$ BAHD1    : num  2.93 2.95 3.01 2.89 2.93 ...
$ OR51B5   : num  1.95 1.87 1.86 1.9 1.87 ...
$ C11orf95 : num  2.78 2.76 2.79 2.55 2.8 ...
$ ATXN8OS  : num  2.27 2.26 2.21 2.36 2.23 ...
$ NRG2     : num  2.28 2.18 2.4 2.37 2.41 ...
$ LOC400655: num  1.44 1.59 1.28 1.52 1.42 ...
$ GJA9     : num  2.09 2.14 2.01 2.16 2.09 ...
$ VPS8     : num  1.84 1.81 1.82 1.84 1.82 ...
$ FLRT2    : num  1.59 1.51 1.61 1.56 1.43 ...
$ C1QTNF7  : num  1.32 1.42 1.39 1.28 1.54 ...
$ KLF8     : num  2.23 2.17 2.23 2.22 2.29 ...
$ CWF19L2  : num  1.24 1.3 1.34 1.4 1.4 ...
$ DEPDC1   : num  1.62 1.77 1.73 1.75 1.76 ...
$ decision : Factor w/ 2 levels "autism","control": 2 2 2 2 2 2 2 2 2 2 ...
```

a) n features = 35

b) n objects per class = 146

c) I suppose? Hard to say.

3

```
autconJohnson <- rosetta(autcon, roc = T)
decision_table <- autconJohnson$main
quality_stats <- autconJohnson$quality
```

4

a)

Cross-validation is the method of splitting a dataset into roughly equal size subsets and reserving one for testing while training on the rest.

Default cross-validations in R.ROSETTA::rosetta is 10.

b)

Default reduction method is Johnson.

The Johnson reduction method is a heuristic method for finding reducts.

c)

Default discretization method is equal frequency.

Equal frequency splits the (ordered) dataset into k-subsets with equal number of objects.

d)

Default number of discretization levels used is 3.

e)

Mean accuracy of the model is 0.804.

f)

To statistically evaluate the model, the accuracy is compared to that of random guessing.

g)

There are a total of 193 rules in the model for decisions “control” and “autism”.

```
top_12_rules_autism <- decision_table |>
  dplyr::filter(decision == "autism", pValue < 0.05) |>
  dplyr::arrange(pValue) |>
  viewRules()
top_12_rules_autism |>
  head(12) |>
  knitr::kable()
```

Table 1: Top 12 (of 67 with p < 0.05) rules for decision “autism” with lowest p-value.

rule	length	accuracy	support	pValue
IF RHPN1(3) THEN autism	1	0.88372	38	0.0000282
IF COX2(3) THEN autism	1	0.87719	37	0.0000579
IF TSPOAP1(3) THEN autism	1	0.88095	37	0.0000579
IF MAP7(2) AND COX2(3) THEN autism	2	0.97826	22	0.0001413
IF ZSCAN18(3) AND PSMG4(3) THEN autism	2	1.00000	21	0.0002896
IF PSMG4(3) AND TSPOAP1(3) THEN autism	2	1.00000	20	0.0005886
IF NCS1(2) AND CSTB(1) THEN autism	2	1.00000	20	0.0005886
IF NPR2(3) AND PSMG4(3) THEN autism	2	1.00000	20	0.0005886
IF SCIN(2) AND ZFP36L2(1) THEN autism	2	1.00000	19	0.0011865
IF TMLHE-AS1(1) AND MSI2(2) THEN autism	2	1.00000	19	0.0011865
IF SCIN(2) AND CWF19L2(2) THEN autism	2	1.00000	19	0.0011865
IF MAP7(2) AND TCP11L1(1) THEN autism	2	1.00000	19	0.0011865

h)

Accuracy is how often the rule is correct out of the objects is is applied to.

Support is how many objects the rule is applied to.

Cuts are the conditions for where to split a feature to differentiate between decisions.

i)

Length of rules range from 1 to 2.

The quality of rules is pretty good. They depend on few features and have high accuracy.

Rules are structured as such that "IF gene THEN [decision]", where the (value-cut comparison) is one of `1 => value < cut_1`, `2 => cut_1 < value > cut_2`, or `3 => value > cut_2`. Interpreting the rule is done by comparing the value for the gene against the cut.

```
top_12_rules_autism |>
  head(12) |>
  knitr::kable()
```

Table 2: Same as Table 1 for some reason. (Are the instructions incorrect?)

rule	length	accuracy	sup-port	pValue
IF RHPN1(3) THEN autism	1	0.88372	38	0.0000282
IF COX2(3) THEN autism	1	0.87719	37	0.0000579
IF TSPOAP1(3) THEN autism	1	0.88095	37	0.0000579
IF MAP7(2) AND COX2(3) THEN autism	2	0.97826	22	0.0001413
IF ZSCAN18(3) AND PSMG4(3) THEN autism	2	1.00000	21	0.0002896
IF PSMG4(3) AND TSPOAP1(3) THEN autism	2	1.00000	20	0.0005886
IF NCS1(2) AND CSTB(1) THEN autism	2	1.00000	20	0.0005886
IF NPR2(3) AND PSMG4(3) THEN autism	2	1.00000	20	0.0005886
IF SCIN(2) AND ZFP36L2(1) THEN autism	2	1.00000	19	0.0011865
IF TMLHE-AS1(1) AND MSI2(2) THEN autism	2	1.00000	19	0.0011865
IF SCIN(2) AND CWF19L2(2) THEN autism	2	1.00000	19	0.0011865
IF MAP7(2) AND TCP11L1(1) THEN autism	2	1.00000	19	0.0011865

```
top_12_rules_autism_cuts <- decision_table |>
  dplyr::filter(decision == "autism", pValue < 0.05) |>
  dplyr::arrange(pValue) |>
  dplyr::select(c(
    features,
    levels,
    tidyselect::starts_with("cut"),
    decision
  )) |>
```

```
  dplyr::select(-cuts)
top_12_rules_autism_cuts |>
  head(12) |>
  knitr::kable()
```

Table 3: Cuts for features as seen in Table 1

features	levels	cut1	cut2	cut3	cut4	decision
RHPN1	3	2.68500	NaN	NaN	NaN	autism
COX2	3	3.76430	NaN	NaN	NaN	autism
TSPOAP1	3	2.71400	NaN	NaN	NaN	autism
MAP7,COX2	2,3	2.38750	2.51800	3.76400	NaN	autism
ZSCAN18,PSMG4	3,3	2.43250	2.47250	NaN	NaN	autism
PSMG4,TSPOAP1	3,3	2.47433	2.71289	NaN	NaN	autism
NCS1,CSTB	2,1	2.30150	2.34875	2.34425	NaN	autism
NPR2,PSMG4	3,3	2.59400	2.47450	NaN	NaN	autism
SCIN,ZFP36L2	2,1	1.52750	1.59750	2.93200	NaN	autism
TMLHE-AS1,MSI2	1,2	1.91950	1.93150	2.06350	NaN	autism
SCIN,CWF19L2	2,2	1.52825	1.59625	1.24750	1.30375	autism
MAP7,TCP11L1	2,1	2.37650	2.51350	2.23600	NaN	autism

j)

Class “control” had one more significant rule than “autism”, with 68 vs 67 out of 135 rules with $p < 0.05$.

The fraction of significant rules are:

- control: $\frac{68}{135} \approx 0.504$
- autism: $\frac{67}{135} \approx 0.496$

```
top_12_rules_control <- decision_table |>
  dplyr::filter(decision == "control", pValue < 0.05) |>
  dplyr::arrange(pValue) |>
  viewRules()
top_12_rules_control |>
  head(12) |>
  knitr::kable()
```

Table 4: Top 12 rules (of 68 with p < 0.05) for decision "control" with lowest p-value.

rule	length	accuracy	support	pValue
IF NCKAP5L(1) AND PPOX(1) THEN control	2	0.95119	20	1.90e-06
IF ZSCAN18(1) AND NPR2(2) THEN control	2	0.98521	19	5.20e-06
IF CAPS2(3) AND CLDN17(2) THEN control	2	1.00000	19	5.20e-06
IF PPOX(1) AND LOC400655(2) THEN control	2	1.00000	18	1.46e-05
IF NCKAP5L(1) AND 234817_at(1) THEN control	2	0.97368	18	1.46e-05
IF MAP7(3) AND ATXN8OS(1) THEN control	2	1.00000	18	1.46e-05
IF NPR2(2) AND TSPOAP1(2) THEN control	2	0.94591	18	1.46e-05
IF MAP7(3) AND NCKAP5L(1) THEN control	2	0.95806	20	2.61e-05
IF MAP7(3) AND 234817_at(1) THEN control	2	0.95227	20	2.61e-05
IF NPR2(2) AND CAPS2(3) THEN control	2	1.00000	17	4.01e-05
IF ZSCAN18(1) AND C11orf95(1) THEN control	2	0.92659	19	6.96e-05
IF RHPN1(1) AND PPOX(1) THEN control	2	0.95000	19	6.96e-05

5

```
plotMeanROC(autconJohnson)
```

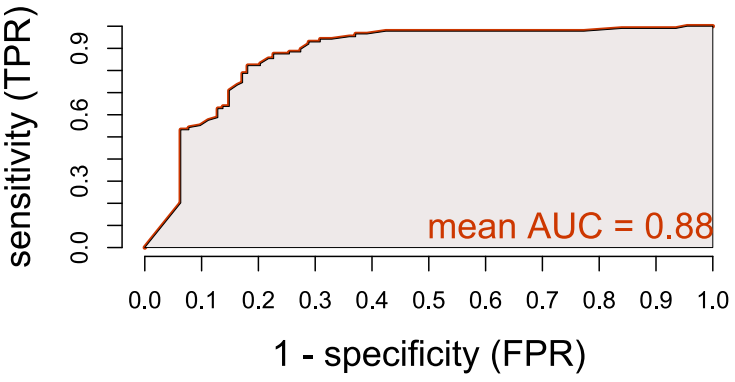

Figure 1: ROC for model.

a) Mean AUC is 0.877.

b)

x-axis shows the false positive rate.

y-axis shows the true positive rate.

c)

The area under the curve (AUC) measures how good the model is at differentiating between decision classes, with AUC = 1 being a perfect consistent system and AUC = 0.5 meaning random guessing.

d)

Because it's robust to skewed data, it becomes less sensitive to minority/rare classes.

e)

Best threshold depends on if it's better to get more false positives but catch all true positives (disease tests where further testing may be done to be sure), or if it's better to get less false positives but miss some true positives (test is very expensive or invasive, but a false negative is no big deal).

6

a)

```
rules_features <- getFeatures(decision_table)
```

b)

Intersecting genes.

```
features_intersect <- intersect(
  rules_features$features$control,
  rules_features$features$autism
)
print(features_intersect)
```

```
 [1] "MAP7"      "NPR2"      "234817_at" "PPOX"      "NCKAP5L"   "ZSCAN18"
 [7] "COX2"      "CAPS2"     "MSI2"      "TCP11L1"   "TSPOAP1"   "B3GNT7"
[13] "CSTB"      "OR51B5"    "PSMD4"     "SCIN"      "MIR646HG"  "NCS1"
[19] "PSMG4"     "RHPN1"     "BAHD1"     "C11orf95"  "C1QTNF7"   "CWF19L2"
[25] "GJA9"      "LOC400655" "TMLHE-AS1" "VPS8"      "CLDN17"    "DEPDC1"
[31] "KLF8"      "ZFP36L2"
```

c)

Unique genes for control.

```
print(unique(rules_features$features$control))
```

```
 [1] "MAP7"      "NPR2"      "234817_at" "PPOX"      "NCKAP5L"   "ZSCAN18"
 [7] "COX2"      "CAPS2"     "MSI2"      "TCP11L1"   "TSPOAP1"   "B3GNT7"
[13] "CSTB"      "OR51B5"    "PSMD4"     "SCIN"      "ATXN8OS"   "MIR646HG"
[19] "NCS1"      "NRG2"      "PSMG4"     "RHPN1"     "BAHD1"     "C11orf95"
[25] "C1QTNF7"   "CWF19L2"   "GJA9"      "LOC400655" "TMLHE-AS1" "VPS8"
[31] "CLDN17"    "DEPDC1"    "KLF8"      "ZFP36L2"
```

Unique genes for autism.

```
print(unique(rules_features$features$autism))
```

```
 [1] "COX2"      "RHPN1"     "MAP7"      "CSTB"      "SCIN"      "TSPOAP1"
 [7] "ZSCAN18"   "PSMG4"     "ZFP36L2"   "NPR2"      "NCKAP5L"   "NCS1"
[13] "TMLHE-AS1" "MIR646HG"  "PPOX"      "CAPS2"     "GJA9"      "PSMD4"
[19] "TCP11L1"   "C11orf95"  "OR51B5"    "B3GNT7"    "C1QTNF7"   "CWF19L2"
[25] "DEPDC1"    "LOC400655" "MSI2"      "234817_at" "BAHD1"     "CLDN17"
[31] "FLRT2"     "KLF8"      "VPS8"
```

7

a)

```
autconJohnson_recalculated <- recalculateRules(autcon, decision_table)
```

b)

Column `supportSetRHS` contains the objects that the rule lead to the right decision for.

c)

The heatmap shows the most significant rule for decision “control”, which involves genes MAP7 and NPR2. Color gradient inside the cells indicate normalized values and the left side legend bar groups the objects by if the rule supports them or not.

```
decision_recalc_control <- autconJohnson_recalculated |>
  dplyr::filter(decision == "control")
plotRule(autcon, decision_recalc_control)
```

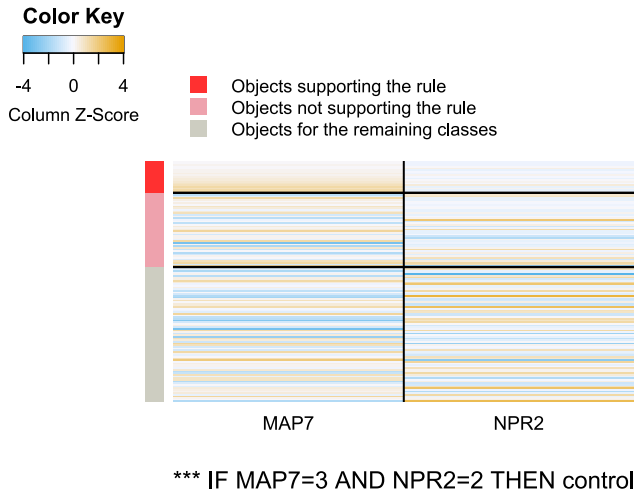

Figure 2: Recalculated model "most significant" rule for decision = control (genes MAP7 & NPR2).

c)

Objects supporting the rule have less variation compared to the rest of the objects.

```
plotRule(
  autcon,
  decision_recalc_control,
  type = "box",
  label = c("high", "medium")
)
```

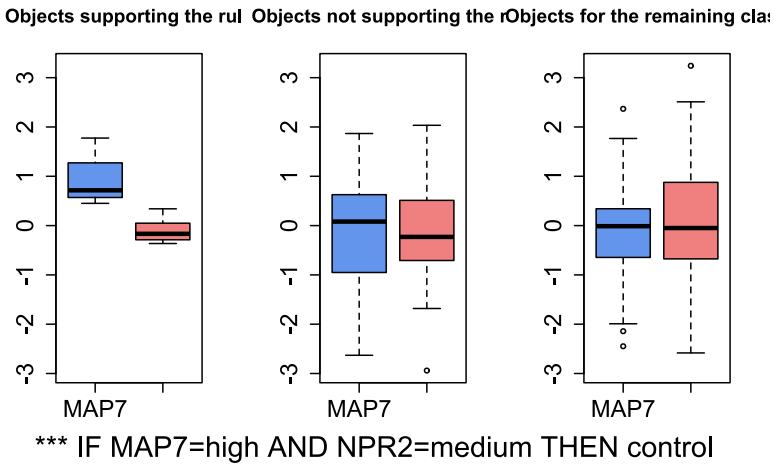

Figure 3: Same rule as for Figure 2.

8

a) Yes.

b)

```
rules_recalc_nonSignif <- decision_recalc_control |>
  dplyr::filter(pValue > 0.05)
```

c)

```
clusterRules(
  autcon,
  rules_recalc_nonSignif
)
```

```
Loading required package: pheatmap
```

```
Warning: package 'pheatmap' was built under R version 4.5.2
```

```
Loading required package: tidyverse
```

```
Warning: package 'tibble' was built under R version 4.5.2
```

```
Warning: package 'tidyr' was built under R version 4.5.2
```

```
Warning: package 'purrr' was built under R version 4.5.2
```

```
Warning: package 'dplyr' was built under R version 4.5.2
```

```
Warning: package 'stringr' was built under R version 4.5.2
```

```
Warning: package 'lubridate' was built under R version 4.5.2
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0
──
✔ dplyr     1.1.4     ✔ readr     2.1.6
✔ forcats   1.0.1     ✔ stringr   1.6.0
✔ ggplot2   4.0.2     ✔ tibble    3.3.1
✔ lubridate 1.9.4     ✔ tidyr     1.3.2
✔ purrr     1.2.1
```

```
── Conflicts ─────────────────────────────────────── tidyverse_conflicts()
──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
Loading required package: grid

Loading required package: gridExtra


Attaching package: 'gridExtra'


The following object is masked from 'package:dplyr':

    combine
```

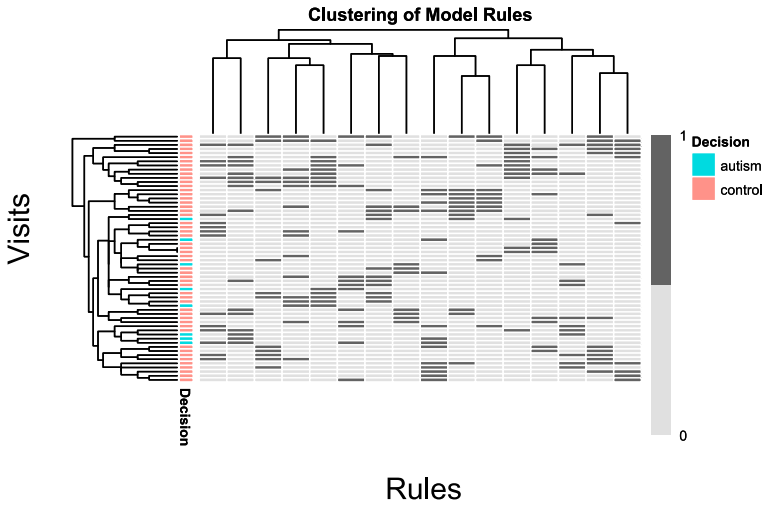

Figure 4: Non-significant rules (p > 0.05) where dark gray = 1 meaning the object supports the rule and light gray = 0 meaning no support.

d)

Described the heatmap annotation in the figure text.

e)

There is no pattern in Figure 4, which is expected since the remaining rules are non-significant.

```
rules_recalc_signif <- decision_recalc_control |>
  dplyr::filter(pValue < 0.05)
clusterRules(
  autcon,
```

```
  rules_recalc_signif
)
```

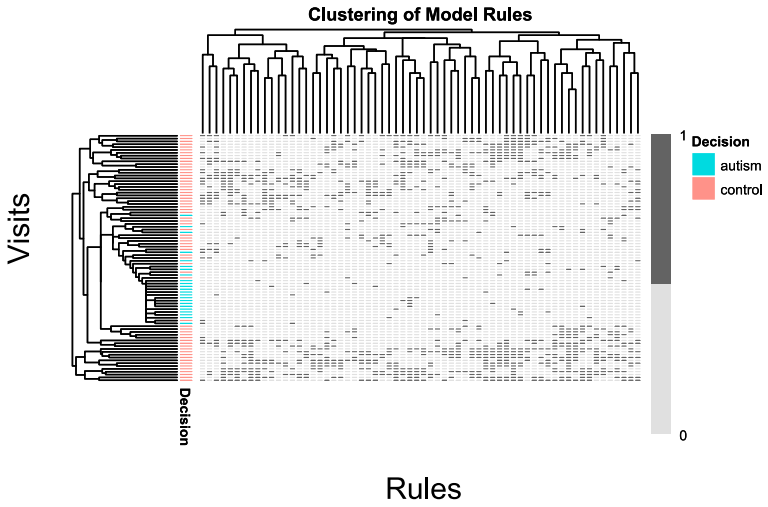

Figure 5: A more interesting case where non-significant ($p > 0.05$) rules were pruned (unlike in the lab instructions).

In Figure 5, there are three clear clusters with one consisting of autism objects with a lower number of objects supporting the rules.

f)

More control objects seem to have more support from the rules compared to autism.

g)

Rules are more effective in classifying control objects than autism objects.