Nested Loop Cross Validation for Classification using `nlcv`

Willem Talloen

August 16, 2024

1 Introduction

Microarrays may provide clinicians with valuable biomarkers for disease status or treatment sensitivity. Extracting molecular signatures from the high-dimensional data is however a difficult and complex task. The aim is to select a small number of features (genes) with high predictive accuracy [2]. *Classification* algorithms in combination with *feature selection* methods serve this purpose.

One of the biggest problems of classification models is the *low reproducibility* of their results [5], primarily due to *overfitting*. To fully consider the pitfall of overfitting, one should do more than only carefully avoiding potential selection bias [1], as obtained results may also be specific for the used training samples or for the used selection/classification methods;

- From seven large studies aimed at predicting prognosis of cancer patients by microarrays, [3] found that the obtained results depended on how the patients were selected in the training set. Because of problems with this aspect of study design, the outcomes of these studies were biased.
- Potential differences in results between classification models will cause signatures to depend highly on the algorithm used to extract the signature. Therefore, the scientist has to validate how much of the reported discrimination can be attributed to a real biological difference: the scientist needs to *disentangle biology and algorithm*[5].

The package `nlcv` provides a framework for robust and reproducible classification while keeping a high sensitivity to detect subtle signals. Its main benefits are;

1. It uses and compares multiple classification models based on the original code from the authors.
2. It estimates predictive accuracy not once, but on multiple random partitions into training and test sets.
3. A balanced partitioning avoids that samples of a certain class are absent in either training or test set in small sample sized studies.
4. A clear separation of feature selection and classification algorithms allows more flexibility and an assessment of the relative impact of the two steps.
5. The use of both a univariate (gene-by-gene) t-test ranking and a multivariate Random forest variable importance ranking allow selecting genes either in isolation as well as in combination.
6. There is no selection bias, as feature selection and classification are applied in combination on the training samples only.

2 Methodology

The package `nlcv` implements two nested cross-validation loops to estimate the misclassification error rate (MCR). Cross-validation is an appropiate instrument to estimate the MCR [4]. First, the models for feature selection and classification are combined to create a "complete classification procedure" [5]. Then the outer cross-validations are performed with this complete classification procedure.

The outer cross-validation loop is used to estimate the misclassification rate and the inner cross-validation loop is used to tune the optimal parameters for a given complete classification procedure [5]. The test set used for estimating the MCR is not included in the cross-validation loop for the tuning of the parameters (see Figure 1). So, as an example, applying 20 outer CV loops is random partitioning the data 20 times into training and test sets, and obtaining 20 MCR based on these 20 different test sets. The default setting of `nlcv` is to split the data into $2/3$ training and $1/3$ test.

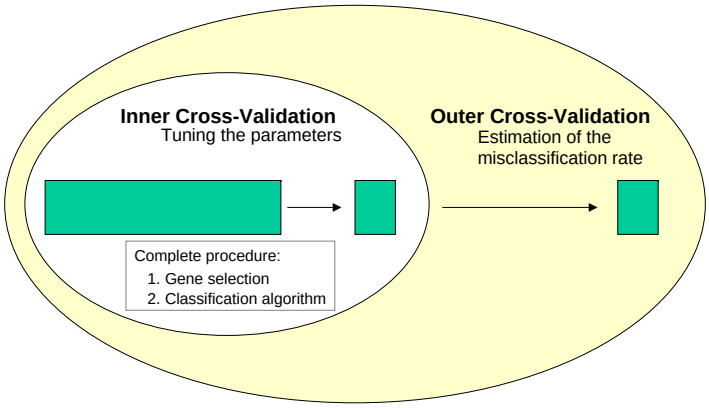

Figure 1: Scheme of nested loop cross-validation, showing that feature selection and classification are within a outer cross-validation loop and therefore do not see the test samples of that outer CV loop.

Feature ranking is done once in every outer cross-validation loop. Then, based on the cut-offs for `x` number of features prespecified by the user, the top `x` features are used for inner cross-validation loops (see Figure 2). At the moment, two feature selection techniques are implemented: t-test and random forest variable importance for ranking the features on relevance in respectively isolation and combination.

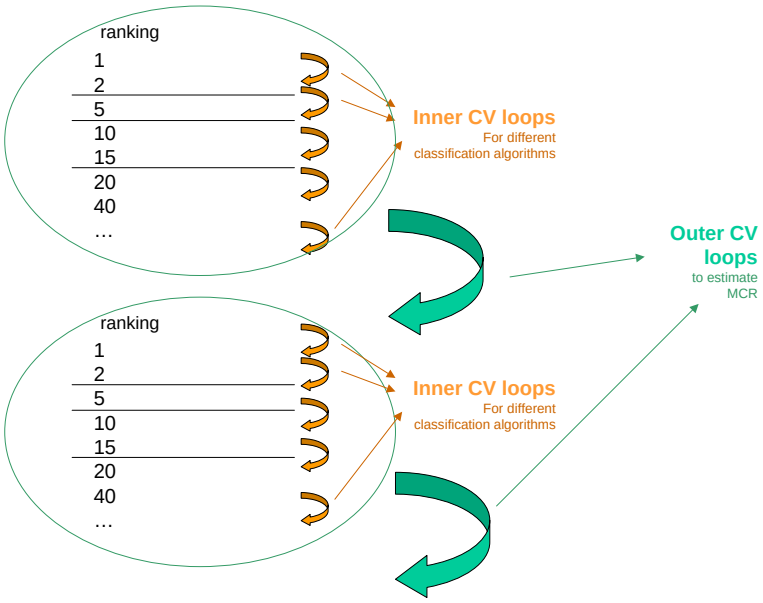

Figure 2: Scheme of nested loop cross-validation, showing that feature ranking is done only once in every outer cross-validation loop. Selection is done as many times as the user specified how many genes should be considered.

This package makes use of state-of-the-art classification algorithms from existing packages uploaded via the wrapper package `MLInterfaces`.

3 Results

3.1 Data Simulation

First we load the package.

```
> library(nlcv)
```

Second, we simulate 6 datasets with different properties using the function `simulateData`. More specifically we generate 40 samples and 1000 features containing;

1. Random data to check whether the obtained results are not over-optimistic.

   ```
   > EsetRandom <- simulateData(nCols = 40, nRows = 1000, nEffectRows = 0, nNoEffectCols = 0)
   ```

2. Data including 10 strongly differentially expressed genes to check whether the signal is detected.

   ```
   > EsetStrongSignal <- simulateData(nCols = 40, nRows = 1000, nEffectRows = 10,
   +         nNoEffectCols = 0, betweenClassDifference = 3, withinClassSd = 0.5)
   ```

3. Data including 5 moderately differentially expressed genes to check whether a more subtle signal can also be detected.

   ```
   > EsetWeakSignal <- simulateData(nCols = 40, nRows = 1000, nEffectRows = 5,
   +         nNoEffectCols = 0, betweenClassDifference = 1, withinClassSd = 0.6)
   ```

4. Data including 5 strongly differentially expressed genes, with some samples having an expression profile like in the opposite class. This to check how outlying samples affect the obtained results. Data with group A having 5 samples behaving like group B.

   ```
   > EsetStrongHeteroSignal <- simulateData(nCols = 40, nRows = 1000, nEffectRows = 5,
   +         nNoEffectCols = 5, betweenClassDifference = 3, withinClassSd = 0.5)
   ```

5. Data including 5 moderately differentially expressed genes, with some samples having an expression profile like in the opposite class. This to check how previous study behaves if the signal is weaker.

   ```
   > EsetWeakHeteroSignal <- simulateData(nCols = 40, nRows = 1000, nEffectRows = 5,
   +         nNoEffectCols = 5, betweenClassDifference = 1, withinClassSd = 0.6)
   ```

We generate 20 samples from class 'A' and 20 from class 'B'. The rows with simulated difference between classes A and B occur in the top of the dataset, and are consequently referred to by Gene.1, Gene.2, etc. The columns that are simulated as belonging to the opposite class occur in the beginning, and are consequently called Sample1, Sample2 to sampleN when N samples were simulated as outliers.

As an illustration, the expression levels of the first gene of the data set *EsetStrongHeteroSignal* are shown in 17. This is clearly a gene with a strong signal (mean difference of 3), and there are clearly three samples (samples1 to 5) that behave as samples from group B.

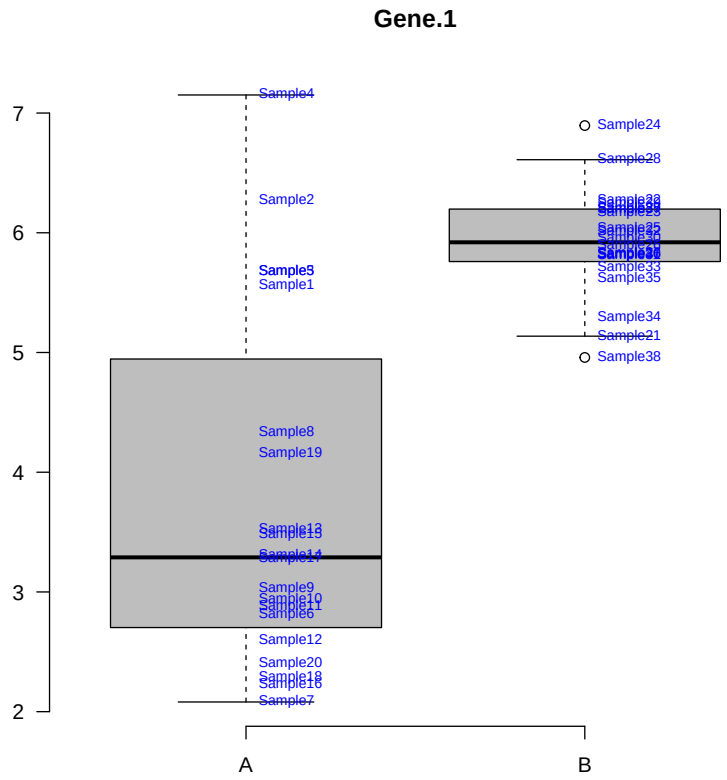

Figure 3: The expression levels of the first gene of the data set EsetStrongHeteroSignal.

3.2 Classification

Let's now run the nlcv. Here we use 2 runs whith t-test selection as an illustration as the number of runs determines the computation time, and as Random Forest selection is computationally more intensive.

```
>         nlcvTT_SS <- nlcv(EsetStrongSignal, classVar = "type", nRuns = 2,
+                 fsMethod = "t.test", verbose = TRUE)
```

As 2 runs is insufficient to obtain accurate estimates of MRC, we use results of previously ran nclv based on 20 runs The computation time of the calculations of the 8 nlcv's, all using 20 runs, was around 1h30 on a laptop.

3.3 Random data without signal.

Let's first simulate a completely random data set. This to check whether the obtained results are indeed robust against overfitting.

Figure 4, created with the code below, shows that all classifiers for all gene set sizes have an average MCR of 0.5. Feature selection on t-test even generates on average worse MCR (0.58) than expected by chance.

```
> # plot MCR versus number of features
> pdf(file = "./graphs/mcrPlot_nlcv_R.pdf", width = 10, height = 5)
>   layout(matrix(1:4, ncol = 2), height = c(6, 1, 6, 1))
>   mcrPlot_RF_R <- mcrPlot(nlcvRF_R, plot = TRUE, optimalDots = TRUE,
+         layout = FALSE, main = 'RF selection')
>   mcrPlot_TT_R <- mcrPlot(nlcvTT_R, plot = TRUE, optimalDots = TRUE,
+          layout = FALSE, main = 'T selection')
>   layout(1)
> dev.off()
```

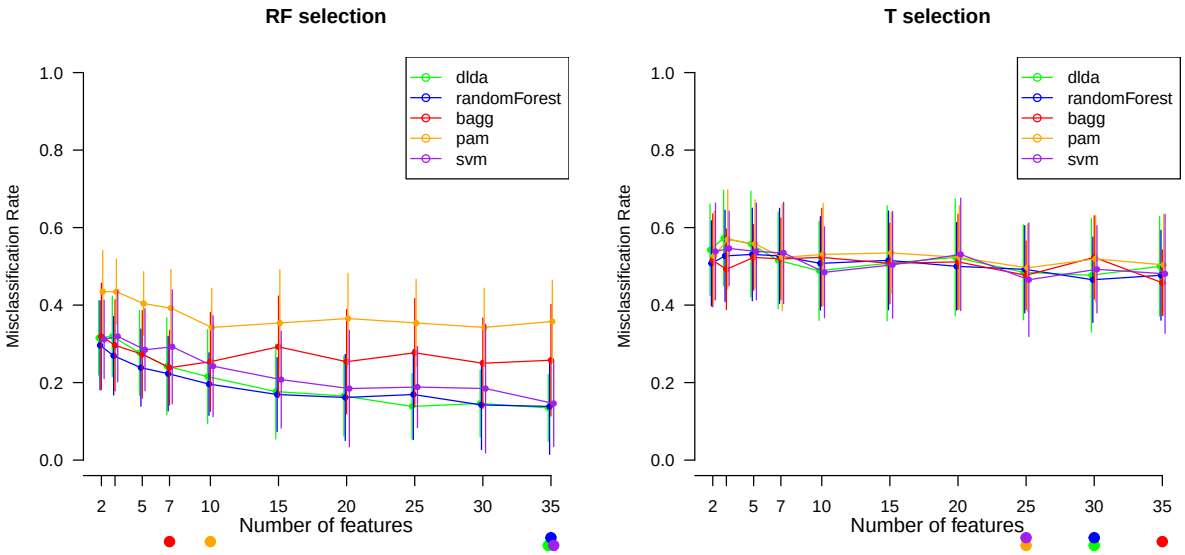

Figure 4: The mean misclassification rate (mcr) and its standard error for each classification technique and number of features, calculated across the runs of the nested loop cross-validation.

Figure 5, created with the code below, shows the probability scores for Random Forest with Variable Importance selection with a gene set of 5 genes. There are as many samples good as bad classified, and more importantly, no single sample (except sample 35) has been always correctly classified or always misclassified.

```
> pdf(file = "./graphs/ScoresPlot_nlcv_R.pdf", width = 10, height = 6)
> scoresPlot(nlcvRF_R, "randomForest", 5)
> dev.off()
```

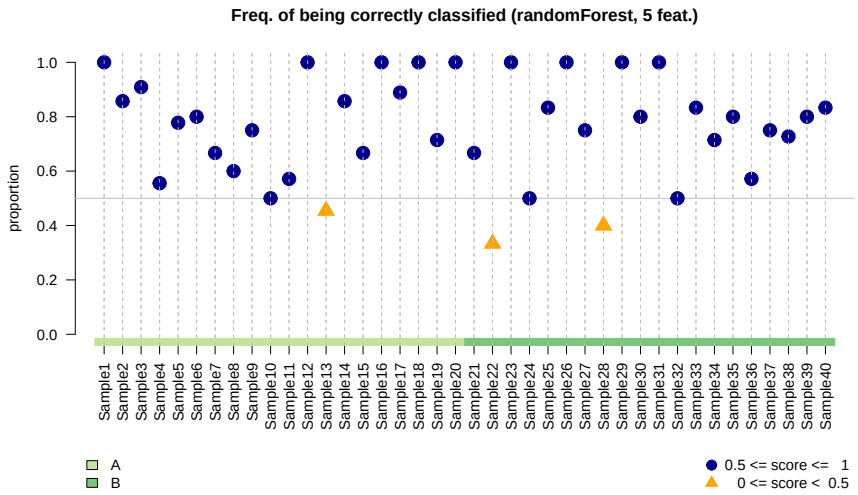

Figure 5: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

Finally, the top 10 most frequently selected genes by RF variable importance 1 shows that no gene is frequently selected. Only gene 410 is half of the time selected.

	percentage.Var1	percentage.Freq
1	Gene.940	70.00
2	Gene.772	60.00
3	Gene.695	55.00
4	Gene.355	50.00
5	Gene.207	40.00
6	Gene.198	35.00
7	Gene.502	35.00
8	Gene.556	35.00
9	Gene.187	30.00
10	Gene.664	30.00

Table 1: Top 10 features across all runs of the nested loop cross-validation.

3.4 Data containing a strong signal.

Next, we simulate a similar random data set (40x10000), but this time we have introduced 10 genes that are strongly differentially expressed between the groups A and B (see Gene.1 in Figure 6 as an example).

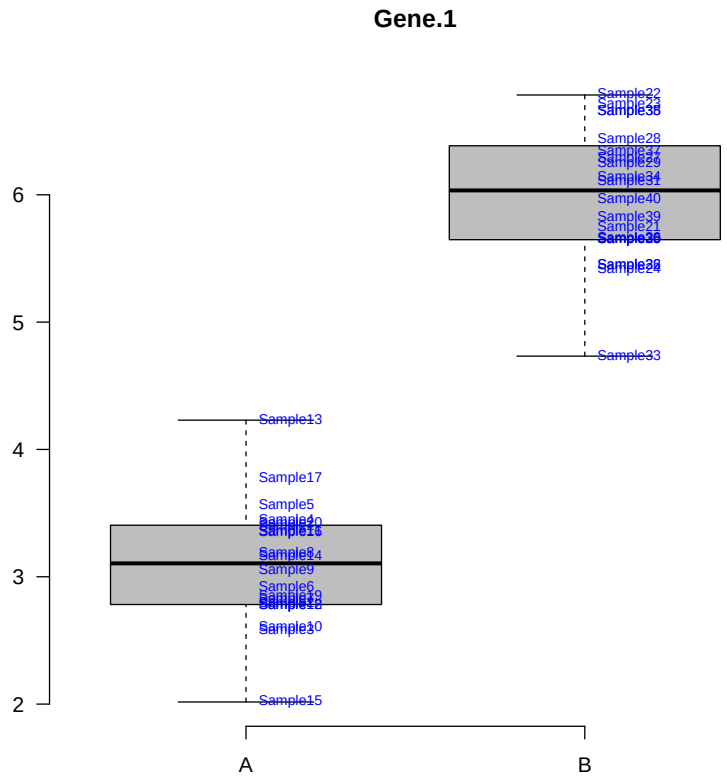

Figure 6: The expression levels of the first gene of the simulated data set.

Figure 7 shows that all classifiers except bagging (MCR of 0.015) have an average MCR of 0 for all gene set sizes.

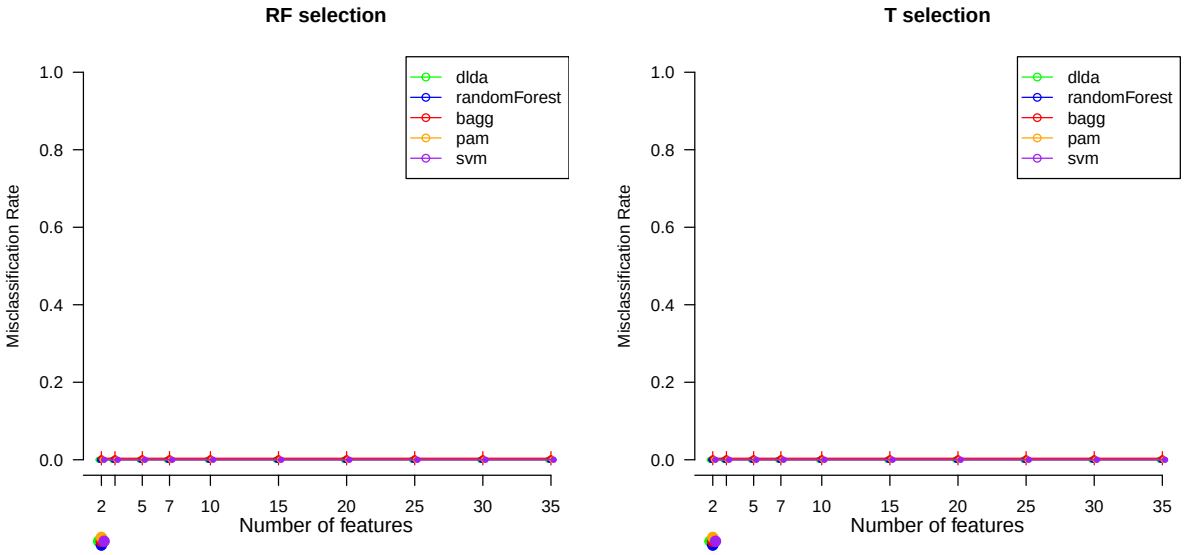

Figure 7: The mean misclassification rate (mcr) and its standard error for each classification technique and number of features, calculated across the runs of the nested loop cross-validation.

Figure 8 indeed shows that all samples were classified correctly in 100with Random Forest with Variable Importance selection with a gene set of 5 genes.

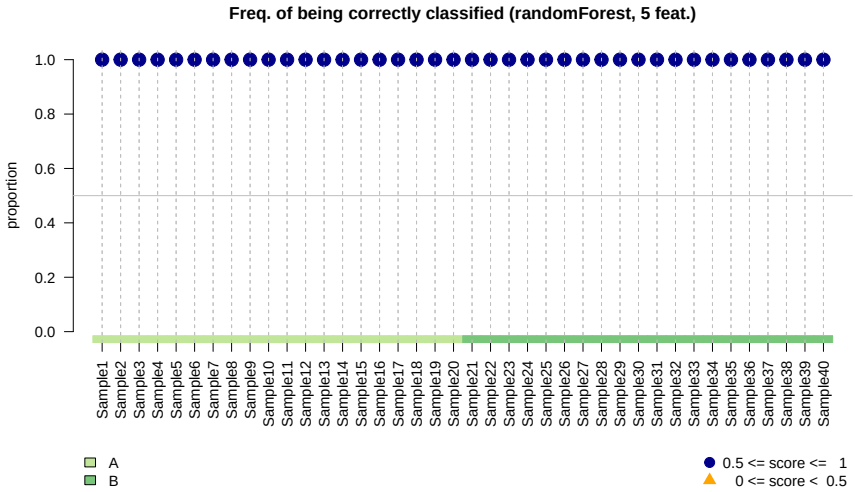

Figure 8: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

Finally, the top 12 most frequently selected genes by RF variable importance (Table 2) shows that the genes with a signal were always selected. The same thing applies for t-test selection.

	percentage.Var1	percentage.Freq
1	Gene.1	100.00
2	Gene.10	100.00
3	Gene.2	100.00
4	Gene.3	100.00
5	Gene.4	100.00
6	Gene.5	100.00
7	Gene.6	100.00
8	Gene.7	100.00
9	Gene.8	100.00
10	Gene.9	100.00
11	Gene.661	95.00
12	Gene.789	55.00

Table 2: Top 20 features across all runs of the nested loop cross-validation.

3.5 Data containing a weak signal.

In a similar simulation, we have introduced 5 genes that are only moderately differentially expressed between the groups A and B (see Gene.1 in Figure 9 as an example).

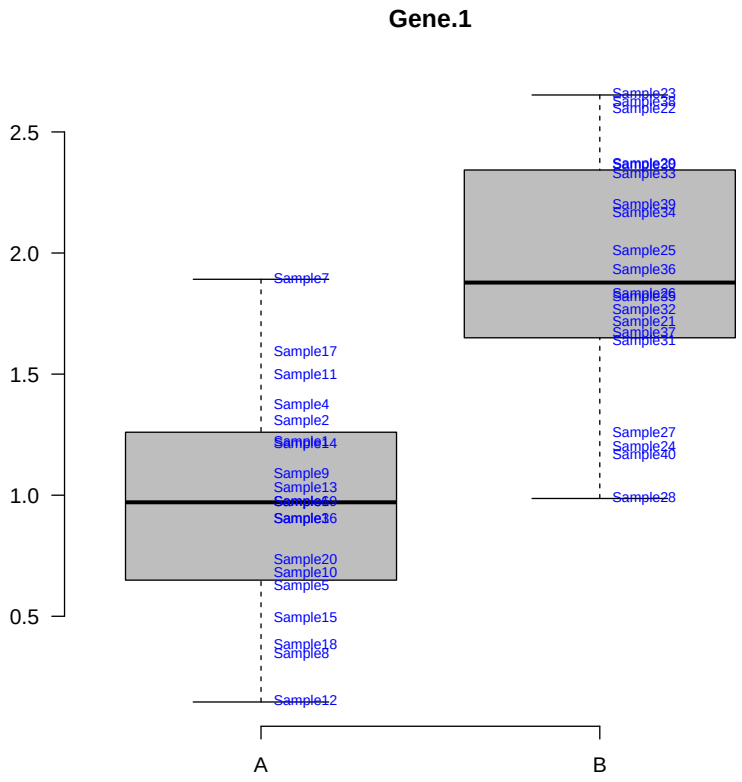

Figure 9: The expression levels of the first gene of the simulated data set.

Figure 10 shows that more variation across the different classifiers, but also an increased variation across runs of the same classifier for the same gene set size. In general, average MCR around 0.3 are obtained. Not all gene set sizes have similar MCRs, but there is a minimum for gene sets containing around 5 to 10 genes. This perfectly fits the bias-variance trade-off expected due to overfitting.

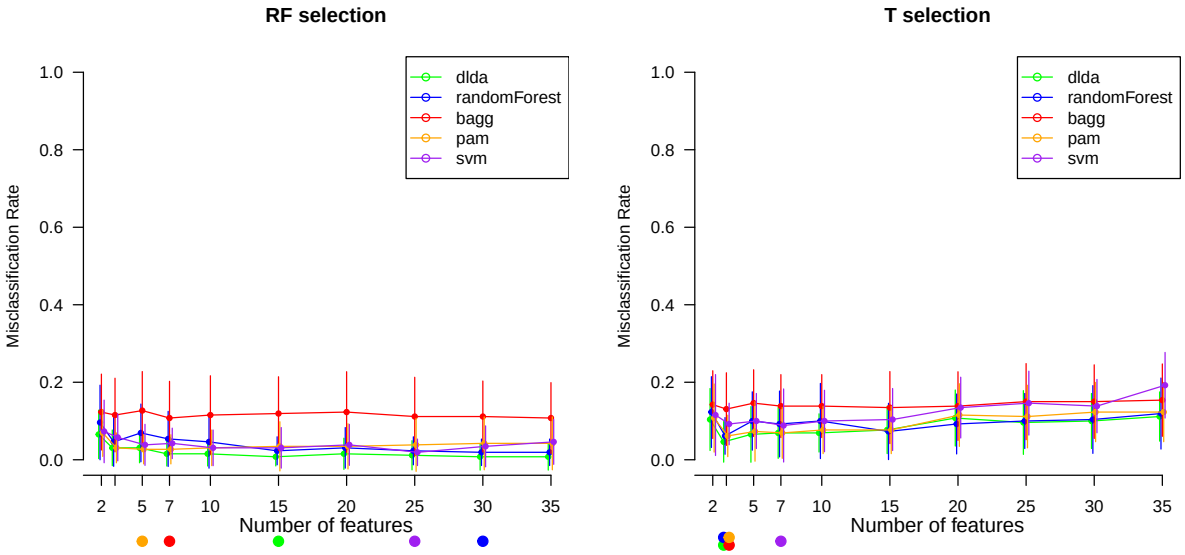

Figure 10: The mean misclassification rate (mcr) and its standard error for each classification technique and number of features, calculated across the runs of the nested loop cross-validation.

Support Vector machines using t-test feature selction performs the best when using 7 genes. Figure 11 shows that all samples except two were classified more than half of the times. Note that if one would use this summarized confidence level to assess classification accuracy, one would even have a MCR of 0.1 (2/20).

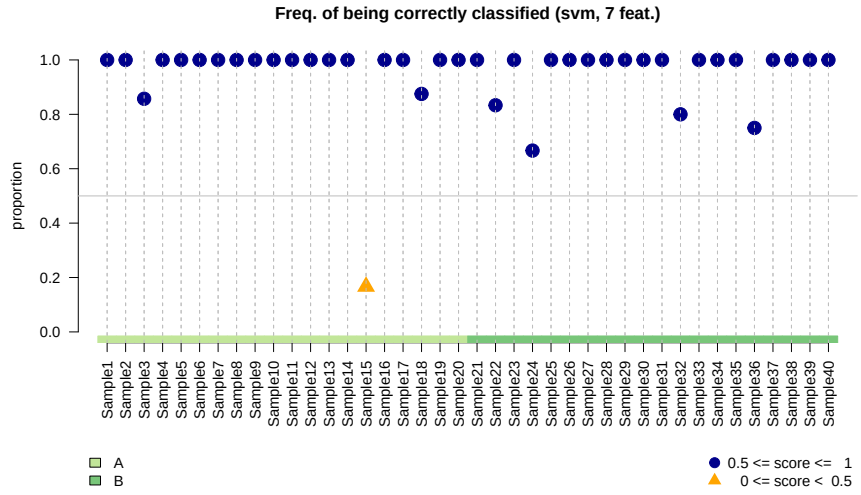

Figure 11: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

The top 7 most frequently selected genes by t-test selection (Table 3) shows that the 5 simulated genes are in the top 6. RF variable importance (Table 4) is is selecting less accurately as expected because the signal was simulated gene-by-gene.

	percentage.Var1	percentage.Freq
1	Gene.2	100.00
2	Gene.3	100.00
3	Gene.4	100.00
4	Gene.5	85.00
5	Gene.1	50.00
6	Gene.282	40.00
7	Gene.133	20.00

Table 3: Top 20 features selected with t-test across all runs of the nested loop cross-validation.

	percentage.Var1	percentage.Freq
1	Gene.2	100.00
2	Gene.3	100.00
3	Gene.4	100.00
4	Gene.282	90.00
5	Gene.5	90.00
6	Gene.1	60.00
7	Gene.975	45.00

Table 4: Top 20 features selected with RF variable importance across all runs of the nested loop cross-validation.

3.6 Data containing a strong and heterogeneous signal.

Besides introducing 5 genes that are strongly differentially expressed between the groups A and B, we also simulate 5 samples of group A to behave like group B (see Gene.1 in Figure 17 as an example).

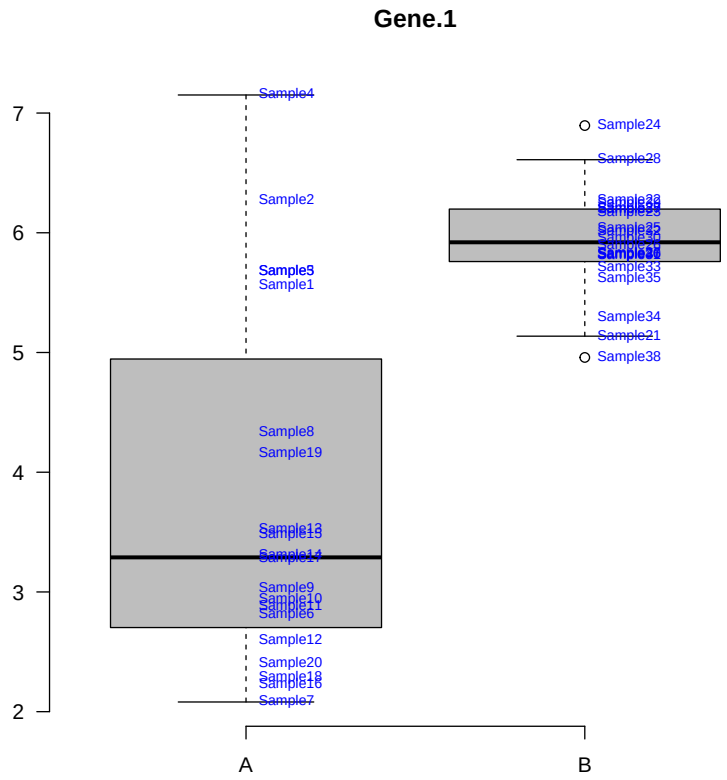

Figure 12: The expression levels of the first gene of the simulated data set.

Figure 13 shows that, despite the strong signal, the overall MCR is not zero but around 0.15-0.2. There is quite some variability within and between the different classifiers. Although not very clearly, MCRs decrease with increasing size towards sizes of 3-5 and afterwards increase, as expected due to overfitting. [? SVM seems to be the classifier that is the most sensitive to overfitting.] Irrespective of the gene set size, PAM always results in the lowest MCRs.

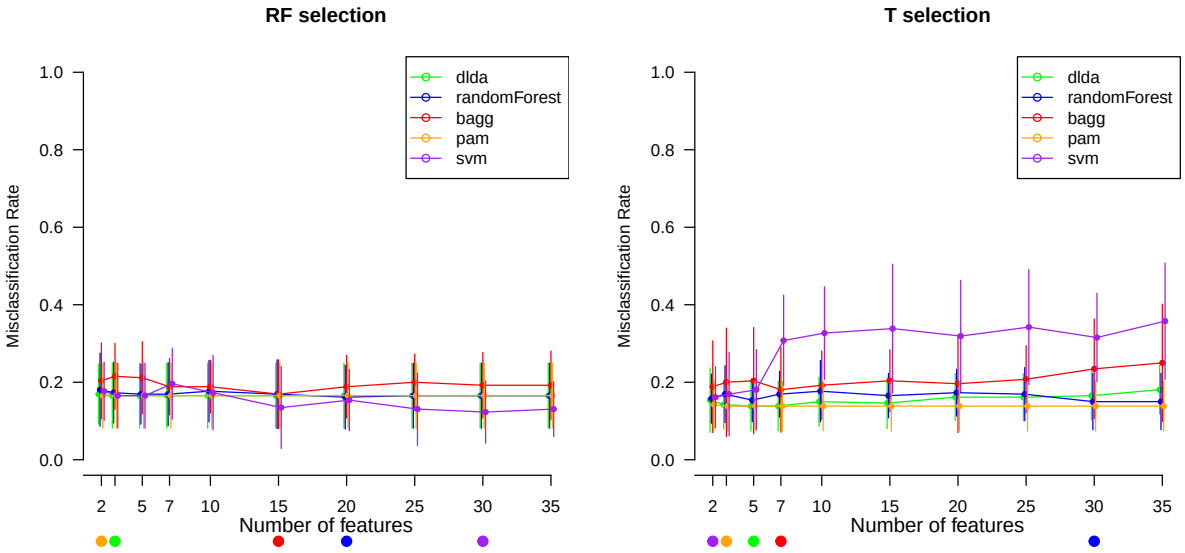

Figure 13: The mean misclassification rate (mcr) and its standard error for each classification technique and number of features, calculated across the runs of the nested loop cross-validation.

Let's look at 3 different 'feature selection-classifier' combinations. Figure 14 shows that PAM with 7 genes selected with t-test classifies all samples correctly. The samples truely belonging to their group are classified 100 % correctly, while the first five samples are 100% classified to the other group - which is also correct as we have simulated these samples to behave in this way.

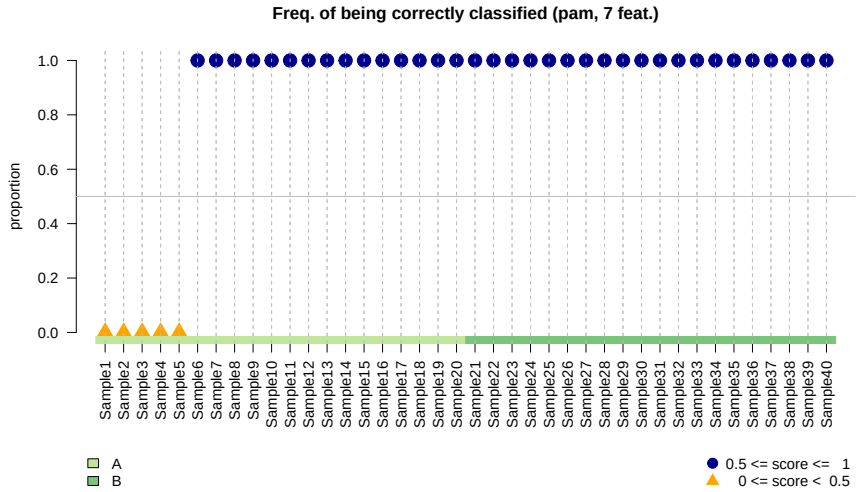

Figure 14: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

Second, using the same 7 genes, Random Forest fails to put all samples in their true group (Figure 15). If one selects the 7 genes using RF variable importance, Random Forest improves things, but is still not as good as the MCR obtained when using PAM (Figure 16. Therefore, PAM seems to be more robust against misspecified samples.

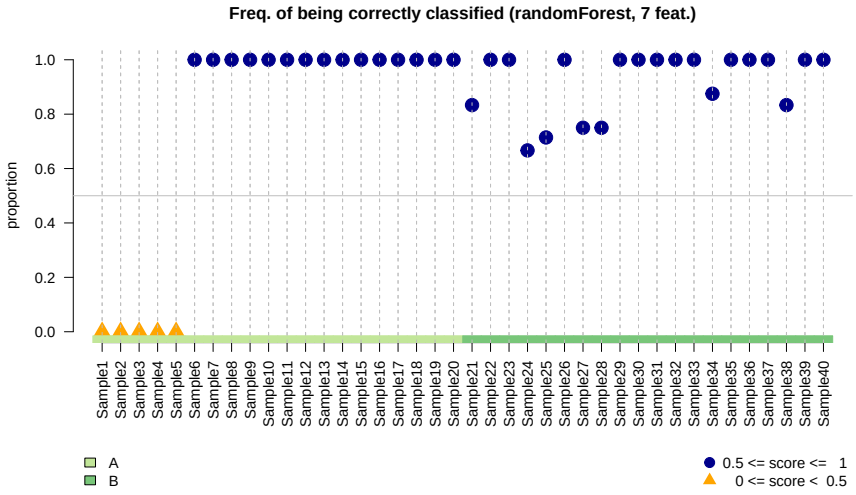

Figure 15: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

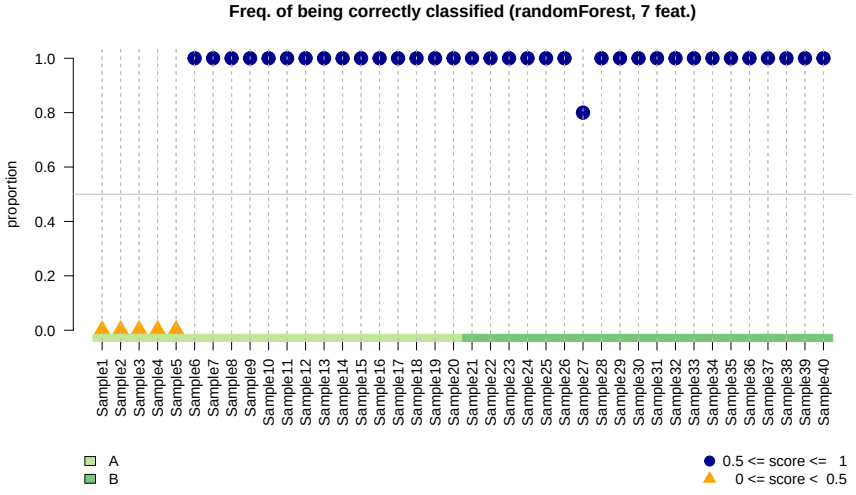

Figure 16: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

The top 7 most frequently selected genes by t-test selection (Table 5) shows that the 5 simulated genes are in the top 6. RF variable importance (Table 6) is is selecting less accurately

as expected because the signal was simulated gene-by-gene.

	percentage.Var1	percentage.Freq
1	Gene.1	100.00
2	Gene.2	100.00
3	Gene.3	100.00
4	Gene.4	100.00
5	Gene.5	100.00
6	Gene.616	35.00
7	Gene.864	20.00

Table 5: Top 20 features selected with t-test across all runs of the nested loop cross-validation.

	percentage.Var1	percentage.Freq
1	Gene.1	100.00
2	Gene.2	100.00
3	Gene.3	100.00
4	Gene.4	100.00
5	Gene.5	100.00
6	Gene.864	45.00
7	Gene.514	15.00

Table 6: Top 20 features selected with RF variable importance across all runs of the nested loop cross-validation.

3.7 Data containing a weak and heterogeneous signal.

For completeness, let's do a similar simulation as previously (5 outlying samples) but with 5 genes that are only moderately differentially expressed between the groups A and B (see Gene.1 in Figure 17 as an example).

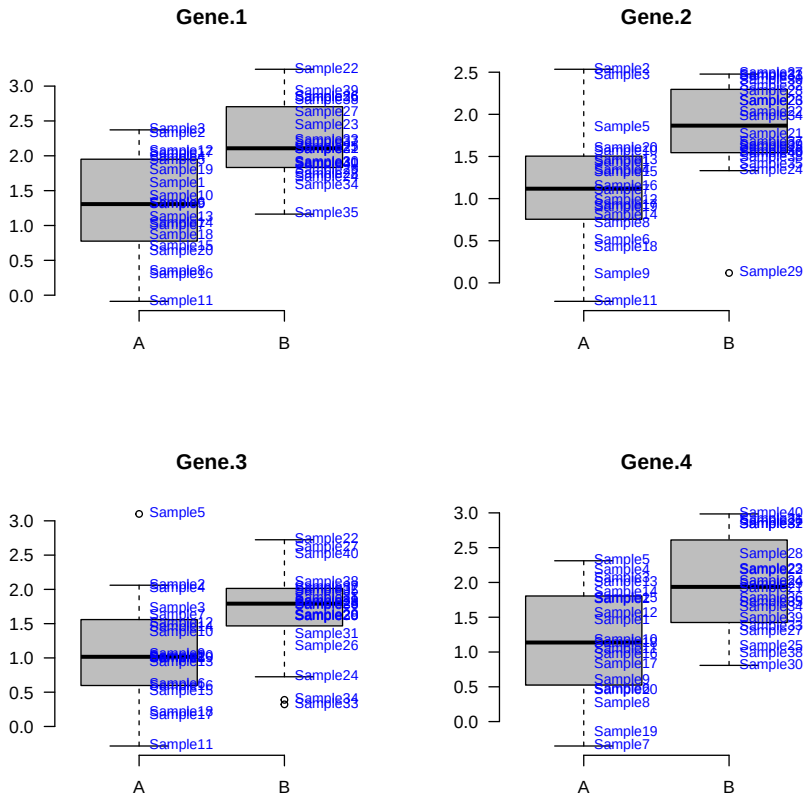

Figure 17: The expression levels of the first gene of the simulated data set.

Figure 18 shows that overall MCR lie around 0.35-0.4. There is quite some variability within and between the different classifiers. Now, PAM does not always result in the overall lowest MCRs.

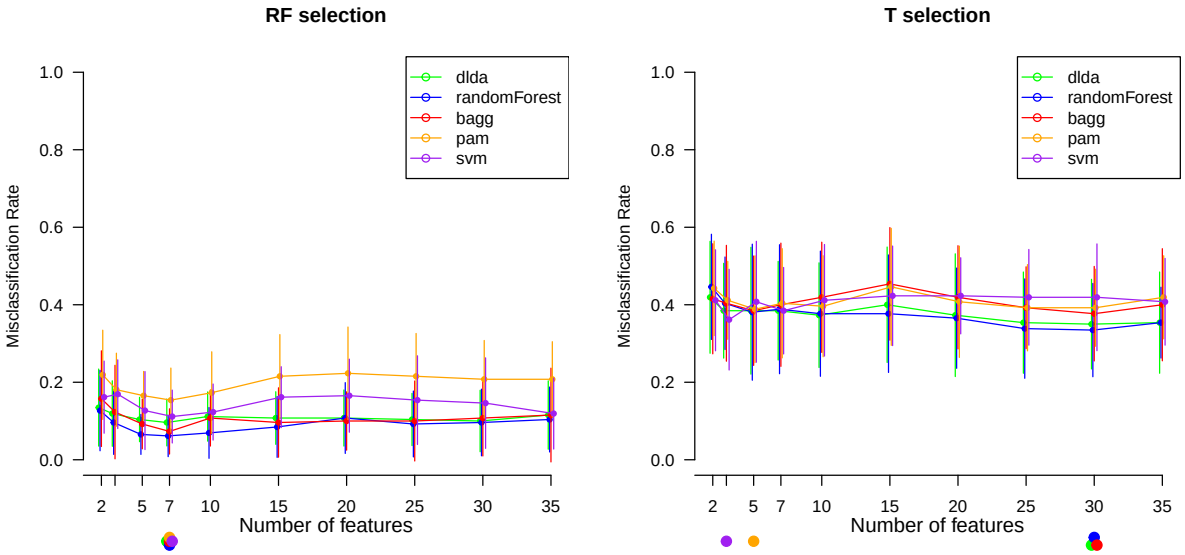

Figure 18: The mean misclassification rate (mcr) and its standard error for each classification technique and number of features, calculated across the runs of the nested loop cross-validation.

Let's look at 3 different 'feature selection-classifier' combinations. Figure 19 shows that PAM with 2 genes (the most optimal gene set size) selected with t-test classifies most samples correctly, although still 5 samples are less than 50% of the time classified correctly. PAM with 10 genes results however in only 2 missclassified samples (Figure 20).

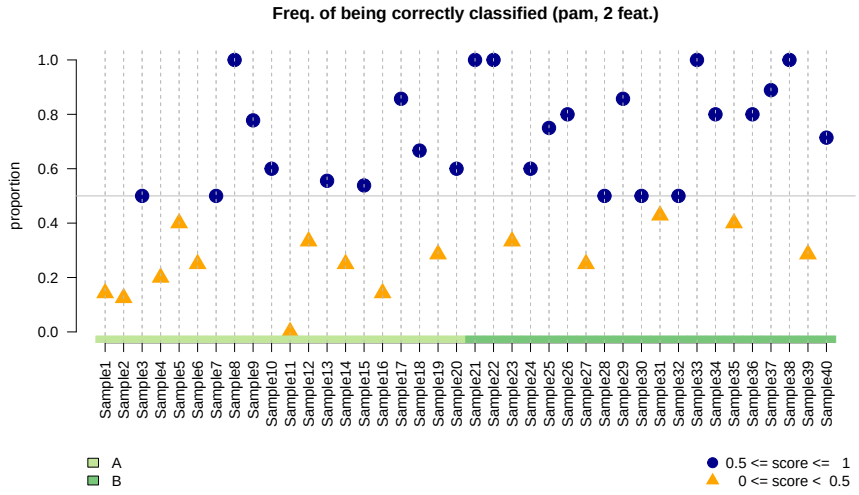

Figure 19: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

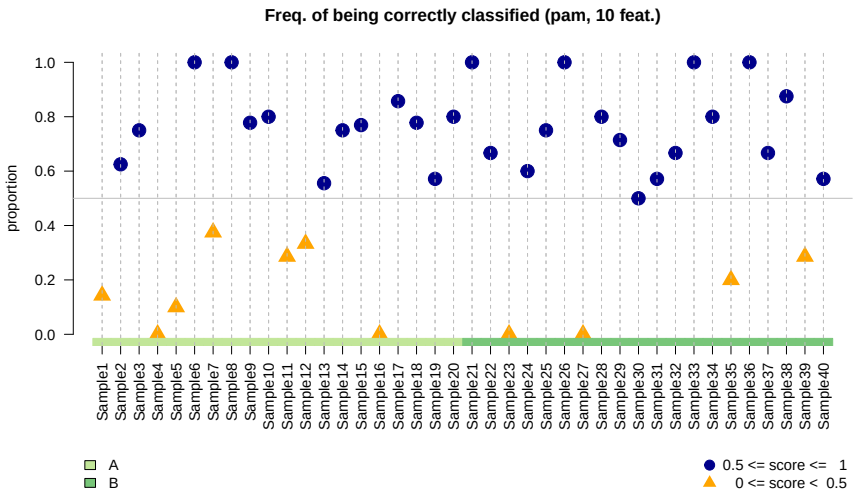

Figure 20: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

Second, using 15 genes (the optimal gene set size for RF), Random Forest does slightly worse (Figure 21). If one selects 5 genes using RF variable importance, Random Forest improves things. Although it has 6 samples that are most of the times missclassified, it succeeds in predicting some samples more often correctly, i.e. more percentages around 100% or around zero % (for the 5 outlying samples) (Figure 22). Therefore, RF seems to be preferred.

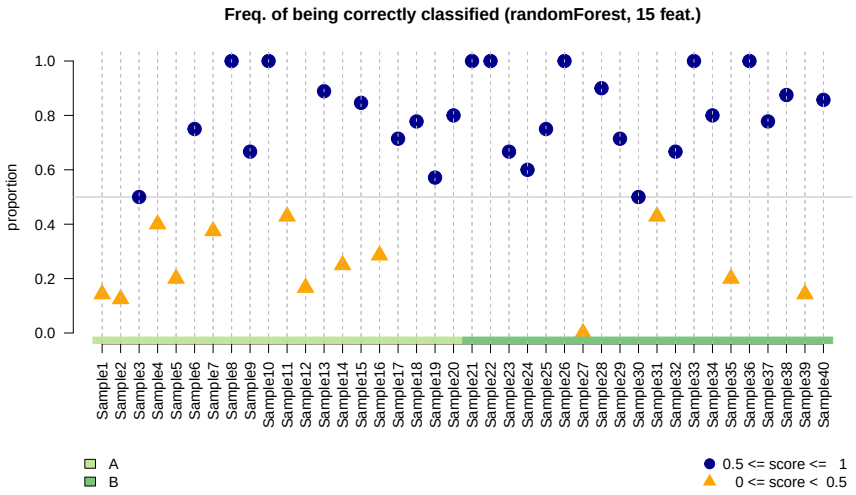

Figure 21: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

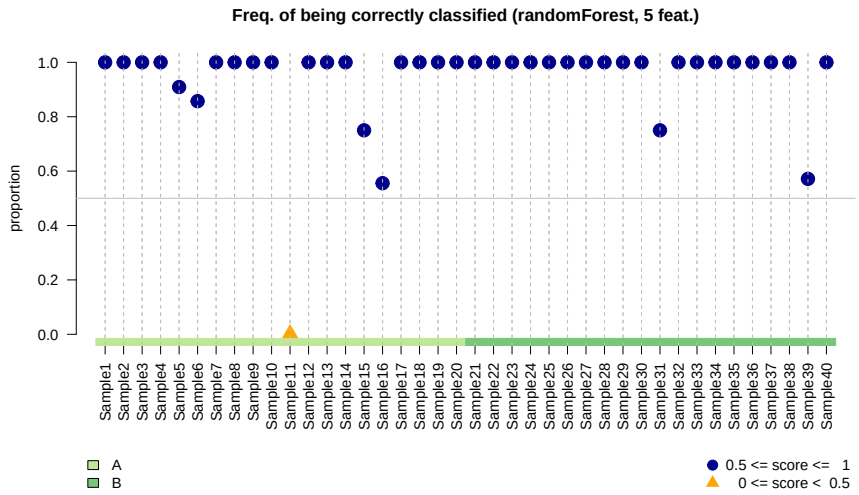

Figure 22: The proportion for each sample of being correctly classified across all runs of the nested loop cross-validation.

The most frequently selected genes by t-test selection contain only 2 of the 5 genes with an effect (Table 7), while RF variable importance contain 4 out of 5 (Table 8). This might be explained because we used here a parametric t-test which is more sensitive for outliers.

	percentage.Var1	percentage.Freq
1	Gene.4	85.00
2	Gene.3	65.00
3	Gene.288	60.00
4	Gene.2	55.00
5	Gene.524	50.00
6	Gene.837	45.00
7	Gene.703	40.00
8	Gene.823	40.00
9	Gene.5	35.00
10	Gene.391	30.00

Table 7: Top 20 features selected with t-test across all runs of the nested loop cross-validation.

	percentage.Var1	percentage.Freq
1	Gene.288	100.00
2	Gene.2	95.00
3	Gene.3	95.00
4	Gene.4	85.00
5	Gene.837	75.00
6	Gene.518	60.00
7	Gene.228	55.00
8	Gene.629	30.00
9	Gene.823	30.00
10	Gene.524	25.00

Table 8: Top 20 features selected with RF variable importance across all runs of the nested loop cross-validation.

A Software used

- R version 4.4.1 (2024-06-14), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Time zone: `Etc/UTC`
- TZcode source: `system (glibc)`
- Running under: `Ubuntu 24.04 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3`
- LAPACK: `/usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.26.so` ; LAPACK version3.12.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, stats4, utils
- Other packages: AnnotationDbi 1.67.0, Biobase 2.65.0, BiocGenerics 0.51.0, IRanges 2.39.2, MLInterfaces 1.85.0, Rcpp 1.0.13, S4Vectors 0.43.2, XML 3.99-0.17, a4Core 1.53.0, annotate 1.83.0, cluster 2.1.6, nlcv 0.3.5, xtable 1.8-4
- Loaded via a namespace (and not attached): Biostrings 2.73.1, DBI 1.2.3, DelayedArray 0.31.11, GenomeInfoDb 1.41.1, GenomeInfoDbData 1.2.12, GenomicRanges 1.57.1, KEGGREST 1.45.1, MASS 7.3-61, Matrix 1.7-0, MatrixGenerics 1.17.0, R6 2.5.1, RColorBrewer 1.1-3, ROCR 1.0-11, RSQLite 2.3.7, S4Arrays 1.5.7, SparseArray 1.5.31, SummarizedExperiment 1.35.1, UCSC.utils 1.1.0, XVector 0.45.0, abind 1.4-5, bit 4.0.5, bit64 4.0.5, blob 1.2.4, buildtools 1.0.0, cachem 1.1.0, class 7.3-22, cli 3.6.3, codetools 0.2-20, compiler 4.4.1, crayon 1.5.3, data.table 1.15.4, digest 0.6.36, e1071 1.7-14, fastmap 1.2.0, foreach 1.5.2, future 1.34.0, future.apply 1.11.2, glmnet 4.1-8, globals 0.16.3, grid 4.4.1, httr 1.4.7, ipred 0.9-15,

iterators 1.0.14, jsonlite 1.8.8, kernlab 0.9-33, knitr 1.48, lattice 0.22-6, lava 1.8.0, limma 3.61.9, listenv 0.9.1, magrittr 2.0.3, maketools 1.3.0, matrixStats 1.3.0, memoise 2.0.1, multtest 2.61.0, nnet 7.3-19, pamr 1.57, parallel 4.4.1, parallelly 1.38.0, png 0.1-8, prodlim 2024.06.25, proxy 0.4-27, randomForest 4.7-1.1, rlang 1.1.4, rpart 4.1.23, shape 1.4.6.1, splines 4.4.1, statmod 1.5.0, survival 3.7-0, sys 3.4.2, tools 4.4.1, vctrs 0.6.5, xfun 0.46, zlibbioc 1.51.1

References

[1] Christophe Ambroise and Geoffrey J McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci U S A*, 99(10):6562–6566, May 2002.

[2] Ramón Díaz-Uriarte and Sara Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

[3] Stefan Michiels, Serge Koscielny, and Catherine Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet*, 365(9458):488–492, 2005.

[4] David F Ransohoff. Rules of evidence for cancer molecular-marker discovery and validation. *Nat Rev Cancer*, 4(4):309–314, Apr 2004.

[5] Markus Ruschhaupt, Wolfgang Huber, Annemarie Poustka, and Ulrich Mansmann. A compendium to ensure computational reproducibility in high-dimensional classification tasks. *Stat Appl Genet Mol Biol*, 3:Article37, 2004.