Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
17 April 2020
Manuscript Category

CS4260 Machine Learning in Bioinformatics Final report

Single-cell transcriptome analysis of chronic myeloid leukemia at diagnosis time

Marthe Veldhuis [1,*], Akash Singh [1], Matus Mikus [1] and Hielke Walinga [1,*]

[1] Technical University Delft

Abstract

The progression of cancer within the bodies of its patients and the myriad responses of the disease to therapy had mostly been a black-box before the advent of genome sequencing. With single-cell transcriptomics, it is now possible to record genetic expression of individual cells, thus offering a peek inside the black-box. Our main contribution in this paper is to identify questions relevant to Chronic Myeloid Leukemia and adapt modern machine learning techniques on a gene expression data set to answer them. Main challenges we faced during this process were prevalence of batch effect (inherent bias in data as a result of sampling protocols), high-dimensional data (23384 genes), and relatively low number of samples (2287 cells). Through the application of batch-effect removal, dimension-reduction, and strategically selected modelling techniques, we attempt to show that there exist distinct biological signals within the gene expression data of cancer cells. We zoom in on these signals through our methodology to answer certain key questions.

Contact: m.s.veldhuis@student.tudelft.nl; a.singh-25@student.tudelft.nl; m.mikus@student.tudelft.nl; h.walinga@student.tudelft.nl

1 Introduction

1.1 Problem

In cancer research, analysis of tumors was historically done at the level of the tissue, which overlooks the natural heterogeneity of cells and their individual response to molecular therapy. These multi-cell approaches lack the precision to identify patterns of gene expression in individual cells, and as a result fail to identify critical mutations. However, recent advances in single-cell transcriptomics allow researchers to develop tools to analyze single-cell data and closely identify individual cell mutations and gene expression. Methods developed on single-cell data offer great promise for molecular-targeted therapy, as it allows for exploration of single-cell therapy resistance, as well as the ability to monitor the progression of a disease on the cellular level. Are there inherent factors in a patient's genetic composition before starting therapy that determine the outcome of the treatment?

1.2 Related work

The work of (Giustacchini *et al.*, 2017) serves as the base of our analysis. The authors' goal was to develop methods for single-cell transcriptome analysis of patients with chronic myeloid leukemia (CML). Furthermore, the authors used their analysis to identify patterns of disrupted cell development, as well as to gain insight into therapy-resistant cell identification. The authors implement methods for answering several research questions, such as detection of aberrant genes occurring in CML, prediction of a cell's response to therapy, and characterization of this response. During the different sections of our analysis, we relate to other literature that defines how to process RNA Sequence data using different (Machine Learning) methods.

1.3 Contribution

The focus of our analysis lies on finding information about patients at diagnosis time, and their future disease progression. For this purpose we have formulated three research questions:

1. What differentiates the gene expression of patients with a poor response to TKI therapy in comparison to good responders?
2. How can we predict good and poor responders to TKI therapy at diagnosis time?
3. What is the probability of a cell progressing to blast crisis from diagnosis?

Each question was tackled by the following authors:

1. Marthe Veldhuis
2. Hielke Walinga
3. Akash Singh, Matus Mikus

2 Methods

2.1 Data Processing

2.1.1 Data sets

The original data sets provided by (Giustacchini *et al.*, 2017) consist of the single-cell gene expressions from RNA sequencing of human bone marrow of chronic myeloid leukemia (CML) patients or normal donors. These samples are accompanied by meta data for each cell, such as patient origin, processing date, in what stage of treatment the samples were taken, and the patient's response to the treatment. The expression values are quantified by Read Per Kilobase Million (RPKM), causing the reads to be normalized for gene length. The total expression data concerns 23384 genes and 2287 cells.

2.1.2 Picking genes of interest

The first challenge we immediately were aware of was the high number of genes. As mentioned in Luecken and Theis (2019), a large number of these genes contain noisy signals which increase computational costs of an analysis without providing any useful information. Therefore, we decided to limit our data for most of the analysis to 5000 genes using a modified version of the approach suggested in (Luecken and Theis, 2019) (note that in 2.2.1 the full dataset is used as required for the used software). We selected all the genes with mean expression (across all the reported cells) greater than the 50^{th} percentile of the mean expression of all genes. This ensures that we eliminate genes with extremely low mean expression values (most likely due to measurement noise) from making it to the selected list of genes. We sort these genes in decreasing order of the ratio of Standard Deviation and Mean of expression (across all reported cells) and select the top 5000 genes.

2.1.3 Batch-effect removal

As indicated in the accompanying metadata, the gene expression data was gathered in 6 different batches. This raises the suspicion of inherent bias being present in the data owing to different experimental procedures/conditions (Also see 2). In order to eliminate this bias, we remove batch-effect on our reduced (5000 genes) data in two steps:

1. Log transforming the gene expression data
2. Centralising data from different batches using ComBat function, thus removing the batch effect

The log transformation was performed because the ComBat function demands log-transformed data. More importantly, using a log_2 transformation, the data becomes more normally distributed, which is favorable for many types of statistical analysis (Lee, 2016; Luecken and Theis, 2019). It also helps to model proportional change, which is typically biologically more relevant as shown by (Danielsson *et al.*, 2015). RNA seq data is inherently quite skewed, because some genes are abundantly transcribed, causing large variation between samples on average as well. This phenomena can be summarized as heteroskedasticity, or mean-variance dependency. By log-transforming, this dependence is reduced (Datta and Nettleton, 2014).

Notice in fig 1.a how the yellow, green, and cyan clusters are quite distinct from each other which indicates prevalence of batch-effect in the data. Results post removal of batch-effect are displayed in fig 1.b where these clusters are diffused throughout the volume of data.

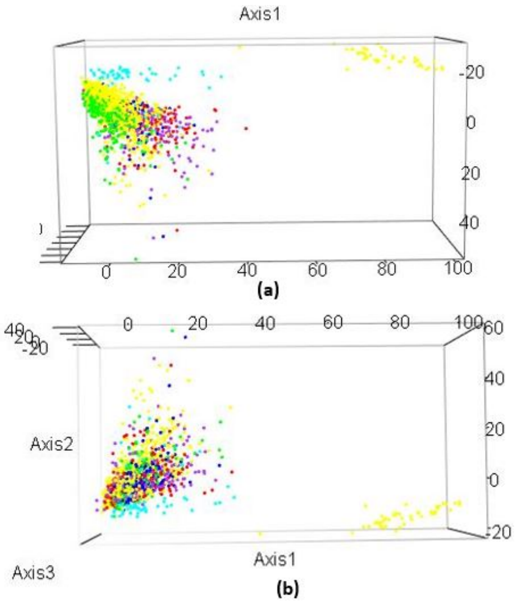


Fig. 1: **(a.) Before batch-effect removal; (b.) Post batch-effect removal**

2.2 Methodology

2.2.1 Differentiating gene expression of poor responder patients in comparison to good responders

For determining what differentiates good and poor responders at diagnosis time, a core step is Differential Expression (DE) analysis (Luecken and Theis, 2019; Conesa *et al.*, 2016).

Multiple packages exists for performing this type of analysis such as DESeq2, edgeR, and limma. However, these all require raw count data as input (Love *et al.*, 2014; McCarthy *et al.*, 2012; Ritchie *et al.*, 2015). Unfortunately, we had no access to the original count data, only the RPKM values. This type of quantification should preferably not be used for a DE type of analysis. However, RPKM data, once log_2 transformed, can still be used to achieve reasonably good results (Datta and Nettleton, 2014) by using limma (MacDonald, 2015). limma has further been shown to perform well under many circumstances, and is also the fastest out of the three to run (Conesa *et al.*, 2016; Luecken and Theis, 2019). Moreover, analysis by DESeq and edgeR is a more fixed process, whereas limma allows steps in the pipeline to be executed separately. This is beneficial to our case, since we do not have the correct input for the initial step, which would transform the raw count data using voom to log2-counts-per-million (logCPM) values (Ritchie *et al.*, 2015). Instead, we skip this step, and use our logRPKM values as input to the next part of the process. In this way, our data is closer to the desired format.

For the rest of the DE analysis, the limma guide by (Ritchie *et al.*, 2015) was used as a pipeline example.

Since limma requires the entire relevant data to estimate the mean-variance relationships, the original data set was used instead of the filtered, and batch-corrected version described in sections 2.1.2 and 2.1.3. Any zero-expressed genes were removed since they do not contain any information. Only cells from patients that were sequenced at diagnosis time and have recorded response to the TKI treatment as good ($n = 11$) or poor ($n = 5$) are included in the data set. In this way, no treatment has influenced the expressions of genes in the cells and the focus lies on if patients have any predisposition at to the TKI treatment. This cuts down the number of genes from 23384 to 23110, and the number of cells from 2287 to 792.

Next, the differences in expression profiles groups were explored using MDS plots (see Figure 2). Here distances between samples are shown as "leading fold change", which corresponds to "the root-mean-square average of the log-fold-changes for the genes best distinguishing each pair of samples."(Smyth, 2020). The top 500 samples with the largest deviations between samples are shown, as a type of unsupervised clustering. Though the good and poor responders do not seem to separate that well (Figure 2a), it is useful to see that there is a clear batch effect (Figure 2b), which should be adjusted for when designing the model in the next steps (Ritchie *et al.*, 2015).

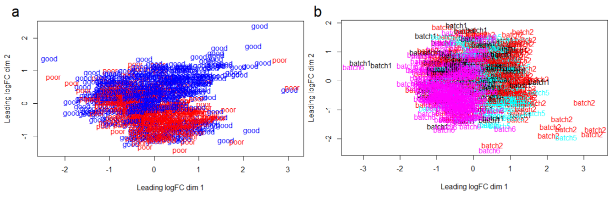


Fig. 2: (**a**) Clustering based on responder status. (**b**) Clustering based on batch

The method limma uses for DE analysis is to fit a linear model to every gene in the data (see Figure 3). So for each gene g, we have a vector of gene expression values y_g. Then, using a design matrix X, these values can be related to some coefficients of interest β_g. This design matrix thus provides a representation of the targets to be modelled. In our case, this is used to model the responder status, as well as to model the covariates that we need to take into account, like the batch effect we noticed in Figure 2b, and patient origin. We also use a contrast matrix, which specifies which comparison of the targets is of interest for our differential expression analysis. We use this to compare poor- to good responders. The final step is the application of Empirical Bayes to estimate the variance for each gene. It combines the data for each gene, with the global variability across all genes, estimated by pooling the ensemble of all genes. In this way, the variance can be modelled more accurately. The results of this analysis can be found in section 3.1.1.

To relate the outcomes of the DE analysis to more tangible results, Gene Set Testing was used to obtain over-represented functions from the obtained fit (Ritchie *et al.*, 2015; Luecken and Theis, 2019; Conesa *et al.*, 2016). Most frequently, the Gene Ontology (GO) project is used to annotate genes to more general functions such as defined by (Mi *et al.*, 2019) as:

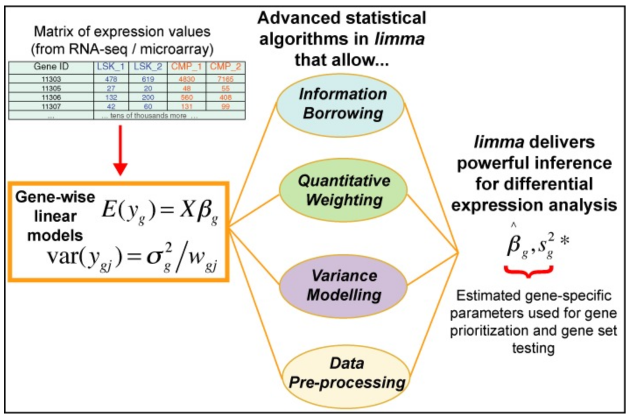

Matrix of expression values (from RNA-seq / microarray)
Gene ID | LSK_1 | LSK_2 | CMP_1 | CMP_2
11303 | 478 | 619 | 4830 | 7165
11305 | 27 | 20 | 48 | 55
11306 | 132 | 200 | 560 | 408
11307 | 42 | 60 | 131 | 99
... | ... tens of thousands more ...
Gene-wise linear models
$E(y_g) = X\beta_g$
$\mathrm{var}(y_{gj}) = \sigma_g^2 / w_{gj}$
Advanced statistical algorithms in *limma* that allow...
Information Borrowing
Quantitative Weighting
Variance Modelling
Data Pre-processing
limma delivers powerful inference for differential expression analysis
$\hat{\beta}_g, s_g^2$ *
Estimated gene-specific parameters used for gene prioritization and gene set testing

Fig. 3: limma linear model

1. Cellular Component = which parts of a cell the gene product is active in.
2. Molecular Function = which molecular activities the gene product relates to.
3. Biological Process = which pathways and processes the gene product contributed to.

In this way, more biologically relevant information can be extracted from the DE analysis results, combining information from multiple genes. limma contains the goana method which operates on the fit from the DE analysis to extract enriched GO terms automatically (Ritchie *et al.*, 2015). In order to obtain the correct ontologies, the gene symbols in the data needed to be converted to Entrez IDs first. Please refer to section 3.1.2 to see the GO results.

Lastly, to get more specific results as well, Gene Set Enrichment Analysis (GSEA) was performed. This analysis is implemented in limma by the romer function which, like goana, uses the obtained fit and Entrez IDs as input. The genes in this analysis are mapped to gene sets from the Molecular Signatures Database collections (MSigDB). This database includes a wider range of sets as compared to the Gene Ontologies, helping to represent more specific information about biological processes and diseases related to our DE analysis (Liberzon *et al.*, 2015). Refer to section 3.1.3 to inspect our GSEA results.

2.2.2 Predicting good and poor responders to TKI therapy at diagnosis time

Instead of figuring out what constitutes the difference between good and bad responders, you could look directly at how to *predict* the response. It is by this way not necessarily to find out what exactly makes up the difference, as this can be a very complex relation that can not easily be shown. So the goal here is, can we predict the response to TKI therapy by looking at the data. To predict this we will be using various supervised learning techniques.

The data set we have is slightly different from a conventional supervised learning design, because the data we have is a nested model. We measured expression in cells, but all these cells are coming from different patients where the response is known at the patient level. This causes a confounding factor on the data, but it also allows us to lessen the need to have a 100% accurate prediction for all cells. Since we only have to predict correctly for each patient.

When doing supervised learning it is essential that you make an appropriate holdout dataset as the error on the training set will very likely be higher than you will see on unseen data. What's more, if the error on the holdout dataset is much higher than on the training set you are likely overfitting. Since we are dealing with nested data it is even more important to make sure that you are making an appropriate test set. If you just select a random set of cells from the data and train on this, your model will learn all differences for the different patients provided while it is much more important it must also perform well enough on data from patients it has never seen before. Therefore to have a good test set it is best if you holdout all cells from different patients.

Since the initial dataset is too big to train the models on we had to compromise a bit and only used the 5000 genes as described in Section 2.1.2. This preselection might not be the best possible gene selection for all models.

We picked the following models to train our data on:

1. **LDA** From the R package *MASS* (Venables and Ripley, 2002).
2. **Random Forest** Breiman's random forest algorithm from the R package *randomForest* (Liaw and Wiener, 2002).
3. **QDA** From the R package MASS (Venables and Ripley, 2002).

4. **SVM** The SVM makes use of the default kernel which is a radial kernel. The method comes from the R package *e1071* (Meyer *et al.*, 2019).
5. **LASSO** The optimal λ is found using cross-validation. This method comes from the package *glmnet* (Friedman *et al.*, 2010).
6. **Ridge** The same applies here as for LASSO.

Reducing the amount of genes For some models that did not have regularization we wanted to reduce the dataset even more and picked the top 500 genes as ranked by the random forest method with the Gini index. This might not be the ideal ranking for all models, as this is mostly the most important genes for the random forest method and it might not perform as well for models that use different methods on the data.

2.2.3 Predicting progression to blast-crisis using the genetic landscape during diagnosis

Giustacchini *et al.* (2017) discuss that the presence of BCR-ABL fusion gene is the only reliable marker for identifying CML-SCs. However, as displayed in table 1, patients with a similar BCR-ABL composition during diagnosis can progress in two different directions (remission or a blast-crisis). Clearly, simply looking at the fraction of BCR-ABL+ cells in a patient's sample is not informative about the future trajectory the patient might follow after TKI therapy. However, being able to predict a patient's state (with a reasonable, if not absolute, degree of confidence) post TKI might help in fine-tuning the therapy to the patient's benefit. Therefore, we decided to apply predictive techniques on the gene expression data to obtain a much smaller number of genes which successfully explain the difference between the two patient sets. As mentioned in the data processing section, we were working with the top 5,000 most variable genes. We started with generating cluster visualisations in reduced dimensions to display the overlap among the following two classes of cells:

1. Belonging to patients who reported remission post TKI
2. Belonging to patients who reported blast-crisis post TKI

Patient ID	**BCR-ABL+ cells during diagnosis**	**BCR-ABL+ cells post TKI**	**Patient Status post TKI**
CML1266	65%	88%	Blast Crisis
CML15	65%	45%	Remission
CML655	86%	27%	Remission
CML940	80%	56%	Remission
OX1931	86%	90%	Blast Crisis

Table 1. Progression of patients with similar BCR-ABL profiles

Because we were approaching this question using multiple independent methodologies, we decided to use UMAP, a non-linear dimension reduction technique, to visualise results of non-linear techniques following the recommendation of (Luecken and Theis, 2019). In order to reduce dimensions of linear results, we decided to use Principal Component Analysis, a linear method. There are 8 patients for whom we have gene expression records from two points in time - during diagnosis and a few months after the initiation of TKI. This data comes from 409 cells which gives us training data of < 409 samples (after setting aside samples for the test set) and 5000 parameters. The unique nature of our data motivated us to use the following methodologies:

1. Random Forest - Belkin *et al.* (2019) discuss experimental evidence of Random Forests performing extremely well when the number of parameters is much higher than total training samples.
2. Support Vector Machine - SVMs are also known to perform well in problems where number of parameters is much higher than total training samples.
3. Regularized General Linear Models (GLM) using the glmnet package.

3 Results and Discussion

3.1 Differences in gene expression of poor responder patients in comparison to good responders

3.1.1 Differential Expression results

Using the Differential Expression analysis specified in section 2.2.1, we obtained a result with a total of 81 down-regulated and 168 up-regulated genes for poor responders with respect to good responders. These results are summarized in Figure 4a. For this figure a p-value cut-off of $10e-7$, and a fold-change cut-off of 0.5 are used. These were chosen as such to show a limited amount of genes, as well as helping account for the small sample size. It is in line with the top DE genes as outputted by limma itself. There are a handful of genes that have both high fold change, as well as low p-values, corresponding to being most significant. When plotting the original log_2 expression values of these top 10 genes in a box plot, the large number of zero-expression caused all quartiles to be completely at 0 (see Appendix II). Since this result is not informative, we chose to create a violin plot instead (Figure 4b). Here, the probability distribution is drawn at each expression value. This shows that these genes do indeed show quite different distributions for poor responders. For example, the CA1 gene is almost entirely zero-expressed for poor responders, while it's expression is quite even for good responders between -0.5 and 1.0. This is further supported by Figure 4c, where the mean expression is plotted. Some further research for the CA1 gene suggests it is a potential oncogene (Zheng *et al.*, 2015). Since it is not feasible to research each individual gene, the following sections may provide more comprehensive results.

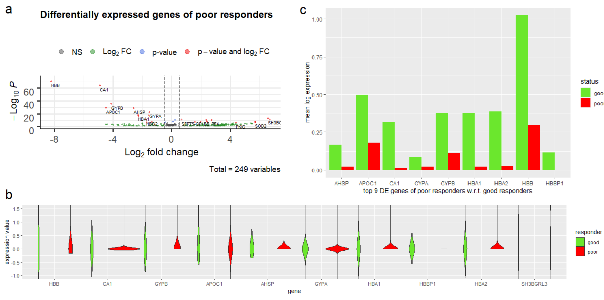


Fig. 4: (**a**) Differentially expressed genes poor responders w.r.t. good responders. (**b**) Comparison expression levels of the top 10 DE genes by responder status. (**c**) Comparison mean expression values of the top 9 DE genes by responder status (SH3BRL3 is left out due to its average expression values being too high compared to the others, skewing the scale)

3.1.2 Gene Ontology results

The results of Gene Ontology analysis can be found in Figure 5. Here, the top GO sets containing the most significant differentially expressed genes for down- (a) and up-regulated (b) genes are shown. Note that they are sorted by p-value, listing the most significant ones at the top and corresponding to a lighter color. The most frequent types of ontologies correspond to heightened oxygen transportation, increased metabolic processes, and decreases in cellular degradation (Mi *et al.*, 2019).

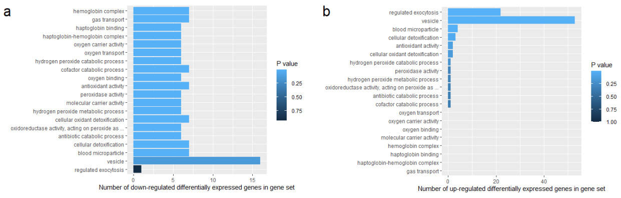


Fig. 5: Gene Ontology sets containing the most significant differentially expressed genes, color-coded by p-value (darker meaning less significant). (**a**) For down-regulated genes. (**b**) For up-regulated genes.

3.1.3 Gene Set Enrichment Analysis results

The results from Gene Set Enrichment Analysis (GSEA) using romer can be found in Figure 6. This plot shows the most significant gene sets associated with the most differentially expressed genes without regard for direction of up- or down regulation. When researched, frequent keywords corresponding to these sets include inflammation, inhibited cell death, etc. The most interesting gene sets seem to be the following, with corresponding main take-away from (MSigDB, 2020):

- GOBERT_OLIGODENDROCYTE_DIFFERENTIATION_DN: promoting dedifferentiation of myelinating cells (R.P. *et al.*, 2009).
- GRAESSMANN_APOPTOSIS_BY_DOXORUBICIN_UP: therapy resistance in breast cancer.
- NUYTTEN_EZH2_TARGETS_DN: promotes the late-stage development of cancer by silencing a specific set of genes.
- GEORGES_TARGETS_OF_MIR192_AND_MIR21: down-regulated expression of tumor suppressors.
- KRIEG_HYPOXIA_NOT_VIA_KDM3A: enhance tumor growth.

These all seem to correspond to some kind of therapy resistance, or relating to quiescent genes as well. This is further explored in section 3.3.

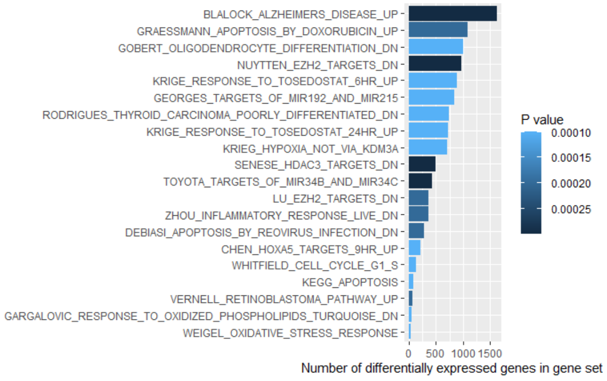


Fig. 6: Gene sets from MSigDB containing the most number of differentially expressed genes from our analysis. Note the color coding for significance (darker meaning less significant)

We would like to note that since log transformed RPKM values were used in this analysis instead of raw counts as required by limma, the results may not be optimal. What is more, some genes were not correctly mapped to their corresponding Entrez ID, causing a loss of about 8% of gene expression information. The gene symbols used might have been outdated, thus inhibiting a correct query. In a more thorough analysis, they might be mapped by hand. It would have been convenient if the data available contained the raw counts and Entrez IDs, because they are most frequently used in analysis. RPKM values can easily be derived from raw counts, where the opposite is not possible. Also, the data only has cells from 16 patients. Preferably for this type of analysis, more samples should be used.

Despite this, when analysing the results in correspondence with the findings of (Giustacchini *et al.*, 2017), as well as interpreting the results with respect to their biological functions, they seem to be quite relevant. For example, Figure 4e in (Giustacchini *et al.*, 2017) shows similar biological functions differentially expressed in good and poor responders. However, their analysis focused mostly on finding DE genes between $BCR-ABL^{+}$ and $BCR-ABL^{-}$, which is where our focus differs.

Lastly, the GSEA gene sets some DE genes belong to, might not be a clear indication of the underlying biological principles. For instance, BLALOCK_ALZHEIMERS_DISEASE_UP relates to up-regulated genes in Alzheimer's patients' brains. It ties into our results due to the proliferation of tumors and processes that apparently are also significant to Alzheimer's patients. This means that further investigation is required before conclusions can be drawn.

3.2 Predicting good and poor responders

	Train: Poor	**Train: Good**	**Test: Poor**	**Test: Good**
LDA	88.5%	82.8%	67.4%	54.1%
Random Forest	78.3%	60.3%	79.1%	45.9%
LDA (500)	90.2%	84.5%	58.1%	48.0%
QDA	100%	100%	9.2%	96.3%
SVM	97.8%	95.2%	68.6%	42.9%
LASSO	89.4%	54.5%	81.4%	54.5%
Ridge	99.5%	97.8%	55.8%	35.7%

Table 2. The percentage correctly classified of the supervised models

Table 2 summarizes the result of the supervised learning on the responder status. The exact distributions can be found in Appendix III in which it is more clear how each model performs on the patient level.

The LDA performs relatively well, but on the test set the scores are mediocre. It doesn't really learn very well on new patients. The random forest approach already performs a bit better.

We also applied LDA to a smaller subset of genes as determined by importance using the random forest model. We can see that this subset reduces the performance of the LDA drastically. The subset is not really an appropriate set for the LDA.

For the other models, QDA is clearly overfitting. The Ridge regression is also a model that seems to be overfitting. Here LASSO actually performs fairly well.

To see how well the LASSO model performs, we created an ROC plot for its performance that is shown in Figure 7. Here we can see that LASSO performs best in the region with a high true positive and a high false positive rate, but that it overall does not perform very well.

3.3 Predicting progression to blast-crisis post TKI

3.3.1 Random Forest

While working with Random Forest, we set aside 25% of total samples as test set. Training a Random Forest model in R on rest of the data gave an out-of-bag error of 0.61% (which we admit, is suspiciously low). A confusion matrix of the trained model on the test set is shown in table 3. As apparent, our model achieves perfect performance even on the test set. To further address the suspicion thus raised, we perform 10-fold cross-validation on the model. The cross-validated model achieves an out-of-bag

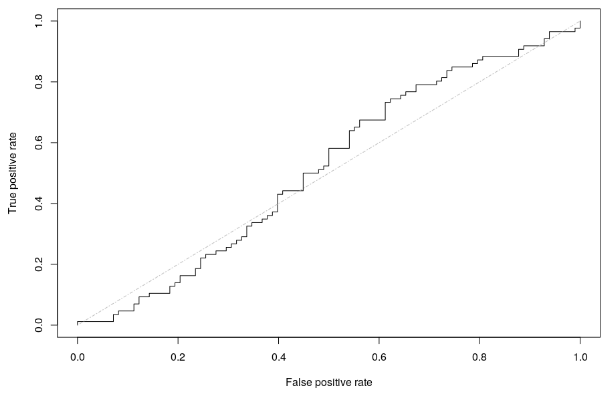


Fig. 7: **ROC curve of the LASSO model**

error of 0.55% with an error variance of scale 10^{-6}. This indicates that the model has not over-fitted and is simply a result of feeding 5000 parameters with <400 training samples to a powerful linear model (Belkin *et al.*, 2019).

	Pre-remission (actual)	**Pre-blast-crisis (actual)**
Pre-remission (predicted)	53	0
Pre-blast-crisis (predicted)	0	50

Table 3. Random Forest - confusion matrix on test set

Going back to the original objective of this exercise - predicting the future state of a recently diagnosed patient using as less genes as possible, we looked at the relative importance of all 5000 genes. Using mean decrease in Gini as our evaluation metric, we picked top 1500 genes which contributed the most to the model's Gini upon addition (on average, these 1500 genes contribute a approximately 73% increase in the model's Gini in total). However, there were no clear winners among these 1500 and none of the genes seemed to contribute significantly more than any other. As a result, we decided to proceed with all 1500. In order to visualise the incremental segregation achieved in the cells of different patient types, we used (as mentioned in section 2.2.3) UMAP which is a non-linear dimension-reduction method. Our reasoning behind this choice was that we wanted to maintain coherence between the methodologies of classification and the subsequent visualisation. Therefore, we chose to use non-linear methods for both, in this instance. Figure 8 compares the clusters obtained from the initial 5000 genes and the 1500 most important genes derived from Random Forest (Green is pre-remission, red is pre-blast-crisis). Results suggest that the biological signal was already present in those 5000 genes (as suspected) and by pruning out the unnecessary (and potentially noisy) genes, we have managed to isolate the signal to some extent. Reducing the number of genes we used here from 1500 to 500 seems to blur the distinction between the two clusters, suggesting the top 500 clusters are not enough to capture the underlying biological signal.

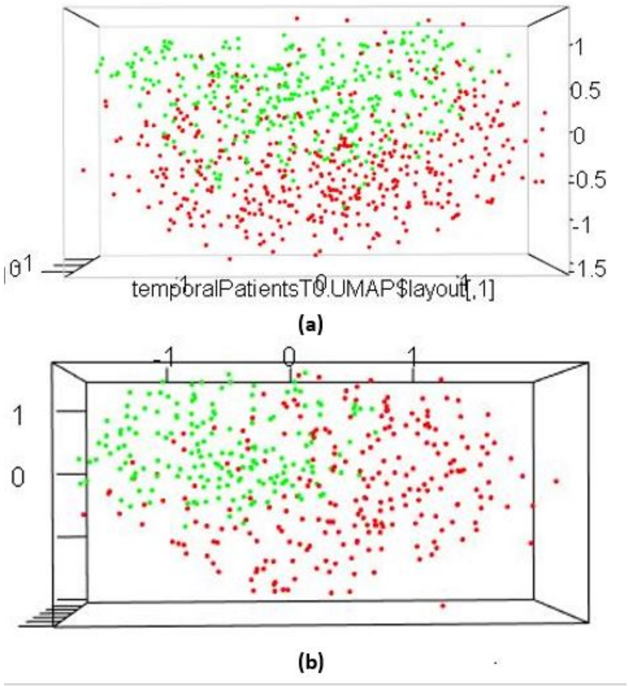


Fig. 8: **(a.) Before Random Forest - 5000 genes; (b.) Post Random Forest - top 1500 genes; Red: pre-blast-crisis, Green: pre-remission**

	Pre-remission (actual)	**Pre-blast-crisis (actual)**
Pre-remission (predicted)	43	4
Pre-blast-crisis (predicted)	10	46

Table 4. SVM - confusion matrix on test set

Looking these selected genes up in GeneCards against the key-word "blast-crisis" returned 940 genes in total, 57 of which were among the top 1000 genes selected by our model. Beer *et al.* (2015) indicate the activity of HDC, MPO, and PRG2, three genes from our top 1500 genes, to be associated with CP-CML cells. These findings provide some evidence that our findings are relevant without making any conclusive promises.

3.3.2 Support Vector Machine

Like in the case of Random Forest, we set aside 25% of samples for testing. To keep the method truly linear, we used a linear kernel for our model. The confusion matrix we obtained exhibited more "realistic" results this time (table 4).

The next obvious instinct at this point was to pick top 1500 most important genes and compare the resulting clusters against the clusters obtained by Random Forest. Since we used a linear kernel, we used the model coefficients to rank the relative importance of genes. As argued in section 2.2.3, to visualise these results we used Principal Component Analysis, to maintain consistency of methodology. Figure 9 displays the clusters thus obtained both before and after implementing SVM. The clusters we obtain are remarkably distinct as compared to the ones obtained before implementing SVM. Hence, despite the slightly lower performance (as compared to Random Forest) on the confusion matrix, this workflow allows us to distinguish between pre-remission and pre-blast-crisis patients with a greater degree of confidence.

Further, we used these 1500 genes to observe cell-clusters of the same patients from the second instance, i.e., when these patients had actually entered remission or blast-crisis. The results are displayed in figure 10

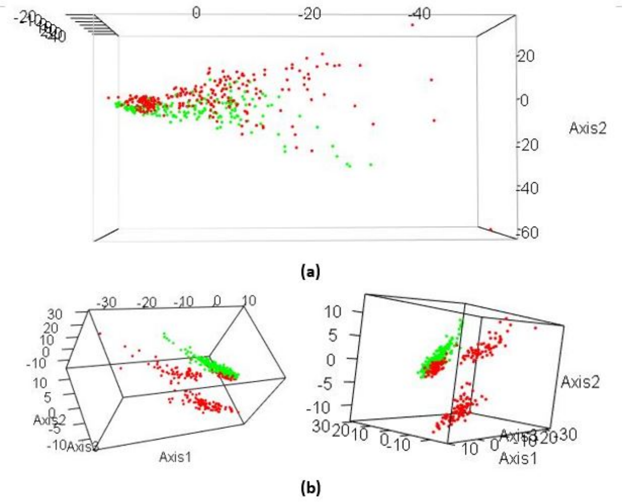


Fig. 9: **(a.) Before SVM - 5000 genes; (b.) Post SVM - top 1500 genes; Red: pre-blast-crisis, Green: pre-remission**

- the pre-blast-crisis cells (orange) cluster close to the blast-crisis cells (red) while the pre-remission cells (violet) cluster in close proximity of remission cells (green). This further indicates that the 1500 genes picked by SVM are indeed capturing a relevant biological signal which distinguishes between the trajectories towards remission and blast-crisis.

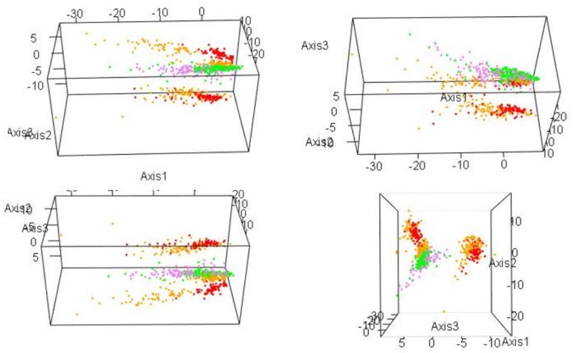


Fig. 10: **Clusters on top 1500 SVM genes: Orange - pre-blast-crisis; Red - blast-crisis; Violet - pre-remission; Green-remission**

At this point, we were curious to see if the top 1500 genes picked by the two methods have any common genes. We find 638 genes which belong to both of these sets. Clustering on these 638 genes is displayed in figure 11 - clusters in the top view were obtained by UMAP and the ones at the bottom are obtained via PCA. Notice that the PCA clusters in figure 11.b are less distinctly clustered in space as compared to the ones seen in figure 9.b.

3.3.3 Regularised General Linear Model

After analysing the performance of Random Forest and SVM on our test data, we have decided to further explore simpler models to see whether a linear model can detect a clear signal in the data. We have chosen the GLM, due to its support for various machine learning staple operations such as cross validation, as well as a variety of tuning parameters and overall efficiency. The GLM has achieved accuracy of 80% on the test data, which points to a non-linear trend in the data which cannot be detected by the GLM. In our experience, the regularization led the model into selecting only 28 genes as significant, with non-zero weights, which we thought was a significant result. These genes overlap heavily with the genes picked by the previous two methods (figure 13). Figure 12 displays clusters obtained by applying PCA on the 28 genes selected by GLM - the cells clusters somewhat sparsely but distinctly in space.

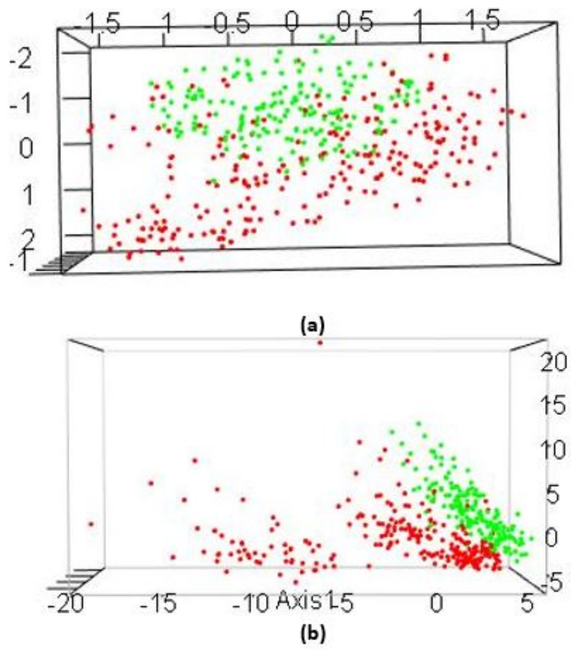


Fig. 11: **Common Most Important Genes among SVM and Random Forest (a.) UMAP clusters; (b.) PCA clusters; Red: pre-blast-crisis, Green: pre-remission**

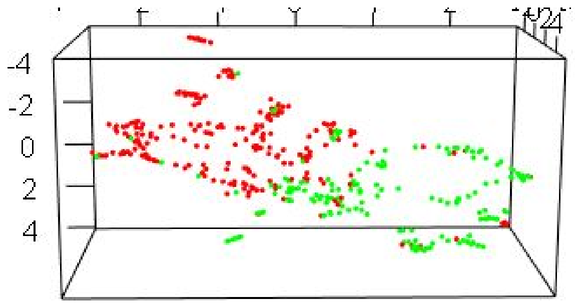


Fig. 12: **PCA clusters on 28 GLM genes; Red: pre-blast-crisis, Green: pre-remission**

4 Conclusion

In conclusion, our findings suggest that already during diagnosis, the genetic profile of a patient contains undeniable clues about the subtleties of their unique medical situation. These clues, despite being a result of highly complex genetic interactions, can be isolated and analysed thanks to recent advances in single-cell transcriptomics and machine learning. Through Differential Expression analysis, we showed that patients with

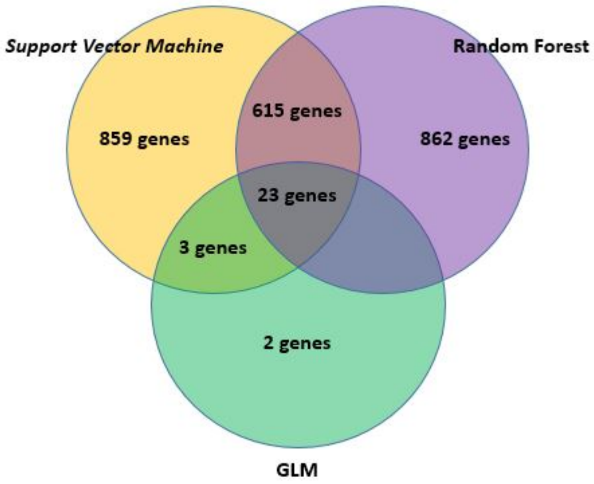


Fig. 13: **Overlap of important genes from the 3 methods**

a poor response to TKI therapy inherently have differentially expressed genes with respect to good responders. By annotating these genes with GO and GSEA annotations, key biological functions were highlighted. Examples include a heightened factor of oxygen transportation, inhibition of cell death and cancer therapy resistance. It might be an interesting approach for the future to combine these sets of genes, try to identify them in future CML patients before starting therapy, and follow their response. The main issue is that more samples need to be collected, which is a good direction for the future.

Figure 13 displays the overlap among the most important genes selected by the methods we discussed throughout section 3.3. This overlap suggests that genetic activity of a patient during diagnosis already contains signals which influence the fate of the patient (remission or blast-crisis) in the future. However, it also appears that this influence is the result of extremely complicated and interwoven genetic processes and not due to anomaly in the function of a single gene.

Despite their surprisingly perfect performance on test sets and during cross-validation, we were not able to obtain from Random Forest cell-clusters as clearly distinct as compared to the ones obtained from linear methods. However, as mentioned earlier, even the linear methods do not identify a single gene which can answer our questions. This suggests that we are indeed trying to study an extremely complicated phenomenon. It will be interesting to obtain single-cell transcriptomics from new patients, predict their trajectory post TKI, and then compare it with the actual results. Additionally, future collaborations with *in vitro* personnel may help focus on a few of these genes, thus refining our understanding of therapy resistant CML cells, an adversary defined by famous oncologist and author, Siddhartha Mukherjee as: *"(down to their innate molecular core) hyperactive, survival-endowed, scrappy, fecund, inventive copies of ourselves"*.

References

Beer, P. A. *et al.* (2015). Disruption of ikaros activity in primitive chronic-phase cml cells mimics myeloid disease progression.

Belkin, M. *et al.* (2019). Reconciling modern machine learning practice and the bias-variance trade-off.

Conesa, A. *et al.* (2016). A survey of best practices for rna-seq data analysis. *Genome Biol*, **17**.

Danielsson, F. *et al.* (2015). Assessing the consistency of public human tissue rna-seq data sets. *Briefings in Bioinformatics*, **16**, 941–949.

Datta, S. and Nettleton, D. (2014). *Statistical Analysis of Next Generation Sequencing Data*, chapter 10.2.1, pages 194–195. Springer. An optional note.

Friedman, J. *et al.* (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**(1), 1–22.

Giustacchini, A. *et al.* (2017). Single-cell transcriptomics uncovers distinct molecular signatures of stem cells in chronic myeloid leukemia. *Nat Med*, **23**, 692–702.

Lee, S. (2016). Convert natural values to log2 values.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, **2**(3), 18–22.

Liberzon, A. *et al.* (2015). The molecular signatures database (msigdb) hallmark gene set collection. *Cell Syst*, **6**, 417–425.

Love, M. *et al.* (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *bioRxiv*.

Luecken, M. D. and Theis, F. J. (2019). Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*.

MacDonald, J. W. (2015). Answer: Using rpkm data in bioconductor for gene expression analysis.

McCarthy, D. J. *et al.* (2012). Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic acids research*, **40**, 4288–4297.

Meyer, D. *et al.* (2019). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-3.

Mi, H. *et al.* (2019). Go enrichment analysis. *Nucleic Acids Res*, **47**, 419–426.

MSigDB (2020). Msigdb home.

Ritchie, M. E. *et al.* (2015). limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, **43**.

R.P., G. *et al.* (2009). Convergent functional genomics of oligodendrocyte differentiation identifies multiple autoinhibitory signaling circuits. *Mol Cell Biol*, **29**, 1538–1553.

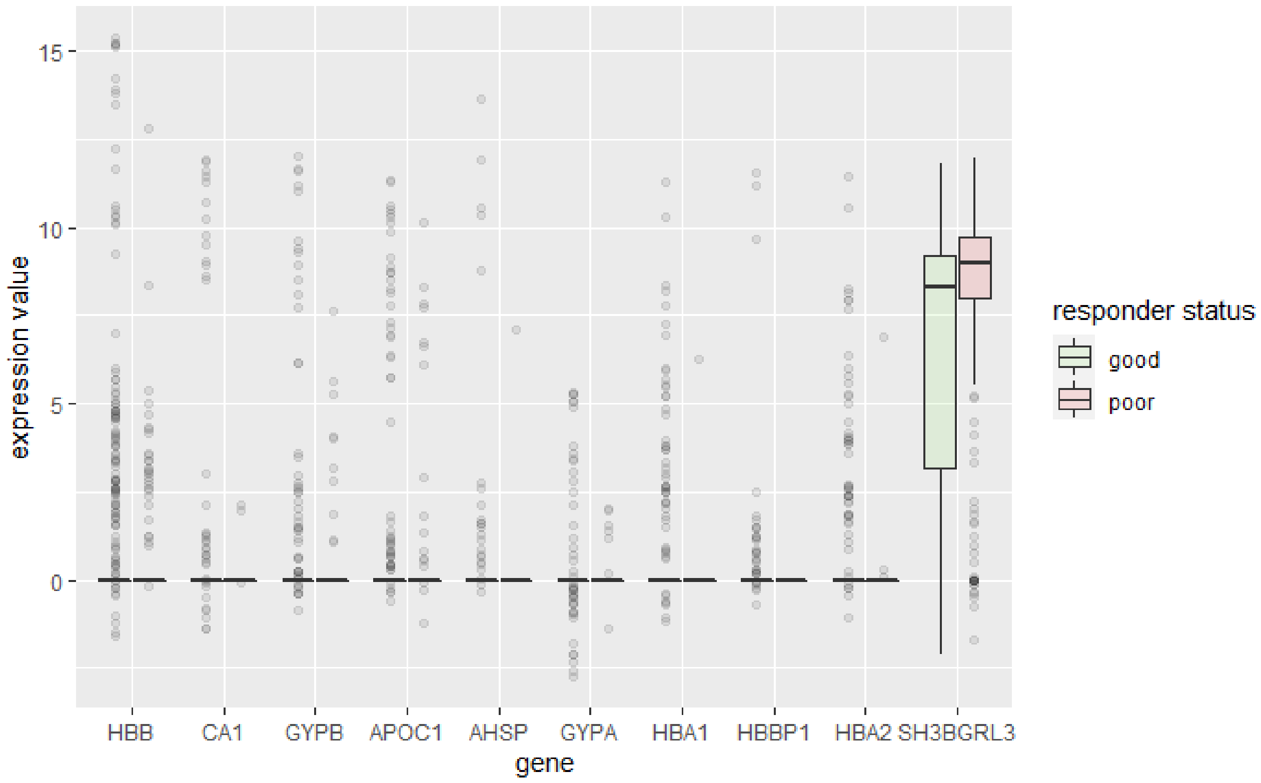
Smyth, G. (2020). plotmds.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Zheng, Y. *et al.* (2015). Ca1 contributes to microcalcification and tumourigenesis in breast cancer. *BMC Cancer*, **15**.

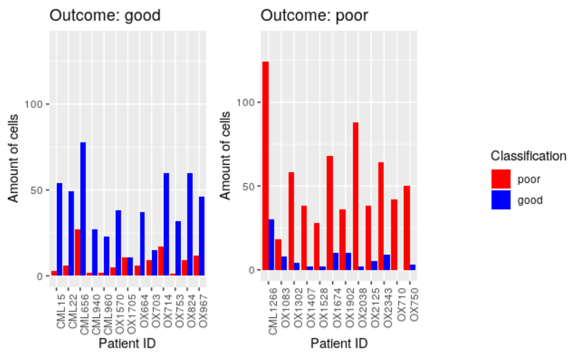
Appendix II

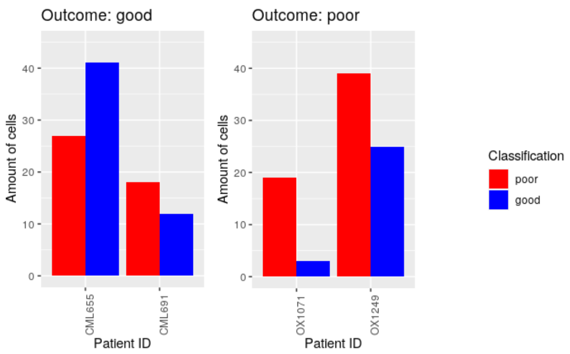
0.1 Box plot of top 10 DE expressed genes’ expression levels

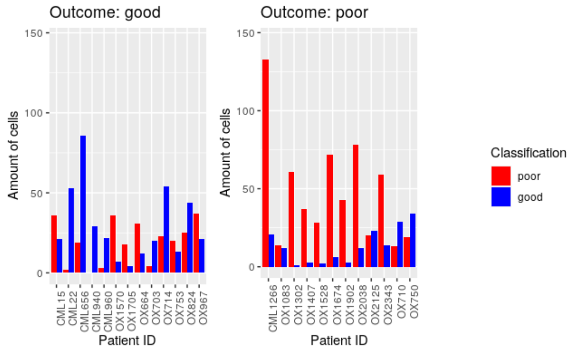


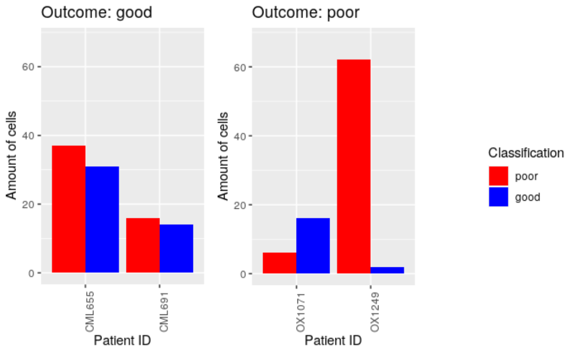
Appendix III

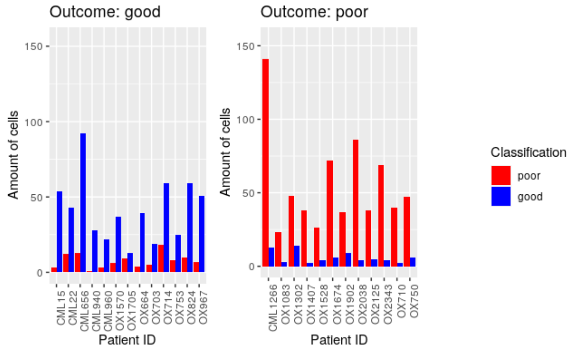
0.2 Supervised learning models on responder status

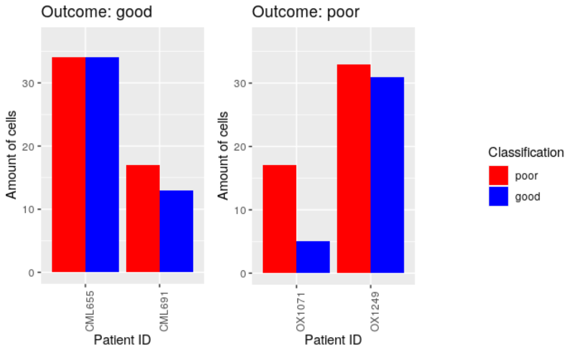

LDA: Train

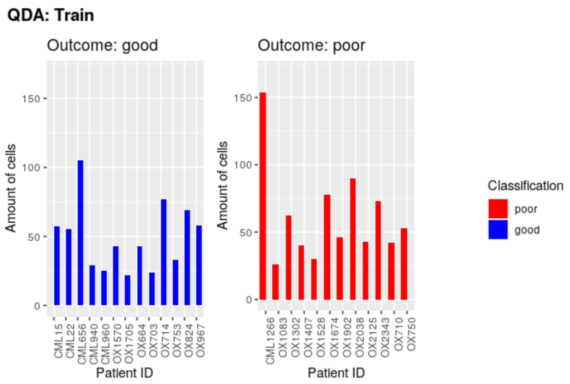

LDA: Test

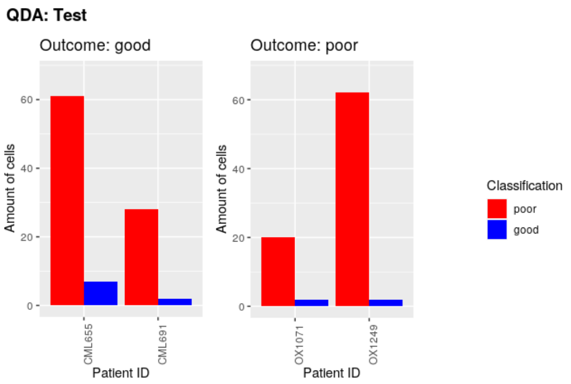

Random Forest: Train

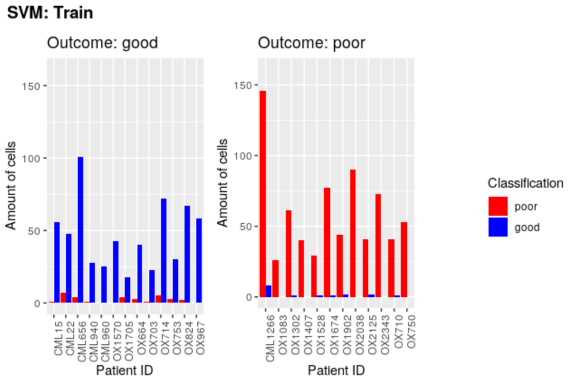

Random Forest: Test

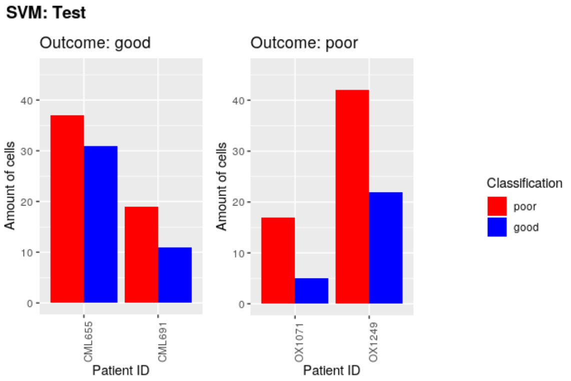

LDA (500): Train

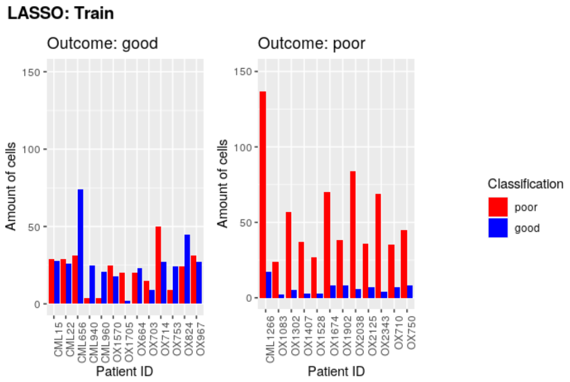

LDA (500): Test

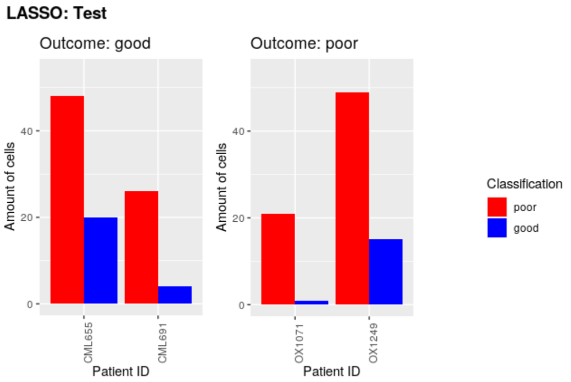

QDA: Train
Outcome: good
Outcome: poor
Amount of cells
150
100
50
0
CML15
CML22
CML656
CML940
CML960
OX1570
OX1705
OX664
OX703
OX714
OX753
OX824
OX967
CML1266
OX1083
OX1302
OX1407
OX1528
OX1674
OX1902
OX2038
OX2125
OX2343
OX710
OX750
Patient ID
Classification
🟥 poor
🟦 good

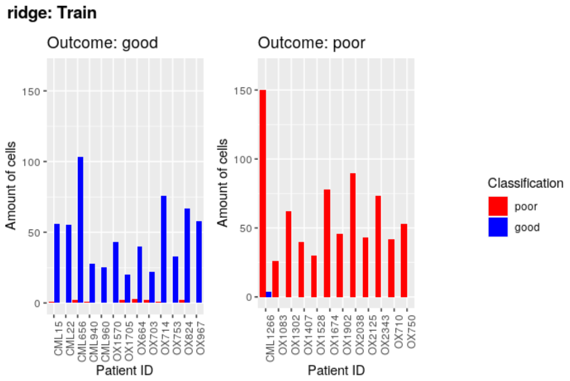

QDA: Test
Outcome: good
Outcome: poor
Amount of cells
60
40
20
0
CML655
CML691
OX1071
OX1249
Patient ID
Classification
🟥 poor
🟦 good

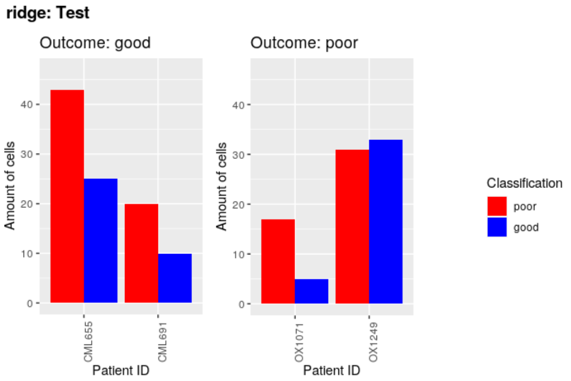

SVM: Train
Outcome: good
Outcome: poor
Amount of cells
150
100
50
0
CML15
CML22
CML656
CML940
CML960
OX1570
OX1705
OX664
OX703
OX714
OX753
OX824
OX967
CML1266
OX1083
OX1302
OX1407
OX1528
OX1674
OX1902
OX2038
OX2125
OX2343
OX710
OX750
Patient ID
Classification
🟥 poor
🟦 good


SVM: Test
Outcome: good
Outcome: poor
Amount of cells
40
30
20
10
0
CML655
CML691
OX1071
OX1249
Patient ID
Classification
🟥 poor
🟦 good


LASSO: Train
Outcome: good
Outcome: poor
Amount of cells
150
100
50
0
CML15
CML22
CML656
CML940
CML960
OX1570
OX1705
OX664
OX703
OX714
OX753
OX824
OX967
CML1266
OX1083
OX1302
OX1407
OX1528
OX1674
OX1902
OX2038
OX2125
OX2343
OX710
OX750
Patient ID
Classification
🟥 poor
🟦 good


LASSO: Test
Outcome: good
Outcome: poor
Amount of cells
40
20
0
CML655
CML691
OX1071
OX1249
Patient ID
Classification
🟥 poor
🟦 good


ridge: Train
Outcome: good
Outcome: poor
Amount of cells
150
100
50
0
CML15
CML22
CML656
CML940
CML960
OX1570
OX1705
OX664
OX703
OX714
OX753
OX824
OX967
CML1266
OX1083
OX1302
OX1407
OX1528
OX1674
OX1902
OX2038
OX2125
OX2343
OX710
OX750
Patient ID
Classification
🟥 poor
🟦 good


ridge: Test
Outcome: good
Outcome: poor
Amount of cells
40
30
20
10
0
CML655
CML691
OX1071
OX1249
Patient ID
Classification
🟥 poor
🟦 good