Machine Learning Engineer Nanodegree

Capstone Project

Title: Classification and Clustering of Paediatric Cancers

Author: Kenneth Y. Wertheim

Date: 8th November 2018

I. Definition

Project Overview

Cancers arise from changes in the genomes of their constituent cells, including single nucleotide substitutions, insertions and deletions of DNA segments, rearrangements of existing DNA segments, and copy number changes of existing DNA segments [1]. These changes are collectively known as somatic alterations.

The somatic alterations in a cancer cell are the cumulative result of multiple mutational processes, such as smoking and exposure to ultraviolet light; they leave characteristic mutational signatures [2]. For example, the nucleotide segment CC (two consecutive cytosine bases) is frequently substituted by the segment TT (two consecutive thymine bases) in skin cancers caused by ultraviolet light [2]. For a cancer genome, for selected somatic alterations, a frequency distribution can be calculated from sequencing data; it is called a mutational catalogue. Given the mutational catalogues of multiple cancers, it is possible to calculate the number of mutational processes responsible for these cancers, decipher their mutational signatures (probability that each mutational process can induce each selected alteration type), and the number of mutations caused by each process in each cancer [2].

Pan-cancer analysis is the analysis of somatic alterations across multiple cancer types, a process that identifies commonalities and differences among cancer types [3]. In a study, such an analysis was performed on 1699 paediatric cancers of six histotypes: B-lineage acute lymphoblastic leukaemia (B-ALL), T-lineage ALL (T-ALL), acute myeloid leukaemia (AML), neuroblastoma (NBL), Wilms tumour (WT), and osteosarcoma (OS) [3]. Experiments were performed to generate whole-genome, whole-exome, and transcriptome sequencing data. Some of these measurements were filtered out for multiple reasons, such as quality issues and exclusion of outliers. By applying the deciphering method [2] on the filtered raw data, 11 mutational processes were identified in 915 of the cancers in the sample, the signature of each process was deciphered, and their activities in individual cancers were calculated. Each signature consists of contributions from 96 types of somatic alterations: nucleotide substitutions in trinucleotide sequences. This processed dataset contains 915 entries. It is 12-dimensional; the 11 activities of mutational processes are features and the histotype is the label.

However, cancers evolve in an iterative process of clonal expansion, mutation, and clonal selection [4]. In other words, for the same cancer, two different mutational catalogues will be obtained at two different stages of cancer progression. In the pan-cancer analysis study [3], the sampled cancers were obtained and assigned their histotypes at diagnosis. There is a chance that a cancer classified as a WT at

diagnosis might evolve and behave more like an OS later. Alternatively, it might evolve without acquiring a different histotype. These issues were not considered in the pan-cancer analysis study. In this project, the processed dataset was used to investigate them.

Problem Statement

Three goals were set for this project.

First, if the histotype of a cancer can be predicted from its signature activities, drastic clonal evolution can also be detected. This was treated as a classification problem in the project. The plan was to train a series of classifiers on the dataset and evaluate them by a suitable metric.

Second, genetic diversity within a histotype may indicate clonal evolution. This was treated as a clustering problem. The plan was to split the dataset into six parts by histotype before looking for hierarchical structures within each.

Third, the suitability of mutational signatures as biomarkers in this context was questioned. This was treated as a dimensionality reduction problem. The plan was to look for latent features hidden in the 11 mutational signatures.

Evaluation Metrics

For the first task, the F1 score was chosen over accuracy because the dataset is imbalanced. If 100 entries are in the testing subset, roughly two OS entries will be in this subset. If a classier classifies both OS entries wrongly and the rest correctly, it will have an accuracy of 98 %, but this score does not indicate the inability of the classifier to identify OS entries. By contrast, the F1 score considers both the precision and recall of the classifier. Its precision is the proportion of positive predictions that are correct, while its recall is the proportion of positive instances that it can detect. The F1 score is the harmonic mean of its precision and recall:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}.$$

It should be noted that for each classifier, the F1 score must be calculated for each histotype. For example, the F1 score for OS depends on a classifier's precision in predicting this histotype and ability to recall OS entries from the dataset. For each classifier, the unweighted average of its six F1 scores was deemed by the author to be the best metric because it considers the six histotypes to be equally important.

For the second task, an internal validation index was chosen because the dataset does not contain any distinguishing labels within a histotype. Two options were considered. The silhouette coefficient is a measure of inter-cluster distances. It ranges from negative one to positive one; if it is close to one, it means the clusters are individually compact and far from each other; if it is close to zero, it means the clusters are overlapping or poorly separated; if it is close to minus one, it means many entries are in the wrong clusters. However, the silhouette coefficient is only an accurate metric for compact, dense, and circular clusters; for example, it does not work well for ring-like clusters. Variance is strictly speaking an objective function for hierarchical clustering with Ward's method rather than a cluster validation index. As an instance of this clustering method, given five pairs of clusters, potentially including singleton clusters, the pair whose variance is the lowest among the five will be merged. If the variances

of the final two mergers in a hierarchical clustering task are averaged, the result will be a metric for the three largest clusters. It was decided that the latter would be used because the shapes of any subclusters were unknown and hierarchical clustering was deemed suitable for the second task due to its ability to reveal hierarchical structures within a histotype.

For the third task, it was decided early on that a principal component analysis would serve the purpose. Any latent features returned by this method come with the amount of variance explained by each latent feature. The percentage of variance explained by a latent feature was adopted as the metric for this task.

II. Analysis

Data Exploration and Visualisation

As explained previously, there are 915 entries in the dataset. Each entry represents one cancer, has 11 features describing the activities of mutational signatures, and has its histotype as its label.

The label is a categorical variable, so label encoding was deemed necessary for this project. An abnormality is the imbalanced distribution of entries between the six histotypes. There are 263 entries for T-ALL, 218 for B-ALL, 197 for AML, 137 for NBL, 81 for WT, and 19 for OS. If the dataset is split randomly into training, validation, and testing subsets, the subsets may not represent the original dataset. For example, there is a chance that no OS entries will be present in the testing subset. It was decided that stratified sampling should be employed to preserve the imbalanced distribution in any validation or testing subsets. Figure 1 shows the distribution of histotypes in the dataset.

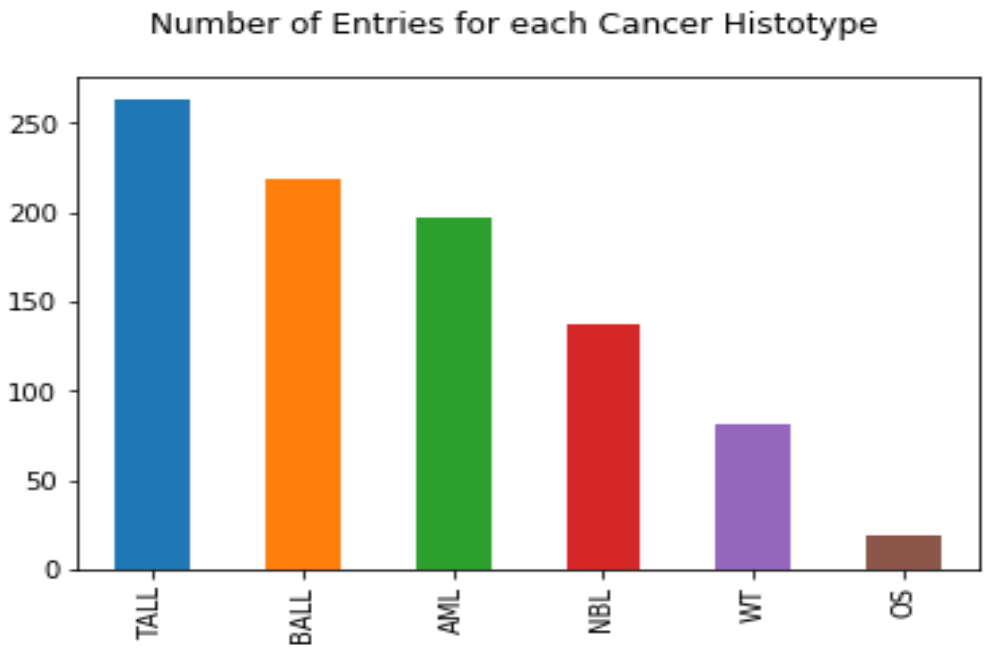

Figure 1: Number of entries for each cancer histotype in the dataset. BALL stands for B-lineage acute lymphoblastic leukaemia; TALL, T-lineage ALL; AML, acute myeloid leukaemia; NBL, neuroblastoma; WT, Wilms tumour; and OS, osteosarcoma.

The 915×11 matrix of features is very sparse in the sense that it contains many zeros. The minimum activity of every signature is zero. The maximum activity of signature 1 is 0.633; signature 2, 0.296;

signature 3, 0.812; signature 4, 0.967; signature 5, 0.777; signature 6, 0.430; signature 7, 0.521; signature 8, 0.756; signature 9, 1.43; signature 10, 0.225; signature 11, 0.426. The distributions of activities are shown in figure 2. That they have different ranges led the author to conclude that normalisation would be necessary before any machine learning methods could be applied to the dataset. Furthermore, each histogram is highly skewed at the lower end. For example, the minimum non-zero activity of signature 1 is 0.000360, orders of magnitude smaller than the maximum (0.633). It is hard to identify any patterns in the dataset as such, so a logarithmic transformation was deemed necessary by the author. Signature 9 is interesting because unlike the other signatures, its maximum activity goes beyond one. Furthermore, only two entries have non-zero activities of this signature. It is possible that errors were made in the sequencing experiments or deciphering process [3]. However, a proper investigation will require one to study the raw sequences, a task way beyond the scope of this project. Therefore, it was decided that this feature would be kept and treated like the others.

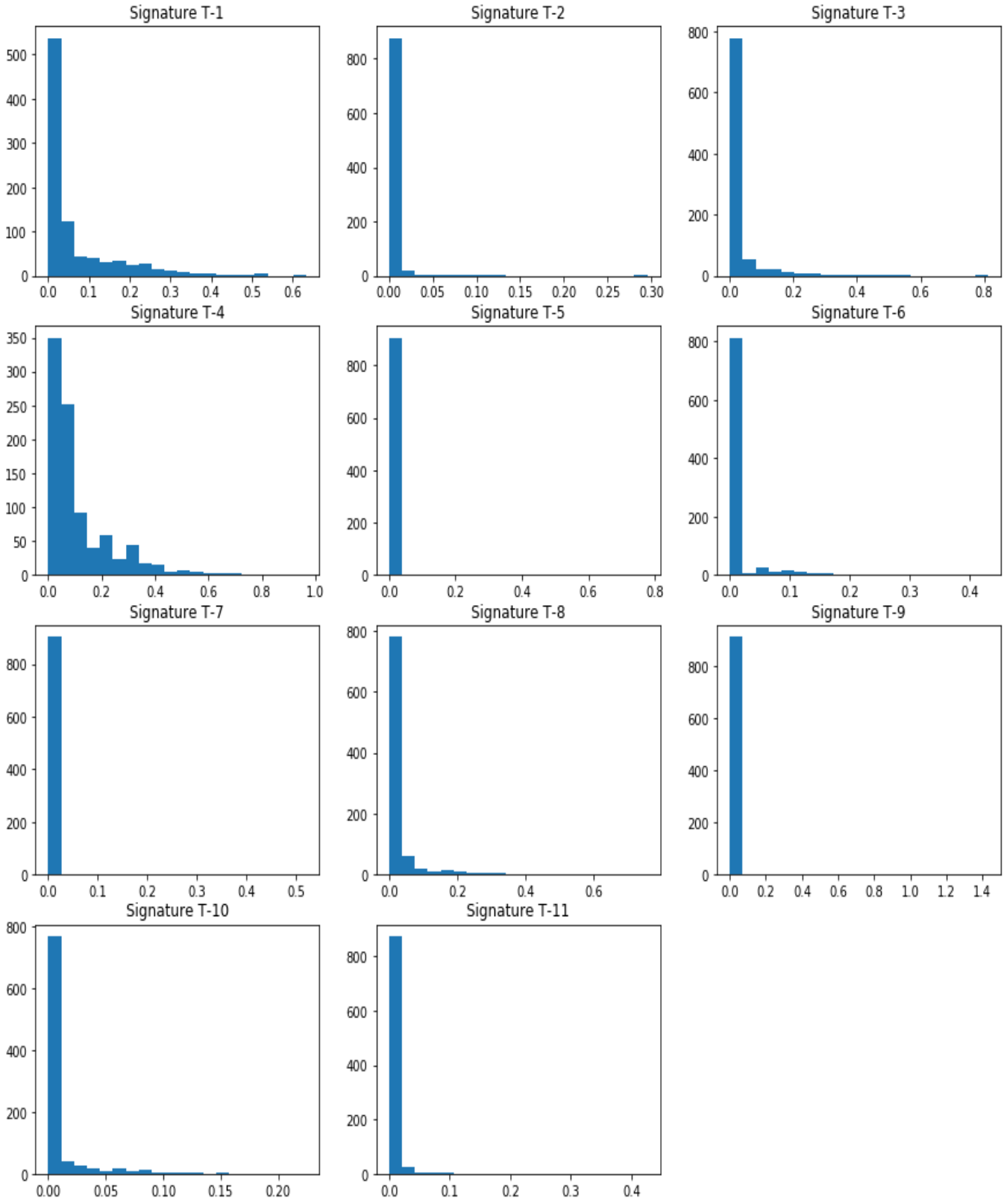

Figure 2: Histograms for the activities of mutational signatures. In each subplot, the x-axis represents the dimensionless activity of a mutational signature and the y-axis represents the number of entries within a certain activity range. Each histogram contains 20 bins.

Algorithms and Techniques

All the classification algorithms discussed in the Machine Learning Engineer Nanodegree were considered for the first task. The overall strategy was to try all of them before optimising the most promising one.

A decision tree model is an application of information theory. The entropy of a dataset is a measure of how heterogeneous it is. If every entry is identical, its entropy will be zero. The information gain associated with a split is the weighted average decrease in entropy following the split. A decision tree is trained by splitting a training dataset in a series of binary decisions in order to maximise the overall information content in the final subsets. The optimal combination of binary decisions provides an algorithm to classify new entries. The main advantages of this class of algorithms are the ease of interpretation and ability to handle non-linear data. The downside is that it is easy to overfit.

A naïve Bayes classifier is a conditional probability model derived from Bayes' theorem. Given a multiclass dataset, it uses the frequency of each class as the corresponding prior probability and the joint distribution of features in that class as the likelihood. Assuming the features are independent, it can then predict the probability that a new entry belongs to a class just from its features. Naïve Bayes algorithms are easy and fast to implement, partly because they do not have any hyperparameters; does not require a large quantity of training data; and can handle many features. However, the assumption of independence is a strong one. In the dataset under consideration, while the mutational processes are independent, they may converge downstream in common cancer genes or pathways. Although this big drawback was recognised in the planning stage, it was decided that a naïve Bayes classifier should be built because of the ease of implementation.

A support vector machine is a non-probabilistic binary linear classifier. It works by creating a gap to separate entries in the feature space and making the gap as wide as possible. If the data are non-linear, it can apply a kernel method to map the entries to a higher-dimensional space where they are separated linearly. A single support vector machine is a binary classifier, but by constructing multiple classifiers (one-against-one or one-against-the-rest), one can combine the results to classify a new entry into one of multiple classes. A support vector machine can handle non-linear data with the kernel trick; provide a unique solution to a problem because it performs convex optimisation, ensuring that the local minimum is also the global minimum; and its hyperparameters can be fine-tuned to avoid overfitting. However, compared to decision trees, the results generated by support vector machines are hard to visualise and interpret, making feature selection more difficult.

A perceptron or neural network combines inputs linearly, adds a bias, and passes the result to an activation function. When the outputs of one layer of perceptrons are fed to another layer of perceptrons and so on, the result is a deep neural network. In a deep neural network, the first layer consists of the inputs, the last layer consists of the outputs, and there can be one or more hidden layers sandwiched between them. If every node in one layer is connected to every node in the next, the architecture is that of a multilayer perceptron. By contrast, architectures such as those of convoluted neural networks can convolve their inputs selectively into features. The 11 features in the dataset under consideration are not spatially related, so a multilayer perceptron is more appropriate for this project. A deep neural network is trained by iterating between feedforward and backpropagation, meaning

training inputs are fed to the network to predict their outputs, and the errors are used to change the weights and biases by gradient descent. Recently, deep learning has outperformed more traditional machine learning algorithms in many domains [5, 6]. However, deep neural networks require a lot of data and computational resources to train. Furthermore, its practice is ahead of the underlying theory, so there is not a unified framework to guide their construction. Although it was decided that a multilayer perceptron would be built, it was not expected to perform well due to the size of the dataset.

AdaBoost is an algorithm which employs an ensemble method. It trains by building a series of weak classifiers, with each weak classifier being tweaked to correct the mistakes made by its predecessors. The weak classifiers are then weighted and combined into a strong classifier. The main advantage of this algorithm is its ability to use diverse algorithms as weak classifiers. The main disadvantage is that overfitting is a concern when there are too many weak classifiers. It was decided that AdaBoost would be used for the refinement of the most promising non- AdaBoost classifier.

A clustering method had to be chosen for the second task. Two major factors guided the selection process. First, the dataset is relatively small, especially the subsets; the OS subset has 19 entries only. Second, the number of subclusters in each histotype was an absolute unknown.

On account of the second factor, k-means clustering and Gaussian mixture models were ruled out. It should be noted, however, that Gaussian mixture modelling is a better choice than k-means clustering because it is probabilistic (soft clustering) and does not favour any cluster shapes (k-means clustering favours compact and circular clusters).

DBSCAN has the advantages of not having the number of clusters as a hyperparameter and not favouring any cluster shapes. However, it also classifies entries in low-density regions as outliers. While this is an advantage when a large and noisy dataset is available, the dataset used in this project is relatively small, so it was also ruled out.

Hierarchical clustering works by treating every entry as a singleton cluster before merging one pair of clusters at a time. The advantages are its deterministic nature and more importantly for this project, the ability to reveal hierarchical structures within a histotype: the second goal. Therefore, it was adopted.

Principal component analysis was chosen over random projection and independent component analysis for the third task. The suitability of mutational signatures as biomarkers of cancer histotypes was being questioned. If a biomarker is bad, it means it is far from the causal factor of a cancer histotype. In other words, the suspicion was that the mutational signatures independently induce distinct somatic alterations, but these alterations affect common histotype-defining genes or pathways. For example, if one patient is exposed to ultraviolet radiation only and another is exposed to tobacco only, both may go on to develop OS cancers if ultraviolet radiation and tobacco mutate the same genes in different ways.

Principal component analysis works by combining features to form latent features that capture the variance in a dataset better; it is a great way of identifying redundancies in the features that come with a dataset. It was decided that this was exactly what had to be done to complete the third task.

Random projection is a computationally efficient way of reducing dimensionality. It is useful when there are many dimensions, a scenario which does not apply to the dataset used in this project. Since

this method picks its basis vectors randomly, they cannot capture as much information as those picked by principal component analysis.

Independent component analysis works by separating multiple signals into statistically independent and non-Gaussian components. Within the context of this project, the signals are the features. Considering the complex nature of cancer signalling pathways [7], the assumption of statistical independence is unlikely to hold. In fact, this method is more suitable for the opposite problem: separating signals in cancer genomes into independent mutational processes. The deciphering method [2] is very similar to independent component analysis and in the paper, the problem is explicitly described as a blind source separation problem.

Benchmark

For the first task, no meaningful benchmarks exist in the literature. The classifiers built in this project are extensions of the deciphering method [2], so they can only be assessed in this context. Since no one has built classifiers for the same purpose, the best benchmark model is a process whereby a histotype is assigned to a cancer with a probability equalling its frequency in the dataset, regardless of its mutational catalogue.

A suitable benchmark for the second task was found in the dataset itself. There are six histotypes in the dataset. It was concluded that hierarchical clustering would be performed on the whole dataset, and the variances in the final five mergers would be averaged. This average would serve as a benchmark for the genetic diversity within the histotypes.

For the third task, a hypothetical set of eigenvectors with the same eigenvalue was chosen. The components represented by them divide the variance in the dataset equally. The hope was that the 11 features could be combined by principal component analysis to form less balanced and more varied latent features.

III. Methodology

Data Preprocessing

After the dataset was loaded into Spyder, an integrated development environment for Python 3.5, all unnecessary columns and rows were dropped from it, including annotations and every column other than the label and 11 features. The remaining columns were renamed to indicate the signatures they represent or that it contains the histotypes. Then, after a copy of the edited dataset was made, the original was split into six subsets by histotype. The original dataset was split into a matrix containing the 11 feature columns and a label vector. The label column was dropped from every histotype subset.

The label vector from the original dataset was label-encoded. After this step, a label vector with encoded histotypes was obtained. In this column, 0 stands for AML; 1, B-ALL; 2, NBL; 3, OS; 4, T-ALL; 5, WT.

The feature matrix and six subsets were logarithmically transformed, followed by normalisation. In the transformed matrices, the values in each column range from zero to one. For example, the sixth entry is a B-ALL cancer; the activities of its first, fourth, and sixth mutational signatures are 0.035, 0.077, and 0.105 respectively, while the remaining mutational signatures are inactive.

A copy of the transformed feature matrix was made for the second and third task. Then, the original transformed feature matrix and label-encoded label vector were split into two subsets, with all correspondences between the matrix and vector being preserved in the subsets. The testing subset contains a quarter of the entries. Stratified sampling was performed to ensure the imbalanced distribution of histotypes was preserved in both subsets.

Implementation

For the first task, the benchmark classifier was built using the scikit-learn model named DummyClassifier. It was fitted to the training data.

The scikit-learn model named DecisionTreeClassifier was used to build a decision tree classifier. This algorithm has three hyperparameters: the maximum tree depth or the maximum number of binary decisions allowed, the minimum number of entries in a leaf node, and the minimum number of entries in a decision node. They were decided by a grid search. A wide range of values were tried at first. For the maximum tree depth, one, five, 10, 15, 20, 25, 50, 100, 200, 400, and 800 were used in the first attempt. The minimum number of entries in a leaf node was informed by the training data. There are 14 or 15 OS entries in the training data, so this minimum had to be set at or below 15 to ensure every leaf node had the potential to become homogeneous after training. In the first round, all the integers from one to 15 were tried. The minimum number of entries in a decision node had to be set above the other minimum, so two, four, eight, 16, 32, and 64 were tried in the first round. A stratified five-fold cross validation was carried out to test each combination of hyperparameters. For each combination, the training data were split into five subsets by stratified sampling; the decision tree was trained and validated five times, each time being validated against a different subset; the five F1 scores were averaged to produce an overall validation F1 score for the combination. After the first round, a validation F1 score of 0.713 was achieved and the three hyperparameters were found to be 20, one, and two. A second round of grid search was performed in the vicinity of these values. The process was repeated one more time. The optimal combination of hyperparameters is 19, one, and two; its validation F1 score was found to be 0.727.

The naïve Bayes algorithm does not contain any hyperparameters, so a grid search was not performed. The scikit-learn model named GaussianNB was fitted to the training data. This implementation of the naïve Bayes algorithm uses a Gaussian distribution to fit continuous, multiclass data to estimate the likelihoods associated with different classes.

A linear support vector machine was built. The scikit-learn model named SVC was used. A grid search was performed to optimise the hyperparameter *C*: the relative significance of minimising the classification error and maximising the margin, with a larger *C* meaning classification is more important. As before, the first attempt involved a wide range: 0.01, 0.1, one, 10, and 100. As before, a stratified five-fold cross validation was carried out for each hyperparameter. The first round resulted in a validation F1 score of 0.679 and a *C* value of 100. After several rounds of grid search, with each round getting narrower in range, a final validation F1 score of 0.681 and *C* value of 44 were obtained.

Another support vector machine was built with a polynomial kernel. The same scikit-learn model was used. This time, the grid search involved two hyperparameters: *C* and *degree*. *C* works just like it does in a linear support vector machine and *degree* is the degree of the polynomial kernel function. A grid

search was performed in the same manner as in the other cases. In the first round, 0.01, 0.1, one, 10, and 100 were all tried for *C*. The degree of the polynomial kernel function was varied from one to ten. The validation F1 score was found to be 0.626; *C*, 100; and *degree*, two. The optimal hyperparameters are 14900 and two; the corresponding validation F1 score is 0.754.

The final support vector machine was built with a radial basis function kernel. The same scikit-learn model was used. This time, the grid search involved two hyperparameters: *C* and *gamma*. *C* works just like it does in the other two support vector machines and *gamma* is inversely proportional to the variance of the Gaussian function used by the algorithm. A grid search was performed as usual. In the first attempt, *C* was set to 0.01, 0.1, one, 10, and 100, while *gamma* was set to 0.01, 0.1, one, 10, and 100. This initial attempt returned *C* at 100 and *gamma* at one; the validation F1 score was found to be 0.754. The optimal hyperparameters are 104 and 0.74; the corresponding validation F1 score is 0.770.

An attempt was made to build a multilayer perceptron using a library called Keras. However, due to the relatively small size of the dataset, deep learning was considered impractical, so it was not refined in detail. The architecture consists of three fully connected layers, an input layers with 11 nodes, a hidden layer with eight nodes, and an output layer with six nodes. The rectifier was chosen to be the activation function of the hidden layer to avoid vanishing gradients in the loss or error function; the softmax function was picked for the output layer to accommodate the presence of six classes. In order to prevent overfitting, an L2 regulariser was added to the hidden and output layers, and dropout (10 %) was applied to the hidden layer. The stochastic gradient descent optimiser with learning rate decay (to aid convergence) and momentum (to avoid local minima) was used to train the model by minimising the categorical cross entropy: the loss or error function. The training data were split into two parts, with a fifth being reserved for validation after every epoch; the remaining data were fed to the neural network in one batch every epoch. Up to 1000000 epochs were allowed, but early stopping was employed to avoid overfitting and save computational resources.

For the second task, hierarchical clustering was performed multiple times. Specifically, Ward's method was employed, meaning the variance between two clusters was used to measure the inter-cluster distance. Because dendrograms were desired for the sake of visualisation, the method named linkage from the SciPy library was used for this task.

It was performed on the copy of transformed feature matrix which had been set aside in the first task. Since there are six histotypes in the dataset, the variances of the final five mergers were averaged to produce a benchmark: 11.3, a proxy of the genetic variations between the six histotypes.

Then, hierarchical clustering was performed on each of the six subsets. The variances of the final five mergers in all cases are shown in table 1. Unsurprisingly, the intra-histotype variations are less than the inter-histotype variations. However, they are within the same order of magnitude. If 50 % of the benchmark is the level of genetic difference beyond which clonal selection is effective, there are five clones within the B-ALL histotype; three, AML; and three, NBL.

That identically labelled entries are not evenly distributed in the feature space is a statistical fact. However, this fact *per se* does not carry much weight. Links must be made to the mutational catalogues, mutated genes, and phenotypes of the cancers before conclusions can be made. However, if the intra-histotype subclusters do represent cancers with different phenotypes, evidence of clonal

evolution may be buried in the dataset. For example, one of the clones with the AML histotype may be the evolutionary ancestor of the rest, meaning the cancers represented by the first clone were sampled at an earlier progression stage than the others.

For the third task, a principal component analysis was performed on the transformed feature matrix. If the 11 original features share the information in the dataset equally, the components should do the same. Figure 3 shows that this is not the case. The first component can explain more than 27 % of the variance in the dataset. The first six components explain almost 90 % of the variance. Biologically, one interpretation is that the 11 mutational processes affect common genes or pathways independently, so some of the mutational processes are redundant. The principal components are latent features, which are potentially the activities of the genes or pathways affected by the mutational processes. However, this is only a hypothesis. To test it, one needs to study the raw sequences, a task way beyond the scope of this project.

Table 1: The variances of the final five mergers in the hierarchical clustering of cancers of different histotypes. B-ALL stands for B-lineage acute lymphoblastic leukaemia; T-ALL, T-lineage ALL; AML, acute myeloid leukaemia; NBL, neuroblastoma; WT, Wilms tumour; and OS, osteosarcoma. The values that exceed half the inter-histotype genetic variations are in bold.

Merger\Histotype	B-ALL	T-ALL	AML	NBL	WT	OS
Final	**8.55**	3.29	**7.44**	**6.82**	4.91	3.23
Second last	**7.06**	3.29	**6.59**	**5.73**	4.25	2.32
Third last	**6.16**	0.993	5.49	4.55	3.99	1.99
Fourth last	**5.98**	0.519	4.58	4.43	3.32	1.44
Fifth last	5.60	0.489	4.42	4.40	2.86	1.36

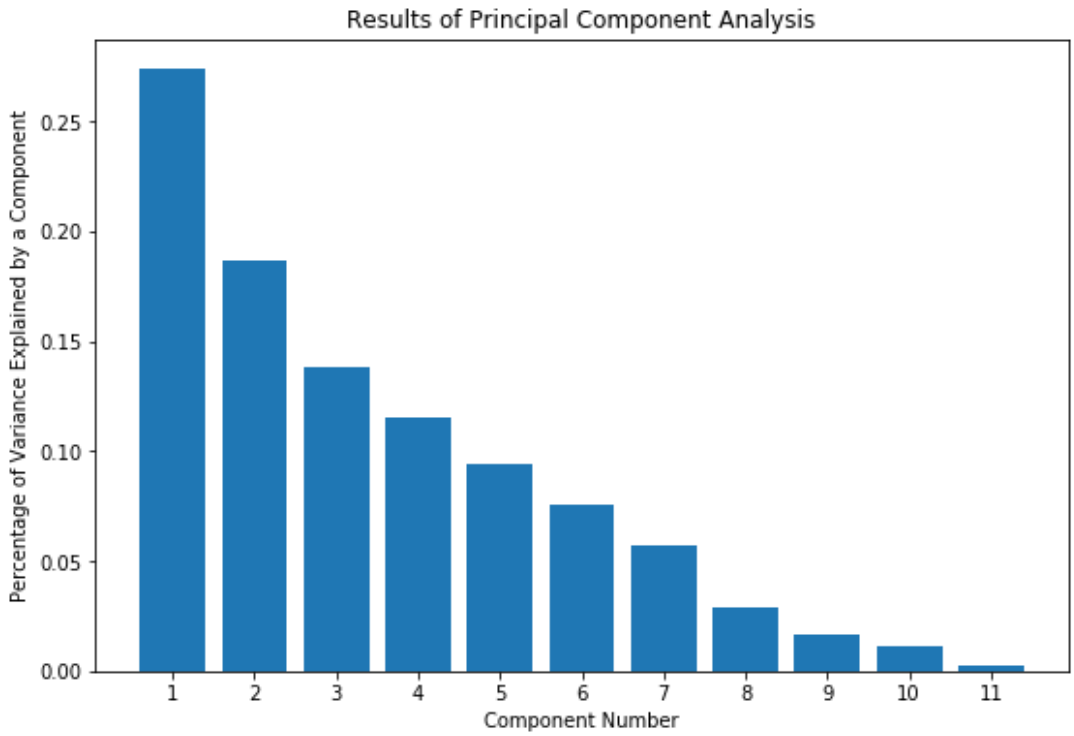

Figure 3: Results of the principal component analysis of the whole dataset. The components are arranged in decreasing order of information content, as indicated by the component numbers. The y-axis shows the percentage of variance in the original dataset that is captured by a component.

Refinement

For the first task, an ensemble method (bagging and boosting) was adopted as the strategy for refinement. The AdaBoost algorithm from the scikit-learn library was used to implement it.

The first decision was of course over the base classifier. The naïve Bayes classifier was quickly ruled out. As explained, the assumption of independence is hard to justify for the dataset under consideration, so it was decided that it was not worth refining. The multilayer perceptron was also ruled out. First, deep learning is not the best method for a dataset with fewer than 1000 entries: the parameters cannot be trained or validated adequately. Second, deep neural networks are complex in the sense that there are many hyperparameters. For any ensemble methods, a simple base classifier is desirable because it allows better control over the ensemble model's complexity: one can simply increase the number of weak learners and ignore other factors. A better way of improving the multilayer perceptron is to experiment with its architecture and hyperparameters, including the optimiser. The support vector machine with a radial basis function kernel was considered because its validation F1 score is higher than the other support vector machines and the decision tree classifier. However, it was ultimately not chosen because it is a strong classifier [8]. The decision tree classifier was picked for refinement. because random forest models had been shown to be effective in genomics [9].

The optimised decision tree classifier has a maximum depth of 19, a minimum number of entries *per* leaf node of one, and a minimum number of entries *per* decision node of two. However, ensemble methods work better with simple base classifiers. Therefore, different maximum depths were tried in a

grid search. The other hyperparameter considered in the grid search was the number of weak learners. For each maximum depth, two, three, four, five, six, seven, eight, nine, 10, 50, 100, 200, 400, 800, and 1000 weak learners were tried at first. The best validation F1 score was found for a maximum depth of seven. Then, with this maximum depth, further rounds of grid search were performed to optimise the number of weak learners. Overall, the best ensemble of decision tree classifiers contains 97 weak learners, each of which has a maximum depth of seven. Its validation F1 score is 0.794.

As explained, refinement of the solutions to the second and third task requires further experimental research or at least a thorough study of the raw sequences from which the dataset was derived.

IV. Results

Model Evaluation and Validation

A series of classifiers were built for the first task, including a decision tree classifier, a naïve Bayes classifier, three support vector machines, a multilayer perceptron, and an ensemble of decision trees. The testing subset of data, which had been set aside until the final evaluation, was used to assess their performance. Features were fed from the testing data to each model, which tried to predict the histotypes. Seven F1 scores were calculated from the results, one for the ability to predict each histotype and one overall score which is an unweighted average of the other six. Table 2 summarises these results and the following bullet points describe the key properties of the classifiers.

- The decision tree classifier has a maximum depth of 19, a minimum number of entries *per* leaf node of one, and a minimum number of entries *per* decision node of two. Its validation F1 score is 0.727; testing F1 score, 0.800.
- The naïve Bayes classifier has a testing F1 score of 0.414.
- The linear support vector machine has a *C* value of 44. Its validation F1 score is 0.681; testing F1 score, 0.686.
- The support vector machine with a second-order polynomial kernel has a *C* value of 14900. Its validation F1 score is 0.754; testing F1 score, 0.764.
- The support vector machine with a radial basis function kernel has a *C* value of 104 and a *gamma* value of 0.74. Its validation F1 score is 0.770; testing F1 score, 0.752.
- The multilayer perceptron has a testing F1 score of 0.755.
- The ensemble classifier contains 97 decision trees whose maximum depth is seven. Its validation F1 score is 0.794; testing F1 score, 0.800.

The results are largely as expected. The naïve Bayes classifier is way worse than the rest and should not be used to categorise new cancers. The linear support vector machine is less predictive than those with kernels, so it should not be used either. The ones with kernels are reliable classifiers except the one with a radial basis function kernel is an unreliable detector of WT cancers; they should be kept for clinical or research use. The only real surprise is the respectable performance of the multilayer perceptron despite the scarcity of data and the lack of refinement; this classifier should be the focus of any future studies. Although it is not a good detector of WT cancers, it is of clinical or research use.

The decision tree classifier and ensemble classifier are equally good according to the metric. Furthermore, they have different strengths. It should be noted that the ensemble classifier has a higher validation F1 score than the decision tree classifier: 0.794 versus 0.727. It is fair to say that the ensemble classifier is more robust than the decision tree classifier despite their identical testing F1 score. On the other hand, the ensemble classifier is an unreliable detector of OS cancers. In fact, it is even worse than the linear support vector machine in this area.

Table 2: The ability of each classifier to predict each histotype. DT represents the decision tree classifier; NB, the naïve Bayes classifier; SVM (linear), the linear support vector machine; SVM (polynomial), the support vector machine with a second-order polynomial kernel; SVM (rbf), the support vector machine with a radial basis function kernel; MLP, the multilayer perceptron; and EC, the ensemble classifier. B-ALL stands for B-lineage acute lymphoblastic leukaemia; T-ALL, T-lineage ALL; AML, acute myeloid leukaemia; NBL, neuroblastoma; WT, Wilms tumour; and OS, osteosarcoma. Each number is a testing F1 score and the final column contains the row averages. The maximum value in each column is in bold. Low values for otherwise reliable classifiers are in red.

Classifier\Histotype	B-ALL	T-ALL	AML	NBL	WT	OS	Overall
DT	**0.782**	**0.985**	0.694	0.857	0.591	**0.889**	**0.800**
NB	0.036	0.892	0.603	0.490	0.186	0.278	0.414
SVM (linear)	0.562	0.903	0.667	0.781	0.313	**0.889**	0.686
SVM (polynomial)	0.722	0.964	0.647	0.845	0.516	**0.889**	0.764
SVM (rbf)	0.739	0.964	0.692	0.857	0.370	**0.889**	0.752
MLP	0.764	0.934	0.667	0.848	0.429	**0.889**	0.755
EC	0.763	**0.985**	**0.727**	**0.871**	**0.703**	0.75	**0.800**

Therefore, the chosen models are the ensemble classifier, decision tree classifier, support vector machine with a second-order polynomial kernel, support vector machine with a radial basis function kernel, and multilayer perceptron. It was decided that the ensemble classifier is the best model, with the decision tree classifier being marginally worse. One caveat is that the latter is a significantly better detector of OS cancers.

In research or clinical use, the five chosen models should be used in conjunction. When a new cancer is presented, the activities of its mutational signatures should be fed to all five classifiers. If the five predictions are unanimous, that will be the result. Otherwise, their predictions should be weighted. If one of the support vector machines or the multilayer perceptron predicts a B-ALL, T-ALL, AML, NBL, or OS cancer, that prediction will count as one vote. If the support vector machine with a second-order polynomial kernel predicts a WT cancer, that prediction will count as one vote too. If the support vector machine with a radial basis function kernel or the multilayer perceptron predicts a WT cancer, that prediction should be treated as half a vote because both classifiers are known to be unreliable detectors of WT cancers. If the decision tree predicts a B-ALL, T-ALL, AML, NBL, or OS cancer, its prediction should be treated as two votes due to its superior performance. However, if it predicts a WT cancer, its

prediction should be treated as 1.5 votes because it is worse than the ensemble classifier but better than the other three in this area. If the ensemble classifier predicts a B-ALL, T-ALL, AML, NBL, or WT cancer, its prediction should be treated as two votes because of its superior performance. However, if it predicts an OS cancer, that prediction should count as half a vote because it is an unreliable detector of OS cancers.

Justification

The benchmark classifier was fed features from the testing data and tried to predict the histotypes. Seven F1 scores were calculated from the results, one for the ability to predict each histotype and one overall score. The unweighted average F1 score is 0.120, while the F1 for predicting AML is 0.226; B-ALL, 0.176; NBL, 0.031; OS, 0; T-ALL, 0.284; and WT, 0.

Overall, the ensemble classifier, the best model out of the five chosen ones, outperforms the benchmark classifier almost seven times. For the individual histotypes, the ensemble classifier is at least three times better than the benchmark classifier. Therefore, the solution is significant enough, especially when the five chosen classifiers are used in the manner described in the last subsection.

The second task was considered speculative from the outset. A benchmark was obtained by performing hierarchical clustering on the whole dataset and averaging the variances of the final five mergers. The result, 11.3, is a measure of the genetic variations between the six histotypes. Within three histotypes; B-ALL, AML, and NBL; the genetic variations are at least half as high as the benchmark. Therefore, the goal of quantifying intra-histotype genetic diversity has been achieved. The hypotheses that there are clones in at least half of the six histotypes and that they may have evolutionary relations are reasonable too. The validation of these hypotheses goes beyond machine learning and enters the realm of experimental research.

The third task was also considered speculative from the outset. The first principal component explains more than 27 % of the variance in the dataset and the first six principal components explain almost 90 % of the variance. This is way higher than the benchmark of roughly 9 % for each component. Therefore, it is reasonable to argue that there are more descriptive biomarkers than the 11 mutational signatures, thus satisfying the third goal. The hypothesis that the latent features are the activities of the mutational processes' common targets is also reasonable. The search of these biomarkers goes beyond the scope of this project and requires a separate study of the raw sequences.

V. Conclusion

Free-Form Visualisation

The hierarchical structures within the histotypes are worth visualisng. In the B-ALL histotype, there are five potential clones whose genetic variations are more than half of the inter-histotype variations. In the AML histotype, three. In the NBL histotype, three. The dendrograms for these three subsets of data are shown in figures 4, 5, and 6.

Dendrograms are powerful tools for visualising evolutionary relations. They help one formulate hypotheses when nothing is known about the hierarchical structures within a dataset, such as the second task. For example, in figure 4, the yellow cluster is close to and significantly bigger than the purple

one. Assuming the cancers were diagnosed and sampled from all progression stages at the same frequency, mutations acquired at an early stage must be overrepresented in the dataset. It is possible that the purple cluster represents a clone that branched off the clone represented by the yellow cluster.

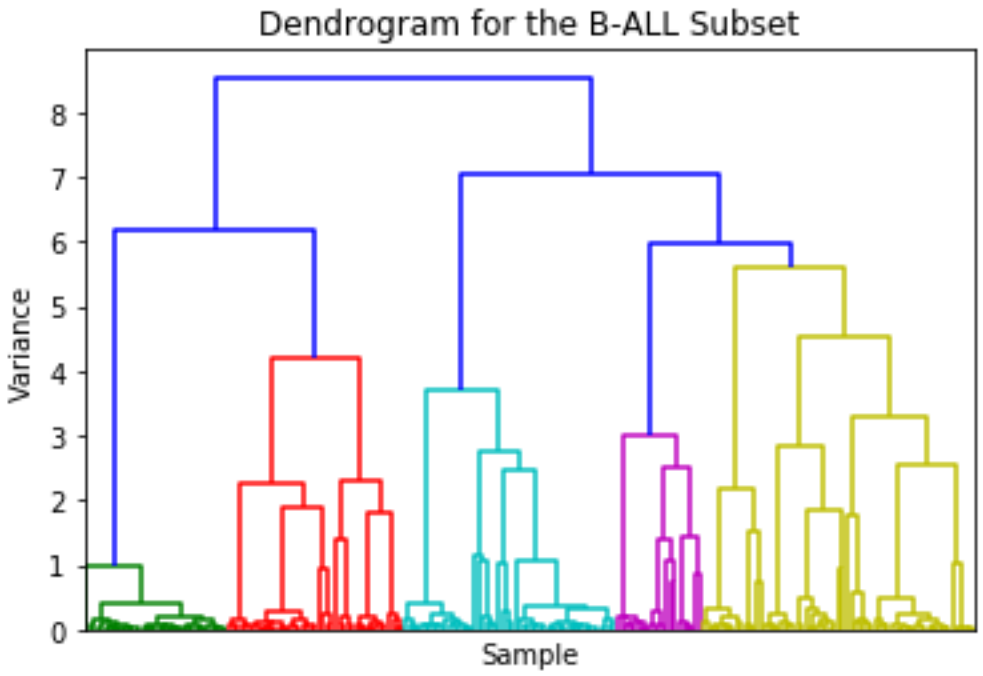

Figure 4: Dendrogram for the B-ALL subset of data. Each node on the x-axis represents a sample recorded in an entry in the subset. The y-axis shows the inter-cluster distances in terms of variance.

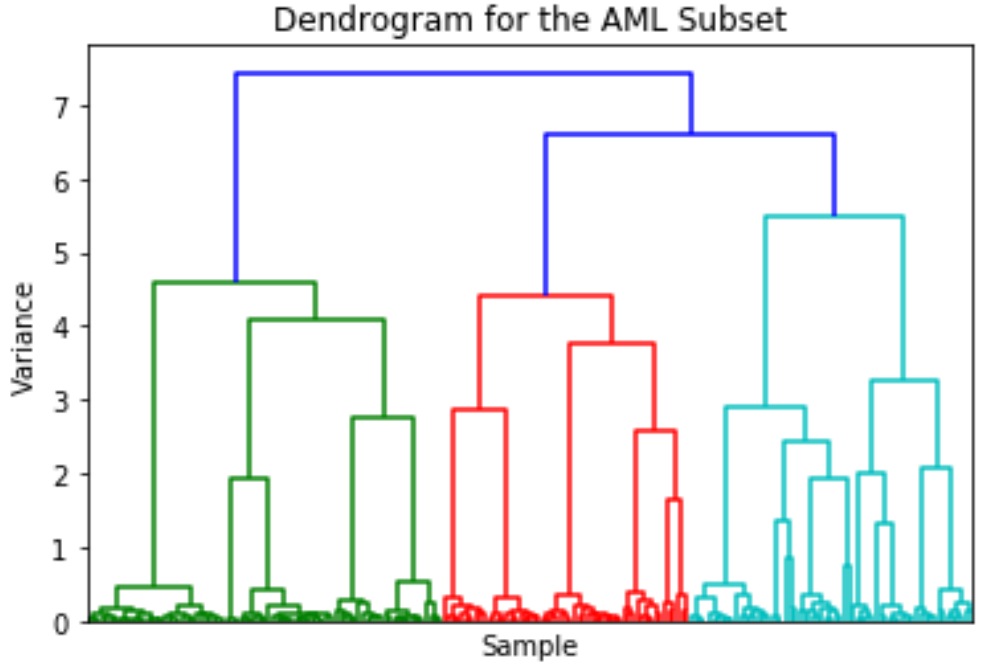

Figure 5: Dendrogram for the AML subset of data. Each node on the x-axis represents a sample recorded in an entry in the subset. The y-axis shows the inter-cluster distances in terms of variance.

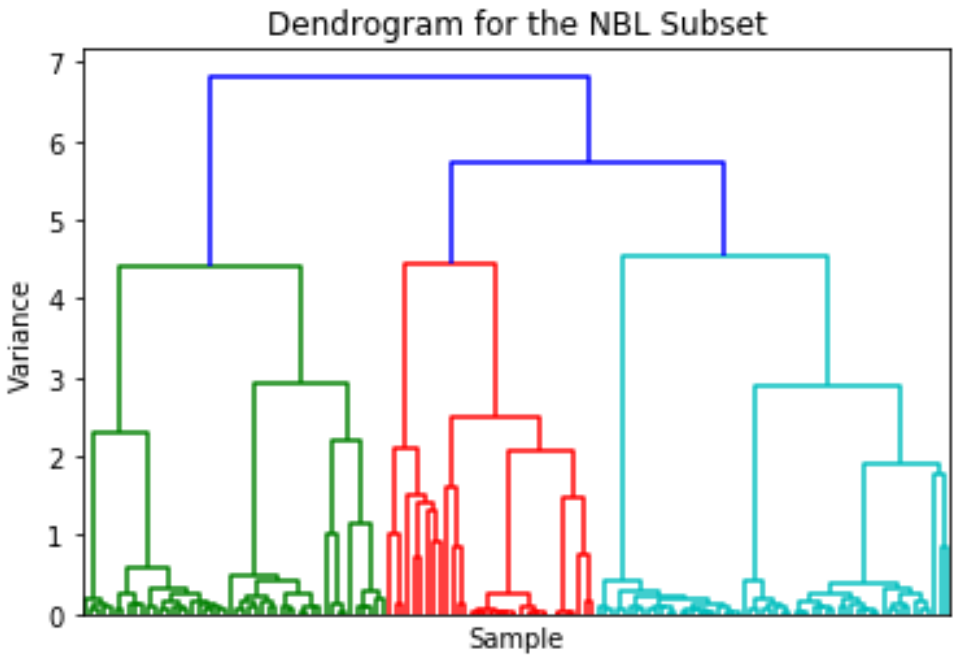

Figure 6: Dendrogram for the NBL subset of data. Each node on the x-axis represents a sample recorded in an entry in the subset. The y-axis shows the inter-cluster distances in terms of variance.

Reflection

The whole process was divided into seven parts. First, I found a dataset in an area I was interested in: cancer genomics. Second, I studied the dataset in order to understand its features and label. This required me to read the journal article about the dataset and the computational method used by the authors to decipher the mutational signatures from raw DNA sequences. Third, I looked for areas of improvement in the study: the evolving nature of cancers, meaning the histotype assigned to a cancer at the time of diagnosis may become irrelevant later; the intra-histotype genetic diversity had not been considered by the authors; and the mutational signatures might not be the best biomarkers for the histotypes in the study. Fourth, I came up with a list of algorithms and techniques which could address these issues. Fifth, I assigned the algorithms and techniques to the three tasks. Sixth, for each task, I watched the Udacity videos about the assigned algorithms and techniques, designed a metric and benchmark, and then applied the algorithms and techniques. Seventh, from the results, I drew conclusions about the three tasks: the best combination of classifiers, a quantitative estimate of the intra-histotype genetic diversity, and the potential existence of common targets of the mutational processes.

I enjoyed learning about the differences between principal component analysis and independent component analysis. Only after learning how the mutational signatures had been deciphered from mutational catalogues did I realise the method is a kind of independent component analysis. Armed with this insight, I realised that I was looking for something more abstract than the mutational catalogues: the common targets of the distinct somatic alterations induced independently by the mutational processes. I also enjoyed using dendrograms to find out potential evolutionary relations; it is a powerful method in the study of clonal evolution.

A challenge I faced was the installation of libraries. For example, I started the project with Python 3.7. When I decided to build a multilayer perceptron, I downloaded TensorFlow; it turned out to be incompatible with Python 3.7. I had to revert to Python 3.5 just to use one library. Another challenge came from the practical nature of machine learning. More than once, I found myself optimising things by trial and error because a theoretical framework was missing.

Improvement

A better classifier can probably be built by optimising the multilayer perceptron. It was not done because of the relatively small size of the dataset, meaning it was not expected to be effective. Furthermore, because of the complexity of deep learning, a standalone project dedicated to this algorithm is needed to optimise it. As it turned out, however, even without refinement and access to more data, the multilayer perceptron was found to have comparable performance to the optimised support vector machines.

After sampling more cancers and expanding the dataset by an order of magnitude, it will be a good idea to experiment with the architecture and hyperparameters of the multilayer perceptron. It is highly likely that it will outperform the new benchmark: the ensemble classifier.

Bibliography

[1] Stratton, M.R., Campbell, P.J. and Futreal, P.A., 2009. The cancer genome. *Nature*, *458*(7239), p.719.

[2] Alexandrov, L.B., Nik-Zainal, S., Wedge, D.C., Campbell, P.J. and Stratton, M.R., 2013. Deciphering signatures of mutational processes operative in human cancer. *Cell reports*, *3*(1), pp.246-259.

[3] Ma, X., Liu, Y., Liu, Y., Alexandrov, L.B., Edmonson, M.N., Gawad, C., Zhou, X., Li, Y., Rusch, M.C., Easton, J. and Huether, R., 2018. Pan-cancer genome and transcriptome analyses of 1,699 paediatric leukaemias and solid tumours. *Nature*, *555*(7696), p.371.

[4] Greaves, M. and Maley, C.C., 2012. Clonal evolution in cancer. *Nature*, *481*(7381), p.306.

[5] Gosasang, Veerachai, Watcharavee Chandraprakaikul, and Supaporn Kiattisin. "A comparison of traditional and neural networks forecasting techniques for container throughput at Bangkok port." *The Asian Journal of Shipping and Logistics*27.3 (2011): 463-482.

[6] Angermueller, Christof, et al. "Deep learning for computational biology." *Molecular systems biology* 12.7 (2016): 878.

[7] Sanchez-Vega, Francisco, et al. "Oncogenic Signaling Pathways in The Cancer Genome Atlas." *Cell* 173.2 (2018): 321-337.

[8] Wang, Yu, and Cheng De Lin. "Learning by Bagging and Adaboost based on support vector machine." *Industrial Informatics, 2007 5th IEEE International Conference on*. Vol. 2. IEEE, 2007.

[9] Chen, Xi, and Hemant Ishwaran. "Random forests for genomic data analysis." *Genomics* 99.6 (2012): 323-329.