MACHINE LEARNING ON SINGLE-CELL RNA-SEQ TO ADVANCE OUR UNDERSTANDING OF CLONAL HEMATOPOIESIS

By

Brian Sharber

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Computer Science

December 16, 2023

Nashville, Tennessee

Approved:

Dr. Douglas C. Schmidt

Dr. Jesse Spencer-Smith

Dr. Alexander Bick

Dr. Jonathan Brett Heimlich

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Clonal Hematopoiesis of Indeterminate Potential (CHIP) is an age-related condition having significant implications on a person's health, as it is associated with an increased risk of hematological malignancies, cardiovascular diseases, and other age-related diseases. Aging brings about changes in hematopoietic stem cells, situated in the bone marrow, leading to the development of genetic mutations. These somatic mutations gradually accumulate in various tissues, given the pivotal role of these stem cells in generating all blood cells. Mutations can be categorized into passenger and driver mutations. Passenger mutations are incidental genetic changes occurring alongside driver mutations, holding little consequence. In contrast, driver mutations are specifically associated with an increased risk of various diseases. Driver mutations instigate clonal expansion, causing an accelerated proliferation of blood cells. While advantageous for the cells, this heightened growth poses potential harm to the overall health of the individual. The significance of CHIP lies in its potential as a predictive biomarker for health outcomes, and has garnered increased attention due to its potential to serve as an early indicator of health risks (2).

From previous studies, we know that there are certain genes that driver mutations occur in the context of CHIP (3). However, these mutations are not uniformly distributed, with around 75% of these mutations occurring in TET2 (Ten-Eleven Translocation 2) and DNMT3A (DNA Methyltransferase 3A), genes that have been previously linked to blood cancer. In the realm of CHIP research, investigation of these two genetic players, and effectively identifying and quantifying the presence of CHIP cells amidst the broader cellular landscape, is challenging.

Single-Cell RNA Sequencing (scRNA-seq) is a groundbreaking genomics technique that examines gene expression within individual cells, revealing intricate cellular diversity and function (4). Applied to CHIP, scRNA-seq uncovers unique RNA expression signatures in mutated blood cells, shedding light on disease mechanisms (5). In order to differentiate CHIP cells from non-CHIP cells, our laboratory at Vanderbilt University Medical Center (VUMC) attempted to utilize single cell sequencing methods to determine the transcriptional phenotype of the mutated cells.

This undertaking proved more challenging than anticipated, primarily due to a technical obstacle. Specifically, the CHIP mutations under scrutiny, DNMT3A and TET2, are not well expressed in terminally differentiated cells such as Peripheral Blood Mononuclear Cells (PBMCs), resulting in scarce RNA transcripts. This predicament indicates conventional methods cannot directly ascertain genetic measurements of DNMT3A and TET2. Therefore, the process of differentiating CHIP cells is not as straightforward as conducting 3'

RNA-seq and identifying cells carrying the mutant RNA transcript. To overcome this hurdle, our laboratory employs a novel approach, utilizing co-occurring mitochondrial DNA Single Nucleotide Variants (SNVs) as barcodes to identify mutant cells within our 3' RNA-seq dataset.

This current methodology makes use of MAESTER (Mitochondrial Analysis and Estimation of Single-cell Transcriptional Error) to accurately attribute genetic variants to specific cells, enabling the identification of CHIP cells (6). It focuses on the mitochondrial genome (mtDNA) and helps identify variants and lineage relationships among single cells. It utilizes a combination of computational techniques to process the data and make inferences regarding cellular diversity and genetic variants.

However, the current methodology is time-consuming and cost-intensive. This prompts a pivotal question: What if the intricate post scRNA-seq processes could be circumvented altogether? This inquiry forms the basis of our study. An approach consisting of training and deploying a machine learning (ML) classifier to proficiently identify CHIP cells could offer a more streamlined and resource-efficient approach at informing and stratifying the risk of patients with CHIP.

1.1 Research Aims

Aim 1: Building a More Cost-Effective Solution - To utilize ML classifiers from sklearn (Random Forest, LinearSVC, etc.) in contrast to current methodologies utilizing MAESTER for identifying CHIP cells. The goal is to determine the economic and logistical advantages of this new process.

Aim 2: Determining the Most Specific Gene/Expression for CHIP - To discover the gene or gene expression that exhibits the highest specificity for CHIP. Identifying the most specific genetic markers for CHIP is essential for precise diagnosis and classification.

Aim 3: Uncovering Distinct RNA Expression Signatures in CHIP Cells - To identify unique RNA expression signatures within CHIP cells and pinpoint a specific cell subtype that serves as the archetypal representation of CHIP in terms of RNA expression. Understanding the specific RNA expression patterns in CHIP cells and identifying a prototypical cell subtype are crucial for characterizing this condition at the molecular level.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Single-Cell RNA Sequencing

In the realm of genomics, Single-Cell RNA Sequencing (scRNA-seq) has emerged as a powerful tool, enabling researchers to delve deep into the intricate world of gene expression at the individual cell level. This technology not only provides insights into cellular diversity but also unravels the underlying molecular mechanisms governing various biological processes. One of the primary challenges in harnessing the potential of scRNA-seq data is managing its complexity. With thousands of genes measured in each cell, these datasets inherently reside in high-dimensional spaces. To navigate this intricate landscape, researchers employ dimensionality reduction techniques. Principal Component Analysis (PCA) is one such approach.

PCA aims to identify linear combinations of genes that capture the primary transcriptional variations between cells. However, PCA is not without its limitations. The limitations of using PCA in scRNA-seq data analysis include its sensitivity to gene expression magnitude, skewness caused by zero counts (cases where a specific gene is not expressed in a particular cell), and reduced effectiveness in high-dimensional scRNA-seq data, necessitating the consideration of alternative techniques such as neural networks and probabilistic models to address these challenges (7).

Another essential aspect of scRNA-seq data analysis is identifying transcriptionally similar cells and grouping them into communities. Neighbor graphs play a pivotal role in this process. These graphs are constructed to link cells that share transcriptional similarities, facilitating downstream analysis and visualization, including the utilization of techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) (8) and Uniform Manifold Approximation and Projection (UMAP) (9) embeddings. These methods are incredibly effective at capturing subpopulations and the inherent structures within scRNA-seq data.

For visualization and in-depth analysis of scRNA-seq data, UMAP and t-SNE are indispensable tools. These techniques create two-dimensional embeddings that represent the underlying structure of the data. What sets UMAP and t-SNE apart is their ability to discount global distances and prioritize the preservation of local neighborhood relationships within the data. As a result, they excel in capturing subpopulations, continuous cellular trajectories, and other intricate structures concealed within the vast landscape of scRNA-seq data. When it comes to exploring feature-to-feature interactions, tools like UMAP and t-SNE provide powerful dimensionality reduction techniques to visualize complex relationships.

However, if our focus is on univariate feature selection, we shift our attention to a different approach.

Univariate feature analysis involves assessing the statistical significance of individual features in isolation, without considering the influence of other features. This evaluation relies on the statistical relationship between each feature and the target variable. This process is typically carried out using pandas dataframes with the programming language Python and class label columns, enabling us to select the most informative features that contribute significantly to predictive models or further analyses. While UMAP and t-SNE help us understand intricate relationships, univariate feature selection helps us pinpoint the essential elements for our specific analytical goals.

Our data science pipeline embarks on the journey of data exploration, analysis, and modeling with univariate feature selection and another goal in mind: to reduce the number of predictors as far as possible without compromising predictive performance. Indeed, this is the goal behind feature selection (10). These objectives drive our study for precision and efficiency in classifying CHIP cells and allow us to uncover the most informative and influential attributes, leaving behind the noise and redundancy that often clutters datasets. By enhancing interpretability, we aim to unlock the patterns hidden within the data while paving the way for precise, cost-effective, and practical diagnostic methodologies.

2.2 Types of Machine Learning Models

Figure 2.1 provides an overview of machine learning algorithms, with this study focusing on supervised learning. The choice of supervised learning is driven by the critical need for interpretability in biological applications. Interpretability plays a pivotal role as it connects computational predictions with biological insights, allowing the identification of key genes, aiding hypothesis generation, and guiding focused experimental design.

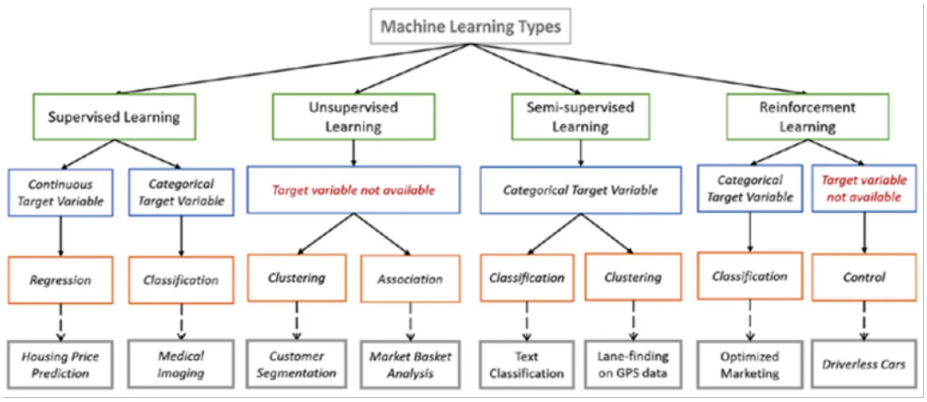


Figure 2.1: Types of Machine Learning Algorithms (1)

In the context of machine learning, supervised learning entails training an algorithm using labeled data to facilitate its ability to make predictions or classify new, unlabeled data by discerning patterns from the

labeled training data. In this research, we harnessed and tested six distinct supervised learning models for the purpose of binary classification.

2.2.1 Bernoulli Naïve Bayes

Naïve Bayes training algorithms are considered rudimentary yet serve as valuable initial steps in classification tasks. These algorithms rely on fundamental probability principles, making a simplistic assumption that all features within a dataset are independent, and subsequently endeavor to classify data. Specifically, the Bernoulli Naïve Bayes model focuses on the presence or absence of features.

2.2.2 AdaBoost

AdaBoost represents a machine learning strategy employing a sequential ensemble technique aimed at data classification. This method combines multiple classifiers to enhance overall predictive accuracy. The core concept involves assigning weights to both the classifiers and data instances in a way that encourages classifiers to prioritize challenging-to-classify observations. AdaBoost uses the algorithm known as AdaBoost-SAMME (11).

2.2.3 Logistic Regression

Binary Logistic Regression models are specifically tailored for distinguishing between two distinct categories, for example, ”Mutant” and ”Wildtype.” The primary objective of this algorithm is to establish a connection between features and the likelihood of a specific outcome. In contrast to linear regression, logistic regression refrains from directly fitting a straight line to the data. Instead, it models the relationship by fitting an S-shaped curve, known as the Sigmoid curve, to the observations.

2.2.4 Support Vector Machine

SVC, the Support Vector Machine classifier implementation using *libsvm* (12), is a powerful tool for classification tasks. It provides robust results for various datasets. LinearSVC (Linear Support Vector Classification) shares similarities with SVC but differs in its underlying implementation. Instead of relying on *libsvm*, LinearSVC is built upon *liblinear* (13). This distinction grants LinearSVC greater flexibility when it comes to selecting penalties and loss functions, making it adaptable to a wide range of problem scenarios. Additionally, LinearSVC exhibits superior scalability when dealing with datasets that encompass a substantial number of samples. This scalability makes it particularly well-suited for handling large and complex datasets where computational efficiency is paramount.

2.2.5 Stochastic Gradient Descent

SGD classifiers represent a class of linear classifiers that employ Stochastic Gradient Descent (SGD) as their training method. By default, these models are configured as linear Support Vector Machines (SVMs). The concept of gradient descent, underpinning SGD, involves an iterative process. It initiates from a randomly chosen point on a mathematical function and incrementally moves down the function's slope in discrete steps, seeking to reach the lowest point, or minimum, of that function. This approach proves particularly valuable when finding optimal points is not feasible through traditional methods like setting the slope of the function to zero.

What sets SGD apart from traditional gradient descent methods is its incorporation of randomness in the descent algorithm. In each iteration, it randomly selects a single data point from the dataset. This stochastic element reduces the overall computational burden, making SGD particularly advantageous when dealing with sizable datasets. Consequently, it enhances the efficiency and applicability of SGD to larger and more complex data scenarios.

2.2.6 Random Forest

Decision Trees represent non-parametric supervised learning techniques employed for both classification and regression tasks. These classifiers can be envisioned as schematic, flowchart-like structures where the journey from the root node to a leaf node signifies the application of classification rules. Each internal node within the tree signifies a test conducted on a particular feature, while each leaf node designates a class label, representing the final decision made after traversing the tree and evaluating all relevant features.

Random Forest constitutes a supervised machine learning algorithm grounded in ensemble learning, a method that combines multiple algorithms of the same type, often utilizing multiple Decision Tree classifiers. This amalgamation results in an assembly of trees, aptly referred to as a "forest." The algorithm's fundamental steps encompass: (a) selecting n random records from the dataset, (b) constructing a decision tree based on this subset, (c) determining the desired number of trees for the algorithm, (d) iterating through steps (a) and (b) until the designated number of trees is achieved, and (e) assigning a category to new records, particularly in the context of classification problems, based on the majority consensus among predictions made by each tree within the forest.

2.3 Evaluation Metrics

Confusion Matrix: A confusion matrix is a table that is often used to evaluate the performance of a classification model. It provides a summary of the predicted and actual class labels for a set of data. Figure 2.2 depicts a straightforward example of a confusion matrix.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 2.2: Confusion Matrix

Precision: Precision gauges the accuracy of positive predictions made by the model by calculating the ratio of true positive predictions (correctly predicted positive instances) to the sum of true positive and false positive predictions.

Recall: This metric assesses the model's ability to correctly identify positive instances by determining the ratio of true positive predictions to the sum of true positive and false negative predictions.

F1-score: The F1-score offers a balanced assessment by taking the harmonic mean of precision and recall. It serves as a useful measure for evaluating the trade-off between precision and recall.

Specificity: Specificity focuses on the model's capacity to correctly identify negative instances, as it quantifies the ratio of true negative predictions (correctly predicted negative instances) to the sum of true negative and false positive predictions.

Accuracy: Accuracy is a fundamental metric used to assess the overall correctness of a model's predictions. It calculates the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset. Accuracy provides an easy-to-understand measure of how well a model performs across all classes, making it a suitable choice when all classes are of equal importance and when there is no significant class imbalance. It is particularly useful in scenarios where achieving a balance between precision and recall is not critical, and the primary goal is to maximize overall correctness. However, it may not be the best metric to use when working with imbalanced datasets, where a class of interest is underrepresented, as high accuracy can be misleading when the model predominantly predicts the majority class.

K-fold Cross Validation: This widely adopted technique is employed for robust performance assessment. It involves dividing the labeled dataset into k equally sized folds and iteratively training and evaluating the model k times, using distinct subsets for training and testing in each iteration. K-fold cross-validation yields a more dependable estimate of the model's performance and helps mitigate the risk of overfitting.

CHAPTER 3

METHODOLOGY

3.1 Data Gathering - scRNA-seq Datasets

The two datasets used in this study collectively serve as both the training and test data for separate CHIP identification tasks. These labelled scRNA-seq datasets were obtained from VUMC as a result of calling MAESTER. The class label column in each dataset, ”Clone”, represents the classification of each cell as either ”Mutant” (indicating the presence of CHIP) or ”Wildtype” (indicating the absence of CHIP). Recognizing the efficiency discrepancy in handling thousands of columns between R and pandas dataframes, we adopted a column-wise reduction strategy.

The TET2 dataset shown in Table 3.1 encompasses data from individuals with a TET2 mutation in their hematopoietic stem cells. This mutation exerts a profound influence, resulting in the transmission of the TET2 mutation to the blood cells generated by the mutated hematopoietic stem cells, which makes up our rows of data. The ”celltype.de” column encompasses various cell types, including CD14 Mono, CD4 T cell, CD8 T cell, NK, gdT, B, CD4 CTL, Treg, DC, ILC, Platelet, Eryth, MAIT, and HSPC. The dataset consists of a comprehensive set of features, including numerical data derived from scRNA-seq of a specific gene Single Nucleotide Polymorphism (SNP), categorical data describing cell types based on reference mapping, and the class label ”Clone”. In total, the dataset comprises 8,585 rows of data.

Features	
Feature	Description
Gene SNP (726 columns)	(Numerical) scRNA-seq data regarding a specific gene SNP.
celltype.de	(Categorical) Cell type based on reference mapping.
Clone	(Class label) Mutant/Wildtype.
Rows of Data	
Class	#Clones
Mutant	4,425
Wildtype	4,160

Table 3.1: Summary of scRNA-seq Data for TET2 Mutated Set

The DNMT3A dataset shown in Table 3.2 encompasses data from individuals with a DNMT3A mutation in their hematopoietic stem cells. Only one cell type, CD14 Mono, is of interest with this dataset, and as such, there is no ”celltype.de” column present. CHIP typically leads to a bias towards myeloid cells. In hematopoiesis, the process of blood formation in the bone marrow, two major cell types are involved:

lymphoid and myeloid cells. When a CHIP mutation is present, it tends to promote a higher proportion of myeloid cells. Therefore, our focus was directed towards investigating these myeloid-skewed cells. The rows of data correlate to CD14 Mono cells that have a DNMT3A mutation in them. In total, the dataset comprises 3,163 rows of data.

Features	
Feature	Description
Gene SNP (2000 columns)	(Numerical) scRNA-seq data regarding a specific gene SNP.
Clone	(Class label) Mutant/Wildtype.
Rows of Data	
Class	#Clones
Mutant	1,478
Wildtype	1,685

Table 3.2: Summary of scRNA-seq Data for DNMT3A Mutated Set

3.2 Pipeline for Determining the Best Classifier for CHIP Detection

To identify the optimal classifier for scRNA-seq data analysis, we utilized the TET2 dataset, primarily due to its larger dataset size in comparison to the DNMT3A dataset. As illustrated in the pseudocode in Figure 3.1, our methodology encompassed a systematic approach of data exploration, pre-processing, and the selection of a range of classifiers from scikit-learn (sklearn). Then, we designed a well-structured pipeline and performed rigorous experimentation. The goal was to pinpoint the classifier that consistently demonstrates superior performance across 30 train/test sessions.

3.2.1 Pre-Processing

Prior to delving into the analysis, we undertake an essential phase of exploratory data analysis and data cleansing. This includes loading the data from the Comma-Separated Values (CSV) file into a pandas dataframe and subsequently removing rows with missing or null values to ensure our analysis is based on complete and reliable data. Additionally, we meticulously eliminate any duplicate entries to eliminate redundancy.

In the realm of scRNA-seq data, a common and pivotal pre-processing step is log normalization. This transformation plays a fundamental role in centering the data and adjusting for right-skewness. Log transformed expression values are widely embraced in the initial analysis of scRNA-seq data due to their straightforward nature and ease of interpretation (14). Log transformed values provide an effective means to approximate log-fold changes in gene expression between individual cells, a metric often more pertinent than raw counts. This becomes especially crucial in processes like clustering and trajectory analysis, where assessing

1. **rna_data**: dataframe converted from data from CSV
2. **class_label**: "Clone"
3. **train/test size**: 0.67/0.33
4. For each row in *rna_data*:
   #Apply cleaning
   4.1. Remove null values.
   4.2. Remove duplicate data.
5. **classification_models**: [Ada Boost, Logistic Regression, LinearSVC, Random Forest, Bernoulli Naive Bayes, SGD]
6. For 30 train/test sessions of the *classification_models*:
   6.1. Randomly shuffle the data.
   6.2. **x_train, x_test, y_train, y_test**: split *rna_data* using train/test size.
   6.3. Use Pipeline object with OneHotEncoder.
   6.4. Use GridSearchCV object applying hyperparameter tuning with 5-fold cross-validation.
   6.5. *GridSearchCV.fit(x_train, y_train)*
   6.6. Inspect average cross-validation score: *np.mean(grid_search.cv_results_['mean_test_score'])*
   6.7. Inspect performance against golden holdout data: *grid_search.best_estimator_.score(x_test, y_test)*
7. Display performance evaluation results: confusion matrix and statistics.
8. Determine best-performing classifier.

Figure 3.1: Pseudocode for Pipeline for Determining the Best Classifier

relative differences in gene expression takes precedence. Furthermore, log transformation helps attenuate the impact of random count fluctuations, particularly in the context of highly abundant genes, where these fluctuations would otherwise introduce substantial yet inconsequential differences between cells.

It is worth noting that the data used in this study was already log transformed before loading the data into a dataframe. Standardization on top of normalization is not necessary in the context of scRNA-seq data, as issues related to scale and variance are addressed from log normalization alone.

3.2.2 Training and Testing of Models

In this analysis, it is important to recognize the balanced nature of false positives and false negatives in the context of CHIP detection. While CHIP is a significant risk factor, it does not provide a definitive indicator of blood cancer, hematological malignancies, cardiovascular diseases, or other age-related diseases. As such, we chose not to weight the data in favor of one outcome over the other, underlining the importance of this research while maintaining a balanced perspective.

To establish reliable and unbiased training and testing sessions for our models, we follow a step-by-step process. Firstly, we randomly shuffle the data to prevent any potential bias. Afterward, the dataset is split

into two segments: the training set, which serves as the foundation for teaching our model, and the test set, which plays a vital role in assessing the model's predictions.

When splitting the data into a training set (0.67%) and testing set (0.33%), we employ stratified sampling on both the "celltype.de" column and the class label column "Clone." This method ensures that the composition of different cell types and the distribution of Mutant and Wildtype labels remains proportional in both the training and test datasets.

To prepare our data for the model's understanding, we implement a "OneHotEncoding" transformation process within a "Pipeline" object. This technique is essential for converting categorical data into a format that the model can work with effectively, improving compatibility and overall algorithm performance.

To fine-tune our model for maximum accuracy, we utilize a "GridSearchCV" object for hyperparameter tuning. This involves an exhaustive search for the most suitable combination of hyperparameters using a n-fold cross-validation approach. The primary goal here is to optimize our model, making it as precise as possible. In this context, we choose 5-fold cross-validation, as the size of our dataset does not necessitate the extra execution time required for a 10-fold cross-validation. Our model then goes through a training phase, where it learns from the data in the training set. It absorbs information and patterns that will enable it to make predictions on unseen data effectively.

To evaluate the model's performance, we consider two key aspects. First, we calculate the average accuracy across five different training and testing splits using 5-fold cross-validation. This metric gives us a robust measure of the model's performance. Second, we examine how well the model performs on the pre-separated test set, often referred to as the "golden holdout" set, as another method of ensuring our model has not overfit on the training data.

In this specific context, we opt for accuracy as our primary metric. This choice aligns with the nature of our data, characterized by a relatively balanced distribution of class labels. Unlike other scenarios where precision-recall or Receiver Operating Characteristic (ROC) curves might be more suitable, accuracy provides a straightforward and comprehensive measure of our model's effectiveness, making it well-suited for our datasets.

To identify the optimal classifier for CHIP detection, we performed 30 independent training sessions for each classifier. We calculated the average cross-validation accuracy for each classifier, a score derived from 5-fold cross-validation applied in each of the 30 runs. We also assessed each classifier's performance against a golden holdout set, meticulously kept separate from the cross-validation process. By analyzing the averaged scores from these diverse evaluations, we identified the classifier consistently demonstrating the greatest robustness and reliability when applied to the dataset.

ML Classifier	%AVG CV Score	%AVG Golden Holdout Set Score
AdaBoost	92.85%	93.62%
LogisticRegression	88.38%	91.44%
LinearSVC	87.39%	91.55%
RandomForest	93.75%	93.95%
BernoulliNaiveBayes	93.55%	93.59%
SGD	89.25%	93.83%

Table 3.3: Models' Accuracy

Taking both the average cross-validation scores and the average golden holdout set scores into consideration, Random Forest emerged as the top-performing classifier, as indicated in Table 3.3. For the feature pruning tasks that lie ahead, we leverage permutation importance, a model-agnostic approach, to effectively reduce noise and uncover the genuinely significant features within scRNA-seq data.

3.3 Pipeline for CHIP Detection

With the best-performing classifier in hand, we move forward to the next phase, building upon the foundation we have established. This involves integrating several additional steps, shown in the pseudocode for the pipeline in Figure 3.2, each designed to refine the model's performance and identify key features in the context of CHIP detection.

One pivotal addition is the integration of permutation importance, an approach recognized for its model-agnostic nature. In this context, "model-agnostic" means that permutation importance is not bound to a specific machine learning model. Instead, it serves as a versatile technique applicable to various models. Permutation importance's core concept is to evaluate feature significance by perturbing features and assessing their impact on the model's performance. Features that, when shuffled, result in a substantial performance drop are deemed more significant.

The primary goal behind introducing permutation importance into our pipeline is feature pruning. We aim to reduce dataset noise, focusing our analysis on the most informative and influential features. Identifying these key features enables us to gain deeper insights into CHIP classification, particularly in the context of the Mutant cells. These features play a crucial role in understanding the genetic components that differentiate Mutant from Wildtype cells in our dataset.

Incorporating permutation importance is a substantial addition to our analysis, significantly impacting the overall execution time. It serves as the rationale behind the development of two distinct pipelines instead of a singular one for both sets of data. To efficiently manage this time-consuming task while balancing thorough

1. **rna_data**: dataframe converted from data from CSV
2. **class_label**: "Clone"
3. **train/test size**: 0.67/0.33
4. For each row in *rna_data*:
   #Apply cleaning
   4.1. Remove null values.
   4.2. Remove duplicate data.
5. **classification_model**: Random Forest
6. For 10 train/test sessions of the *classification_model*:
   6.1. Randomly shuffle the data.
   6.2. **x_train, x_test, y_train, y_test**: split *rna_data* using train/test size.
   6.3. Use Pipeline object with OneHotEncoder.
   6.4. Use GridSearchCV object applying hyperparameter tuning with 5-fold cross-validation.
   6.5. *GridSearchCV.fit(x_train, y_train)*
   6.6. Display performance evaluation results with Confusion Matrix.
   6.7. Apply permutation importance.
   6.8. **pruned_rna_data**: Grab top 30 features indicated by permutation importance. Drop all other features besides *class_label* from *rna_data*.
   6.9. Repeat 6.2-6.6 using *pruned_rna_data*.
7. Display key features and their counts, based on their appearance among the top 30 most significant features across 10 train/test sessions.
8. Perform final train/test sessions with key features with variations based on count.
9. Inspect final performance evaluation results with Confusion Matrix.

Figure 3.2: Pseudocode for the Pipeline for CHIP Detection

analysis and practicality, we modify our approach and deliberately reduce the number of train/test sessions from 30 to 10. This adjustment allows us to save time while still generating a substantial volume of results.

Throughout these 10 runs, we track and record key features consistently deemed highly significant. These features, demonstrating their relevance across multiple sessions, become the focal point of our analysis. The final step involves conducting a single train/test session using these top features, enabling us to extract valuable insights while efficiently managing computational resources.

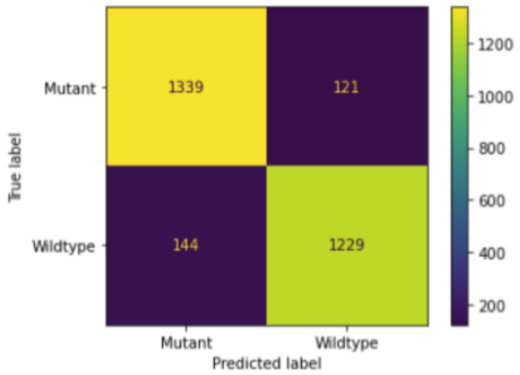
CHAPTER 4

RESEARCH FINDINGS AND DISCUSSION

4.1 TET2 Dataset

Using the CHIP Detection pipeline, we first identify the top 30 most significant features across 10 train/test sessions. Subsequently, we perform final train/test sessions while varying the feature count. The objective is to maintain high accuracy while increasing the feature count, retaining only those features that consistently appear across multiple runs. To gain further insights into the model's accuracy and identify potential areas of improvement, we employ a confusion matrix, a powerful tool for assessing classification results.

In the TET2 dataset, we observed that when using features with a count greater than 3, the Random Forest classifier accurately classified CHIP cells 91% of the time. However, when we increased the feature count to more than 4, the accuracy dropped to 79%. This suggests that the ideal feature count is 3 or greater, as these features are the most influential for prediction. The meticulous feature selection process that guided us through dataset refinement resulted in the reduction of the initial 727 columns to a more focused and streamlined set of 13 key columns. Figure 4.1 illustrates the confusion matrix for the Random Forest model against the testing data, as well as the accuracy, precision, recall, and F1-scores. A visual representation of key features (all representing gene SNPs) is provided in Figure 4.2.



	precision	recall	f1-score	support
Mutant	0.90	0.92	0.91	1460
Wildtype	0.91	0.90	0.90	1373
accuracy			0.91	2833
macro avg	0.91	0.91	0.91	2833
weighted avg	0.91	0.91	0.91	2833

Figure 4.1: Final Confusion Matrix - TET2 Dataset

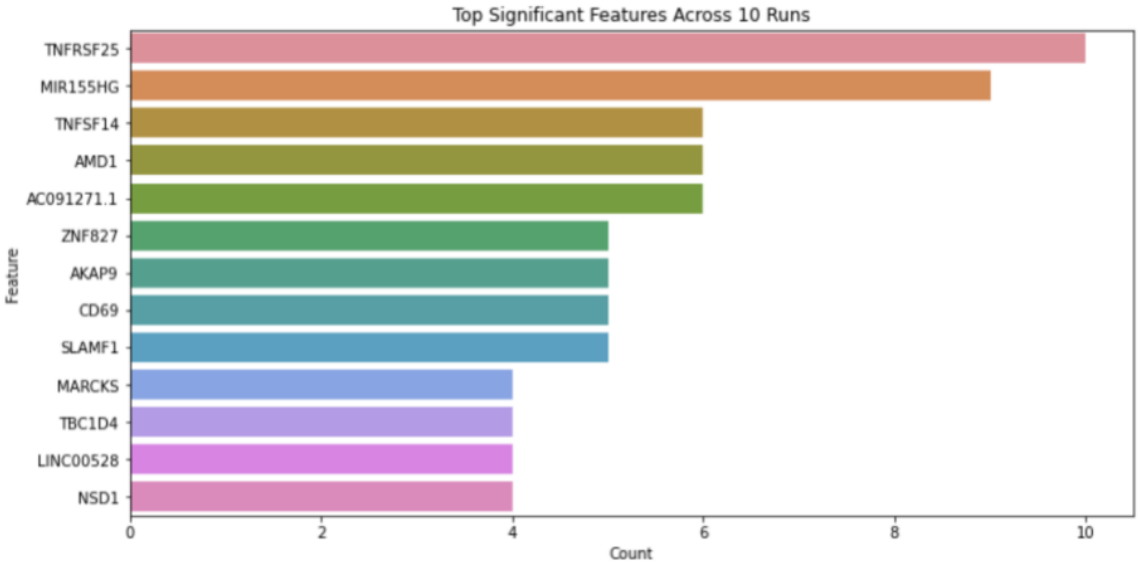
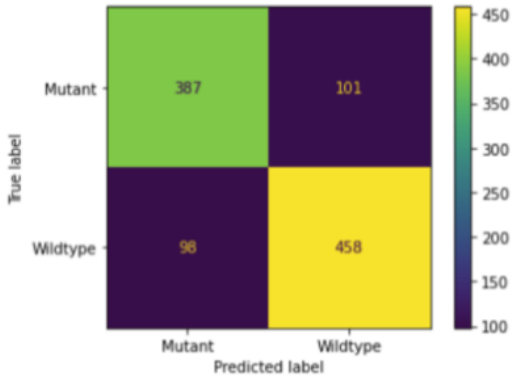


Figure 4.2: Final Significant Features - TET2 Dataset

4.2 DNMT3A Dataset

In the DNMT3A dataset, we conducted a similar series of runs with varying feature counts. When using features with a count greater than 3, the Random Forest classifier accurately classified CHIP cells 82% of the time. Subsequently, in runs with features exceeding a count of 4, the accuracy increased to 83%. Further attempts with features having a count greater than 5 resulted in a maintained accuracy of 81%. These outcomes suggest that the accuracy differences between feature counts of 3, 4, and 5 are negligible, indicating that features with a count greater than 5 can be utilized to inspect key features effectively. Our pipeline streamlined this dataset from its initial 2000 columns to a more concentrated selection of 3 essential columns. Figure 4.3 below illustrates the confusion matrix for the Random Forest model against the testing data, as well as the accuracy, precision, recall, and F1-scores. Considering the accuracy and recognizing the disparity in sample size compared to the TET2 dataset, the identification of these key features highlights the potential benefits of incorporating additional data to further refine the subsetting process for CD14 Mono cells. A visual representation of crucial features is presented in Figure 4.4.



	precision	recall	f1-score	support
Mutant	0.80	0.79	0.80	488
Wildtype	0.82	0.82	0.82	556
accuracy			0.81	1044
macro avg	0.81	0.81	0.81	1044
weighted avg	0.81	0.81	0.81	1044

Figure 4.3: Final Confusion Matrix - DNMT3A Dataset

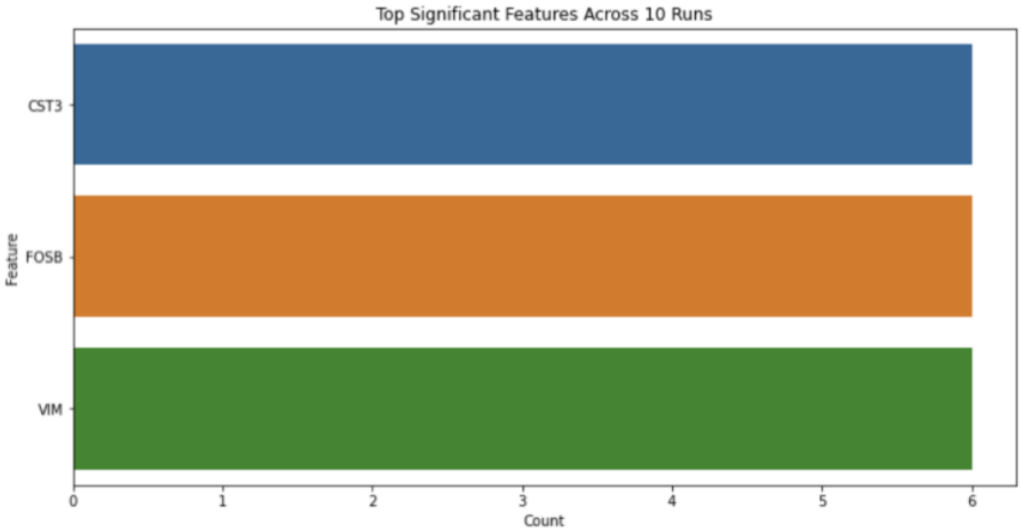


Figure 4.4: Final Significant Features - DNMT3A Dataset

CHAPTER 5

CONCLUDING REMARKS

Through the analytical pipeline we built, we successfully reduced both scRNA-seq datasets from their many initial columns to just several key columns, leading to reduced noise, computation efficiency, dimensionality reduction, and easier data interpretation. While the term ”redundancy” may not be entirely fitting, the dataset contains substantial overlap and interrelated information among its features. This discovery underscores the importance of employing careful feature selection and dimensionality reduction techniques, as it indicates the potential for optimizing the dataset to streamline and enhance the efficiency of our analysis.

As our analysis unfolded, we noted a fascinating alignment between some of the genes that surfaced prominently in this specialized CD14 Mono subset and the findings from earlier differential expression analyses within our laboratory. This alignment strengthens the significance of these genes in characterizing CHIP cells and their distinctive RNA expression signatures.

The presented study aims to harness the power of machine learning classifiers on scRNA-seq expression data to advance our comprehension of CHIP. This research endeavor not only enhances our understanding of CHIP cells but also pinpoints genes specific to CHIP, potentially holding diagnostic and therapeutic significance. The methodology proposed here is not confined to this specific dataset; it can be seamlessly applied to a broader spectrum of single-cell RNA expression data analysis, making it a valuable and versatile contribution to the realm of genomics research.

5.1 Challenges Faced

Aim 1: Building a More Cost-Effective Solution - Expanding upon the findings, it is imperative to consider the intrinsic challenges posed by scRNA-seq data. This type of data can be incredibly sparse, often comprising tens of thousands of columns. While R has the capability to handle such data efficiently, pandas, does not share this advantage. The process of loading data with tens of thousands of columns and thousands of rows into a pandas dataframe can be exceptionally time-consuming, sometimes taking several days. To circumvent this, we implemented a column-wise data reduction strategy, allowing us to work with a more manageable dataset.

While acknowledging the fact that R has a tidymodels package for machine learning, our decision to opt for Python with pandas dataframes stems from the extensive support and rich ecosystem of libraries available. Python, coupled with pandas, provides a robust framework that seamlessly integrates with an array of libraries, offering a comprehensive tool set for machine learning tasks. The versatility and community

support surrounding Python, including popular libraries like scikit-learn and TensorFlow, played a pivotal role in our choice.

Despite the limitations imposed by data reduction, it is important to recognize the potential for cost-effectiveness in the long run. A key element of this approach could involve training a classifier rigorously and then saving it as a pickle file (with a .pkl extension). This saved classifier could be easily loaded and utilized for the identification of new rows of information in future scenarios. While current methodologies relying on MAESTER are costly and time-consuming, our objective is to integrate a robust and efficient classifier into our pipeline. This step serves as a critical advancement toward optimizing cost-efficiency, minimizing expenditures, and accelerating CHIP cell identification. As we continue to refine this approach, we aim to transition to a more streamlined and practical method in the future.

Aim 2: Determining the Most Specific Gene/Expression for CHIP - Through our CHIP Identification pipeline, we aimed to identify the genes that exhibit the highest specificity for CHIP, recognizing their pivotal role in achieving precision in the diagnosis and classification of this condition. Exploration with the TET2 dataset successfully pinpointed 13 key genes that play a pivotal role in the identification of CHIP cells vs. non-CHIP cells using a Random Forest model that accurately classified CHIP cells 91% of the time.

These key features predominantly encompass genetic SNPs, highlighting their significance in the classification of CHIP cells. The genes and expressions associated with these SNPs represent insights that researchers and clinicians can delve into to gain a more profound understanding of CHIP and its underlying genetic components.

Aim 3: Uncovering Distinct RNA Expression Signatures in CHIP Cells - To enhance the specificity of our analysis, we narrowed our focus to CD14 Mono cells within the DNMT3A dataset, a cell subtype of particular interest in the context of CHIP. This strategic narrowing down of our focus allowed us to delve deeper into the unique RNA expression patterns characterizing CHIP cells. Exploration with the DNMT3A dataset successfully pinpointed 3 key features that play a pivotal role in the identification of CHIP cells vs. non-CHIP cells using a Random Forest model that accurately classified CHIP cells 81% of the time.

This precise subset came with its own set of intricacies. Upon reducing our dataset exclusively to CD14 Mono cells, a distinct data balance issue came to the forefront. The resulting dataset was characterized by a significant abundance of Mutant cells and a notable scarcity of Wildtype cells. While this configuration provided valuable insights into the RNA expression profiles of CHIP cells, it raised a fundamental challenge regarding data balance.

To address this concern in future analyses, we should consider the implementation of two viable techniques. One approach involves the fabrication of data using techniques like bootstrapping or Synthetic Minority Over-sampling Technique (SMOTE) (15), allowing the introduction of synthetically balanced data

points. The alternative method revolves around data duplication, ensuring a relative equilibrium between Wildtype and Mutant cell counts. The overarching goal is to maintain a balanced dataset, thereby enabling the continued use of accuracy as a reliable metric for performance assessment. However, it is noteworthy that other evaluation metrics such as ROC and precision-recall curves can be considered if they align better with the specific analytical goals.

CHAPTER 6

FUTURE WORK

Our current study serves as a robust stepping stone for future research endeavors. As we look ahead, several avenues for refining our analysis and extending its scope come to the forefront. Here are some key aspects to consider for future improvements and research directions:

- **Diverse Classifiers:** One potential avenue for improvement involves the incorporation of different classifiers from the sklearn library. By diversifying the set of classifiers used for training, we can assess how different models perform and continue to select the most suitable one for the task.
- **Ensemble Learning:** The use of an ensemble learning classifier, such as a "Vote Classifier" that combines the predictions of the top-performing classifiers, may improve accuracy. However, it is important to note that this approach can reduce the classifier's ability to generalize to data it has not encountered outside the training dataset.
- **Exploration of Mislabelled Data:** An investigation into mislabelled data within each dataset could uncover potential patterns and insights.
- **Addressing Data Imbalance:** In scenarios of data imbalance, methods such as "RandomUnderSampler" play a vital role in equalizing datasets, particularly when one class has a notable majority over the other. This action enhances model performance and mitigates potential biases. In future analyses where data scarcity is an issue, synthetic data generation through bootstrapping or data duplication could be considered to ensure an even distribution of Wildtype and Mutant cell counts. These strategies lead to a balanced dataset, allowing continued utilization of accuracy as a performance metric.
- **Alternative Performance Metrics:** If the data exhibits severe imbalance, it may be necessary to transition from using accuracy as the primary performance metric. Instead, metrics like ROC curves or precision-recall curves can provide a more informative evaluation of model performance.
- **Cross-Validation:** Increasing the number of folds in cross-validation (e.g., 10-fold) is a computationally-intensive option but is typically not necessary unless working with exceptionally large datasets. For most standard-sized datasets, a 5-fold cross-validation strategy is sufficient.
- **Dimensionality Reduction Techniques:** While dimensionality reduction techniques like Recursive Feature Elimination with Cross-Validation (RFECV) and Principal Component Analysis (PCA) can be

useful, they are not essential. Instead, we can opt for model-agnostic approaches to feature selection and pruning, such as LIME or SHAP.

- **Model-Agnostic Approaches:** Besides permutation importance, exploring other model-agnostic methods like Local Interpretable Model-Agnostic Explanations (LIME) (16) and SHapley Additive exPlanations (SHAP) (17) can offer a more comprehensive understanding of feature importance.
- **New Feature Creation:** Beyond "OneHotEncoding" categorical data, the creation of new features should only be considered if they provide substantial and meaningful information. It is important to avoid introducing noise into the dataset through unnecessary feature generation.
- **New Categorical Columns:** The inclusion of new categorical columns should be approached with caution. Additional categorical variables may not contribute significantly to the analysis and can potentially introduce noise.
- **Null Value Handling:** In scenarios where null values are present, the use of a "SimpleImputer" within the pipeline can be considered as an alternative to removing null-valued rows beforehand. This approach allows us to retain valuable data while addressing missing values effectively.
- **Incorporating New Data:** If additional data becomes available, it could significantly enrich the analysis and contribute to the overall effectiveness of the CHIP Identification system.

These considerations and potential enhancements serve as a road map for future iterations of our CHIP identification pipeline, offering opportunities to refine and expand our approach for improved results.

REFERENCES

[1] "Types of machine learning algorithms you should know." https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861.

[2] S. Jaiswal, P. Fontanillas, J. Flannick, A. Manning, P. V. Grauman, and B. G. e. a. Mar, "Clonal hematopoiesis and risk of atherosclerotic cardiovascular disease," *New England Journal of Medicine*, vol. 377, no. 2, pp. 111–121, 2017.

[3] A. G. Bick, J. S. Weinstock, S. K. Nandakumar, C. P. Fulco, E. L. Bao, and S. M. Z. et al., "Inherited causes of clonal haematopoiesis in 97,691 whole genomes," *Nature*, vol. 586, no. 7831, pp. 763–768, 2020.

[4] V. Svensson, R. Vento-Tormo, and S. A. Teichmann, "Exponential scaling of single-cell rna-seq in the past decade," *Nature Protocols*, vol. 13, no. 4, pp. 599–604, 2019.

[5] G. Genovese, A. K. Kähler, R. E. Handsaker, J. Lindberg, S. A. Rose, and S. F. e. a. Bakhoum, "Clonal hematopoiesis and blood-cancer risk inferred from blood dna sequence," *New England Journal of Medicine*, vol. 371, no. 26, pp. 2477–2487, 2014.

[6] T. E. Miller, C. A. Lareau, J. A. Verga, E. A. K. DePasquale, V. Liu, and D. S. et al., "Mitochondrial variant enrichment from high-throughput single-cell rna sequencing resolves clonal populations," *Nature Biotechnology*, vol. 40, pp. 1030–1034, July 2022.

[7] P. V. Kharchenko, "The triumphs and limitations of computational methods for scrna-seq," *Nature Methods*, vol. 18, pp. 723–732, July 2021.

[8] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[9] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[10] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press, June 2019.

[11] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[12] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[13] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, Aug 2008.

[14] A. Lun, "Overcoming systematic errors caused by log-transformation of normalized single-cell rna sequencing data," *BioRxiv*, p. 404962, 2018.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority oversampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[17] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, 2017.