# Feature Selection in High Dimensional RNA Sequencing Data

**Soren Dunn**[1] **Navya Gupta**[1] **Machi Takeda**[1] **Jeff Xue**[1]
[1]University of Illinois at Urbana-Champaign
sorend2@illinois.edu navyag3@illinois.edu
mtakeda2@illinois.edu jxue22@illinois.edu

## Abstract

Precise classification of cancer types is a central problem for cancer therapy and diagnosis. In recent years there has been an increasing interest in high-accuracy tumor classification based on RNA sequencing data. In this project we follow the methodology of Mohammed et al. [2021] in classifying tumor types from among the most diagnosed cancer types among women (breast, lung, colorectal, thyroid, and ovarian) but test a greater diversity of gene selection methods. Our highest accuracy methods yield better-performing test set accuracy on held out data than the original paper. Thus our proposed methods can aid in the detection and diagnosis of cancer in women and consequently aid in early treatment to improve survival

## 1 Introduction and Related Work

Although recent years have shown declining mortality for various cancer types, there remain more than 1.9 million yearly incidents of cancer with more than 500 thousand projected cancer deaths in the United States alone in 2023 Siegel et al. [2023]. Recent trends have shown greater decreases in cancer incident rates in men rather than women, highlighting the importance of cancer type classification in women in particular. Moreover, classification of cancer based solely on morphological characteristics has been shown to have several severe limitations including bias by experts Golub et al. [1999].

To help combat these limitations, recent years have shown an increasing use of RNA sequencing gene expression data in cancer type classification. The use of gene expression data has also been shown to lead to higher accuracy classification of cancer types García-Díaz et al. [2019]. The limitation of using gene expression data for cancer type classification mainly stems from its high dimensionality compared to available sample sizes. By far the two largest cancer databases which provide gene sequencing data (the publicly available Cancer Genome Atlas Project and the paid NCI Genomic Data Commons) only consist of 11,429 and 9,114 cases respectively across all cancer types Nassif et al. [2022]. For comparison, the database with the next highest case number (METABRIC) only consists of a mere 543 cases. In contrast, there are over 19 thousand coding genes in the human genome, resulting in most analysis of RNA sequencing data having a much higher number of features than available samples.

Two main types of feature selection methods have previously been applied for RNA sequencing data. The first of these methods are those that apply some sort of filter to score features according to some measure of divergence or correlations and then select features by setting a threshold on these values. These include methods based on correlation, chi-squared tests, and mutual information Guo et al. [2020]. Though computationally efficient, these methods often overlook more complex interactions between the genes which may be useful for prediction.

The second main type of feature selection methods involve some sort of model training and later selection for important predictors in the model. These include regularized logistic regression, LASSO and ridge regression, and random forests. In this project we compare the performance of both filter and training based feature selection methods.

In addition to employing a variety of feature selection methods, previous papers have also used a variety of trained models for tumor-type classification. Commonly used methods include multiplayer perceptrons, logistic regression, convolutional neural networks, support vector machines, and random forests Nassif et al. [2022]. Additionally, classification methods mainly come in two types: multiclass classification and tumor type classification and binary cancer diagnosis. The second of these usually results in higher accuracy: likely due to the highly imbalanced nature of these datasets and binary classification being an easier prediction problem than multiclass classification. Previous papers have used either imaging or gene expression data for their predictions. Imaging data has yielded high accuracy for binary classification, but gene expression data has yielded higher accuracy for tumor type classification Nassif et al. [2022] . One downside of previously published research is that most papers target different prediction problems and employ varying datasets and feature selection methods. This makes comparisons between previous papers difficult.

## 2 Methodology

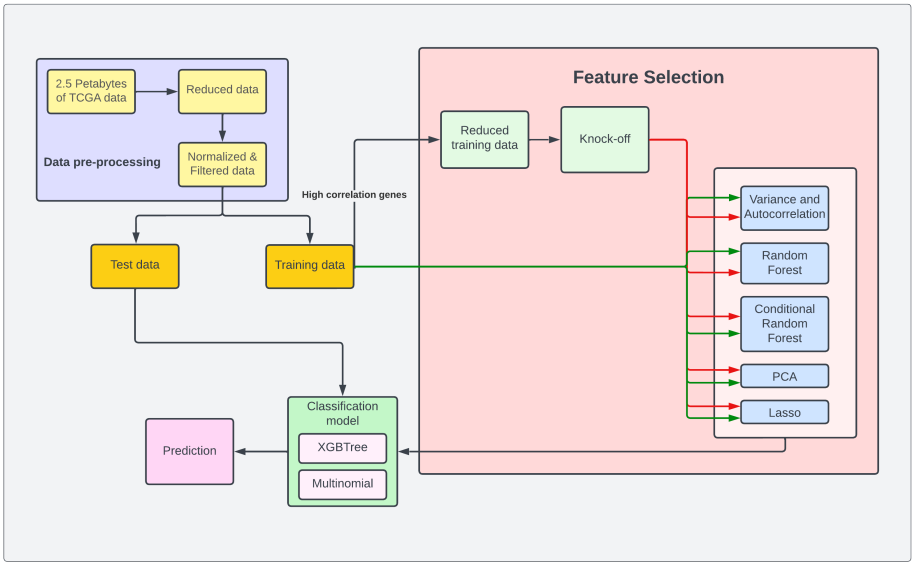

Figure 1: Outline of methodology. Data preprocessing steps are discussion in Section 2.2, feature selection methods in Section 2.3, and classification models in Section 2.5. Feature selection methods are employed both with and without controlling false discovery rate using knockoffs (see Section 2.4.3)

### 2.1 Datasets

We select five of the most commonly diagnosed cancer types among women (breast, lung, colorectal, thyroid, and ovarian) which together constitute 62 percent of all newly diagnosed cancer cases among women in 2023 and around 50 percent of all deaths Nassif et al. [2022]. We download a selection of gene expression quantification data from the The Cancer Genome Atlas Project (TCGA) which correspond to primary tumors among women from the five projects corresponding to our cancer types of interest (BRCA, LUAD, COAD, THCA, and OV respectively). The initially downloaded data contains unnormalized expression quantification values for 60660 genes spanning 2406 patient samples.

## 2.2 Data Pre-Processing

The majority of the pre-processing discussed in this section follows the methodology of Mohammed et al. [2021] in order to allow comparison with their results. The main differences in the methodology stem from the fact that we do not apply the last of their pre-processing steps (differential sequencing analysis) due to time constraints. However this step only eliminates 2,250 features in their analysis and if anything reduces the performance of our feature selection methods and final performance results due to having a higher feature dimension before applying our feature selection methods.

The initial gene pool is first subsetted to only RNA sequence values corresponding to coding genes. This reduction corresponds to the intuition that the RNA expression values which are most likely to contribute to cancer are the ones corresponding to coding genes since coding genes are the ones which actually produce proteins.

Next, we performed gene normalization using the TCGAanalyze_Normalization function which performs within lane, between lane, and count normalization of the RNA sequencing data to adjust for gene level effects and distributional difference between the different lanes (which correspond to the physical locations the RNA sequencing data was read). After normalization, we filtered the genes to those with mean intensity across the samples higher than 0.25 . This allowed us to eliminate genes with low intensities which are consequently likely to be irrelevant for the cancer type classification Mohammed et al. [2021].

The final size of our dataset (14,648 genes and 2406 patient samples still differs slightly from Mohammed et al. [2021] pre-differential sequence analysis dataset (14,899 genes from 2166 patient samples), but this is likely only due to differences in the TCGA database since 2021. However since the number of genes and patients only differs slightly from the original paper, performance on the two datasets should still be largely comparable.

Out of our final dataset we set aside 722 samples (which corresponds to approximately 30% of our total data) for our testing set and retain 1684 samples for our training set. Our final training set contains 766 breast cancer cases, 301 ovarian cancer cases, 263 thyroid cancer cases, 201 lung cancer cases, and 153 colorectal cancer cases and our testing set contains a similar distribution of cancer types. The fact that the number of samples across the varying cancer types are not too imbalanced means that accuracy is a reasonable performance metric for this dataset. The testing set is set aside in order to prevent overfitting of our models to the data since we are testing a high number of different feature selection methods (this train-test mirrors the procedure used by Mohammed et al. [2021]). We apply all feature selection methods on our training set and train two different classification methods on the subsets selected with these different methods. We take the gene subsets selected from the five feature selection methods with highest performance on our training data as the subsets for training classification models on the testing dataset for final evaluation of subset performance.

## 2.3 Feature Selection

Feature Selection addresses the challenge posed by high dimensionality. Its primary goal is to meticulously select pertinent features while discarding irrelevant, redundant, and noisy ones. This process aims to derive an optimal subset from the original features, striving for superior performance without altering or transforming the data.

After preprocessing, the RNASeq gene expression data contained 14,648 dimensions or features, which remained substantial relative to the sample size of 1683 in the training set, therefore we required feature selection algorithms to decrease the number of genes or features which would further allow us to analyze the data better.

In our report we have dealt with several feature selection including principal component analysis, Lasso regression, random forest and conditional random forest. We have compared the accuracy of these feature selection methods when applied with our classification models with and without the integration of the knockoff filter.

### 2.3.1 PCA

Principal Component Analysis is a statistical procedure that uses an orthogonal transformation to convert a set of possibly correlated variables into a set of linearly uncorrelated variables called

Principal components. In our gene samples, PCA converts genes into several PCs, and the importance of accumulated Principal Component variances is described as follows:

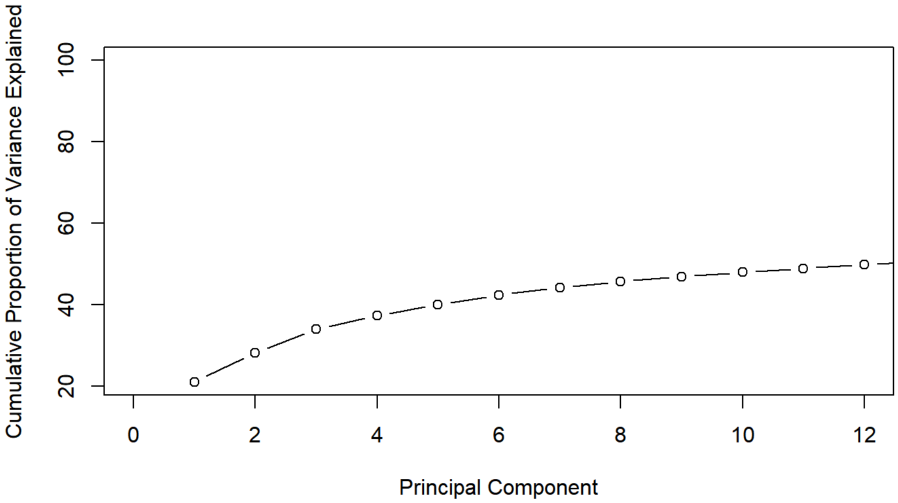

Figure 2: PCA Cumulative Variances

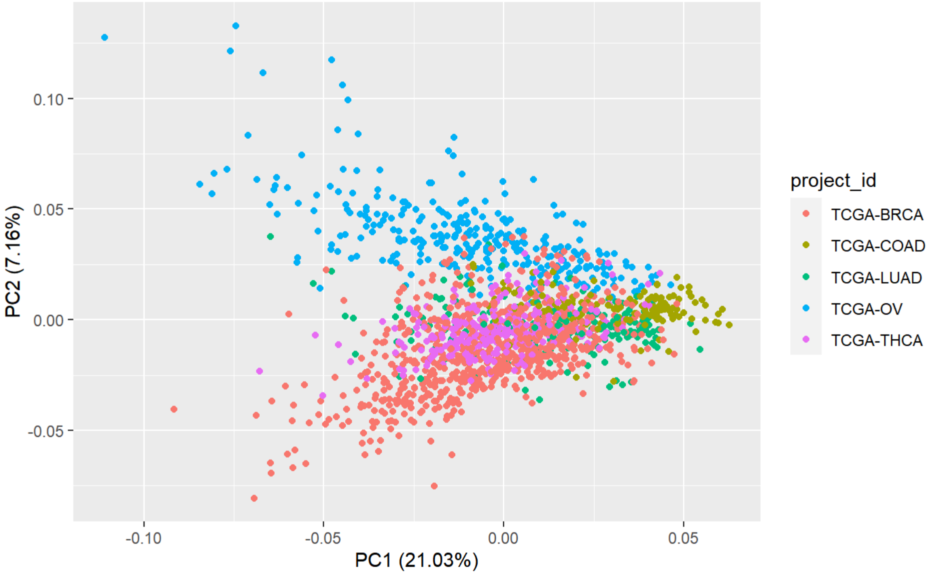

Figure 3: Plot of samples against the first two principles components. Points colored by project (which correspond to the different cancer types). Points show marginal separation based on cancer type

We chose to select the top 12 principal components produced, and it explain around 54 percent of gene sample variance. The top 12 prediction gives perfect prediction accuracy on the training data.

The advantage of PCA is that it is an effective dimension-reduction tool, which limits 14648 gene variables to 12 PC variables effectively. By choosing the top two PCs, we can generalize graphs to visualize the classifications among samples.

The disadvantage of PCA is it limits the interpretation of our gene variables. It would convert the gene variables to the top PCs, and We do not know which specific genes are being selected by Principal Components.

### 2.3.2 Lasso

The RNAseq after Data pre-processing and filtering had 14648 variables of featured genes, which was still huge given that we have 1684 observations in our sample. Therefore, LASSO regression was used to select the number of genes that allow us to effectively analyze the data. LASSO performs both variable selection and regularization to enhance the prediction accuracy. It does this by imposing a constraint on the absolute values of the model parameters, effectively driving some of them to zero. The variables corresponding to the zeroed parameters are excluded from the model.

In the case of Multinomial Response, the log-likelihood penalize function used to shrink coefficients as follows:

$$l(\{\beta_{0k}, \beta_k\}_{k=1}^{K}) = -\left[\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K} y_{il}(\beta_0 + x_i^T\beta_k) - \log\left(\sum_{k=1}^{K} e^{\beta_0 + x_i^T\beta_k}\right)\right)\right] + \lambda\left[\sum_{j=1}^{p} ||\beta_j||_1\right] \tag{1}$$

Where K is a level greater than 2, the last term is the lasso penalty. In our case, LASSO was implemented using R glmnet package. We selected the Lambda value by minimizing the Multinomial Deviance:

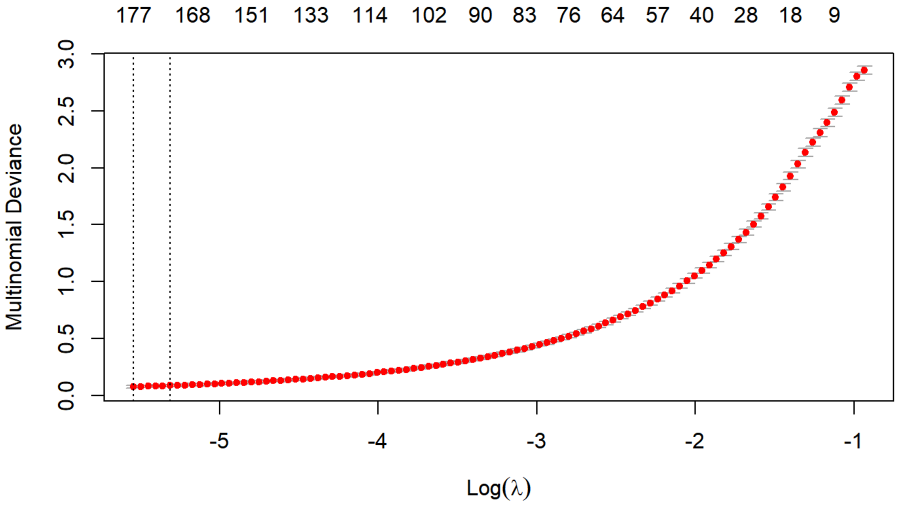

Figure 4: Lasso Deviance Graph

The optimal value we had for $\log(\lambda)$ is -5.542027. As a result, we have selected 177 genes among 14648 genes in our sample. The model accuracy on the training set is 1.

The advantage of using LASSO is it has a very efficient algorithm to compute the result and define the important gene variables without consuming too much computational power. Another advantage of using Lasso is that it avoids overfitting the training data, and provides a more accurate and generalized prediction of the testing data.

The disadvantage of LASSO is it would automatically eliminate highly correlated variables during variable selection, which can make it hard to understand correlations among genes.

### 2.4 Baseline Feature Selection Methods

Several simple baseline feature selection methods were applied based on previous work. One would expect genes with high correlations with the outcomes to potentially serve as important features for prediction. Due to this intuition, we first one-hot encoded the tumor types and then averaged the correlations between the tumor types and these outcomes and picked the top 20 genes with highest average correlations with these outcomes. A second baseline feature selection method employed was to take the top 20 genes with most variance. This basline helped us make sure our methods were performing more interesting variable selection than simply taking the genes with the most variation accross the samples and discarding gene without variation across samples. Though unlikely to lead to the best predictive performance, these baselines served as computationally efficient baselines to compare the more computationally-intense but likely higher-performing feature selection methods to.

#### 2.4.1 Random Forest

Random forests are frequently employed in data science workflows for feature selection due to their tree-based techniques, which inherently rank features based on their ability to enhance node purity. This ranking involves assigning a status to each feature, indicating whether they reduce impurity across all trees (referred to as GINI impurity). The nodes that contribute the most substantial decreases in impurity are typically found at the beginning of the trees, while those with minimal impurity reduction are situated towards the end of the trees. Consequently, by pruning trees below a specific node, we can generate a subset comprising the most significant features.

**Feature Importance:**

Random Forest provides a measure of feature importance based on the decrease in impurity or information gain when splitting a node using a particular feature. This importance score is calculated during the training process and can be used for feature selection.

**Gini Impurity:** The Gini impurity measures the probability of incorrectly classifying a randomly chosen element if it were randomly labeled. For a node $m$ with $K$ classes, the Gini impurity is calculated as:

$$I_G(m) = 1 - \sum_{k=1}^{K} p_{mk}^2$$

where $p_{mk}$ is the proportion of class $k$ observations in node $m$. Once we found the features with the most importance we fed them through a recursive feature elimination algorithm to remove features, based on Darst et al. [2018] that could potentially have a higher importance due to high correlation, we chose the bottom 10% of the features with the least importance are removed. The choice of 10% was due to computational constraints. Variables removed in each step are assigned ranks based on their sequential elimination order and the respective importance scores at that point. The comparison of importance scores is conducted within individual runs and not across runs. e process iterates by utilizing the reduced dataset from the preceding step, repeating the elimination procedure until we reach the required number of features, we achieved this with the help of the *rfe* package of R.

**Advantages of Random Forests:**

1. **Reduced Overfitting Risk:** While decision trees tend to overfit by tightly fitting all samples in the training data, a robust number of trees in a random forest mitigates this issue. Averaging uncorrelated trees decreases overall variance and prediction error, preventing overfitting of the classifier.

2. **Simplified Evaluation of Feature Importance:** Random forests offer an uncomplicated means of assessing variable importance. Methods such as Gini importance, mean decrease in impurity (MDI), and permutation importance (MDA) measure a variable's contribution to the model. These metrics analyze the impact of excluding a variable on the model's accuracy, facilitating the understanding of feature significance.

**Limitations of Random Forests:**

1. **Time-Consuming Processing:** Although capable of handling large datasets for more accurate predictions, random forest algorithms can be time-consuming due to the computation required for each decision tree.

2. **Increased Resource Requirements:** Handling larger datasets in random forests demands more resources for data storage and computation, presenting challenges in implementation.

### 2.4.2 Conditional Random Forest

Another feature selection method we have used is a non-parametric method called the conditional random forest (CRF) developed by Torsten Hothorn and Zeileis [2006]. Similar to the popular random forest algorithms, CRF is an ensemble of decision trees. The decision trees used in CRF are called the conditional inference trees (CIT).

**CRF Algorithm:**

While the popular random forest algorithms, such as the one that uses the Classification and Regression Trees (CART), can only be constructed using a bootstrap sample, either a bootstrap sample or a random sub-sample of size up to 2/3 of the whole training set can be used for the CIT. In a CIT, the decisions are made by testing a null hypothesis derived from the conditional distribution of the statistics that capture the association between the response variable and the predictor variables. More specifically, let $H_0^j : Y$ is independent of $X_j$ for $j = 1, ..., p$, where $p$ is the number of predictors in the design matrix. The global null hypothesis for the CIT is that this holds for all the predictors in the sample set, so the global null hypothesis can be written as $H_0 = \cap_{j=1}^m H_0^j$, where $m$ is the number of predictors in the sample set.

Let $\sigma_t$ be a permutation of the response $Y$, and $S = \{\sigma_t\}$ be the set of all permutations. Let $K$ denote the number of levels in the response $Y$, and define $B_k = \{i | y_i = k, i = 1, ..., n\}$ for $k = 1, ..., K$. Since our study is concerned with data where predictors are the gene expressions, we will only consider the case where the predictors are continuous variables. We can then define a set of vectors

$$R_j = \left( \sum_{i \in B_1} x_{ij}, \sum_{i \in B_2} x_{ij}, ..., \sum_{i \in B_K} x_{ij} \right)^T \tag{2}$$

which measure the association between the response $Y$ and the predictor $X_j$.

Let $\mu_j = \mathbb{E}(R_j | S)$ and $\Sigma_j = \text{cov}(R_j | S)$ be the conditional expectation and the conditional covariance of $R_j$, respectively, given all possible permutations of $Y$. Define a statistic $U_j = (R_j - \mu_j)^T \Sigma_j^{-1} (R_j - \mu_j)$. Here, $\Sigma_j^{-1}$ is the inverse or the Moore-Penrose general inverse of $\Sigma_j$ (Torsten Hothorn and Zeileis [2006]).

Using the above definitions, we can now illustrate the algorithm for constructing the CRF with $B$ CITs:

```
1. Construct the CITs $T_b$ for $b = 1, ..., B$ by repeating the following steps $B$ times:
   a. Draw a sample $X_b^*$ by either bootstrap sampling or by random sub-sampling up to 2/3 of the of the training set
   b. Grow a CIT $T_b$ by repeating the following steps until the global null hypothesis $H_0$ is accepted:
      i. Randomly select $m$ predictors from the $p$ predictor variables
      ii. Compute the vectors $R_j$ for the selected predictors
      iii. Permute $Y$ and compute $\mu_j$, $\Sigma_j$, and $U_j$ for all possible permutations
      iv. Using the statistic $U_j$, compute the p-value $p_j$ for the test with $H_0^j$
      v. Compare the p-value of the global test, $p = \min\{p_j\}$, and a pre-selected $\alpha$-level for the test
         - If $p \geq \alpha$, stop growing the tree
         - If $p < \alpha$, select the predictor that has the minimum p-value $p_j$ to continue splitting
2. Output the collection of trees $\{T_b\}_{b=1}^B$
```

We used the R function `cforest` in the `moreparty` package to build the CRF.

**Feature Importance:**

To select the most important features, we used the `fastvarImp` function in the `moreparty` package, which calculates the feature importance measures for conditional random forests. The `fastvarImp` function computes the feature importance based on the decrease in node impurity (e.g., Gini index or variance) caused by splitting on each predictor variable, similar to the feature importance calculation explained in section 2.4.2.

**Advantages of Conditional Random Variable:**

In addition to the advantages mentioned for the random forests, the CRFs have the following advantages:

1. **Nonlinear Relationships:** The CRFs perform well when there are nonlinear relationships (higher order terms and interactions) between the response and the predictor variables, because CRFs consider interactions between features when making splits in the trees.
2. **Multicollinearity:** The CRFs handle multicollinearity more effectively than other methods since the CRFs consider the relationships between the predictor variables and the selected variables based on their conditional association with the response variable.

**Limitations of Conditional Random Forests:**

In addition to the limitations mentioned for the random forests, the CRFs have the following limitations:

1. **Assumption of Conditional Independence:** The CRF algorithm assumes that the predictor variables are conditionally independent given the response variable, so if this assumption is violated, the feature selection using CRF may be suboptimal.
2. **Performance for Simple and Additive Relationships:** While the CRFs perform well when there exists nonlinear relationship between the response and the predictor variables, it may not perform optimally when the relationships are simple (linear) and additive (no interactions), and other feature selection methods may be better choices.

### 2.4.3 Integration of knockoff filter

For most of the above variable selection methods, the exact number of features selected has been somewhat arbitrary. To apply a more principled approach to the selected number of features for each of the feature selection methods, we combined several of the previously applied methods with a knock-off filter Candes et al. [2017], which provides a general framework for controlling the false discovery rate when performing variable selection. We will present the general formulation of the variable selection problem used for the knock-off filter below followed by an explanation of the particular setup used in our application of knock-offs.

The goal in the knock-off filter variable selection problem is to identify important predictors $X_j$ for a response variable $Y$ from $p$ potential explanatory variables represented as $\mathbf{X} = (X_1, \dots, X_p)$. Given $n$ samples $(X_{i,1}, \dots, X_{i,p}, Y_i)_{i=1}^n$, we seek to determine which predictors significantly influence the response.

We assume the conditional independence of the responses $Y_i$ given their corresponding predictors $(X_{i,1}, \dots, X_{i,p})$, such that:

$$Y_i | (X_{i,1}, \dots, X_{i,p}) \sim F_{Y|X}, \quad i = 1, \dots, n,$$

where $F_{Y|X}$ is some conditional distribution.

Practically, $F_{Y|X}$ may only depend on a subset $S \subset \{1, \dots, p\}$ of predictors, rendering $Y$ conditionally independent of all other variables given $\{X_j\}_{j \in S}$.

A predictor is termed null if $Y$ is independent of $X_j$, conditionally on all other predictors $X_{-j} = \{X_1, \dots, X_p\} \setminus \{X_j\}$. The set of null variables is denoted by $H_0$. A *relevant* predictor is one for which $j \notin H_0$.

The aim is to identify as many relevant variables as possible while controlling the false discovery rate (FDR). For a selection rule that chooses a predictor subset $\hat{S}$, the FDR is defined as:

$$FDR = \mathbb{E}\left[\frac{|\hat{S} \cap H_0|}{\max(1, |\hat{S}|)}\right].$$

Two key paradigms are considered in knock-off based variable selection: Model-X and Fixed-X. Each has different assumptions regarding the explanatory variables $\mathbf{X}$.

In the Model-X paradigm, the explanatory variables are considered random and drawn i.i.d. from some known distribution $F_X$. The joint distribution of all variables is given by:

$$(\mathbf{X}_i, Y_i) i.i.d. \sim F_{XY} = F_{Y|X} \circ F_X, \quad i = 1, \ldots, n,$$

where the conditional distribution $F_{Y|X}$ can be arbitrary and unknown, and no restrictions are imposed on $n$ and $p$ (the problem can be high-dimensional).

For knockoff methods to be effective in this paradigm both the covariate distribution $F_X$ must be known and knockoff construction must ensure that swapping any subset $S \subseteq \{1, \ldots, p\}$ of the original and knockoff variables $(\mathbf{X}, \tilde{\mathbf{X}})$ does not change the distribution, and that $Y$ remains conditionally independent of $\tilde{\mathbf{X}}$ given $\mathbf{X}$.

The Model-X paradigm is well-suited for high-dimensional RNA gene selection because it allows for the direct modeling of variability in RNA expression as a random process. Since the distribution $F_X$ can be estimated from large-scale genomic databases, the knockoff method can create proper knockoffs that respect the complex correlation structure typical in such data. Additionally, because the paradigm does not require specification of $F_{Y|X}$, it can accommodate complex and potentially unknown relationships between genes and the response variable, which is common in genomic applications. Furthermore, the allowance for high-dimensional settings where $p \gg n$ is particularly relevant in our context, where the number of genes (predictors) is much larger than the number of samples.

For these reasons we applied Model-X knock-offs with second-order Gaussian knock-offs. These estimate the mean and covariance of the rows of X instead of using the true parameters from which the variables were sampled. This choice of knock-off construction was chosen as it was reasonably computationally efficient while still allowing high-fidelity knock-off construction.

With the knock-offs we selected various test statistics based on the previously-applied selection methods. These included the total decrease in node impurities from splitting on a variable versus its knockoff averaged across all trees, the difference in standardized coefficient estimates between the variable and its knockoff for LASSO, taking the difference in chi-squared statistic between the variable and its knockoff when converting each sample value to whether or not it has above median abundance for its gene,and the difference in averaged correlation with the one-hot encoded tumor type classifications. Since computing the second-order Gaussian knock-offs for the genes ended up being quite computationally expensive for a large number of genes, quite aggressive preselection of variables needed to be applied before variable selection with knockoffs. To do this, we first subsetted the genes to the 500 with highest variances in order to have an intitial filter to remove genes with low abundance variability which would be unlikely to be predictive of the tumor type.

Initially an FDR control of 0.1 was applied to all feature selection methods. However this resulted in the LASSO, correlation, and variance based selection methods selecting 0 genes. In order for these methods to still select genes while applied the knockoff filter, the FDR cutoff for these methods was raised to 0.5 which resulted in 30, 22, and 10 selected genes, respectively.

**Limitations of Knockoffs:**

1. **Knockoffs computationally expensive:** Calculation of knockoffs was computationally expensive so large amounts of feature selection were required to be applied even before application of knockoffs. This resulted in all knockoff based methods performing worse than their non-knockoff based counterparts.
2. **Choice of FDR Cuttoff:** Even though employing knockoffs allowed control of a sepecified FDR rate for variable selection, the specific choice of false discovery rate was still somewhat

arbitrary. Additionally the same FDR rate could not be applied across all the feature selection methods because some methods would select many more features with the same FDR control than others. Thus use of knockoffs for variable selection only seems worthwhile given a principled need for a specific FDR cuttoff rate.

### 2.5 Classification

Two different classification models were trained using the selected feature subsets. Each were implemented with 5-fold cross validation. Mohammed et al. [2021] used 10-fold cross validation, but we limited to 5-fold cross validation due to computational limitations.

The first classification method applied was gradient boosting with 100 trees and max tree depth of 5.

The classifier is defined by an ensemble of decision trees, where each tree is constructed to minimize a loss function given the current model. The final classification model, $F(x)$, is a sum of the weak learners (in this case depth-1 trees):

$$F(x) = \sum_{n=1}^{N} f_n(x), \quad where \quad f_n : \mathcal{X} \to \mathbb{R} \tag{3}$$

Where $f_n$ represents the $n$-th tree and $\mathcal{X}$ denotes the feature space.

This method was applied due to its wide applicability and previous competitive performance in cancer type classification with RNA expression data Nassif et al. [2022] while being less computationally expensive than deep-learning based approaches.

The second method applied was penalized multinomial logistic regression with L2 regularization. Given $N$ data points $\{x_i, y_i\}_{i=1}^{N}$ where $x_i$ is the feature vector of the $i$-th data point, and $y_i$ is the corresponding class label such that $y_i \in \{1, 2, ..., K\}$ for $K$ classes (which in our case correspond to the 5 cancer types), the penalized objective function $J(\theta)$ of a multinomial logistic regression model with L2 regularization is as follows:

$$J(\theta) = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{k=1}^{K} 1\{y_i = k\} \log \frac{e^{\theta_k^T x_i}}{\sum_{j=1}^{K} e^{\theta_j^T x_i}}\right] + \frac{\lambda}{2}\sum_{k=1}^{K}\sum_{j=1}^{d} \theta_{kj}^2 \tag{4}$$

Where:

- $\theta$ are the parameters to be learned for each class $k$ and feature $j$.

- $\theta_k$ is the parameter vector for class $k$.

- $\theta_{kj}$ is the parameter for feature $j$ in class $k$.

- $1\{\cdot\}$ is an indicator function that is 1 if $y_i = k$ and 0 otherwise.

- $\lambda$ is the regularization strength, which controls the size of the shrinkage applied to the parameters: higher values of $\lambda$ lead to more significant penalization.

- $d$ is the number of features in each feature vector $x_i$.

The first part of $J(\theta)$ is the negative log-likelihood (also known as the cross-entropy loss) of the multinomial logistic regression model. The second part is the L2 regularization term that penalizes the squared magnitude of the coefficients.

These models were trained on the features selected using each of the previously described feature selection methods on the training dataset. The best performing of these gene subsets were then chosen for evaluation on the testing dataset where these two models were again trained for each data subset. Results for these classification models are reported in Figures 6, 7, and 8.

Feature importances for the classification models were found by using the decrease in impurity for random forests and the normalized coefficients for the penalized logistic regression. The different classification models each had completely different genes among their top features. This is not surprising since, as shown in Figure 5, the subsets contained almost exclusively different subsets of genes in any case, but this further demonstrates that the few genes which were present in multiple subsets were not the ones predominantly behind predictive performance of the models.

## 3 Results

Among all of the subsets from different feature selection methods we considered, we found that the most selected features only occur in a single one of the subsets. This shows that different feature selection methods selected almost completely different subsets. The distribution of selected features among different subsets can be seen in Figure 5.

We found that the best-performing model was the gradient boosting trees model using top 50 features from the CRF with 300 trees, which had a 0.9988 testing accuracy. The next two best-performing models were the gradient boosting with the filtered LASSO having the accuracy of 0.9973, and then the gradient boosting with the top 50 features from random forest with 100 trees having the accuracy of 0.9959. Figure 6 shows the testing accuracy of all the models we developed. We can see that classification models with gradient boosting trees performed better than the models with the multinomial logistic regression for all the feature selection methods except for the PCA. The top three models we developed also performed better than the best model developed in the baseline paper, which can be seen in Figure 7. Figure 8 shows the results of testing on a subset of the test dataset. This allows us to see greater differences in model performances to increase our confidence in which feature selection methods performed best. We found that gradient boosting models with filtered LASSO and random forest still performed better than the other models. We were not able to test the models with CRF on the subset of test data, but it is likely that it will still perform well, based on the results of the gradient boosting models with filtered LASSO and random forest.

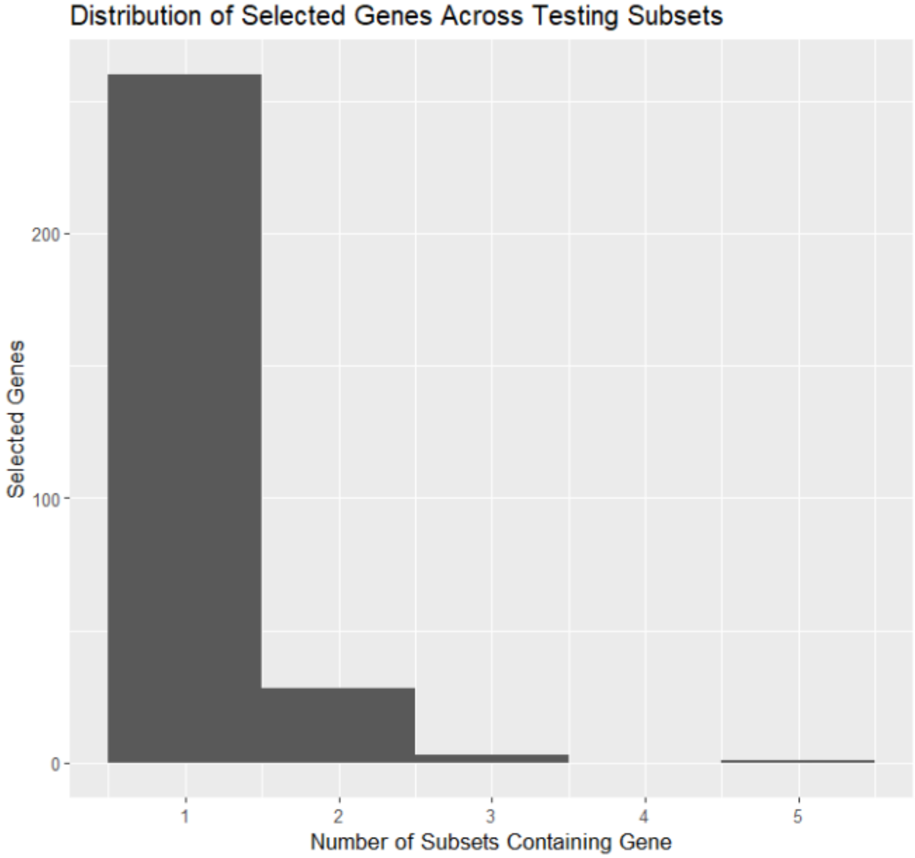

Figure 5: Number of subsets each gene selected in the feature subsets applied to the testing data was present in

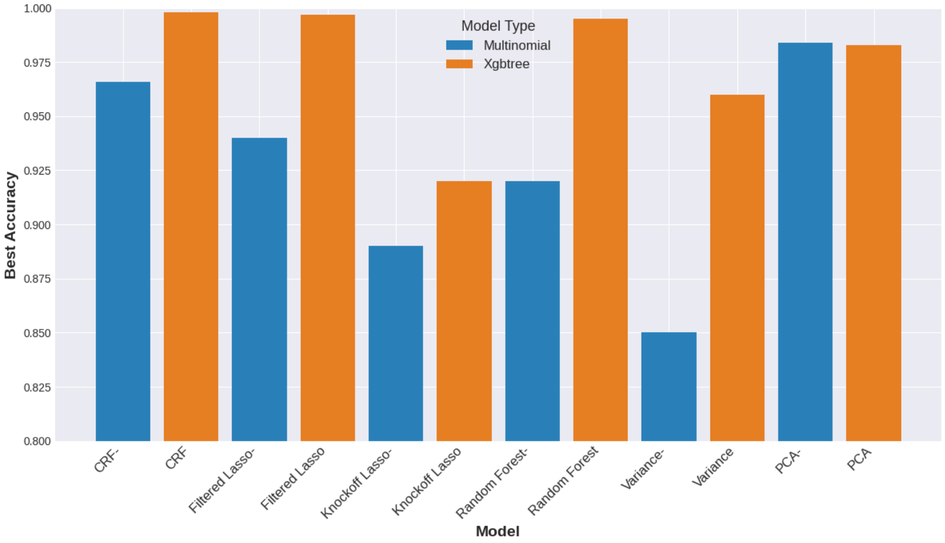

Figure 6: Accuracy across different feature selection methods (full test set)

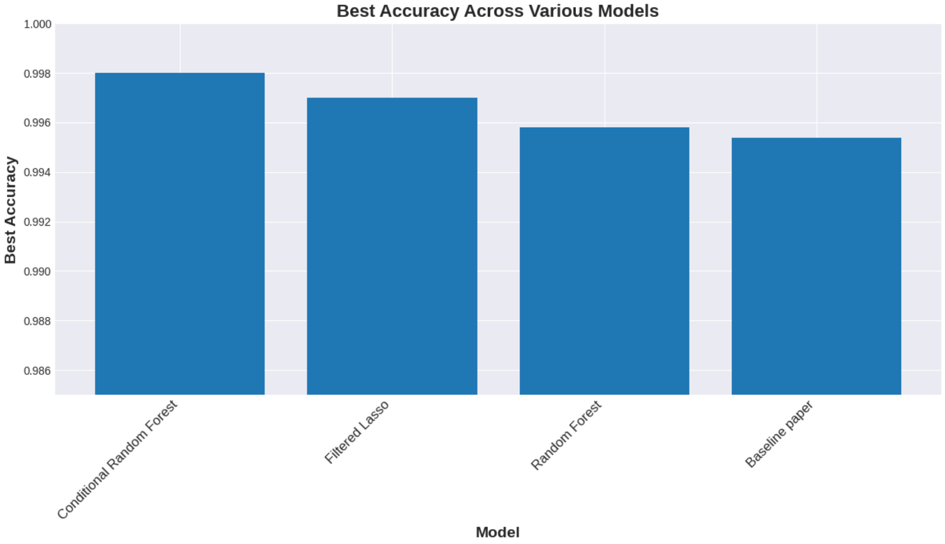

Figure 7: Accuracy comparison with baseline

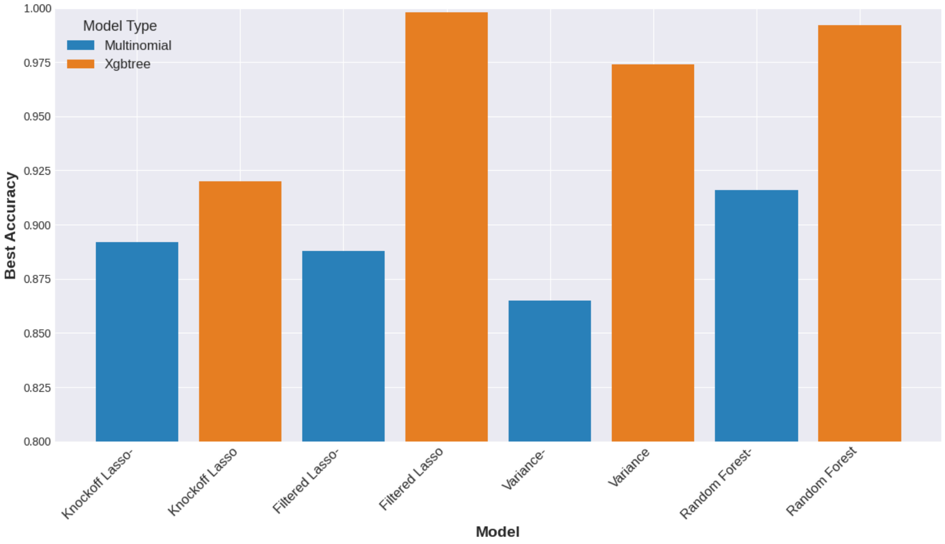

Figure 8: Accuracy across different feature selection methods (subset of test set)

## 4 Conclusion

In this study, we introduced various methods for selecting relevant genes associated with cancer development to effectively classify the five most prevalent cancers in women: breast cancer, colon adenocarcinoma, lung adenocarcinoma, ovarian cancer, and thyroid cancer, using the RNASeq gene expression dataset. We employed the feature selection techniques : Principal component analysis, lasso, random forest and conditional random forest along with the integration of a knockoff filter combined with the techniques to combat the arbitrary nature of the features selected in each of the them. Between the classification models tested and the feature selection methods, we found our most successful approach utilized a combination of conditional random forest and gradient boosting.

The results indicate that three of the proposed feature selection techniques (conditional random forest, filtered lasso, and random forest) combined with gradient boosting outperform the baseline model in Mohammed et al. [2021]. This suggests that our proposed models have a higher potential to significantly enhance the early detection and diagnosis of cancer in women, enabling timely intervention and ultimately improving survival.

To further this work, we would like to work on a highly specific classification model involving a two-layer neural network potentially trained via reinforcement learning and recording the results with conditional random forest as the feature selection technique. Future work should also integrate additional omics data, such as DNA sequencing, epigenetic data, or proteomic data, to provide a more comprehensive understanding of cancer biology.

## Contributions

All authors contributed substantially to this work. Soren, Navya, Machi and Jeff participated in the design of the study. Navya peformed feature selection with random forests and conducted inference on the resulting models along with the literature review for the baseline. Machi performed feature selection with conditional random forests and conducted inference on the resulting models. Jeff ran the LASSO and PCA feature selection methods and conducted inference on the resulting models. Soren acquired and preprocessed the data, performed the baseline and knockoff feature selection methods, and ran the classification models. All authors drafted and reviewed the drafts of this manuscript and that of the presentation done in class and approved the final version for submission.

## References

Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: Model-x knockoffs for high-dimensional controlled variable selection, 2017.

Burcu F. Darst, Kristen C. Malecki, and Corinne D. Engelman. Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC Genetics*, 19(Suppl 1):65, 2018. doi: 10.1186/s12863-018-0633-8. URL `https://doi.org/10.1186/s12863-018-0633-8`.

Pilar García-Díaz, Isabel Sánchez-Berriel, Juan A Martínez-Rojas, and Ana M Diez-Pascual. Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-Seq data. *Genomics*, 112(2):1916–1925, November 2019.

T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenbeek, J P Mesirov, H Coller, M L Loh, J R Downing, M A Caligiuri, C D Bloomfield, and E S Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999.

Juncheng Guo, Min Jin, Yuanyuan Chen, and Jianxiao Liu. An embedded gene selection method using knockoffs optimizing neural network. *BMC Bioinformatics*, 21(1):414, Sep 2020. ISSN 1471-2105. doi: 10.1186/s12859-020-03717-w. URL `https://doi.org/10.1186/s12859-020-03717-w`.

Mohanad Mohammed, Henry Mwambi, Innocent B. Mboya, Murtada K. Elbashir, and Bernard Omolo. A stacking ensemble deep learning approach to cancer type classification based on tcga data. *Scientific Reports*, 11(1):15626, Aug 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-95128-x. URL `https://doi.org/10.1038/s41598-021-95128-x`.

Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, Yaman Afadar, and Omar Elgendy. Breast cancer detection using artificial intelligence techniques: A systematic literature review. *Artif Intell Med*, 127:102276, March 2022.

Rebecca L. Siegel, Kimberly D. Miller, Nikita Sandeep Wagle, and Ahmedin Jemal. Cancer statistics, 2023. *CA: A Cancer Journal for Clinicians*, 73(1):17–48, 2023. doi: https://doi.org/10.3322/caac.21763. URL `https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21763`.

Kurt Hornik Torsten Hothorn and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006. doi: 10.1198/106186006X133933. URL `https://doi.org/10.1198/106186006X133933`.