Classification of Cancer RNA-Seq Data

ISYE 6740 Project Proposal

Kirpa Singh (GTID: 1445)

Problem Statement

RNA-Seq is a sequencing technique that is used to determine the quantity of RNA in a biological sample[1]. In other words, this technique allows us to investigate the transcriptome, which is the set of all RNA transcripts in a cell. The transcriptome contains vital information regarding when and where each gene is turned on in an organism's cells and tissues. Counting the number of transcripts in a cell tells us the amount of gene activity (also known as gene expression)[2]. Different cells show different patterns of gene expression. Thus, understanding transcripts is crucial to understanding the biology of a cell and differences in gene expression that may indicate disease.

Sequencing the cancer transcriptome with RNA sequencing provides us with information on gene expression in tumors. Cancer RNA-Seq can help with identifying gene expression and mutational profiles associated with tumor types. Monitoring gene expression and transcriptome changes can help in understanding tumor classification[3].

In this report, I will be using a RNA-Seq dataset that contains gene expression levels for patients that have different types of tumors. The tumors represented in the data include Breast invasive carcinoma (BRCA), Kidney renal clear cell carcinoma (KIRC), Colon adenocarcinoma (COAD), Lung adenocarcinoma (LUAD), and Prostate adenocarcinoma (PRAD). Based on the knowledge that different gene expression levels may be associated with different types of tumors, I will be using this data to perform classification of the different tumor types. I hypothesize that there will be some similarity in gene expression features for samples in the same tumor class, and these gene expression levels will likely differ from other classes.

Data Source

The gene expression cancer RNA-Seq dataset, which is part of the RNA-Seq (HiSeq) PANCAN data set, is a random extraction of gene expressions of patients having different types of tumor: BRCA, KIRC, COAD, LUAD and PRAD. This dataset contains 801 patient samples. Each sample has 20,531 attributes. These attributes are RNA-Seq gene expression levels measured by illumina HiSeq platform. This data can be found in the UCI Machine Learning Repository[4]. The repository contains two different files. The first file contains the bulk of the data, including the 801 samples and their 20,531 attributes. The second file is a labels file which gives the tumor class for each sample (BRCA, KIRC, COAD, LUAD, or PRAD). A small snapshot of the data can be seen in Figure 1. This figure just shows a small subset of the gene expression features, as there are too many to show in one image.

	gene_0	gene_1	gene_2	gene_3	gene_4	gene_5	gene_6	gene_7	gene_8	gene_9	...	gene_20522	gene_20523	gene_20524	gene_20525	gen
0	0.0	2.017209	3.265527	5.478487	10.431999	0.0	7.175175	0.591871	0.0	0.0	...	8.210257	9.723516	7.220030	9.119813	12
1	0.0	0.592732	1.588421	7.586157	9.623011	0.0	6.816049	0.000000	0.0	0.0	...	7.323865	9.740931	6.256586	8.381612	12
2	0.0	3.511759	4.327199	6.881787	9.870730	0.0	6.972130	0.452595	0.0	0.0	...	8.127123	10.908640	5.401607	9.911597	9
3	0.0	3.663618	4.507649	6.659068	10.196184	0.0	7.843375	0.434882	0.0	0.0	...	8.792959	10.141520	8.942805	9.601208	11
4	0.0	2.655741	2.821547	6.539454	9.738265	0.0	6.566967	0.360982	0.0	0.0	...	8.891425	10.373790	7.181162	9.846910	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
796	0.0	1.865642	2.718197	7.350099	10.006003	0.0	6.764792	0.496922	0.0	0.0	...	9.118313	10.004852	4.484415	9.614701	12
797	0.0	3.942955	4.453807	6.346597	10.056868	0.0	7.320331	0.000000	0.0	0.0	...	9.623335	9.823921	6.555327	9.064002	11
798	0.0	3.249582	3.707492	8.185901	9.504082	0.0	7.536589	1.811101	0.0	0.0	...	8.610704	10.485517	3.589763	9.350636	12
799	0.0	2.590339	2.787976	7.318624	9.987136	0.0	9.213464	0.000000	0.0	0.0	...	8.605387	11.004677	4.745888	9.626383	11
800	0.0	2.325242	3.805932	6.530246	9.560367	0.0	7.957027	0.000000	0.0	0.0	...	8.594354	10.243079	9.139459	10.102934	11

Figure 1. Snapshot of RNA-Seq Gene Expression Dataset

Figure 2 shows the different tumor class types represented in the dataset and their frequency. There is clearly a much higher number of samples for BRCA tumors, which may make the classification models more accurate for this tumor type. Given the large number of samples in this dataset and multiple classes, there are a number of ways to experiment with classification for the different tumor types. Knowing the tumor classes for the different samples allows us to create supervised learning models.

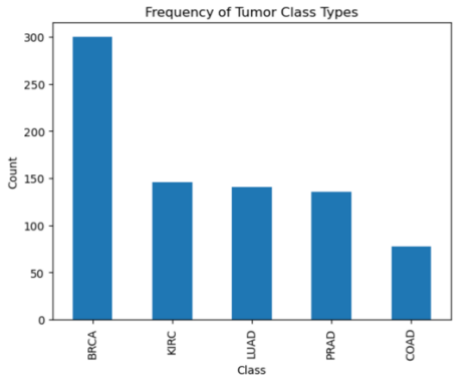

Figure 2. Frequency of Tumor Class Types

Methodology

Principal Component Analysis

To begin analyzing this dataset, it would be helpful to find a way to represent the data visually using the very large number of features provided. One simple method is to run Principal Component Analysis on the data to transform the data from 20,531 features to 3 features. The 3-dimensional representation can then be plotted and further analyzed. One of the main benefits of PCA is that it allows you to summarize complex data and capture its most important aspects. It aims to preserve as much variance from the original gene expression data as possible in the reduced number of features[5]. PCA can be done using the scikit-learn PCA function, which allows you to select the desired number of components. After running PCA on the data, I will plot all 801 samples of 3-D data. In order to show the different classes clearly on the scatter plot, I will convert the 'class' column of the dataset from strings ('BRCA', 'KIRC', 'LUAD', 'PRAD', 'COAD') to digits (1, 2, 3, 4, 5) so that a different color can be assigned to each

class. It will be interesting to see if there are very distinct groups in the resulting plot, whether or not the groups are very spaced out, and what groups lie closed to each other than others due to more similarity in features.

Logistic Regression

Logistic regression is a parametric classification model that predicts a binary outcome using a set of independent variables based on the input data features. Because there are 5 tumor types, this will be a multinomial logistic regression model. This model will be used to classify the samples based on tumor type. The class labels will have already been converted into integers for PCA, so I can use these integer labels (1, 2, 3, 4, 5) as the response variables for the logistic regression model. To start, the data will have to be split into training and testing data. I plan to use a 80-20 train-test split, so the model will be trained on 80% of the data and tested on the remaining 20%. Before training the model, it's important to make sure the best possible hyperparameters are used. We want to avoid overfitting the model to the training data while also maintaining accuracy. It is important for the model to be general enough to so that it has high predictive power on new data, particularly for a complex dataset like this with such a large number of features.

Cross-validation allows us to train and evaluate the performance of the model on multiple folds of data using varying values for a particular hyperparameter. The parameter of interest in the case of scikit-learn LogisticRegression function is C, which is the inverse of regularization strength. Smaller values of C specify stronger regularization (less weight given to training data)[6]. Cross-validation will be done using the LogisticRegressionCV function in scikit-learn, where a range of C values and number of folds can be specified. I will do 5-fold cross-validation with the default Cs = 10 which creates a grid of values to try for C. Because there are more than 2 possible classes, this is a multinomial logistic regression problem, so I will also set the variable multi_class = 'multinomial'.

The nature of logistic regression is binary, so I will have a separate set of coefficients for each of the 5 classes (i.e. 5 different models). The intention behind logistic regression is to see which genes have a greater effect on the likelihood of a sample belonging to a certain class. For example, it may be the case that feature #200 has a very high coefficient and thus high correlation with the BRCA class. I'll be looking at the coefficients with highest magnitude in each of the 5 models as this provides valuable insight into how different gene sequences factor into different tumors classes. I'll also be looking at the overall accuracy score, confusion matrix, and the individual accuracy scores for each of the 5 classes.

K-Nearest Neighbor

KNN is another useful method for classification. Unlike logistic regression, KNN supports non-linear solutions. KNN is an appealing algorithm for studying gene expression because it is adaptable to different data types and irregular feature spaces. It is often described as a lazy algorithm because it does not make any generalizations using the training data, which can be an asset[7]. KNN stores all training data and classifies new data based on a similarity metric. KNN can be run using the scikit-learn KNeighborsClassifier function. I will be using Minkowski distance as the similarity metric for this function. The number of neighbors used to classify new data can be defined using the n_neighbors variable. I will use trial-and-error to find an appropriate value for n_neighbors. I'll be using the same 80-20 train-test split data from the logistic regression model so the two classification models can be easily

compared. As before, I will be using the test data to calculate the overall accuracy score, display the confusion matrix, and calculate accuracy scores by class.

Evaluation and Final Results

The results of running PCA on the data show very distinct grouping of the different tumor types based on the reduced number of features. A plot of the 3-dimensional data is shown in Figure 3. Despite the number of features for each sample being reduced from 20,531 to 3, clearly much of the classifying information has been maintained in the reduced form. The data has been color-coded by class, so it is easy to see how each class forms a cluster, with the exception of a few stray points.

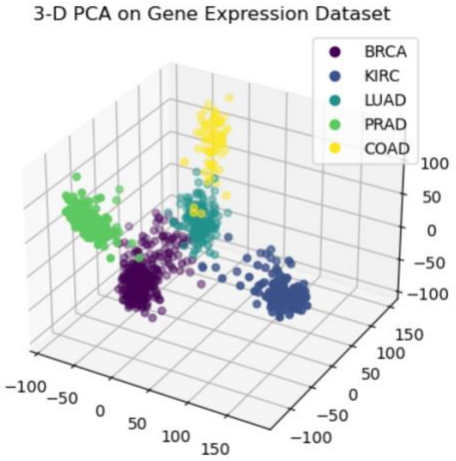

Figure 3. 3-D PCA on gene expression dataset

Another representation of the 3-D data is shown in Figure 4. This is the same plot but from a different angle. The reason for plotting multiple angles is to get a better picture of which classes are the closest in distance (similarity). There seems to be some overlap between the BRCA and LUAD tumor classes, while COAD, PRAD, and KIRC seem to be quite far apart and distinct, with some exceptions. These plots indicate that certain aspects of the gene expression levels from the original dataset are highly unique for all 5 classes, with particularly distinguishable information for the COAD, PRAD, and KIRC classes.

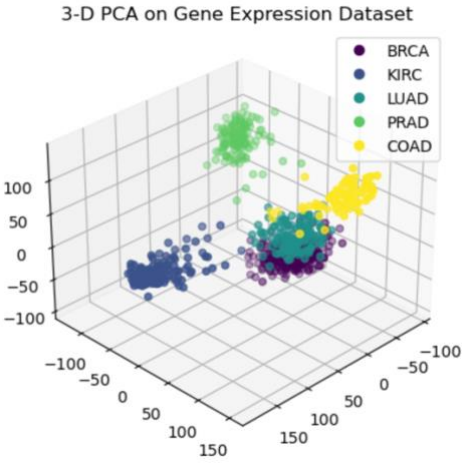

Figure 4. 3-D PCA on gene expression dataset (different angle)

The logistic regression model proved to be quite accurate in its classification of new data. After running 5-fold Cross Validation on the logistic regression model, I settled on a hyperparameter value of C = 0.0001 for all 5 classes, and the resulting predictions had a 99.379% accuracy rate overall when predicted on the 20% of the data reserved for testing. The confusion matrix for the logistic regression model is shown in Figure 5. As you can see, only 1 of the tested points was misclassified. This test point was misclassified as BRCA but was actually a LUAD sample. Looking at the PCA representation of the data from Figures 3 and 4, this makes sense given that LUAD and BRCA are two of the closest, most overlapping classes. Apart from LUAD having an accuracy score of 96.875%, all other classes had accuracy scores of 100%.

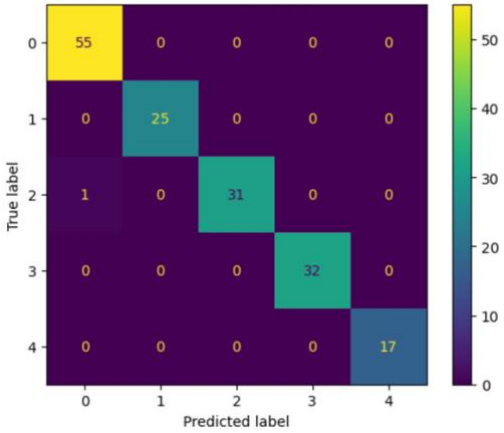

Figure 5. Logistic regression model confusion matrix

Using the .coef_ feature for logistic regression models in scikit-learn, I was able to create a data frame of the coefficients for all features for each class. Most coefficient were of extremely low magnitude (< 1e-7), others were closer to 0.005. I found the absolute value maximum coefficient for each column (class) and found its corresponding feature ID using idxmax(). The genes with the highest magnitude coefficient for each class (highest impact on probability of a sample being part of the class) are as follows: BRCA - gene_15589, KIRC - gene_3439, LUAD - gene_15898, PRAD - gene_9176, COAD - gene_5829. Once again a similarity can be seen between BRCA and LUAD as the most impactful genes are located very close to one another, whereas KIRC, PRAD, and COAD's are spread apart. Because the goal of this model is successful classification, this project wont delve deeper into research on the different genes and the mutational profiles of the different tumor types, but this would be an interesting topic to study in a future project. More information on the specific genes can be found by searching the feature names in the NIH National Library of Medicine database[8].

Finally, a K-Nearest Neighbor model was built as another method of classifying tumor types. As explained in the Methodology section, KNN has a very different approach to training and testing than logistic regression. Training it a much simpler process for KNN because it does not generalize the data to a function, however, testing is usually a more computationally expensive process as the model has to find the closest data points for each new piece of test data. The overall accuracy rate of the KNN model when predicted on the test data was again 99.379%. The confusion matrix is shown in Figure 6. Unlike the logistic regression model, which misclassified a LUAD sample, this model misclassified a KIRC tumor as BRCA. Looking again at the PCA plot, there are a couple KIRC data points close to the BRCA region, so this is understandable. The error rate is so minimal for both classification models that attempts at adjusting the models to correct these classifications may lead to overfitting.

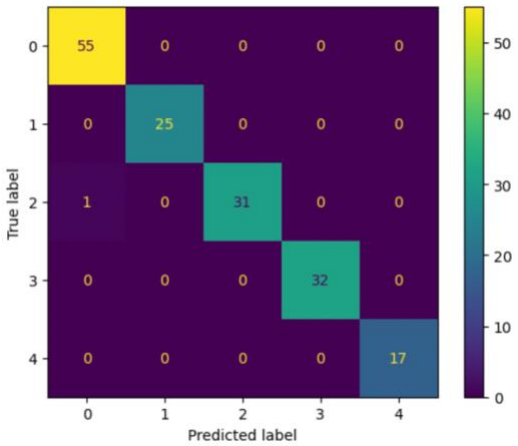

Figure 6. KNN model confusion matrix

Conclusion

Analyzing large datasets of gene expression for tumor classification is one of the many useful applications of Machine Learning in the field of Biology. The human genome is vast with so much information yet to be researched. The RNA-Seq PANCAN dataset is a perfect example of the wealth of information to be found in gene expression data. As shown in the PCA representation of this dataset, the 5 tumor classes represented in the dataset have distinguishable patterns of gene expression such that very clear groups are formed. The results of my analysis show that the dataset can accurately classify test data at a rate of approximately 99.379%. While this percentage may differ for different subsets of gene expression data, this is still an undeniably successful classification rate, which proves the utility of the RNA-Seq dataset. It is unsurprising that the KNN model had a similarly high classification rate given that we have seen how distinct the tumor class groups are, and given how well the linear logistic regression model was able to perform.

A noteworthy finding of the results were the coefficients on the logistic regression model. Many of the variables in logistic regression had almost negligible coefficients, but some stood out as having a more significant impact on the classifier. It would be interesting to take a closer look at the other gene expression variables with high coefficients above a certain threshold and compare how many of these overlap amongst the different tumor classes. Additionally, I would be interested in learning more information about the genes that were randomly extracted for this dataset and any correlation between location of genes that were most significant in the logistic regression model.

The aim of this project was to use Machine Learning models to classify the RNA-Seq gene expression dataset by tumor type using different methods including PCA, Logistic Regression, and K-Nearest Neighbor. The results show that this is an effective method of analyzing genetic data and making predictions. As the field of Bioinformatics continues to grow, this will surely be an area of further research and advancements.

References

1. *RNA-seq: Basics, applications and Protocol*. Genomics Research from Technology Networks. (n.d.). https://www.technologynetworks.com/genomics/articles/rna-seq-basics-applications-and-protocol-299461
2. *Transcriptome Fact sheet*. Genome.gov. (n.d.). https://www.genome.gov/about-genomics/fact-sheets/Transcriptome-Fact-Sheet#:~:text=What%20can%20a%20transcriptome%20tell,and%20tissues%20of%20an%20organism
3. *Cancer RNA sequencing*. Cancer Transcriptome Analysis with RNA-Seq. (n.d.). http://www.illumina.com/areas-of-interest/cancer/research/sequencing-methods/cancer-rna-seq.html#:~:text=Sequencing%20the%20coding%20regions%20or,important%20component%20of%20gene%20regulation
4. *Gene expression cancer RNA-seq*. UCI Machine Learning Repository. (n.d.). https://archive.ics.uci.edu/dataset/401/gene+expression+cancer+rna+seq
5. Avcontentteam. (2023, September 13). *PCA: What is Principal Component Analysis & How It Works? (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/
6. Rithp. (2023, January 4). *Logistic Regression and Regularization: Avoiding Overfitting and Improving Generalization.* Medium. medium.com/@rithpansanga/logistic-regression-and-regularization-avoiding-overfitting-and-improving-generalization-e9afdcddd09d.
7. Chatterjee. (2023, June 5). *A Quick Introduction to KNN Algorithm*. Great Learning. https://www.mygreatlearning.com/blog/knn-algorithm-introduction/#:~:text=KNN%20is%20non%2Dparametric%20since,points%20to%20make%20any%20generalisation.
8. *National Center for Biotechnology Information*, U.S. National Library of Medicine, www.ncbi.nlm.nih.gov/gene.