Tumor Type Classification Using Gene Expression Data and Linkages to Potential Biomarkers

Jeffrey Yeung
yjeffreyeung@gmail.com
Scripps Research Institute
La Jolla, CA

Abstract

The Cancer Genome Atlas (TCGA) has analyzed over 11,000 tumor expressions from 33 forms of cancer to better understand the causes of various cancers. The project has both engineering and scientific goals. The scientific goals attempt to further research in early cancer detection; the engineering goals attempt to apply different machine learning classification techniques to determine whether gene expression is truly indicative of disease presence. Our project explores these engineering goals and develops models to test whether the most discriminative genes point to existing biomarkers. We replicate the results of an effective CNN model and propose the use of two other methods that prove to be equally effective at tackling this problem.

Keywords: *Deep Learning, Convolutional Neural Network, Adaboost, Variational Autoencoder, Dimensionality Reduction, Cancer Tumor Classification, Gene Expression*

1. Introduction

Cancer is among the leading causes of death worldwide. In 2018, there were over 18 million new cases and 9.5 million cancer-related deaths [1]. Changes in the DNA sequence of the genomes of cells are the main cause of cancer occurrence [10]. Genes are part of a DNA sequence that contains complex instructions to instruct cells to produce particular proteins. Researchers use RNA-seq a methodology to measure gene expression patterns [9]. Studies show analyses of RNA-seq expression may yield clues to the roots of cancer and linkages to potential biomarkers for each tumor. For example, analyzing of RNA-seq expression has been broadly used to analyze and improve the diagnosis of breast cancer and prostate cancer [4].

The Cancer Genome Atlas (TCGA) characterized over 11,000 different tumor tissues and matched normal samples spanning 33 various cancer types. They released the Genomic Data Commons (GDC) Data Portal that is a robust data-driven platform and incorporates cancer data information such as the RNA-seq expression data of each sample. Researchers could access these expression data to learn the cause of different cancers and find linkages to tumor-specific biomarkers by classifying tumor types.

Prevalent tumor type classification and tumor-specific biomarkers provide for a faster and more reliable cancer diagnosis. However, little research is done in developing machine learning classification models for tumor types, mainly due to the curse of the high dimensionality of the genomic dataset [4]. Existing models demand extensive training on data that usually belong to disparate classes of samples where there is a high chance of redundancy. Moreover, typical gene analysis methods focus on single cancer types by matching both the normal and tumor samples from the same tumor type, where it loses the correlation between features of all other cancer types.

Our work addressed these challenges using deep neural networks (DNN) as they can be utilized as a suitable option to provide superior performance and deal with high dimensionality in the above-mentioned cases. DNNs have been widely used in a variety of computer vision applications, such as image classification [**?**], image recognition [7], and object detection[6]. These networks provide multi-layered data-friendly models that are highly efficient in learning hidden features, and finally, use a supervised or unsupervised approach for classification or regression purposes. Therefore, these DNNs are also suited for scenarios concerning the high dimensionality issue.

Our work consists of three main models.

First, we validated the existing model motivated by one such DNN network to discover top genes for each tumor [5]. In this model, a Convolutional Neural Network (CNN) is applied as feature importance can be mapped back through gradient activation mappings as shown in Figure 1. This gave a way to see which genes are influencing classification the most and more importantly shedding light on which genes affect the formation of certain cancers. However, to the best of our knowledge, this work is not peer-reviewed

nor does it contain working out of the box code hence why we first validate their results as a baseline.

Second, we generated an Adaptive Boosting method to directly extract the most distinct features of each class. We used a one-vs-all architecture in order to accomplish this. Final classification was done by finding the maximum output of each one-vs-all classifier. Our motivation behind using Adaboost was to have an effective supervised learning method that would be trained one feature at a time to compare with the DNN models.

Third, we used a Variational Auto-encoder (VAE) in order to reduce the dimensionality of the data to the most distinctive features. The reason the VAE implementation was chosen over other implementations of auto-encoders is because the latent Gaussian representation of the data ensures that the latent space gradient is smooth (which discourages large irregularities between features in the latent space). This data was then used in order to train a feedforward neural network with a softmax output for classification.

Lastly, as a baseline study we implemented a simple PCA to compare its results with those from our implemented methods. We expected PCA to not perform as well due to the extremely high dimensionality of the dataset, and features of high covariance may not always be indicative of important genes.

Finally, using these three networks, we explored if gene expression is actually indicative of disease by its ability to discriminate between different tumors by testing them on the TCGA dataset. We also tested the ability of our model to classify tumor types by determining if the most discriminative or variant genes point to already existing or novel biomarkers.

The remaining of this paper is organized as follows.

In section 2, we reviewed the related work of biological methods to find keygens linked to different cancers, and classified various tumor type classification.

In section 3, we described the data we use, and the algorithms that we applied for developing the classification models.

In section 4, we presented the ability of these methods to classify tumor types and compared the accuracy of these tumor type classification models.

Lastly, we concluded our completed work with insights to the implications of our results, as well as suggestions for future expansions.

2. Related Work

Various methods of analytics have been applied to find trends or connection between the vast amount of biological and omics data to diseases or phenotypical variation. These methods are typically aimed at a specific variation or disease and aim to find the key biomarkers driving such issues. In all pasts works, groups had to deal with the notoriously large feature space of such data and brought up interesting methods to ameliorate it.

In [3] they use traditional biological methods to find key genes linked to breast cancer. This gives a good metric to compare against when analyzing our results. However, to address the high feature space of the data they only use the known variants (GWAS data) occurring for each gene. This is a very small subset of data available. In most biological problems there is such a vast amount of data with a huge dimensionality that these problems point to methods of machine and deep learning to help deal with such issues. In [12] a Variational Autoencoder is used to help reduce the dimensionality of RNA sequence data (20k features) to a latent space of 100 dimensions. This method though relies of cutting out 15k of the features already to fit the network.

Furthermore in [8] and [2] convolutional neural network methods are used to filter through all of the features and draw key points in each method. In [8] a two dimensional CNN is applied to images of RNA sequence data. Still the authors of [8] artificially reduce the dimensionality of the data and shape it into images to fit widely use image techniques already established. It works in the classification for their results but it also introduces a second spatial dimensions for the CNN that actually doesn't exist. This creates correlation between certain features that is not representative of the actual data. A CNN is applied in this manor due to the capabilities of CNNs to recover the activation mappings. These mappings are a way to determine which features are actually useful in the network for classification. The idea behind this is that the genes/features that are import for classification are actually biomarkers in real life and point to the underlying mechanisms of such diseases. This is a novel approach although there are some oddities in the method.

Lastly, as stated earlier in [2], a convolutional nerual network is also used. However the method of the convolutions are in a single dimension. This is done for DNA sequence data. Since sequence data is so long and redundant. The idea is that there are repeatable patterns and trends in all runs of DNA that can be identified by the single dimensional convolutions. The method that is done in this paper is for identifying specific biomarkers on the DNA sequences. These are the promoter and enhancer regions where proteins bind. This method of single dimension scanning convolutions are also novel application if their field. However it looses long run relationships between features over long distances. It can only identify local features.

3. Methods

The various related works mentioned above take different forms of deep learning techniques and apply them to different datasets. Here we extend on their ideas and focus

specifically on the TCGA cancer dataset.

3.1. Adaboost

Since the goal was to learn the most discriminant features, one approach we decided to analyze was Adaboost. This approach directly extracted the most discriminant features one at a time without making any assumptions about dependency of one feature on another. Additionally, it was crucial to find features unique to each class, thus we decided on one-vs-all classifiers. This approach produced results comparable to the others with respect to multiclass classification error.

3.1.1 Algorithm

This section will explain the Adaboost algorithm. 33 one-vs-all classifiers are trained using decision stumps. Each classifier is trained separately. For each classifier, the algorithm begins by initializing the cost function $g^{(t)} = 0, t = 0$ and the thresholds $T = (0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1)$.

Each iteration does the following 4 things:

First, compute

$$k_i = exp(-y_i g^{(t)}(x_i)) \tag{1}$$

Second, compute the $u(x_i)$ that maximizes

$$\alpha_t(x) = argmax_u \sum_i y_i u(x_i) k_i \tag{2}$$

where $u(x_i)$ is defined as

$$u(x; j, l) = \begin{cases} 1 & x_j \geq T(l) \\ -1 & otherwise \end{cases} \tag{3}$$

Thus the algorithm needs to find one feature and one threshold per iteration. This optimal $u(x)$ will be referred to as $\alpha_t(x)$. It is also important to consider $-u(x)$, which is the opposite of $u(x)$. This function may be the best maximizer.

Third, the step size is computed by

$$w_t = \frac{1}{2} log(\frac{1-\epsilon}{\epsilon}) \tag{4}$$

where

$$\epsilon = \frac{\sum_{i|y_i \neq \alpha_t(x_i)}(k_i)}{\sum_i (k_i)} \tag{5}$$

And lastly, update

$$g^{(t+1)} = g^{(t)} + w_t \alpha_t(x) \tag{6}$$

Iterate until training error for each 1-vs-all classifier reaches zero. Final classification is done by finding the maximum $g(x)$ produced by the classifiers.

3.2. Convolutional Neural Network

https://www.overleaf.com/project/5c8e0a937e47d72f651d3c87
The method of classification following a CNN architecture is used because of the ability to analyze the activation mappings as shown in Figure 1. The CNN architecture is a powerful feature extractor that uses the filters to find local relationships between genes. For the architecture the data is first down sampled to reach a perfect square of 102 such that the images can be reshaped into squares.The top 10404 varying features are reshaped to make the images. The network architecture is as follows in figure 1.

In the work presented by [5], they removed the genes data with low variance features for feature reduction purposes. Next, they embedded the high-dimension expression data into a 2-D image with the size of 102x102. For the CNN, they developed a 3 layer network and for testing purposes, they implied 10-fold cross-validation to test the performance. Then, they generated heat-maps for all the classes showing outstanding genes.

In the future we wanted to implement the single dimensional convolutions like what is done in [2]. The original data is single dimensional and hence it makes sense to keep those spatial relationships. Furthermore we would be able to append gene wide variant intensities.

3.3. Variational Autoencoder

At a high level view, a VAE consists of an encoder and decoder. The encoder serves to represent the matrix in a latent space often of different dimension. The decoder takes these latent space representation and maps them back to the original input space. The difference between VAE and other encoders are the assumptions that are placed on the encoder in order to ensure that the latent space is smooth.[**?**] This smoothness in the latent space is useful as it is more likely to provide more biologically useful results.

3.3.1 Algorithm

In this application, we choose a latent space of $k = 100$. In the VAE formulation, however, we also mandate that the latent space (whose samples are denoted as z) represent a Gaussian posterior $q_\theta(z|x)$. The set of parameters that encodes this distribution are represented by Θ. The encoder takes this latent space representation of an input z and returns an approximation of the original input by learning a set of weights W such that the latent vector maps back to the original input as $\hat{x} = Wz$ (Note that the z here is augmented by 1 in order to incorporate the bias). Assuming that the encoder has some formulation $p_W(x|z)$, the objective function for the VAE is formulated as follows:

$$L(\Theta, W) = \mathbb{E}\left[\log p_W(x|z)\right] - \mathbb{KL}\left(q_\theta(x|z)||p(z)\right) \tag{7}$$

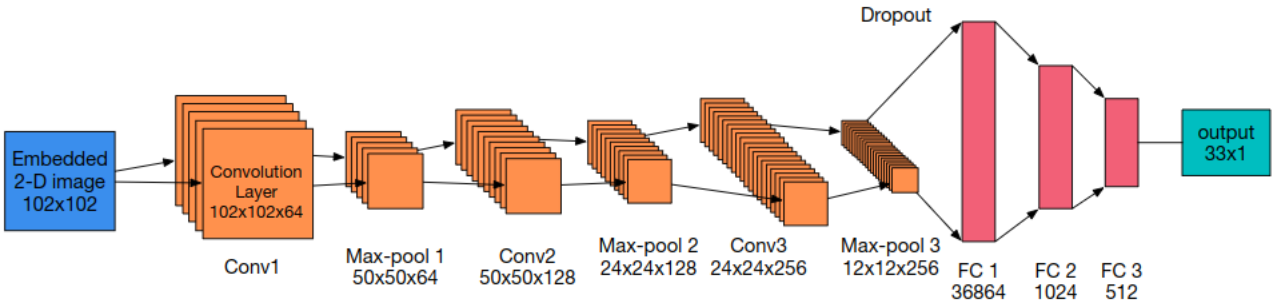

Figure 5: The architecture of our convolutional neural network.

Figure 1. CNN Network Architecture adopted from [8]

Notice that the objective function attempts to find maximize the log-likelihood of the decoder while simultaneously ensuring that the encoder remains similar to the true latent distribution $p(z)$. In the VAE formulation, we assume that $p(z)$ is sampled from a normal with zero mean and identity covariance as it keeps the objective easy to differentiate. In order to train the network, gradient descent is used to optimize the parameters starting from the objective. The updates are of the typical form with α representing the learning rate:

$$\Theta^{n+1} = \Theta^n + \alpha \nabla_\Theta L(\Theta, W) \tag{8}$$

$$W^{n+1} = W^n + \alpha \nabla_W L(\Theta, W) \tag{9}$$

3.3.2 Evaluation

With these updates in mind, we can then proceed to train the network until the data begins to show signs of overfitting. After the network is trained, the latent space representation is used in a two-layer feedforward network with a softmax output. The number of hidden neurons was chosen to be 20; although, it could have varied anywhere between 20 to 40 and maintained the same results. The derivations for the feed-forward network will not be covered here as there is sufficient prior knowledge from previous assignments in order to understand the training and update procedures.

3.4. Principal Component Analysis

A central problem surrounding this project is the high dimensionality of the genomic dataset. A common data dimensionality reduction technique is the Principal Component Analysis algorithm (PCA). PCA is technique used to find patterns in data of high dimensionality, and as a baseline verification, we implemented a simple PCA to compare with the results from our other methods.

3.4.1 Algorithm

The goal of PCA is to transform a given dataset $\mathbf{X}$ of dimension p to a smaller data set $\mathbf{Y}$ of a lower dimension L. To begin performing PCA we must ensure the data is arranged as a set of n data vectors represented as row vectors, each with p number of columns. First, compute the mean of each column $j = 1, 2..., p$ and subtract it from each row value in the corresponding column. Mean subtraction is important to minimize the mean square error ofapproximating the data.

Next, compute the covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$ and find the eigenvalues and the eigenvectors of the covariance matrix. This can be computed from $D = V^{-1}\Sigma V$ where D is the diagonal matrix with the eigenvalues on the diagonal indices, and the matrix V is the matrix of eigenvectors corresponding to the eigenvalues of Σ.

Once the covariance matrix is computed, obtain the top n principal components corresponding to the columns of the eigenvector matrix V of the top n highest eigenvalues as V'.

The final reduced data is calculated as

$$Y = V'^T(X - \mu_X) \tag{10}$$

4. Experiments

4.1. Adaboost

After mean and variational filtering, the Adaboost algorithm described in Section 3.1.1 was run on 80% of the dataset and validated using the remaining 20%. Final multiclass accuracy was 95.16%, which is very close to the performance achieved by the DNN methods.

Looking at the confusion matrix (See Figure 2,) there were some misclassifications. COAD and READ were a common misclassification, which was also observed in the CNN paper [8]. These tumors are spatially very close in the body, and this was a trend observed with some of the other misclassified tumors as well. Figure 3 shows that COAD,

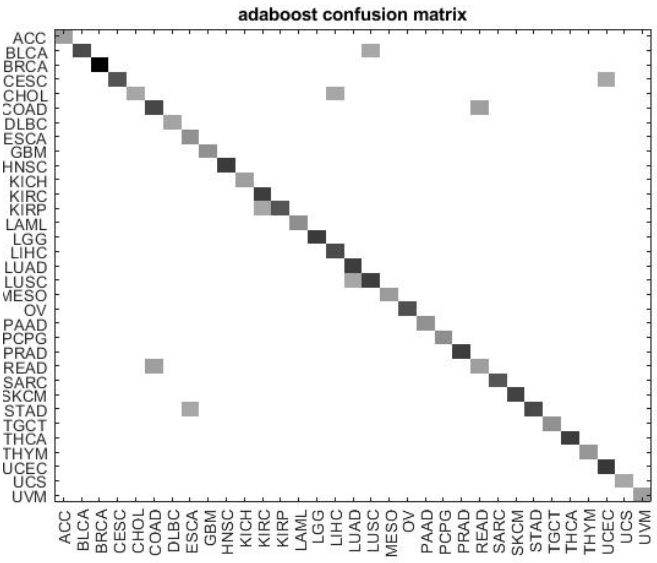

Figure 2. Confusion Matrix of Adaboost Method

READ, LUAD, and LUSC had a large amount of features selected compared to the other classifiers, so their misclassifications may also be due to overfitting. Other misclassifications, such as CHOL, may simply be due to extremely small sample size.

While most tumors were classified using genes that related to the affected part of the body, some classifiers selected genes that did not relate to the tumor region. This could be due to the 1-vs-all classifier model. While a certain gene may separate one tumor from most of the others,

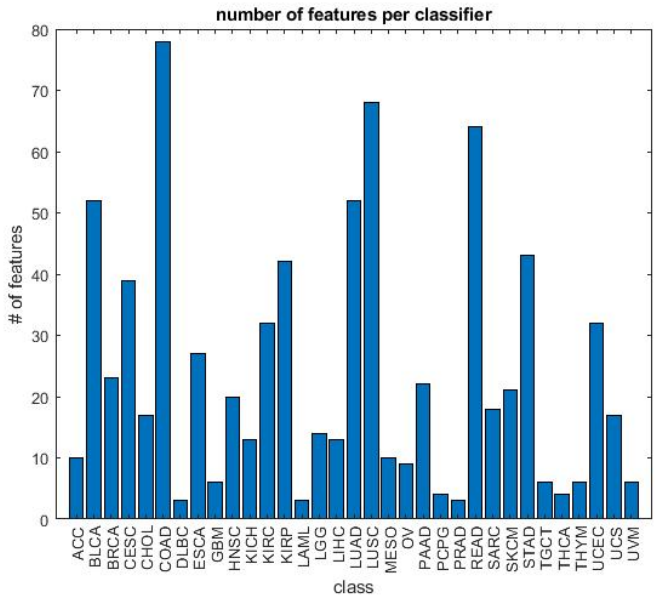

Figure 3. Features Selected By Class of Adaboost

a gene shared between two tumor types should not always be picked, so the classifier may stray from only picking spatially relevant features. Additionally, some genes may have been selected so that the classifier can fine tune against one problematic class. A result where the model chose genes as expected is with PRAD, with every selected gene directly relating to the affected body region. The selected genes were KLK2, NKX3-1, and SLC45A3. A case where this model gave unexpected results was ACC, where the classifier selected a mixture of genes relating to the tumor region and other non-tumor regions.

4.2. Convolutional Neural Network

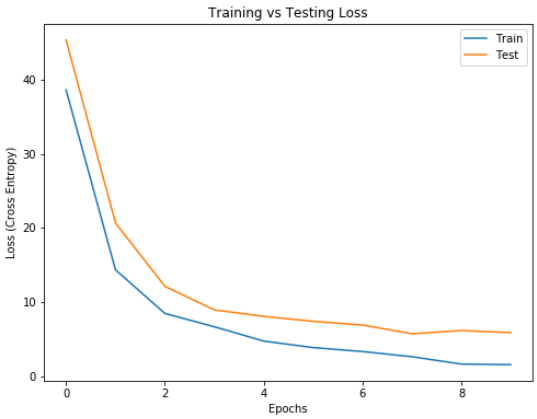

Figure 4. Training versus testing loss of CNN training

Our convolutional neural network was able to achieve a testing accuracy of 95.6%. The training reached a reasonable loss within 10 epochs of the data. Training was stopped after the loss decreased after successive runs by a certain amount. This was due to prevent over fitting. The activation maps are then run on the images which are shown in the appendix below. There is also included the activation maps, gradients and original image. In the activation maps we analyzed the top expressed values. These genes actually match genes that are important for specific cancers and tissues. For example for the ACC cancer the top most expressed gene was CYP11B1 which is extremely important in adrenal glands where the cancer for ACC is found.

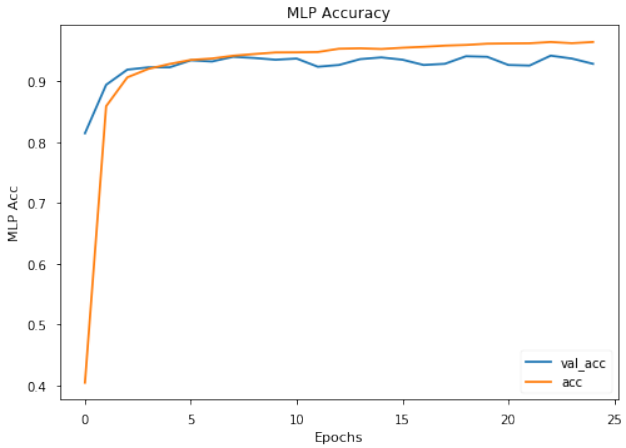

Figure 6. The accuracy during training of the MLP

4.3. Variational Autoencoder

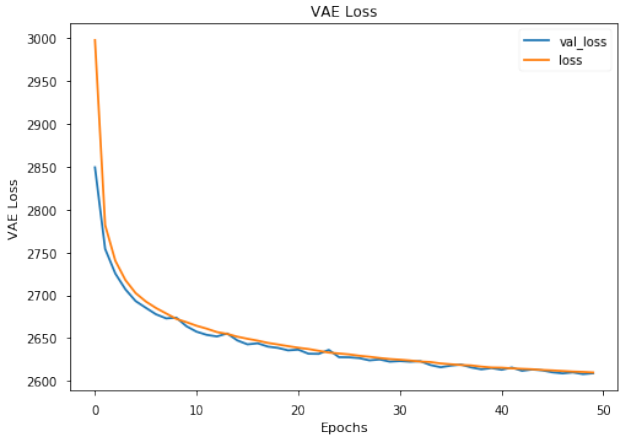

Figure 5. Training and validation loss values during VAE training

As seen in Figure 5, training did reduce the loss of the overall VAE model. One thing to note is that the loss is still particularly high; the reason for this is because the model used did not normalize the loss across all samples. Doing so would give a more realistic interpretation of the model.

After training the VAE, the encoder was stripped from the overall model and used in order to encode the entire dataset into the latent space. The accuracy of the feedforward network during training can be seen in Figure 6. The testing accuracy of the final network model was found to be 94.9%. Given this accuracy, it is possible to extract features of interest from the classification network and map them back to the original space by considering the decoder of the VAE. As an example, the first cancer type was highly dependent on feature 30 of the latent space. The VAE then reveals that the top contributing genes for this feature were shown in Table 1. Notice that the genes with asterisks are known to be strongly connected with a specific type of cancer while the others are genes that are generally associated with cancer. This means that the network can be used to get a biological understanding of which genes are most likely to cause cancer.

Class 1 Most Relevant Genes
MOGAT3
PLA2G12B
AFP*
ALPI
LECT2*

Table 1. The most represented (positive) genes for the top feature of the first cancer type.

4.3.1 t-distributed stochastic neighbor embedding

t-distributed stochastic neighbor embedding (t-SNE) is a machine learning algorithm mainly used for visualization of high dimensional data. As its name implies, t-SNE embeds high-dimensional data into a lower dimensional space, usually two or three dimensions, to allow for visualization. t-SNE consists of two stages. First, it constructs a probability distribution over pairs of high-dimensional objects while attempting to maintain relative distances between all the points in space. Second, it defines a probability distribution over the points in the lower dimensional map and minimizes the Kullback-Leibler divergence between each pairs of distributions with respect to the locations of the points in space [11]. The result is then visualized as clusters, and in this project, since there are 33 different tumor types, there are 33 clusters corresponding to each.

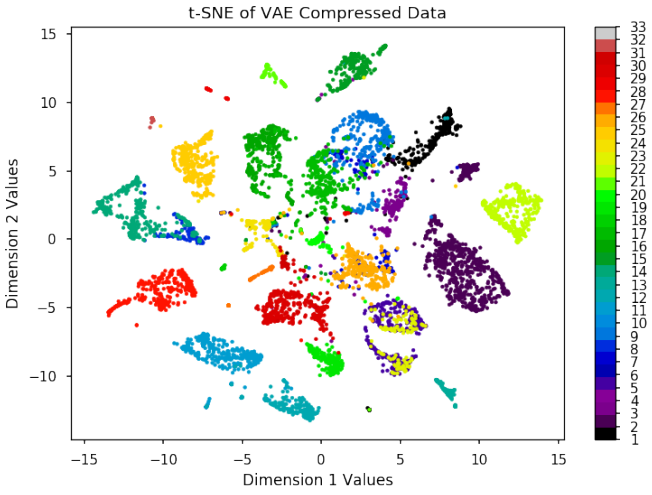

Figure 7. Clusters of t-SNE compressed data

4.4. Principal Component Analysis

As a baseline study of our dataset, PCA proved to be less effective than our other implemented methods. The final training accuracy was 93.78%, while the validation and testing accuracies dropped significantly to 22.25% and 33.62%, respectively. We believe that the drastic loss of accuracies is due to the extremely high dimensionality of the dataset, so PCA is unable to differentiate between the important features since it only takes account into features of large variances, which may not always be the most indicative features.

5. Conclusion

Our results upheld the findings of the CNN paper [8] and we were able to achieve similar results with other machine learning models. This paper is significant because this is an under-explored field and these models can be applied in further research or can be improved. An addition of a control group of healthy tissue samples would be one avenue to explore with this dataset. We are especially interested in seeing the our proposed models applied to other datasets that examine differences between disease types as well as datasets that incorporate extra features such as genomics data, clinical data, and proteomics data.

Adaboost performed surprisingly well on this dataset, although some of the features selected do not seem functionally relevant to the tumors they classify. This method should be tested on another dataset in order to fully test the selected features. Additionally, one idea to potentially improve the Adaboost model is to first classify groups of tumors based on proximity in the body, then create a second layer of intra-group classifiers using the same method. This should lead to a layer of spatially significant features and then a second layer of more unique features.

The Variational Autoencoder's latent space representation of the dataset has shown that it has accurately maintained the most distinctive genes from the original dataset. This is exhibited by the high accuracy achieved from the classification model used to evaluate the latent space. Furthermore, the direct connections revealed by tracing back activations through the networks show that the latent space model can be used to determine connections between cancer types and specific genes.

Overall, we were satisfactory with our project and the results were inline with our expectations. We believe that our methods can be extended to procedures and datasets beyond tumor expressions and help improve the medical and scientific community in disease detection. Genetic information will always be high dimensional, and this paper may serve as an invaluable starting point for many classifications done in the future.

6. Appendix

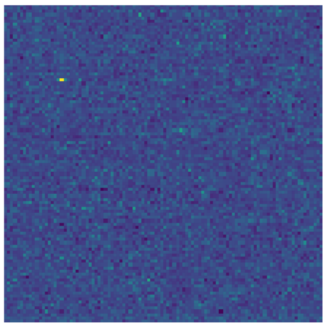

Figure 8. Regular Image class ACC

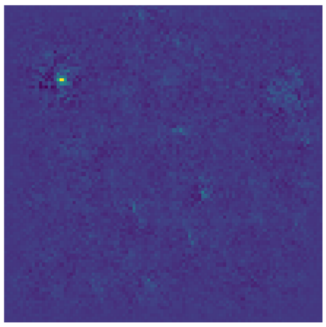

Figure 9. Gradient Mapping class ACC

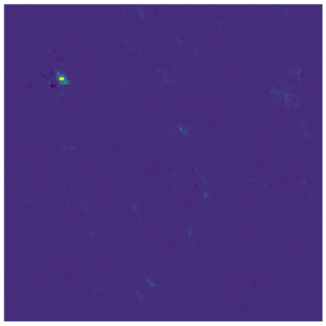

Figure 10. Gradient-weighted Class Activation Mapping class ACC

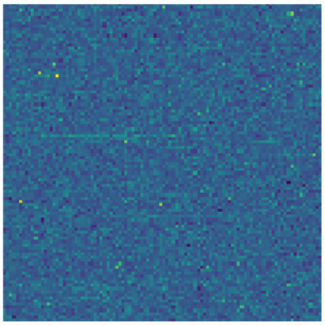

Figure 11. Regular Image class BRCA

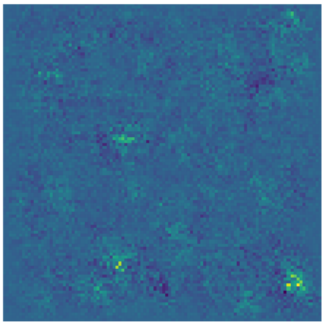

Figure 12. Gradient Mapping class BRCA

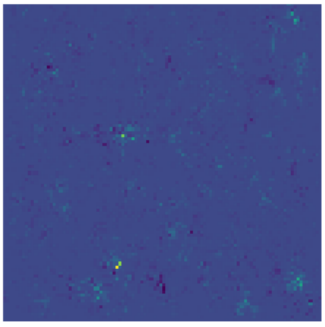

Figure 13. Gradient-weighted Class Activation Mapping class BRCA

References

[1] All-cancers-fact-sheet. [online]. available: http://gco.iarc.fr/today/fact-sheets-cancers,. Accessed: 17- Mar- 2019.

[2] M. W. B. Alipanahi, A. Delong and B. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*.

[3] J. Baxter, O. Leavy, N. Dryden, and O. Fletcher. Capture hi-c identifies putative target genes at 33 breast cancer risk loci. *NCBI*, 2018.

[4] R. R. Bhat, V. Viswanath, and L. X. Deepcancer: Detecting cancer through gene expressions via deep generative learning. *CoRR*, abs/1612.03211, 2016.

[5] L. Boyu and H. Anamul. Deep learning based tumor type classification using gene expression data. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 89–96, 2018.

[6] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[7] R. S. He K., Zhang X. and S. J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] B. Lyu and A. Haque. Deep learning based tumor type classification using gene expression data. 2018.

[9] U. Nagalakshmi, Z. Wang, C. Waern, K. Shou, G. M. Raha, D., and M. Snyder. The transcriptional landscape of the yeast genome defined by rna sequencing. *Science*, 320:1344–9, 2008.

[10] M. R. Stratton, P. J. Campbell, and P. A. Futreal1. The cancer genome. *The cancer genome, Nature*, pages 458(7239), 719–24, 2009.

[11] G. van der Maaten, L.J.P.; Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[12] G. Way and C. Green. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pacific Symposium on Biocomputing*, 2018.