Gene Attribution in Cancer Classification

Irene Pham

December 2020

1 Abstract

The Cancer Genome Atlas [2] stands as one of the largest and richest repositories of cancer data publicly available. The large amount of data allows for the application of deep learning as done by Lyu et al. [4] who achieved 95% accuracy on cancer type classification. Based on this, and the development of a wide variety of gradient based attribution methods, this paper seeks to apply gradient-based attribution methods to the model designed by Lyu et al. to identify informative features. The ability of each attribution method to identify information features is then tested by training the model on the genes with the highest attribution given. In addition, the agreement between the attribution methods will be examined. Finally, a literature review of the genes with the highest attribution will be done to see if they are known biomarkers or are related to certain cancer types. Successful identification of genes characteristic to certain cancer types could contribute to novel biomarker identification.

2 Introduction

Next Generation Sequencing (NGS) and increased computational capabilities have dramatically changed the way biological and pharmaceutical research is done. Although NGS is fairly new considering the first NGS method, 454 Sequencing, was only developed in 2005, the large impact of NGS can be explained by the low cost of sequencing and variety of sequencing data these recent developments have allowed. One of the most common types of sequencing is mRNA sequencing which measures gene expression level [1].

With the rise of NGS, large amounts of sequencing data have been generated and in some cases been made available to the public. One of these datasets is The Cancer Genome Atlas (TCGA) which contains the results of mRNA sequencing for over 10,000 cancer samples for 33 cancer types [2]. This provides a rich dataset for use in machine learning and neural network approaches.

Several articles have been published describing various neural network based architectures used to perform cancer type classification. Motsavi et al. evaluated the ability of multiple neural network models to classify by cancer type [3]. The first consisted of a 1D inputs with 1D convolutions followed by pooling and a series of fully connected linear layers. The second consisted of a 2D input - created by reshaping the 1D samples - followed by a convolutional layer, max pooling, and a series of linear fully connected layers. Classification accuracy was approximately 95% and 94% for each model respectively [3]. Lyu et al. also

developed a neural network classifier that achieved 95% accuracy [4]. The model consumes a 2D input and has 3 convolutional layers each followed by max pooling and batch normalization. After the convolutional layers, dropout is applied before feeding into a series of fully connected linear layers [4].

Given that models with high classification accuracy have been developed, the next task of interest would be to identify genes (features) that are important in determining the class label of an input. This information would particularly helpful for biomarker discovery and drug development. Biomarkers are any sort of biological signal (such as high levels of a certain substance in the body, a tumor, or abnormal gene expression) that could be used to diagnose a disease [5]. Genes important in cancer type classification could be potential biomarkers for certain cancer types. In addition, they could be targets of cancer therapies.

Convolutional neural networks (CNN) have gained popularity due to their ability to perform a wide variety of tasks including image classification and recognition. Popular CNN architectures for image recognition include VGG, AlexNet, and Inception [6][7][8]. However, due to the black box nature of these models researchers have developed various attribution methods which attempt to make the decision making process behind the models more understandable and explainable [9] [10] [11]. The goal of attribution is often to determine what features (or pixels for images) are important for the model to make its decisions. There are several broad categories of interpretability methods including gradient based methods (also called saliency or back propagation based methods) and perturbation (occlusion) based methods [12]. At a high level gradient based methods calculate the gradient with respect to the input and input features with larger gradients have higher attribution. Based on these gradients, heat-maps (also called saliency maps) are often generated. Many gradient based attribution methods exist including Integrated gradients, SmoothGrad, Guided GradCAM, and "vanilla" gradients [13] [14] [15] [9]. Gradient based attribution methods are often applied to image recognition models since a rough evaluation can be done by inspecting the saliency map to see if it resembles the input.

Evaluating attribution methods empirically is generally challenging due to the fact that it's hard to differentiate between model errors and attribution method errors [12]. Despite these challenges, several papers have proposed evaluation metrics [12][16] although there is no one single evaluation metric agreed on in the field. A few articles have also proposed qualitative properties that attribution methods should have [13][] although, once again, these properties and their precise definitions are generally not agreed upon by the community. The most common forms of evaluation are visual inspection or analysis by a domain expert.

The simplest form of gradient based attribution methods was proposed by Simonyan et al. and is known as "vanilla gradients" [9]. Attributions for each feature of the input is simply the gradient flowing back to it.

Sundararajan et al. later proposed Integrated Gradients [13]. To calculate the integrated gradient for an input, a baseline input, x', must be identified. The baseline input must have no information. This would be a black image for image inputs or the zero embedding vector for NLP applications. Then the gradient for each point along the straight line from the baseline to the input is calculated and averaged to get the integrated gradient. Sundararajan et al. states that integrated gradients is a valid method because it satisfies

2 desirable properties for attribution methods: sensitivity and implementation invariance. Sundararajan et al. states that a method satisfies Sensitivity if "for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution". On the other hand, implementation invariance is met if the attribution method will assign the same attribution for 2 functionally equivalent networks. Two functionally equivalent networks give the same output for the same input for all possible inputs Sundararajan et al. then explains that most of the popular attribution methods violate either sensitivity or implementation invariance but integrated gradients does not [13].

Given the considerable accuracy of the cancer type classifier designed by Lyu et al. and the recent developments in gradient based attribution methods, this project attempts to apply integrated gradients and vanilla gradients to the classifier by Lyu et al. A brief overview of the experimental process can be seen in Figure 1. First the data will be preprocessed and the classifier by Lyu et al. will be recreated. Next, the vanilla gradients and integrated gradients will be computed for each input and averaged by cancer type and by the entire dataset. Then, the model by Lyu et al. will be retrained with only the top K percent of features with the highest attribution identified by each method. Afterwards, the difference in attributions assigned by each method will be compared. Finally, the genes with the highest attribution will be researched to see if they have any known association with cancer.

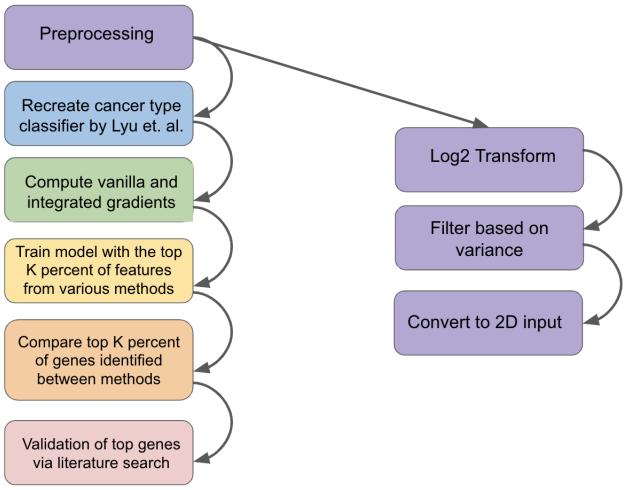

Figure 1: An overview of workflow performed in this project.

3 Experimental Design

3.1 Dataset

TCGA data was acquired from the Google Drive folder shared by Lyu et al. as a supporting document. Lyu et al. had downloaded and combined all normalized level 3 RNA seq expression data for all 33 types of cancer types using the Broad Institute Firehouse portal and combined it into a nice CSV file that can be found in the aforementioned site. The file

contains 10,446 samples for 20,531 genes. As seen in Figure 3, samples from some cancer types are more prevalent than others.

3.2 Preprocessing

The preprocessing steps that Lyu et al. used were replicated. First the data was transformed using x = log2(x+1). Then, the genes were filtered based on variance. A variance threshold of 1.18 was used to decrease the number of genes to 10,382 as Lyu et al. had done. As Lyu et al. had noted the range of values was still extremely high so the samples were then scaled to be between 0 and 255. Each sample was then folded into a 2D input of size 102 x 102 to be fed into the neural network, with 22 zeros added at the beginning. Before reshaping the inputs into 2D, Lyu et al. had reordered the features by chromosome so that genes on the same chromosome were located next to each other. However, this step was not performed. Lyu et al. had obviously scaled the data to be between 0 and 255 and folded it into a square input to mimic image data, however the logical reasoning behind this is not clear based on what the data is and how the features are associated with each other. Perhaps Lyu's preprocessing steps to image data mimicry by the great performance achieved.

3.3 Classifier Recreation

Lyu et al. designed a CNN classifier and this model was recreated based on the description presented in the paper and the source code they released.

As seen in Figure 2, taken from Lyu et al, the model consists of 3 convolutional layers, all with kernels of size 3, followed by a set of fully connected layers. Max pooling and batch normalization were used after convolutional layers with max pooling kernel sizes of 2 and strides of 2. As shown in Figure 2 the first convolution based half of the architecture is as follows : convolutional layer with 64 filters, max-pooling, batch normalization, convolutional layer with 128 filters, convolutional layer with 256 filters, max pooling, batch normalization. The second, linear based half of the neural network is as follows: dropout with a rate of 25%, and 3 linear layers of size 36864, 1024, and 512 each.

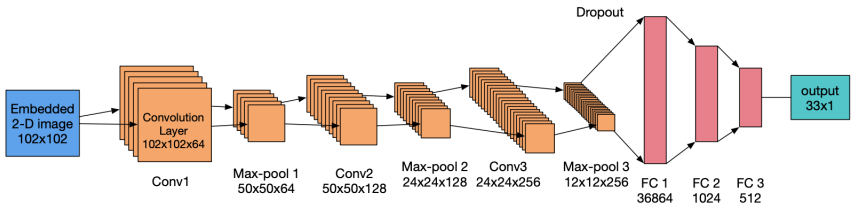

Figure 2: The CNN classifier designed by Lyu et al. This image is taken from the article by Lyu et al.[4]

The model was trained for 10 epochs with a learning rate of 0.0001, an Adam optimizer, and cross entropy loss. The test, validation, train split was 10%, 10%, and 80% respectively.

The replication of the model Lyu et. al. created resulted in 92% accuracy on the test set and 95% accuracy on all data. This corresponds to the 95% accuracy that Lyu et

al reported. Additional results and metrics for the model will be provided in the results section.

3.4 Vanilla and Integrated Gradients

Next, vanilla and integrated gradients were calculated to determine which genes were contributing most to classification. Vanilla gradient was implemented at a batch level by accessing the grads for the inputs through the PyTorch Autograd module. The average gradient for each class (cancer type) was calculated and stored. Average gradients were calculated on a class level so that the genes contributing to the classification of each cancer type could be examined downstream if desired.

Integrated gradients was also implemented from scratch in batch form. To calculate the integrated gradient, a baseline input which represents an input with no information is needed. As Sundarajan et al. mentions, for images this would be a black image [13]. Although Lyu et al. does process the data so that it mimics an image (2D square matrix, scaled between 0 and 255) using a black image as a baseline in this application does not make sense [4]. A black image where values are 0 would represent no expression for any of the genes. However, in this application, this doesn't translate into no information. Some genes are normally expressed at a certain level in non-cancerous samples so a value of zero would be abnormal and indicative of under-expression (possible cancer even). Thus, the baseline was chosen to be a sample where the value of each feature was the mean value of the feature for the entire dataset. This baseline better approximates the normal level of gene expression. However, considering that the dataset (obviously) contains cancerous samples where expression of certain genes may be higher or lower than usual, this baseline does not perfectly approximate normal gene expression levels. In addition, for genes that are over or under expressed in all cancer types, the average values will not mimic and non-cancerous sample. Thus, a better baseline would be a gene expression data from a person without cancer.

The original source code for integrated gradients uses 50 steps, meaning that for a single baseline and input, the vanilla gradient for 50 points on the line between the baseline and input are calculated and averaged. However, due to computational resource constraints and the need to compute the integrated gradients for over 10,000 samples, only 10 steps were used.

Although Sundaranjan et al. does not explicitly mention it in the publication for integrated gradients, the accompanying source code indicates that to compute the integrated gradient for a single input, multiple variants of the baseline are used. The integrated gradient from each baseline to the input is then computed and average. The released code creates variations of the baseline by multiplying the baseline by an equally sized matrix of random values. To create variations of the baseline, the mean and standard deviation for each feature was computed and values were randomly sampled from that normal distribution. Neither the integrated gradients article or source code specifies how many variants of the baseline or "random baseline trials" were used but due to computational resource limitations, 3 random baseline trials were used in this project.

3.5 Evaluating Performance of Model with Top K Features with Highest Attribution

Next, to compare the ability of vanilla gradients and integrated gradients to distinguish genes important for classification, subsets of the genes with highest attribution from each method were used to train a model. More specifically, the top 1, 2, 5, 10, 20, 50, 70, and 90 percent of genes identified by each method were used to train the same model by Lyu et al. In addition, the standard deviation for each feature was calculated and the top 1, 2, 5, 10, 20, 50, 70, and 90 percent of genes with the highest standard deviation were used to train the model by Lyu et al. The rationale behind this is that cancer is often caused by the over or under expression of genes. Thus, the genes with the largest standard deviations contain the most information - information that is used to distinguish between cancer types. As a sanity check, datasets where 1, 2, 5, 10, 70, and 90 percent of the total features were randomly selected were made and used to train the model. The accuracy of each model on the test set can be seen in Figure 4 and will be discussed in the results section.

3.6 Agreement Between Attribution Methods

Finally, the similarity of the attribution methods were compared by calculating the percent difference between the set of genes with the highest attribution identified. The set of top genes identified by the 2 attribution methods were compared to each other. These 2 sets were also compared to a set made by choosing genes with the highest standard deviation. This was done to examine if gradient based attribution methods were any better at identifying informative genes than selection using statistical analysis.

4 Results and Discussion

As previously mentioned, the classifier designed by Lyu et al. had an overall accuracy of 95% on the entire data set and an accuracy of 92% on the test dataset. Figure 3 (under the column "Accuracy with All Features") shows per-class accuracy. The model has poorer accuracy on READ and CHOL samples, 64% and 73% respectively. However, this aligns with the results produced by Lyu et al.

Full Cancer Name	Cancer Type Abbrev	Number of Samples	Class Label	Accuracy - All Features	Accuarcy Top 20% Integrated Grads	Accuracy 20% Random
Adrenocortical carcinoma	ACC	79	0	0.975	1	1
Bladder urothelial carcinoma	BLCA	427	1	0.899	0.976	0.928
Breast invasive carcinoma	BRCA	1212	2	0.991	0.991	0.983
Cervical and endocervical cancers	CESC	309	3	0.926	0.709	0.354
Cholangiocarcinoma	CHOL	45	4	0.733	0	0
Colon adenocarcinoma	COAD	328	5	0.921	0.61	0.903
Lymphoid Neoplasm Diffuse Large B-cell Lymphoma	DLBC	48	6	1	1	1
Esophageal carcinom	ESCA	196	7	0.908	0.388	0.166
Glioblastoma multiforme	GBM	171	8	0.655	0.944	0.944
Head and Neck squamous cell carcinoma	HNSC	566	9	0.979	0.933	0.889
Kidney Chromophobe	KICH	91	10	0.802	0.538	0.923
Kidney renal clear cell carcinoma	KIRC	606	11	0.983	0.939	0.863
Kidney renal papillary cell carcinoma	KIRP	323	12	0.954	0.846	0.846
Acute Myeloid Leukemia	LAML	173	13	0.994	1	1
Brain Lower Grade Glioma	LGG	530	14	0.992	0.977	1
Liver hepatocellular carcinoma	LIHC	423	15	0.986	1	0.979
Lung adenocarcinoma	LUAD	576	16	0.962	0.927	0.945
Lung squamous cell carcinoma	LUSC	552	17	0.935	0.796	0.694
Mesothelioma	MESO	87	18	0.943	0.875	1
Ovarian serous cystadenocarcinoma	OV	307	19	0.827	0.962	0.962
Pancreatic adenocarcinoma	PAAD	183	20	0.995	0.827	0.9655
Pheochromocytoma and Paraganglioma	PCPG	187	21	1	1	1
Prostate adenocarcinoma	PRAD	550	22	1	0.983	1
Rectum adenocarcinoma	READ	105	23	0.638	0.3	0
Sarcoma	SARC	265	24	0.977	0.892	0.857
Skin Cutaneous Melanoma	SKCM	473	25	0.975	1	0.981
Stomach adenocarcinoma	STAD	450	26	0.967	0.948	0.974
Testicular Germ Cell Tumors	TGCT	156	27	0.981	0.812	0.937
Thyroid carcinoma	THCA	568	28	1	1	1
Thymoma	THYM	122	29	0.984	0.923	0.923
Uterine Corpus Endometrial Carcinoma	UCEC	201	30	0.995	0.772	0.909
Uterine Carcinosarcoma	UCS	57	31	0.93	0.666	0.666
Uveal Melanoma	UVM	80	32	0.725	1	1

Figure 3: The per-class accuracy of the model designed by Lyu et al. when trained with all features, the top 20% of features with the highest attribution assigned by integrated gradients, and 20% of features which were randomly selected. Values highlighted in red indicate noticeably worse accuracy than the model trained with all features. Purple values indicate higher accuracy. The full name, abbreviation, and number of samples for each of the 33 types of cancer in the dataset is also given as reference.

For each attribution method, the top 1, 2, 5, 10, 20, 50, 70, and 90 percent of features with the highest attribution were used to train the model. Figure 4 shows the accuracy on the test set for each of these trials. As mentioned earlier, the same process was repeated using random selection as a sanity check. Genes were also ordered by highest standard deviation to see how a purely statistical method (note this not an attribution method) compared. Random selection serves as a control while integrated gradients, vanilla gradients, and standard deviation can be considered experimental. Figure 5 shows the data in tabular form as the left hand side of the graph may be hard to read. When using 1 percent of the genes integrated gradients has the best performance at 85% followed by vanilla gradients and standard deviation both at 83% and finally the baseline at 79%. While these results make sense - the random selection trial should have the lowest percentage it must be noted that 78% is remarkably high considering only 103 features were randomly selected and there are 33 cancer types in total. This unexpectedly high accuracy is very questionable. Of the 3 experimental methods, as the percent of features used to train the model increases, integrated gradients generally remains the method with the best performance followed by vanilla gradients (with the noticeable exception at 20%) and finally standard deviation. It must be noted however that the difference in performance is quite marginal - just a few

percentage points so it's hard to tell if the slight increases and decreases in performance is noise or not. As for random selection, accuracy starts off very low but then spikes at 10% and 20% to be at or above that of the other 3 experimental methods before dropping. This is very unexpected. The accuracy of the random baseline should gradually increase as the percent of features used increases. It is possible that the sudden increase at 10% and 20% occurred by chance - perhaps very informative genes were chosen. However, these results are still suspicious and perhaps, next time, to get the accuracy for X% of features for the random selection method, multiple trials should be performed and the accuracies averaged.

Given that performance for the best performing method - integrated gradients - began to plateau at around 90% at when 20% of the top features were used, the per-class accuracy of this trial on the entire dataset was recorded and can be seen in Table 1. For comparison with the control group, the per-class accuracy of the model trained with 20% of genes, chosen randomly, was also reported in Table 1. The values highlighted in red indicate noticeably worse performance and those in purple indicate better performance in comparison to the model trained with all features. There are many red values indicating that although the accuracy of the models with 20% of the features was decent, accuracies for some of the classes suffered greatly. Most notably, the accuracy for COAD, ESCA, and READ dropped considerably when a smaller number of features was used. Interestingly, Lyu et al. also found and discussed the difficulties classifying COAD and READ [4]. COAD and READ affect parts of the body which are very close to each other (the colon and rectum respectively) so this could explain the general difficulty distinguishing between these 2 types of cancer. Interestingly, there were a few classes - GBM and UVM - where accuracy when the model was trained with less features was actually better than training with all features. Perhaps using less features eliminated noise in the data allowing for better classification.

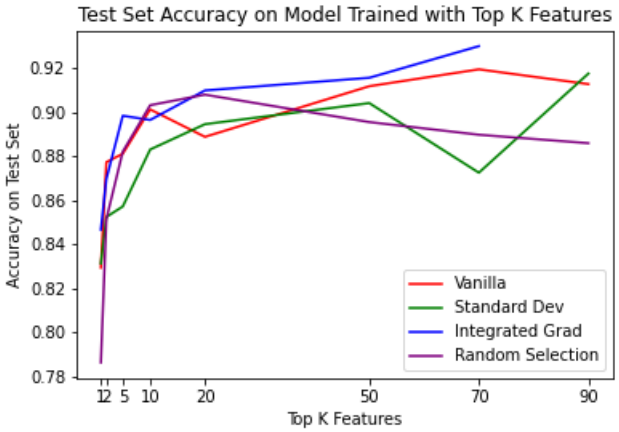

Figure 4: Test set accuracy on a model trained with the top 1, 2, 5, 10, 20, 50, 70, and 90 percent of features. The top genes for vanilla and integrated gradients were selected by choosing the genes with the highest attribution. The top genes for the standard deviation method were selected by choosing genes with the highest standard deviation. As a sanity check, the model was trained with a subset of features of equal size that were randomly chosen.

Percent of Genes	Vanilla Gradients	Integrated Gradient	Standard Deviation	Random Selection
1	83%	85%	83%	79%
2	88%	87%	85%	85%
5	88%	90%	86%	88%
10	90%	90%	88%	90%
20	89%	90%	89%	91%
50	91%	92%	90%	90%
70	92%	93%	87%	89%
90	91%		92%	88%

Figure 5: The data in Figure 4 in tabular form for ease of reading. Note, there is no accuracy for a model trained with the top 90% of genes identified by integrated gradients because GPU usage limits had been reached on Colab again!

The agreement between the 3 experimental methods - vanilla gradients, integrated gradients, and standard deviation was compared by calculating the percent difference between the top set of genes identified by each method. Figure 6 shows the percent difference for the set of top K percent of genes for all pairing of methods. Very shockingly, the percent difference for the top 1% of genes is very high - above 90% for all possible comparisons. As expected the percent difference gradually decreases, and eventually reaches 0, as the percentage of genes compared increases. The very high percent difference is extremely shocking because even when only 1% of the features are used to train a classifier, the accuracy is still considerable for all 3 experimental methods - between 83 and 85%. If there were no errors in the results, this would suggest that multiple sets of 100 or so genes (1% of the total number of genes) could be used to classify all 33 cancer types. This is very, very unlikely. The results for the comparison between vanilla and integrated gradient is also surprising since, given that they are both gradient based methods, they would be expected to produce similar attributions. Possible sources of error include buggy implementations of vanilla and integrated gradients. Sources of error and future improvements will be discussed in the next section.

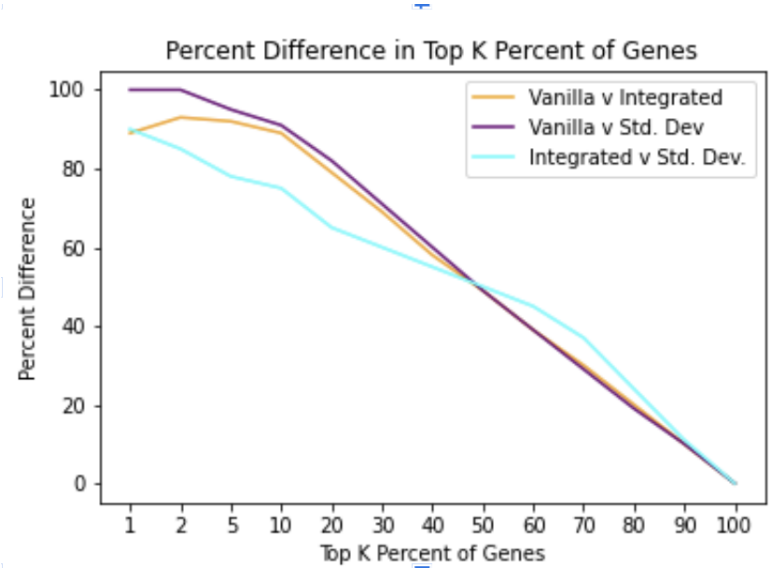

Figure 6: The percent difference between the set of top K percent of genes identified by vanilla gradient, integrated gradient, and standard deviation. All possible pairwise comparisons, 3 total, were made resulting in 3 separate lines.

Finally, the genes given the highest attribution for all breast cancer sample inputs were researched to see if they had any relation to cancer and the results are shown in Table 3. Breast cancer was chosen since it contains the most samples in the dataset and breast cancer has been widely studied. Given that the attribution for breast cancer inputs are being looked at, one would expect the genes with the highest attribution to be differentially expressed in breast cancer and the model uses information from that feature to classify the sample. Of the 3 genes identified by vanilla gradients, 2 were designated by The Protein Atlas [19] as having low cancer specificity meaning they were expressed at a fairly equal rate across all cancer types. One of the genes, GATA3 is however known to be enriched in breast cancer. This possibly indicates poor performance of vanilla gradients although more genes identified need to be examined to come to a better conclusion The 3 genes with the highest attribution assigned by integrated gradients are all different than the 3 identified by vanilla gradients. This is unexpected given that both are obviously gradient based methods. Of the 3 genes, 2 are linked to breast cancer while the other ANXA2P2 is actually linked to gliomas and hepatocellular carcinoma. However, it must be noted that ANXA2P2 is a pseudogene and does not have an entry in The Protein Atlas. However the gene which it originates from - ANXA2 - also shows low cancer specificity. Interestingly, the top 3 genes identified by integrated gradients for breast cancer are also the top 5 genes identified by integrated gradients for all samples.

Method	Gene Name	Notes From The Protein Atlas
Vanilla 1	LCP1	low cancer specificity
Vanilla 2	GATA3	enriched in breast and uroethelial cancer
Vanilla 3	ANXA6	low cancer specificity
Integrated 1	ASXL2	expressed at a higher rate in breast cancer
Integrated 2	RPS18	favorable prognosis marker for breast cancer
Integrated 3	ANXA2P2	linked to glioma and hepatocellular carcinoma

Figure 7: The top 3 genes with the highest average attribution assigned by vanilla gradient or integrated gradients for all breast cancer inputs. A brief description of the information about the gene and its relation to different cancer types and breast cancer is also provided based on information from The Protein Atlas [18]. *ANXA2P2 is a pseudogene with no entry in The Protein Atlas so this data was obtained from another published research article [19].

5 Conclusions and Future Work

As noted in the results and discussion section, lots of results were very surprising and very questionable. This was noticed as the project was being completed and the code was reviewed for errors and all experiments were rerun (a couple times actually). However the results did not significantly change and errors in the code may still be present. Getting a second pair of eyes on the code may help catch some bugs. Given that there is a surprisingly high percent difference in the top 100 genes identified by vanilla and integrated gradients it is possible that there are errors in their implementation or aggregation. This is particularly probable for integrated gradients. Although the implementation was guided by the source code accompanying the paper by Sund, the implementation was modified to be batched as computing gradients one input at a time was extremely, extremely slow.

Using Colab also created some challenges. The experiments used a lot of computational power - so much that I was prevented from connecting to GPU on 2 of my accounts (luckily I had a third!). However, this caused some file shuffling which was not fun (and possibly led to bugs).

6 Challenges and Non-Design Related Sources of Error

In this project, the cancer classifier by Lyu et. al. was reimplemented and vanilla gradients and integrated gradients were calculated. The ability of these two gradient based attribution methods were compared by training a model with only the top k% of features identified by each method. The same process was also performed with features with a higher standard deviation having greater weight. As a control, the same process was performed on randomly selected features. Although the model trained on the subset of features identified by integrated gradients generally performed the best, the differences in accuracy were marginal and simply randomly selecting a subset of features performed fairly well also. Thus the results are inconclusive and questionable. The percent difference between the top k percent of genes identified by vanilla gradients, integrated gradients and standard deviation was also recorded. These results were even more surprising and strongly indicate the possibility of error.

As briefly mentioned throughout the experimental design sections, there are several steps that may have caused error and could be improved. When computing integrated gradients using non-cancerous samples as a baseline and using more steps and random trials may produce better results.

Many extensions of this project are possible. Comparison with occlusion, a non-gradient based attribution method, was originally planned but due to time constraints was not completed. Using a library such as Captum.ai to perform occlusion or calculate integrated gradients or to verify the implementation used in the project would be useful as well. A more robust analysis of the top genes identified for each class and the entire data set could also be conducted. Specifically, more genes could be researched and pathway analysis could be performed to see if the genes with high attribution are part of a cancer-associated pathway. It is possible that genes could be very close towards the beginning of a pathway so that a simple Google search would not identify them as being related to a specific type of cancer.

7 Supporting Documents and Artifacts

Please see the GitHub repository belwo for information about how to access source code and other artifacts.

https://github.com/irenepham99/TCGA

8 References

1. Kulski, J. K. (2016). Next-Generation Sequencing — An Overview of the History, Tools, and "Omic" Applications. Next Generation Sequencing - Advances, Applications and Challenges. doi:10.5772/61964 Chang, K., Creighton, C., Davis, C. et al. The Cancer Genome Atlas Pan-Cancer analysis project. Nat Genet 45, 1113–1120 (2013). https://doi.org/10.1038/ng.2764

2. Mostavi, M., Chiu, Y., Huang, Y.et al.Convolutional Neural Network models for cancer type prediction based on gene expression. BMC Med Genomics 13, 44 (2020). https://doi.org/10.1186/s12920-020-0677-2

3. Lyu, B., Haque, A. (2018). Deep Learning Based Tumor Type Classification Using Gene Expression Data. doi:10.1101/364323 Strimbu K, Tavel JA. What are biomarkers? Curr Opin HIV AIDS. 2010 Nov;5(6):463-6. doi: 10.1097/COH.0b013e32833ed177. PMID: 20978388; PMCID: PMC307862

4. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

5. Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). "ImageNet classification with deep convolutional neural networks" (PDF). Communications of the ACM. 60 (6): 84–90. doi:10.1145/3065386. ISSN 0001-0782.

6. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

7. Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR, 2013.

8. Shrikumar, Avanti, Greenside, Peyton, Shcherbina, Anna, and Kundaje, Anshul. Not just a black box: Learning important features through propagating activation differences. CoRR, 2016.

9. Springenberg, Jost Tobias, Dosovitskiy, Alexey, Brox, Thomas, and Riedmiller, Martin A. Striving for sim- plicity: The all convolutional net. CoRR, 2014.

10. M. Ancona, E. Ceolini, A. C. Oztireli, and M. H. Gross, "A unified view of gradient-based attribution methods for deep neural networks," CoRR, vol. abs/1711.06104, 2017. [Online]. Available: http://arxiv.org/abs/1711.06104

11. M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," arXiv preprint arXiv:1703.01365, 2017.

12. Smilkov, Daniel, Thorat, Nikhil, Kim, Been, Viegas, Fernanda, Wattenberg, Martin Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825, 2017.

13. Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh,and Dhruv Batra. Grad-cam: Why did you say that? arXiv preprint arXiv:1611.07450, 2016.

14. Mengjiao Yang and Been Kim. 2019. Benchmarking Attribution Methods with Relative Feature Importance. Computing Research Repository, arXiv:1907.09701.

15. Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? ArXiv, abs/2004.03685.

16. Uhlén M, Fagerberg L, Hallström BM, Lindskog C, Oksvold P, Mardinoglu A, Sivertsson Å, Kampf C, Sjöstedt E, Asplund A, Olsson I, Edlund K, Lundberg E, Navani S, Szigyarto CA, Odeberg J, Djureinovic D, Takanen JO, Hober S, Alm T, Edqvist PH, Berling H, Tegel H, Mulder J, Rockberg J, Nilsson P, Schwenk JM, Hamsten M, von Feilitzen K, Forsberg M, Persson L, Johansson F, Zwahlen M, von Heijne G, Nielsen J, Pontén F. Proteomics. Tissue-based map of the human proteome. Science. 2015 Jan 23;347(6220):1260419. doi: 10.1126/science.1260419. PMID: 25613900.

17. Wang, Y., Liu, X., Guan, G., Xiao, Z., Zhao, W., Zhuang, M. (n.d.). Identification of a Five-Pseudogene Signature for Predicting Survival and Its ceRNA Network in Glioma. Frontiers in Oncology, 9, 1059. doi:10.3389/fonc.2019.01059