Interpreting how machine learning models make predictions in biological studies

Yongbing Zhao[1]*, Jinfeng Shao[2] and Yan W Asmann[1]*

1 Department of Quantitative Health Sciences, Mayo Clinic, Jacksonville, FL 32224, USA

2 The Laboratory of Malaria and Vector Research, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Rockville, MD, 20852, USA

* To whom correspondence should be addressed:

Yongbing Zhao: im@ybzhao.com

Yan W Asmann: asmann.yan@mayo.edu

ABSTRACT

Interpreting how the machine learning models make decisions is a new method to explore meaningful rules. However, it still lacks an understanding of the applicability of different model explainers in biological study. To address this question, we made a comprehensive evaluation on various explainers, and analyzed their performance and biological preference by quantifying the contribution of individual gene in the models trained to predict tissue type from transcriptome. Additionally, we also proposed a series of optimization strategies to improve the performance of different explainers. Interestingly, all explainers can be classified into three groups based on their outputs on different neural network architectures. With explainers from the group II, we found that the top contributing genes in different tissues exhibit tissue-specific manifestation and are potential biomarkers for cancer research. In summary, this work provides a novel insight and general guidance for exploring biological mechanisms by interpreting machine learning models.

INTRODUCTION

Recent years, many tools based on machine learning models have been developed and applied in biological studies, most of which are developed for predictions. For example, AlphaFold was developed to predict protein 3D structure from sequence profile [1], P-NET was used to predict cancer state from molecular data[2], CEFCIG predicted cell identity regulator from histone markers[3] and so on. Additionally, machine leaning models can predict different biological profiles from the same or similar kind of datasets, depending on which kind of "known" output is paired with the input data when training the model. For instance, a variety of models have been developed to predict ncRNA[4], nucleosome[5], chromatin accessibility, activity and states[6-9] from genome sequence.

Although these tools made great success in various topics, biologists are still curious about how a particular machine learning model makes decision, and which features of the input data play an important role in the model output? By far, there are two popular methods to estimate the contribution of each input feature to the model output: 1) Perturb the input data, and then compare outputs from original and perturbated inputs; 2) Use backpropagation to measure the importance of each feature in the input data [10-12]. The first method is intuitive but computationally inefficient especially when checking each input feature, and there is also the risk of underestimating feature contribution[13]. By contrast, the latter can measure the contribution of all input features with "one-shot", and many model explainers based on the idea of the latter were proposed and developed in the field of computer science [14]. Benefiting from these model explainers, computational biologists discovered the syntax of transcription factor (TF) binding motifs by interpreting models trained to predict chromatin accessibility [15, 16]; and screened cancer marker genes from models of cancer type classification[17-19]. There is no doubt that these explorations have witnessed the potential of interpretable models in discovering meaningful biological mechanisms. However, a noteworthy problem is that results from different model explainers are highly variable[17]. Since these model explainers are not specifically designed for biological data or studies, it is critical to evaluate their applicability in biology. By far, there is still lack of comprehensive understanding of these explainers in biological studies. To fill this gap, we optimized and assessed the performance of different model explainers and analyzed their biological preferences. To minimize the impact of model performance, we applied these explainers on well-trained models, predicting tissue type from

gene expression data, because these models have the highest predictive performance on the biological questions we have tested. In short, this study will provide a comprehensive guidance for applying interpretable machine learning in biological studies.

RESULTS

Overview of model interpretability

In this study, we formulated a specific question to instantiate application of interpretable models in biological study. Can we quantify the contributions of individual gene to tissue type and disease status? Two steps were implemented to estimate attributes with respect to each gene of the input sample. First, we built a neural network model, and then trained the model with transcriptomes and their tissue/cell type names or statuses. The model is trained to predict cell or tissue type from transcriptome. We built models based on two types of neural networks, convolutional neural network (CNN) and multiple layer perceptron (MLP) (Architectures are detailed in the Methods). Generally, CNN is more complex than MLP. Next, we incorporated model explainer to estimate how much each individual gene attributes the model's decision or prediction. In this step, model explainer computed quantitative scores with respect to each gene's attribution, which is named as gene contribution score. We implemented eight popular model explainers and their variants commonly used in computer vision, so that we can assess and compare their applicability and performances on the same pretrained model. These explainers include Saliency, InputXGradient, GuidedBackprop, IntegratedGradients, DeepLift, DeepLiftShap, GuidedGradCam, and GuidedGradCam++ (**Supplementary Table S1**) [13, 14, 20-25]. Since GuidedGradCam, and GuidedGradCam++ were developed for CNN specially, only the first six explainers were tested on MLP.

We used 27,417 RNA-Seq samples from GTEx and TCGA projects to train CNN or MLP-based models. These samples were isolated from 82 distinct normal and cancer tissues and cell types (**Supplementary Table S2**). After training, the prediction accuracy of all models is comparable, with a median value of 97.2% for CNN and 97.8% for MLP. Due to the restriction of convolutional layer, models based on CNN require that, as input data, gene expression scores should be organized with a fixed gene order in a matrix. Therefore, we considered different gene orders for the CNN-based models, including sorting genes as their genomic coordinates[18]. However, testing results indicate that gene order does not affect model performance in terms of prediction accuracy. To estimate applicability and performance of the eight explainers, samples from four out of 82 different tissues were used for testing, including normal liver, lung, ovary and pancreas.

Direct use of explainers from computer vision causes poor reproducibility

Randomness is often challenging in machine learning, which is present in both model training and model interpretability [26, 27]. For this reason, we measured both intra-model and inter-model reproducibility of each explainer. During the testing, each explainer was applied on pretrained models based on CNN or MLP separately.

Firstly, we applied each explainer on the same pretrained model 5 times independently, and then checked correlations of gene contribution scores among the 5 replicates, as well as overlap in the top 100 contributing genes between any two replicates. We found that reproducibility between any two replicates, in terms of Spearman's correlation on gene contribution scores and overlap in the top 100 contributing genes, are low in most explainers on both CNN and MLP-based models (**Figure 1, Supplementary Figure S1**), except GuidedGradCam++ which was used to identify cancer markers [17]. Next, we further applied each explainer on different pretrained models, which were trained based on the same model architecture, the same training data, similar hyperparameters, and have comparable prediction accuracy. We also checked the reproducibility of gene contribution scores calculated from different pretrained models. It was found that the reproducibility of all explainers is significantly reduced, including GuidedGradCam++. For CNN-based models, it is interesting that gene orders have little impact on prediction accuracy, but significant impact on the reproducibility of model interpretability. For MLP-based models, though there is no significant change on the reproducibility between intra-model and inter-model, both Spearman's correlation and overlaps in the top 100 contributing genes are very low.

Reproducibility tests show that gene contribution scores vary greatly across different runs on both CNN and MLP-based models. Moreover, these tests are based on the comparisons within the same explainer, so it is expected that the situation across explainers will be worse. Therefore, it suggests that model explainers from computer vision may not be directly applied in biological questions. One possible reason could be that biological data like transcriptome are focusing on a single gene, which requires higher resolution in model interpretability, so that result data is very sensitive to random noise.

Optimization on model interpretability

Given that it is not feasible to directly transfer model explainers from computer vision to biology, we asked whether these explainers can be optimized and adjusted for biological studies. For this purpose, we borrowed a de-noising strategy widely used in computer vison, "SmoothGrad:

removing noise by adding noise" [28]. Instead of estimating gene contribution scores in a sample with "one shot", SmoothGrad proposes to estimate multiple times on one sample while adding random noise into the expression data of estimated sample each time, and then average all results as the final gene contribution scores. We tested whether reproducibility between different pretrained models would benefit from SmoothGrad. Unfortunately, this optimization strategy does not improve reproducibility, instead it lowers the performance of all explainers running on both CNN and MLP, except Saliency on MLP and GuidedBackprop on CNN (**Figure 2A and Supplementary Figure S2)**. For Saliency on MLP, the improvement saturates when repeat number goes to 50, while performance of GuidedBackprop plateaus when the number increases to 30. Inspired by SmoothGrad, we asked whether it will be beneficial if we just simply repeat without adding random noise into the expression data. Results of the simulation indicate that simple repeat significantly improves performance of all explainers on both CNN and MLP, but GuidedGradCam and GuidedGradCam++ (**Figure 2B and Supplementary Figure S3)**. For most explainers, improvement saturates within 20 repeats on CNN, and the number increases to 40 on MLP.

DeepLift, DeepLiftShap and IntegratedGradients require a reference as baseline when estimating gene contribution scores, where the reference is an artificial transcriptome sample and is randomly generated. In computer vision, black image (all zeros as input) or random scores are widely used, while scrambled genomic sequence is demonstrated better in motif identification from regulatory elements [13]. In this study, we compared four kinds of references, named as reference zero, normal, universal, and specific. Reference zero and normal are equivalent to black image and random scores respectively. For reference universal and specific, we estimated mean (mu) and standard deviation (sigma) of each gene's expression level across samples, and then randomly generated a score based a truncated normal distribution N (mu, sigma). Reference universal uses samples from all 82 tissues, while reference specific only uses samples from a specific tissue or cell type. We tested performance of these four kinds of references individually, as well as whether using multiple references would improve reproducibility among models. Simulation results indicate that reference zero is predominant on CNN-based models, while universal is better than others on MLP-based models (**Supplementary Figure S4)**. Additionally, this result is consistent on all the three explainers. Next, we tested the impact of multiple references on reproducibility. Since using multiple references with zero is equivalent to simple repeat with one reference, these two kinds of optimization cannot contribute to reproducibility additively. Therefore, we compared

reproducibility by combing simple repeat with multiple references as reference normal, universal, or specific, with the reproducibility by combing simple repeat with single reference as zero (which is equivalent to multiple references as zero without simple repeat). Interestingly, we found that reference zero still outperforms the other three kinds of references on CNN-based models (**Supplementary Figure S5)**. Similar as simple repeat, improvement saturates when number of references as zero reach to 20 on CNN-based models, while the number of references as universal goes to 60 on MLP-based models (**Figure 2C**).

Considering that the reproducibility was significantly improved by repeating the interpreting step multiple times on the same model, we wondered whether it would be beneficial to aggregate outputs from different models. Therefore, we applied optimal conditions of each explainer on CNN or MLP-based models (**Supplementary Table S2**), estimated gene contribution scores on each pretrained model individually, and then averaged results from a certain number of models. Results show that aggregating models can significantly increase reproducibility among different replicates (**Figure 2D, and Supplementary Figure S6**). Especially, Spearman's correlations for DeepLift, DeepLiftShap and IntegratedGradients nearly reach to 1.0 on MLP. By and large, the reproducibility of all explainers is significantly increased on both CNN and MLP-based models after aggregating models (**Figure 2E-F, and Supplementary Figure S7**). Of note is that model aggregation has extremely strong impact on the reproducibility of all explainers on CNN-based models in terms of overlap in top 100 contributing genes. For most explainers on MLP-based models, Spearman's correlations on gene contribution scores are higher than 0.9, and over 90% of top 100 contributing genes overlap between replicates on the same explainer.

In short, gene contribution scores are highly reproducible on the same explainer with specific optimized conditions. Reproducibility of the top 100 contributing genes is better on MLP-based models than those on CNN-based models. One possible reason is that CNN-based models is much more complicated than MLP-based models and is hard to be interpreted.

Consistency across model explainers

To test consistency of gene contribution scores across explainers, we extracted the top 100 contributing genes identified by different explainers with and without model aggregation respectively, and then checked their overlaps (**Supplementary Table S3**). For both CNN and MLP-based models, model aggregation does not only improve reproducibility within the same explainer, but also consistency across explainers. However, the top 100 contributing genes from

CNN-based models with model aggregation does not overlap with those genes on MLP-based models with or without model aggregation. Moreover, within CNN-based models, the top 100 contributing genes with model aggregation does not overlap with those without model aggregation either, which suggests that model aggregation makes these explainers to highlight completely different genes. By contrast, top contributing genes on MLP-based models are highly consistent between with and without model aggregation.

Intriguingly, the measurement of reproducibility within the same explainer and across explainers highlights three representative groups from the combinations composed of explainers, model types (CNN or MLP), and optimization approaches (with or without model aggregation) (**Figure 3**). The three groups are group I: DeepLift, DeepLiftShap, GuidedBackprop, InputXGradient, and IntegratedGradients on CNN-based models with model aggregation; group II: DeepLift, DeepLiftShap, InputXGradient, and IntegratedGradients on MLP-based models with model aggregation; and group III: GuidedBackprop and Saliency on MLP-based models with model aggregation.

Expression status of top contributing genes

Given that explainers on three representative groups emphasize distinctive genes, it is important to explore the biological relevance of the top contributing genes. Since contribution scores were derived from gene expression, we analyzed Spearman's correlation between gene contribution scores and their expression level (**Supplementary Figure S8**). It is expected that there will be high correlation in InputXGradient, because gene expression level is a cofactor used to compute contribution score. There is weak correlation in all explainers in both group II and group III, except InputXGradient. Conversely, there is very strong correlation in group I. However, it is incomprehensible that there is weak negative correlation in GuidedBackprop on CNN-based models, and model aggregation strengthens the negative correlation.

Additionally, we also checked overlaps between the top 100 contributing genes and the top 100 expressed genes for all explainers on both CNN and MLP-based models (**Supplementary Figure S9**). In liver, nearly 50% of the top contributing genes overlap with top expressed genes in those explainers of group II, while the numbers are less than 10% in both group I and group III (**Figure 4A**). Strikingly, model aggregation eliminates overlaps when the top 100 contributing genes were calculated on CNN-based models. Another noticeable finding is that though

Spearman's correlation between gene contribution scores and expression level are extremely high in group I, majority of the top contributing genes are not highly expressed in this group.

Since different tissues have a unique phenotype, we wondered whether the top contributing genes from different tissues will also exhibit distinct expression profiles. Heatmap analysis shows that there are clear tissue-specific manifestations in group II (shown as DeepLift, MLP with model aggregation), and the patterns are very weak in both group I (shown as DeepLift, CNN with model aggregation) and group III (shown as Saliency, MLP with model aggregation) (**Figure 4B**). In addition, the total gene number in group III is much lower than those in group I and II, after removing redundant genes from the top 100 contributing genes across tissues. This suggests that the top 100 contributing genes are mostly shared across tissues in group III, which was also validated by comparison across tissues in all explainers (**Supplementary Table S4**). Among the three groups, the top contributing genes in both group I and II are tissue specific, and genes in group III are highly shared across tissues (**Figure 4C**).

Considering that the top contributing genes in group I and II are mostly tissue specific, we are curious how the top contributing genes are related with tissue-specifically (TS) expressed genes. For this propose, we identified TS genes across 82 tissues and cell types used in model training. It was found that about 70% of the top contributing genes overlap with TS genes in group II in liver (**Figure 4D**). The fractions vary across tissues (**Supplementary Figure S10A**), since there are different numbers of TS genes in each tissue type (**Supplementary Figure S10B**). The percentages drop to less than 10% in both group I and group III. Interestingly, model aggregation also diminishes overlaps with TS genes in most explainers on CNN-based models.

In addition, since many of the top contributing genes in group I are expressed at comparable level across tissues, we asked whether the top contributing genes are related to housekeeping (HK) genes. We also identified all HK genes across 82 tissue and cell types. Comparison results show about 10% of top contributing genes overlap with HK genes in group I, in which the number was also reduced by model aggregation (**Figure 4E, and Supplementary Figure S11**). Conversely, no overlap was found in both group II and group III, except InputXGradient.

Enrichment of top contributing genes on biological functions

To further explore function of top contributing genes, we performed Gene Ontology (GO) enrichment analysis with these genes. Results show that no enrichment was found on genes

identified by all explainers in group I. Enrichment of genes identified by group II are mostly also tissue specific (**Supplementary Table S5**). For example, genes in liver are enriched in molecular function as lipoprotein and lipoprotein lipase related activities, while genes in pancreas are enriched in binding of oligosaccharide, peptidoglycan and so on. Additionally, in group II, results within DeepLift, DeepLiftShap and IntegratedGradients are slightly more consistent than InputXGradient. It is expected that enrichment of genes from group III are similar across tissues, since top contributing genes are highly overlapped. GO enrichment analysis shows that top contributing genes in group III are enriched in CCR7 chemokine receptor binding, neuropeptide hormone activity, neuropeptide receptor binding, and DNA-binding transcription activator activity. Subsequently, we checked how top contributing genes are related with transcription factors (TFs) and TF cofactors. It was found that there are about 20 genes that overlap with TFs in group III, which is more than 2-fold enrichment than random baseline (**Figure 4F**). By contrast, genes in group II show depletion in TFs in liver, but 1.5-fold enrichment in Ovary (**Supplementary Figure S12**). No enrichment or depletion was found in group I, except GuidedGradCam++. As for TF cofactors, there are low overlap in all three groups (**Supplementary Figure S13**).

Top contributing genes in cancers

From group II, we found that top contributing genes are tissue specific, and their expression level also exhibit tissue-specific manifestations. Therefore, we asked how the expression pattern of the top contributing genes changes from normal to cancer tissues. To address this question, we matched normal and cancer samples for liver, lung, ovary and pancreas from GTEx and TCGA data, and compared expression level of top contributing genes between normal and cancer tissues. It was found that about 40% to 80% of the top contributing genes are differentially expressed genes between normal and cancer tissues in DeepLift on MLP (group II), which is about 2 times fold over random baseline (**Figure 4G**). The fractions range from 30% to 60% in Saliency on MLP (group III), about 1.5 times fold over random baseline. Group I was not included for the analysis, since previous analyses show that model aggregation eliminates many features common in explainers on CNN-based models without model aggregation and explainers from group II and III, and no biological enrichment was found in the top contributing genes identified by group I.

Interestingly, differentially expressed top contributing genes are segregated into two distinct populations in group II (**Figure 4H, and Supplementary Figure S14**). Specifically, the top

contributing genes specific to normal tissues are downregulated in cancer, while those specific to cancer are upregulated in cancer. For example, Glypican-3 (*GPC3*), a member of heparan sulfate proteoglycans family, is one of top contributing genes in liver cancer but not in normal liver. *GPC3* is often observed to be highly elevated in hepatocellular carcinoma and is a target for diagnosis and treatment of hepatocellular carcinoma [29]. However, similar pattern was not found in Saliency on MLP-based models, because top contributing genes are mostly shared in both normal and cancer tissues. Together, the expression profiles suggest that the top contributing genes in group II will be potential marker genes in cancer research.

CONCLUSIONS

The beauty of interpreting machine learning model is that it converts the complex mathematical rules learned by neural networks into biological rules, which provides new insights to explore biological questions. In this study, we illustrated model interpretability in biological studies with an example of interpreting models trained to predict tissue types and status based on transcriptome. To facilitate application of interpretable machine learning model, we proposed a series of optimization strategies and demonstrated the biological preference of different model explainers. We believe this guidance will bring broad applications of interpretable machine learning models in biological studies.

Typically, complicated models are not easily interpreted[30], which is also witnessed by the poor performance on interpreting CNN-based models. Therefore, it will be better to use a simple neural networks architecture with comparable performance in prediction. The top contributing genes detected by explainers in group II exhibits tissue-specific manifestation in both gene sets and gene expression profile, which is consistent with prior to knowledges about tissue specificity and cell identity[31-34]. From this perspective, explainers in group II are more suitable for biological study, especially when exploring biological questions based on transcriptomic data. Recent years, single-cell RNA-Seq technique has been widely applied to different tissue types and diseases, leading to many well-defined sub-populations in each tissue [35, 36]. Although this study assessed model interpretability on transcriptome from bulk RNA-Seq, the optimization strategies proposed here can be also applied to single-cell transcriptome to quantify “individual gene contribution” and explore important genes in each sub-population. It is expected that interpretable machine learning models will also benefit understandings on tissue specificity and heterogeneity, disease mechanisms and cellular engineering at single-cell resolution.

METHODS

Human transcriptome collection and processing

In total, 27,417 RNA-Seq samples are used in our study, among which 17,329 and 10,088 RNA-Seq data were collected from GTEx and TCGA projects respectively[37]. These samples are from 47 distinct primary normal tissues and 2 cell lines (with prefix GTEx_ in the tissue code) and 33 different primary cancer tissues (with prefix TCGA_ in the tissue code), respectively. All pre-processed TCGA and GTEx RNA-Seq data were downloaded from GTEx Portal (phs000424.v8.p2) and Recount2 database[38] respectively. For TCGA data, only primary tumor samples were used for further analysis. Tissue type names remain the same as TCGA and GTEx Projects. In each sample, the expression level scores of 19,241 protein-coding genes were normalized as log2(TPM + 1) and then imported for further analysis.

Convolutional neural networks (CNN) model

We used a five-layer convolutional neural network to build a CNN model, which includes three convolutional layers, one global average pooling layer and one fully connected layer sequentially. Each layer includes 64, 128, 256, 256 and 82 channels respectively. The kernel sizes for the three convolutional layers are 5, 5, and 3 respectively, and each convolutional layer is followed by max pooling with kernel size of 2. Batch normalization and ReLU are applied immediately after max pooling in each convolutional layer and global average pooling layer.

As the input of the CNN model, normalized expression scores of 19,241 genes were transformed into a 144X144 matrix, and zero-padding was used at the bottom of the matrix. The final fully connected layer comes with an 82-dimension output, which matches to 82 different tissue types.

Multilayer perceptron (MLP) model

There is only one hidden layer in the MLP model, and 128 units are used in the hidden layer. Batch normalization and rectified linear unit activation function (ReLU, which can be presented as $f(x) = \max(0, x)$) are applied immediately after the hidden layer. There are 19,241 variables in the input layer, which matches the number of genes used for the analyses. The output layer assigns a phenotype probability-like score for each of 82 tissue types.

Model training

All samples in a tissue type were randomly partitioned as 9:1 ratio, with 90% of samples used as training data and the remaining 10% treated as testing data. In each epoch, up-sampling was employed to avoid imbalance caused by different sample numbers per tissue types. Adam optimizer on cross entropy loss was utilized to update the weights of the neural network. After hyperparameter optimization, an initial learning rate of 0.0006 was used for CNN model, and 0.001 was used for MLP model. Batch size of 256 was used for both CNN and MLP. If there is no improvement for 5 sequential epochs, the learning rate will be reduced by 0.25. L2 regularization was applied with a lambda score of 0.001. A fixed dropout of 0.25 was applied before the output layer in the MLP model, while dropout of 0.25 was applied before the global average pooling layer in the CNN model.

To check and optimize reproducibility in model explanation, we selected 60 well-trained models based on slightly different parameters but similar performance. In the CNN model, genes are organized in to 2-D matrix with a fixed order as input. In this study, gene order in the CNN model was also considered. With the same gene order, we selected 5 well well-trained models based on slightly different parameter but similar performance. With different gene orders, we selected 60 well well-trained models based on slightly different parameter but similar performance.

Estimate model performance

5-fold cross-validation was used to estimate the model performance for both MLP and CNN. 5 groups of datasets were prepared for further training and evaluation, and each group of datasets includes training dataset and test dataset. Dataset preparation for each group is as follows. In the beginning, we randomly split all samples in each tissue into 5 parts. For the 1st group: combine the 1st part of samples in each tissue as test datasets and all the remaining parts of samples in each tissue as training dataset. For the 2nd group: combine the 2nd part of samples in each tissue as test datasets and all the remaining parts of samples in each tissue as training dataset. The same strategy was also applied to the other groups. The same hyperparameters were used to train models based on the training dataset of different groups separately. Pretrained models were then used to estimate the corresponding test dataset. Estimated results from different groups were combined, in which the number of combined samples is equal to the total sample. Lastly, all metrics about performance were calculated based on the combined estimated results.

Model explanation

To estimate how much each gene contributes to the model prediction, we used eight different model explainers and variants, which are DeepLift, DeepLiftShap, GuidedBackprop, GuidedGradCam, GuidedGradCam++, InputXGradient, IntegratedGradients, and Saliency. All these explainers were implemented based on Captum package (https://github.com/pytorch/captum). As output, each explainer estimates contribution scores with respect to each of the 19,241genes.

Reference preparation

In this study, we tested four kinds of references, which are named as zero, normal, universal and specific. 1) For zero, we assigned expression level of each gene to 0. 2) For norm, expression level of each gene was randomly generated from a truncated normal distribution ***N*** (0,1), and all values are restricted between 0 and 1. 3) For universal, expression level of each gene was randomly generated from a truncated normal distribution ***N*** (μ, σ), and all values are restricted between 0 and σ. μ and σ were calculated based on expression values of a specific gene across all samples from all tissues, and σ is standard deviation. 4) For specific, each tissue uses their tissue-specific references. The strategy to generate specific reference is similar to universal reference. Instead of using all samples from all tissues, only samples from a specific tissue were used to estimate μ and σ. In the reference testing, 2,000 different references were generated for normal, universal and specific separately. For zero, we just repeated the same reference 2,000 times.

Simulation for optimal number of pseudo-samples generated for each sample

The simulation process was performed on 5 different pretrained models as follows, and sample X will be taken as an example. Step 1: we generated 50 pseudo-samples based on sample X by randomly adding noise to each gene's expression level with normal distribution ***N*** (0,1). Step 2: we estimated gene contribution scores for all genes in each pseudo-sample. To estimate gene contribution scores based on n pseudo-samples, we randomly selected n replicates out of 50, and the final gene contribution score for a specific gene was calculated based on mean of n scores. Step 3: repeat step 1) and 2) on 5 different pretrained models respectively. Step 4: For sample X, there will be 5 replicates of gene contribution scores based on the same number of pseudo-samples. For the 5 replicates based on n (n = 1, 2, …, 100) pseudo-samples, we calculated Spearman's Correlation coefficient on gene contribution scores from any two replicates, and this operation was carried out on all C_5^2 combinations. Based on the above

method, we obtained the relationship between number of pseudo-samples and Correlation coefficient on any two replicates.

Simulation for optimal repeat number on the same model

The simulation process was performed on 5 different pretrained models as follows, and sample X will be taken as an example. Step 1: First, we estimated gene contribution scores for all genes in sample X 50 times respectively, and there were 50 replicates for sample X. To estimate gene contribution scores by n times repeats, we randomly selected n replicates out of 50, and the final gene contribution score for a specific gene was calculated based on mean of n scores. Step 2: repeat step 1) on 5 different pretrained models respectively. Step 3: For sample X, there will be 5 replicates of gene contribution scores based on the same repeat number. For the 5 replicates based on n (n = 1, 2, …, 100) times repeats, we calculated Spearman's Correlation coefficient on gene contribution scores from any two replicates, and this operation was carried out on all C_5^2 combinations. Based on the above method, we obtained the relationship between repeat number and Correlation coefficient on any two replicates.

Simulation for optimal number of references

The simulation process was performed on 5 different pretrained models as follows, and sample X will be taken as an example. Step 1: we estimated gene contribution scores for all genes in sample X with 1, 2, 3 ... 100 reference samples respectively, and the reference samples were randomly selected from the 2,000 background samples pool. Step 2: repeat step 1) on 5 different pretrained models. Step 3: For sample X, there will be 5 replicates of gene contribution scores based on the same number of reference samples but different pretrained models. For the 5 replicates based on n (n = 1, 2, …, 100) reference samples, we calculated Spearman's Correlation coefficient on gene contribution scores from any two replicates, and this operation was carried out on all C_5^2 combinations. Based on the above method, we obtained the relationship between number of reference samples and Correlation coefficient on any two replicates. For each type of reference, we repeated the above simulation process individually.

Simulation for optimal number of aggregated models

The simulation process was performed on 60 different pretrained models as follows, and sample X will be taken as an example. Step 1: we estimated gene contribution scores for all genes in sample X on each pretrained models respectively. Step 2: to estimate gene contribution scores by aggregating n (n = 1, 2, …, 20) models, we randomly selected n replicates out of 60, and the

final gene contribution score for a specific gene was calculated based on mean of n scores. Step 3: repeat step 2) K times, where $K = \max(\frac{60}{n}, 4)$. Step 4: For sample X, there will be K replicates of gene contribution scores based on the same number of aggregated models. For these K replicates based on n (n = 1, 2, …, 20) aggregated models, we calculated Spearman's Correlation coefficient on gene contribution scores from any two replicates, and this operation was carried out on all C_K^2 combinations. Based on the above method, we obtained the relationship between repeat number and Correlation coefficient on any two replicates.

Gene classification

Tissue-specific (TS) expressed genes were identified by the tool TissueEnrich with group "Tissue-Enhanced" [39]. In each tissue, median expression level of each gene was calculated across all samples, and housekeeping genes are defined as genes with TPM ≥1 and less than 2-fold change on median expression level among all tissue types [40].

Gene Ontology enrichment analysis

Genes of interest were extracted and imported into the Gene Ontology online tool for GO enrichment analysis with the options "molecular function" or "biological process" and "Homo sapiens" checked. [41, 42].

Annotation of transcription factor (TF) and TF cofactors

All TFs and TF cofactors were downloaded from animalTFDB [43]. In total, there were 1,666 TFs and 1,026 TF cofactors.

Differentially expressed genes between normal and cancer

Mann–Whitney U test (two-sided) was used to compare gene expression between normal and cancer tissues. Differentially expressed genes should satisfy the following criteria: FDR ≤ 0.001 and fold change ≥ 3.

REFERENCES

1. Jumper, J., et al., *Highly accurate protein structure prediction with AlphaFold.* Nature, 2021. **596**(7873): p. 583-589.
2. Elmarakeby, H.A., et al., *Biologically informed deep neural network for prostate cancer discovery.* Nature, 2021. **598**(7880): p. 348-352.
3. Xia, B., et al., *Machine learning uncovers cell identity regulator by histone code.* Nat Commun, 2020. **11**(1): p. 2696.
4. Chantsalnyam, T., et al., *ncRDeep: Non-coding RNA classification with convolutional neural network.* Comput Biol Chem, 2020. **88**: p. 107364.
5. Zhang, J., W. Peng, and L. Wang, *LeNup: learning nucleosome positioning from DNA sequences with improved convolutional neural networks.* Bioinformatics, 2018. **34**(10): p. 1705-1712.
6. Nair, S., et al., *Integrating regulatory DNA sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts.* Bioinformatics, 2019. **35**(14): p. i108-i116.
7. Kelley, D.R., et al., *Sequential regulatory activity prediction across chromosomes with convolutional neural networks.* Genome Res, 2018. **28**(5): p. 739-750.
8. Kelley, D.R., J. Snoek, and J.L. Rinn, *Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks.* Genome Res, 2016. **26**(7): p. 990-9.
9. Avsec, Z., et al., *Effective gene expression prediction from sequence by integrating long-range interactions.* Nat Methods, 2021. **18**(10): p. 1196-1203.
10. Talukder, A., et al., *Interpretation of deep learning in genomics and epigenomics.* Briefings in Bioinformatics, 2020.
11. Zhou, J. and O.G. Troyanskaya, *Predicting effects of noncoding variants with deep learning-based sequence model.* Nat Methods, 2015. **12**(10): p. 931-4.
12. Torng, W. and R.B. Altman, *3D deep convolutional neural networks for amino acid environment similarity analysis.* BMC Bioinformatics, 2017. **18**(1): p. 302.
13. Shrikumar, A., P. Greenside, and A. Kundaje. *Learning important features through propagating activation differences*. in *International Conference on Machine Learning*. 2017. PMLR.
14. Lundberg, S.M. and S.-I. Lee. *A unified approach to interpreting model predictions*. in *Advances in Neural Information Processing Systems*. 2017.
15. Avsec, Z., et al., *Base-resolution models of transcription-factor binding reveal soft motif syntax.* Nat Genet, 2021.
16. Kim, D.S., et al., *The dynamic, combinatorial cis-regulatory lexicon of epidermal differentiation.* Nature Genetics, 2021.
17. Karim, M., et al., *OncoNetExplainer: Explainable Predictions of Cancer Types Based on Gene Expression Data.* arXiv preprint arXiv:1909.04169, 2019.
18. Lyu, B. and A. Haque. *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*. 2018. ACM.
19. Li, Y., et al., *A comprehensive genomic pan-cancer classification using The Cancer Genome Atlas gene expression data.* BMC Genomics, 2017. **18**(1): p. 508.
20. Simonyan, K., A. Vedaldi, and A. Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps.* arXiv preprint arXiv:1312.6034, 2013.
21. Shrikumar, A., et al., *Not just a black box: Learning important features through propagating activation differences.* arXiv preprint arXiv:1605.01713, 2016.

22. Springenberg, J.T., et al., *Striving for simplicity: The all convolutional net.* arXiv preprint arXiv:1412.6806, 2014.
23. Sundararajan, M., A. Taly, and Q. Yan. *Axiomatic attribution for deep networks*. in *International Conference on Machine Learning*. 2017. PMLR.
24. Selvaraju, R.R., et al. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. in *Proceedings of the IEEE International Conference on Computer Vision*. 2017.
25. Chattopadhay, A., et al. *Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks*. in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018. IEEE.
26. Hartley, M. and T.S.G. Olsson, *dtoolAI: Reproducibility for Deep Learning.* Patterns (N Y), 2020. **1**(5): p. 100073.
27. Fan, F., et al., *On Interpretability of Artificial Neural Networks: A Survey.* arXiv preprint arXiv:2001.02522, 2020.
28. Smilkov, D., et al., *Smoothgrad: removing noise by adding noise.* arXiv preprint arXiv:1706.03825, 2017.
29. Guo, M., et al., *Glypican-3: A New Target for Diagnosis and Treatment of Hepatocellular Carcinoma.* J Cancer, 2020. **11**(8): p. 2008-2021.
30. Carvalho, D.V., E.M. Pereira, and J.S. Cardoso, *Machine learning interpretability: A survey on methods and metrics.* Electronics, 2019. **8**(8): p. 832.
31. Toyoda, M., et al., *Defining cell identity by comprehensive gene expression profiling.* Curr Med Chem, 2010. **17**(28): p. 3245-52.
32. Ye, Z. and C.A. Sarkar, *Towards a Quantitative Understanding of Cell Identity.* Trends Cell Biol, 2018. **28**(12): p. 1030-1048.
33. Sonawane, A.R., et al., *Understanding Tissue-Specific Gene Regulation.* Cell Rep, 2017. **21**(4): p. 1077-1088.
34. Uhlen, M., et al., *Proteomics. Tissue-based map of the human proteome.* Science, 2015. **347**(6220): p. 1260419.
35. Morris, S.A., *The evolving concept of cell identity in the single cell era.* Development, 2019. **146**(12).
36. Stuart, T. and R. Satija, *Integrative single-cell analysis.* Nat Rev Genet, 2019. **20**(5): p. 257-272.
37. Consortium, G.T., *The Genotype-Tissue Expression (GTEx) project.* Nat Genet, 2013. **45**(6): p. 580-5.
38. Collado-Torres, L., et al., *Reproducible RNA-seq analysis using recount2.* Nat Biotechnol, 2017. **35**(4): p. 319-321.
39. Jain, A. and G. Tuteja, *TissueEnrich: Tissue-specific gene enrichment analysis.* Bioinformatics, 2019. **35**(11): p. 1966-1967.
40. Eisenberg, E. and E.Y. Levanon, *Human housekeeping genes, revisited.* Trends Genet, 2013. **29**(10): p. 569-74.
41. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.* Nat Genet, 2000. **25**(1): p. 25-9.
42. The Gene Ontology, C., *The Gene Ontology Resource: 20 years and still GOing strong.* Nucleic Acids Res, 2019. **47**(D1): p. D330-D338.
43. Hu, H., et al., *AnimalTFDB 3.0: a comprehensive resource for annotation and prediction of animal transcription factors.* Nucleic Acids Res, 2019. **47**(D1): p. D33-D38.

FIGURE LEGEND

Figure 1: **Performance of different model explainers without optimization**

A) Spearman's correlation on gene contribution scores (top) and overlap in the top 100 contributing genes (bottom) in liver among replicates from the same pretrained model, different pretrained models with the same gene order, and different pretrained models with different gene orders on CNN-based models. **B**) Spearman's correlation on gene contribution scores (top) and overlap in the top 100 contributing genes (bottom) in liver among replicates from the same pretrained model and different pretrained models on MLP-based models.

Figure 2: **Optimization on different model explainers. A-D**) Spearman's correlation on gene contribution scores in liver among replicates from different pretrained models with different gene orders on CNN-based models (top) and from different pretrained models on MLP-based models (bottom). **A**) performance of multiple repeats with adding noise; **B**) performance of simple repeat; **C**) performance of multiple references, reference zero and reference universal were applied on CNN and MLP-based models respectively; **D**) performance of model aggregation. **E**) Spearman's correlation on gene contribution scores (top) and overlap in the top 100 contributing genes (bottom) among replicates generated from different pretrained models with different gene orders based on CNN-based models. The analyses were carried out with three different optimization strategies respectively: without optimization, with optimized conditions for each explainer but without model aggregation, and with optimized conditions for each explainer and with model aggregation. **F**) same as e) but based on MLP-based models.

Figure 3: **Overlaps in the top 100 contributing genes across explainers with and without model aggregation**. Three representative groups are marked by black bars.

Figure 4: Biological relevance and expression profiles of top 100 contributing genes. **A**) Overlaps between the top 100 contributing genes and the top 100 expressed genes in liver samples. Explainers from group I, II and III are marked by with black bars. **B**) Expression profiles of the top 100 contributing genes in liver, lung, ovary and pancreas identified by DeepLift on CNN-based models with model aggregation (representative of Group I), DeepLift on MLP-based models with model aggregation (Group II), and Saliency on MLP-based models with model aggregation (Group III). **C**) Overlaps in the top 100 contributing genes among liver, lung, ovary and pancreas identified by DeepLift on CNN-based models with model aggregation (Group I), DeepLift on MLP-based models with model aggregation (Group II), and Saliency on

MLP-based models with model aggregation (Group III). **D**) Overlaps between top 100 contributing genes and tissue-specifically (TS) expressed genes in liver samples. **E**) Overlaps between the top 100 contributing genes and housekeeping (HK) genes in liver samples. **F**) Percentage of the top 100 contributing genes are transcription factors (TFs) in liver samples (left) and ovary samples(right). Dashed lines are random baseline for enrichment analysis. **G**) Percentages of the top 100 contributing genes are differentially expressed between normal and cancer tissues. Dashed lines are random baseline for enrichment analysis. **H**) Expression level of different types of top 100 contributing genes between normal and cancer ovary tissues, including top genes only in cancer, only in normal and in both.

Figure 1

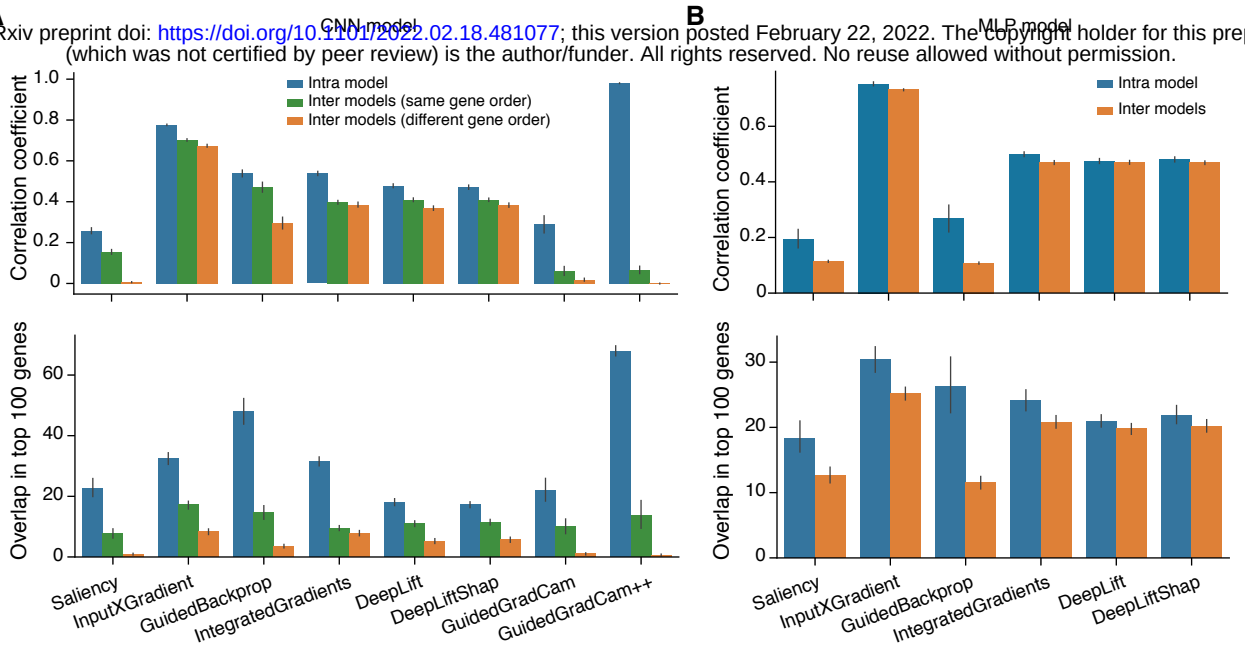


Figure 2

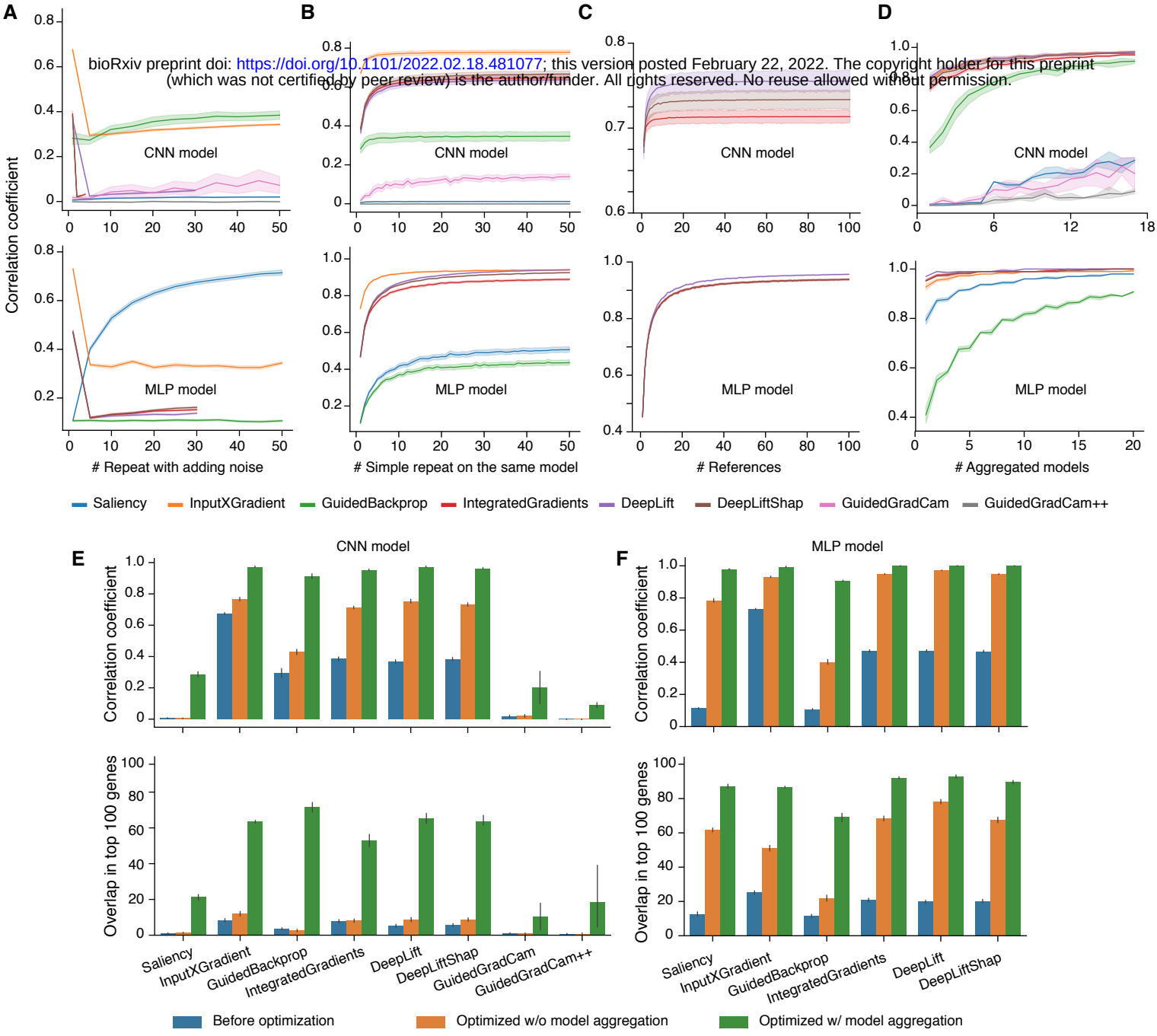
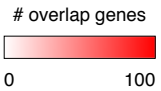


Figure 3



	CNN,DeepLift	CNN,DeepLiftShap	CNN,GuidedBackprop	CNN,InputXGradient	CNN,IntegratedGradients	CNN,Saliency	CNN,GuidedGradCam	CNN,GuidedGradCam++	CNN-Agg,DeepLift	CNN-Agg,DeepLiftShap	CNN-Agg,GuidedBackprop	CNN-Agg,InputXGradient	CNN-Agg,IntegratedGradients	CNN-Agg,Saliency	CNN-Agg,GuidedGradCam	CNN-Agg,GuidedGradCam++	MLP,DeepLift	MLP,DeepLiftShap	MLP,IntegratedGradients	MLP,InputXGradient	MLP,GuidedBackprop	MLP,Saliency	MLP-Agg,DeepLift	MLP-Agg,DeepLiftShap	MLP-Agg,IntegratedGradients	MLP-Agg,InputXGradient	MLP-Agg,GuidedBackprop	MLP-Agg,Saliency
									Group I														**Group II**				**Group III**	
CNN,DeepLift	16	14	8	15	14	4	4	4																				
CNN,DeepLiftShap	14	13	7	14	12	3	4	4																				
CNN,GuidedBackprop	8	7	5	8	7	2	2	3																				
CNN,InputXGradient	15	14	8	15	14	4	4	4																				
CNN,IntegratedGradients	14	12	7	14	12	3	4	4																				
CNN,Saliency	4	3	2	4	3	1	1	2																				
CNN,GuidedGradCam	4	4	2	4	4	1	1	2																				
CNN,GuidedGradCam++	4	4	3	4	4	2	2	2																				
CNN-Agg,DeepLift	1	1	1	1	1	1	1	1	71	72	69	66	62	27	14	5												
CNN-Agg,DeepLiftShap	1	1	1	1	1	0	0	1	72	73	71	64	61	27	14	6												
CNN-Agg,GuidedBackprop	1	1	1	1	1	1	1	1	69	71	78	64	61	27	13	6												
CNN-Agg,InputXGradient	1	1	1	1	1	1	1	1	66	64	64	68	59	29	14	6												
CNN-Agg,IntegratedGradients	1	1	1	1	1	1	0	1	62	61	61	59	57	27	12	6												
CNN-Agg,Saliency	1	1	1	1	1	1	1	1	27	27	27	29	27	25	7	6												
CNN-Agg,GuidedGradCam	1	1	1	1	1	1	1	1	14	14	13	14	12	7	9	2												
CNN-Agg,GuidedGradCam++	0	0	1	0	0	0	1	1	5	6	6	6	6	6	2	16												
MLP,DeepLift	17	15	7	15	14	3	3	4	1	0	2	2	2	1	1	0	77	70	70	44	11	15						
MLP,DeepLiftShap	16	14	6	15	14	3	3	4	1	1	2	2	2	1	1	0	70	66	65	42	11	14						
MLP,IntegratedGradients	16	14	6	15	14	3	3	4	1	0	2	2	2	1	1	0	70	65	66	42	11	14						
MLP,InputXGradient	21	18	9	20	18	4	5	5	1	1	1	1	1	1	1	1	44	42	42	50	9	10						
MLP,GuidedBackprop	3	2	2	3	2	2	1	1	0	0	1	1	1	0	0	0	11	11	11	9	22	34						
MLP,Saliency	3	3	2	3	2	2	1	2	0	0	1	1	1	0	0	0	15	14	14	10	34	59						
MLP-Agg,DeepLift	17	15	7	15	14	3	3	4	1	0	1	1	2	1	1	0	84	75	74	45	12	16	96	94	92	55	18	18
MLP-Agg,DeepLiftShap	17	15	7	15	14	3	3	4	1	0	1	1	2	1	1	0	83	75	74	45	12	15	94	93	90	55	17	17
MLP-Agg,IntegratedGradients	17	15	7	16	14	3	3	4	1	0	1	1	2	1	1	0	82	74	75	45	12	16	92	90	92	55	18	18
MLP-Agg,InputXGradient	25	22	12	25	22	5	6	6	1	0	1	1	1	1	1	0	53	50	50	63	10	11	55	55	55	88	12	12
MLP-Agg,GuidedBackprop	3	3	2	4	3	2	2	2	0	0	1	1	1	0	0	0	16	15	15	11	37	64	18	17	18	12	74	78
MLP-Agg,Saliency	3	3	3	4	3	2	2	2	0	0	1	1	1	1	0	0	17	15	16	11	39	70	18	17	18	12	78	86

Figure 4