Genomics Proteomics Bioinformatics 20 (2022) 899–911

Genomics Proteomics Bioinformatics

www.elsevier.com/locate/gpb
www.sciencedirect.com

ORIGINAL RESEARCH

Assessment and Optimization of Explainable Machine Learning Models Applied to Transcriptomic Data

Yongbing Zhao [1,*], **Jinfeng Shao** [2], **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*

Received 10 November 2021; revised 5 June 2022; accepted 25 July 2022
Available online 3 August 2022

Handled by Feng Gao

KEYWORDS

Machine learning;
Model interpretability;
Gene expression;
Marker gene;
Omics data mining

Abstract Explainable artificial intelligence aims to interpret how **machine learning** models make decisions, and many model explainers have been developed in the computer vision field. However, understanding of the applicability of these model explainers to biological data is still lacking. In this study, we comprehensively evaluated multiple explainers by interpreting pre-trained models for predicting tissue types from transcriptomic data and by identifying the top contributing genes from each sample with the greatest impacts on model prediction. To improve the reproducibility and interpretability of results generated by model explainers, we proposed a series of optimization strategies for each explainer on two different model architectures of multilayer perceptron (MLP) and convolutional neural network (CNN). We observed three groups of explainer and model architecture combinations with high reproducibility. Group II, which contains three model explainers on aggregated MLP models, identified top contributing genes in different tissues that exhibited tissue-specific manifestation and were potential cancer biomarkers. In summary, our work provides novel insights and guidance for exploring biological mechanisms using explainable machine learning models.

Introduction

In recent years, many tools based on machine learning models have been developed and applied to biological studies, most of which are developed for predictions. For example, AlphaFold was developed to predict protein 3D structure from amino acid sequences [1], P-NET was used to predict cancer treatment-resistance state from molecular data [2], and CEFCIG can

* Corresponding authors.
E-mail: im@ybzhao.com (Zhao Y), asmann.yan@mayo.edu (Asmann YW).

Peer review under responsibility of Beijing Institute of Genomics, Chinese Academy of Sciences / China National Center for Bioinformation and Genetics Society of China.

https://doi.org/10.1016/j.gpb.2022.07.003

predict cell identity regulators from histone markers [3]. Additionally, machine learning models can predict different biological features from one single data type, depending on which feature 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], and chromatin accessibility/activity/state [6–9] from genome sequences.

Although these tools have been greatly successful in various biological topics, biologists are still curious about how a machine learning model makes a decision and which features of the input data play important roles in the model output. To answer these questions, explainable artificial intelligence (XAI) programs have recently emerged to enable the development of models that can be understood by humans [10,11]. These XAI methods can also be applied to interpret machine learning models obtained from biological data by quantifying feature contributions to model prediction [12,13]. The two most popular approaches to estimate the contribution of each input feature to the model output are: 1) perturbing the input data and comparing outputs between the original and perturbated inputs; and 2) using backpropagation to measure the importance of each feature in the input data [14–16]. The former is intuitive but computationally expensive, especially when exhaustively estimating all input features, and there is also the risk of underestimating feature contribution [17]. By contrast, the latter can measure the contribution of all input features in "one shot". Consequently, many model explainers based on backpropagation were proposed and developed in the field of computer science and computer vision [18]. Benefiting from these model explainers, computational biologists discovered the syntax of transcription factor (TF) binding motifs by interpreting models trained to predict chromatin accessibility [19,20]; and screened for cancer marker genes from models of cancer type classification [21–23]. There is no doubt that these explorations have showcased the potential of interpretable models in discovering meaningful biological mechanisms. However, the remaining problem is that results from different model explainers are highly variable [21]. Since these model explainers were not specifically designed for biological data, it is critical to evaluate their applicability in biology. Currently, there is still a 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 relevance. To minimize the impact of model performance on the assessment of explainers, we tested explainers on well-trained models for predicting tissue types and cancer/normal statuses from gene expression data. In summary, this study provides comprehensive guide for applying interpretable machine learning to biological studies.

Results

Overview of model interpretability

In this study, we formulated a specific question to instantiate the application of interpretable models to biological data. Can we quantify the contribution of individual genes to tissue type and disease status? Two steps were implemented. First, we built neural network models and trained the models with transcriptomes as input and tissue type and disease status of the transcriptome sample as the prediction output. Models were built based on two types of neural networks, convolutional neural network (CNN) and multilayer perceptron (MLP), and model architectures are detailed in the methods section. In general, CNN is more complex than MLP. Next, we applied model explainers to compute a quantitative score of each gene's contribution to the model's prediction, which are named gene contribution scores. We tested eight popular model explainers and their variations commonly used in computer vision and assessed and compared their applicability and performances on each pre-trained model. These explainers are: gradients (Saliency), Input x Gradient (InputXGradient), guided backpropagation (GuidedBackprop), Integrated Gradients (IntegratedGradients), DeepLIFT (DeepLift), approximating SHAP values using DeepLIFT (DeepLiftShap), Guided Grad-CAM (GuidedGradCam), and Guided Grad-CAM++ (GuidedGradCam++) (Table S1) [17,18,24–29]. Since GuidedGradCam and GuidedGradCam++ were developed for CNN specifically, only the first six explainers were tested on MLP.

We used 27,417 RNA-seq samples from The Genotype-Tissue Expression (GTEx) and The Cancer Genome Atlas (TCGA) projects to train CNN or MLP-based models. These samples were from 82 distinct normal and cancer tissues and cell types (Table S2). After training, the prediction accuracy of all models was comparable, with a median value of 97.2% for CNN and 97.8% for MLP. The convolutional layers of the CNN models require that, as input data, gene expression values should be organized with a fixed gene order in a 2D matrix. Therefore, we tested various gene orders for the CNN-based models, *e.g.*, sorting genes according to their genomic coordinates [22]. Results indicated that gene order did not affect model performance in terms of prediction accuracy. Although models were trained with all 82 different normal and cancer tissues, results from four normal tissues (liver, lung, ovary, and pancreas) are reported here to illustrate the applicability and performance of the eight explainers.

Direct use of explainers from computer vision resulted in poor reproducibility

Randomness is often challenging in machine learning, which is present in both model training and model interpretation [30,31]. For this reason, we measured both intra-model and inter-model reproducibility of each explainer. During the testing, each explainer was applied to pre-trained models based on CNN or MLP, respectively.

First, we tested intra-model reproducibility by applying an explainer to the same pre-trained model 5 times (5 replicates per model per explainer), and for each explainer, we checked correlations of gene contribution scores as well as the pairwise overlap of the top 100 contributing genes between replicates. We found that intra-model reproducibility, in terms of Spearman's correlation of gene contribution scores and overlap of the gene IDs among the top 100 contributing genes, is low for most explainers on both CNN- and MLP-based models (results from normal liver samples are shown in **Figure 1**; results from normal lung, ovary, and pancreas are shown Figure S1). The exception was GuidedGradCam++, which was previously used to identify cancer marker genes [21].

As shown in Figure 1, we also tested inter-model reproducibility by applying each explainer to five different models with comparable prediction accuracies (5 models per explai-

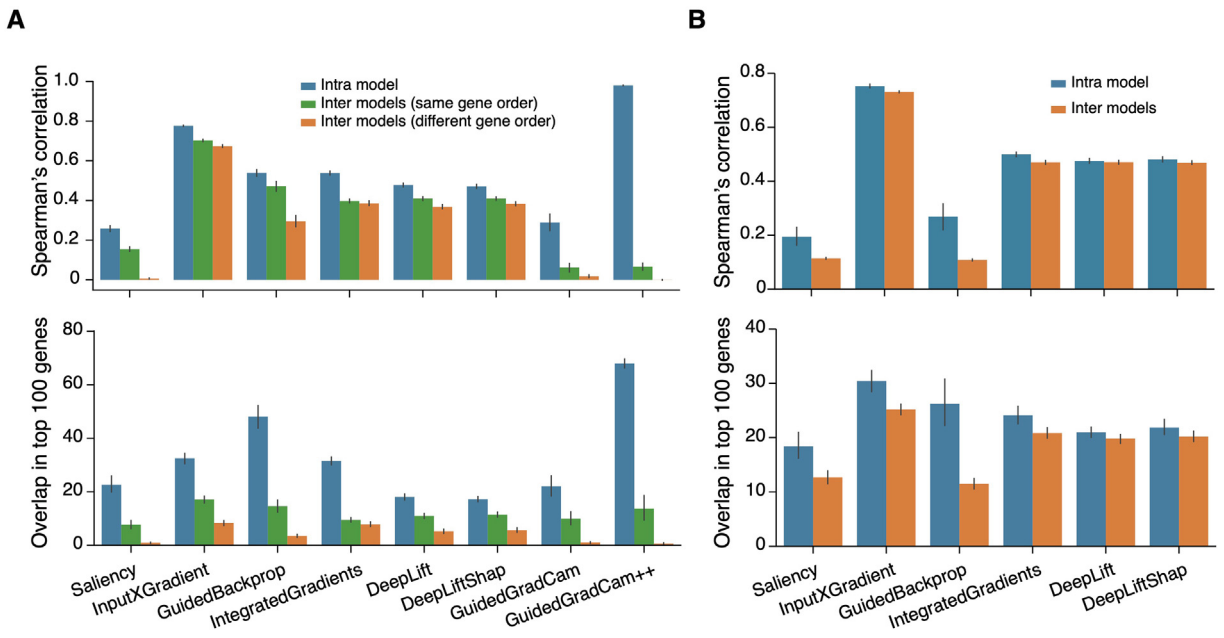

Figure 1 Performance of different model explainers without optimization

A. Spearman's correlation of gene contribution scores (upper panel) and overlap in the top 100 contributing genes (lower panel) in liver among replicates from the same pre-trained model, different pre-trained models with the same gene order, and different pre-trained models with different gene orders on CNN-based models. **B.** Spearman's correlation on gene contribution scores (upper panel) and overlap in the top 100 contributing genes (lower panel) in liver among replicates from the same pre-trained model and different pre-trained models on MLP-based models. CNN, convolutional neural network; MLP, multilayer perceptron; Saliency, gradients; InputXGradient, Input x Gradient; GuidedBackprop, guided backpropagation; IntegratedGradients, Integrated Gradients; DeepLift, DeepLIFT; DeepLiftShap, approximating SHAP values using DeepLIFT; GuidedGradCam, Guided Grad-CAM; GuidedGradCam++, Guided Grad-CAM++.

ner). These five models were trained using the same model architecture and training data set but with slightly different hyperparameters. For CNN models, we also tested the impact of different input gene orders since CNN requires organizing input genes in a 2D matrix. The inter-model reproducibility of all explainers, including GuidedGradCam++, was significantly lower than those of the intra-models. For CNN-based models, even though the gene order had little impact on prediction accuracy, it had a significant impact on the model reproducibility, especially regarding the overlap of top contributing genes. For MLP-based models, the reproducibility of intra-model and inter-model tests were similar; however, both Spearman's correlation and overlap of the top 100 contributing genes were very low.

In summary, gene contribution scores vary greatly intra- and inter-models for both CNN and MLP. These tests were performed per explainer. We expect that reproducibility across different explainers would be much worse. Therefore, model explainers developed for computer vision may not be directly applied to answering biological questions. Model interpretability in computer vision aims to identify visual features consisting of multiple similar pixels in an area, and variations within the area have a limited impact on the outcome. However, interpreting biological data such as the transcriptomes requires single-gene resolution since genes within an area were arbitrarily placed together, therefore, the results are very sensitive to random noise.

Optimization of model interpretability

Since it is not feasible to directly transfer model explainers from computer vision to biology, we tested whether these explainers can be optimized and adjusted for biological data. First, we borrowed a de-noising strategy widely used in computer vision, "SmoothGrad: removing noise by adding noise" [32]. Instead of estimating gene contribution scores in a sample in "one pass", SmoothGrad calculates the contribution of a gene by averaging contribution scores from multiple explanation estimates per sample by adding random noise into the expression data each pass. Unfortunately, the strategy of SmoothGrad did not improve inter-model reproducibility. On the contrary, it lowered the performance of all explainers on both CNN and MLP except for Saliency on MLP and GuidedBackprop on CNN (**Figure 2**A, Figure S2). For Saliency on MLP, the improvement saturates when the number of repeat estimates reaches 50, while the performance of GuidedBackprop plateaued after 30 repeats. Next, we tested whether repeating the explanation without adding random noise into the expression data, which we defined as a simple repeat, would be beneficial. Results of the simulation indicated that repeats without adding noise significantly improved the performance of all explainers on both CNN and MLP, except for Saliency and GuidedGradCam++ on CNN (Figure 2B, Figure S3). For most explainers, improvement saturated after 20 repeats for CNN and after 40 repeats for MLP.

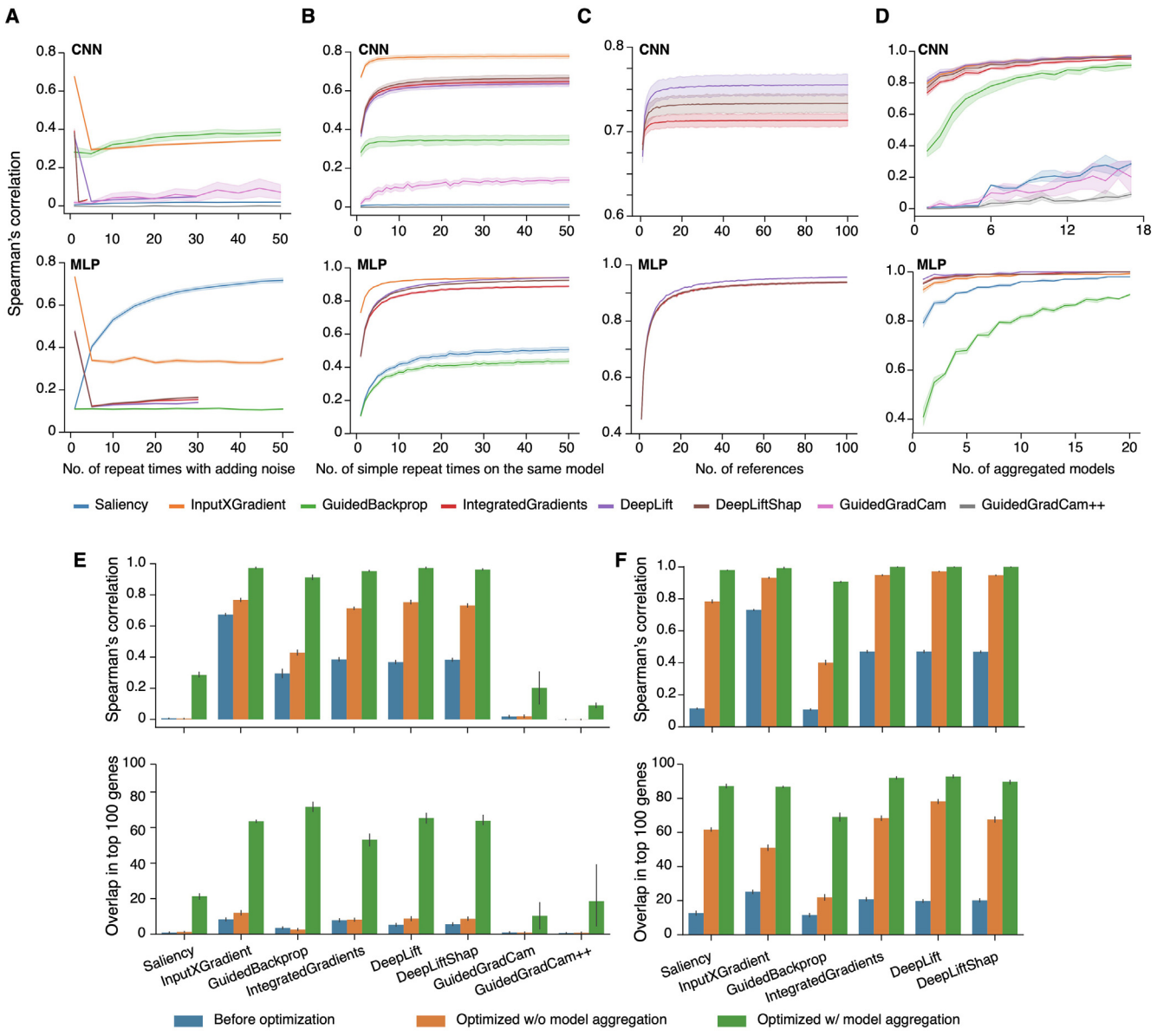

Figure 2 Optimization of different model explainers

Spearman's correlation of gene contribution scores in liver from CNN models (upper panels) and MLP (lower panels). **A.** Performance by averaging contribution scores from multiple estimates per sample by adding random noise into the expression data each pass. **B.** Performance from running the same model multiple times and averaging contribution scores without adding random noise (simple repeats). **C.** Performance of repeated reference zero for CNN, and a number of references randomly selected from 2000 simulated reference universal for MLP models, for three explainers that require reference samples. **D.** Performance of model aggregations. **E.** Spearman's correlation on gene contribution scores (upper panel) and overlap in the top 100 contributing genes (lower panel) among replicates from different pre-trained models with different gene orders based on CNN-based models. The analyses were carried out with three different optimization strategies: 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. w/o, without; w/, with.

DeepLift, DeepLiftShap, and IntegratedGradients require a reference baseline when estimating gene contribution scores, where a reference is a synthetic, randomly generated transcriptome. In computer vision, a black image (all-zero input) or random pixel values are often used as references, and for motif identification of regulatory elements, the scrambled genomic sequence was demonstrated as good reference [17]. In this study, we compared four types of references named reference zero, normal, universal, and specific. Reference zero and normal are equivalent to black image and random pixel values, respectively. For reference universal and specific, we estimated the mean (μ) and standard deviation (σ) of each gene's expression levels across samples and then randomly generated a value based on a truncated normal distribution

N (μ, σ). Reference universal uses samples from all 82 tissue types, while reference specific uses samples from a specific tissue type.

We tested the performance of these four kinds of references individually as well as in combination, to evaluate whether multiple references would improve reproducibility. First, for individual references, simulation results showed that reference zero is the best for CNN-based models, while universal is preferred for MLP-based models (Figure S4). The result is consistent with all the three explainers that required references. Next, we tested the impact of multiple references on reproducibility. Since the effect of using multiple references with zero is equivalent to that of simple repeat with single reference zero, these two kinds of optimization, using multiple references zero and simple repeat, 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 zero (which is equivalent to multiple references zero without simple repeat). Interestingly, we found that reference zero still outperformed the other three kinds of references on CNN-based models (Figure S5). Similar as simple repeat, improvement saturates when the number of references zero reaches 20 on CNN-based models, while the number of references universal goes to 60 on MLP-based models (Figure 2C).

Since the intra-model reproducibility was significantly improved by repeating the explanation process multiple times and averaging contribution scores from different estimates (Figure 2B), we next tested the benefits of inter-model aggregation. We applied optimized parameters of each explainer on CNN or MLP-based models (Table S1), estimated gene contribution scores on each pre-trained model individually, and then averaged inter-model results. Indeed, aggregating models significantly increased reproducibility (Figure 2D, Figure S6). Spearman's correlations for DeepLift, DeepLiftShap, IntegratedGradients, and Saliency reached nearly 1.0 on MLP. In general, the reproducibility of all explainers was significantly improved on both CNN and MLP-based models after aggregating models (Figure 2D–F, Figure S7). Of note, the model aggregation had the most significant impact and improvement on the reproducibility of all explainers on CNN-based models in terms of overlaps between the top 100 contributing genes (Figure 2E, lower panel). For most explainers on MLP-based models, Spearman's correlations on gene contribution scores from model aggregation were higher than 0.9, and over 90% of the top 100 contributing genes overlapped between replicates on the same explainer (Figure 2F).

To summarize, gene contribution scores were highly reproducible from the same explainer with optimized parameters. Reproducibility of the top 100 contributing genes was better on MLP-based models than those on CNN-based models. One possible reason is that CNN-based models are much more complex than MLP-based models and can be hard to be interpreted.

Consistency across model explainers

So far, we've tested the performance within each explainer. To test the consistency of gene contribution scores across different explainers, we checked the overlap of the top 100 contributing genes identified by different explainers with and without model aggregation (Table S3). Within CNN or MLP models, model aggregation did not only improve reproducibility within the same explainer, but also across explainers. However, the top 100 contributing genes from CNN-based models with model aggregation did not overlap with those from MLP-based models with or without model aggregation. Moreover, within CNN-based models, the top 100 contributing genes with model aggregation did not overlap with those without model aggregation, which suggests that model aggregation resulted in explainers identifying a completely different set of genes. By contrast, top contributing genes from MLP-based models were highly consistent with and without model aggregations. We further explored why model aggregation had different impacts on MLP and CNN. We first defined the top 100 contributing genes from optimized parameters with model aggregations as the baseline of comparison for each explainer. Next, we compared the top 100 contributing genes from each explainer without optimization to the baseline. It was found that, without optimization on MLP-based models, top contributing genes shared by two or more replicates had a higher overlap with the baseline than that between top contributing genes from individual replicates and the baseline (Figure S8). However, this conclusion was not seen on CNN-based models. This result suggests that the top contributing genes from different MLP models are convergent, while those from different CNN models are very divergent.

Intriguingly, the measurement of reproducibility highlighted three representative groups, each with different explainers, model types (CNN or MLP), and aggregation status (**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. Next, we delved into the top contributing genes identified by these three groups.

Expression status of top contributing genes

It is important to understand the biological relevance of the top contributing genes by different model architectures (MLP *vs.* CNN) and different explainers. Since contribution scores were derived from the input gene expression values, we first calculated Spearman's correlation between gene contribution scores and expression levels (Figure S9). We expected a high correlation for genes identified by InputXGradient because gene expression level is a cofactor in computing gene contribution scores by InputXGradient. Indeed, we found weak correlations of all explainers in both groups II and III except for InputXGradient. Conversely, strong correlations were observed from 4 explainers in group I: InputXGradient, IntegartedGradients, DeepLift, and DeepLiftShap. However, it is puzzling that GuidedBackprop from group I showed negative correlations for unknown reasons.

Additionally, we checked overlaps between the top 100 contributing genes and the top 100 expressed genes on both CNN- and MLP-based models (Figure S10). In liver, nearly 50% of

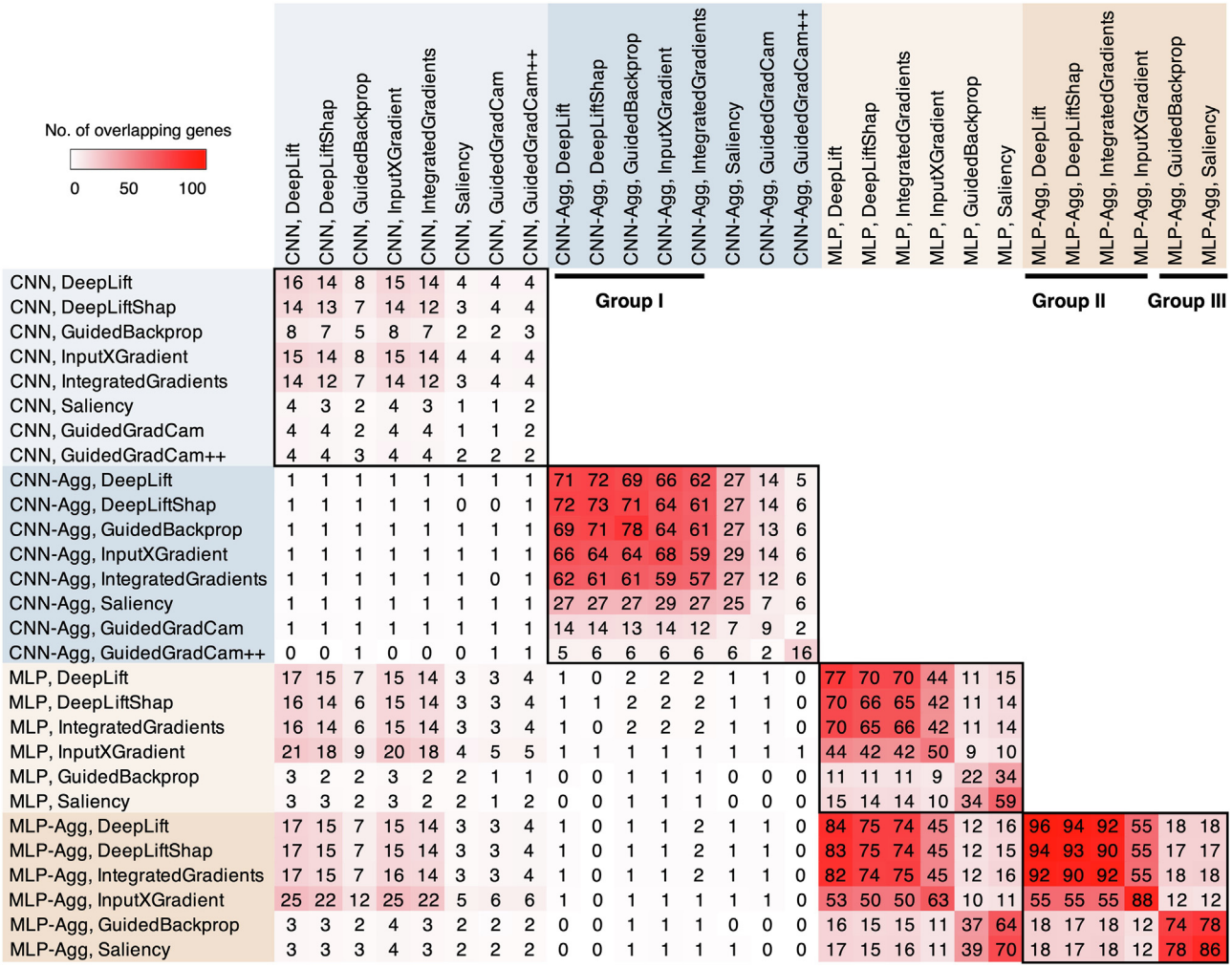

Figure 3 Overlap of the top 100 contributing genes across explainers with and without model aggregation
Three representative groups (I, II, and III) are marked by black bars. Aggregation of CNN models or MLP models is shown as CNN-Agg or MLP-Agg.

the top contributing genes from group II overlapped with top expressed genes, while the overlaps were less than 10% in groups I and III (Figure 4A). Group II, as described above, are DeepLift, DeepLiftShap, InputXGradient, and IntegratedGradients on MLP-based models with model aggregation. Strikingly, model aggregation eliminated the already moderate overlaps in CNN-based models (group I). Another noticeable finding for group I is that though Spearman's correlation between gene contribution scores and expression level was very high, the majority of the top contributing genes were not highly expressed.

Since different tissues have distinct phenotypes, we wondered whether the top contributing genes of different tissue types exhibit distinct expression profiles. Heatmap of the top contributing genes clearly demonstrated tissue-specific (TS) manifestations for group II (represented by DeepLift on MLP with model aggregation), and the patterns were much weaker in both group I (represented by DeepLift on CNN with model aggregation) and group III (represented by Saliency on MLP with model aggregation) (Figure 4B). In addition, the total number of genes from group III is much lower than that of both groups I and II after removing redundant genes from the top 100 contributing genes across tissues. This suggests that the top 100 contributing genes were largely shared across tissues in group III, which was validated by comparison across tissue types in all explainers (Table S4). Among the three groups, the top contributing genes in both groups I and II are TS, while the top contributing genes in group III are highly shared across tissues (Figure 4C).

Considering that the top contributing genes in groups I and II were mostly TS, we are curious how the top contributing genes are related to tissue specifically expressed genes (TS genes). For this purpose, we identified TS genes across all 82 tissues and cell types, which were 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 (Figure S11A), since there are different numbers of TS genes in each tissue type (Figure S11B). The percentages drop to less than 10% in both groups I and III. Interestingly, model aggregation also significantly reduced overlaps with TS genes in most explainers on CNN-based models.

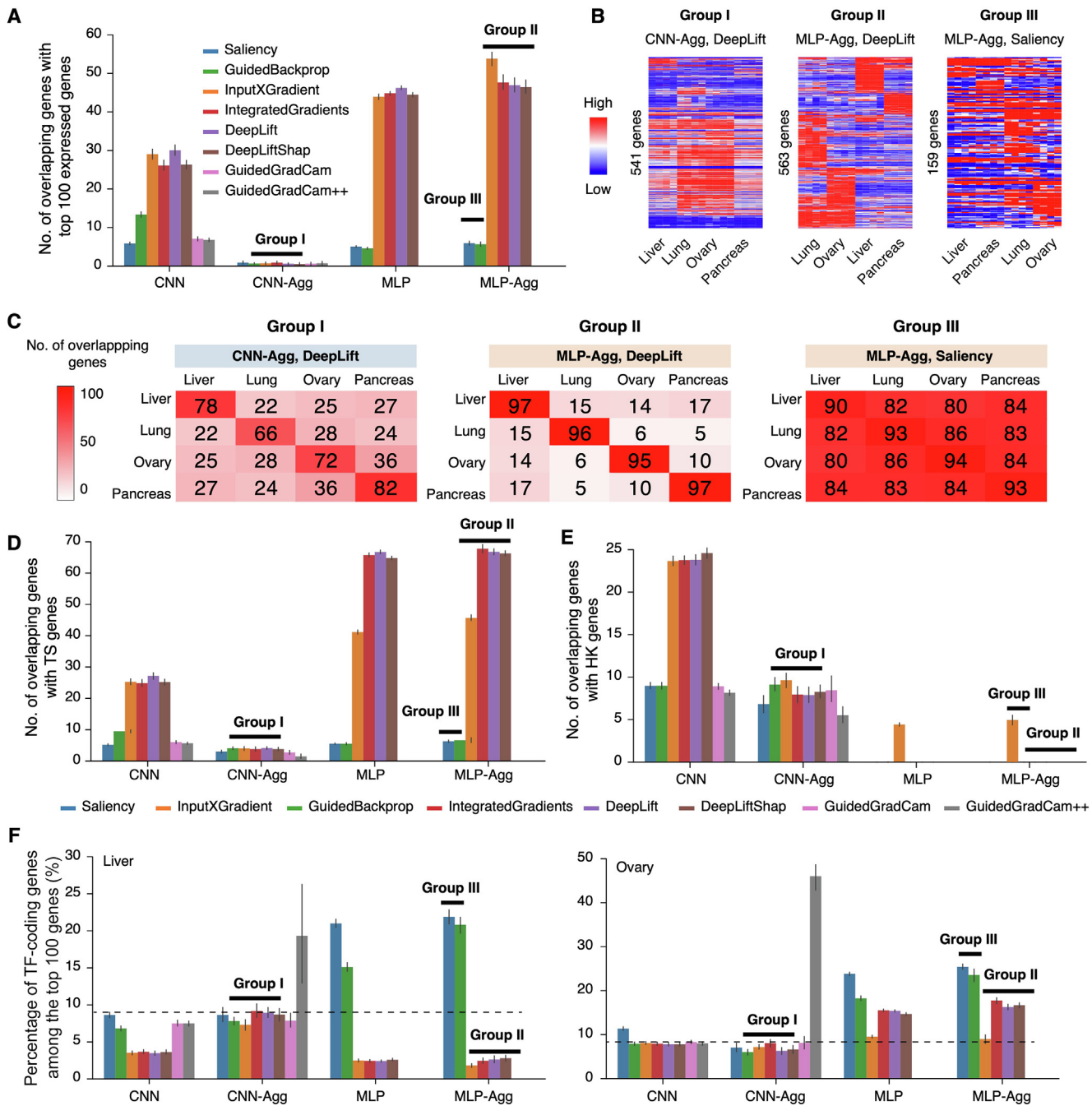

Figure 4 Biological relevance of the top 100 contributing genes

A. Overlaps between the top 100 contributing genes and the top 100 highest expressed genes in liver samples. Explainers that belong to groups I, II, and III are marked by horizontal 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 the top 100 contributing genes and TS genes in liver samples. **E.** Overlaps between the top 100 contributing genes and HK genes in liver samples. **F.** Percentages of TF-coding genes among the top 100 contributing genes in liver samples (left panel) and ovary samples (right panel). Dashed lines indicate the overlap percentage by random chance. TS, tissue-specific; HK, housekeeping; TF, transcription factor.

In addition, we observed that many of the top contributing genes (in group I particularly) are expressed at comparable levels across tissues. We investigated the relationships between the top contributing genes and housekeeping (HK) genes. Results showed that about 10%–20% of top contributing genes overlap with HK genes in group I, and the overlap was also further reduced by model aggregation (Figure 4E, Figure S12). Conversely, no overlap was found in both groups II and III, except for the explainer InputXGradient.

Enrichment of top contributing genes in biological functions

To understand the biological functions of the top contributing genes, we performed gene ontology (GO) enrichment analysis. No enrichment was found on genes identified by all explainers in group I. The enriched GO terms by genes from group II were mostly unique for each tissue type and TS functions (Table S5). For example, enriched GO terms in liver are molecular functions related to lipoprotein and lipoprotein lipase activities, while GO terms enriched in pancreas are associated with the binding of oligosaccharides, peptidoglycan and so on. Additionally, in group II, results among DeepLift, DeepLiftShap and IntegratedGradients are slightly more agreeable compared to that from InputXGradient. For group III, we expected similar GO terms enriched across tissue types since top contributing genes from different tissues highly overlapped. This turned out to be the case. GO enrichment analysis showed that the 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 across tissues. Next, we checked how top contributing genes are related to TFs and TF cofactors. We found about 20 genes overlapping with TFs in group III, which is more than 2-fold enrichment than by random chance (Figure 4F). By contrast, genes in group II showed depletion of TFs in liver, but 1.5-fold enrichment in ovary (Figure 4F, Figure S13). No enrichment or depletion was found in group I, except for genes identified by GuidedGradCam++ . As for TF cofactors, there is low overlap in all three groups (Figure S14).

Top contributing genes in cancers

Group II's top contributing genes are TS with TS manifestations of expression values. Therefore, we asked how the expression pattern of the top contributing genes changed from normal to cancer tissues. We compared normal and cancer samples of liver, lung, ovary, and pancreas from GTEx and TCGA, and studied the expression differences of top contributing genes. About 40%–80% of the top contributing genes were differentially expressed genes between normal and cancer tissues identified by DeepLift on MLP (group II), which is about twice more than the random chance (**Figure 5**A). The percentages ranged from 30% to 60% by Saliency on MLP

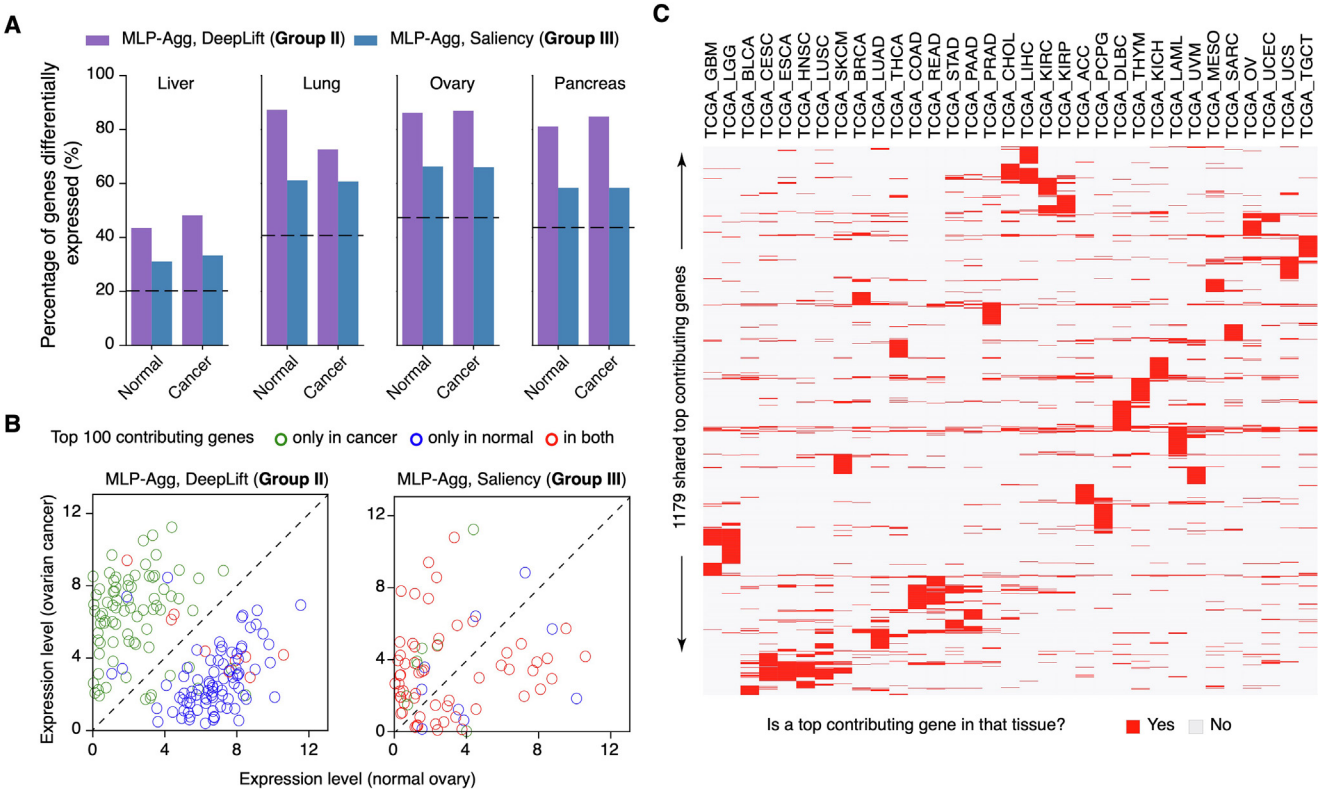

Figure 5 Top contributing genes in cancers

A. Percentages of the top 100 contributing genes which were differentially expressed between normal and cancer tissues. Dashed lines indicate the percentage of differentially expressed genes by random chance. **B.** Expression levels of the top 100 contributing genes which were observed only in cancer samples, only in normal samples, and in both normal and cancer samples, of genes identified by DeepLift on aggregated MLP models (shown as MLP-Agg, representative of group II, left panel) and by Saliency on aggregated MLP models (representative of group III, right panel). **C.** Heatmap of the 1179 shared top contributing genes from 33 TCGA cancer types demonstrated tissue specificity.

(group III), which is about 1.5-fold over random chance. Group I was not included in this analysis since model aggregation eliminated many features in common from different explainers on CNN-based models, and no biological enrichments were found in the top contributing genes.

Interestingly, differentially expressed top contributing genes are segregated into two distinct populations in group II (Figure 5B, Figure S15). 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 the heparan sulfate proteoglycans family, is one of the 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 [33]. However, similar segregation was not found in group III (Saliency on MLP-based models) because top contributing genes from group III were mostly shared between normal and cancer tissues. Together, the expression profiles suggest that the top contributing genes in group II might be potential cancer biomarkers. We identified the top 100 contributing genes in individual samples of all 33 different cancer types and named those shared by two or more samples of the same cancer type as shared top contributing genes. In total, 1179 genes were identified as shared top contributing genes. As expected, these shared top contributing genes are mostly TS (Figure 5C). Among these shared top contributing genes, we further studied the known oncogenes and tumor suppressor genes in OncoKB [34]. Heatmap analysis showed that some oncogenes and tumor suppressor genes are shared by multiple cancer types, such as *SFRP2*, while the others are specific to one or very few cancers (Figure S16).

Discussion

The beauty of interpreting machine learning models is that it converts the complex mathematical rules learned by neural networks into biological rules and provides new insights into biology. To facilitate the application of an interpretable machine learning model, we established a series of optimization steps and compared the biological relevance of different model explainers. Since the tests in this study were based on models that predict tissue types from transcriptomes, applications using other types of biological data or different model architectures may require further investigations. In addition, even though the current optimizations demonstrated good performance on MLP-based models for a subset of explainers, some important genes may still be missing from the top contributing genes. For example, a machine learning model might choose only one of two highly correlated genes to use for prediction. Alternatively, the contribution of two highly correlated genes might be diluted if the model chooses to use both genes, and thus, the contribution here might not reflect biological importance. These factors might partly explain the low reproducibility of individual single models and why improvement could be made by aggregation of models. Overall, we believe this study will provide novel insights to optimize interpretable machine learning in biological studies.

A recent paper pointed out five potential pitfalls of applying machine learning in genomics: 1) distributional differences; 2) dependent examples; 3) confounding; 4) leaky preprocessing; and 5) unbalanced class [35]. These technical challenges of applying machine learning models to genomics data are nontrivial and should be paid close attention to in addition to the optimization strategies we laid out in this study.

Typically, complicated models are not easily interpretable [36], which is also confirmed by the poor performance when interpreting CNN-based models. In this study, the optimized strategy significantly increased the interpretability on MLP-based models for a subset of explainers, but not on CNN-based models for any explainers. The aggregated CNN model approach should perhaps be categorized into a new modeling strategy, which is similar to the "averaging" of models. The "averaging" strategy indeed mitigated randomness to some extent but didn't show biological relevance. Therefore, even if models of different architectures had comparable prediction performances, it's probably preferable to use models with relatively simpler architectures for model interpretation.

The top contributing genes detected by explainers in group II (DeepLift, DeepLiftShap, InputXGradient, and IntegratedGradients on MLP-based models with model aggregation) exhibited TS manifestation in both gene ontology and expression profile, which is expected based on prior knowledge about tissue specificity and cell identity [37–40]. Therefore, explainers in group II are more suitable for biological study, especially when exploring biological questions based on transcriptomic data. In recent years, single-cell RNA-seq technology has been widely applied to different tissue and diseases, leading to the discovery of many well-defined sub-cell populations [41,42]. Although this study assessed model interpretability on bulk RNA-seq transcriptomes, the optimization strategies proposed here can also be applied to single-cell transcriptomes to quantify individual gene contribution and identify important genes in each sub-population. It is expected that interpretable machine learning models will also benefit understanding of tissue heterogeneity, disease mechanisms, and cellular engineering at single-cell resolution.

Materials and methods

Human transcriptome collection and processing

A total of 27,417 RNA-seq samples were used in our study, among which 17,329 and 10,088 samples were collected from GTEx and TCGA, respectively [43]. These samples are from 47 distinct primary normal tissues and 2 cell lines (with prefix GTEx_ in the tissue code) and 33 different primary tumors (with prefix TCGA_ in the tissue code). Pre-processed TCGA and GTEx RNA-seq gene expression level data were downloaded from GTEx Portal (phs000424.v8.p2) and Recount2 database [44], respectively. For TCGA data, only primary tumor samples were included. Names of tissue types (normal or cancer) remained the same as defined by TCGA and GTEx projects. For each sample, the expression levels of 19,241 protein-coding genes were normalized to $\log_2$ (TPM + 1), where TPM denotes Transcript Per Million, and then used for analyses.

The architecture of CNN models

We used a five-layer CNN to build the CNN models, which included three convolutional layers, one global average pooling layer, and one fully connected layer sequentially. Each

layer included 64, 128, 256, 256, and 82 channels, respectively. The kernel sizes for the three convolutional layers were 5, 5, and 3, respectively, and each convolutional layer was followed by max-pooling with a kernel size of 2. Batch normalization and rectified linear unit activation function [ReLU, which can be presented as $f(x) = \max(0, x)$] were applied immediately after max-pooling of each convolutional layer and global average pooling layer.

As the input of the CNN model, normalized expression values of 19,241 protein-coding genes from a sample were transformed into a 144×144 matrix, and zero padding was used at the bottom of the matrix. The final fully connected layer produced a vector of 82-probability-like scores, each corresponding to one of the 82 tissue types (normal or cancer).

The architecture of MLP models

There was only one hidden layer in the MLP models with 128 units. Batch normalization and ReLU were applied immediately after the hidden layer. There were 19,241 variables in the input layer, each corresponding to one of the 19,241 genes. The output layer assigns a probability-like score for each of the 82 tissue types.

Model training

All samples in a tissue type were randomly partitioned at a 9:1 ratio, with 90% of samples used as training data and the remaining 10% as testing data. In each epoch, up-sampling was employed to avoid imbalance caused by different sample sizes between 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 models, and 0.001 was used for MLP models. A batch size of 256 was used for both CNN and MLP. If there was no improvement for 5 sequential epochs, the learning rate was 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 models, while a dropout of 0.25 was applied before the global average pooling layer in the CNN models.

To optimize reproducibility in model explanation, we selected 60 well-trained models with slightly different parameters but of similar performances. In the CNN model, genes were organized into 2D matrix with fixed orders as input. In this study, gene orders in the CNN model were also experimented. For testing of the same gene order, we selected 5 well well-trained models with slightly different parameters but of similar performances. To study different gene orders, we selected 60 well-trained models with slightly different parameters but of similar performances.

Model performance estimation

Five-fold cross-validation was used to estimate the model performance for both MLP and CNN. Five groups of datasets were prepared, and each included a training dataset and a test dataset. Dataset preparation for each group was as follows. First, we randomly split all samples in a tissue type into 5 parts. Each group used one of the 5 parts as a test dataset, and the remaining four parts were combined into the training dataset. The same hyperparameters were used to train models based on the training dataset of each group separately. The trained models were then used to estimate the test dataset of the same group. Estimated results from five groups were combined, and all metrics about performance were calculated based on the combined results.

Model explanation

To estimate how much each gene contributes to the model prediction, we used eight different model explainers and variations, which are DeepLift, DeepLiftShap, GuidedBackprop, GuidedGradCam, GuidedGradCam++, InputXGradient, IntegratedGradients, and Saliency. All these explainers were implemented based on the Captum package (https://github.com/pytorch/captum). All explanations were based on well-trained CNN and MLP models. In addition, dropout was enabled to increase the diversity of model architecture, which helps measure the impact of model variations and uncertainties during the model explanation. As output, each explainer estimated contribution scores for each of the 19,241 genes.

Reference preparation

In this study, we tested four kinds of references which are named as zero, normal, universal, and specific. 1) For reference zero, we assigned the expression level of each gene to 0. 2) For reference normal, the expression level of each gene was randomly generated from a truncated normal distribution $N(0, 1)$, and all values were restricted between 0 and 1. 3) For reference universal, the expression level of each gene was randomly generated from a truncated normal distribution $N(\mu, \sigma)$, and all values are restricted between 0 and σ. Here, μ and σ were calculated based on expression values of this gene across all samples from all tissues, and σ is the standard deviation. 4) Reference specific was generated the same way as reference universal except that only samples of the same tissue types were used. In the reference testing, 2000 different references were generated for reference normal, universal, and specific separately. For zero, we repeated the same reference 2000 times.

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

A simulation was performed on 5 different pre-trained models as follows, using sample X 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 the mean of n scores. Step 3: repeat steps (1) and (2) on 5 different pre-trained 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 ($n = 1, 2, \ldots, 100$) 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 aforementioned method, we

obtained the relationship between a number of pseudo-samples and the correlation coefficient on any two replicates.

Simulation for optimal repeat number on the same model

The simulation process was performed on 5 different pre-trained models as follows, using sample X 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 the mean of n scores. Step 2: repeat step (1) on 5 different pre-trained 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 ($n = 1, 2, \ldots, 100$) times of 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 aforementioned method, we obtained the relationship between repeat number and correlation coefficient on any two replicates.

Simulation for an optimal number of references

The simulation process was performed on 5 different pre-trained models as follows, using sample X 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 2000 background samples pool. Step 2: repeat step (1) on 5 different pre-trained 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 pre-trained models. For the 5 replicates based on n ($n = 1, 2, \ldots, 100$) 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 aforementioned method, we obtained the relationship between a number of reference samples and the correlation coefficient on any two replicates. For each type of reference, we repeated the aforementioned simulation process individually.

Simulation for an optimal number of aggregated models

The simulation process was performed on 60 different pre-trained models as follows, using sample X as an example. Step 1: we estimated gene contribution scores for all genes in sample X on each pre-trained model, respectively. Step 2: to estimate gene contribution scores by aggregating n ($n = 1, 2, \ldots, 20$) models, we randomly selected n replicates out of 60, and the final gene contribution score for a specific gene was calculated based on the mean of n scores. Step 3: repeat step (2) K times, where $K = \max\left(\frac{60}{n}, 4\right)$. 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 ($n = 1, 2, \ldots, 20$) 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 aforementioned method, we obtained the relationship between repeat number and correlation coefficient on any two replicates.

Gene classification

Tissue specifically expressed genes were identified by the tool TissueEnrich with the group "Tissue-Enhanced" [45]. In each tissue, the median expression level of each gene was calculated across all samples, and HK genes are defined as genes with TPM $\geq$ 1 and less than 2-fold change on median expression level among all tissue types [46].

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 [47,48].

Annotation of TF and TF cofactors

All TFs and TF cofactors were downloaded from animalTFDB [49]. In total, there were 1666 TFs and 1026 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: false discovery rate (FDR) $\leq$ 0.001 and fold change $\geq$ 3.

Code availability

Scripts used to test model interpretability are based on Python and are freely available at https://github.com/zhaopage/model_interpretability.

CRediT author statement

Yongbing Zhao: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Jinfeng Shao:** Investigation, Writing - review & editing. **Yan W. Asmann:** Funding acquisition, Investigation, Writing - review & editing. All authors have read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Acknowledgments

The results here are in whole or part based upon data generated by the TCGA research network: https://www.cancer.gov/tcga. The data used for the analyses described in this study were obtained from the GTEx portal as

phs000424.v8.p2. We would like to thank Edward Asmann, Nancy L. Terry, National Institutes of Health (NIH) library editing service for reviewing the manuscript.

Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.gpb.2022.07.003.

ORCID

ORCID 0000-0002-9917-7425 (Yongbing Zhao)
ORCID 0000-0001-7227-6776 (Jinfeng Shao)
ORCID 0000-0002-8896-2647 (Yan W. Asmann)

References

[1] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature 2021;596:583–9.
[2] Elmarakeby HA, Hwang J, Arafeh R, Crowdis J, Gang S, Liu D, et al. Biologically informed deep neural network for prostate cancer discovery. Nature 2021;598:348–52.
[3] Xia B, Zhao D, Wang G, Zhang M, Lv J, Tomoiaga AS, et al. Machine learning uncovers cell identity regulator by histone code. Nat Commun 2020;11:2696.
[4] Chantsalnyam T, Lim DY, Tayara H, Chong KT. ncRDeep: non-coding RNA classification with convolutional neural network. Comput Biol Chem 2020;88:107364.
[5] Zhang J, Peng W, Wang L. LeNup: learning nucleosome positioning from DNA sequences with improved convolutional neural networks. Bioinformatics 2018;34:1705–12.
[6] Nair S, Kim DS, Perricone J, Kundaje A. Integrating regulatory DNA sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts. Bioinformatics 2019;35:i108–16.
[7] Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, Snoek J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res 2018;28:739–50.
[8] Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res 2016;26:990–9.
[9] Avsec Z, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods 2021;18:1196–203.
[10] Angelov PP, Soares EA, Jiang R, Arnold NI, Atkinson PM. Explainable artificial intelligence: an analytical review. Wiley Interdiscip Rev Data Min Knowl Disc 2021;11:e1424.
[11] Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 2020;58:82–115.
[12] Anguita-Ruiz A, Segura-Delgado A, Alcala R, Aguilera CM, Alcala-Fdez J. eXplainable artificial intelligence (XAI) for the identification of biologically relevant gene expression patterns in longitudinal human studies, insights from obesity research. PLoS Comput Biol 2020;16:e1007792.
[13] Alonso JM, Casalino G. Explainable artificial intelligence for human-centric data analysis in virtual learning environments. International Workshop on Higher Education Learning Methodologies and Technologies Online 2019:125–38.
[14] Talukder A, Barham C, Li X, Hu H. Interpretation of deep learning in genomics and epigenomics. Brief Bioinform 2020;22:bbaa177.
[15] Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. Nat Methods 2015;12:931–4.
[16] Torng W, Altman RB. 3D deep convolutional neural networks for amino acid environment similarity analysis. BMC Bioinformatics 2017;18:302.
[17] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. Int Conf Mach Learn 2017:3145–53.
[18] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 2017:4765–74.
[19] Avsec Z, Weilert M, Shrikumar A, Krueger S, Alexandari A, Dalal K, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. Nat Genet 2021;53:354–66.
[20] Kim DS, Risca VI, Reynolds DL, Chappell J, Rubin AJ, Jung N, et al. The dynamic, combinatorial *cis*-regulatory lexicon of epidermal differentiation. Nat Genet 2021;53:1564–76.
[21] Karim M, Cochez M, Beyan O, Decker S, Lange C. OncoNetExplainer: explainable predictions of cancer types based on gene expression data. arXiv 2019;1909.04169.
[22] Lyu B, Haque A. Deep learning based tumor type classification using gene expression data. Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics 2018:89–96.
[23] Li Y, Kang K, Krahn JM, Croutwater N, Lee K, Umbach DM, et al. A comprehensive genomic pan-cancer classification using the cancer genome atlas gene expression data. BMC Genomics 2017;18:508.
[24] Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv 2013;1312.6034.
[25] Shrikumar A, Greenside P, Shcherbina A, Kundaje A. Not just a black box: learning important features through propagating activation differences. arXiv 2016;1605.01713.
[26] Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: the all convolutional net. arXiv 2014;1412.6806.
[27] Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. Proc 34th Int Conf Mach Learn 2017;40:3319–28.
[28] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. Proc IEEE Int Conf Comput Vis 2017;2017:618–26.
[29] Chattopadhay A, Sarkar A, Howlader P, Balasubramanian VN. Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. IEEE Winter Conf Appl Comput Vis 2018;2018:839–47.
[30] Hartley M, Olsson TSG. dtoolAI: reproducibility for deep learning. Patterns (N Y) 2020;1:100073.
[31] Fan F, Xiong J, Li M, Wang G. On interpretability of artificial neural networks: a survey. IEEE Trans Radiat Plasma Med Sci 2021;5:741–60.
[32] Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M. Smoothgrad: removing noise by adding noise. arXiv 2017;1706.03825.
[33] Guo M, Zhang H, Zheng J, Liu Y. Glypican-3: a new target for diagnosis and treatment of hepatocellular carcinoma. J Cancer 2020;11:2008–21.
[34] Chakravarty D, Gao J, Phillips S, Kundra R, Zhang H, Wang J, et al. OncoKB: a precision oncology knowledge base. JCO Precis Oncol 2017;1:1–16.
[35] Whalen S, Schreiber J, Noble WS, Pollard KS. Navigating the pitfalls of applying machine learning in genomics. Nat Rev Genet 2021;23:169–81.

[36] Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: a survey on methods and metrics. Electronics 2019;8:832.
[37] Toyoda M, Hamatani T, Okada H, Matsumoto K, Saito H, Umezawa A. Defining cell identity by comprehensive gene expression profiling. Curr Med Chem 2010;17:3245–52.
[38] Ye Z, Sarkar CA. Towards a quantitative understanding of cell identity. Trends Cell Biol 2018;28:1030–48.
[39] Sonawane AR, Platig J, Fagny M, Chen CY, Paulson JN, Lopes-Ramos CM, et al. Understanding tissue-specific gene regulation. Cell Rep 2017;21:1077–88.
[40] Uhlen M, Fagerberg L, Hallstrom BM, Lindskog C, Oksvold P, Mardinoglu A, et al. Tissue-based map of the human proteome. Science 2015;347:1260419.
[41] Morris SA. The evolving concept of cell identity in the single cell era. Development 2019;146:dev169748.
[42] Stuart T, Satija R. Integrative single-cell analysis. Nat Rev Genet 2019;20:257–72.
[43] GTEx Consortium. The Genotype-Tissue Expression (GTEx) project. Nat Genet 2013;45:580–5.
[44] Collado-Torres L, Nellore A, Kammers K, Ellis SE, Taub MA, Hansen KD, et al. Reproducible RNA-seq analysis using recount2. Nat Biotechnol 2017;35:319–21.
[45] Jain A, Tuteja G. TissueEnrich: tissue-specific gene enrichment analysis. Bioinformatics 2019;35:1966–7.
[46] Eisenberg E, Levanon EY. Human housekeeping genes, revisited. Trends Genet 2013;29:569–74.
[47] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene Ontology: tool for the unification of biology. Nat Genet 2000;25:25–9.
[48] The Gene Ontology Consortium. The Gene Ontology resource: 20 years and still GOing strong. Nucleic Acids Res 2019;47:D330–8.
[49] Hu H, Miao YR, Jia LH, Yu QY, Zhang Q, Guo AY. AnimalTFDB 3.0: a comprehensive resource for annotation and prediction of animal transcription factors. Nucleic Acids Res 2019;47:D33–8.