Predicting DNA accessibility in the pan-cancer tumor genome using RNA-seq, WGS, and deep learning

Kamil Wnuk
NantOmics
Culver City, CA

Jeremi Sudol
NantOmics
Culver City, CA

Shahrooz Rabizadeh
NantOmics
Culver City, CA

Christopher Szeto
NantOmics
Santa Cruz, CA

Charles Vaske
NantOmics
Santa Cruz, CA

Abstract

DNA accessibility, chromatin regulation, and genome methylation are key drivers of transcriptional events promoting tumor growth. However, understanding the impact of DNA sequence data on transcriptional regulation of gene expression is a challenge, particularly in noncoding regions of the genome. Recently, neural networks have been used to effectively predict DNA accessibility in multiple specific cell types [14]. These models make it possible to explore the impact of mutations on DNA accessibility and transcriptional regulation.

Our work first improved on prior cell-specific accessibility prediction, obtaining a mean receiver operating characteristic (ROC) area under the curve (AUC) $= 0.910$ and mean precision-recall (PR) AUC $= 0.605$, compared to the previous mean ROC AUC $= 0.895$ and mean PR AUC $= 0.561$ [14].

Our key contribution extended the model to enable accessibility predictions on any new sample for which RNA-seq data is available, without requiring cell-type-specific DNase-seq data for re-training. This new model obtained overall PR AUC $= 0.621$ and ROC AUC $= 0.897$ when applied across whole genomes of new samples whose biotypes were held out from training, and PR AUC $= 0.725$ and ROC AUC $= 0.913$ on randomly held out new samples whose biotypes were allowed to overlap with training.

More significantly, we showed that for promoter and promoter flank regions of the genome our model predicts accessibility to high reliability, achieving PR AUC $= 0.838$ in held out biotypes and PR AUC $= 0.908$ in randomly held out samples. This performance is not sensitive to whether the promoter and flank regions fall within genes used in the input RNA-seq expression vector.

Finally, we utilize this tool to investigate, for the first time, promoter accessibility patterns across several cohorts from The Cancer Genome Atlas (TCGA) [27].

1 Introduction

DNA accessibility plays a key role in the regulatory machinery of DNA transcription. Locations where DNA is not tightly bound in nucleosomes, detectable as DNase I hypersensitivity sites (DHSs), render the sequence accessible to other DNA-binding proteins, including a wide range of transcription factors (TFs). DHS sites are cell specific and play a crucial role in determining cell-selective transcriptional events.

Furthermore, genome wide association studies (GWAS) have revealed that the vast majority of genetic variants significantly associated with many diseases and traits are located in non-coding regions [8].

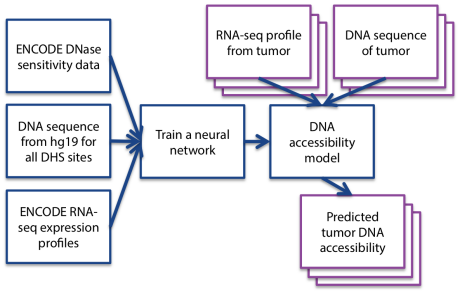

Figure 1: Overview of our pipeline from training to application. Once a neural network is trained, we can apply our accessibility prediction to any new data for which whole genome sequencing and RNA-seq are available.

Among such non-coding single nucleotide polymorphisms (SNPs), well over half affect DHS sites [19]. Thus variable access to DNA regulatory elements not only plays a key role in normal cell development, but also in altered expression profiles associated with disease states [31, 8].

In an effort go beyond association studies and gain deeper insight into how changes in DNA sequence data impact transcriptional regulation, predictive models have been developed for a multitude of genomic phenomena. Several works have recently made significant advances in accuracy of such DNA-sequence-based prediction tasks by applying neural network models to problems such as transcription factor binding [2, 34, 20, 17], promoter-enhancer interactions [24], DNA accessibility [34, 14], and DNA methylation states [3].

One common issue that limits the broad applicability of these models is the cell-type-specific nature of many of the underlying biological mechanisms, such as DHS sites. In all of the above predictive model examples the issue was addressed by either training a separate model for each cell type or by having a single model output multiple cell-type-specific (multi-task) predictions. This made it difficult to apply the models to new data and limits them from being integrated into broader scope pathway models [30].

Our work builds on a recently successful neural network for predicting DNA accessibility [14] and focuses on overcoming the barrier to broad applicability due to cell-specific phenomena. We began by making some initial improvements to the baseline model (Sections 2.1 – 2.2). Based on the utility of RNA-seq as a signal for cell type clustering and classification, we then hypothesized that given paired RNA-seq and DNase-seq input data a neural network should be capable of learning to appropriately modulate its prediction. Beyond eliminating the need for a distinct trained model or unique output per cell type, replacing a discrete categorization with a continuous signature gave the model access to similarity information in the space of cell types. This made it possible to make DNA accessibility predictions for previously unseen cell types whose gene expressions were similar but unique from samples in the training data.

We present the new model in Section 2.3 and the dataset collected for training in Section 2.4. We evaluated the model's performance on samples held out from training in Section 3.1 and applied whole genome accessibility prediction to 6 different cancer cohorts from The Cancer Genome Atlas (TCGA) in Section 3.3. To the extent of our knowledge this is the first time that a prediction of DNA accessibility has been applied to whole genomes in any TCGA cancer cohort. We provide a first view of the predicted accessibility landscape across several cancers using high dimensional data visualization tools.

2 Predicting DNA accessibility

2.1 Cell-type-specific model

The Basset neural network architecture recently demonstrated state of the art results on DNA accessibility prediction [14]. They factored the cell-specific DHS site issue into their work in two ways. First, they created a binary matrix of biosample types and their respective accessibility for a list of genomic sites. The universal list of (potentially accessible) sites was found by agglomeratively

clustering all overlapping DNase-seq peaks across all samples before training. The model's final layer was then set up as a multi-task output, with a distinct prediction unit for each biotype.

Input to the Basset model was a 600 base-pair segment of DNA represented as a one-hot code. The input was fed through a hierarchy of three convolutional (conv) layers, each succeeded by max pooling, followed by two fully connected layers. Batch normalization was utilized at all layers, and a max norm constraint was applied for regularization of all weights during training. The final output layer consisted of 164 sigmoid units, so for every input sequence the model produced a unique prediction for each of the training cell types. The model architecture is illustrated in Figure 3

We used TensorFlow [1] to implement the Basset model, with minor changes, as our baseline. First, we used Adam [15] instead of RMSProp [28] to optimize network parameters. We also found that use of a dynamic decay rate (that increased over the course of training) for updating moving averages in batch normalization led to a model with competitive performance more quickly than when using a fixed decay. Compared to an ROC AUC $= 0.895$ reported on the test set in [14] and confirmed by applying the pre-trained model provided by the authors, our baseline implementation achieved a mean ROC AUC $= 0.903$.

2.1.1 Baseline dataset

All cell-specific models described in this work were trained and evaluated following the exact procedure of the Basset network [14], using DNase-seq peak data from 164 sample types obtained from ENCODE [7] and Roadmap Epigenomics [21] projects.

A universal set of potential accessibility sites was created by greedy merging of overlapping peaks across all DNase-seq data samples. For each site, a binary vector was used to label its accessibility state in each of the 164 cell types. Data was then split by genomic site so that 70,000 peak locations were held out for validation, 71,886 for testing, and the remaining 1.8 million sites were used for training.

2.2 Convolutional layer factorization

Following the success of many works demonstrating that deeper hierarchies of small convolutional kernels tend to improve neural network performance [23, 26, 11], we experimented with factorization of large convolutional layers in the baseline model. We focused experiments on factorizations that maintained the effective region of influence of the original layers and did not significantly increase the overall number of network parameters.

We found that factorization of layers closest to the data was the most significant for improving accuracy (Figure 2). When only the second convolutional layer was factorized the speed of learning improved during the early epochs of training, but final accuracy was not noticeably affected compared to our baseline implementation. An overall improvement in both rate of learning and final accuracy was achieved when both the first and second convolutional layers were factorized, leading to a mean ROC AUC $= 0.910$ (Figure 3).

Furthermore, despite following the same training procedure and taking no additional steps to account for class imbalance, our final model's mean PR AUC $= 0.605$, which compared favorably to the mean PR AUC$= 0.561$ reported as the best result obtained by the Basset model.

There is a multitude of alternate model architectures (such as densely connected convolutional networks[13] and LSTMs [12]) and additions (such as attention mechanisms [4, 32]) which we believe are likely to benefit performance, however they have been left for future evaluation, as the key contribution of this work is to move beyond the cell-type-specific limitations of DNA sequence classifiers.

2.3 Making predictions in new cell types without re-training

With the models described thus far, each new type of cell or tissue encountered required the network to first be trained with DNase-seq peaks measured from the new type of sample before any predictions could be made. This issue is due to the discretization of cell types and can be avoided by instead using a numerical signature based on which cells and tissues can be characterized. Having such a signature as a parallel input would enable the network to leverage similarity and structure in the space

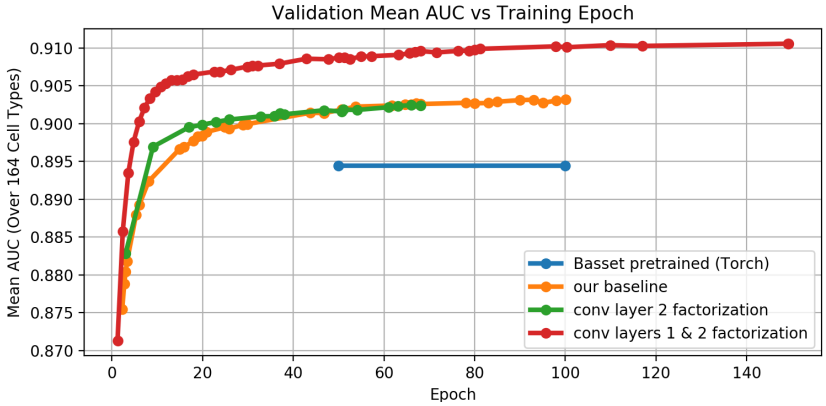

Figure 2: Training of cell-type-specific model architectures: mean ROC AUC across 164 cell types in the validation set versus training epoch. The result obtained by the pre-trained model provided by the authors of Basset [14] is shown for reference, but since we did not know the number of training epochs an arbitrary range was selected for display. We explored independent factorization of the second convolutional layer of the baseline model, and achieved the best performance when both the first and second convolutional layers were factorized.

of cell types and learn how DNA accessibility is modulated in a more general way (by a sample's coordinates in the cell signature space).

Several studies have demonstrated that gene expression levels estimated from RNA-seq can be used to disambiguate cell types [25, 5, 6]. Also, DNase-seq and microarray based gene expression levels from matched samples were found to cluster similarly according to biological relationships, and many DHS sites were found to significantly correlate with gene expressions [22]. We qualitatively observed similar biologically meaningful neighborhood relationships in both DNase-seq and RNA-seq when visualizing new data collected from ENCODE (Figure 4, Section 2.4).

All the above suggest the possibility that there is structure in the space of gene expression that may be informative for how accessibility is modulated in different cell and tissue types. We therefore chose to use gene expression levels derived from RNA-seq as the supplementary signature to incorporate into the model. RNA-seq also has the benefit of being one of the most commonly available measurements across large data sources of interest in research such as TCGA [27] or the Genotype-Tissue Expression (GTEx) project [9].

There were many possible strategies for selecting the subset of genes for our input signature. However to initially avoid optimizing in this space we relied on the prior work of the Library of Integrated Network-based Cellular Signatures (LINCS) and utilized their curated L1000 list of genes[18]. Future exploration should be done to investigate use of an autoencoder that leverages a more complete set of genes instead of a manually curated subset.

The cell-specific models (Sections 2.1–2.2) demonstrated that multi-task outputs could share common convolutional layers and provide an accurate prediction of DNA accessibility across distinct sample types. Thus we expected that if a signature was discriminative of cell type it was likely to be sufficient to integrate it into the network after the convolutional layers. We evaluated a direct concatenation of the RNA-seq gene expression vector with output from the convolutional layers (pictured in Figure 3), as well as first passing the new input through an additional fully connected layer (depth$= 500$) before concatenation with conv layer outputs (evaluation in Figure 5). The former turned out to be the better performing model of the two.

2.4 ENCODE DNase-seq and RNA-seq dataset

DNA accessibility is one of many complex factors that eventually determine expression at the level of RNA-seq, which makes the relationship not trivially invertible. Knowing expression levels does not uniquely define the pattern of DHS sites. Our extended model learns a most likely mechanism with which the DNA sequence immediately surrounding a potential DHS site determines its accessibility in the context of observed gene expression levels. Thus when accessibility prediction is applied

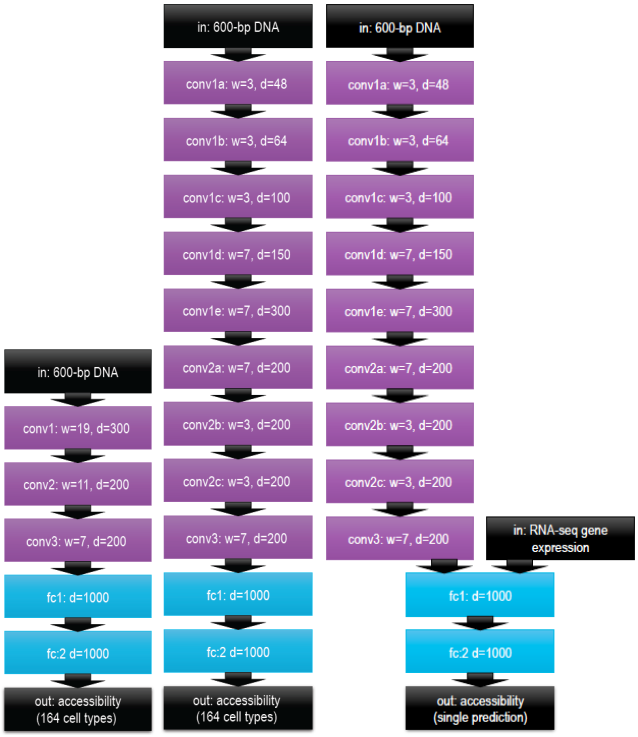

Figure 3: Architectures for the baseline (left), final cell-specific (center), and final RNA-seq signature (right) models. Max pooling with equal size and stride is applied after each conv layer in the baseline model. Sizes of the pooling layers following conv1, conv2, and conv3 are 3, 4, 4 respectively.

across the whole genome it can be viewed as an approach that inverts gene expression to obtain most likely DHSs, constrained only by local sequence information.

In order to train a model for predicting accessibility in the context of gene expression levels it was necessary to build a new dataset where both DNase-seq and RNA-seq were available for a large and diverse collection of different cell types. Data from the ENCODE project was initially collected at the start of January 2017 for all biotypes for which RNA-seq and DNase-seq measurements were both available. In order to capture a greater diversity of biosample types, gene quantifications from RNA-seq files with the following ENCODE labels were collected: "RNA-seq", "polyA mRNA", "polyA depleted", "single cell". All files with "ERROR" audit flags were rejected. We kept files with "insufficient read depth," and "insufficient read length" warnings. Despite being below ENCODE project standards, we believe the available read depths and lengths in warning situations were likely to be less of an issue when it comes to differentiating cell types [6], and preferred to accept more potential noise in favor of a larger diversity of sample types.

The final step of data preparation involved assigning associations between specific RNA-seq and DNase-seq files within the same biotype. In cases where there existed multiple exact matches of biosample accession between the two file types, associations were restricted to such exact matches. If exact biosample accessions did not match, two file types were associated if it could be verified that they originated from the same tissue sample, cell line, or patient. This eliminated several biotypes for which no such correspondences existed. Both technical and biological replicates were treated as independent samples of the same biotype since we wanted to put the burden of learning non-trivially-invertible aspects of noise on the neural network model.

The dataset was refined at the end of August 2017, as several samples that had been part of our training and testing data were revoked by the ENCODE consortium due to quality concerns and updates. The final dataset consisted of 74 unique biotypes.

We partitioned the paired samples into training, validation, and test sets with a file distribution as illustrated in Table 1. The validation set was held constant, while the training and test sets included

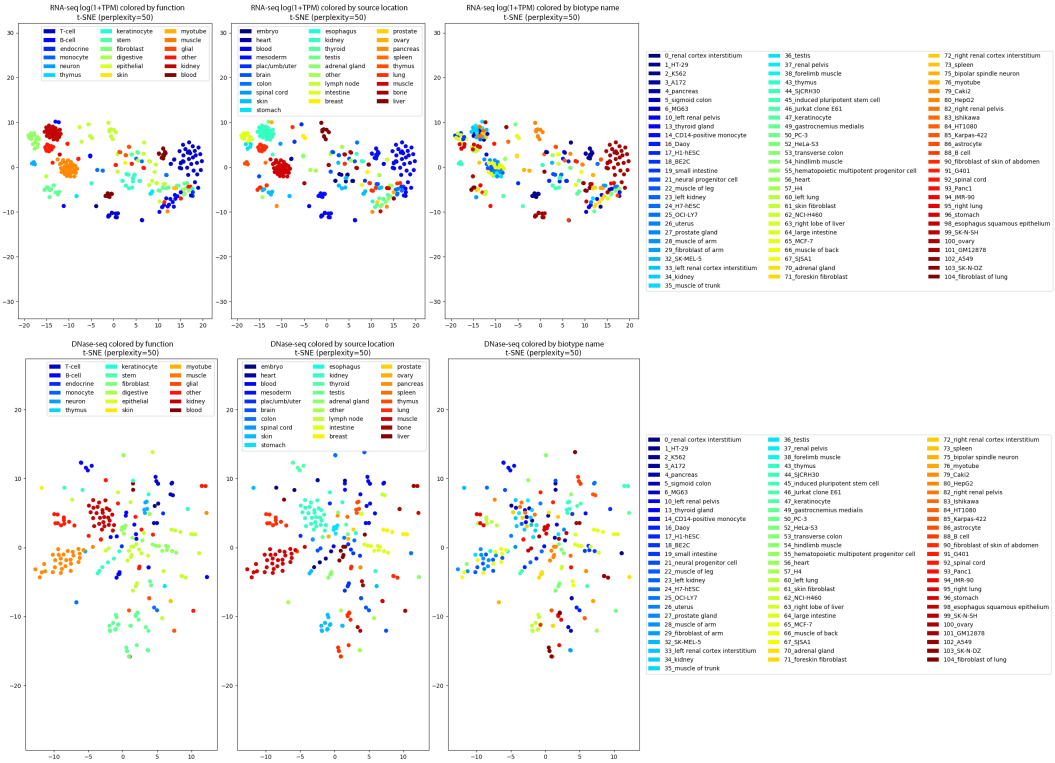

Figure 4: t-SNE [29] visualization of RNA-seq gene expressions (top) and binary DNase-seq peaks (bottom) from our ENCODE dataset. Each point corresponds to all sample information derived from a distinct file. Color codes illustrate rough categorizations of biological function (left), sample source location (center), and specific ENCODE biotype names (right). In both datatypes there are visible clusters of categories such as muscle, kidney, lung, intestine/stomach, etc. We also notice a few other categories, such as fibroblasts and endothelial cells, appear as several scattered local clusters, suggesting that our data lives on a manifold difficult to visualize in two dimensions. These qualitative groupings are consistent with the observations [22] discussed in Section 2.3.

two variations. The first test set consisted of randomly held-out samples, only one of which ended up being a biotype not represented in the first training set. The second partitioning was specifically selected such that all samples in the test set were from biotypes not represented in the training or validation data. The latter was meant to more accurately simulate the intended application scenario and was thus the main focus in our analysis.

Table 1: Number of file types per dataset partition

Partition	DNase-seq files	RNA-seq files
Validation set	11	12
Train set 1	195	277
Train set 2	198	281
Test set 1	14	15
Test set 2	11	11

We utilized the same greedy merge methodology described in Basset [14] on all DNase-seq samples in our training sets to obtain a set of all potential sites of accessible DNA along the whole genome. As previously, we used a fixed length of 600 base pairs centered at DHS peaks to define each site. Blacklisted sites at which measurements were suggested to be unreliable were excluded [16]. This led to a total of 1.71 million sites of interest in the case of the main (held out biotype) data split, and 1.75 million sites in the alternate data split. Using all sites across all available DNase-seq files, this produced a total 338.7 million training examples in the main data split.

As in other recent work on DNA-based prediction tasks [2, 14, 24, 20] the sequence for each genomic site was obtained from human genome assembly hg19/GRCh37. To faithfully represent the true biology training should have utilized whole genome sequencing data for each of the samples in our dataset, since DNA sequences of all unique biosamples vary and using data from the reference genome introduces non-random noise in the input space. It is difficult to know the impact of this noise, but irrespective of task we expect that more accurate models of sequence function can be obtained by using data specific to each sample's genotype. Unlike models with multi-task outputs our architecture can easily support such training without any changes, but unfortunately whole genome data for all samples was not available to us at the time.

During training data was balanced per batch due to a $14:1$ ratio of negative to positive examples. Each batch sampled an equal amount of accessible and non-accessible sites without replacement, such that one pass through all available negative training examples constituted multiple randomly permuted passes through all positive training examples. In situations where a DNase-seq file had more than a single matching RNA-seq file, sites from that DNase-seq file were randomly assigned to one of the multitude of corresponding RNA-seq expression vectors each time they were selected for a training batch.

To generate a validation set that was manageable to evaluate frequently we selected 40,000 random samples from each of accessible and non-accessible sites per validation DNase-seq file. This resulted in a set of 440,000 validation examples that were used to estimate ROC AUC throughout training.

However, we also evaluated prediction performance across whole genomes (all potential DHSs) of all validation samples. In cases where multiple RNA-seq file matches existed, predictions across the entire genome were evaluated once for every possible DNase-seq and RNA-seq file pair. This gave a better characterization of performance on the intended application, especially as captured by PR AUC, which is less misleading in the presence of data imbalance. Results on the test set were evaluated across whole genomes following the same procedure. Using this paradigm the total number of examples for validation was 20.5 million and 22.2 million for testing in the held out biotype data split.

3 Results

3.1 ENCODE evaluation

We trained several alternative versions of our model and reported validation results over the course of training in Figure 5 and Figure 6. We found that adding a fully connected layer before concatenating RNA-seq data to output from the convolutional layers performed consistently worse than direct concatenation without the fully connected layer.

Transfer learning consistently shortened the training time across model variants. However, our most impactful changes were increasing the batch size (from 128 to 512, and finally to 2048), and decreasing the learning rate (from 0.001 to 0.0001).

The cell-specific models (Sections 2.1-2.2) had multi-task outputs so that each training sample provided an information-rich gradient based on multiple labels for backpropagation. Since using RNA-seq inputs eliminated the need for multi-task outputs, each sample now only provided gradient feedback based on a single output. The batch size increase was thus intended to compensate for this change in output dimension to produce a more useful gradient for each batch.

The learning rate decrease, on the other hand, was guided by the observation that training was reaching a point of slow improvement before even a single full pass through all negative training examples. Our new dataset was also significantly larger than that used to train cell-specific models.

In transfer learning using weights learned from the corresponding data splits before final cleanup of revoked files was more effective on the final data than transfer of conv layer weights from the best cell-specific model. Since some of the revoked samples featured a very high rate of DHS peaks, the pre-revoke dataset included many more sites of interest (2.7 million). This meant that aside from many additional negative examples, a fair number of potentially accessible sites also had differently centered peaks. This added positional noise could have encouraged model robustness similarly to how conv nets for images are often improved when training examples are randomly translated and

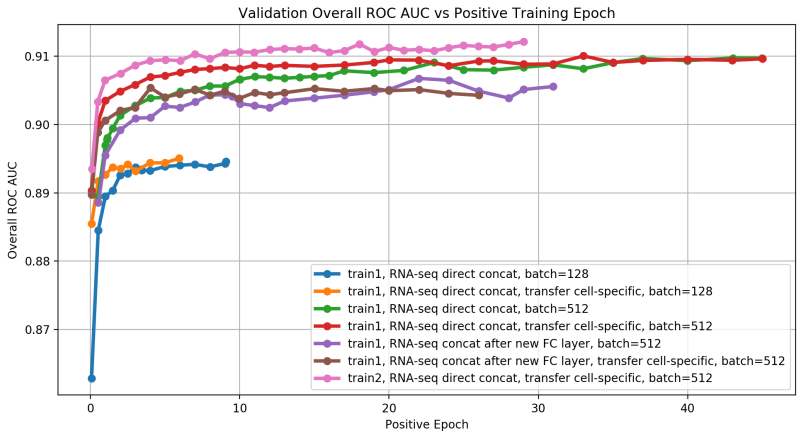

Figure 5: Overall ROC AUC for the small validation set over number of passes through all positive examples (positive epochs) for model architectures using RNA-seq input. We experimented with adding a fully connected (FC) layer of depth 500 before concatenating gene expressions with outputs from the conv layers. However, increasing the batch size and initializing the conv layers with weights from our final cell-specific model (transfer) improved performance most. Models trained on set 1 showed similar validation performance as those trained on set 2 with the same hyperparameters. This evaluation was done before the final dataset revision which revoked several suspected low quality samples, yet still provided valuable feedback for model selection.

cropped. Thus the utility of small random sequence shifts by several nucleotides during training may be worth evaluating in the future when applying conv nets to DNA sequence data.

Our final model, as illustrated in Figure 3, was initialized with weights learned from the prior iteration of the dataset, before the final revoked files were removed. In turn, those models were initialized with conv layer parameters from our best performing cell-specific model (Section 2.2). An effective batch size of 2048 was used for training (2 GPUs processing distinct batches of 1024), with a Adam [15] learning rate of 0.0001 and a 0.25 fraction of positive to negative samples in every batch.

Table 2: Whole genome validation results for our final model trained on set 1 (t1) and set 2 (t2)

Sample biotype	ROC AUC (t1)	PR AUC (t1)	ROC AUC (t2)	PR AUC (t2)
CD14-positive monocyte	0.888	0.559	0.889	0.563
muscle of arm	0.774, 0.959	0.654, 0.808	0.783, 0.960	0.671, 0.811
fibroblast of arm	0.898, 0.900	0.806, 0.809	0.898, 0.900	0.808, 0.811
SJCRH30	0.875	0.727	0.875	0.730
foreskin fibroblast	0.953	0.774	0.953	0.771
right renal pelvis	0.967	0.833	0.968	0.836
spinal cord	0.947	0.714	0.946	0.713
Panc1	0.957	0.713	0.958	0.711
right lung	0.958	0.781	0.958	0.782
A549	0.902	0.735	0.900	0.734
mean biotype AUC	0.915	0.743	0.916	0.745
overall AUC	**0.912**	**0.721**	**0.913**	**0.725**

Table 2 shows that final model performance on the validation set, both overall and by biotype, was consistent across each of the two training partitions with respect to both ROC AUC as well as PR AUC. Table 3 and Table 4 summarize the results of applying our model across whole genomes, at all potential DHS sites. For biotypes with more than a single file pair in the test set, each sample's results are listed.

It was not surprising that overall the model performed worse on completely new biotypes, however we were pleased to see that even in the more challenging scenario the overall PR AUC was higher than the best cell-specific models evaluated on known biotypes (Section 2.2). Note that several of

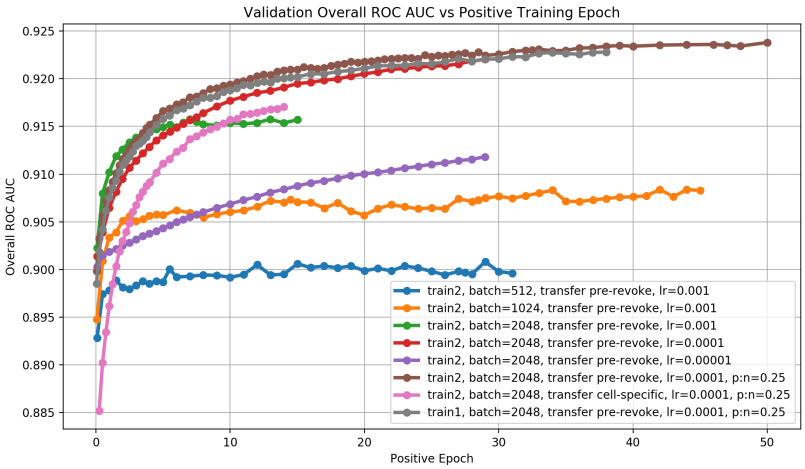

Figure 6: Overall ROC AUC for the small validation set over positive training epochs for models trained after the final dataset revision. A further increase in batch size as well as a decreased learning rate led to additional significant improvements. Changing the fraction of positive samples per training batch (from 0.5 to 0.25) also slightly improved both ROC AUC as well as PR AUC in whole genome validation. Transfer of weights learned before final revoking of data (Figure 5) was a more effective initialization than transfer learning from our final cell-specific model. Finally, we again confirmed that the same hyperparameters led to good validation performance across both training partitions.

the results in Table 4 were within similar ranges as predictions whose sample types overlapped with training.

Table 3: Test set 1 whole genome results

Sample biotype	ROC AUC	PR AUC
A172	0.959	0.721
left renal pelvis	0.967	0.838
small intestine	0.951, 0.926	0.737, 0.571
muscle of arm	0.968	0.843
forelimb muscle	0.968	0.808
keratinocyte	0.939	0.644
skin fibroblast	0.948, 0.947	0.770, 0.763
large intestine	0.964	0.727
muscle of back	0.967, 0.954	0.853, 0.854
adrenal gland	0.957	0.743
SK-N-DZ	0.898	0.652
fibroblast of lung	0.942	0.840
mean biotype AUC	0.950	0.758
overall AUC	**0.947**	**0.748**

3.2 Performance by genomic site

To better understand the performance characteristics and limitations of our model we broke down our ENCODE validation and test results by genomic site type. Exon, protein coding exon, intragenic, and intergenic regions were derived from annotations defined by GENCODE v19 [10], and promoter and flank region annotations were obtained from ENSEMBL [33].

Table 5 details the distribution of annotations applied to the 1.71 million sites considered in the held out biotype training set, as well as the percent of all positive samples that fall within each annotation type. The third column also reports the percentage of samples per each annotation type that are positive. Note that a single site may overlap with more than one annotation. For brevity, here we

Table 4: Test set 2 whole genome results

Sample biotype	ROC AUC	PR AUC
left kidney	0.965	0.778
OCI-LY7	0.899, 0.899, 0.886, 0.886	0.654, 0.654, 0.655, 0.654
prostate gland	0.865	0.516
hindlimb muscle	0.943	0.824
spleen	0.913	0.582
astrocyte	0.919, 0.944	0.787, 0.613
fibroblast of skin of abdomen	0.964	0.826
G401	0.739, 0.846	0.459, 0.516
mean biotype AUC	0.898	0.655
overall AUC	**0.897**	**0.621**

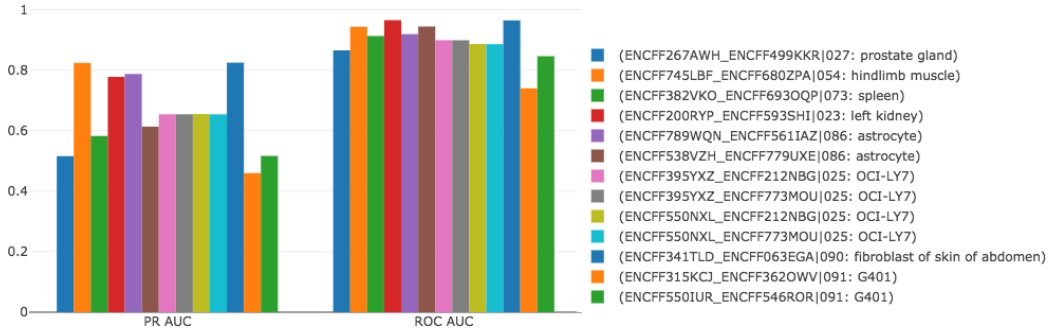

Figure 7: PR AUC and ROC AUC results on the test set of held out biotypes (set 2) shown per whole genome sample. Since ROC AUC is affected by data imbalance, the PR AUC metric is a better evaluation of whole genome performance.

report details of the held out biotypes partition (train/test set 2) only and leave details of set 1 for supplementary material.

Table 5: Distribution of potentially accessible sites by annotation

Site type	Percent of all sites	Percent of all pos examples	Percent per site type that are pos
exon	3.47	5.08	9.74
protein coding exon	2.75	4.11	9.94
intragenic	49.94	45.25	6.04
intergenic	39.89	34.11	5.70
promoter and flank	6.37	30.33	31.75
other	5.39	5.20	6.43

We found that even for samples in which the model performed poorly overall, predictions within promoter and flank regions consistently attained a high level of accuracy (Figure 7, Figure 8, and Figure 10), achieving PR AUC $= 0.838$ over all held out biotypes (test set 2) and PR AUC $= 0.908$ over randomly held out samples (validation set).

We also confirmed that the accuracy of these predictions was independent of whether the promoter and flank sites overlapped with the regions of genes used in our RNA-seq input vector (Figure 9, Figure 10).

Selecting a threshold for classification of only promoter and flank sites such that precision is 80% (20% false discovery rate) on the held out biotype test set, our trained model recalls 65.3% of accessible promoter regions, with a false positive rate of 10%. Applying this same threshold to the validation set where biotypes are allowed to overlap with the training set the model achieves a

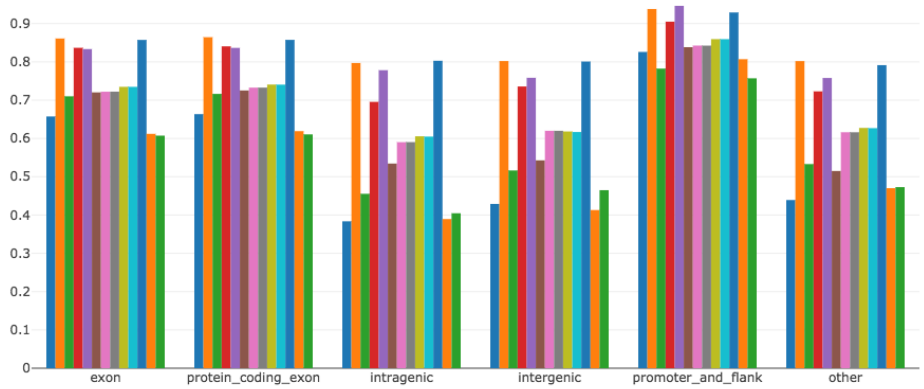

Figure 8: PR AUC results on the test set of held out biotypes (set 2) split per whole genome sample and broken down by genomic site type. Performance on promoter and flank regions was consitently high, even for samples where overall results were lowest. Graph coloring is the same as defined in the legend of Figure 7.

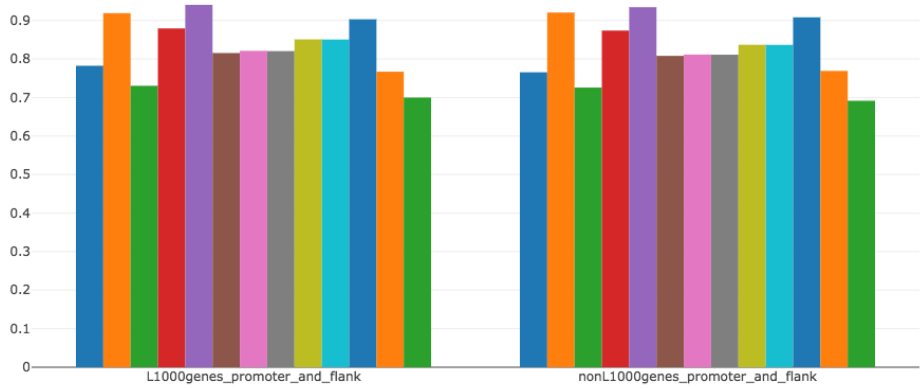

Figure 9: Promoter and flank PR AUC results on the test set of held out biotypes (set 2) split per whole genome sample and broken down by input gene set (L1000) membership. No clear performance difference was observed when promoter and flank regions were split into those that do and do not overlap the RNA-seq input gene set. Graph coloring is the same as defined in the legend of Figure 7

precision of 93.4%, recalling 62.6% of accessible promoter regions, and has a false positive rate of only 3.5%.

3.3 Application to cancer

Once trained, we were able to apply our model to new datasets where whole genome sequence information and RNA-seq were available as illustrated in Figure 1. As a first example we applied our model to 6 cohorts from TCGA in Figure 14: Lung Adenocarcinoma (LUAD), Lung Squamous Cell Carcinoma (LUSC), Kidney Chromophobe (KICH), Kidney Clear Cell Carcinoma (KIRC), Kidney Papillary Cell Carcinoma (KIRP), Breast Cancer (BRCA)

We restricted our predictions on TCGA samples to the subset of potentially accessible sites that overlapped promoter and flank annotations, since performance on those predictions was high across all tests. For consistency with the analysis presented in Sections 3.1-3.2 all TCGA results were obtained by applying our best model trained on set 2 (held out biotypes).

To construct a predicted accessibility profile for each TCGA sample we first applied all somatic SNP, insertion (INS), and deletion (DEL) mutations to any affected sites. However, before looking at

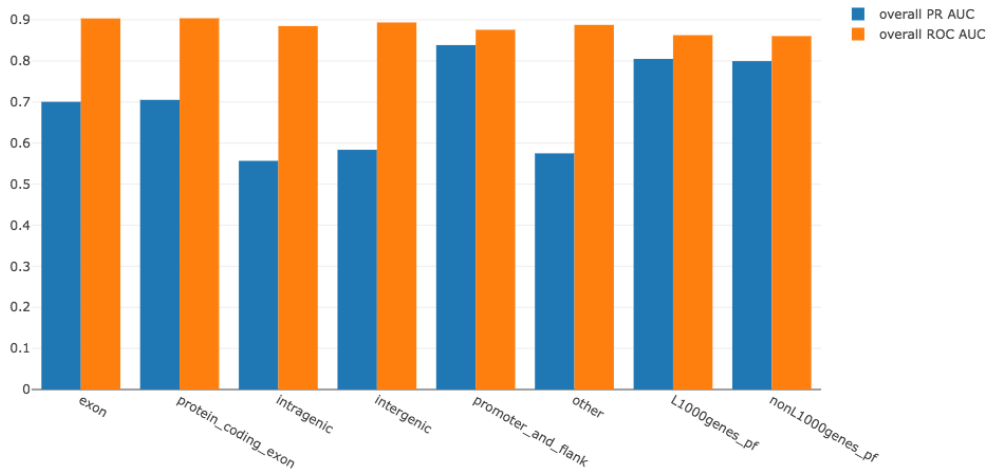

Figure 10: Overall results on the test dataset of held out biotypes (set 2) broken down by site type and L1000 gene set membership.

Table 6: Test set 2 promoter and flank results across whole genomes

Sample biotype	ROC AUC	PR AUC
left kidney	0.949	0.905
OCI-LY7	0.869, 0.868, 0.864, 0.864	0.842, 0.842, 0.859, 0.859
prostate gland	0.897	0.826
hindlimb muscle	0.935	0.938
spleen	0.867	0.782
astrocyte	0.925, 0.914	0.946, 0.838
fibroblast of skin of abdomen	0.951	0.929
G401	0.798, 0.828	0.807, 0.757
overall AUC	**0.876**	**0.839**

the global scope and comparing accessibility profiles, we first sought to understand the impact of mutations on our set of genomic interest regions.

Across all samples in the above cohorts for which we had whole genome data 3172 interest regions had a single SNP, 78 had 2 SNPs, and only 9 regions had between 3 and 5 SNPs. A total of 465 sites included an insertion or deletion (INDEL) mutation, and only 7 sites featured both an INDEL as well as an SNP. Figure 11 shows the total number of mutations per cohort, normalized by each cohort's patient count.

For each sample site affected by at least one mutation, we computed the change in predicted accessibility before and after each type of mutation was applied. Figure 12 shows the distribution of changes due to SNPs only, INDELs only, and all mutations applied across all samples. INDEL mutations showed a larger variance in how much they impacted the accessibility score, which is unsurprising since they typically impact a greater number of base pairs.

Applying the 80% precision threshold, we also investigated how frequently each type of mutation caused accessibility decision changes (Figure 13). Among all mutations that led to changes in classification INS and DEL mutations were the most frequent causes of a decision flip. Notably, of all promoter and flank sites affected by INDELs, 5.46% resulted in changed classification outcomes.

After applying all mutations, predictions from all promoter and flanks sites were stacked into a single vector per sample to form the accessibility profiles for all the samples in our 6 TCGA cohorts. We visualized the relationships between TCGA accessibility profiles using t-SNE in Figure 14, which suggests that looking at cancers from the viewpoint of DNA accessibility offers different

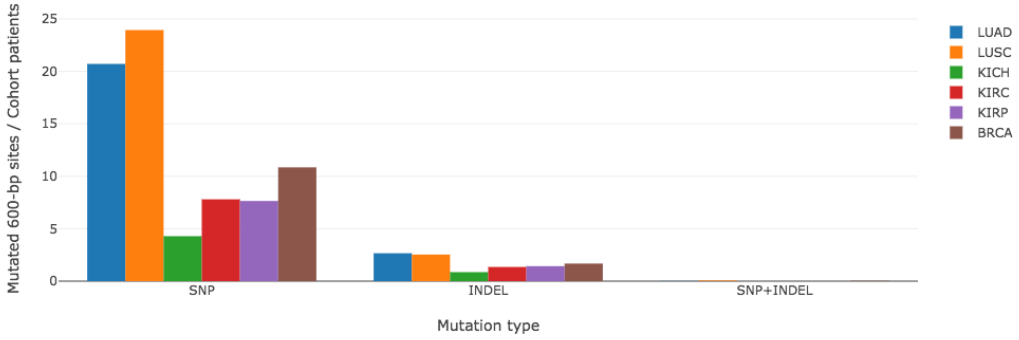

Figure 11: Number of mutated promoter and flank sites normalized by number of patients analyzed per cohort. There were only a total of 7 sites where both SNP and INDEL mutations occurred together (4 in BRCA, 2 in LUSC, 1 in LUAD), they are therefore hardly visible in this plot.

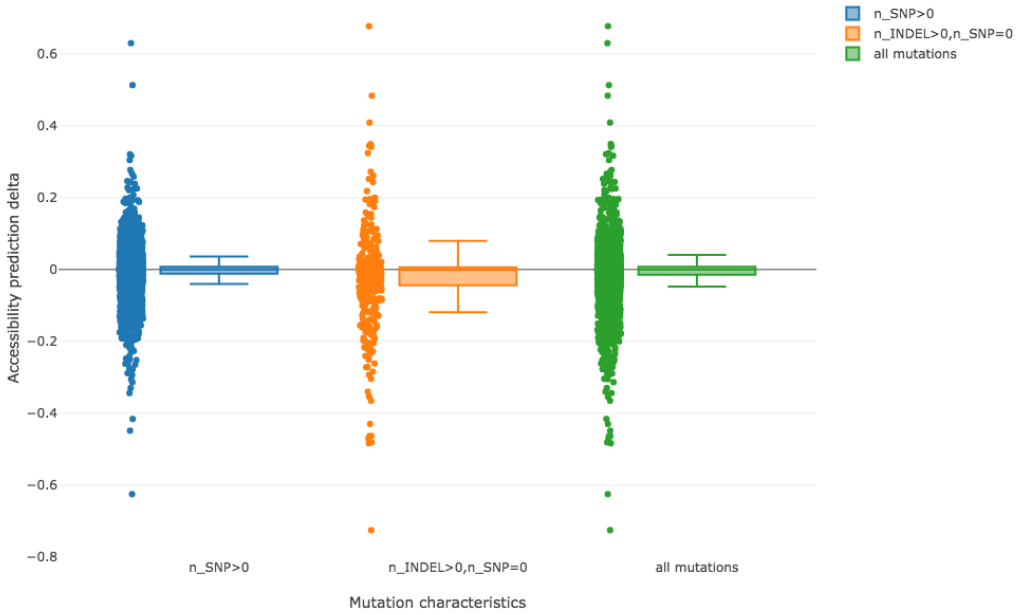

Figure 12: Box plots showing impact of mutations on predicted accessibility score at 600 base-pair promoter and flank sites. The leftmost plot shows the change in predicted accessibility due to regions affected only by SNP mutations (n_SNP>0), followed by regions affected only by INS or DEL mutations (n_INDEL>0,n_SNP=0), and finally all sites where mutations occurred considered together (all mutations). We found that INS and DEL mutations have a broader variance when it comes to impact on accessibility prediction than SNP mutations.

relationships and sub-categories than RNA-seq. Further investigation is ongoing to determine what clinical significance can be obtained from patterns in the pan-cancer accessibility profile space.

4 Conclusions

We have demonstrated that predictive models operating on DNA sequence data can learn to handle cell-specific behavior in a way that allows application to new sample types without re-training. In our models, gene expression from RNA-seq was added as an input to allow learning to exploit cell-type similarity. We showed that these models were capable of achieving consistently high performance for predictions at promoter and flank regions of the genome. This work thus enables a new tool for analysis of tumor genomes across different cell and tissue types and has provided the first glimpse of

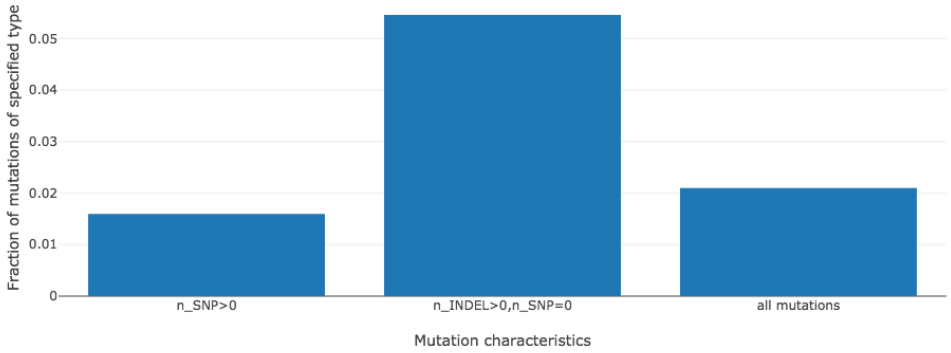

Figure 13: Fraction of mutated sites with a certain category of mutation that end up flipped as a result of considering mutations versus just using hg19/GRCh37 reference genome.

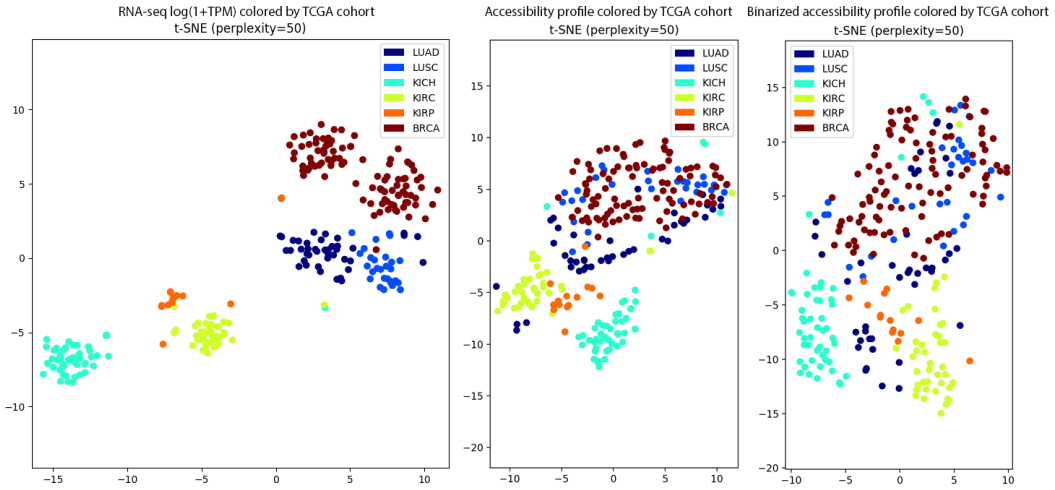

Figure 14: t-SNE [29] visualization of L1000 gene expression level vectors from RNA-seq (left), raw predicted accessibility profile values (center), and binarized accessibility profile after applying the 80% precision threshold (right) in samples from six TCGA cohorts. In the RNA-seq space we see a clear distinction between basal-like versus luminal A/B and HER2-enriched breast cancers (BRCA). In predicted accessibility space (center and right) the lung (LUAD,LUSC) and breast (BRCA) cancer samples appear to have some common accessibility characteristics.

DNA accessibility across TCGA data. Further work is needed to understand the clinical relevance of DNA accessibility predictions across cancer genomes.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M.Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[3] C. Angermueller, H. J. Lee, W. Reik, and O. Stegle. Deepcpg: accurate prediction of single-cell dna methylation states using deep learning. *Genome Biology*, 18(1):67, 2017.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] A. Breschi, S. Djebali, J. Gillis, D. D. Pervouchine, A. Dobin, C. A. Davis, T. R. Gingeras, and R. Guigó. Gene-specific patterns of expression variation across organs and species. *Genome Biology*, 17(1):151, 2016.

[6] A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, M. W. Szcześniak, D. J. Gaffney, L. L. Elo, X. Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):13, 2016.

[7] ENCODE Project Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[8] B. Deplancke, D. Alpern, and V. Gardeux. The genetics of transcription factor dna binding variation. *Cell*, 166(3):538–554, 2016.

[9] GTEx Consortium. The genotype-tissue expression project. https://gtexportal.org.

[10] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, et al. Gencode: the reference human genome annotation for the encode project. *Genome research*, 22(9):1760–1774, 2012.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[14] D. R. Kelley, J. Snoek, and J. L. Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] A. Kundaje. DAC Blacklisted Regions. https://www.encodeproject.org/annotations/ENCSR636HFF/.

[17] J. Lanchantin, R. Singh, Z. Lin, and Y. Qi. Deep motif: Visualizing genomic sequence classifications. *arXiv preprint arXiv:1605.01133*, 2016.

[18] Library of Integrated Network-based Cellular Signatures (LINCS). L1000 genes, GEO accession GPL20573. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL20573.

[19] M. T. Maurano, R. Humbert, E. Rynes, R. E. Thurman, E. Haugen, H. Wang, A. P. Reynolds, R. Sandstrom, H. Qu, J. Brody, et al. Systematic localization of common disease-associated variation in regulatory dna. *Science*, 337(6099):1190–1195, 2012.

[20] D. Quang and X. Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, 44(11):e107–e107, 2016.

[21] Roadmap Epigenomics Consortium et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.

[22] N. C. Sheffield, R. E. Thurman, L. Song, A. Safi, J. A. Stamatoyannopoulos, B. Lenhard, G. E. Crawford, and T. S. Furey. Patterns of regulatory activity across diverse human cell types predict tissue identity, transcription factor binding, and long-range interactions. *Genome research*, 23(5):777–788, 2013.

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

[24] S. Singh, Y. Yang, B. Poczos, and J. Ma. Predicting enhancer-promoter interaction from genomic sequence with deep neural networks. *bioRxiv*, page 085241, 2016.

[25] P. H. Sudmant, M. S. Alexis, and C. B. Burge. Meta-analysis of rna-seq expression data across species, tissues and studies. *Genome biology*, 16(1):287, 2015.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[27] TCGA Research Network. The cancer genome atlas. http://cancergenome.nih.gov/.

[28] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[29] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[30] C. J. Vaske, S. C. Benz, J. Z. Sanborn, D. Earl, C. Szeto, J. Zhu, D. Haussler, and J. M. Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using paradigm. *Bioinformatics*, 26(12):i237–i245, 2010.

[31] Ty C Voss and Gordon L Hager. Dynamic regulation of transcriptional states by chromatin and transcription factors. *Nature Reviews Genetics*, 15(2):69–81, 2014.

[32] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[33] D. R. Zerbino, S. P. Wilder, N. Johnson, T. Juettemann, and P. R. Flicek. The ensembl regulatory build. *Genome Biology*, 16(1):56, Mar 2015.

[34] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature methods*, 12(10):931–934, 2015.