Drawing Biological Understanding From Machine Learning

Forest Yang

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-187
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-187.html

August 30, 2024

Drawing Biological Understanding From Machine Learning

by

Forest Yang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Laurent El Ghaoui, Chair
Assistant Professor Nilah Ioannidis
Associate Professor Adam Yala

Summer 2024

Drawing Biological Understanding From Machine Learning

Abstract

Drawing Biological Understanding From Machine Learning

by

Forest Yang

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Laurent El Ghaoui, Chair

Large biological data, such as medical imaging and single-cell level genomic data, are rich sources of biological information. Machine learning is a tool to extract that information into a usable form, whether it be predictions for some prediction task or insights drawn from the model. We explore three applications of machine learning to biology. One is on using deep learning to perform metastatic cancer prognosis from CT images by predicting lesion-level risks. We use the lesion-level risks to show that the model captures clinically known indicators of risk. Next, we utilize the DeepLIFT and TF-MoDISco neural network interpretation techniques to understand how DNA shape affects transcription factor binding. Overall, we find that sequence features are more important for distinguishing bound sites, but that shape features can modulate binding affinity. Finally, we the test the CellOracle and SCENIC+ gene regulatory network inference frameworks in the context of reprogramming fibroblasts to pluripotent cells, to prioritize key factors in reprogramming and recover their effects on differentiation.

To my friends and family

Contents

List of Figures

List of Tables

Acknowledgments

I thank my advisor, Laurent El Ghaoui, for his support and guidance. Even though I really struggled with research, his input at each meeting gave me a bit more perspective each time, allowing me to mature as a researcher. He pushed me to improve my communication skills and inspired me through his positive attitude and belief in me. I hope I gleaned some of his resourcefulness in tackling problems and his optimization wizardry. I thank Nilah Ioannidis and Adam Yala for being on my dissertation committee, and Nilah for generously welcoming me into her group and setting up the collaboration that became the second project in this thesis. I learned a ton of biology thanks to her. I also thank Adam for his insight on medical imaging models. Lastly, I thank Sanmi Koyejo for being on my qual committee, advising me for a year at UIUC, and continuing to support me.

I thank Jean Nguyen and Shirley Salanio, for fielding my questions throughout the PhD, and making things seem possible during the push to schedule my qual and advance to candidacy. I thank Judy Smithson for doing the same for my dissertation talk and filing my dissertation.

I am grateful to Nilah's group (Ni-lab): Aniketh Janardhan Reddy, Ayesha Bajwa, Daniel Lewinsohn, Eyes Robson, Margarita Geleta, Oberon Dixon-Luinenburg, Pooja Kathail, Ruchir Rastogi, Ryan Chung, Ryan Keivanfar, Sergio Emilio Mares, and Sindy Li, for being very welcoming and teaching me a ton of biology. Also, Ryan Keivanfar was superb to work with in the DNA shape project.

I thank Armin Askari, Fangda Gu, Alicia Tsai, and Zihao Chen for the camaraderie and fun conversations about random topics and PhD life as Laurent's students, which gave me the information and optimism to proceed in the face of uncertainty.

I thank all my other friends: Alan Kwok, Andrew Xu, Justin Lizama, Sabrina Shie, Yiqun Chen, Sinho Chewi, Siqi Liu, Daniel Raban, Jasmine Gass, Richard Li, Natalie Fugate, Keiran Paster, James Ding, Gwen Chang, Chris Ki, Jonathan Xu, Liang Zhou, and others, for their perspectives and fun times eating, cooking, walking around, watching anime and movies, playing video games, tennis, and climbing.

I thank Efe Aras, my housemate for most of my PhD, for the fun trips to different restaurants in Berkeley and SF, always being open to talk, letting me drag him to movies and climbing, playthroughs of games like 13 Sentinels, 9 Hours 9 Persons 9 Doors, and Diablo II (NOT League of Legends. Unless...), and all that I learned from living with him. His presence enabled me to keep going through the lowest points, when any actual progress seemed impossible.

Finally, I thank my brother Lingfeng and his wife Jasmine for their love and concern, and hosting me at their home in South Bay on many weekends. Thanks to my cat Onyx for being so cute and lovable. I thank my mom Xuan and my dad Shudong for their vast love and support, and all the fun experiences they gave me as a kid.

Chapter 1

Introduction

Biology, the study of life, holds deep mysteries that govern how our bodies function, change, and reproduce. Life emerged beginning from cyanobacteria as early as 3.4 billion years ago and has gone through huge developments in diversity and complexity over time [9]. Modern humans have only existed for a small fraction of the time that life existed – around 300,000 years – and the age of human flourishing for an even smaller fraction – around 11,700 years. Biological datasets containing complex information have the potential to illuminate the mechanisms of life that evolved over 3.4 billion years, by having machine learning models extract patterns from them.

Biological data, from radiographic images of the body to genomic and epigenomic assays of cells, are rapidly growing in number, availability, and resolution. The cost of sequencing an entire human genome has gone from $7 million in 2007 to $600 in 2022 [126], and the rise of single-cell data gives an incredibly detailed view of each individual cell. At the same time, the fields of machine learning and data science are booming. The question of how to apply these powerful modeling techniques in the real world is a salient one. Common industrial applications involve using these models, including neural networks, to perform profitable tasks, like advertisement optimization, task automation, and recommendation systems. More ambitious aims such as self-driving, drug discovery, and automatic medical diagnoses are also hotly being pursued, though machine learning approaches for these pursuits are only just being deployed in real applications. Considering the rate of growth of biological data and the amount of effort diverted towards singularly profit-oriented enterprises, there is likely far more that can be done with biological data.

In this thesis, I will focus on methods for understanding biology using machine learning, including methods that use deep neural networks. This involves not only training a deep neural network to accomplish some predictive task, but also interpreting the model to understand what it has learned. Neural networks automatically learn complex patterns that map each input to a prediction of its accompanying output; since networks can contain hundreds of thousands to even hundreds of billions of parameters (by today's standards, a network of a couple million parameters is considered lightweight), they can automatically uncover relationships that a human could not. The network's predictions alone can certainly

be useful, if we want to predict the output on new inputs. However, only using the model for predictions without any interpretation misses out on understanding the basis for the model's predictions. This is dangerous as it could lead to deployment of a model that has only learned incidental patterns that are not truly relevant to the task at hand, and may delay the discovery of novel biological insights already found by the model. Therefore, I also focus on interpreting the models trained in this thesis.

Specifically, we train a deep neural network to perform metastatic lung cancer prognosis by predicting lesion-level risks and aggregating them. Model interpretation using the lesion-level risks shows the model captures relevant risk factors. We also interpret a deep neural network trained on sequence and shape features to predict transcription factor binding, making use of the discriminative patterns learned by the model to derive insights on the role of DNA shape in transcription factor binding. Lastly, we test methods that infer gene regulatory networks from single-cell data on a single-cell dataset containing gene expression and accessibility data for fibroblasts being reprogrammed to induced pluripotent stem cells.

Chapter 2

Lesion Prioritization for Cancer Prognosis

2.1 Background

Cancer prognosis

Cancer is the second-leading cause of death in the US and within cancer, lung cancer is the leading cause of death [108]. Clearly, improving the treatment of this disease is of pivotal importance for people's health. Cancer prognosis refers to providing an outlook on the survival of the patient. The clinical stage of a cancer is already prognostic: for example, stage I lung cancer has a 5-year survival rate of about 80%, while stage III lung cancer has a 5-year survival rate of about 25% [81]; more granular stages such as stage IB can provide a more precise prognosis. Prognosis is useful for informing the patient and informing what kind of treatment to give the patient; poorer prognoses may call for more intensive treatment regimens. The good news is that survival rates for cancer are steadily improving due to the constant development of new and improved treatments via clinical trials. Computational models of prognosis can play a role in accelerating treatment development and therefore increases in survival rate.

TNM staging Clinical stage is obtained through a system called *TNM staging* [98]. T, N, and M represent characteristics of the cancer that are individually measured and then combined to produce an overall stage. T (Tumor) describes the size and level of invasion into adjacent tissues of the primary tumor, and ranges from T0 to T4. N (Node) denotes the extent that the cancer has spread to nearby lymph nodes, which are nodes in the body that filter fluid in the body for disease, and this measure ranges from N0 to N3. M (Metastasis) indicates whether the cancer has metastasized, or spread to a different location, with M0 denoting no metastasis and M1 denoting metastasis. The combination of these characteristics is combined into a stage from I through IV. For example, a T1N1M0 classification might

result in a stage II diagnosis. The most severe diagnosis, stage IV, is synonymous with an M1 classification, or metastatic cancer.

RECIST and treatment response assessment A related notion to prognosis is treatment response assessment. The goal here is to determine whether the cancer is shrinking or weakening in response to treatment. The clinical standard for this is RECIST 1.1 [84], which compares the proportional change in sum of diameters of up to 5 lesions chosen by a radiologist. A decrease in the sum of diameters by $\geq 30\%$ corresponds to treatment response, while an increase by $\geq 20\%$ corresponds to progressive disease. In the interest of implementing potentially life-saving drugs as soon as possible, treatment response and disease progression defined according to RECIST are FDA-recognized endpoints for assessing the efficacy of drugs in clinical trials [99], because survival time, while the gold standard of efficacy, takes a long time to observe.

However, the RECIST criterion suffers from inter-reader variability (estimated 95% confidence interval: $[-18.6\%, 25.4\%]$ around the measured change [86]) due to measurement error and some arbitrariness in the choice of lesions. Furthermore, novel therapies like molecular targeted therapy and immunotherapy produce different patterns of response in cancer that may call for different criteria [85]. For example, in gastrointestinal stromal tumors treated with a tyrosine kinase inhibitor, the Choi criteria, which also incorporate tumor density, were shown to have better correlation with survival than RECIST [100]. By learning to perform prognosis from data, a neural network could extract better measures of survival that are tuned to a particular combination of cancer and treatment type.

Survival analysis

A good reference for survival analysis is [1].

Main definitions Survival analysis is the branch of statistics that deals with estimating the time to an event, T. In oncology, the event considered is usually death, recurrence of disease, or progression of disease; cancer prognosis is simply survival analysis applied to cancer patients. The main object of interest is the *survival function*, $S(t)$, defined as the probability of surviving past time t:

$$S(t) := \mathbb{P}(T > t).$$

The standard setup is that we observe n survival times $t_1, t_2, \ldots, t_n \in \mathbb{R}$, potentially accompanied by feature vectors $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$. Furthermore, some of the times are *censored*, meaning they do not truly denote the time to the event, but represent a time when the subject dropped out of the study, since subjects do not always remain under observation. We let $\delta_1, \delta_2, \ldots, \delta_n$ denote whether the event was observed for the ith subject:

$$\delta_i = \begin{cases} 1 & \text{event was observed for subject } i \\ 0 & \text{subject } i \text{ was censored.} \end{cases}$$

Censoring is assumed to be independent of survival times for statistical reasons. Were it not for censoring, we could estimate $S(t)$ with the empirical survival function: $\hat{S}(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{t_i > t\}$. However, in the presence of censoring, this is an underestimate, because the true survival time of a censored subject i may be greater than t despite their censoring time t_i being less than t.

An object that is more convenient for estimating survival when there is censoring is the *hazard function*, $\lambda(t)$. This is defined as the density of the event occurring at time t conditioned on survival up until time t. Thus, assuming that T is continuous with density $f(t)$, we have:

$$\lambda(t) = \lim_{\Delta t \downarrow 0} \frac{\mathbb{P}(t \le T < t + \Delta t | T \ge t)}{\Delta t} = \lim_{\Delta t \downarrow 0} \frac{\mathbb{P}(t \le T < t + \Delta t)}{S(t)\Delta t} = \frac{f(t)}{S(t)}.$$

Note that the hazard function $\lambda(t)$ and survival function $S(t)$ are equivalent, in that one may be obtained from the other. To obtain $\lambda(t)$ from $S(t)$, we can observe that since $f(t) = \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}(T \le t) = \frac{\mathrm{d}}{\mathrm{d}t}(1 - S(t)) = -S'(t)$, we have

$$\lambda(t) = \frac{-S'(t)}{S(t)}.$$

To obtain $S(t)$ from $\lambda(t)$, we note that

$$\lambda(t) = \frac{-S'(t)}{S(t)} = -\frac{\mathrm{d}}{\mathrm{d}t}\log(S(t)) \implies \log(S(t)) - \log(S(0)) = -\int_0^t \lambda(s)\,\mathrm{d}s$$
$$\implies S(t) = e^{-\int_0^t \lambda(s)\,\mathrm{d}s} = e^{-\Lambda(t)},$$

because $S(0) = \mathbb{P}(T > 0) = 1$ and $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$ denotes the cumulative hazard function. Note that a constant hazard $\lambda(t)$ corresponds to an exponential distribution with rate parameter λ: $f(t) = \lambda e^{-\lambda t}$.

Basic estimators: Kaplan-Meier estimator and Cox model The *Kaplan-Meier estimator* [55] is a nonparametric estimator of $S(t)$ using $\{(t_i, \delta_i)\}_{i=1}^n$. Features $\{x_1, \ldots, x_n\}$ are not used. It is based on the idea that

$$S(t) = \mathbb{P}(T > t) \approx \mathbb{P}(T > t_{(j)}) = \mathbb{P}(T > t_{(1)})\frac{\mathbb{P}(T > t_{(2)})}{\mathbb{P}(T > t_{(1)})} \cdots \frac{\mathbb{P}(T > t_{(j)})}{\mathbb{P}(T > t_{(j-1)})},$$

where $t_{(i)}$ is the ith smallest survival time, $t_{(j)}$ is the greatest t_i such that $t_i \le t$, and for simplicity we assume there are no ties. $\frac{\mathbb{P}(T > t_{(i)})}{\mathbb{P}(T > t_{(i-1)})} = \mathbb{P}(T > t_{(i)} | T > t_{(i-1)})$ is the probability of survival past time $t_{(i)}$ given survival past the previous time point $t_{(i-1)}$. A natural estimate of this quantity is the number surviving at time $t_{(i)}$ divided by the number surviving just before time $t_{(i)}$: $1 - \frac{\delta_{(i)}}{n-i+1}$, where $\delta_{(i)}$ is the event indicator corresponding to time $t_{(i)}$. Thus,

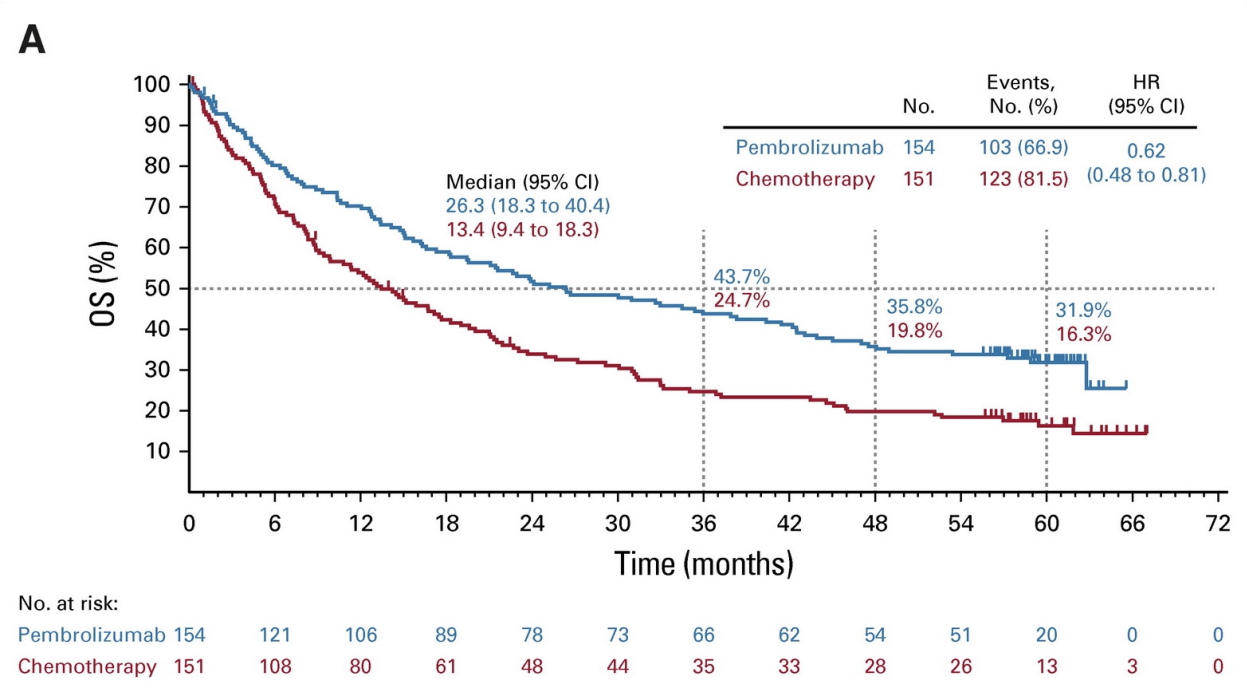


Figure 2.1: Kaplan-Meier curves from a recent study (2021) comparing immunotherapy and chemotherapy applied to patients with non-small-cell metastatic lung cancer [93]. As shown by the estimated survival curves, outcomes are better with the immunotherapy.

the Kaplan-Meier estimator is

$$\hat{S}(t) = \prod_{t_{(i)} \leq t} \left(1 - \frac{\delta_{(i)}}{n - i + 1}\right).$$

Censored individuals contribute to the denominator of the fractions at times when they are in the study, and do not contribute to the denominator at times when they have left the study. This still makes use of censored individuals to estimate $S(t)$, and its derivation is motivated by the hazard $\lambda(t)$. Example Kaplan-Meier curves for lung cancer patients treated with Pembrolizumab and chemotherapy[93] are shown in Figure 2.1.

Given additionally features that are associated with each subject $\{(t_i, \delta_i, x_i)\}_{i=1}^n$, the proportional hazards assumption states that the features affect the hazard function multiplicatively based on some parameter $\beta \in \mathbb{R}^d$:

$$\lambda(t|X) = \lambda_0(t)e^{\beta^\top X}$$

where we have additionally specified the form of the multiplicative factor as $e^{\beta^\top X}$. In a seminal work [25], Cox proposed to maximize the *partial log-likelihood* function in order to

find β:

$$L(\beta) = \prod_{i=1}^{n} \frac{\delta_{(i)} e^{\beta^\top x_{(i)}}}{\sum_{j=i}^{n} e^{\beta^\top x_{(j)}}}. \tag{2.1}$$

Together with the proportional hazards assumption, this constitutes the celebrated Cox model of survival analysis.

Deep learning for survival analysis

General approaches A simple approach to performing survival analysis with deep neural networks is to crudely convert the problem to a binary classification problem, a common problem type in deep learning, by binarizing survival times according to some cutoff, for example, the median [70], or 2 years [60]. A downside is that this approach throws away censored data that was censored before the cutoff. Another simple approach is to directly regress against survival times [118], though again this does not properly handle censoring. Some existing approaches that are more tailored to survival analysis are:

Faraggi-Simon, DeepSurv, Cox-nnet [32, 57, 21]: The linear model inside the exponential function in the Cox model is replaced with a neural network. Thus, instances of $e^{\beta^\top x_i}$ become $e^{f_\theta(x_i)}$ where f_θ is a neural network. The partial log-likelihood following this replacement is optimized with gradient descent.

Batched PLL [66]: For the individuals in the sum in the denominator of the partial log-likelihood (2.1), Kvamme, Borgan, and Scheel (2019) propose to only include that are in the current batch, and provide mathematical justification for this approach. This allows dealing with large datasets and larger architectures, which cannot all fit in a single batch.

Nnet-survival [38]: Gensheimer and Balasubramanian (2019) propose a survival loss function which divides the entire time interval into smaller sub-intervals and considers the relationship of the survival time to each sub-interval.

Evaluation metrics The dominant metric for evaluating survival analysis models is C-index. It can be stated as follows:

$$\text{C-index} = \frac{\text{concordant pairs}}{\text{concordant pairs} + \text{discordant pairs}}. \tag{2.2}$$

A concordant pair is a pair of data points whose outcome ordering agrees with its predicted outcome ordering. A discordant pair is a pair of data points whose outcome ordering disagrees with its predicted outcome ordering. The outcome is a survival time t_i. The prediction could be a risk score, s_i (it could also be a predicted survival time, in which case we would reverse the following comparisons between s_i and s_j). i and j are concordant if $t_i > t_j$ and $s_i < s_j$. They are discordant if $t_i > t_j$ and $s_i > s_j$. The C-index can be interpreted as the probability that if we draw a random pair of subjects, the prediction agrees with reality, i.e.

$$\text{C-index} = \mathbb{P}(s_i < s_j | t_i > t_j).$$

Thus, a C-index of 0.70 can be interpreted as the model being right 70% of the time on a random pair of patients.

Another survival analysis metric is the Brier score, and both Brier score and C-index can be adjusted for censoring using the inverse-probability-of-censoring weighting (IPCW) [27] though at least in image-based deep learning for cancer prognosis, these have not seen significant use.

A related metric to C-index is the area under the receiver operating characteristic curve, or AUROC, often further abbreviated to AUC. While C-index deals with real-valued outcomes, the AUC deals with binary outcomes, but can be understood as a special case of C-index in the binary case. In fact, the AUC is known to equal to the probability that a randomly chosen positive sample has a higher prediction than a randomly chosen negative sample:

$$\text{AUC} = \mathbb{P}(p_i > p_j | y_i = 1,\, y_j = 0).$$

Setting $y_i = t_i$ and $s_i = -p_i$, we exactly get back the probability definition of C-index. Thus, AUC is a special case of C-index with binary survival times.

Cancer imaging Prior works in deep image-based cancer prognosis examine CT scans containing images of cancer. Most deal with early stage cancer, in which the cancer has not really spread beyond the primary lesion. Thus, their modus operandi is to extract an image crop containing the primary lesion with some border, and to feed it to a convolutional neural network (or potentially, a vision transformer, though this has not been explored as much) to predict survival. Some works deal with 2D input patches, others deal with 3D input patches. Also, some works handle more than one time-point, in which case a crop of the lesion at each time point is extracted, and each is converted to a vector embedding, with a recurrent neural network such as an LSTM being used to integrate information from embeddings across timepoints. A previous work [70] that, like this thesis, deals with metastatic cancer, considers lesions of at least 1cm in diameter and performs a size-weighted average of their embeddings, and feeds that to their classification head to obtain their prediction. However this results in them only considering up to 10 lesions per patient. Table 2.1 summarizes prior work.

Goal and approach

Our goal is to design a neural network that functions similarly to a radiologist performing TNM staging or using RECIST – it inspects radiographic images of cancer, typically a CT scan, and based on the features of the cancer, outputs a measurement correlated with the future survival of the patient. Furthermore, we aim to bake in an inherent degree of interpretability to the network, as interpretability is especially important in medical applications. We focus on metastatic cancer, where a patient can have numerous lesions, because this is a challenging and underexplored problem – most prior works applying deep learning to cancer prognosis focus on stages I-III cancer, inputting only one lesion to the network (Table 2.1).

Ref	Cancer	Treatment	N_P	N_L	N_T	Model	Loss	AUC / C-index
[47]	Lung, Stage I-III	Radio-therapy	771	1	1	Custom 3DCNN	Binary (2 year)	0.70 AUC
[127]	Lung, Stage III	Radio– & chemo-therapy	179	1	2–4	ResNet +GRU	Binary (2 year)	0.74 AUC
[60]	Lung, Stage I-II	Surgery	800	1	1	Custom 3DCNN	Nnet-survival[38]	0.74 C
[70]	Colorectal, Stage IV	Chemo– & targeted therapy	1028	1–10	2–4	Inception +BiLSTM	Binary (1 -year)	0.649 C
[130]	Pancreatic*	Unknown	205	1	1	3D-ResNet18 +LSTM	NPLL	0.683 C

Table 2.1: Previous approaches on deep cancer prognosis from images. N_P, N_L, N_T stand for the number of patients, number of lesions considered per patient, and number of scans taken at different time points considered respectively. If the model has "3D" in its name, inputs are 3D lesion-centered CT patches. Otherwise, inputs are 2D patches. *: used contrast-enhanced CT.

Thus, we develop the framework of lesion prioritization. This involves simultaneously predicting survival and identifying high risk lesions. The philosophy behind this approach is that the survival-based training objective can be exploited to automatically guide assignment of risk scores to lesions, and in turn, the lesion-level risk scores lend interpretability to the model. We implement this approach on synthetic and real data, showing in the synthetic case that the approach can successfully recover which lesions are defined as malignant, and in the real case, we show that our approach outperforms deep learning approaches on a dataset of about two hundred patients, and we derive insights by interpreting the lesion-level risk scores.

2.2 Outcome-aware Object Detection on Synthetic Lesions

Introduction

We seek a method to, given radiology scans, simultaneously localize tumors and provide a survival prognosis for each patient. One lens through which to view this task is *outcome-*

aware object detection: can we detect tumors in such a way that makes use of our survival predictions? Conversely, we can also view the task as *object-aware outcome prediction*: performing a survival prognosis that is informed by identified tumors.

To test approaches in a controlled environment, where we know exactly how much a given tumor affects survival time, we model tumors as randomly generated ellipses with green ellipses being designated as benign and red ellipses as malignant. The survival time is modeled as an exponential random variable whose parameter is determined by the number and area of the red ellipses in the image. We further simulate the difficulty of the real life problem by adding noise to the images to make them resemble medical images more. Having multiple tumors or lesions per image models metastatic cancer.

To tackle this task, we make use of a classic object detection architecture, the region proposal network from Faster R-CNN [94], which achieved state of the art object detection performance and very fast prediction speed by predicting bounding box classification and regression outputs across the entire image by using different position-wise vectors from the same convolutional encoding. In addition to the existing bounding box classification and regression heads, we introduce two additional heads, an *outcome-relevant* bounding box classification head, and a risk head. We train the network to perform two tasks: object detection (bounding box classification and regression losses), and survival prognosis (negative partial log-likelihood loss). Furthermore, we apply L1 regularization to the outputs of the outcome-relevant classification head, so that it assigns probabilities of 0 to non-outcome-relevant objects and thus acts as a relevant object selector.

We experiment with different weightings for the components of our loss function. On the clean data, we find that for a wide range of weightings, the model's predicted risks have C-index close to the optimum, which is achieved by the ground truth risks that generated the survival times. With a reasonable weighting on the object detection component of the loss and the L1 penalty, the model achieves near perfect object detection performance and "outcome-relevant" object detection performance. In the presence of noise, survival prognosis and object detection performance are well-preserved, but the outcome-relevant object detection performance drops noticeably. This underscores the challenge of learning to detect the more survival-relevant lesions in complex, noisy medical images.

Synthetic data construction

Clean image construction Each clean image consists of red and green ellipses randomly scattered against a white background. To generate the ellipses for an image, first, the number of ellipses is sampled from a Poisson(5) distribution. The color of each ellipse is randomly selected to be red or green with equal probability. The center of each ellipse is chosen uniformly at random from the square $[-4, 4] \times [-4, 4]$. The width is drawn from a $N(0.7, 0.0625)$ distribution truncated from below to ensure non-trivial size, and the height is set to be a Uniform$(0.5, 1.5)$ variable times the width. The orientation, or angle of the ellipse is drawn uniformly at random from $[0, 360°)$. Finally, the ellipses are drawn into the image one-by-one with these parameters selected randomly in this way independently of previous

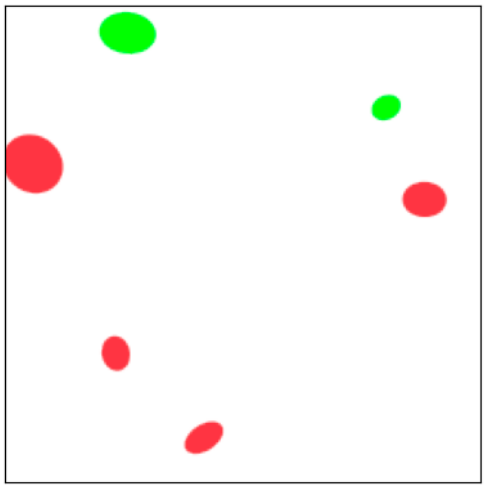


Figure 2.2: Example clean image of 7 randomly generated lesions.

ellipses, but if the current ellipse overlaps another ellipse, the current ellipse is resampled. In the unlikely event this sampling repeatedly fails due to the previous ellipses covering most of the available space, the entire sampling is restarted. An example of an image randomly generated in this manner is shown in Figure 2.2.

Survival model We model the survival time for each image using a proportional hazards assumption with constant baseline hazard λ_0 and a multiplicative factor, or hazard ratio, that only depends on the red ellipses in the image.

$$\lambda(t \mid \text{Image}) = \lambda_0 e^{\beta_1 n_{\text{red}} + \beta_2 A_{\text{red}}}. \tag{2.3}$$

This causes the survival time of each image to be exponentially distributed:

$$T \mid \text{Image} \sim \text{Exp}(\lambda_0 e^{\beta_1 n_{\text{red}} + \beta_2 A_{\text{red}}}).$$

n_{red} is the number of ellipses that are red, and A_{area} is the fraction of the area of the image that is red. We set $\beta_1 = 0.5$ and $\beta_2 = 20$. For each image, we set its survival time by sampling from its survival time distribution. The choice of the baseline hazard λ_0 simply determines the scale of the survival times and therefore does not affect results; scaling λ_0 by a constant c corresponds to scaling the sampled survival times by $\frac{1}{c}$. The resulting C-index between sampled survival times and ground truth hazards (2.3) was 0.761, and represents the best possible C-index a predictor could get.

Based on the ground truth values of $\beta = [\beta_1 \ \beta_2]^\top$, we can interpret each additional red ellipse as multiplying the hazard by $e^{0.5} = 1.65$, and each 1% of the image that is red as multiplying the hazard by $e^{0.2} = 1.22$. Note that the survival time is independent of the green ellipses in the image. This encodes the fact that precisely the red ellipses are the malignant lesions in the image, responsible for decreasing the patient's survival time.

Adding noise to clean images To obtain more realistic looking images, we perform the following steps to generate noisy images from clean images. In Figure 2.3 which demonstrates the process, the shown ellipses are larger than they would be in the actual dataset, as hopefully this makes the transformations easier to visualize.

1. Grayscale: for a particular color, either red or green, we extract only the lesions of that color and convert to a grayscale image, where the lesions have the value 1 and the background 0.

2. Jumble: we fray the edges of each ellipse by applying the following "jumble" filter:

$$C = \frac{1}{45}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 3 & 3 & 1 \\ 1 & 3 & 5 & 3 & 1 \\ 1 & 3 & 3 & 3 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

   Note that the entries of C define a probability distribution. We apply C by replacing each pixel with a random pixel in the 5×5 grid around it according to the distribution defined by C. This can be thought of as a "sampling" version of a convolution operation.

3. Smooth: we smooth things over by applying C again, this time as a standard convolutional filter.

4. Gaussian noise: we add correlated Gaussian noise to the image for texture. We set the covariance between two pixels to be $\sigma^2 \exp(-\|x - x'\|^2/5)$ where x, x' are the locations of the two pixels. For efficiency, for a given pixel, we only consider correlations in a 9×9 region around it, and during implementation, we generate each 3×3 block of the image at a time, scanning across the first row of 3×3s, then going to the next row, etc. When generating each 3×3 block, we sample from the Gaussian defined above conditioned on the values of the blocks to its left, above, and above left, which have already been generated. We set the mean intensity of red ellipses to 450 and the mean intensity of green ellipses to 400, and vary the noise parameter σ^2.

Architecture and losses

We modify the region proposal network of [94], adding to the existing classification head β and regression head $\hat{\beta}$, the outcome aware classification head $\hat{\beta}$ and risk prediction head Z. To review, at the end of some convolutional encoder, a feature map of dimension (a, f, h, w) is output, where (h, w) are the spatial dimensions, a is the number of "anchor boxes" (base bounding boxes per position, a concept introduced by [94]), and f is the number of features at each anchor box $\times$ position. Each head operates on the f-dimensional embedding at each

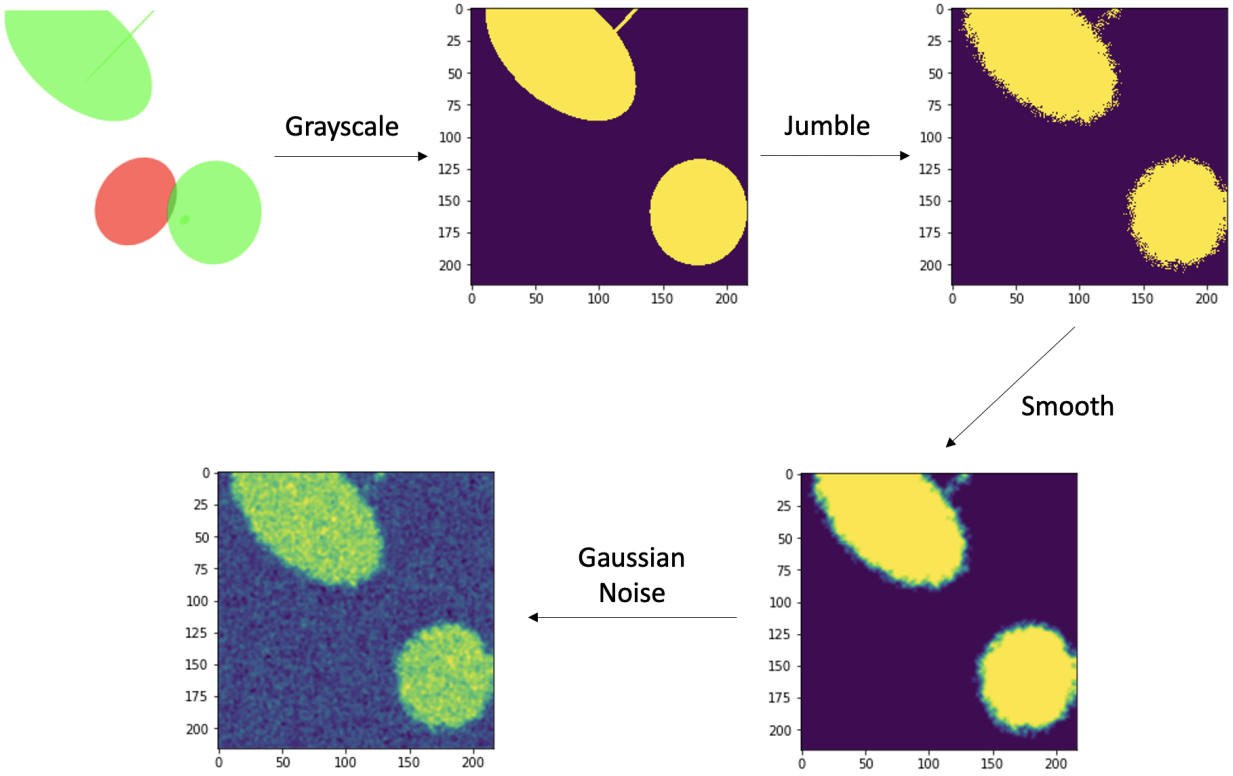


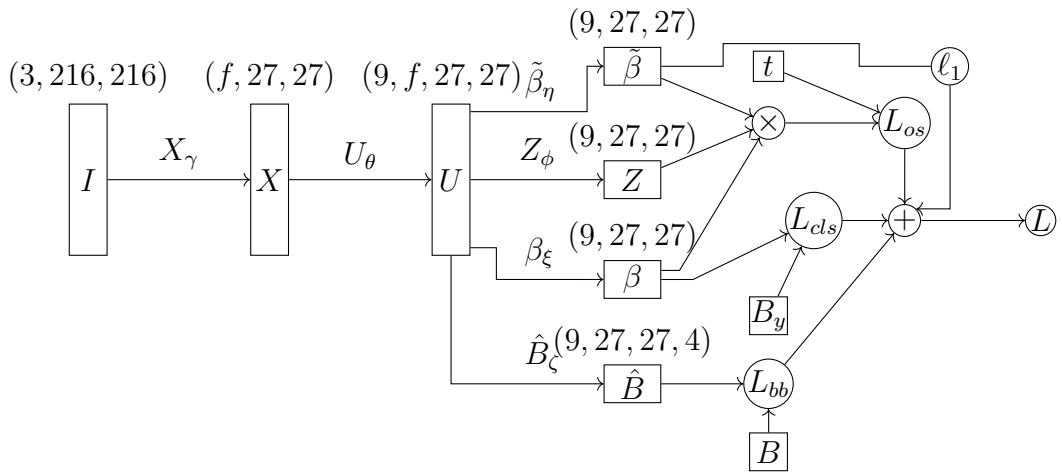
Figure 2.3: Demonstration of noise-adding procedure to images of synthetic lesions. The process is done for each color, red and green, individually, and the results are summed together. The process is shown for the green lesions in the above image.

anchor box × position individually, producing a prediction for each anchor box × position. β predicts the probability of an actual object being present. $\hat{B}$ predicts precise bounding box coordinates. $\tilde{\beta}$ predicts the probability of an *outcome-relevant* object being present. Z predicts the risk score of the object in the bounding box. A schematic of the resulting network is shown in Figure 2.4.

Our loss function is shown below:

$$L = L_{os}(\langle Z, \beta \odot \tilde{\beta} \rangle, t) + \alpha(L_{bb}(\hat{B}, B) + L_{cls}(\beta, B_y)) + \mu\|\tilde{\beta}\|_1. \tag{2.4}$$

L_{os} is the negative partial log-likelihood loss, restricted to the samples in the current batch (as justified in [66]). The overall predicted risk for an image (the log hazard ratio, analogous to $\beta^\top x$ in the Cox model $\lambda(t \mid x) = \lambda_0(t \mid x)e^{\beta^\top x}$) is generated by $\langle Z, \beta \odot \tilde{\beta} \rangle$. In other words, we multiply the risks Z with the object probabilities β and the outcome-aware object probabilities $\tilde{\beta}$ position-wise, then sum it all up. L_{bb} is the regression loss based on log-transformed bounding box coordinates relative to the current anchor box from [94], and L_{cls} is a simple binary classification loss – the sum of these two components is the object



$$L = L_{os}(\langle Z, \beta \odot \tilde{\beta} \rangle, t) + \alpha(L_{bb}(\hat{B}, B) + L_{cls}(\beta, B_y)) + \mu\|\tilde{\beta}\|_1.$$

Figure 2.4: Schematic of outcome aware object detection network used on synthetic lesion data. In our implementation, the convolutional encoder, which maps $I \to U$, contains 8 convolutional layers, similar to a VGG, each head contains 1-3 position-wise fully connected layers, and we set the number of channels to $f = 256$. The dimensions output by each layer reveal the amount of downsampling at each stage.

detection component which is weighted by a parameter α. Then, apply an L1 penalty on $\tilde{\beta}$ weighted by a parameter μ, encouraging $\tilde{\beta}$ to be as sparse as possible. Since the only other place $\tilde{\beta}$ appears is $\langle Z, \beta \odot \tilde{\beta} \rangle$, $\tilde{\beta}$ will ideally be 0 wherever β is 0, because the result at that position would be 0 regardless. Thus, $\tilde{\beta}$ can be viewed as selecting a subset of the objects selected by β. Additionally, $\tilde{\beta}$ will be encouraged to be 0 at objects that do not contribute to the true risk, because accurate risk predictions need not assign weight to irrelevant objects. It will not be able to be 0 at relevant objects, because this would harm the survival prediction performance. Together, this encourages $\tilde{\beta}$ to output the probability (or if the interpretation of "probability" at this point has been marred, a nonnegative weight in $[0, 1]$, a "psuedo-probability") of an outcome-relevant object being present at each position.

Results

To test our approach, we train our model on a train set of 3,000 images and a validation set of 2,000 images and evaluate on a test set of 10,000 images. The inputs to the loss (2.4) are the outputs of our architecture, $\beta, \hat{B}, \tilde{\beta}$, and Z, which are derived solely from the input image I, the ground truth bounding box information B (coordinates) and B_y (presence labels), and survival times t, for all the samples in the batch. We use Adam [62] with a constant

learning rate of 0.001. We first evaluate our approach on clean data to get a sense of how our approach performs in an idealized, noiseless setting on three tasks: survival prognosis, object detection, and outcome-relevant object detection. Furthermore, we vary α, the object detection weight, and μ, the L1 penalty weight, from 0.1 to 300, in order to see how different weightings of loss function components affect performance on different tasks. The evaluation metrics used are:

1. C-index: the standard survival analysis metric, as defined in (2.2).

2. mAP@50: the mean average precision at 50, a classic object detection metric used in standard benchmarks [30, 69]. It computes the average precision of bounding boxes ranked by object probability to detect ground truth bounding boxes, counting true positives as boxes with an IOU of ≥ 0.5 with a ground truth box. After a predicted box has been assigned to a ground truth box, the ground truth box is removed. We use non-maximal suppression to prune the list of predicted bounding boxes. Since we only have one object class (lesion), mAP is equivalent to AP.

3. Red mAP@50: the *red* or *relevant* mean average precision at 50. This replaces the ground truth bounding boxes with the bounding boxes of the red ellipses, which are the relevant objects. Instead of using predicted object probabilities (β) to rank bounding boxes, we use predicted relevant object probabilities ($\tilde{\beta}$).

Our results on clean data are summarized in Table 2.2. The top table shows the performance when $\mu = 0$, i.e. there is no L1 penalty, and α is varied from 0.1 to 300. As expected, for very low values of α ($\alpha = 0.1, 1$), the object detection performance as exhibited by mAP@50 suffers, because it is being overwhelmed by the survival prediction task. For very high values of α ($\alpha = 100, 300$), the prognosis performance drops slightly, from a C-index of ≥ 0.760 to 0.757, showing that too high of a weight on object detection does interfere with prognosis, but only slightly. For a sweet spot of $\alpha \in \{10, 30, 50\}$, both very high prognosis performance (essentially hitting the theoretical maximum 0.761 on this data) and detection performance (mAP@50 $= 0.99$) are achieved. Another observation is that since there is no L1 penalty, $\tilde{\beta}$ has no motivation to select outcome-relevant objects, so the performance at outcome-relevant object detection measured by red mAP@50 is poor.

The bottom half of Table 2.2 analyzes the effect of varying μ from 0.1 to 300, with $\alpha = 50$ fixed. We observe that for low values of μ ($\mu \in \{0.1, 1, 10\}$), $\tilde{\beta}$ learns to distinguish relevant objects (red ellipses) from non-relevant objects (green ellipses) very well without disturbing performance on the other two tasks, as reflected by red mAP@50 values of close to 0.99. Moreover, it has done so through learning $\tilde{\beta}$ to minimize the negative partial log-likelihood with risk formulation $\langle Z, \beta \odot \tilde{\beta}$ while putting an L1 constraint on $\tilde{\beta}$, as opposed to explicitly supervising $\tilde{\beta}$ with which objects are relevant. For higher values of μ, e.g. $\mu \in \{100, 300\}$, counterintuitively, the red mAP@50 performance drops to below 0.7. An explanation could be that too high of an L1 penalty forces $\tilde{\beta}$ to become 0 even on relevant objects, though interestingly, the C-index does not suffer at this point. Alternatively, perhaps the extreme

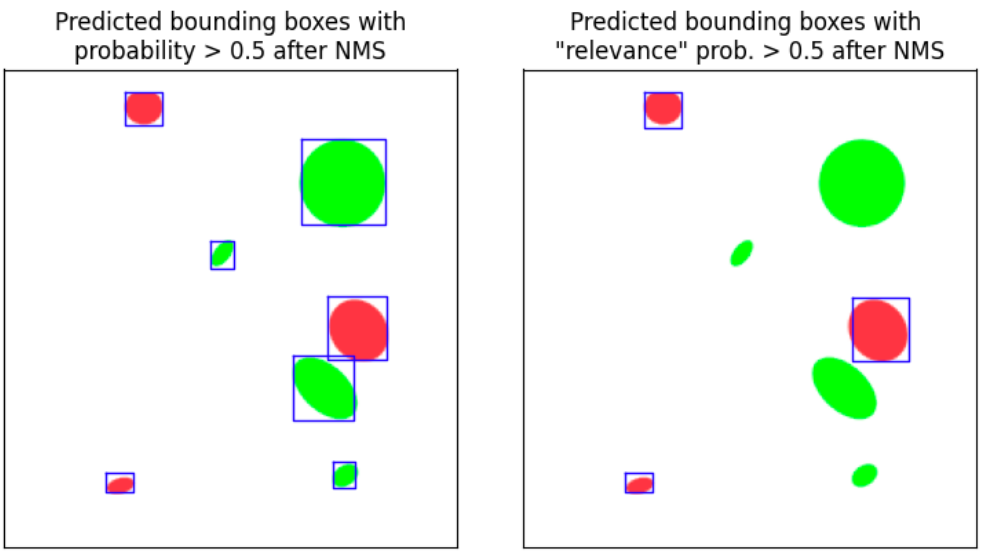


Figure 2.5: Example detection by a model with $\alpha = 50, \mu = 50$.

L1 penalty causes $\tilde{\beta}$ to behave in erratically in some sense, where it loses the ability to properly order locations in the image according to their probability of containing an object. Somehow, this is compensated for by Z, so the prognosis performance stays the same. An example detection and outcome-aware detection for the same input by the $\alpha = 50, \mu = 50$ model are shown in Figure 2.5.

	$\mu = 0$						
	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 10$	$\alpha = 30$	$\alpha = 50$	$\alpha = 100$	$\alpha = 300$
C-index	0.762	0.762	0.761	0.760	0.760	0.757	0.757
mAP@50 (β)	0.158	0.909	0.989	0.990	0.990	0.990	0.990
Red mAP@50 ($\tilde{\beta}$)	0.036	0.115	0.061	0.158	0.187	0.114	0.073

	$\alpha = 50$						
	$\mu = 0.1$	$\mu = 1$	$\mu = 10$	$\mu = 30$	$\mu = 50$	$\mu = 100$	$\mu = 300$
C-index	0.760	0.757	0.760	0.760	0.758	0.759	0.759
mAP@50 (β)	0.990	0.990	0.990	0.990	0.990	0.989	0.989
Red mAP@50 ($\tilde{\beta}$)	0.987	0.984	0.989	0.705	0.989	0.687	0.638

Table 2.2: Survival prognosis, object detection, and outcome-relevant object detection results on clean data.

Next, we investigate the performance of the $\alpha = 50, \mu = 50$ network on noisy images at different noise levels ($\sigma^2 \in \{100, 500, 1000, 5000, 10000\}$). An example of each level of noise as well as plots showing the performance on each task for each level of noise are shown in Figure 2.6. Performance drops on all three tasks, but at worst the C-index drops from about 0.760 to about 0.750, and mAP@50 from 0.99 to 0.988. In comparison, the red mAP@50 drops precipitously, from about 0.99 to under 0.65. This points to an interesting conclusion: when a task has direct supervision, as in the case of the prognosis and object detection tasks, its performance is relatively robust to noise. However, when a task is supervised indirectly or semi-supervised, as is $\tilde{\beta}$, which is meant to learn to predict relevant-object-probability through survival prediction plus an L1 penalty, performance on it is more vulnerable to noise.

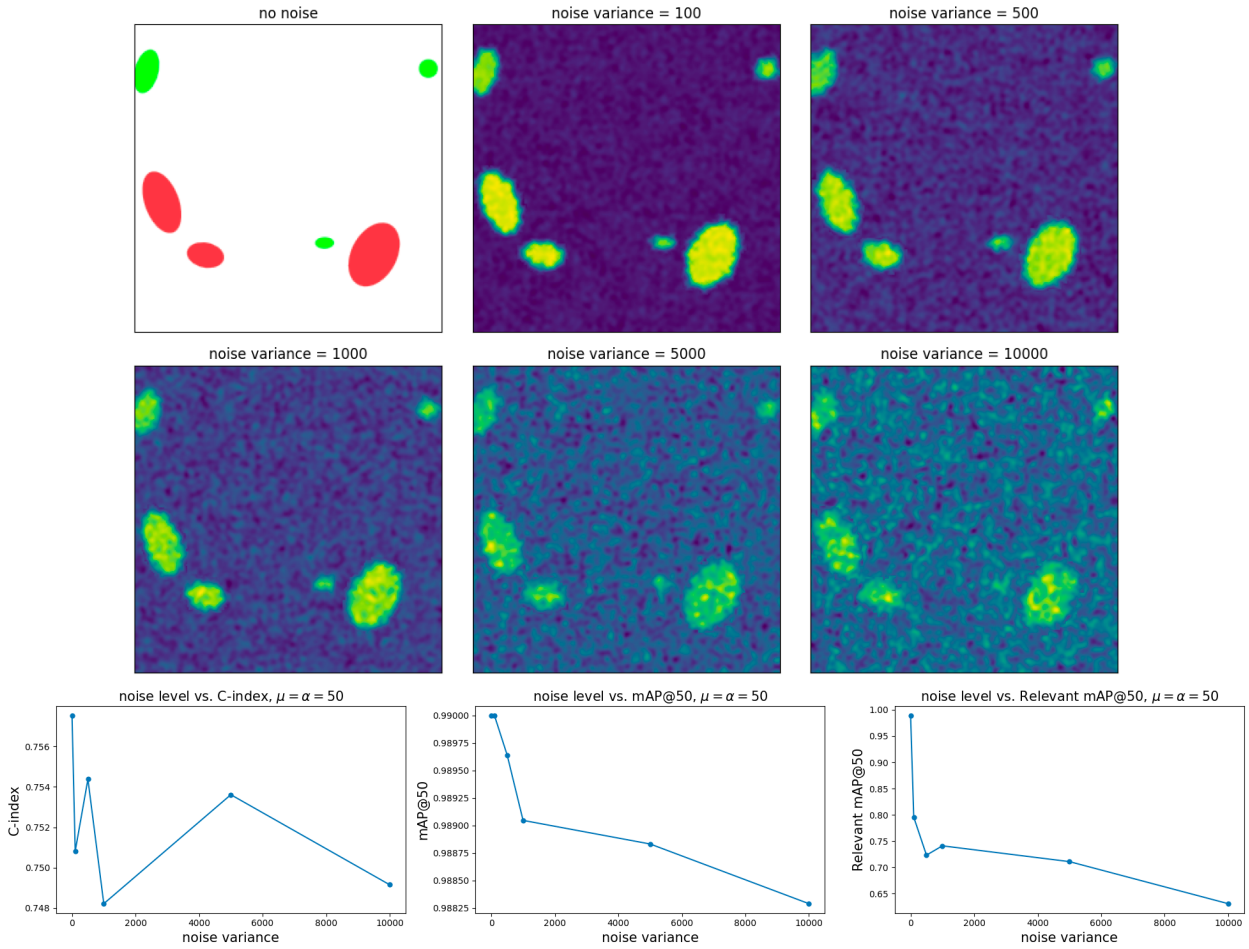


Figure 2.6: Top: example of different noise variances applied to the same input image. The mean intensity of red ellipses is 450 and that of green ellipses is 400. Bottom: Plots of performance on each task, varying the noise variance.

Discussion

Our goal was to design a neural network architecture that could predict an outcome while performing object detection, and on top of that detect specifically which objects are relevant to predicting the outcome. We addressed this with a modified region-proposal network with an additional outcome-relevant object classification head and an outcome contribution head (in our case, since the final predicted outcome was a risk score, this head predicted a risk score for each position of the image). Just by training on outcomes (survival times) and bounding boxes of all objects (red and green ellipse bounding boxes), without explicitly telling the model which objects were relevant, our model was able to identify the relevant objects (red ellipses) with very high accuracy on a clean dataset.

However, the performance of the model on relevant object detection was brittle to both the L1 regularization parameter μ and to noisiness in the images. In contrast, performance on survival prognosis and relevant object detection was quite stable. It appears that our indirect method of learning which objects are outcome-relevant is more brittle than losses that involve direct supervision for the desired task. Perhaps in future work, the formulation of our architecture and loss function could be iterated upon to produce a more robust learning of relevant objects, which would be crucial to deployment on actual medical images which contain lots of irrelevant to slightly relevant information that is effectively noise. On the other hand, our results also suggest that having doctors explicitly label which lesions are more relevant for the survival of the patient can perform better than trying to infer relevance from image and survival data alone.

2.3 Metastatic Lung Cancer Prognosis via Deep Image-Based Lesion Prioritization

Introduction

Lung cancer is the leading cause of cancer death worldwide due to its high incidence coupled with high mortality rate, and it is caused by smoking in about 80-90% [107] of cases.

Here, we design and evaluate a deep cancer prognosis model, which is a deep neural network that estimates cancer patient survival from computed tomography (CT) scans. It differs from prior deep cancer prognosis models in that it targets metastatic cancer, where a patient can have numerous lesions present in their CT scan, and it has an extra layer of interpretability. Accurate prognoses can better inform treatment decisions, accelerate treatment development, and an interpretable model could help understand the disease and instill confidence in the predictions. A metastatic cancer prognosis model is particularly useful for treatment development, since most cancer treatments being developed are focused on metastatic cancer. Towards these goals, we propose metastatic cancer prognosis by deep image-based lesion prioritization.

Lesion prioritization assigns a risk to each individual lesion, where risk signifies the impact of a lesion on patient survival. A simple, intuitive, and clinically utilized measure of the risk of a lesion is its size, hence the use of size-based Response Evaluation Criteria in Solid Tumors (RECIST) [84] and TNM staging clinical criteria. We hypothesize that a deep learning model could produce more informed risk estimates by accounting for factors other than lesion size.

Lesion-level risk predictions, as opposed to previous approaches which only predict patient-level risk, lend interpretability. A doctor could peruse the lesions identified as high-risk and, in case these contradict medical intuitions, know to trust the model less, or in case these conform to medical intuitions, have more confidence in the model. In more ambiguous cases, high risk lesions selected by the model may offer new insight as to what constitutes a dangerous lesion. For treatment development, changes in high risk lesions at followup may be more informative about treatment efficacy than RECIST, where the choice of which lesions to measure is subjective.

Our approach uses a convolutional neural network (CNN) to predict lesion risks and formulates patient risk as an aggregation of lesion risks. Thus, in the process of learning to correlate patient risk with survival, the model learns to correlate lesion risk with the impact of the lesion on survival. Our contributions are as follows:

- We propose and implement lesion prioritization, which predicts lesion-level risks and then aggregates them to predict patient survival, as a framework for deep metastatic lung cancer prognosis.
- We show that our model outperforms alternative deep learning approaches in a low data regime ($\sim$200 patients) on metastatic lung cancer.
- By using the predicted lesion risks for model interpretation, we find that the model predicts higher risks for lesions outside the lung, particularly in bone. Furthermore, we found that predicted lesion risk is predictive of whether a lesion will grow, suggesting sensitivity of the model to a proliferation phenotype.

Dataset

Our dataset consists of 258 patients with non-small cell lung cancer from a phase III clinical trial. 198 (77%) of cases are stage IV, and 246 (95%) are adenocarcinoma (Table 2.3). The median survival is 1 year and 6.5 months, with observed survival for 186 patients and censoring times for the other 72. The full survival distribution is shown in Figure 2.7.

The lesions of each patient are segmented by one to four radiologists (average: 2.92), each radiologist producing a distinct segmentation mask. The median of the per-patient lesion count, averaged over radiologists, is 10.7, with interquartile range [5, 19.4], reflecting a high number of lesions per patient due to the advanced levels of cancer present (Table 2.4. The median spacing of a CT in mm is $(0.78125, 0.78125, 5)$ and the median shape is $(512, 512, 131)$.

Stage	Count
IV	198 (77%)
III	23 (9%)
I-II	31 (12%)
Unk.	6 (2%)

Treatment	Count
A	92 (36%)
B	84 (33%)
C	82 (32%)

Table 2.3: The cancer stage and treatment composition of the dataset. The patients are from a phase III clinical trial testing the three treatments.

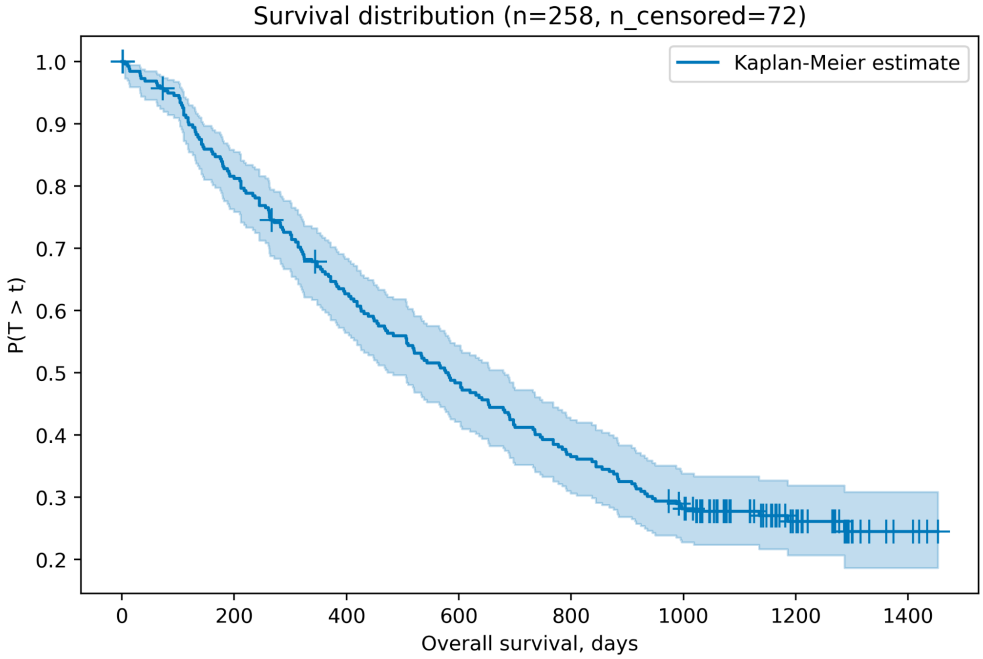


Figure 2.7: The survival distribution estimated using the Kaplan-Meier method from 258 patients with non-small cell lung cancer.

Percentile	Min	10th	25th	50th	75th	90th	Max
Number of lesions	1	2.8	5	10.7	19.4	36.8	232

Table 2.4: Number of lesions per patient. The number of lesions per patient is equal to the number of connected components in a radiologist's segmentation, averaged over radiologists.

Lesions vary widely in size, shape, and location. Axial slices from the chest of two example CT images are shown in Figure 2.8. A breakdown of lesions by organ is shown in Table 2.5. Lesions occur frequently in the lung, mediastinum, bone, liver, and unlabeled regions, with rare occurrences in the spleen, kidney, and stomach. An ideal lesion scoring model models risk for a wide range of lesion characteristics and surrounding tissues.

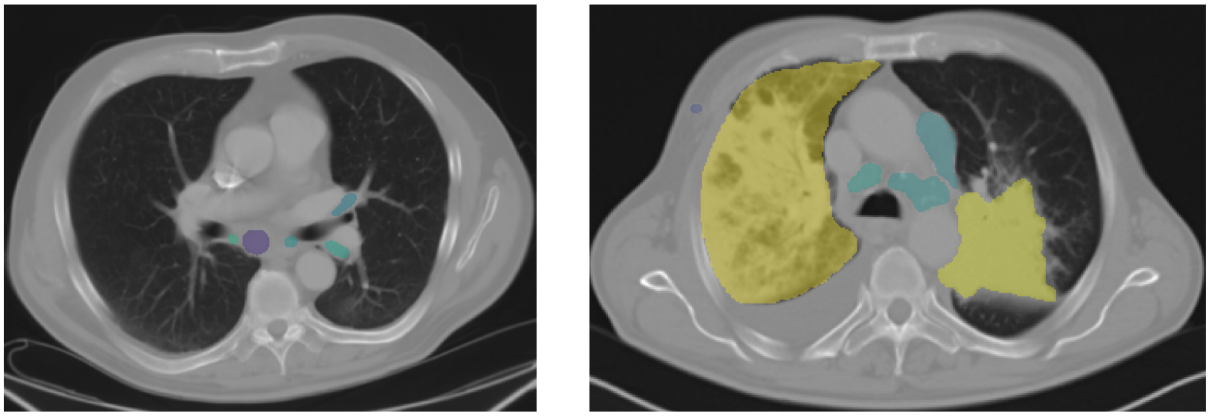


Figure 2.8: Chest axial slices with lesions highlighted. Note the small, round nature of the lesions of the patient on the left and the large, irregular nature of the lesions of the patient on the right. The patient on the left had a survival time of 283 days and the patient on the right had a survival time of 855 days.

	Other	Lung	Mediastinum	Liver	Spleen	Kidney	Stomach	Bone
% of lesions	23.6%	35.0%	18.6%	8.5%	0.2%	0.2%	0.3%	13.6%

Table 2.5: Distribution of lesions across organs. Lesions occur frequently in the lung, mediastinum, bone, liver, and unlabeled regions, with rare occurrences in the spleen, kidney, and stomach.

Method

Conceptually, our approach maps each lesion of a patient to a lesion-level risk score, and aggregates the lesion risk scores to form a patient-level risk score, trained using the negative proportional log likelihood (NPLL) loss [57, 66]. We extract a $2d \times 2d$ patch of each lesion of a patient where d is the lesion bounding box diameter and resize it to 72×72. If the patch would go out of image bounds, we shrink it isotropically until it is no longer so. Furthermore, each image is normalized to lie within $[0, 1]$ using the minimum and maximum

CT voxel values $(-1024, 3071)$. Then we feed the lesion patch to a ResNet18 lesion scoring network to obtain risk score r_{l_i} and the lesion volume to a fully connected size scaler network to obtain scaling factor s_{l_i}. The number of input channels of the ResNet18 is set to 1 and the number of output features is set to 1 because we are using grayscale images and outputting a single scalar value, the lesion risk. The size scaler network has two hidden layers, each with 128 neurons, and GeLU activation on each hidden neuron and output neuron. The patient risk score is the sum of the scaled lesion risks:

$$r_p = \sum_{i=1}^{nl_p} r_{l_i} s_{l_i}$$

where nl_p is the number of lesions in the annotation. The rationale is that a vision network (here, a ResNet18) can identify image-based features that distinguish between risky and less risky lesions to produce a risk r_{l_i}, and we inject size information, which is clearly relevant to survival, by scaling by the size-based scaling factors s_{l_i}.

In practice, we limit the number of lesions considered per patient annotation to 40. During training, we consider each annotation as a distinct example, but in evaluation, we average scores produced by different annotations for the same patient. A schematic of the approach is shown in Figure 2.9.

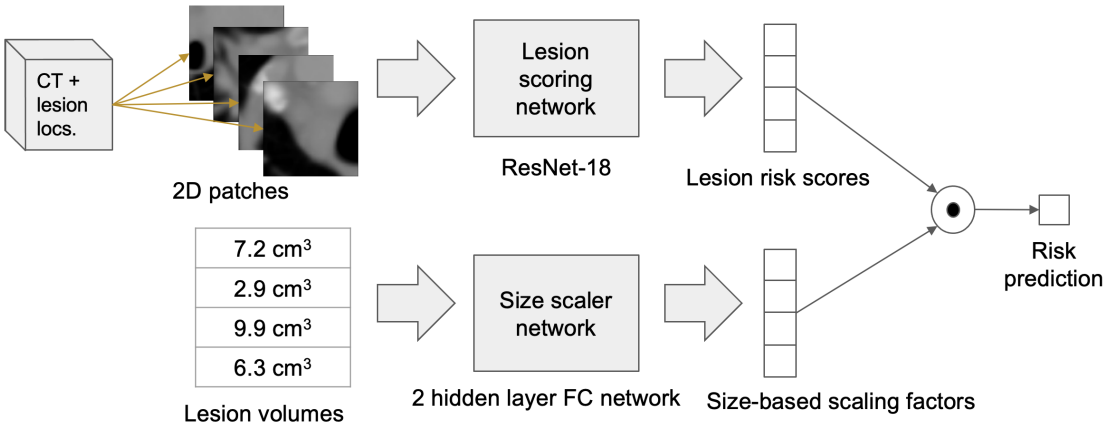


Figure 2.9: A schematic of the proposed approach. The patient risk scores are trained using the NPLL loss.

During training (and on the validation set), we perform standard data augmentations to each lesion patch, namely: random shifting, scaling, flipping, and rotation. Specifically, independently for both dimensions, shifting by a random fraction in $[-0.25, 0.25]$, scaling by a random factor in $[0.85, 1.15]$, flipping with probability 0.5, and finally, applying a random multiple-of-90 degree rotation. Also, during training, lesions are extracted based the annotation of a random radiologist to generate the prediction. During testing, no data

augmentation is done, and we average the predictions resulting from all the annotations to produce the final test prediction.

For accurate evaluation, we run 4-fold cross validation, partitioning the dataset into 4 equally sized parts with matched joint survival $\times$ treatment distributions, using two parts for training and one for validation and test each, cycling the parts 4 times to obtain 4 splits. With 258 patients, we felt that a sample size of one-fourth (≈ 64) of the patients was appropriate to obtain meaningful C-indices on the validation set and test set, which limited the number of folds we considered to 4.

Results

Overall, we found our model components to be performant, and the approach of aggregating lesion risks across all or near-all lesions to outperform other deep CT-based prognosis approaches on metastatic cancer.

Comparison to pretraining

We assessed the efficacy of a ResNet18 trained from scratch for the lesion scoring network by comparing with pretrained models finetuned on our task – a ResNet50 pretrained on ImageNet, and a ResNet50 pretrained on radiological image tasks [76]. Overall, while pretraining obtained higher training performance, the test and validation performance were not significantly higher. Interestingly, the ImageNet-pretrained network overfitted severely, while the radiological image-pretrained network only overfitted slightly, though the validation and test performances were similar. Pretraining on ImageNet, which is a very large collection of diverse natural images, likely leads to learning a larger repertoire of image features, allowing the model to grasp patterns correlated with the outcome more quickly. However, most of these features likely have little to do with the relationship between radiological image and outcome, leading to poor generalization. See Figure 2.10 and the first three rows of Table 2.6.

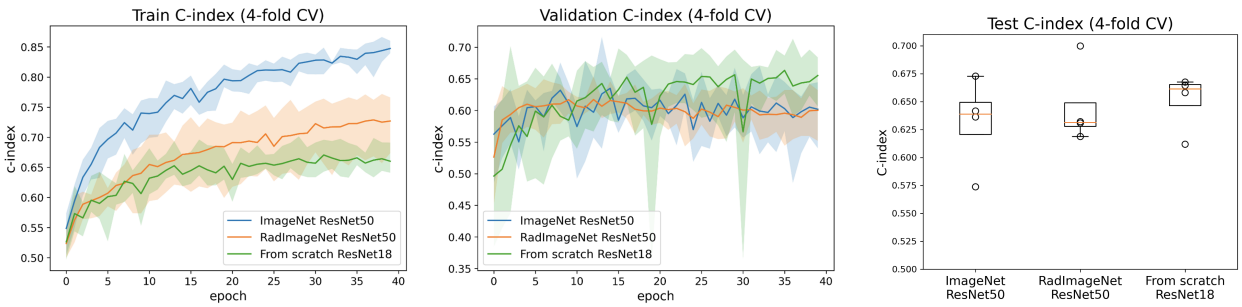


Figure 2.10: Training, validation, and test C-indices of lesion networks trained from a pretrained initialization and ResNet18 trained from scratch.

Size scaler ablations

Next, we assessed the effect of the size scaler on performance. In the "no FC network" ablation, we replaced the fully connected size scaler network with multiplication by a learned constant, changing the scaling factor from a nonlinear to a linear function of volume. In another ablation, we removed the size scaler entirely and simply summed the lesion risks to generate patient risk. The ablation results are shown graphically in Figure 2.11 and summarized as metrics in rows 4 and 5 of Table 2.6.

A linear function of volume (0.658 test C-index) performed similarly to a nonlinear one (0.662), but removing size-based scaling altogether performed slightly worse (0.633). This emphasizes the benefit of including lesion size information for prognosis, though there may be room to optimize the way it is incorporated. On the other hand, performance does not drop too much, suggesting our method may still be applied when size information is unavailable. However, the large epoch-to-epoch fluctuations of the validation curves, particularly the one for the "no size scaler" model, present a challenge for comparing method performance.

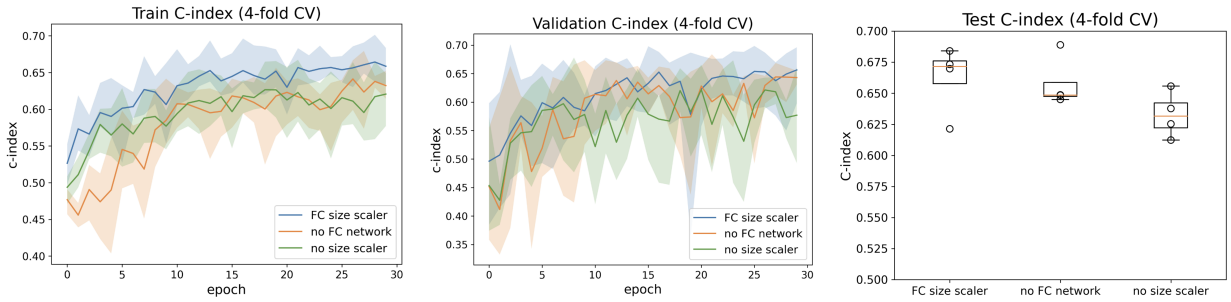


Figure 2.11: Training, validation, and test C-indices of different size scaler settings.

Comparison to alternative models

Though a main benefit of lesion-level risks is interpretability, we compared the survival prediction performance of our approach with that of other viable deep learning approaches to ensure a reasonable level of performance.

Our approach outperformed other deep learning approaches on our data. The alternatives were, predicting survival from a size-weighted average embedding of the 5 largest lesions, similarly to [70], and using whole CT volumes as input. When using whole CT volumes, we resampled all volumes to a spacing of $(0.78125, 0.78125, 5)$, used a $(480, 480, 120)$-shaped crop, and performed analogous data augmentations to the ones done on 2D lesion patches.

Our method outperformed both approaches (Table 2.6). Outperforming lesion averaging suggests that in metastatic cancer, accumulating risk over all lesions is more accurate than basing risk on a summary derived from large lesions. Finally, the whole volume approach likely performed the worst due to data scarcity. Our dataset has 258 patients, while a

successful whole volume-based approach in lung cancer detection used over 10,000 patients [77]. Learning generalizable patterns from whole volumes necessitates a large amount of training data (unique patients), due to the extremely high amount of information contained within an entire CT scan ($480 \times 480 \approx 27.6$ million features per image, granted a large fraction are blank voxels), the bulk of which is likely irrelevant to survival prediction.

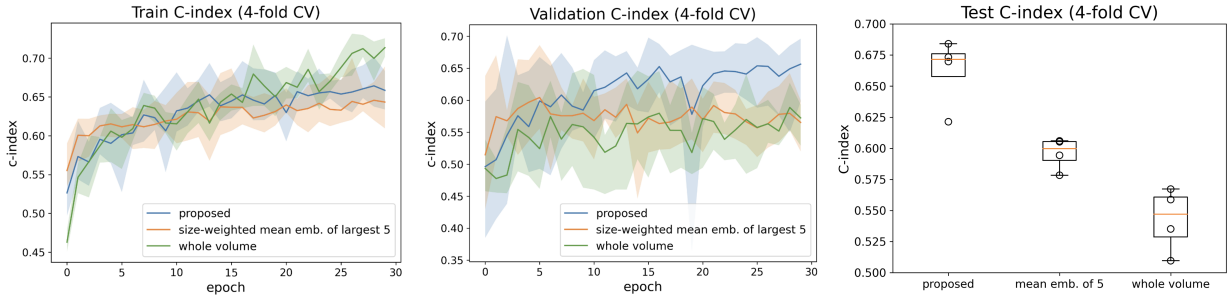


Figure 2.12: Training, validation, and test C-indices of different deep CT-based survival prediction approaches.

	Train	Valid	Test
ImageNet ResNet50*	0.782±0.06	0.662±0.03	0.631±0.04
RadImgNet ResNet50*	0.660±0.03	0.640±0.01	0.645±0.03
Proposed	0.658±0.02	0.656±0.04	0.662±0.02
No FC network	0.632±0.02	0.643±0.01	0.658±0.02
No size scaler	0.621±0.03	0.577±0.06	0.633±0.02
Mean of largest 5	0.643±0.03	0.565±0.03	0.567±0.03
Whole volume	0.713±0.01	0.572±0.03	0.535±0.02

Table 2.6: Model C-indices from 4-fold cross validation. For the starred models, the best validation checkpoint was chosen for evaluation, while for the other models, the last model checkpoint (after 30 epochs) was chosen.

Consistency of risks across runs

Here we assess how consistent our model is by computing the correlation between lesion-level risks from models trained on different folds. We also do the same for patient-level risks. Because of our 4-fold cross validation procedure, each patient and their lesions constitutes a training example for two models, a validation example for one model, and a test sample for one model. Thus, we are correlating training predictions and validation and test predictions between different models, which still provides a measure of the generalizability of our method. The lesion-level correlations are shown in Figure 2.13 and the patient-level correlations are in Figure 2.14.

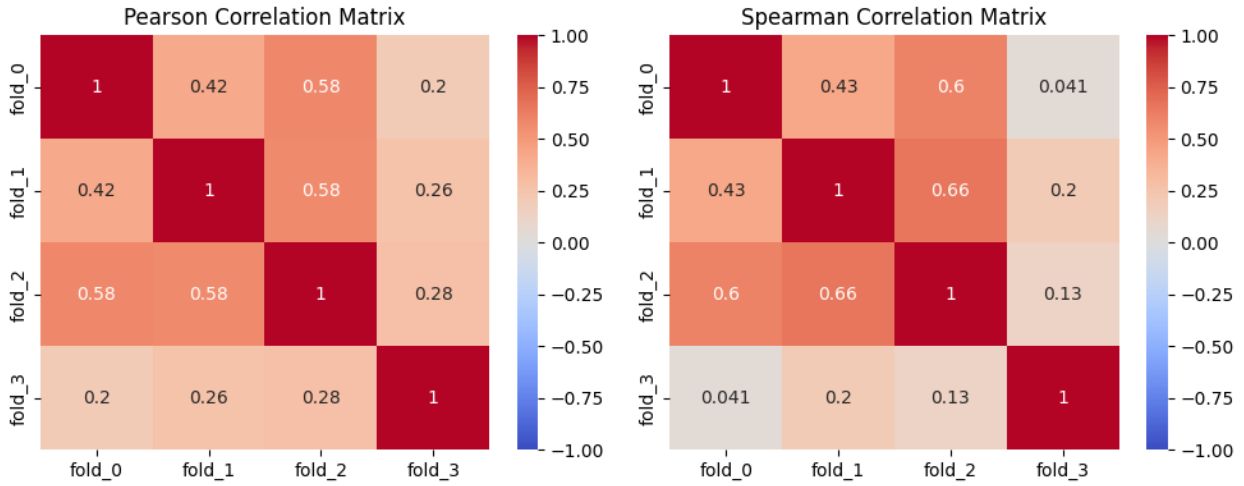


Figure 2.13: Correlations between lesion-level risks predicted by 4 models trained via 4-fold cross validation.

The lesion-level correlations are moderately positive, indicating our model learns generalizable image-based notions of risk. The pearson correlations exceed the spearman correlations, suggesting that while there might be substantial uncertainty in the risk of a large proportion of lesions, very high risk lesions are reliably predicted as high risk. As might be expected, the patient-level correlations are higher than the lesion-level correlations, due to the denoising effect of aggregating lesion risks to form an overall assessment of risk for the patient.

Interpretation using lesion risks

By assigning risk at the lesion level, one can better understand the medical reasoning of the model and what kinds of lesions predict a better or worse prognosis. The following interpretability analyses all use lesion risks predicted by a trained model from an arbitrarily chosen split on its test set.

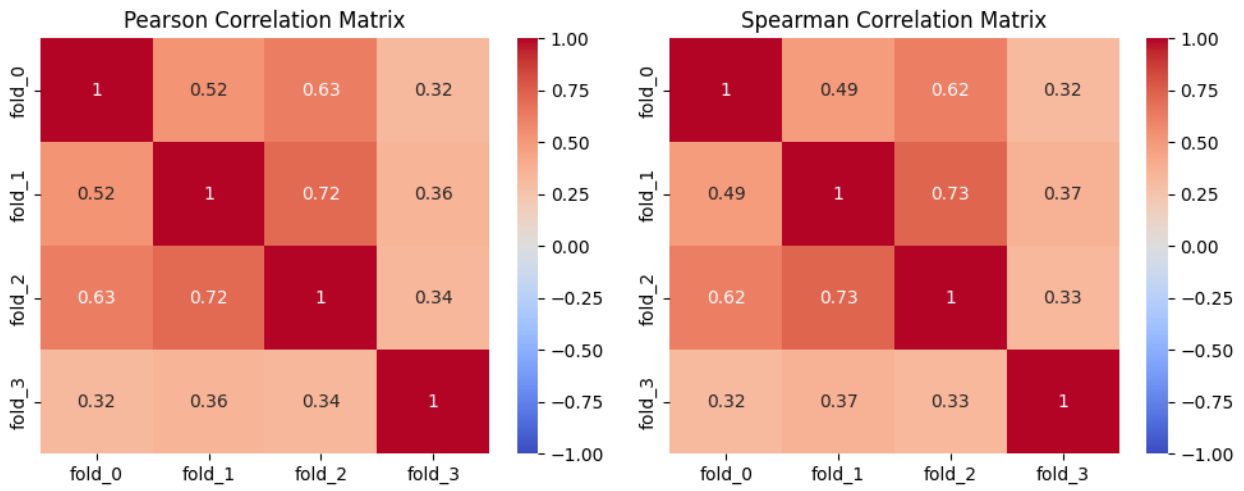


Figure 2.14: Correlations between patient-level risks predicted by 4 models trained via 4-fold cross validation.

Lesion risks outside the lung are higher

The distributions of lesion risks in each organ were obtained using radiologist-annotated organ masks and plotted in Figure 2.15. Interestingly, the risk distribution in the lung is lower than in other organs, recapitulating the increased risk associated with metastasis. Furthermore, the model highlights bone lesions as particularly high risk, agreeing with the observed association of bone metastases with morbidity [26].

Lesion risk is correlated with lesion growth

Intuitively, risky lesions are more likely to grow. To see whether this notion is borne out in the risk predictions of our model, we tested the hypothesis that the mean risk of growing lesions exceeds the mean risk of shrinking lesions. We labeled a lesion as growing or shrinking if it came from an annotation with a second timepoint available (89.6%) based on whether its volume increased in the second timepoint. Note interreader variability is high – the probability a pair of radiologists agrees on whether the same lesion grows or shrinks is 0.827, while the probability a lesion shrinks is 0.673.

Because lesion risks from the same patient are likely correlated, instead of using a t-test, we used a mixed linear model including patient identity coefficients to test our hypothesis. Let r_{ij} denote the risk of the jth lesion of the ith patient, y_{ij} the binary variable equal to 1 if l_{ij} grew and 0 otherwise, c_i the risk coefficient for ith patient, and w the risk coefficient of lesion growth. Then we fit the mixed linear model

$$r_{ij} \sim c_i + w y_{ij} + \epsilon_{ij}$$

where ϵ_{ij} is Gaussian noise. We did not use a t-test because lesion risks from the same patient are likely to be correlated.

The p-value on the test set was 0.006, indicating significantly higher risk among growing lesions. The risk distributions are shown in Figure 2.16. Thus, we extracted from the model that growing lesions negatively predict survival, and are partially identifiable from CT image features.

Visualization of high and low risk lesions

We visualized the top 5 highest predicted risk lesions and the top 5 lowest predicted risk lesions on the test set in Figure 2.17. We observe that the highest risk lesions are bone lesions, agreeing with our earlier observation that predicted risks in bone are the highest. The lowest risk lesions are well contained within the lung as opposed to near the edges, suggesting that the model picked up on high-margin containment in the original organ as a predictor of low risk cancer. This agrees with lesions exhibiting a higher degree of infiltration into neighboring tissues being staged more severely.

Risk saliency maps

Class activation maps (CAM) [135] are a visualization method applicable to convolutional neural networks whose final section consists of global average pooling followed by a linear layer. They highlight regions of the input image based on their contribution to the prediction. CAMs are visualized for various image patches containing lesions in Figure 2.18. In some cases where the model predicts a low risk, the entire input is highlighted as low risk, as in the image in the first row. This could correspond to the neighboring tissue playing a more significant role in the model's prediction than the lesion itself. In higher risk cases, sometimes the regions highlighted as high risk are slightly offset relative to the actual lesion in the image, like in rows 3 and 4, whereas other times they are focused on the actual lesion, like in rows 2 and 5. Further research is necessary to determine whether this is due to the detection of biologically significant features at the tumor boundary, or due to difficulty identifying the lesion in the image.

Discussion

By formulating patient risk as the aggregation of lesion risks, we were able to design an intepretable and data-efficient deep learning model for metastatic lung cancer prognosis. This approach can be directly extended to other forms of metastatic cancer and other forms of disease which present as multiple lesions in the same patient. We found a more granular formulation of the prognosis task, like NPLL or nnet-survival, to have better performance than binary classification, which discards information in the labels. Summing lesion risks (scaled based on size) is intuitive and injects an inductive bias as to how patient risk is formed that alleviates with data scarcity relative to the high dimensionality of the CT images.

Also, we were able to confirm that the model is truly learning survival-related patterns by interpreting the lesion risks. We found that it recapitulates the danger of metastasis and predicts growing lesions as riskier, even after accounting for within-patient correlation. The saliency maps on top of the lesion patches are difficult for a layperson to interpret. From the maps, it appears as though the model detects lesions, though this is hard to confirm, because the lesions are in the center of each patch, making it easier for them to be highlighted on accident. Having a radiologist interpret the saliency maps would help shed light on what biological features the model focuses on.

A challenge is that the limited number of patients makes training and evaluating models difficult, as hold-out performance is prone to sudden, high fluctuations. We did not find stochastic weight averaging [51] or an exponential moving average of model weights from previous epochs to be helpful, as this usually decreased performance. Advancements in stabilizing models would be very helpful in high complexity, low data regimes like ours.

2.4 Related work

Multiple instance learning

Our approach can be cast in the framework of multiple instance learning, with lesions as the instances. To our knowledge, this is first time a multiple instance learning approach has been applied to metastatic cancer prognosis, viewing lesions as instances. However, applying multiple instance learning to histopathological images to diagnose cancer subtype [48] or perform cancer prognosis [131] is indeed an established, related, approach. In this case though, the images are of cancer tissue at microscopic resolution where the cells can be seen, not a CT scan of the body. The instances here are patches of the extremely large 2D histopathological images.

Cancer diagnosis and risk models

A related task is to diagnose cancer from medical images. A model that classifies skin cancer with dermatologist-level performance [29] has been obtained, as well as a highly accurate (AUC > 0.9) breast cancer detection model from mammography images [104].

A perhaps more related task to ours is cancer risk prediction – predicting the chance of later developing cancer based on a current medical scan. Predicting the time at which one will develop cancer is akin to prognosis, with the event "death" replaced with "cancer development." This was done successfully for lung cancer risk using whole 3D CT volumes (Sybil, [78]), and breast cancer risk using mammography images (Mirai, [128]), both achieving C-indices of around 0.75. Both this problem and ours are survival analysis problems that have to do with cancer, but this one is pre-cancer while ours is post-cancer, and our solution explicitly utilizes existing lesions. Furthermore, the aforementioned risk models had training data available for on the order of tens of thousands of patients, allowing them to effectively

learn from entire images without overfitting. Data from metastatic cancer patients is scarcer, but these studies highlight the potential of larger scale approaches for more accurate and powerful cancer prognosis models.

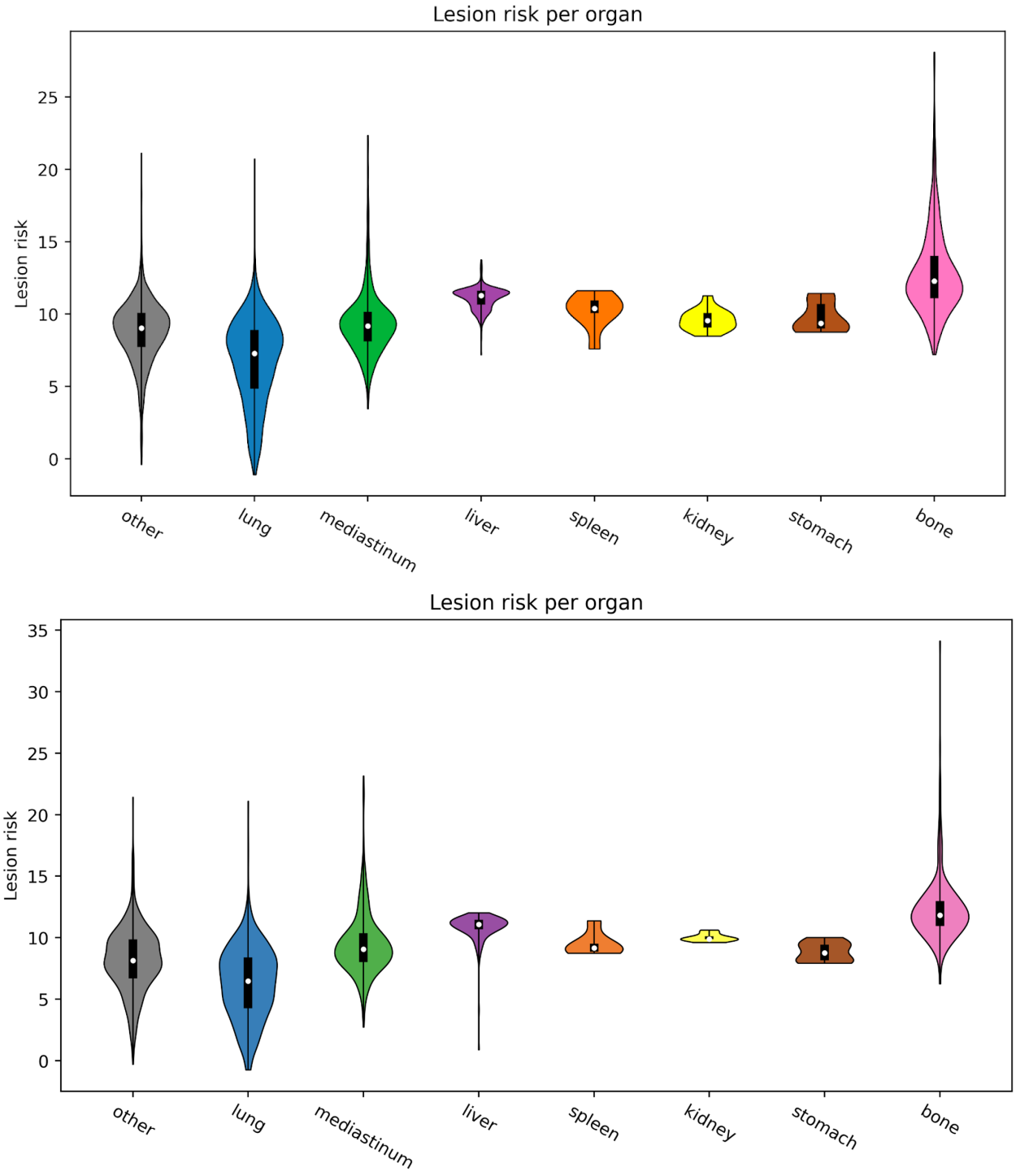


Figure 2.15: Violin plots showing the distribution of predicted lesion risks in each organ. Top: train, bottom: test. The distributions for the train lesions and test lesions are similar.

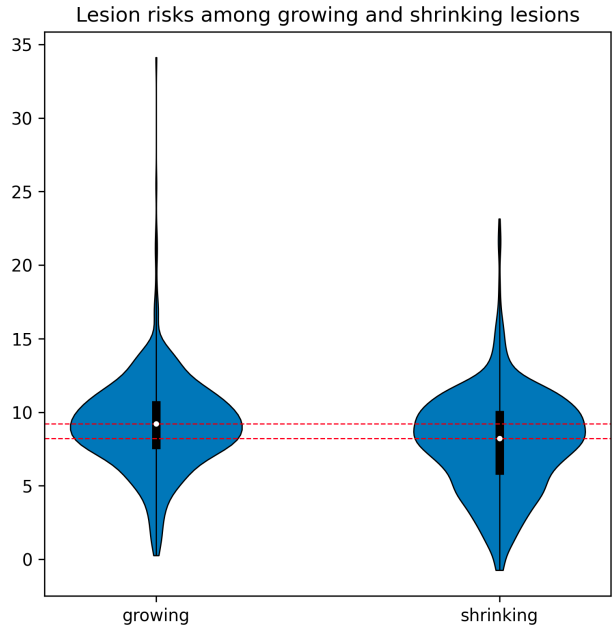


Figure 2.16: Risk distributions for growing and shrinking lesions. The white dots represent medians.

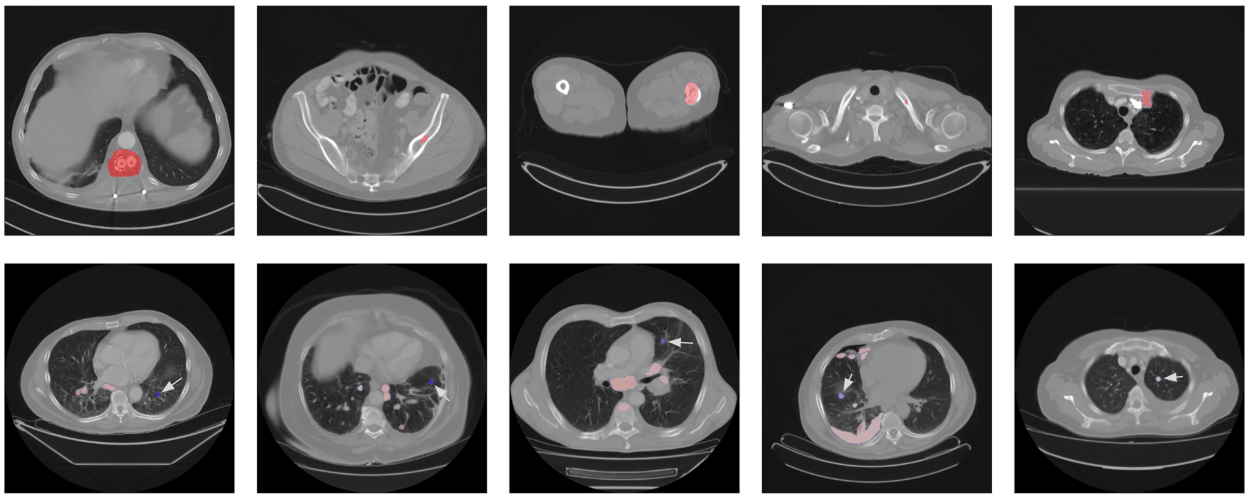


Figure 2.17: The top row shows the 5 highest risk lesions on the test set. The bottom row shows the 5 lowest risk lesions on the test set. Lesions, including other ones in the same image, are colored according to their risks, with blue denoting low risk and red denoting high risk. Due to the small size of the low risk lesions, arrows pointing to them are drawn.

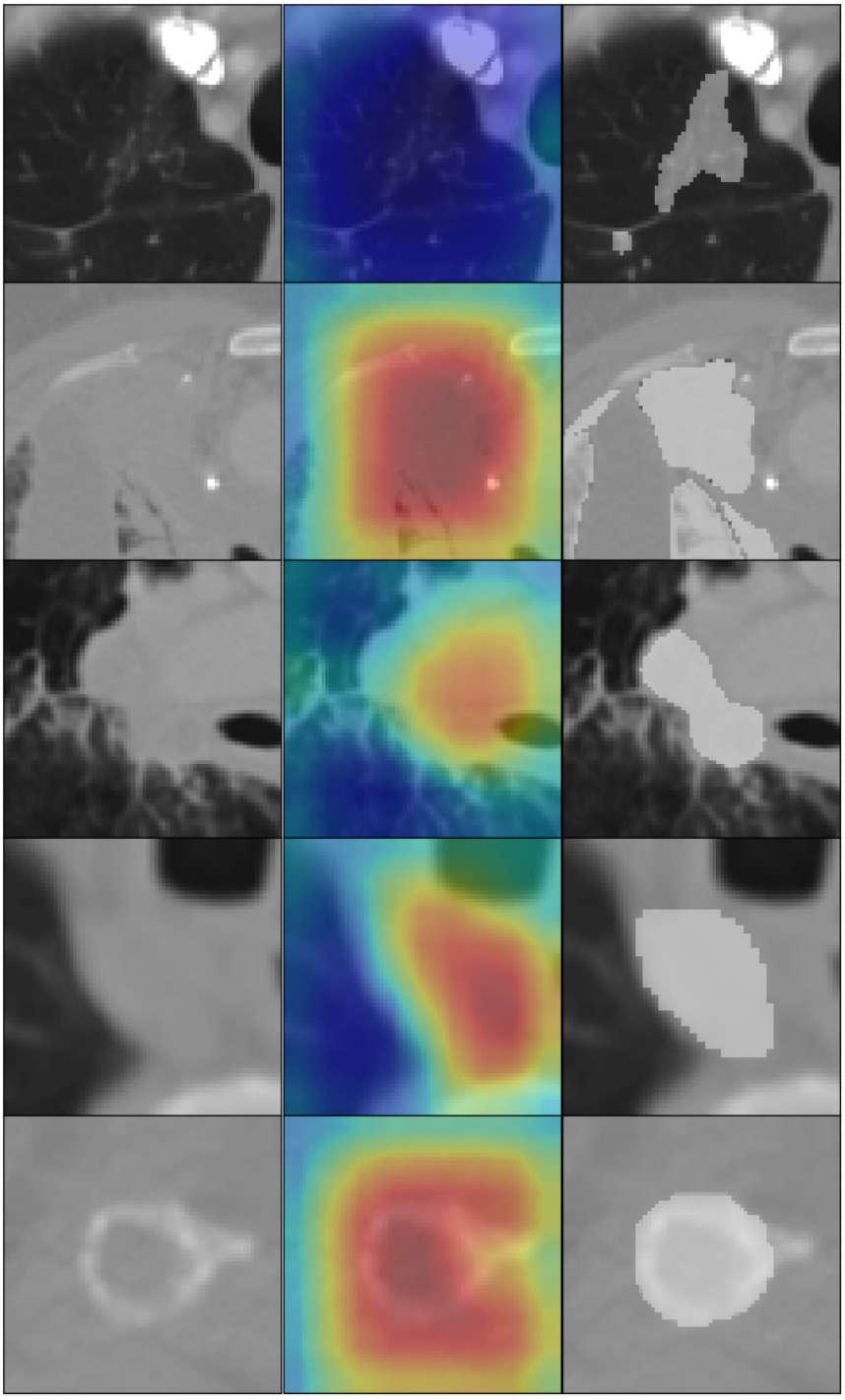


Figure 2.18: Class activation maps (CAM) [135] with respect to risk predictions for an array of lesions. First column, input image; second column, CAM overlaid on image; third column, lesion segmentation overlaid on image. For CAMs, red denotes high risk; blue denotes low risk.

Chapter 3

Interpreting DNA shape in a deep TF binding model

This work was done with Ryan Keivanfar, a PhD candidate in computational biology at UC Berkeley and UCSF.

Overview Here we input DNA shape in addition to DNA sequence to a deep neural network to predict transcription factor (TF) binding. Then, we utilize interpretation methods like DeepLIFT [105] and TF-MoDISco [106] to understand how DNA shape features are used to predict TF binding.

3.1 Background

TF binding and gene regulation

TFs are proteins that bind to DNA to regulate gene expression at the transcriptional level. Cells perform functions via proteins, and each protein is created by transcription of a gene into mRNA, followed by translation of the mRNA into protein. Thus, TFs control the function of a cell. TF binding can activate DNA that is in an inactive or closed state [133], induce specific DNA conformations that affect gene expression [114], and recruit other proteins, including the critical pre-initiation complex that holds down RNA polymerase as it transcribes the gene [112] (Figure 3.1). In addition to activating functions, TFs can also serve repressive functions [97]. By better understanding TF binding mechanisms, we can better understand regulatory biology and obtain a clearer functional map of the human genome.

TF binding is often experimentally measured through an assay called ChIP-seq, which stands for chromatin immunoprecipitation with sequencing. In this assay, first, formaldehyde is applied to cross-link the TF of interest to DNA at locations where it has bound. Then, the DNA is sonicated, shearing it apart at locations where it is not cross-linked. This leaves behind DNA fragments that were bound to the TF or other proteins. In order to target

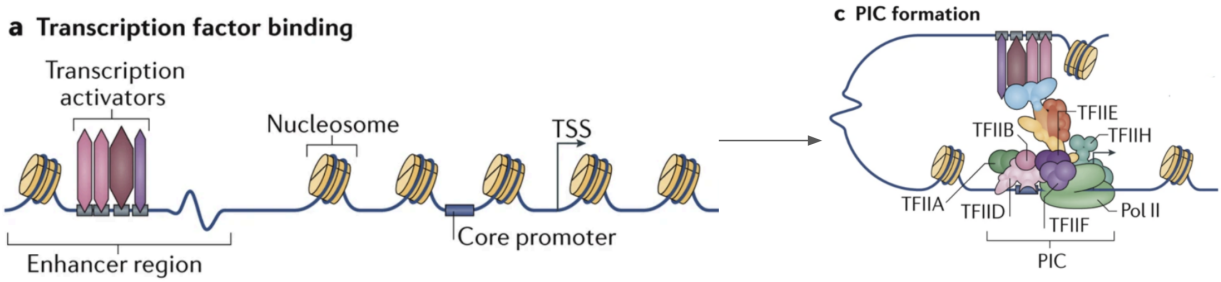


Figure 3.1: Activator TFs recruit the transcription pre-initiation complex, causing the gene in the figure to be transcribed. Taken from [112]. An enhancer is a regulatory region far from the transcription start site (TSS), while a promoter is the regulatory region immediately preceding the TSS.

fragments that specifically bound the TF of interest, an antibody for the TF is used with incubation and centrifugation. Finally, the relevant DNA fragments are un-linked from the protein, sequenced, and mapped to the genome to determine the binding sites of the TF. A typical post-processing step is peak-calling, where the resulting read counts (the number of times each base of the genome appeared in some fragment) are converted into peaks, or intervals in the genome that are significantly enriched with reads, and thus likely represent true binding sites for the TF of interest.

In this work, we mainly derive TF binding labels from ChIP-seq experiment peaks. Other TF binding assays exist, but there is a wealth of ChIP-seq data available from genomic data consortia such as the Encyclopedia of DNA Elements (ENCODE) [24] and Roadmap Epigenomics [11]. Previous seminal works in genomic deep learning [136, 8] used these data as well. Other assays include ChIP-exo [95] and ChIP-nexus [44], which increase the resolution of ChIP-seq reads by trimming the ends of cross-linked fragments with exonuclease digestion. CUT&Run [109] and CUT&TAG [58] use an enzyme to cut DNA instead of sonication. These assays find binding sites in vivo, that is, in DNA from cells from a living organism. There are also in vitro assays, such as systematic evolution of ligands by exponential enrichment (SELEX) and protein binding microarrays (PBMs) [10], which find binding sites in short, artificial strands of DNA called oligonucleotides. Here, the goal is to more precisely isolate the specificity of a TF by looking at "pure" sequences without interference from complex genome interactions.

Sequence and shape readout

Traditionally, TF binding is understood through TFs' preferred sequence motifs, which are short patterns made up of the four DNA nucleotides, A (adenine), C (cytosine), G (guanine), and T (thymine). For example, the Max transcription factor, which is a member of the basic

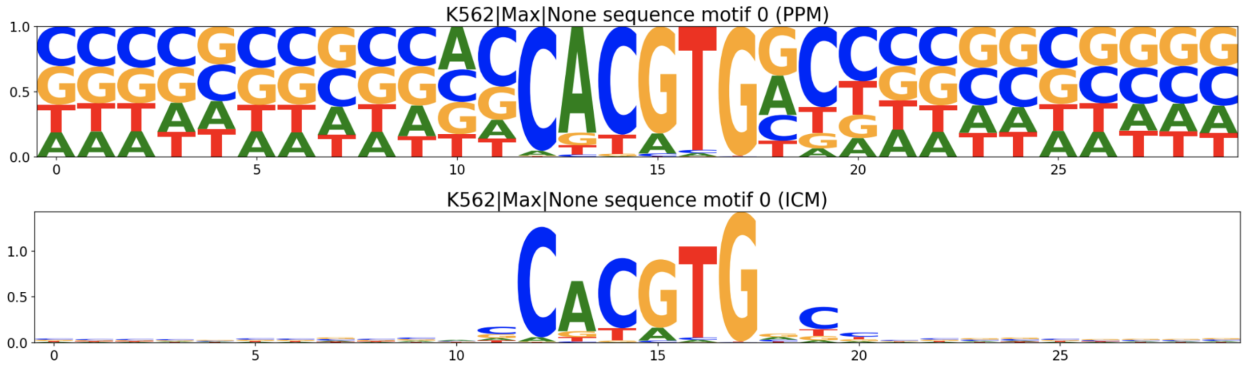


Figure 3.2: Example of the position probability matrix and information content matrix representation of a sequence motif.

helix-loop-helix (bHLH) family, has the sequence motif CACGTG. USF1, also a bHLH TF, has the same sequence motif, highlighting the redundancy that exists between certain pairs of TFs' motifs. A more precise representation of a motif is its position probability matrix, which contains the probability of observing each nucleotide at each position. For visualizing the importance of each base, this matrix is often converted to the information content matrix, which weights each position by its specificity (2 minus base 2 entropy). An example is shown in Figure 3.2. Note that DNA is double stranded, and we are actually representing both strands with the sequence of one. We could just as well represent the same stretch of DNA with the sequence of the other strand, but reversed, to maintain the biological 5' to 3' direction. Thus, a motif is functionally equivalent to its reverse complement, e.g. TCAGCA for TGCTGA.

To make things difficult, however, TFs with very similar sequence motifs can have very different binding locations along the genome. For example, two yeast bHLH transcription factors, Tye7 and Cbf1, were found to have very similar sequence motifs but very different bound regions [39]. It was shown that differences in their binding sites were explained by DNA shape values flanking the motif or equivalently by considering dinucleotides (2-mers, combinations of two nucleotides) in the flanking regions. Furthermore, it was also recently found that in addition to their motifs, TFs weakly bind to short tandem repeats, which are repetitive stretches of DNA such as CACACA... [46]. Alternative forms of TF binding have gained attention recently and warrant further study [101], as we still lack a comprehensive understanding of TF binding targets and mechanisms.

An established dichotomy in TF binding is that of sequence readout versus shape readout. Sequence readout refers to when TFs recognize, or physically contact, the bases of DNA, through chemical interactions between the amino acid residues of the TF and the bases themselves. The double-helix structure of DNA has two grooves, or spaces between the two strands that wind around together with the strands — the minor groove and the major

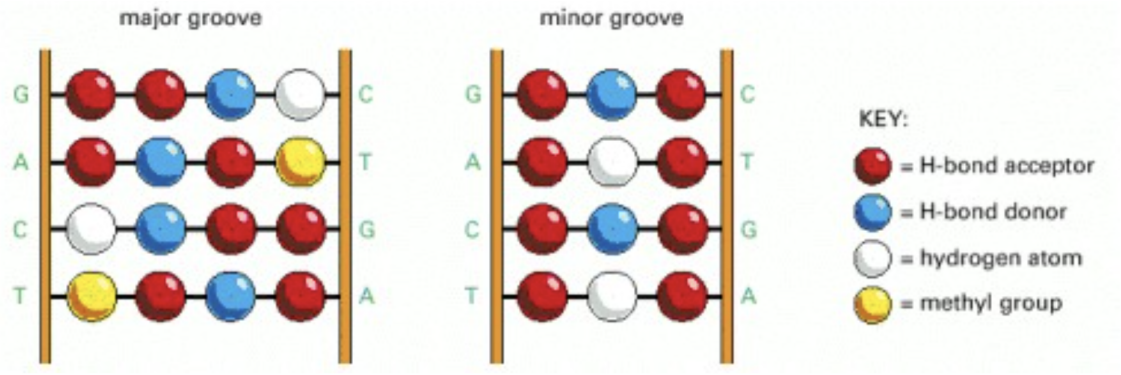


Figure 3.3: The chemical pattern corresponding to each basepair, which is a sequence of hydrogen bond acceptors, donors, inert hydrogens, and methyl groups, as viewed in the major and the minor groove. The uniqueness of the patterns in the major grooves allows the bases to be recognized in the major groove from their chemical patterns. The same is not true of the patterns in the minor groove, as G-C and C-G have the same minor groove pattern. Taken from Chapter 7 of *Molecular Biology of the Cell, 4th editon* [5].

groove (imagine ascending DNA as if it were a twisted ladder — the side of the ladder you are on is the groove you are in). One side is the minor groove, and the other side is the major groove). Each of the four valid basepairs, A-T, C-G, G-C, and T-A, exposes a different chemical pattern in the major groove, allowing a TF to recognize a particular sequence of bases by contacting the major groove [5]. However, the minor groove chemical patterns are redundant, leading researchers to conclude that sequence readout primarily happens in the major groove [96]. Figure 3.3 displays the chemical patterns of the different basepairs in the minor and major groove.

Shape readout refers to when TFs recognize geometric parameters of the double helix called shape features, as opposed to the bases themselves. Shape features include helical twist, minor groove width, propeller twist, and roll. These shape features are depicted in Figure 3.4. Electrostatic potential is also included as a shape feature, since it has been found relevant to DNA shape [12]. The repertoire of shape features has been expanded to include ones like buckle and shear [52] but for simplicity, here, we focus on the earlier five. These basepair resolution features are obtained from the DNAshapeR package [121]. DNAshapeR is essentially a 5-mer-to-shape lookup table, where each 5-mer (ordered combination of 5 nucleotides, e.g., GGTCA), is mapped to its corresponding shape feature values. These values were obtained by averaging over occurrences of the 5-mer in Monte Carlo simulations [115], and were shown to agree well with experimentally measured DNA shape. The computational efficiency of this method is a plus; rather than running expensive simulations, predicting DNA shape amounts to scanning a length 5 window across the sequence. Similarly to sequence motifs, one can consider shape motifs. ShapeMF [102] is a method that finds

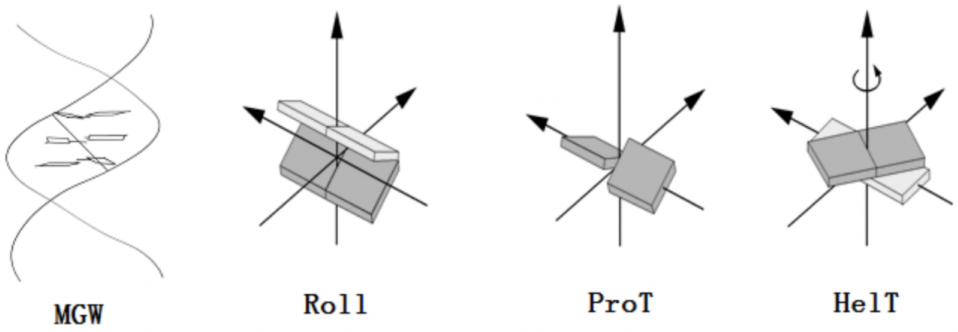


Figure 3.4: Diagram illustrating what geometric aspect of DNA each shape feature measures. Taken from the supplement of [115]. Helical twist is inter-basepair (inter-bp) rotation around the helical axis. Propeller twist is intra-basepair (intra-bp) rotation. Roll is inter-bp rotation around the basepair axis.

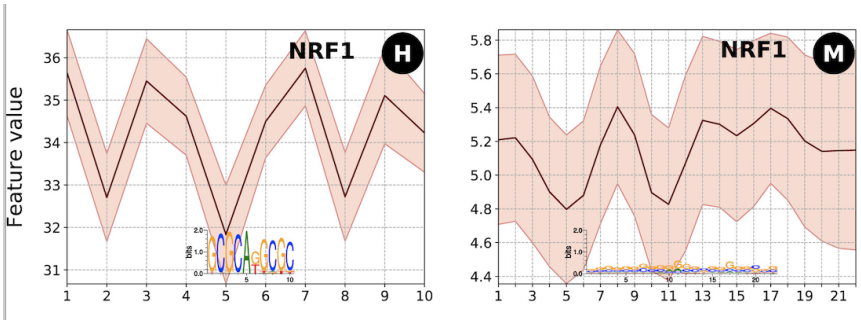


Figure 3.5: Example of a helical twist and a minor groove width shape motif for the NRF1 transcription factor. Taken from from [102], which proposed the ShapeMF algorithm.

shape motifs from a set of sequences. Shape motifs are visualized by showing a measure of central tendency (e.g. mean) at each position along with the spread (e.g. standard deviation) as a shaded width at each position. Example shape motifs are shown in Figure 3.5.

Finding cases of shape readout has been approached in various ways. One approach has been comparing the performance of sequence-only models to that of sequence + shape models of transcription factor binding and showing that the addition of shape features produces a gain in performance, suggesting evidence of shape readout, but this has heretofore mainly been done for shallow models [52, 39, 137, 75]. Furthermore, these works find that including 2-mer or 3-mer features more or less recovers the performance of adding shape features. This shows that the information contained in shape features is simliar to that which is contained in k-mer ($k \geq 2$) features. One could argue that if a specific combination of nucleotides is required to help explain binding, and its individual component nucleotides are

insufficient, then it is more likely that the combination induces binding through a favorable DNA conformation, rather than by reading each individual base, because in that case we would expect the individual component nucleotides to be predictive. Another approach is to show that the DNA shape of the preferred sequence is similar to the DNA shape of the bound sequence [3, 115], based on the logic that the sequence might be preferred because it is already physically close to its configuration when bound by the TF. In a work specifically aiming to deconvolve the contribution of sequence and shape [2], amino acids of the Exd-Scr protein complex that have a narrow minor groove preference at a particular position in the complex's binding site were mutated, and this was shown to result in preference for a wider minor groove. Combined with the fact that the amino acid contacts to the narrow minor groove were found to not form hydrogen bonds in earlier work [53], this was considered a strong case of shape readout. Nonetheless, many of the cases presented for shape readout have been rather implicit in nature, and I believe that clarifying the definition of shape readout with more explicit examples of what it actually is would help guide explorations in this field.

Genomic deep learning interpretability

Here, we assume we have a neural network that has been trained on genomic sequences to predict some property, such as transcription factor binding, gene expression, or accessibility (whether the inputted DNA is *accessible*, i.e. active in a particular cell type). We wish to *interpret* the model, or understand the basis for its decisions. Since much is unknown about the mapping from genomic sequence to function, this could be used to discover new genomic mechanisms. Identification of previously characterized phenomena would also be useful as a form of validation of both model and yet-to-be-confirmed theories. Since neural networks have been demonstrated to learn highly complex patterns, there is great potential for extracting insights by interpreting a neural network trained on some genomic prediction task.

DeepLIFT

DeepLIFT [105] is an *attribution method*, which means that it assigns a score to each input feature representing its contribution to the prediction. More precisely given an input vector $x \in \mathbb{R}^d$ whose prediction is $\hat{y} = f(x)$ and a "reference" input $x_0 \in \mathbb{R}^d$ whose prediction is $\hat{y}_0 = f(x_0)$, DeepLIFT computes an attribution vector $a \in \mathbb{R}^d$ such that the ith attribution is the contribution of the ith input feature to the prediction. This is reinforced by having the attributions sum to the difference from reference:

$$\sum_{i=1}^{d} a_i = \hat{y} - \hat{y}_0,$$

so that we can interpret a_i as how much x_i "personally" contributes to the difference of the prediction from the reference prediction.

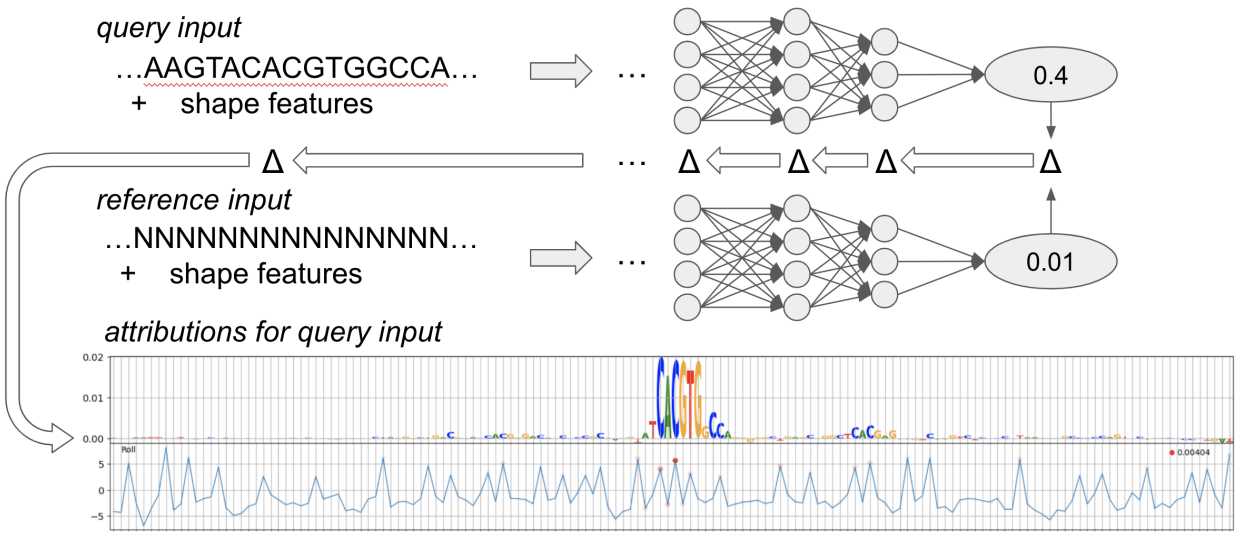


Figure 3.6: Schematic of DeepLIFT.

DeepLIFT operates through a backpropagation-like procedure where attributions starting from the output layer get passed back to the previous layer with some involvement of the weights connecting the layers and the neural activations of the input and the reference input. In fact, it was shown that DeepLIFT can be viewed precisely as backpropagation with the derivative of the activation function replaced with the slope between the input and reference input activations [6]. Thus, DeepLIFT computes a sort of smoothed gradient, helping with the issue of neuron saturation causing gradients to vanish. Comparison with the activations of the reference input allows one to more comprehensively capture the contribution of a particular neuron. A schematic of DeepLIFT is shown in Figure 3.6.

In practice, when working with DNA sequences, input times gradient is a cruder, perhaps more accessible, yet still effective attribution method. Mathematically, this means taking $x \odot \nabla_x f(x)$ as the attributions, where $\odot$ denotes element-wise multiplication and $x \in \{0,1\}^{4\times L}$ is a one-hot encoded DNA sequence (each column represents a base at a position in the sequence, with a 1 for the entry corresponding to the base at that position and a 0 for the entry of every other base). Example attributions computed from DeepLIFT vs. from input times gradient are compared in Figure 3.7.

TF-MoDISco

TF-MoDISco [106] is a method to compute motifs from attributions for genomic inputs. It can be broken down at a high level into three steps.

1. **Seqlet extraction**

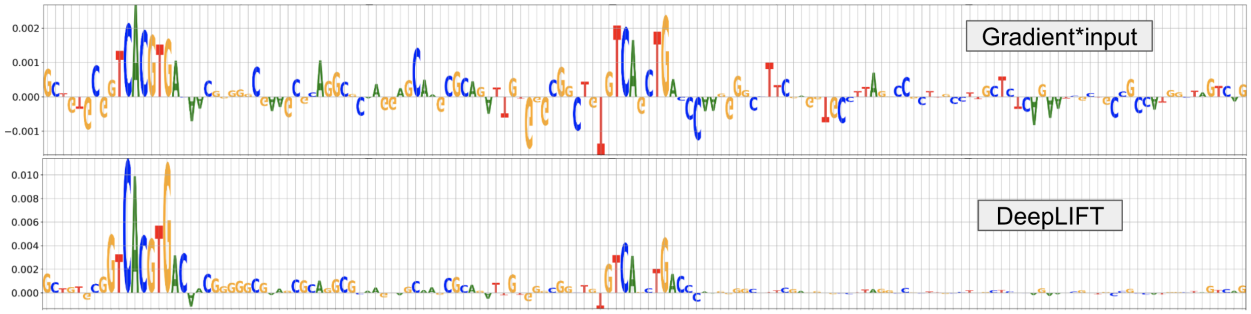


Figure 3.7: Comparison of DeepLIFT and input × gradient attributions for an input sequence. In this example, DeepLIFT tamps down gradient noise between the two attribution peaks that denote regions of importance to the model.

- The first step is to extract high attribution windows called seqlets from the provided input sequences. To do so, the total attribution in every window in every input sequence is computed, and a threshold is determined by fitting a Laplacian distribution to the distribution of window attributions and choosing the optimal threshold that discriminates between the actual distribution and this Laplacian null. Then, seqlets are found by iteratively taking windows passing the threshold from highest attribution to lowest, and removing any seqlets which overlap a taken seqlet by over 50%.

2. **Similarity-based clustering of seqlets**

   - Then, a similarity graph is constructed by computing the continuous Jaccard similarity between each pair of seqlets. Actually, for computational efficiency, the Jaccard similarity, which is relatively expensive, is only computed for a seqlet's 500 nearest neighbors according to a cheaper k-mer based similarity metric. Then, seqlets are clustered based on this graph to produce clusters. Each cluster will correspond to a motif.

3. **Aggregation to form motifs**

   - To generate the motif for a cluster, the seqlets in a cluster are aligned and averaged in attribution space to produce the motif.

There are many additional implementation details, such as running the clustering twice using seqlets updated from the previous clustering, and a step that splits apart clusters that contain meaningfully different subclusters. For a comprehensive description of the algorithm, see the manuscript [106] as well as the clean implementation of the algorithm, tfmodisco-lite [103]. A schematic of the simplified breakdown of TF-MoDISco is provided in Figure 3.8.

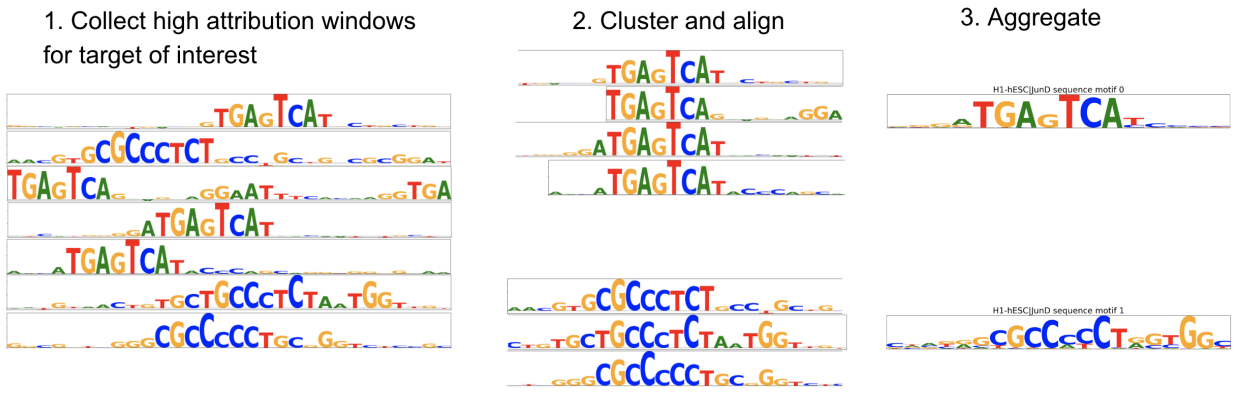


Figure 3.8: A simplified schematic of TF-MoDISco

.

In the default representation of a TF-MoDISco motif, the height of each letter equals the average attribution score given to that nucleotide at that position across the aligned seqlets in the cluster. Thus, the height of a letter corresponds to how important it is for the genomic task. Interestingly, when we use DeepLIFT attributions with respect to TF binding classificaton tasks, the TF-MoDISco motifs are extremely similar to the standard information content matrices used to visualize classical sequence motifs. Figure 3.9 shows the information content matrix representation of the CTCF motif from the CIS-BP [125] database of transcription factor motifs, along with an attribution-based CTCF motif from TF-MoDISco.

Because one has access to the underlying sequences of the seqlets in a TF-MoDISco cluster, it is straightforward to construct a classical motif representation from a TF-MoDISco cluster instead of an attribution-based one. One can simply count the frequency of each nucleotide at each position, convert the frequencies to probabilities by dividing by the number of seqlets, and then convert the probability matrix into the information content matrix. Mathematically, assuming a motif of length L, letting $F \in \mathbb{N}^{4\times L}$ denote the frequency matrix, $P \in [0,1]^{4\times L}$ the probability matrix, $I \in \mathbb{R}^{4\times L}$ the information content matrix, and $(x^{(k)})_{k=1}^{n} \subset \{0,1\}^{4\times L}$ the one-hot encoded sequences of the n seqlets, we have

$$
\begin{aligned}
F_{ij} &= \sum_{k=1}^{n} x_{ij}^{(k)} \\
P_{ij} &= \frac{1}{n} F_{ij} \\
I_{ij} &= P_{ij} \log\left(\frac{P_{ij}}{b_i}\right),
\end{aligned}
$$

where b_i is the background frequency of the ith nucleotide, and is around 0.25 if all four nucleotides occur similarly often.

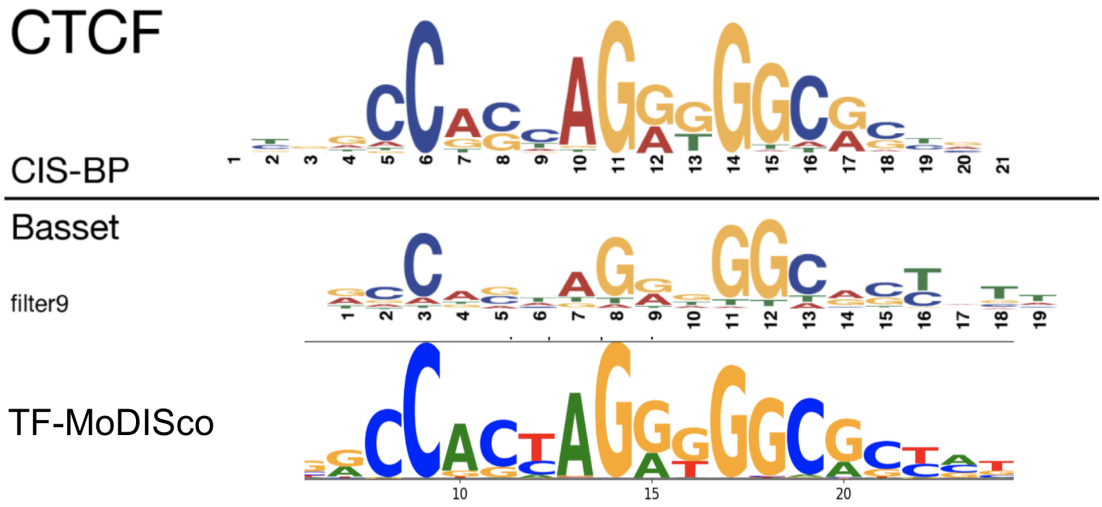


Figure 3.9: The top two rows are a figure taken from the Basset paper [59]. The top row displays the CTCF motif from the CIS-BP database, where letter heights equal per-position information content times the probability of each nucleotide. The middle row are the weights of a filter from the Basset convolutional neural network, indicating that neural networks learn parameters that resemble motifs. The bottom row is a motif extracted by TF-MoDISco from the attributions of a model trained to predict CTCF ChIP-seq. Though letter heights here represent DeepLIFT attribution, the precise variations in letter heights match those of the official information content-based motif very well.

3.2 Overview of the DeepShape model

Architecture

Here we present the DeepShape model, which takes in both bp-level sequence and shape features for an input DNA sequence and outputs binding probabilities for a list of given targets. DeepShape is a modified version of DeeperDeepSEA [19], which is a simple but effective convolutional neural network that only takes in sequence features. Figure 3.11 shows a diagram of the unmodified DeeperDeepSEA architecture. Sequence features come as a 4 by sequence length one-hot encoding of the DNA sequence, while shape features come as a 5 by sequence length real-valued matrix, with each row containing the values of a particular shape feature along the sequence. DeepShape processes both of these inputs in separate input branches, and then uses their combined embeddings to perform various genomic classification tasks. The 5 shape features we use are minor groove width (MGW), helical twist (HelT), propeller twist (ProT), roll (Roll), and electrostatic potential (EP).

The modifications made to DeeperDeepSEA to obtain DeepShape are as follows. First, the first two convolutional layers were copied to obtain two separate input branches of two

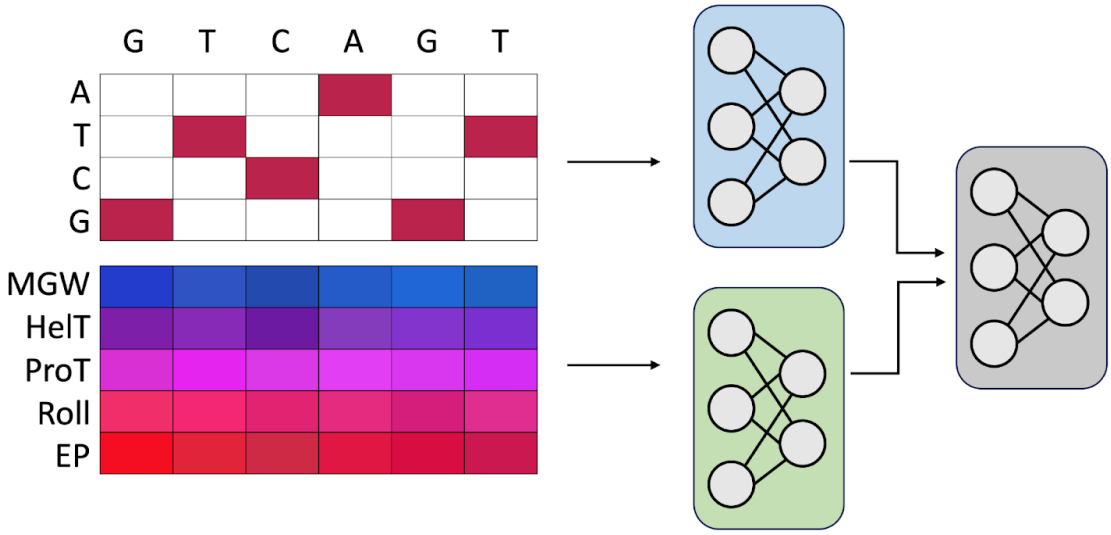


Figure 3.10: Schematic overview of the DeepShape model. DeepShape is a convolutional neural network that takes in DNA sequence and DNA shape features in separate initial branches. The outputs of each branch are concatenated and fed to the rest of the network to predict genomic outputs.

convolutional layers each. The sequence input was fed into one branch and the shape input was fed into the other, and the outputs of each branch were subsequently combined and fed into the rest of the network. This captures the idea that different modalities may require some initial processing to arrive at compatible representations, and a schematic of this is presented in Figure 3.10. Furthermore, a hyperparameter search over filter width and number of filters per convolutional layer found that a constant 450 filters per convolutional layer with filter width 4 performed better than the original configuration in which the number of filters increased along the network according to the sequence 320-480-960, and the filter width was 8. Thus, the convolutional parameters were changed to the higher performing ones.

Prediction task details

DeepShape can be trained on any task that involves predicting one or multiple binary labels for an input genomic sequence. We focus on an instance of DeepShape that takes in 1000bp sequences from the human genome as input, and predicts whether the center 200bp of the input sequence has at least 100bp of overlap with peaks from the 919 DNase accessibility, histone modification ChIP-seq, and TF ChIP-seq experiments used to train DeepSEA [136].

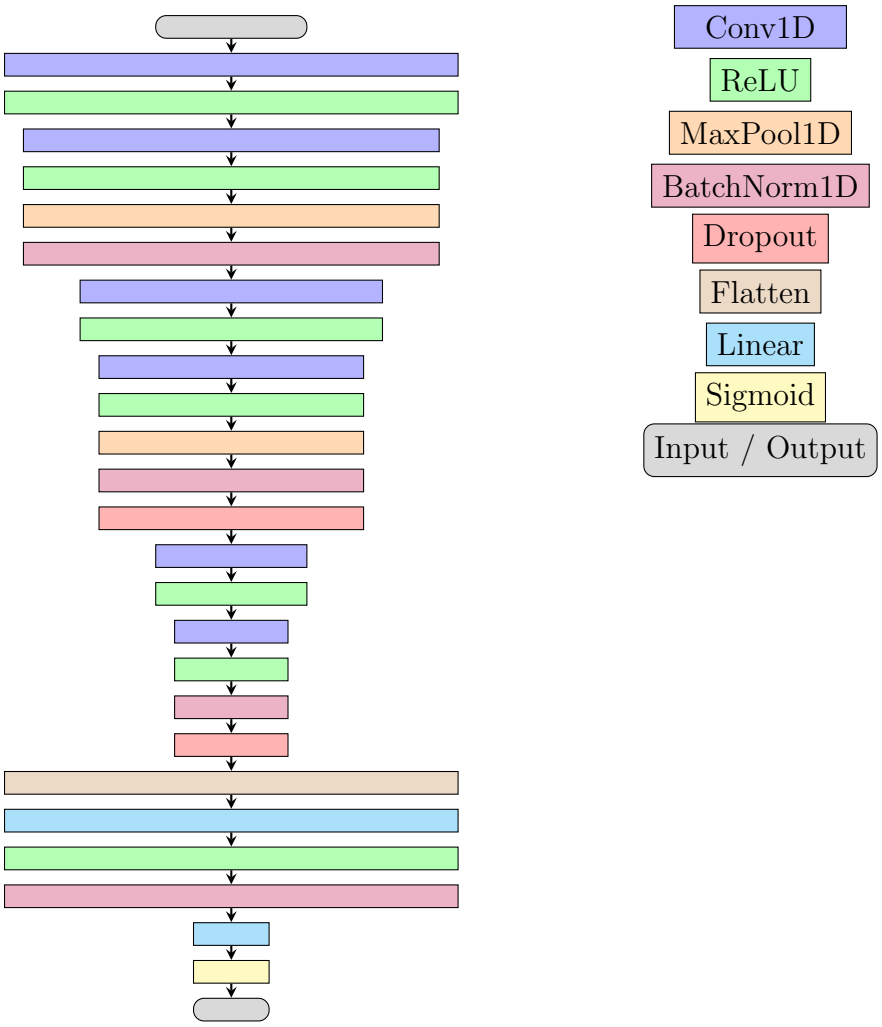


Figure 3.11: Diagram of the DeeperDeepSEA architecture. It consists of 6 convolutional layers followed by flattening and fully connected layers. ReLU [37], batch norm, [50], max pooling, and dropout [113] are employed throughout the network.

The output for an input sequence is a length 919 vector containing the probability of overlap for each target. For TF ChIP-seq experiments, peaks correspond to binding sites, so the network thus predicts the probability of TF binding around the center of the input sequence.

We utilize the Selene [19] library, a PyTorch-based genomic deep learning framework. 1000bp intervals in the human genome are sampled randomly from a predefined set of larger human genome intervals that are likely to contain TF activity in some cell type. Each shape feature is normalized to lie within the range $[0, 1]$. The 919-target model is trained for 960,000 steps with a batch size of 64 using stochastic gradient descent (SGD), with a constant learning rate of 0.08, a weight decay parameter of 1e-6 [65], and a momentum parameter of 0.9. The loss function is the standard binary cross entropy loss used for binary classification problems. We split the data into training, validation, and test sets based on chromosome, with chromosomes 6 and 7 reserved for validation and chromosomes 8 and 9 for test. When training different replicates, we change the validation and test chromosomes.

Model performance

In Figure 3.12, we compare the performance of the 919-target DeepShape over 6 replicates to that of DeeperDeepSEA, modified to use the same filter parameters as DeepShape. The only difference between the two models is that DeepShape takes in shape features in addition to sequence features via a separate input branch. We use the area under the receiver operating characteristic curve (ROC AUC, or AUC for brevity) and average precision to measure binary classification performance. We observe that under both metrics, both models obtain about the same performance. Therefore, the addition of shape features did not produce a meaningful gain in performance. This is in line with previous works which demonstrated that shape features perform similarly to using k-mer sequence features in shallow models where $k \geq 2$ [52, 39, 137, 75].

We conclude that shape features are mostly redundant in the context of deep models, which can recover the relevant shape information from sequence. This may be perceived as an expected result, as the shape features are directly derived from the surrounding 5-mer [115], but note that pretrained features have been found to significantly improve downstream performance in vision [20] and text [92], and one can think of these precomputed shape features as pretrained features. Perhaps the simplicity of the 5-mer lookup table prevents them from contributing new information.

Nonetheless, because of the interest in shape readout as a binding mechanism, interpreting DeepShape remains valuable for understanding how shape features predict TF binding. First, we permute the shape features between different input sequences in the same test batch, destroying the relationship between shape features and outcome, to see how this impacts model performance on all targets. We expect to see a larger drop in performance for targets that rely more on shape information. The results of this perturbation experiment are shown in Figure 3.13, and as expected, performance drops across the board, with larger drops for certain targets. Thus, relevant information is being extracted from the shape fea-

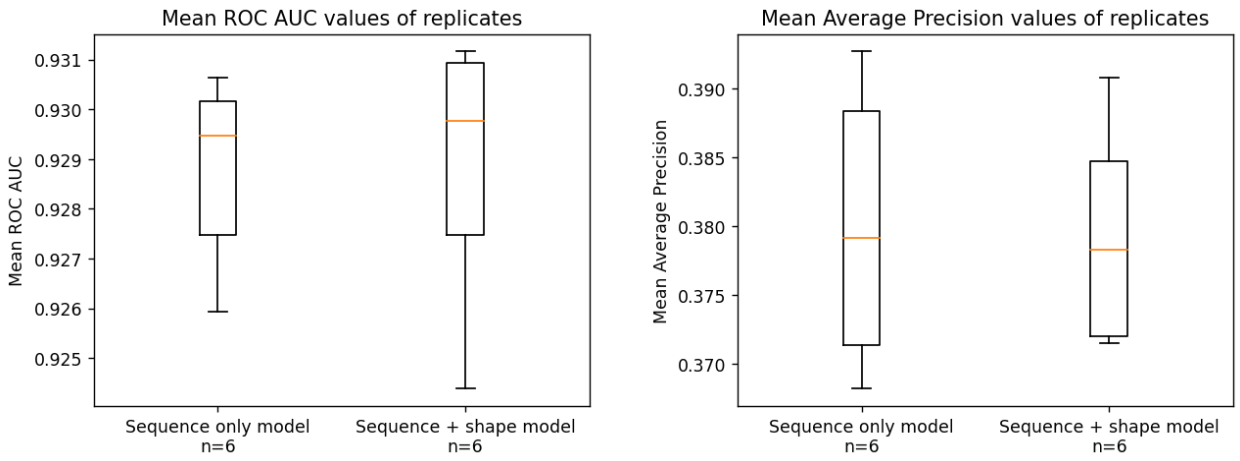


Figure 3.12: Performance of DeepShape (sequence + shape) and DeeperDeepSEA (sequence only) modified to have the same filter parameters as DeepShape across 6 replicates for the 919-target setup. The ROC AUC and average precision are computed for each target, and then averaged across targets to produce the scalar performance measure achieved by a replicate. Both models perform about the same, suggesting that the shape features are not adding new information.

tures. In particular, the HepG2|MafK transcription factor target experiences a large drop, suggesting that shape information is particularly relevant in MafK binding in HepG2 cells.

3.3 Attribution-based DeepShape analyses

In this section, we use DeepLIFT [105] to compute target-specific attributions for DeepShape trained on 919 targets and we run TF-MoDISco [106] on DeepLIFT attributions to extract motifs. In order to extract shape motifs in addition to the standard sequence motifs, we quaternize each shape feature using feature quartiles as cutoffs, to put them in the same format as one-hot encoded sequence features. This allows us to run TF-MoDISco on shape features with minimal modifications to the TF-MoDISco code. We use the implementation of DeepLIFT from Captum[1] [63] and tf-modiscolite [103].

[1]There is a bug in Captum's max-pooling attributions, as noted here: `https://x.com/jmschreiber91/status/1782837223064539515`. We implemented our own, lightweight fix.

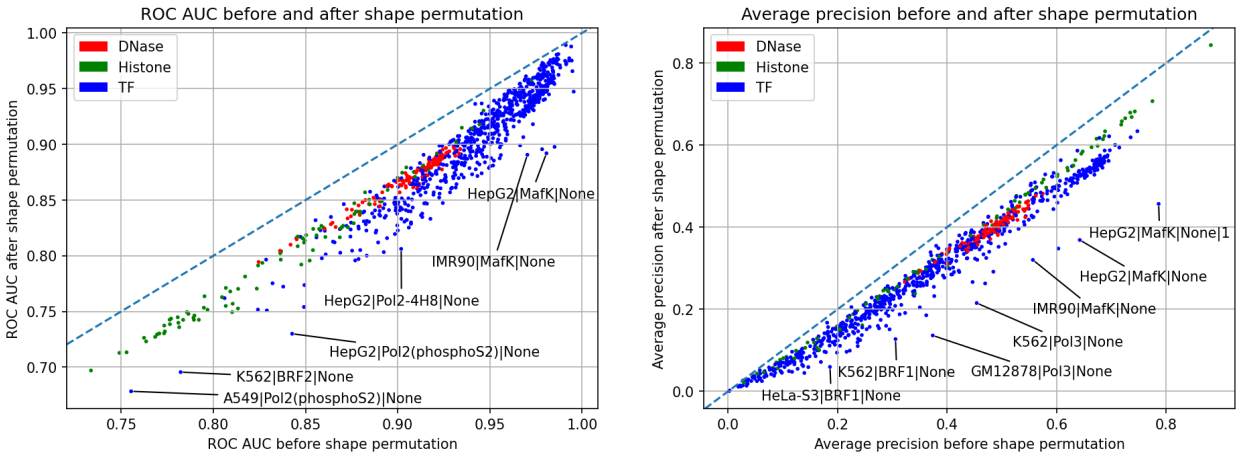


Figure 3.13: Performance before (x-axis) and after (y-axis) permuting shape features, averaged over the 6 replicates for each of the 919 targets. ROC AUC (left) and average precision (right) are shown. The dashed lines are $y = x$, so the vertical distance below the line equals the drop in performance after shape permutation. All of the dots lie below the line, indicating a consistent decrease in performance. TFs have higher AUCs than histone modification targets, but lower average precision. This is because average precision is mediated by the frequency of a target, and histone modifications are more frequent in the genome than TF binding. Higher AUCs for TF targets means that it is easier to tell a random positive from a random negative for TF binding than for histone modifications. MafK, BRF, and Pol targets show pronounced drops, suggesting higher relevance of shape information for these DNA-binding proteins. Very high AUC TF targets do not drop by much, suggesting that these TFs possess very obvious sequence motifs and do not require much shape information to distinguish their binding.

Target-level shape attribution is mostly a constant fraction of sequence attribution

We compare the usage of sequence features versus shape features in predicting TF binding, based on DeepLIFT attributions. Hypothetically, TFs that employ shape readout would have a higher ratio of shape to sequence attribution than TFs that primarily use sequence readout. Figure 3.14 plots the overall sequence and shape attribution for each TF target. Most targets hew to the line of best fit, where shape attribution is approximately 0.27 times sequence attribution, indicating that sequence features are generally more relevant for predicting TF binding.

Interestingly, the ratio of shape to sequence attribution is quite consistent across different TFs, suggesting uniformity in the influence of shape readout relative to sequence readout. However, such uniformity in the actual biophysics of binding across different TFs seems

unlikely. Instead, this uniformity might result from redundant information being extracted from both feature types. If shape features were simply a noisy version of sequence features, we would also expect them to have a small, constant fraction of the sequence features' attribution. Targets above the line, such as HepG2|MafK, have higher shape-to-sequence attribution ratios, implying a greater role for shape readout in their binding. Thus, HepG2|MafK is included in further analyses.

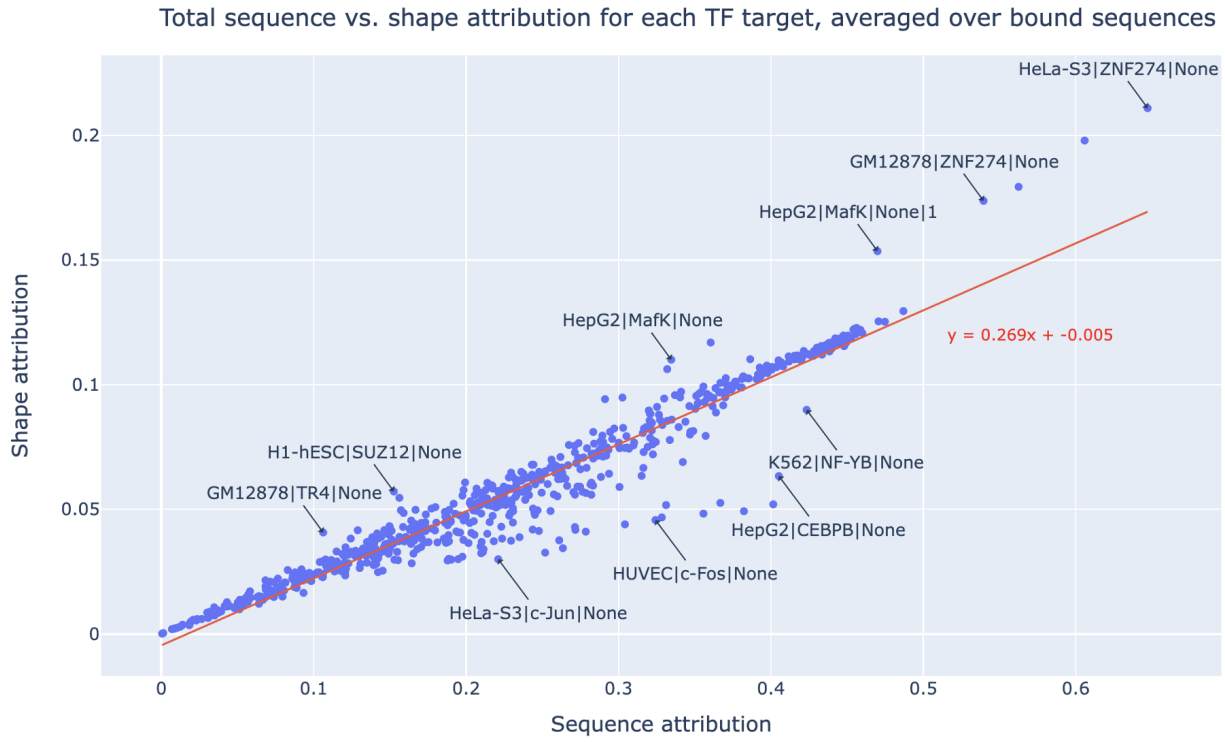


Figure 3.14: Overall sequence and shape attribution per TF target, averaged over 6 replicates. We compute the overall sequence attribution for a target as follows. For each positive (i.e., bound by the given TF in the given cell type) sequence, we sum all sequence attributions across the sequence to obtain a single value, and average this value across positive sequences. We obtain the overall shape attribution for a TF target analogously. The best-fit line, $y = 0.269x - 0.005$, captures most of the targets. We label some targets which deviate from the line.

Shape motifs tend to highlight extreme values and brief segments

Figure 3.15 displays example motifs obtained by inputting DeepLIFT attributions to TF-MoDISco to explain HepG2|MafK binding predictions. The sequence motif (Figure 3.15a) matches the canonical TGCTGA(G/C)TCAGCA palindromic MafK motif [56], showcasing

the ability of deep learning to learn true biological motifs, and the ability of interpretation methods to extract them. On the other hand, the extracted Roll motif (Figure 3.15b) may represent a new biological pattern for MafK binding. Notably, it mainly highlights three Roll value peaks. Thus, high Roll values within a MafK binding site appear to facilitate MafK binding.

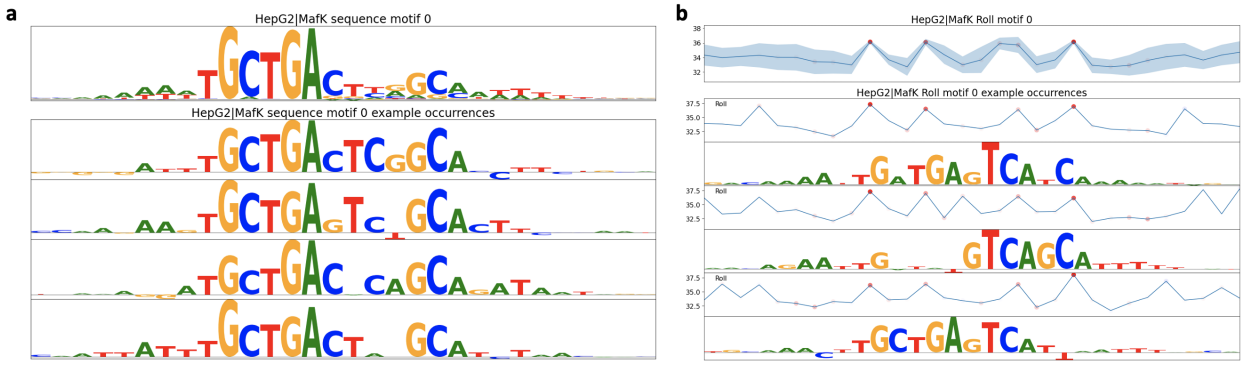


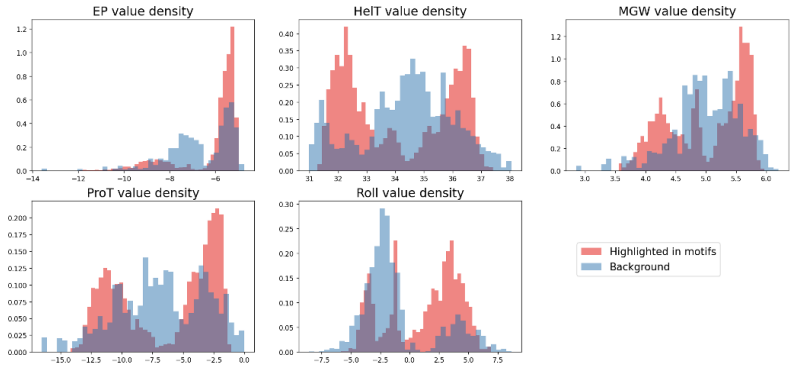
Figure 3.15: A sequence and a shape motif found by TF-MoDISco. Example occurrences of each motif are listed underneath the motif. For shape motif occurrences, the underlying sequence is shown underneath the shape. **a**, Sequence motif 0 for HepG2|MafK. The sequence motif matches the canonical TGCTGA(G/C)TCAGCA palindromic MafK motif. **b**, Roll motif 0 for HepG2|MafK.

Figure 3.16a shows that in fact, shape motifs tend to highlight extreme feature values in general. For each shape feature, the distribution of highlighted feature values compiled across the motifs of different targets is shown to be skewed toward the extremes of the background distribution of feature values. This suggests that DNA shape at binding sites tends to have extreme characteristics, even in its unbound state. Figure 3.16b shows that sequence motifs tend to have more highlighted positions than shape motifs, suggesting that sequence motifs better capture the classic conception of a motif being a contiguous stretch of DNA that transcription factors bind to, whereas shape motifs often highlight specific positions or small fragments of DNA that subtly increase binding affinity.

A flanking shape motif increases binding affinity to the canonical HepG2|MafK sequence motif

Next, we examine the interaction between a sequence motif and a ProT motif identified by TF-MoDISco for HepG2|MafK (Figure 3.17). The sequence motif matches the canonical TGCTGA(C/G)TCAGCA MafK motif, while the ProT motif features a high attribution region with a peak followed by a pronounced dip. High attribution positions are visually represented by tall letters for sequence motifs and red dots for shape motifs. The ProT motif

a



b

	Average number of highlighted positions per motif
Sequence	5.46
EP	3.72
HelT	3.35
MGW	3.02
ProT	3.19
Roll	3.37

Figure 3.16: Analysis of highlighted positions in motifs. Here, a highlighted position is defined as a position in the motif with a high attribution relative to the rest of the motif. **a**, The distribution of shape feature values in the background (blue) vs. the distribution of shape feature values at highlighted positions in motifs. **b**, The average number of highlighted positions in a motif for each feature type.

often co-occurs with the sequence motif at the end of the sequence motif and the nearby flanking region. Here, an occurrence of a motif is defined as a seqlet within the corresponding TF-MoDISco motif cluster. In Figure 3.17d, examples of co-occurrences show that the ProT dip corresponds to an A-tract or T-tract flanking the sequence motif, which previous studies have shown significantly affect DNA shape [96]. The ProT motif may therefore constitute an instance of A-tracts/T-tracts influencing TF binding through their effects on DNA shape.

While informative, sequence motifs alone do not fully capture TF binding. Nearby DNA shape can influence TF binding, potentially enabling binding to a weak sequence motif or preventing binding to a strong sequence motif. To evaluate the role of the ProT motif, we compared the experimental binding affinity of sequence motif-only occurrences with that of sequence motif–ProT motif co-occurrences. As shown in Figure 3.17c, strong sequence motif occurrences that co-occur with the ProT motif have higher binding affinity as measured by ChIP-seq enrichment signal (ChIP-score) than those that do not co-occur with the ProT motif (p-value = 1.0e-3, one-sided t-test). This corroborates previous findings that shape features in flanking regions can modulate binding to a canonical sequence motif [39, 129].

Sequence motifs outweigh shape motifs for predicting TF binding

To compare the predictive value of sequence and shape motifs for TF binding, we extracted the five most predictive sequence motifs and five most predictive shape motifs for each TF target. Then, we used the combined set of 10 motifs to predict TF binding from sequence. Each 400bp input sequence was featurized by representing each motif as a probability distri-

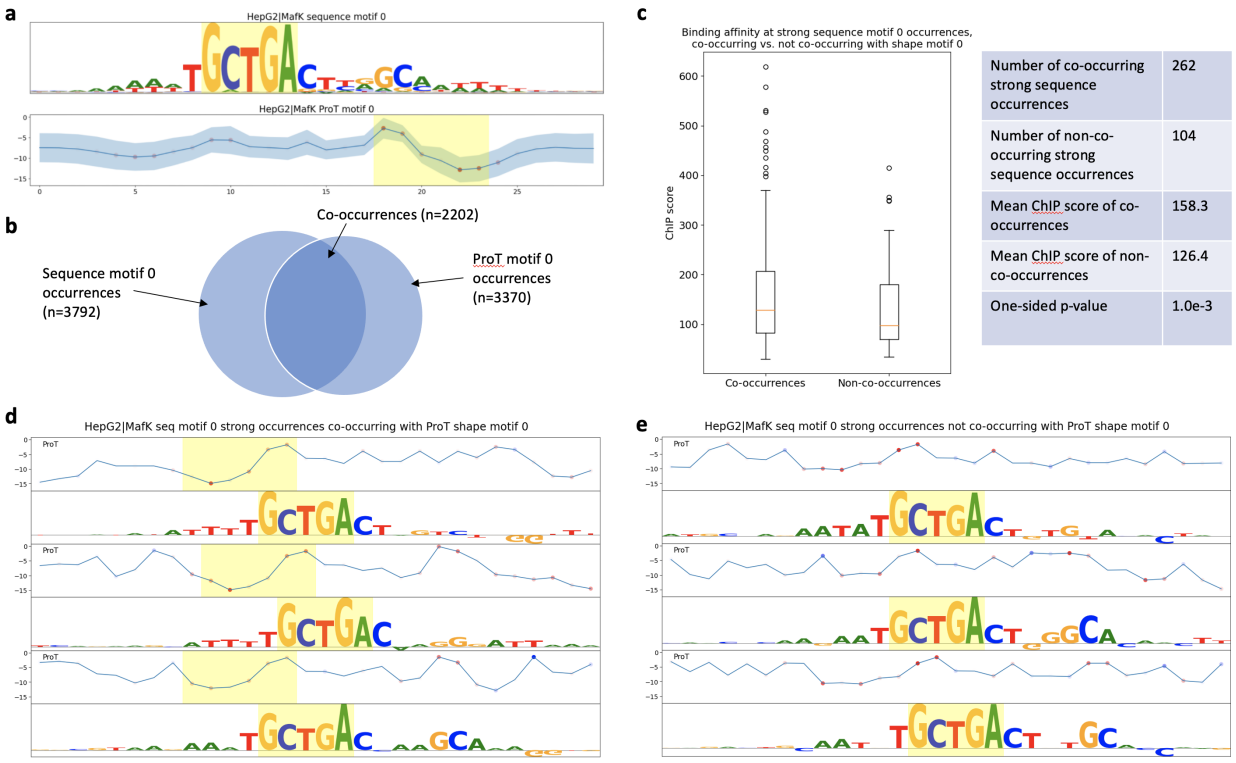


Figure 3.17: Co-occurrence analysis of a HepG2|MafK sequence motif and ProT motif found by TF-MoDISco. **a**, Sequence logo for the sequence motif (top) and average profile with spread for the ProT motif (bottom). High attribution regions of motifs are highlighted. **b**, Number of occurrences of each motif and co-occurrences. A co-occurrence was defined as a pair of occurrences from each motif, such that their highlighted regions overlap or are within 10bp of each other. **c**, ChIP-seq scores of strong sequence motif occurrences (>90 percentile Pearson similarity of attributions) that co-occur and do not co-occur with the ProT motif. **d**, Three examples of strong sequence motif occurrences that co-occur with the ProT motif. For each occurrence, the ProT sequence is shown on top, and the underlying nucleotide sequence is shown below. Highlighted regions correspond to the highlighted region in the motif. **e**, Three examples of strong sequence motif occurrences that do not co-occur with the ProT motif.

bution at each position based on motif cluster seqlets, calculating the motif vs. background log-likelihood ratio for each window [41], and taking the maximum score across all windows. This represents the strength of the strongest match to the motif within the sequence. Given a set of M motifs, this featurizes the sequence as an M-dimensional vector. We run logistic regression and gradient boosting on the 10-dimensional feature vectors resulting from these motifs. This process is diagrammed in Figure 3.18a.

In about 90% of cases (Figure 3.18c, left), both logistic regression and gradient boosting importance scores identified a sequence motif as the most important input feature, indicating that TF binding is primarily captured by sequence motifs. However, there was little ($< 30\%$) overlap between targets where a shape motif had the highest logistic regression importance score and targets where a shape motif had the highest gradient boosting importance score. This suggests that these TF targets do not have a particular affinity for shape motifs, but have binding that is hard to explain with a single motif in general.

In Figure 3.18d, we evaluated the impact of ablating each motif from the gradient boosting classifier and retraining on the remaining nine motifs. Performance drops across targets are grouped by the rank of the drop within sequence motifs or shape motifs. The highest sequence motif AUC drop significantly exceeds others, showing that a single sequence motif is often the primary determinant of TF binding. The sequence motif with the second highest AUC drop typically belongs to another TF and represents the effect of co-binding. Despite sequence motifs' primary role in TF binding, shape motifs still add some predictive value. This is evidenced by the highest AUC drop from ablating a shape motif still being significant (Figure 3.18d). If shape motifs did not add predictive value, ablating one would not result in a noticeable positive AUC drop.

HepG2|MafK sequence-level attributions vary in the ratio of shape attribution to sequence attribution

In Figure 3.19a, the total sequence and shape attributions for individual sequences bound by HepG2|MafK are plotted. We observe a moderate variation in the ratio of shape to sequence attribution across individual sequences, unlike the uniform pattern seen across TF targets. A low shape attribution ratio sequence (Figure 3.19c) features a strong MafK sequence motif, with a matching Roll motif, but its ProT sequence does not match the ProT motif, as it lacks the characteristic dip of the ProT motif. Conversely, a high shape attribution ratio sequence (Figure 3.19b) has a weaker sequence motif, but stronger shape features. Its Roll sequence contains high attribution peaks, like the Roll motif, and its ProT sequence exemplifies the ProT motif dip and has many high attribution positions. This suggests that the shape-to-sequence attribution ratio captures differences in binding mechanisms across sequences.

High and low shape-to-sequence attribution ratio sequences are associated with different chromatin states

To capture any systematic biological differences between low and high shape-to-sequence attribution ratio sequences, we examine their chromatin state annotations. We obtain annotations for the human HepG2 cell line from ChromHMM [28], a hidden Markov model that uses histone modification and CTCF binding marks as input, and outputs promoter, enhancer, transcription region, and heterochromatin annotations across the genome. A bound

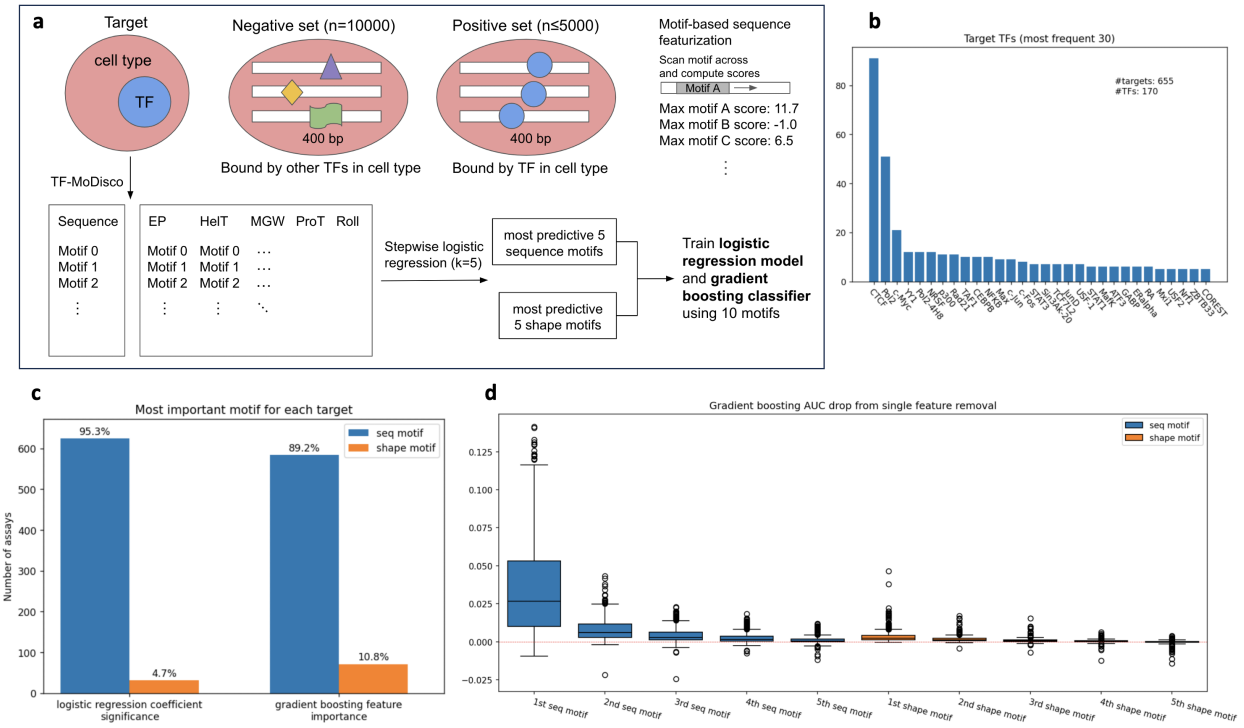


Figure 3.18: Predicting TF binding with TF-MoDISco-derived sequence motifs and shape motifs. **a**, Diagram of motif-based binding prediction experiment setup. **b**, TF composition of TF targets. **c**, The percentage of TF targets where out of the 10 selected motifs, a sequence / shape motif was deemed most important, according to logistic regression coefficient significance and gradient boosting feature importance. **d**, Performance drops resulting from ablating each individual motif and retraining the gradient boosting model. "ith sequence (shape) motif" refers to the sequence (shape) motif for that TF target which resulted in the ith highest AUC drop.

sequence was labeled as the annotation with the greatest overlap with its central 400bp region, which is also the location of the binding site. We then compare the annotations of low and high ratio sequences for the HepG2|MafK and HepG2|CEBPB targets in Figure 3.20 and Figure 3.21 respectively. HepG2|MafK as an overall target had a high shape-to-sequence attribution ratio, while the HepG2|CEBPB target had a low overall shape-to-sequence attribution ratio (Figure 3.14). However, both targets demonstrate interesting variation in chromatin properties as attribution ratio varies within their bound sequences.

High shape ratio HepG2|MafK-bound binding sites more often occur in heterochromatin and less often in enhancers than their low ratio counterparts (Figure 3.20c,d). Oppositely, high shape ratio HepG2|CEBPB-bound binding sites more often occur in enhancers and less often in heterochromatin than low ratio ones (Figure 3.21c,d). Interestingly as well,

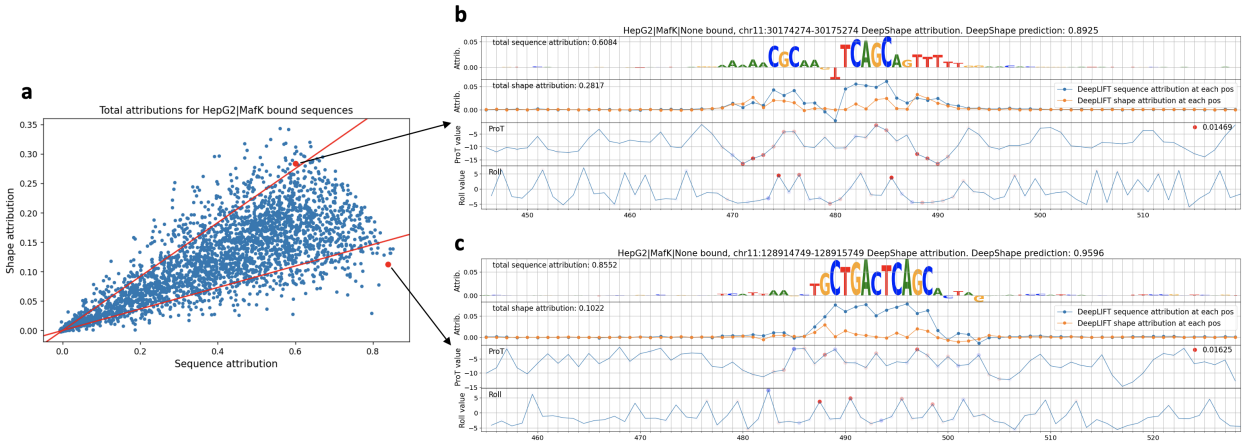
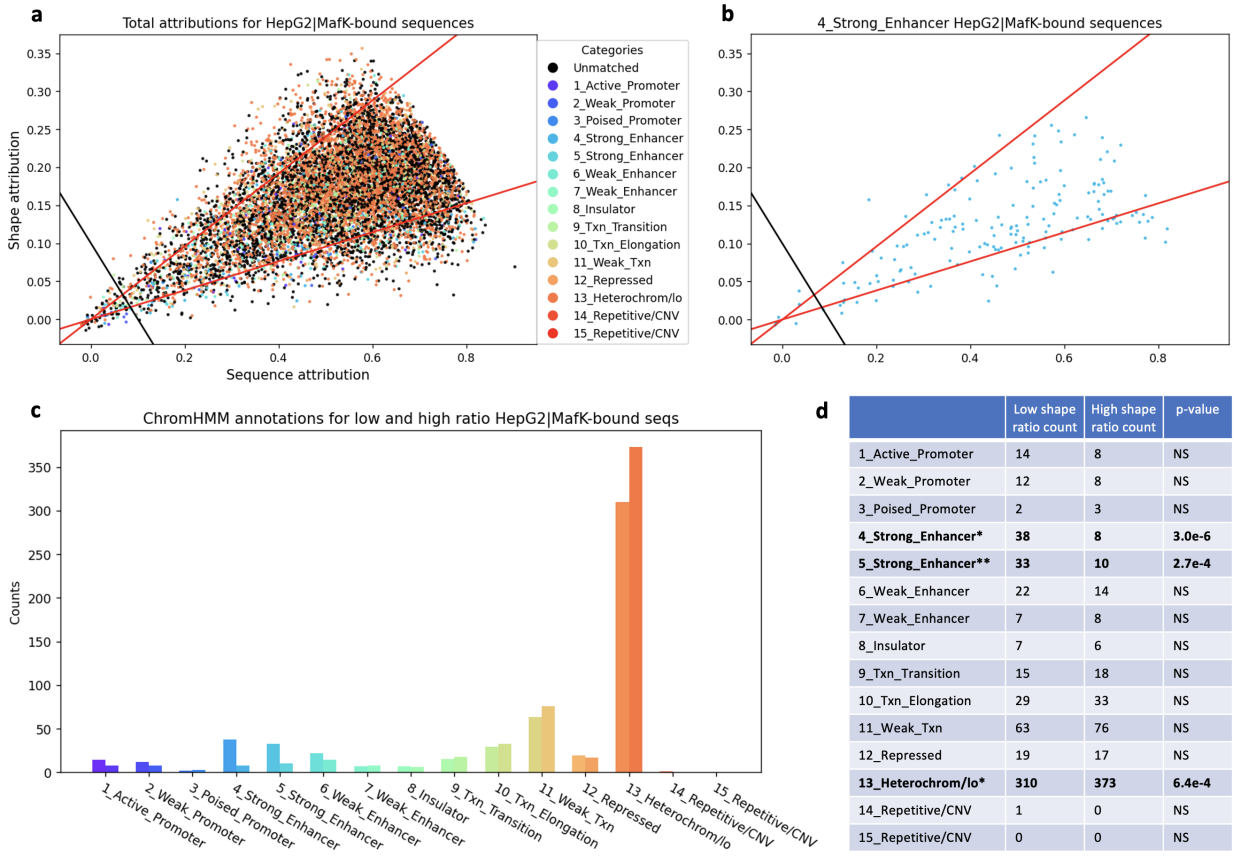


Figure 3.19: Sequence-level sequence and shape attributions of HepG2|MafK-bound sequences. **a**, Each HepG2|MafK-bound sequence is plotted as a point. The x-value is the sum of all sequence attributions across the sequence and the y-value is the sum of all shape attributions across the sequence. Sequences with higher shape attribution to sequence attribution ratios lie above the steeper red line and those with lower ratios lie below the less steep red line. **b**, A high ratio sequence, where shape attribution makes up more of the total attribution. **c**, A low ratio sequence, where total attribution is dominated by sequence attribution.

low shape ratio HepG2|CEBPB-bound sites occur in transcribed regions more often than high ratio ones. Together, this shows that the shape-to-sequence attribution ratio is related to biological properties of the sequence, but it can have opposite relationships in different targets. The prevalence of HepG2|MafK binding sites in heterochromatin may be related to the known role of MafK as a silencer that acts by remodeling chromatin [35].

High shape-to-sequence attribution ratio sequences differ from low ratio sequences in motif occurrences

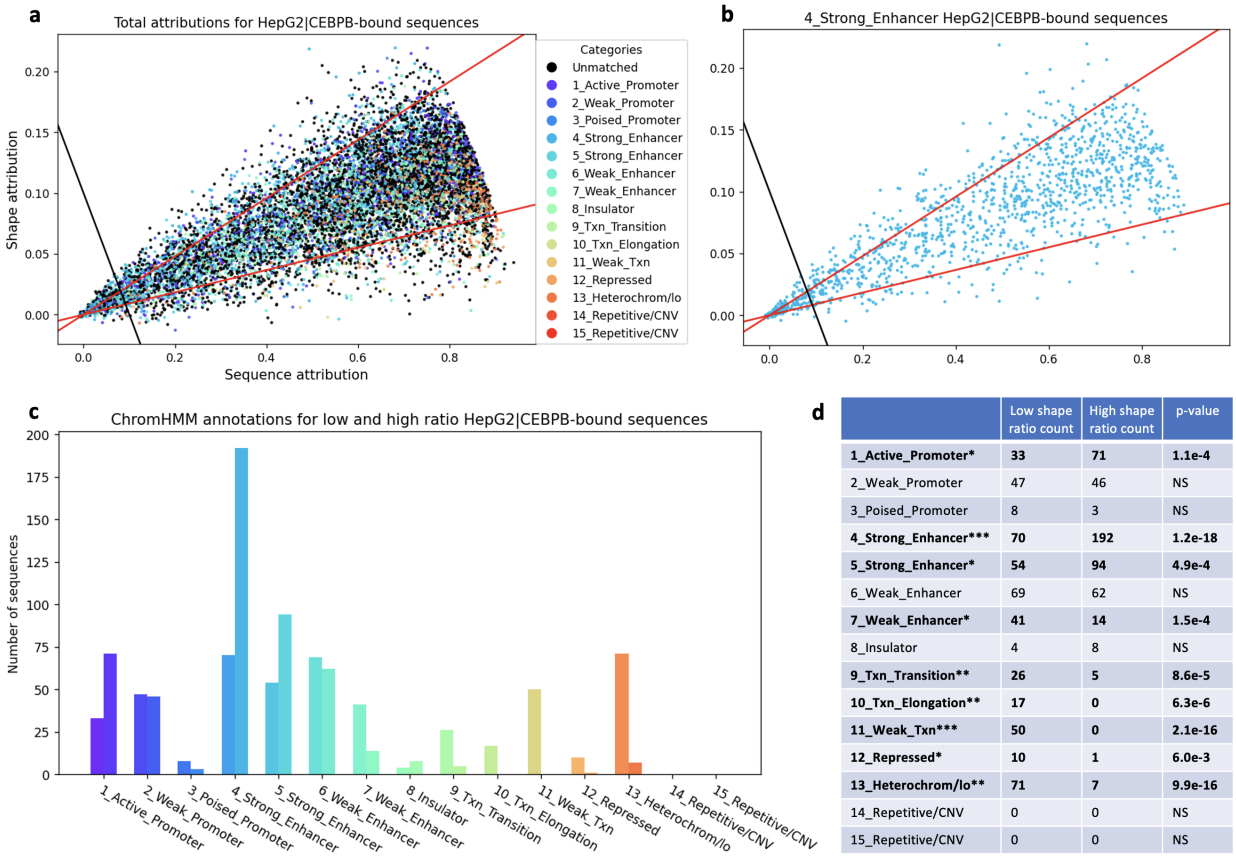
Because a high shape-to-sequence attribution ratio indicates that shape features contribute a higher proportion of DeepShape's prediction, we suspect that shape information is more important in high ratio sequences. However, it is helpful to further characterize high ratio sequences to better understand how DNA shape is impacting TF binding. A hypothesis is that high shape-to-sequence attribution ratio sequences have a higher ratio of shape motifs to sequence motifs than low ratio sequences. We test this hypothesis by computing the number of sequence and shape motif occurrences in high and low ratio sequences, for HepG2|MafK and HepG2|CEBPB. For each target, we restrict to the 5 most predictive motifs per feature type in terms of AUC, and count the number of occurrences in the high and low ratio



	Low shape ratio count	High shape ratio count	p-value
1_Active_Promoter	14	8	NS
2_Weak_Promoter	12	8	NS
3_Poised_Promoter	2	3	NS
4_Strong_Enhancer*	**38**	**8**	**3.0e-6**
5_Strong_Enhancer**	**33**	**10**	**2.7e-4**
6_Weak_Enhancer	22	14	NS
7_Weak_Enhancer	7	8	NS
8_Insulator	7	6	NS
9_Txn_Transition	15	18	NS
10_Txn_Elongation	29	33	NS
11_Weak_Txn	63	76	NS
12_Repressed	19	17	NS
13_Heterochrom/lo*	**310**	**373**	**6.4e-4**
14_Repetitive/CNV	1	0	NS
15_Repetitive/CNV	0	0	NS

Figure 3.20: ChromHMM annotations of HepG2|MafK-bound sequences. **a**, Total sequence and shape attribution plotted for each HepG2|Mafk-bound sequence. Each sequence (dot) is colored with its assigned ChromHMM state. Sequences that did not overlap any annotation are labeled "Unmatched". High ratio sequences ($n = 989$) lie above the higher slope red line, and low ratio sequences ($n = 989$) lie below the lower slope red line. Only sequences with a minimum attribution sum, those above the black line, are considered. **b**, Total sequence and shape attribution plot for HepG2|Mafk-bound sequences annotated as ChromHMM state 4, strong enhancer. **c**, Number of low shape-to-sequence ratio sequences (left bar at each x-axis position) and high shape-to-sequence ratio sequences (right bar) annotated as each state. **d**, Number of low and high ratio sequences for each state as a table, with p-values from a two-sided Fisher's exact test.

subsets identified previously, each being around 1000 sequences. To call occurrences, we use the position-wise probability distribution representation of each motif to compute log likelihood ratio scores of windows across 400bp input sequences, like in the previous section on using motifs to predict TF binding, and use the 95th percentile over a negative set of sequences of the maximum motif score across the sequence as the occurrence threshold.



	Low shape ratio count	High shape ratio count	p-value
1_Active_Promoter*	**33**	**71**	**1.1e-4**
2_Weak_Promoter	47	46	NS
3_Poised_Promoter	8	3	NS
4_Strong_Enhancer***	**70**	**192**	**1.2e-18**
5_Strong_Enhancer*	**54**	**94**	**4.9e-4**
6_Weak_Enhancer	69	62	NS
7_Weak_Enhancer*	**41**	**14**	**1.5e-4**
8_Insulator	4	8	NS
9_Txn_Transition**	**26**	**5**	**8.6e-5**
10_Txn_Elongation**	**17**	**0**	**6.3e-6**
11_Weak_Txn***	**50**	**0**	**2.1e-16**
12_Repressed*	**10**	**1**	**6.0e-3**
13_Heterochrom/lo**	**71**	**7**	**9.9e-16**
14_Repetitive/CNV	0	0	NS
15_Repetitive/CNV	0	0	NS

Figure 3.21: ChromHMM annotations of HepG2|CEBPB-bound sequences. **a**, Total sequence and shape attribution plotted for each HepG2|CEBPB-bound sequence. Each sequence (dot) is colored with its assigned ChromHMM state. Sequences that did not overlap any annotation are labeled "Unmatched". High ratio sequences ($n = 949$) lie above the higher slope red line, and low ratio sequences ($n = 949$) lie below the lower slope red line. Only sequences above the black line are considered. **b**, Total sequence and shape attribution plot for ChromHMM state 4, strong enhancer-annotated HepG2|CEBPB-bound sequences. **c**, Number of low shape-to-sequence ratio sequences (left bar at each x-axis position) and high shape-to-sequence ratio sequences (right bar) annotated as each state. **d**, Number of low and high ratio sequences for each state as a table, with p-values from a two-sided Fisher's exact test.

The resulting total motif count across all 5 motifs considered for each feature type in low and high ratio sequences is shown in Table 3.1. For the HepG2|MafK target, as one might expect, shape motifs occur much more frequently in high ratio sequences, while sequence motifs occur about equally often in high and low ratio sequences. Thus, for this target, higher importance of shape information can be explained by the DNA shape exhibiting

HepG2|MafK

	Sequence	EP	HelT	MGW	ProT	Roll
MOC in LRS	2120	425	961	579	887	627
MOC in HRS	2114	2226	2017	1735	1739	1539

HepG2|CEBPB

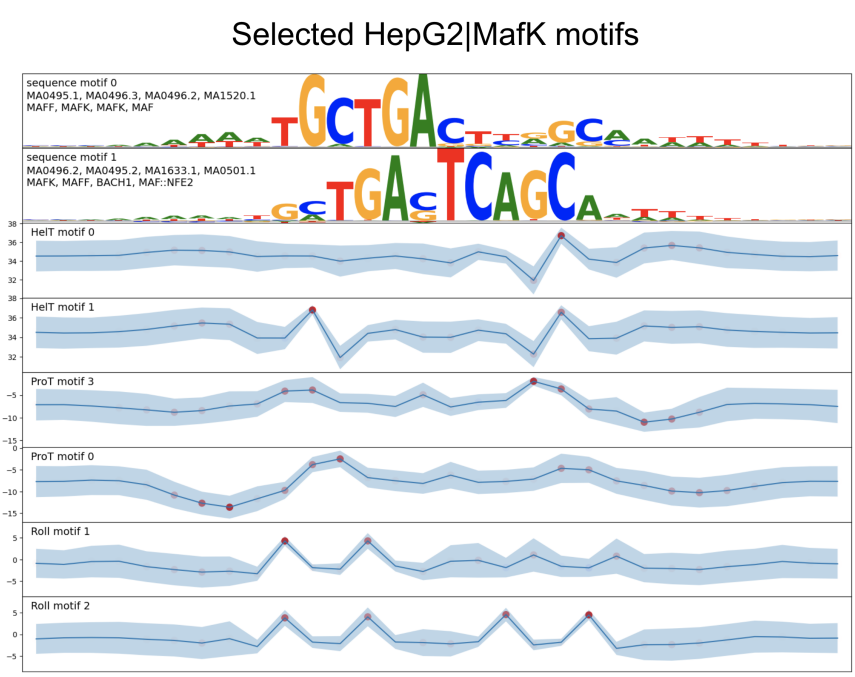
	Sequence	EP	HelT	MGW	ProT	Roll
MOC in LRS	885	1083	610	1242	682	1009
MOC in HRS	361	264	421	271	342	370

Table 3.1: Motif occurrence counts in low and high shape-to-sequence attribution ratio HepG2|MafK and HepG2|CEBPB sequences. MOC: motif occurrence count, LRS: low ratio sequences, HRS: high ratio sequences.

more shape motifs. For the HepG2|CEBPB target, counterintuitively, not just sequence motifs, but shape motifs too occur much more frequently in low ratio sequences. Thus, low ratio sequences are characterized by strong motif occurrences. Perhaps for this target, shape motifs capture different variants of the main CEBPB sequence motif, which occurs starkly more often in low ratio sequences (846 vs. 166 for low vs. high ratio sequence, Figure 3.23).

In Figure 3.22, we visualize some specific HepG2|MafK motifs as well as occurrence counts for each visualized motif in low and ratio HepG2|MafK sequences. We find that palindromic motifs (sequence motif 1, HelT motif 1, ProT motif 3, Roll motif 2) are skewed towards low ratio sequences, while asymmetric motifs are skewed towards high ratio sequences. This suggests that the occurrence of a full, palindromic, TGCTCA(G/C)TGAGCA MafK sequence motif results in a low shape-to-sequence attribution ratio due to binding obviously being explained by the occurrence of this strong sequence motif. Palindromic shape motifs are also likely to accompany occurrences of the palindromic sequence motif. In Figure 3.23, we perform the same visualization for HepG2|CEBPB. Here, we see that the main CEBPB sequence motif has a very strong skew towards low ratio sequences, again supporting the idea that low ratio sequences often contain strong sequence motif occurrences, in which shape information is not needed. Interestingly, the palindromic CEBPB shape motifs are less skewed towards low ratio sequences, which is the opposite of what was seen for HepG2|MafK. Perhaps this is due to the sequence motif itself being almost but not quite palindromic in most of its occurrences.

In Figure 3.24, we examine the relative positioning of HepG2|Mafk shape motifs with respect to the main sequence motif, sequence motif 0, when they co-occur. To do so, we compute a pileup of each shape motif's high attribution region relative to the sequence motif over all co-occurrences. We find, looking at the top plot, shape motifs all occur less



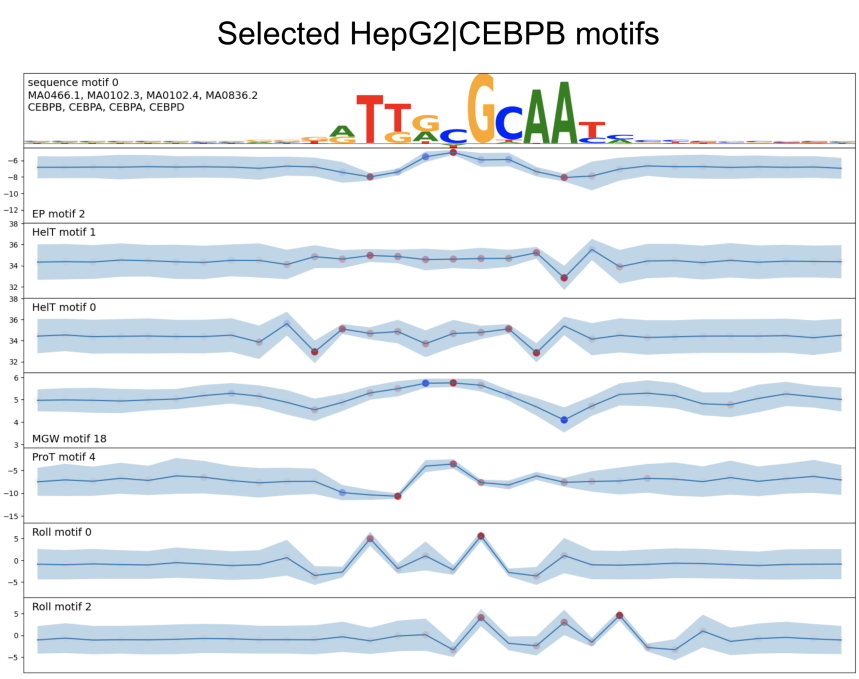
Count in low ratio sequences	Count in high ratio sequences	AUC
725	959	0.975
806	192	0.820
290	758	0.882
565	552	0.834
549	392	0.864
79	856	0.822
98	510	0.773
259	314	0.758

Figure 3.22: Selected predictive HepG2|MafK motifs and occurrence counts in high and low ratio HepG2|MafK-bound sequences.

frequently, normalized for sequence motif occurrences, in negative sequences. In the bottom plot, we see that high ratio sequences feature much higher occurrences of the asymmetric shape motifs, suggesting that they could be compensating for weaker sequence motifs. On the other hand, occurrences of the palindromic shape motifs are similarly or more frequent in low ratio sequences. Interestingly, for Roll motif 2, there is a low but noticeable peak on the weak side of the sequence motif for high ratio sequences, suggesting that the weak side of a sequence motif can be compensated for by the proper shape, and that this manifests as a higher shape ratio sequence.

3.4 Discussion

DeepShape is a highly accurate deep predictor of TF binding (∼0.945 AUROC averaged over 6 replicates, 690 TF targets) taking both sequence and shape features as input. Though shape features may not significantly improve performance, they are still utilized in meaningful ways, as exhibited by consistent performance drops upon shape feature permutation. Examining targets with particularly high shape attribution or performance drops following shape permutation (e.g., HepG2|MafK) can identify cases in which shape information is particularly relevant in TF binding.



Count in low ratio sequences	Count in high ratio sequences	AUC
846	166	0.907
284	93	0.654
178	116	0.666
116	135	0.657
404	53	0.614
319	98	0.643
175	116	0.667
483	73	0.642

Figure 3.23: Selected predictive HepG2|CEBPB motifs and occurrence counts in high and low ratio HepG2|CEBPB-bound sequences.

We find that extreme shape feature values in a few binding site positions are generally how shape features contribute to the model's binding predictions. This points to DNA shape effecting binding in a different manner than DNA sequence. The increased binding affinity of shape motif–sequence motif co-occurrences as compared to sequence motif-only occurrences corroborates that DNA shape modulates TF binding preferences.

Nonetheless, we find that sequence motifs are the main driver of TF binding in our experiment comparing shape and sequence motifs for TF binding prediction. Also, the total shape attribution of most TFs being so close to 0.27 of the total sequence attribution suggests that for most TFs, the model utilizes sequence and shape features in redundant ways, with sequence attribution being higher suggesting higher signal density in sequence features.

Comparing high shape-to-sequence attribution ratio bound sequences to low ratio ones for HepG2|MafK and HepG2|CEBPB, we observe an interesting discrepancy: active regions are more prevalent in low ratio sequences for HepG2|MafK, but more prevalent in high ratio sequences for HepG2|CEBPB. The behavior for HepG2|MafK seems to be explained by the fact that MafK homodimers are repressors due to the lack of an activation domain [56], and based on looking at HepG2|MafK sequence motifs 0 and 1, high ratio sequences appear to correspond to homodimeric sites, while low ratio sequences appear to correspond to AP-1 heterodimers. We did not find a similar explanation for HepG2|CEBPB, though we observed

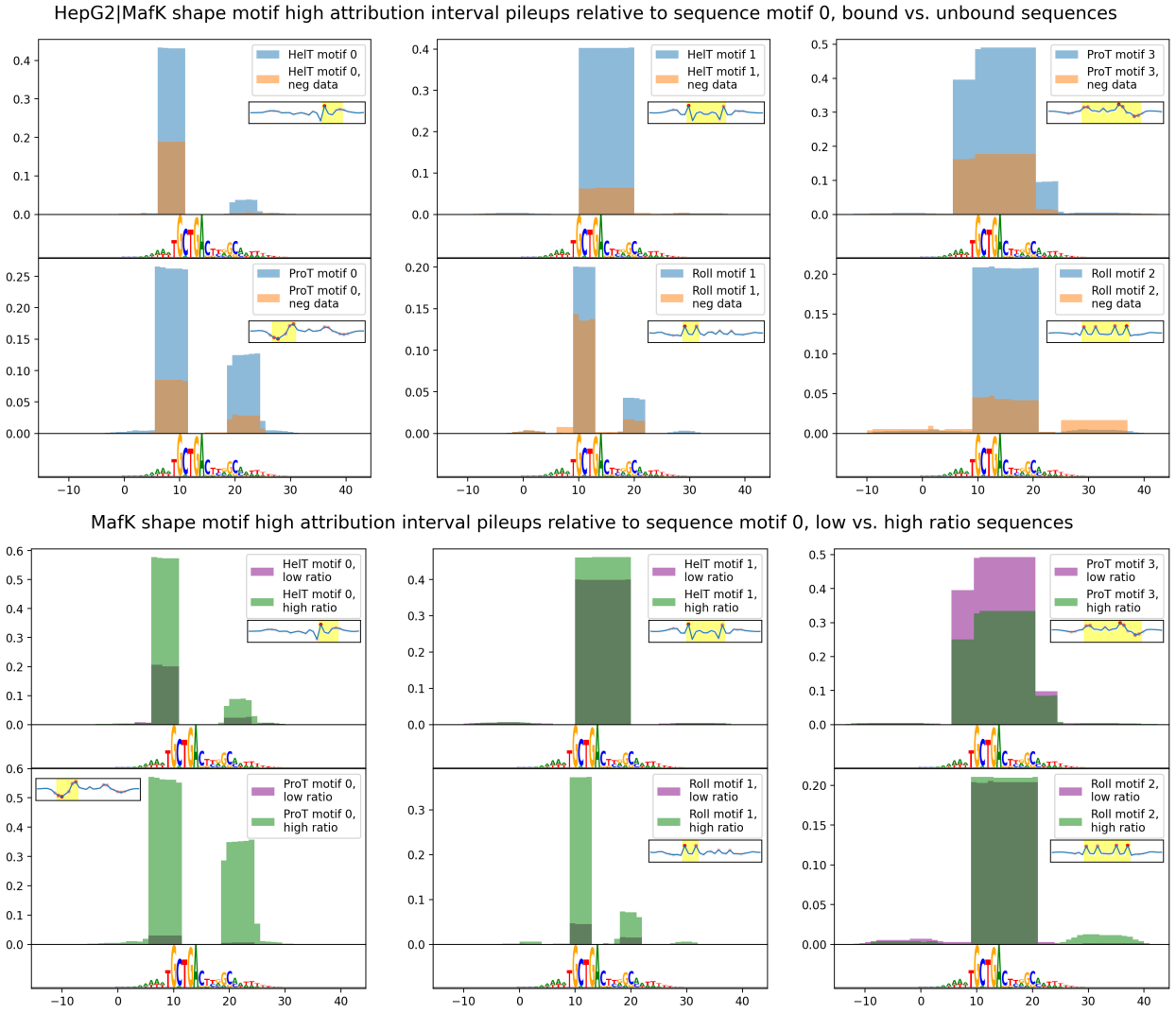


Figure 3.24: Pileups of HepG2|MafK shape motif occurrences (specifically, the high attribution portion of the shape motif, which is highlighted in the insets), relative to sequence motif 0, when they co-occur. The y axis is shape motif occurrence count divided by sequence motif occurrence count. The sequence motif is visualized under each pileup to clearly illustrate what part of the sequence motif each shape motif co-occurs with. Top: positive (bound) sequences vs. negative (unbound) sequences, bottom: low shape-to-sequence attribution ratio positive sequences vs. high ratio positive sequences.

that high ratio HepG2|CEBPB binding sites were more palindromic in sequence and shape, and that low ratio sites often had ATGA for the initial part of the motif instead of ATTG.

This work shows that by employing interpretation methods such as DeepLIFT and TF-

MoDISco, input shape features can be used to understand the effect of DNA shape on TF binding. Overall, we find that shape plays a secondary role to sequence, but highlights interesting discrepancies in types of binding sites. Still though, the role of shape is difficult to cleanly disentangle from that of sequence, which may be inevitable due to redundancy in the information provided by these features.

Chapter 4

Benchmarking Gene Regulatory Networks

4.1 Background

Overview of gene regulatory networks

All of an individual's cells contain the same DNA sequence, yet they vary dramatically in appearance and function. Stomach cells resist acid and absorb nutrients, bone cells form strong, rigid structures, and brain cells conduct electrical impulses. Variation in function despite the same genetic code is possible because in different cell types, genes can be expressed in varying amounts. In a given cell type, genes that are relevant to the function of the cell type are significantly expressed, while genes that are only relevant in other cell types are weakly or not expressed. Gene expression is controlled by regulators, which are biological entities that have the ability to influence gene expression through a variety of mechanisms. The resulting networks of genes, and the regulators that control them, are termed gene regulatory networks.

To be more precise, the expression level of a gene refers to the amount of protein product that is being produced from the gene. According to the fundamental law of molecular biology known as the central dogma, a gene is first *transcribed* into messenger RNA (mRNA) by RNA polymerase, and then the messenger RNA is *translated* into protein by a ribosome (Figure 4.1). The gene itself is merely a section of DNA that serves as a blueprint for a protein; proteins are the primary agents that enact cellular functions. One would ideally quantify the expression of a gene as the amount of the protein it codes for that is present in the cell. However, often, gene expression is instead quantified as the amount of mRNA transcribed from the gene present in the cell, because mRNA can be measured at a much more massive scale than protein. The amount of mRNA is an imperfect proxy for the amount of protein, because mRNA still needs to be translated, and different mRNAs have different translation efficiencies. Nonetheless, based on the current state of technology, mRNA-based assays are the main tool for generating large gene expression datasets.

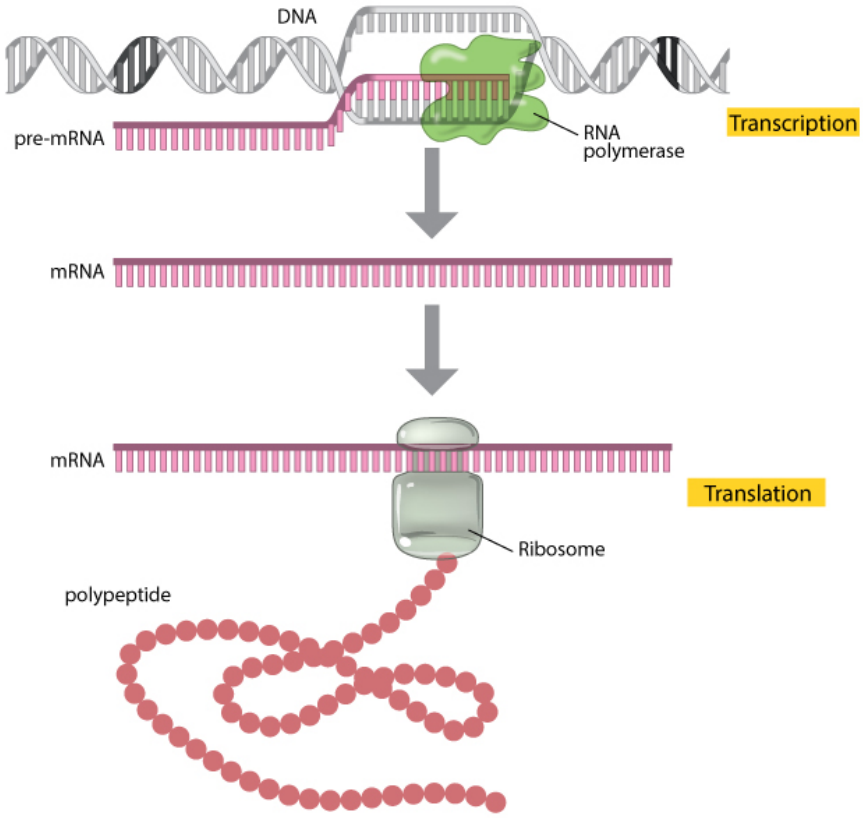


Figure 4.1: The central dogma of molecular biology. Taken from [23].

Transcription factors bind to DNA to control the rate of transcription of their target genes. Certain transcription factors form the pre-initiation complex that is essential for stabilizing RNA polymerase and unwinding DNA so that it can be transcribed (recall Figure 3.1). Other transcription factors can interact with this complex through protein-protein interactions [112] to increase transcription. In eukaryotes, DNA is packaged into nucleosomes, which are coils of DNA wrapped around complexes of histones. DNA that is tightly packed into nucleosomes (heterochromatin) is inaccessible to the transcriptional machinery. Transcription factors can activate transcription by displacing nucleosomes, releasing the DNA and allowing it to be transcribed. Transcription factors can also recruit chromatin modifiers that bestow activating marks on chromatin, such as histone acetylation and histone phosphorylation [79]. In addition to activating roles, transcription factors can serve as repressors, by interfering with RNA polymerase during transcription initiation [97], blocking activators by binding to their binding sites, and recruiting repressive chromatin modifications like histone deacetylation and histone methylation [120]. To complicate matters further, a transcription factor can act as an activator in certain contexts and as a repressor in others

[22].

Transcription factors (TFs) are themselves proteins that are coded for by a gene. Thus, a TF can and often does regulate other TFs, even itself. TFs that serve crucial roles in the maintenance or progression of a particular cell type are called *master regulators* and often regulate a high number of target genes. For example, GATA-1 is considered a master regulator of red blood cell development [36]. Because TFs regulate transcription by binding to DNA, which is detectable through assays and motif scans, and expression assays measure the presence of TFs as well, the regulators considered in current gene regulatory network inference methods are restricted to TFs.

More formally, a gene regulatory network is a graph where each node represents a gene that may or may not code for a TF, and an edge is placed between TF A and gene B if and only if A regulates the expression of B. In that case, we say that B is a target of A. Conceptually, this graph is inherently directed, with the edge pointing from A to B, and current methods strive to infer directionality based on binding information. Past methods based on correlation metrics could not infer directionality between two genes that are both TFs. One may also associate a sign and a magnitude to each edge, denoting the polarity and strength of the relationship respectively.

Networks depend on cellular context, as the genes involved in blood cell differentiation differ from those that underlie neuronal differentiation. We desire to infer these networks because they provide a picture of the regulatory processes for a particular cellular setting. If the network reveals a TF to target many genes, it is implicated as a key regulator. This can lead to a better understanding of the cellular setting, especially if that TF was not known to be a key regulator before. A gene regulatory network can also be used to simulate the effect of a change in transcription factor expression levels, by propagating the changes along the network edges to determine the resulting gene expression profile. Kamimoto et al. [54] pioneered this approach to recover the cell-fate-effects of perturbing transcription factors in hematopoiesis and zebrafish embryonic development.

Gene regulatory network (GRN) inference has broad and highly impactful applications. Cancer occurs when cells mutate and divide uncontrollably, developing ways of bypassing mechanisms that kill aberrant cells. Analyzing the gene regulatory networks under which cancer develops can pinpoint the cause of cancer, leading to improved treatment and prevention. Alternatively, analyzing altered, cancerous regulatory networks can similarly unveil how the cancer survives and develops to more advanced states, leading to better treatments for slowing cancer progression. Gene regulatory networks can be similarly applied to other diseases whose causes are not well understood. Another application is organoids, which are synthetic tissues grown to resemble a particular tissue for the purpose of studying disease or replacing tissue. Knowing how transcription factors drive differentiation into a particular tissue is directly addressed by GRNs and can improve generation of these tissues. Finally, GRNs could aid assisted reproduction. In *in vitro fertilization*, a key problem is constructing a cellular environment that will produce healthy egg cells, which calls for knowing which transcription factors drive egg cell differentiation [119]. Furthermore, *in vitro gametogenesis* (IVG), the production of gametes via reprogramming of a somatic cell to a pluripotent or

stem-like state, followed by differentiation into a functional gamete, has been achieved successfully in mice [82]. IVG is hotly being pursued in both humans and animals, and GRNs have the potential to increase its efficiency.

Challenges in GRN inference

While promising, GRN inference is a challenging task. In 2012, the 5th Dialogue on Reverse Engineering Assessment and Methods (DREAM5) tested 35 different GRN inference methods on recovering known TF-gene interactions in a synthetic network, E. coli, and yeast [73]. Each of the 35 methods got about 2.5% AUPRC on recovering the yeast network, not much better than random, highlighting the difficulty of inferring true regulatory interactions in eukaryotic organisms from expression data alone. The poor performance in this early era of GRN inference mainly had to do with the limited amount of data used to infer the networks and the lack of integration of TF binding information. Nowadays, single-cell RNA-seq (scRNA-seq) provides thousands of expression profiles per experiment, making for much richer expression datasets, and current GRN inference methods also integrate accessibility data to infer causality.

However, there are still fundamental challenges. Current methods lack time resolution in accounting for the effect of TF expression on target gene expression. To an extent, this is captured by modeling pseudotime, but such methods still do not explicitly model TF binding, transcription, translation, and protein and mRNA decay times. Furthermore, a host of regulatory biology is ignored by only focusing on transcription factors. The effects of cell-cell communication by signal transduction [7], small RNAs [87] and post-translational modifications are unaccounted for by only focusing on TF regulators. Also, the inherent noisiness of biological data may pose a challenge.

Finally, GRN inference can be ill-defined and difficult to evaluate. The meaning of "regulates" is difficult to pin down and can include recruiting transcriptional machinery, making regulatory regions accessible, recruiting activating or repressing marks, or blocking other factors from binding. Designing an assay to verify all of these TF-gene interactions so as to obtain a ground truth network is difficult. It is also unclear, mathematically, how to combine a chain of direct effects to obtain a net indirect effect. Multiplying edge weights and summing over possible paths is done to compute the total indirect effect of A on B [33], but it is hard to tell if this is valid. Also, intermediate gene expression changes may change the network topology itself, rendering its reuse at different steps of the paths invalid. Methods may also neglect regulation by theft, in which a TF regulates a gene by stealing its regulators. For example, in fibroblast reprogramming, OCT4 and SOX2 make new regions accessible, which draw AP-1 to themselves, stealing AP-1 from regions near fibroblast genes [83].

Review of GRN inference approaches

Correlation methods One approach is to compute the correlation between the expression of each pair of genes, using some metric. A TF is inferred to regulate a gene if the expression levels of the TF and the gene are highly correlated across samples. Metrics used include pearson correlation, spearman correlation, mutual information, and partial correlation [67, 16, 31, 74, 61]. This type of approach has the advantage of being simple, but is blind to which gene regulates which if both are TFs due to metrics being symmetric, and may be inaccurate because correlations typically do not account for the effect of other variables. Thus, a TF that has a subtle but important effect on a gene may have a low correlation and be missed. Some correlation methods employ techniques to address these issues, such as employing the data processing inequality to prune away mutual information-based edges that represent indirect regulations [74], and using partial correlation from the inverse covariance matrix [61], which for Gaussians is nonzero if and only if the variables are dependent conditioned on all other variables. This tries to highlight interactions whose effects cannot be explained through other interactions.

Regression methods Another approach is to learn a predictor that predicts gene expression from the TF expression, and to use TF-gene coefficients or importance scores as the edge weights. Commonly used models include linear regression with L_1 regularization [111, 43, 68], random forests [49], and gradient boosted trees [80]. As a given gene in a given cell type only has a few TF regulators out of a large pool of potential TFs, this is a sparse regression problem, which is why the L_1 penalty is used. The L_1 penalty is a convex penalty with sparsity-inducing properties. However, methods based on nonconvex L_0 and $L_{\frac{1}{2}}$ penalties have also shown promise [91]. The linear regression problem can be formulated as follows:

$$Y = XW + \varepsilon,$$

where $Y \in \mathbb{R}^{m \times n}$ is a matrix containing gene expression values for n genes over m samples, $X \in \mathbb{R}^{m \times p}$ is a matrix containing gene expression values for p TFs, $W \in \mathbb{R}^{n \times p}$ is a learned matrix of coefficients that describe the effect of the expression of each TF on the expression of each gene, and $\varepsilon \in \mathbb{R}^{m \times n}$ is the error matrix in this model. Letting $M_{\cdot,j}$ denote the jth column of a matrix and assuming an L_1 penalty, this leads to the objective

$$\min_{W_{\cdot,j} \in \mathbb{R}^p} \|Y_{\cdot,j} - XW_{\cdot,j}\|_2^2 + \lambda \|W_{\cdot,j}\|_1$$

for each gene $j = 1, 2, \ldots, n$. If using a random forest or gradient boosted trees, letting $M_{i,\cdot}$ denote the ith row of a matrix, the objective is instead

$$\min_{f_j} \sum_{i=1}^{n} (Y_{ij} - f_j(X_{i,\cdot}))^2$$

for each $j = 1, 2, \ldots, n$, where f_j is optimized over the corresponding function class (random forest or gradient boosted trees). For assigning a signed weight to each potential interaction

in linear regression models, the coefficient W_{ij} itself can be used. For an unsigned importance score, the absolute value $|W_{ij}|$, or the coefficient p-value can be used. A previous work infers networks at various levels of sparsity and uses the first sparsity level at which the interaction exists as a more sophisticated, but potentially more principled measure [91]. For random forests, the cumulative variance explained by the variable averaged over the trees in the forest is used [49]. One must be careful to ensure importance scores are comparable across different genes. For example, for the random forest measure, the variance of each gene is first normalized to 1.

ODE-based methods Other methods view changes in gene expression from the perspective of differential equations, and derive inference models from them [13, 124]. In practice, the resulting GRN inference algorithms work out to a kind of regression method. For example, Inferelator [13] considers the kinetic equation

$$\tau \frac{\mathrm{d}y}{\mathrm{d}t} = -y + g(\beta^\top z(x))$$

where y represents the expression level of a particular gene, x is a vector containing the expression levels of putative regulators, and z is a vector derived from x. β is a learned vector of coefficients, τ is the time constant, and g is a thresholding function:

$$g(\beta^\top x) = \begin{cases} \beta^\top x & y_{\min} \leq \beta^\top x < y_{\max} \\ y_{\min} & \beta^\top x < y_{\min} \\ y_{\max} & \beta^\top x \geq y_{\max} \end{cases}.$$

Thus, for steady state conditions in which $\frac{\mathrm{d}y}{\mathrm{d}t} = 0$, they obtain the equation

$$y = g(\beta^\top z(x)),$$

and for time series data with time points $t_1, t_2, \ldots, t_T$, they obtain the equation

$$\tau \frac{y_{m+1} - y_m}{t_{m+1} - t_m} + y_m = g(\beta^\top z(x)).$$

They solve these equations using an L_1-regularized linear regression method, LARS, with the labels as the quantities on the left hand side and the predictions as on the right hand side. For the time series data, the time constant τ is determined by alternately minimizing the loss over τ and the regression parameters until convergence. Inferelator 2.0 [72] improves the dynamical modeling aspect using Markov chain Monte Carlo methods, obtaining improved predictive performance.

Single-cell and integrative methods With the advent of single-cell data [123], modern GRN inference methods take the same methodology as previous ones but apply them to

single-cell data, and include an additional step of identifying TF binding sites to disambiguate causality [18, 4, 40, 110, 34, 134, 124, 14]. Various ways of finding binding sites include incorporating TF binding data such as ChIP-seq, incorporating (single-cell) accessibility data, such as (sc)ATAC-seq, and motif scanning. An earlier work showed that incorporating ChIP-seq data into sparse regression models greatly improved recovery of ground truth interactions, but their evaluation is somewhat confounded by overlap between the data defining the ground truth and the data incorporated into their model [91]. SCENIC [4], one of the first of the modern methods mentioned above, applies GENIE3 or GRNBoost2 to scRNA-seq data and additionally runs RcisTarget [4] to filter for direct interactions, which searches for TF binding motifs near gene TSSs. CellOracle [54] incorporates single-cell accessibility data by linking distal enhancers to promoters using Cicero [17]. TFs are linked to these regions by and scanning them for motifs. Then, the TFs linked to the regions that are linked to a gene (the promoter and enhancers linked to the promoter) are used to predict the expression of the gene. Overall, these integrative approaches significantly improve GRN inference accuracy over previous approaches only utilizing gene expression data.

Approaches to GRN evaluation

Unfortunately, there is no broadly agreed-upon benchmark for evaluating different GRN inference methods, though benchmarks like DREAM5 [73] and BEELINE [89] exist. This is a big problem, because without a broadly accepted benchmark, it is difficult to truly know how methods perform relative to each other. With so many different methods out there, each with their own specific implementation, choosing one becomes a paralyzing task. A challenge here is that different methods take in different sets of data; some use single-cell data, some use multi-ome data, some use pseudotime, etc. This makes it hard to compile datasets that a broad range of methods can be evaluated fairly on.

One approach to evaluating inferred GRNs is to compile a set of verified TF-target interactions and compare the ranking returned by each method against this set. For example, datasets of interactions compiled from the literature, like ImmGen [45] and the integrated Stem-cell Molecular interactions Database (iScMiD) [71], exist. Furthermore, TF-target regulations can be obtained through experiments in which a TF perturbation experiments. The list of differentially expressed genes can be considered a broad set of targets, including indirect targets. Direct targets can be identified by intersecting differentially expressed genes with the perturbed TF ChIP-seq binding peaks, and one-level-down indirect targets by finding genes whose TSSs contain a motif of a direct target [90]. However, this approach does not detect distal regulation. Also, the relative expression changes of the differentially expressed genes is not taken into account for evaluation, even though this could be useful for evaluating inferred GRNs.

Another evaluation approach is to test recovery of known biological phenomena. This involves inferring a GRN from a certain cellular context, drawing conclusions from the GRN, and comparing them against true biology. This was done to evaluate Pando in the context of human brain organoids [34], and CellOracle was able to recover known drivers and their

effects in mouse hematopoiesis and zebrafish embryogenesis [54]. This type of evaluation is promising, as it is evaluating GRNs on what they are ultimately likely to be used for, but this is prone to confirmation bias, as nothing prevents a researcher from only presenting model-derived conclusions that agree with what is known. It is important to improve the soundness of this type of evaluation by predefining a set of biological conclusions to be recovered, and testing recovery of these predefined conclusions.

4.2 Recovering reprogramming factors

Overview

Takahashi and Yamanaka [116] discovered that overexpressing the four transcription factors OCT4, SOX2, KLF4, and MYC (OSKM) could reprogram mouse skin cells into pluripotent stem cells that were similar to embryonic stem cells. The induction of human pluripotent stem cells using the same four factors followed soon after [117], as well as using a different set: OCT4, SOX2, NANOG, and LIN28 [132]. These induced pluripotent stem cells (iPSCs) theoretically have the ability to differentiate into any type of cell in the human body, giving them great potential for medical research and use. Other TF combinations capable of iPSC induction have since been discovered, such as SALL4, NANOG, ESRRB, and LIN28 (SNEL) [15], OCT4, SOX2, and KLF4 (OSK) [15], and SOX2, KLF4, and MYC (SKM) [122]. Interestingly, these combinations generally lead to fewer iPSCs than OSKM, but a greater fraction of high quality iPSCs.

One test of GRN inference methods is their ability to recover the effects of these reprogramming factors, given single-cell data containing pluripotent stem cells. This tests whether GRNs are useful for identifying TFs that can achieve a particular cell fate. CellOracle [54] explicitly aims to predict the effect of TF perturbations on cell differentiation, visualizing the overall effect as a vector field on top of the cell population. Thus, here, we apply CellOracle to a single cell atlas of human fibroblasts being reprogrammed to iPSCs [83]. This entails constructing GRNs from the data with CellOracle, running network analysis and perturbation simulation, and analyzing the results in light of the fact that the above combinations of factors can reprogram mature cells to pluripotent stem cells.

Dataset of human fibroblasts reprogrammed with OSKM

We use a dataset containing scRNA-seq and scATAC-seq data of human skin cells (fibroblasts), reprogrammed by overexpressing OSKM via a Sendai virus system [83]. The scRNA-seq data contains gene expression data for 59,378 quality-controlled cells, but to shorten computation time we downsample to 12,000 cells. Similarly, we downsample the scATAC-seq data, which contains accessibility values for 62,599 quality-controlled cells across 530,910 genome regions, to 1,000 cells, as the scATAC-seq data processing step of CellOracle is very time consuming. A UMAP dimensionality reduction of the scRNA-seq cells colored by cell

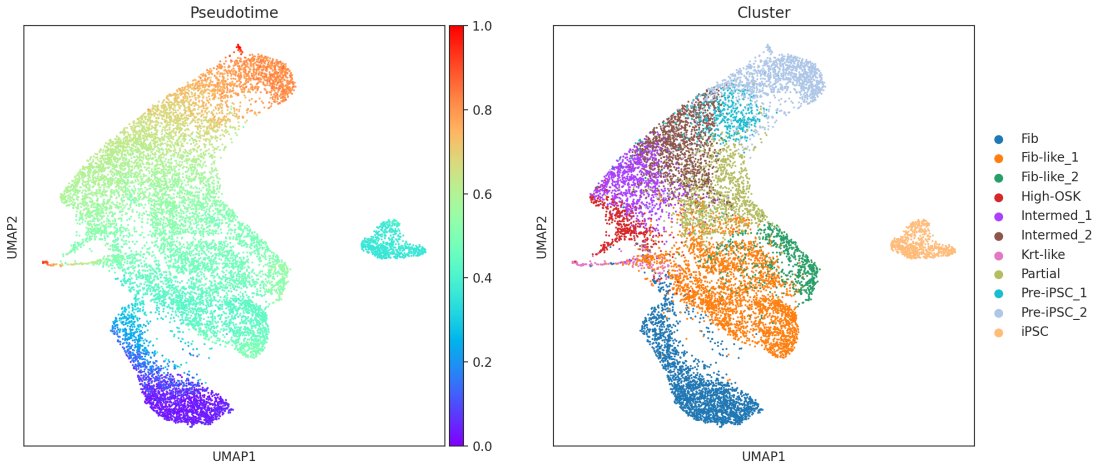


Figure 4.2: Dimension-reduced (UMAP) fibroblast reprogramming single-cell gene expression data (scRNA-seq). Fibroblasts (blue, Fib) travel upward in the visualization during reprogramming, progressing through fibroblast-like, intermediate, and pre-iPSC states. Successfully reprogrammed iPSCs lie in a distinct cluster far from the other cells.

type and pseudotime is shown in Figure 4.2. Pseudotime is an estimate of the position of a cell along its differentiation trajectory, with pseudotime values lying between 0 (initial state) and 1 (final differentiated state). The UMAP coordinates and cell type clusters were provided by the authors in their data, while pseudotime was computed from the scRNA-seq data using diffusion pseudotime [42] through the CellOracle package.

Results of CellOracle

We apply CellOracle [54] to the human fibroblast reprogramming dataset. First, a base GRN is constructed from the scATAC-seq data by linking regions to co-accessible genes with Cicero [88], linking TFs to regions by motif scanning, and then linking TFs to the regions' genes. Then, CellOracle trains a linear model to predict gene expression from the expression of TFs linked to each gene. A different set of linear models is trained for each cluster of cells. The linear regression coefficients are used to further prune the GRN links. Finally, the resulting GRNs are used to simulate the effect of knocking out a TF by setting its expression to 0, and mapping the resulting change in expression to a change in cell identity. Also, the degree of each TF node is used to prioritize major regulators.

We use CellOracle to simulate the effect of knocking out each OSKM factor individually, and the effect of knocking them all out together. The results are shown in Figure 4.3. Since OSKM drives differentiation towards the iPSC-like clusters at the top of the embedding,

we would expect the effect of knocking these factors out to go against this differentiation, represented by downward arrows and purple squares in the visualizations. We can think of a purple region in a TF-KO visualization as a cluster of cells that the TF helps reprogram. Interestingly, we find that each OSKM TF does not uniformly help all cells reprogram. Each has its green regions, in which it may be in fact hindering reprogramming. However, their purple and green regions are complementary, and the effect of the combined knockout (bottom right) is the closest to all purple, which could explain why the combination of all four factors is effective for reprogramming.

Figure 4.3: The pseudotime differentiation field, and the differentiation fields from simulating knocking out each of the OSKM factors individually, and all together. Purple denotes negative inner product between the pseudotime and KO field (knockout goes against differentiation) while green denotes positive inner product (knockout accelerates differentiation).

Next, we expand our scope to all TFs and see how well CellOracle can prioritize OSKM as reprogramming factors. To do so, we calculate the negative perturbation score for the 100 TFs with the highest degree averaged over clusters. The negative perturbation score, which is a functionality already provided by CellOracle, computes the sum of the inner products between KO effect map and pseudotime map over all purple regions (negative inner product regions) in the KO effect map. A bigger (more negative) negative perturbation score indicates

a higher contribution to the differentiation of fibroblasts to iPSCs. The TFs with the top 30 negative perturbation scores (we negate them so that they are positive, for a more intuitive score) are shown in Figure 4.4.

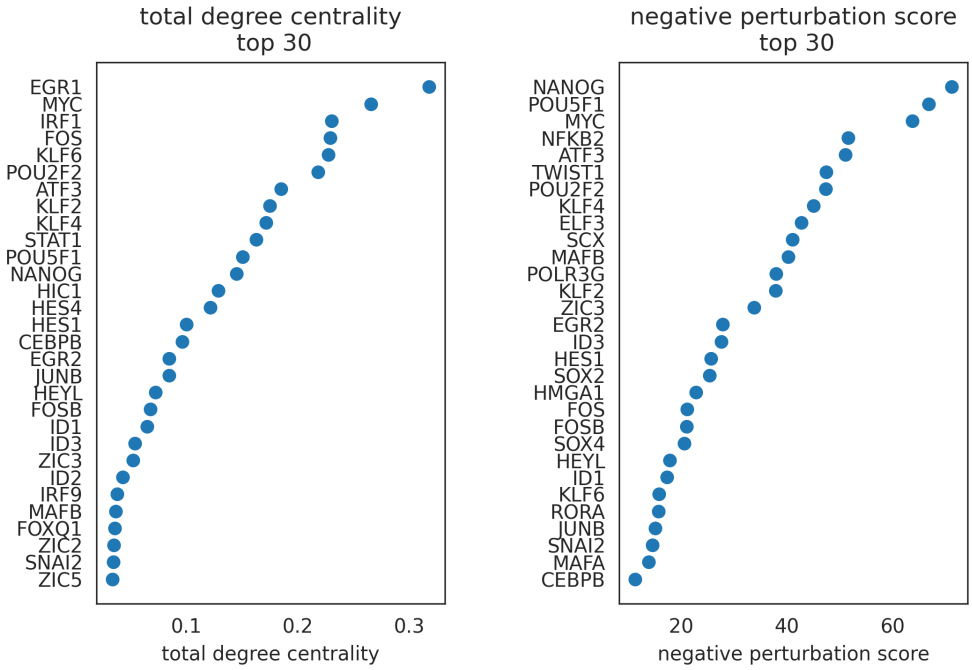


Figure 4.4: Left: highest degree TFs, averaged over clusters (both in-degree, number of its regulators and out-degree, number of its targets, are counted). Right: highest negative perturbation score TFs.

We find that OSKM are all included in the top 30 negative perturbation score TFs (OCT4 is aliased as POU5F1). They are ranked 2, 18, 8, and 3, respectively. The negative perturbation score seems to better measure reprogramming potential than total degree, for which OKM are ranked 11, N/A, 9, and 2, respectively. However, it would be difficult to recover the specific combination of OSKM just from the list. This points to a limitation of using GRNs to prioritize combinations of factors for reprogramming. Biological domain knowledge is needed, though the GRN-based approach could potentially be improved by ranking perturbations of combinations of TFs, and improving modeling such that the effect of complex, combinatorial perturbations can be adequately captured.

Results of SCENIC+

Next we run SCENIC+ [14] on the fibroblast reprogramming dataset. Since SCENIC+ requires multiome data and the fibroblast reprogramming scRNA-seq and scATAC-seq datasets

are separate, we use the provided Harmony integration embedding [64] to assign each scATAC-seq cell the expression data of its nearest scRNA-seq cell in the embedding. Briefly, SCENIC+ computes topics from the accessibility matrix using LDA, and binarizes the topics to generate region sets. Also, SCENIC+ computes differentially accessible regions for clusters in the data to generate additional region sets. Then, a comprehensive set of regions is subjected to a motif scan, wherein the regions are ranked by motif score for each motif. Then this ranking database is used to find enriched motifs for each region set; the TFs of the enriched motifs are assigned to the corresponding regions. Regions are linked to genes by training a gradient boosting model to predict gene expression from region accessibility and taking the most important regions for each gene. A TF-to-gene expression gradient boosting model where all TFs are used is also trained. To assemble the eRegulons (a TF and its TF-region-gene triplets), each combination of TF-gene and region-gene relationship signs is considered, namely, positive-positive, positive-negative, negative-positive, and negative-negative, determined by the correlation coefficient between TF expression or region accessibility and gene expression. For each TF and sign combination, TF-region-gene triplets such that the TF-gene and region-gene pairs satisfy the sign combination are considered. The resultant target genes are pruned based on ranking them by the TF-to-gene gradient boosting important scores. eRegulons with less than 10 target genes are discarded.

We sort the eRegulons by number of target genes in Figure 4.5. Direct refers to eRegulons based on motifs that are directly attributed to the TF through a ChIP-seq experiment in the same species. Extended refers to motifs annotated to the TF through orthology or motif similarity. There were 102 direct eRegulons and 58 extended eRegulons. In addition, we combine the genes in all the eRegulons associated to a TF, and sort TFs by total target gene count in the neighboring plot. We observe that OSKM are perhaps better prioritized by degree here (ranks 5, 12, 9, 10) than in the CellOracle networks (ranks 11, N/A, 9, and 2).

Next, we use SCENIC+ to simulate the effect of knocking out OSKM. SCENIC+ follows the same procedure as CellOracle, except instead of a linear model, a gradient boosting model is used to predict gene expression from the expression of TFs regulating the gene. Furthermore, CellOracle repeatedly applies their linear models to a *difference* vector to update it, in which the entry corresponding to the knocked-out TF is set to 0 at each iteration. On the other hand, SCENIC+ repeatedly applies their GB models to the *expression* vector with the entry corresponding to the knocked-out TF set to 0 at the beginning of the first iteration. The SCENIC+ final difference vector is taken as the difference between the final expression vector and the initial expression vector.

The results of SCENIC+ OSKM-KO simulation are shown in Figure 4.6. Unlike the effects predicted by CellOracle, the effects predicted by SCENIC+ for each factor are more similar to each other, though they still are all predicted to inhibit reprogramming upon knockout. This may stem from CellOracle's application of the TF perturbation at each iteration of simulation, vs. SCENIC+ only applying it at the beginning. Biologically, applying it at every iteration corresponds to permanent or long-lasting perturbation, while applying it only once corresponds to an instantaneous perturbation.

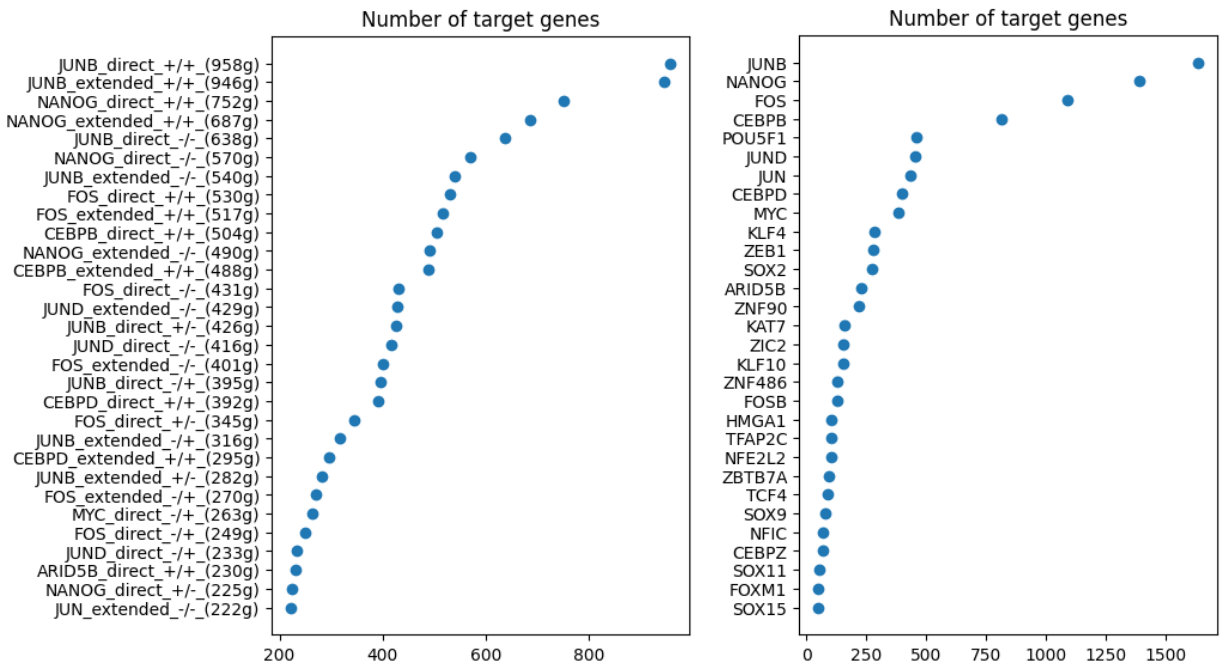


Figure 4.5: Top 30 eRegulons and TFs found by SCENIC+ on the fibroblast reprogramming dataset sorted by number of target genes.

Discussion

Through application to a fibroblast reprogramming dataset, we show that single-cell-based GRN inference methods like CellOracle and SCENIC+ can identify important gene regulators and their predicted effects on cell identity. There were differences in the network structures inferred by the two methods, with the overall highest degree node for CellOracle, EGR1, not even appearing in the top 30 for SCENIC+. Both methods still were able to recover the OSKM reprogramming factors when prioritizing TFs by degree, and negative perturbation score (predicted amount of inhibition of reprogramming) for CellOracle. SCENIC+'s use of topic modeling for accessibility may capture broader patterns of regulation than CellOracle's linking of enhancers to promoters based on co-accessibility. Furthermore, SCENIC+ utilizes the accessibility to predict gene expression and form links in the network, while CellOracle only uses TF expression, granted the TFs are selected based on accessibility. SCENIC+ learns a single network encompassing regulatory patterns across the cell population, while CellOracle learns a different network for each cluster. In terms of simulation, CellOracle predicted notable differences between the roles in reprogramming played by different factors, while SCENIC+ did not reveal as large differences. This is likely due to the different network structures learned by each method.

To better benchmark GRN inference methods for simulating effects on cell identity,

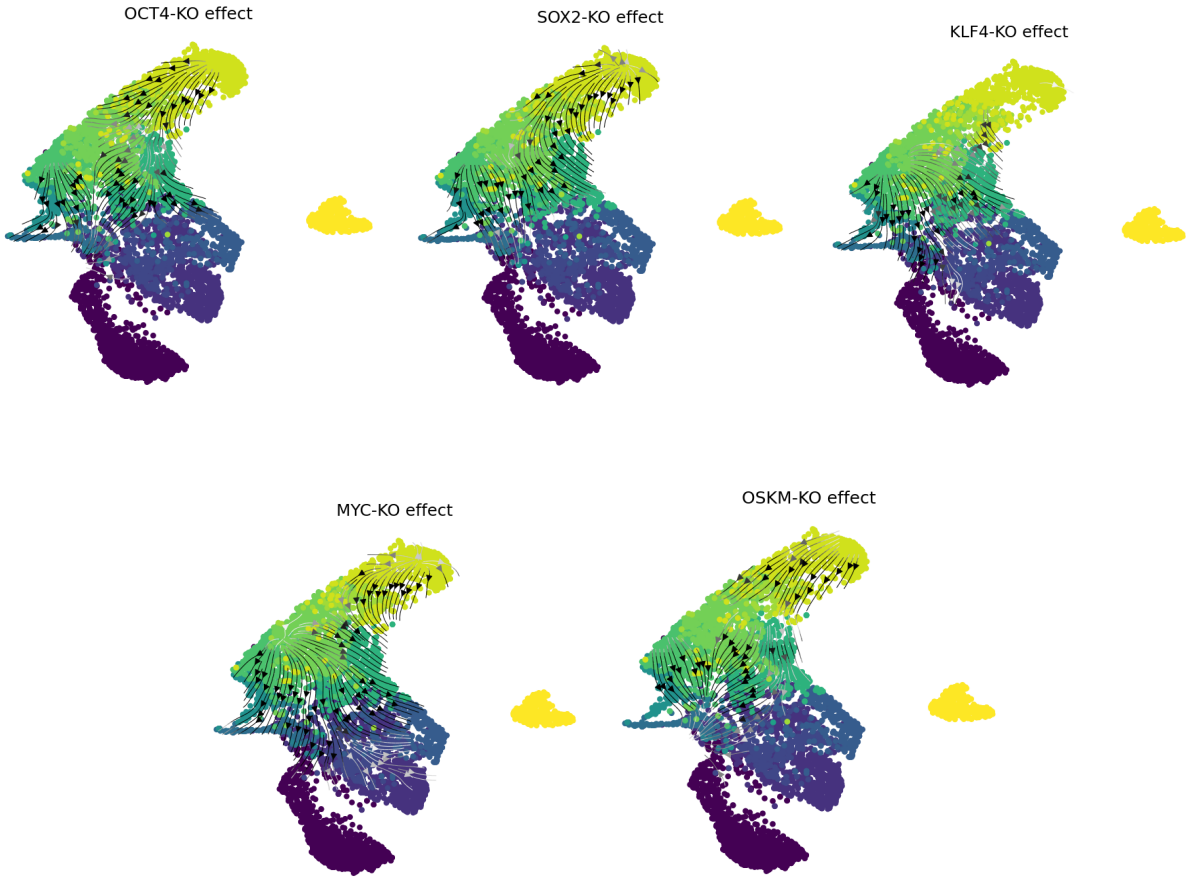


Figure 4.6: SCENIC+ simulated effects of OSKM knockout.

datasets in which the precise trajectory in cell-embedding space induced by a particular perturbation has been determined experimentally are needed, so that the accuracy of different methods can be quantitatively compared. Testing a greater volume of methods and datasets would also help understand the overall utility of GRN inference methods, though potentially challenging due to differences in datasets and required inputs for different methods. For further investigating cellular reprogramming, applying GRN inference methods to a dataset containing naturally differentiating pluripotent stem cells would show whether the role of OSKM in reprogramming could be recovered by observing "natural" data.

Bibliography

[1] Odd O. Aalen, Ørnulf Borgan, and S. Gjessing. *Survival and event history analysis: a process point of view*. Statistics for biology and health. New York, NY: Springer, 2008. ISBN: 9780387202877.

[2] Namiko Abe et al. "Deconvolving the recognition of DNA shape from sequence". en. In: *Cell* 161.2 (Apr. 2015), p. 307. DOI: 10.1016/j.cell.2015.02.008. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4422406/.

[3] Ariel Afek et al. "DNA mismatches reveal conformational penalties in protein–DNA recognition". en. In: *Nature* 587.7833 (Nov. 2020), pp. 291–296. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-020-2843-2. URL: https://www.nature.com/articles/s41586-020-2843-2.

[4] Sara Aibar et al. "SCENIC: single-cell regulatory network inference and clustering". en. In: *Nature Methods* 14.11 (Nov. 2017), pp. 1083–1086. ISSN: 1548-7105. DOI: 10.1038/nmeth.4463. URL: https://www.nature.com/articles/nmeth.4463.

[5] Bruce Alberts et al. "DNA-Binding Motifs in Gene Regulatory Proteins". en. In: *Molecular Biology of the Cell. 4th edition*. Garland Science, 2002. URL: https://www.ncbi.nlm.nih.gov/books/NBK26806/.

[6] Marco Ancona et al. "Towards better understanding of gradient-based attribution methods". In: (Mar. 2018). DOI: 10.48550/arXiv.1711.06104. URL: http://arxiv.org/abs/1711.06104.

[7] Erick Armingol et al. "Deciphering cell–cell interactions and communication from gene expression". en. In: *Nature Reviews Genetics* 22.2 (Feb. 2021), pp. 71–88. ISSN: 1471-0064. DOI: 10.1038/s41576-020-00292-x. URL: https://www.nature.com/articles/s41576-020-00292-x.

[8] Žiga Avsec et al. "Effective gene expression prediction from sequence by integrating long-range interactions". en. In: *Nature Methods* 18.10 (Oct. 2021), pp. 1196–1203. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-021-01252-x. URL: https://www.nature.com/articles/s41592-021-01252-x.

[9] M. J. Benton. *The history of life: a very short introduction*. Very short introductions. Great Clarendon Street, Oxford: Oxford University Press, 2008. ISBN: 9780199226320.

[10] Michael F. Berger and Martha L. Bulyk. "Protein Binding Microarrays (PBMs) for the Rapid, High-Throughput Characterization of the Sequence Specificities of DNA Binding Proteins". In: *Methods in molecular biology (Clifton, N.J.)* 338 (2006), pp. 245–260. ISSN: 1064-3745. DOI: 10.1385/1-59745-097-9:245. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2690637/.

[11] Bradley E Bernstein et al. "The NIH Roadmap Epigenomics Mapping Consortium". In: *Nature biotechnology* 28.10 (Oct. 2010), pp. 1045–1048. ISSN: 1087-0156. DOI: 10.1038/nbt1010-1045. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3607281/.

[12] Eric P. Bishop et al. "A Map of Minor Groove Shape and Electrostatic Potential from Hydroxyl Radical Cleavage Patterns of DNA". en. In: *ACS chemical biology* 6.12 (Dec. 2011), p. 1314. DOI: 10.1021/cb200155t. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3241897/.

[13] Richard Bonneau et al. "The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo". In: *Genome Biology* 7.5 (May 2006), R36. ISSN: 1474-760X. DOI: 10.1186/gb-2006-7-5-r36. URL: https://doi.org/10.1186/gb-2006-7-5-r36.

[14] Carmen Bravo González-Blas et al. "SCENIC+: single-cell multiomic inference of enhancers and gene regulatory networks". en. In: *Nature Methods* 20.9 (Sept. 2023), pp. 1355–1367. ISSN: 1548-7105. DOI: 10.1038/s41592-023-01938-4. URL: https://www.nature.com/articles/s41592-023-01938-4.

[15] Yosef Buganim et al. "The Developmental Potential of iPSCs Is Greatly Influenced by Reprogramming Factor Selection". In: *Cell stem cell* 15.3 (Sept. 2014), pp. 295–309. ISSN: 1934-5909. DOI: 10.1016/j.stem.2014.07.003. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4170792/.

[16] A. J. Butte and I. S. Kohane. "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements". eng. In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* (2000), pp. 418–429. ISSN: 2335-6928. DOI: 10.1142/9789814447331_0040.

[17] Junyue Cao et al. "The single-cell transcriptional landscape of mammalian organogenesis". en. In: *Nature* 566.7745 (Feb. 2019), pp. 496–502. ISSN: 1476-4687. DOI: 10.1038/s41586-019-0969-x. URL: https://www.nature.com/articles/s41586-019-0969-x.

[18] Thalia E. Chan, Michael P. H. Stumpf, and Ann C. Babtie. "Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures". eng. In: *Cell Systems* 5.3 (Sept. 2017), 251–267.e3. ISSN: 2405-4712. DOI: 10.1016/j.cels.2017.08.014.

[19] Kathleen M. Chen et al. "Selene: a PyTorch-based deep learning library for sequence data". en. In: *Nature Methods* 16.4 (Apr. 2019), pp. 315–318. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0360-8. URL: https://www.nature.com/articles/s41592-019-0360-8.

[20] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: arXiv:2002.05709 (June 2020). arXiv:2002.05709 [cs, stat]. DOI: 10.48550/arXiv.2002.05709. URL: http://arxiv.org/abs/2002.05709.

[21] Travers Ching, Xun Zhu, and Lana X. Garmire. "Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data". en. In: *PLOS Computational Biology* 14.4 (Apr. 2018), e1006076. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006076. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006076.

[22] Maria Ciofani et al. "A validated regulatory network for Th17 cell specification". eng. In: *Cell* 151.2 (Oct. 2012), pp. 289–303. ISSN: 1097-4172. DOI: 10.1016/j.cell.2012.09.016.

[23] Suzanne Clancy and William Brown. "Translation: DNA to mRNA to Protein". In: *Nature Education* 1.1 (2008), p. 101.

[24] The ENCODE Project Consortium. "An integrated encyclopedia of DNA elements in the human genome". en. In: *Nature* 489.7414 (Sept. 2012), pp. 57–74. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature11247. URL: https://www.nature.com/articles/nature11247.

[25] D. R. Cox. "Regression Models and Life-Tables". en. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 34.2 (Jan. 1972), pp. 187–202. ISSN: 1369-7412, 1467-9868. DOI: 10.1111/j.2517-6161.1972.tb00899.x. URL: https://academic.oup.com/jrsssb/article/34/2/187/7027194.

[26] Chiara D'Antonio et al. "Bone and brain metastasis in lung cancer: recent advances in therapeutic strategies". In: *Therapeutic Advances in Medical Oncology* 6.3 (May 2014), pp. 101–114. ISSN: 1758-8340. DOI: 10.1177/1758834014521110. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3987652/.

[27] Gaohong Dong et al. "The inverse-probability-of-censoring weighting (IPCW) adjusted win ratio statistic: an unbiased estimator in the presence of right censoring". In: *Journal of biopharmaceutical statistics* 30.5 (Sept. 2020), pp. 882–899. ISSN: 1054-3406. DOI: 10.1080/10543406.2020.1757692. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7538385/.

[28] Jason Ernst and Manolis Kellis. "ChromHMM: automating chromatin-state discovery and characterization". en. In: *Nature Methods* 9.3 (Mar. 2012), pp. 215–216. ISSN: 1548-7105. DOI: 10.1038/nmeth.1906. URL: https://www.nature.com/articles/nmeth.1906.

[29] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". en. In: *Nature* 542.7639 (Feb. 2017), pp. 115–118. ISSN: 1476-4687. DOI: 10.1038/nature21056. URL: https://www.nature.com/articles/nature21056.

[30] Mark Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". en. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-009-0275-4. URL: http://link.springer.com/10.1007/s11263-009-0275-4.

[31] Jeremiah J. Faith et al. "Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles". en. In: *PLOS Biology* 5.1 (Jan. 2007), e8. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.0050008. URL: https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0050008.

[32] David Faraggi and Richard Simon. "A neural network model for survival data". en. In: *Statistics in Medicine* 14.1 (Jan. 1995), pp. 73–82. ISSN: 0277-6715, 1097-0258. DOI: 10.1002/sim.4780140108. URL: https://onlinelibrary.wiley.com/doi/10.1002/sim.4780140108.

[33] Soheil Feizi et al. "Network deconvolution as a general method to distinguish direct dependencies in networks". en. In: *Nature Biotechnology* 31.8 (Aug. 2013), pp. 726–733. ISSN: 1546-1696. DOI: 10.1038/nbt.2635. URL: https://www.nature.com/articles/nbt.2635.

[34] Jonas Simon Fleck et al. "Inferring and perturbing cell fate regulomes in human brain organoids". en. In: *Nature* 621.7978 (Sept. 2023), pp. 365–372. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05279-8. URL: https://www.nature.com/articles/s41586-022-05279-8.

[35] Nitsan Fourier et al. "MafK Mediates Chromatin Remodeling to Silence IRF8 Expression in Non-immune Cells in a Cell Type-SpecificManner". eng. In: *Journal of Molecular Biology* 432.16 (July 2020), pp. 4544–4560. ISSN: 1089-8638. DOI: 10.1016/j.jmb.2020.06.005.

[36] Y Fujiwara et al. "Arrested development of embryonic red cell precursors in mouse embryos lacking transcription factor GATA-1." In: *Proceedings of the National Academy of Sciences of the United States of America* 93.22 (Oct. 1996), pp. 12355–12358. ISSN: 0027-8424. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC37995/.

[37] Kunihiko Fukushima. "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements". In: *IEEE Transactions on Systems Science and Cybernetics* 5.4 (Oct. 1969), pp. 322–333. ISSN: 2168-2887. DOI: 10.1109/TSSC.1969.300225. URL: https://ieeexplore.ieee.org/document/4082265.

[38] Michael F. Gensheimer and Balasubramanian Narasimhan. "A Scalable Discrete-Time Survival Model for Neural Networks". In: *PeerJ* 7 (Jan. 2019). arXiv:1805.00917 [cs, stat], e6257. ISSN: 2167-8359. DOI: 10.7717/peerj.6257. URL: http://arxiv.org/abs/1805.00917.

[39] Raluca Gordân et al. "Genomic regions flanking E-box binding sites influence DNA binding specificity of bHLH transcription factors through DNA shape". In: *Cell reports* 3.4 (Apr. 2013), pp. 1093–1104. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2013.03.014. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3640701/.

[40] Jeffrey M. Granja et al. "ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis". en. In: *Nature Genetics* 53.3 (Mar. 2021), pp. 403–411. ISSN: 1546-1718. DOI: 10.1038/s41588-021-00790-6. URL: https://www.nature.com/articles/s41588-021-00790-6.

[41] Charles E. Grant, Timothy L. Bailey, and William Stafford Noble. "FIMO: scanning for occurrences of a given motif". eng. In: *Bioinformatics (Oxford, England)* 27.7 (Apr. 2011), pp. 1017–1018. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btr064.

[42] Laleh Haghverdi et al. "Diffusion pseudotime robustly reconstructs lineage branching". en. In: *Nature Methods* 13.10 (Oct. 2016), pp. 845–848. ISSN: 1548-7105. DOI: 10.1038/nmeth.3971. URL: https://www.nature.com/articles/nmeth.3971.

[43] Anne-Claire Haury et al. "TIGRESS: Trustful Inference of Gene REgulation using Stability Selection". In: arXiv:1205.1181 (May 2012). arXiv:1205.1181 [q-bio, stat]. DOI: 10.48550/arXiv.1205.1181. URL: http://arxiv.org/abs/1205.1181.

[44] Qiye He, Jeff Johnston, and Julia Zeitlinger. "ChIP-nexus enables improved detection of in vivo transcription factor binding footprints". en. In: *Nature Biotechnology* 33.4 (Apr. 2015), pp. 395–401. ISSN: 1546-1696. DOI: 10.1038/nbt.3121. URL: https://www.nature.com/articles/nbt.3121.

[45] Tracy S. P. Heng et al. "The Immunological Genome Project: networks of gene expression in immune cells". en. In: *Nature Immunology* 9.10 (Oct. 2008), pp. 1091–1094. ISSN: 1529-2916. DOI: 10.1038/ni1008-1091. URL: https://www.nature.com/articles/ni1008-1091.

[46] Connor A. Horton et al. "Short tandem repeats bind transcription factors to tune eukaryotic gene expression". en. In: *Science* 381.6664 (Sept. 2023), eadd1250. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.add1250. URL: https://www.science.org/doi/10.1126/science.add1250.

[47] Ahmed Hosny et al. "Deep learning for lung cancer prognostication: A retrospective multi-cohort radiomics study". eng. In: *PLoS medicine* 15.11 (Nov. 2018), e1002711. ISSN: 1549-1676. DOI: 10.1371/journal.pmed.1002711.

[48] Le Hou et al. "Patch-Based Convolutional Neural Network for Whole Slide Tissue Image Classification". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 2424–2433. DOI: 10.1109/CVPR.2016.266. URL: https://ieeexplore.ieee.org/document/7780635/;jsessionid=D5D56F1F9600730D70B5C22F0434A42A.

[49] Vân Anh Huynh-Thu et al. "Inferring Regulatory Networks from Expression Data Using Tree-Based Methods". en. In: *PLOS ONE* 5.9 (Sept. 2010), e12776. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0012776. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0012776.

[50] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: arXiv:1502.03167 (Mar. 2015). arXiv:1502.03167 [cs]. DOI: 10.48550/arXiv.1502.03167. URL: http://arxiv.org/abs/1502.03167.

[51] Pavel Izmailov et al. "Averaging Weights Leads to Wider Optima and Better Generalization". In: arXiv:1803.05407 (Feb. 2019). arXiv:1803.05407 [cs, stat]. DOI: 10.48550/arXiv.1803.05407. URL: http://arxiv.org/abs/1803.05407.

[52] Li J et al. "Expanding the repertoire of DNA shape features for genome-scale studies of transcription factor binding". en. In: *Nucleic acids research* 45.22 (Dec. 2017). ISSN: 1362-4962. DOI: 10.1093/nar/gkx1145. URL: https://pubmed.ncbi.nlm.nih.gov/29165643/.

[53] Rohit Joshi et al. "Functional Specificity of a Hox Protein Mediated by the Recognition of Minor Groove Structure". en. In: *Cell* 131.3 (Nov. 2007), pp. 530–543. ISSN: 00928674. DOI: 10.1016/j.cell.2007.09.024. URL: https://linkinghub.elsevier.com/retrieve/pii/S0092867407012123.

[54] Kenji Kamimoto et al. "Dissecting cell identity via network inference and in silico gene perturbation". en. In: *Nature* 614.7949 (Feb. 2023), pp. 742–751. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05688-9. URL: https://www.nature.com/articles/s41586-022-05688-9.

[55] E. L. Kaplan and Paul Meier. "Nonparametric Estimation from Incomplete Observations". en. In: *Journal of the American Statistical Association* 53.282 (June 1958), pp. 457–481. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.1958.10501452. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501452.

[56] Fumiki Katsuoka and Masayuki Yamamoto. "Small Maf proteins (MafF, MafG, MafK): History, structure and function". In: *Gene* 586.2 (July 2016), pp. 197–205. ISSN: 0378-1119. DOI: 10.1016/j.gene.2016.03.058. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4911266/.

[57] Jared Katzman et al. "DeepSurv: Personalized Treatment Recommender System Using A Cox Proportional Hazards Deep Neural Network". In: *BMC Medical Research Methodology* 18.1 (Dec. 2018). arXiv:1606.00931 [cs, stat], p. 24. ISSN: 1471-2288. DOI: 10.1186/s12874-018-0482-1. URL: http://arxiv.org/abs/1606.00931.

[58] Hatice S. Kaya-Okur et al. "CUT&Tag for efficient epigenomic profiling of small samples and single cells". en. In: *Nature Communications* 10.1 (Apr. 2019), p. 1930. ISSN: 2041-1723. DOI: 10.1038/s41467-019-09982-5. URL: https://www.nature.com/articles/s41467-019-09982-5.

[59] David R. Kelley, Jasper Snoek, and John L. Rinn. "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks". en. In: *Genome Research* 26.7 (July 2016), pp. 990–999. ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.200535.115. URL: http://genome.cshlp.org/lookup/doi/10.1101/gr.200535.115.

[60] Hyungjin Kim et al. "Preoperative ct-based deep learning model for predicting disease-free survival in patients with lung adenocarcinomas". en. In: *Radiology* 296.1 (July 2020), pp. 216–224. ISSN: 0033-8419, 1527-1315. DOI: 10.1148/radiol.2020192764. URL: http://pubs.rsna.org/doi/10.1148/radiol.2020192764 (visited on 09/07/2023).

[61] Seongho Kim. "ppcor: An R Package for a Fast Calculation to Semi-partial Correlation Coefficients". In: *Communications for statistical applications and methods* 22.6 (Nov. 2015), pp. 665–674. ISSN: 2287-7843. DOI: 10.5351/CSAM.2015.22.6.665. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4681537/.

[62] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: arXiv:1412.6980 (Jan. 2017). DOI: 10.48550/arXiv.1412.6980. URL: http://arxiv.org/abs/1412.6980.

[63] Narine Kokhlikyan et al. *Captum: A unified and generic model interpretability library for PyTorch*. 2020. arXiv: 2009.07896 [cs.LG].

[64] Ilya Korsunsky et al. "Fast, sensitive and accurate integration of single-cell data with Harmony". en. In: *Nature Methods* 16.12 (Dec. 2019), pp. 1289–1296. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0619-0. URL: https://www.nature.com/articles/s41592-019-0619-0.

[65] Anders Krogh and John A. Hertz. "A simple weight decay can improve generalization". In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS'91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Dec. 1991, pp. 950–957. ISBN: 9781558602229.

[66] Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. "Time-to-Event Prediction with Neural Networks and Cox Regression". In: (Sept. 2019). arXiv:1907.00825 [cs, stat]. DOI: 10.48550/arXiv.1907.00825. URL: http://arxiv.org/abs/1907.00825.

[67] Peter Langfelder and Steve Horvath. "WGCNA: an R package for weighted correlation network analysis". In: *BMC Bioinformatics* 9.1 (Dec. 2008), p. 559. ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-559. URL: https://doi.org/10.1186/1471-2105-9-559.

[68] Sophie Lèbre et al. "Statistical inference of the time-varying structure of gene regulation networks". In: *BMC Systems Biology* 4.1 (Sept. 2010), p. 130. ISSN: 1752-0509. DOI: 10.1186/1752-0509-4-130. URL: https://doi.org/10.1186/1752-0509-4-130.

[69] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". en. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 9783319106021. DOI: 10.1007/978-3-319-10602-1_48.

[70] Lin Lu et al. "Deep learning for the prediction of early on-treatment response in metastatic colorectal cancer from serial medical imaging". en. In: *Nature Communications* 12.1 (Nov. 2021), p. 6654. ISSN: 2041-1723. DOI: 10.1038/s41467-021-26990-6. URL: https://www.nature.com/articles/s41467-021-26990-6 (visited on 09/07/2023).

[71] Ben D. Macarthur, Avi Ma'ayan, and Ihor R. Lemischka. "Systems biology of stem cell fate and cellular reprogramming". eng. In: *Nature Reviews. Molecular Cell Biology* 10.10 (Oct. 2009), pp. 672–681. ISSN: 1471-0080. DOI: 10.1038/nrm2766.

[72] Aviv Madar et al. "The inferelator 2.0: A scalable framework for reconstruction of dynamic regulatory network models". In: *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Sept. 2009, pp. 5448–5451. DOI: 10.1109/IEMBS.2009.5334018. URL: https://ieeexplore.ieee.org/document/5334018.

[73] Daniel Marbach et al. "Wisdom of crowds for robust gene network inference". en. In: *Nature Methods* 9.8 (Aug. 2012), pp. 796–804. ISSN: 1548-7105. DOI: 10.1038/nmeth.2016. URL: https://www.nature.com/articles/nmeth.2016.

[74] Adam A. Margolin et al. "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context". In: *BMC Bioinformatics* 7.1 (Mar. 2006), S7. ISSN: 1471-2105. DOI: 10.1186/1471-2105-7-S1-S7. URL: https://doi.org/10.1186/1471-2105-7-S1-S7.

[75] Anthony Mathelier et al. "DNA shape features improve transcription factor binding site predictions in vivo". en. In: *Cell systems* 3.3 (Sept. 2016), p. 278. DOI: 10.1016/j.cels.2016.07.001. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5042832/.

[76] Xueyan Mei et al. "RadImageNet: An Open Radiologic Deep Learning Research Dataset for Effective Transfer Learning". en. In: *Radiology: Artificial Intelligence* 4.5 (Sept. 2022), e210315. ISSN: 2638-6100. DOI: 10.1148/ryai.210315. URL: http://pubs.rsna.org/doi/10.1148/ryai.210315.

[77] Peter G. Mikhael et al. "Sybil: A Validated Deep Learning Model to Predict Future Lung Cancer Risk From a Single Low-Dose Chest Computed Tomography". en. In: *Journal of Clinical Oncology* 41.12 (Apr. 2023), pp. 2191–2200. ISSN: 0732-183X, 1527-7755. DOI: 10.1200/JCO.22.01345. URL: https://ascopubs.org/doi/10.1200/JCO.22.01345.

[78] Peter G. Mikhael et al. "Sybil: A Validated Deep Learning Model to Predict Future Lung Cancer Risk From a Single Low-Dose Chest Computed Tomography". en. In: *Journal of Clinical Oncology* 41.12 (Apr. 2023), pp. 2191–2200. ISSN: 0732-183X, 1527-7755. DOI: 10.1200/JCO.22.01345. URL: https://ascopubs.org/doi/10.1200/JCO.22.01345.

[79] Jaime L. Miller and Patrick A. Grant. "The Role of DNA Methylation and Histone Modifications in Transcriptional Regulation in Humans". In: 61 (2013), pp. 289–317. ISSN: 0306-0225. DOI: 10.1007/978-94-007-4525-4_13. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6611551/.

[80] Thomas Moerman et al. "GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks". eng. In: *Bioinformatics (Oxford, England)* 35.12 (June 2019), pp. 2159–2161. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/bty916.

[81] Kendall K. Morgan. *Lung Cancer Survival Rates & Stages*. en. URL: https://www.webmd.com/lung-cancer/lung-cancer-survival-rates (visited on 06/06/2024).

[82] Kenta Murakami et al. "Generation of functional oocytes from male mice in vitro". en. In: *Nature* 615.7954 (Mar. 2023), pp. 900–906. ISSN: 1476-4687. DOI: 10.1038/s41586-023-05834-x. URL: https://www.nature.com/articles/s41586-023-05834-x.

[83] Surag Nair et al. "Transcription factor stoichiometry, motif affinity and syntax regulate single-cell chromatin dynamics during fibroblast reprogramming to pluripotency". In: *bioRxiv* (Oct. 2023), p. 2023.10.04.560808. DOI: 10.1101/2023.10.04.560808. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10592962/.

[84] "New response evaluation criteria in solid tumours: revised RECIST guideline (version 1.1)". eng. In: 45 (Jan. 2009), pp. 228–247. ISSN: 1879-0852. DOI: 10.1016/j.ejca.2008.10.026.

[85] Mizuki Nishino. "Tumor Response Assessment for Precision Cancer Therapy: Response Evaluation Criteria in Solid Tumors and Beyond". en. In: *American Society of Clinical Oncology Educational Book* 38 (May 2018), pp. 1019–1029. ISSN: 1548-8748, 1548-8756. DOI: 10.1200/EDBK_201441. URL: https://ascopubs.org/doi/10.1200/EDBK_201441.

[86] Mizuki Nishino et al. "New Response Evaluation Criteria in Solid Tumors (RECIST) Guidelines for Advanced Non–Small Cell Lung Cancer: Comparison With Original RECIST and Impact on Assessment of Tumor Response to Targeted Therapy". en. In: *American Journal of Roentgenology* 195.3 (Sept. 2010), W221–W228. ISSN: 0361-803X, 1546-3141. DOI: 10.2214/AJR.09.3928. URL: https://www.ajronline.org/doi/10.2214/AJR.09.3928.

[87] Theresa Phillips. "Small Non-coding RNA and Gene Expression". In: *Nature Education* 1.1 (2008), p. 115.

[88] Hannah A. Pliner et al. "Cicero Predicts cis-Regulatory DNA Interactions from Single-Cell Chromatin Accessibility Data". In: *Molecular Cell* 71.5 (Sept. 2018), 858–871.e8. ISSN: 1097-2765. DOI: 10.1016/j.molcel.2018.06.044. URL: https://www.sciencedirect.com/science/article/pii/S1097276518305471.

[89] Aditya Pratapa et al. "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data". eng. In: *Nature Methods* 17.2 (Feb. 2020), pp. 147–154. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0690-6.

[90] Jing Qin et al. "ChIP-Array: combinatory analysis of ChIP-seq/chip and microarray gene expression data to discover direct/indirect targets of a transcription factor". eng. In: *Nucleic Acids Research* 39.Web Server issue (July 2011), W430–436. ISSN: 1362-4962. DOI: 10.1093/nar/gkr332.

[91] Jing Qin et al. "Inferring gene regulatory networks by integrating ChIP-seq/chip and transcriptome data via LASSO-type regularization methods". eng. In: *Methods (San Diego, Calif.)* 67.3 (June 2014), pp. 294–303. ISSN: 1095-9130. DOI: 10.1016/j.ymeth.2014.03.006.

[92] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018. URL: https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035.

[93] Martin Reck et al. "Five-Year Outcomes With Pembrolizumab Versus Chemotherapy for Metastatic Non–Small-Cell Lung Cancer With PD-L1 Tumor Proportion Score". en. In: *Journal of Clinical Oncology* 39.21 (July 2021), pp. 2339–2349. ISSN: 0732-183X, 1527-7755. DOI: 10.1200/JCO.21.00174. URL: https://ascopubs.org/doi/10.1200/JCO.21.00174.

[94] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: arXiv:1506.01497 (Jan. 2016). arXiv:1506.01497 [cs]. DOI: 10.48550/arXiv.1506.01497. URL: http://arxiv.org/abs/1506.01497.

[95] Ho Sung Rhee and B. Franklin Pugh. "ChIP-exo: A Method to Identify Genomic Location of DNA-binding proteins at Near Single Nucleotide Accuracy". In: *Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.]* 0 21 (Oct. 2012), 10.1002/0471142727.mb2124s100. ISSN: 1934-3639. DOI: 10.1002/0471142727.mb2124s100. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3813302/.

[96] Remo Rohs et al. "Origins of specificity in protein-DNA recognition". eng. In: *Annual Review of Biochemistry* 79 (2010), pp. 233–269. ISSN: 1545-4509. DOI: 10.1146/annurev-biochem-060408-091030.

[97] F. Rojo. "Mechanisms of transcriptional repression". eng. In: *Current Opinion in Microbiology* 4.2 (Apr. 2001), pp. 145–151. ISSN: 1369-5274. DOI: 10.1016/s1369-5274(00)00180-6.

[98] Ryan D. Rosen and Amit Sapra. "TNM Classification". eng. In: *StatPearls*. Treasure Island (FL): StatPearls Publishing, 2024. URL: http://www.ncbi.nlm.nih.gov/books/NBK553187/.

[99] Kathleen Ruchalski et al. "A Primer on RECIST 1.1 for Oncologic Imaging in Clinical Drug Trials". In: 3 (May 2021), e210008. ISSN: 2638-616X. DOI: 10.1148/rycan.2021210008. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8183261/.

[100] Kathleen Ruchalski et al. "Imaging response assessment for oncology: An algorithmic approach". In: *European Journal of Radiology Open* 9 (June 2022), p. 100426. ISSN: 2352-0477. DOI: 10.1016/j.ejro.2022.100426. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9184854/.

[101] Md. Abul Hassan Samee. "Noncanonical binding of transcription factors: time to revisit specificity?" en. In: *Molecular Biology of the Cell* 34.9 (Aug. 2023). Ed. by Doug Kellogg, pe4. ISSN: 1059-1524, 1939-4586. DOI: 10.1091/mbc.E22-08-0325. URL: https://www.molbiolcell.org/doi/10.1091/mbc.E22-08-0325.

[102] Md. Abul Hassan Samee, Benoit G. Bruneau, and Katherine S. Pollard. "A De Novo Shape Motif Discovery Algorithm Reveals Preferences of Transcription Factors for DNA Shape Beyond Sequence Motifs". en. In: *Cell Systems* 8.1 (Jan. 2019), 27–42.e6. ISSN: 24054712. DOI: 10.1016/j.cels.2018.12.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S2405471218304757.

[103] Jacob Schreiber. *tfmodisco-lite*. https://github.com/jmschrei/tfmodisco-lite. 2023.

[104] Li Shen et al. "Deep Learning to Improve Breast Cancer Detection on Screening Mammography". en. In: *Scientific Reports* 9.1 (Aug. 2019), p. 12495. ISSN: 2045-2322. DOI: 10.1038/s41598-019-48995-4. URL: https://www.nature.com/articles/s41598-019-48995-4.

[105] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". en. In: *Proceedings of the 34th International Conference on Machine Learning*. PMLR, July 2017, pp. 3145–3153. URL: https://proceedings.mlr.press/v70/shrikumar17a.html.

[106] Avanti Shrikumar et al. "Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5". In: (2018). DOI: 10.48550/ARXIV.1811.00416. URL: https://arxiv.org/abs/1811.00416.

[107] David A. Siegel et al. "Proportion of Never Smokers Among Men and Women With Lung Cancer in 7 US States". In: *JAMA Oncology* 7.2 (Feb. 2021), pp. 302–304. ISSN: 2374-2437. DOI: 10.1001/jamaoncol.2020.6362. URL: https://doi.org/10.1001/jamaoncol.2020.6362.

[108] Rebecca L. Siegel, Angela N. Giaquinto, and Ahmedin Jemal. "Cancer statistics, 2024". en. In: *CA: A Cancer Journal for Clinicians* 74.1 (Jan. 2024), pp. 12–49. ISSN: 0007-9235, 1542-4863. DOI: 10.3322/caac.21820. URL: https://acsjournals.onlinelibrary.wiley.com/doi/10.3322/caac.21820.

[109] Peter J Skene and Steven Henikoff. "An efficient targeted nuclease strategy for high-resolution mapping of DNA binding sites". In: *eLife* 6 (Jan. 2017). Ed. by Danny Reinberg, e21856. ISSN: 2050-084X. DOI: 10.7554/eLife.21856. URL: https://doi.org/10.7554/eLife.21856.

[110] Claudia Skok Gibbs et al. "High-performance single-cell gene regulatory network inference at scale: the Inferelator 3.0". en. In: *Bioinformatics* 38.9 (Apr. 2022). Ed. by Anthony Mathelier, pp. 2519–2528. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btac117. URL: https://academic.oup.com/bioinformatics/article/38/9/2519/6533443.

[111] E. P. van Someren et al. "Least absolute regression network analysis of the murine osteoblast differentiation network". eng. In: *Bioinformatics (Oxford, England)* 22.4 (Feb. 2006), pp. 477–484. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti816.

[112] Julie Soutourina. "Transcription regulation by the Mediator complex". eng. In: *Nature Reviews. Molecular Cell Biology* 19.4 (Apr. 2018), pp. 262–274. ISSN: 1471-0080. DOI: 10.1038/nrm.2017.115.

[113] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. ISSN: 1533-7928. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[114] Ralph Stadhouders, Guillaume J. Filion, and Thomas Graf. "Transcription factors and 3D genome conformation in cell-fate decisions". en. In: *Nature* 569.7756 (May 2019), pp. 345–354. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1182-7. URL: https://www.nature.com/articles/s41586-019-1182-7.

[115] Zhou T et al. "DNAshape: a method for the high-throughput prediction of DNA structural features on a genomic scale". en. In: *Nucleic acids research* 41.Web Server issue (July 2013). ISSN: 1362-4962. DOI: 10.1093/nar/gkt437. URL: https://pubmed.ncbi.nlm.nih.gov/23703209/.

[116] Kazutoshi Takahashi and Shinya Yamanaka. "Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors". eng. In: *Cell* 126.4 (Aug. 2006), pp. 663–676. ISSN: 0092-8674. DOI: 10.1016/j.cell.2006.07.024.

[117] Kazutoshi Takahashi et al. "Induction of Pluripotent Stem Cells from Adult Human Fibroblasts by Defined Factors". en. In: *Cell* 131.5 (Nov. 2007), pp. 861–872. ISSN: 00928674. DOI: 10.1016/j.cell.2007.11.019. URL: https://linkinghub.elsevier.com/retrieve/pii/S0092867407014717.

[118] Zhenyu Tang et al. "Deep Learning of Imaging Phenotype and Genotype for Predicting Overall Survival Time of Glioblastoma Patients". In: *IEEE transactions on medical imaging* 39.6 (June 2020), pp. 2100–2109. ISSN: 0278-0062. DOI: 10.1109/TMI.2020.2964310. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7289674/.

[119] Evelyn E. Telfer et al. "Making a good egg: human oocyte health, aging, and in vitro development". en. In: *Physiological Reviews* 103.4 (Oct. 2023), pp. 2623–2677. ISSN: 0031-9333, 1522-1210. DOI: 10.1152/physrev.00032.2022. URL: https://journals.physiology.org/doi/10.1152/physrev.00032.2022.

[120] Gerald Thiel, Michael Lietz, and Mathias Hohl. "How mammalian transcriptional repressors work". eng. In: *European Journal of Biochemistry* 271.14 (July 2004), pp. 2855–2862. ISSN: 0014-2956. DOI: 10.1111/j.1432-1033.2004.04174.x.

[121] Chiu Tp et al. "DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding". en. In: *Bioinformatics (Oxford, England)* 32.8 (Apr. 2016). ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btv735. URL: https://pubmed.ncbi.nlm.nih.gov/26668005/.

[122] Sergiy Velychko et al. "Excluding Oct4 from Yamanaka Cocktail Unleashes the Developmental Potential of iPSCs". en. In: *Cell Stem Cell* 25.6 (Dec. 2019), 737–753.e4. ISSN: 19345909. DOI: 10.1016/j.stem.2019.10.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S1934590919304230.

[123] Allon Wagner, Aviv Regev, and Nir Yosef. "Revealing the vectors of cellular identity with single-cell genomics". en. In: *Nature Biotechnology* 34.11 (Nov. 2016), pp. 1145–1160. ISSN: 1546-1696. DOI: 10.1038/nbt.3711. URL: https://www.nature.com/articles/nbt.3711.

[124] Lingfei Wang et al. "Dictys: dynamic gene regulatory network dissects developmental continuum with single-cell multiomics". en. In: *Nature Methods* 20.9 (Sept. 2023), pp. 1368–1378. ISSN: 1548-7105. DOI: 10.1038/s41592-023-01971-3. URL: https://www.nature.com/articles/s41592-023-01971-3.

[125] Matthew T. Weirauch et al. "Determination and inference of eukaryotic transcription factor sequence specificity". eng. In: *Cell* 158.6 (Sept. 2014), pp. 1431–1443. ISSN: 1097-4172. DOI: 10.1016/j.cell.2014.08.009.

[126] Kris A. Wetterstrand. *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP).* URL: https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data.

[127] Yiwen Xu et al. "Deep learning predicts lung cancer treatment response from serial medical imaging". eng. In: *Clinical Cancer Research: An Official Journal of the American Association for Cancer Research* 25.11 (June 2019), pp. 3266–3275. ISSN: 1557-3265. DOI: 10.1158/1078-0432.CCR-18-2495.

[128] Adam Yala et al. "Toward robust mammography-based models for breast cancer risk". en. In: *Science Translational Medicine* 13.578 (Jan. 2021), eaba4373. ISSN: 1946-6234, 1946-6242. DOI: 10.1126/scitranslmed.aba4373. URL: https://www.science.org/doi/10.1126/scitranslmed.aba4373.

[129] Lin Yang et al. "TFBSshape: a motif database for DNA shape features of transcription factor binding sites". eng. In: *Nucleic Acids Research* 42.Database issue (Jan. 2014), pp. D148–155. ISSN: 1362-4962. DOI: 10.1093/nar/gkt1087.

[130] Jiawen Yao et al. *Deepprognosis: preoperative prediction of pancreatic cancer survival and surgical margin via contrast-enhanced ct imaging.* arXiv:2008.11853 [cs, eess]. Aug. 2020. DOI: 10.48550/arXiv.2008.11853. URL: http://arxiv.org/abs/2008.11853 (visited on 09/07/2023).

[131] Jiawen Yao et al. "Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks". In: *Medical Image Analysis* 65 (Oct. 2020), p. 101789. ISSN: 1361-8415. DOI: 10.1016/j.media.2020.101789. URL: https://www.sciencedirect.com/science/article/pii/S1361841520301535.

[132] Junying Yu et al. "Induced pluripotent stem cell lines derived from human somatic cells". eng. In: *Science (New York, N.Y.)* 318.5858 (Dec. 2007), pp. 1917–1920. ISSN: 1095-9203. DOI: 10.1126/science.1151526.

[133] Kenneth S. Zaret and Jason S. Carroll. "Pioneer transcription factors: establishing competence for gene expression". In: *Genes & Development* 25.21 (Nov. 2011), pp. 2227–2241. ISSN: 0890-9369. DOI: 10.1101/gad.176826.111. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3219227/.

[134] Shilu Zhang et al. "Inference of cell type-specific gene regulatory networks on cell lineages from single cell omic datasets". en. In: *Nature Communications* 14.1 (May 2023), p. 3064. ISSN: 2041-1723. DOI: 10.1038/s41467-023-38637-9. URL: https://www.nature.com/articles/s41467-023-38637-9.

[135] Bolei Zhou et al. "Learning Deep Features". In: (2015). DOI: 10.48550/arXiv.1512.04150. URL: http://arxiv.org/abs/1512.04150.

[136] Jian Zhou and Olga G Troyanskaya. "Predicting effects of noncoding variants with deep learning–based sequence model". en. In: *Nature Methods* 12.10 (Oct. 2015), pp. 931–934. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/nmeth.3547. URL: https://www.nature.com/articles/nmeth.3547.

[137] Tianyin Zhou et al. "From the Cover: Quantitative modeling of transcription factor binding specificities using DNA shape". en. In: *Proceedings of the National Academy of Sciences of the United States of America* 112.15 (Apr. 2015), p. 4654. DOI: 10.1073/pnas.1422023112. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4403198/.