Interpretable Machine Learning Methods with Applications in Genomics

HARVARD UNIVERSITY
Graduate School of Arts and Sciences

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the

Department of Biostatistics

have examined a dissertation entitled

"Interpretable Machine Learning Methods with Applications in Genomics"

presented by Matthew Ploenzke

candidate for the degree of Doctor of Philosophy and hereby certify that it is worthy of acceptance.

Signature ....

Typed name: Prof. Rafael Irizarry

Signature ....GParmigiani..

Typed name: Prof. Giovanni Parmigiani

Signature ....

Digitally signed by Danielle Braun
DN: cn=Danielle Braun, o=Harvard University, ou, email=dbraun@mail.harvard.edu, c=US
Date: 2020.07.01 17:28:40 -04'00'

Typed name: Dr. Danielle Braun

Signature ....

Typed name: Prof. Peter Koo

Date: July 1, 2020

Interpretable Machine Learning Methods with Applications in Genomics

A DISSERTATION PRESENTED
BY
MATTHEW PLOENZKE
TO
THE DEPARTMENT OF BIOSTATISTICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
BIOSTATISTICS

HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS
JULY 2020

Thesis advisor: Professor Rafael Irizarry Matthew Ploenzke

Interpretable Machine Learning Methods with Applications in Genomics

Abstract

A primary goal in biology is understanding the relationship between genomic sequence and cell state or function. Pharmacogenomic experiments, for instance, measure how different genomic profiles correlate with cell survival under varying drug dosages, thus finding genomic markers and signatures associated with effective therapy. ChIP-seq experiments, on the other hand, isolate proteins and/or transcription factors (TF) bound to the genome and subsequently measure genomic sequence variability with these TFs across different conditions. In both these cases the fundamental problem formulation is set up with some genomic input space, X, and interest lies in associations with some outcome Y. How one defines either X or Y for any given application has a tremendous downstream effect on the conclusions drawn. The focus of this dissertation is the development of methods for three -omics applications which address the importance of defining X and Y in a data driven manner. Improved model interpretability and an agreement with intuition highlight the benefit of such an approach for each application. A multi-level model is detailed in the pharmacogenomics application to show the effect assuming an outcome variable Y is a continuous univariate random variable when in fact Y follows a two-component mixture distribution. Estimated associations between X and Y are compared under the differing assumptions, as well as bivariate measures of association such as those between the Y collected in one experiment and those collected in another. The second application uses weight constraints and regularization to illustrate how the inherent structure of the genomic sequence X, namely being composed of a string of nucleotides, allows one to transform X into a set of learnable sequence motifs using the first layer weights in convolutional neural networks (CNNs). These feature extractors allow one to encode prior information into the sequence-function analysis and extract interpretable sequence motifs after fitting the model. The final results again focus on CNNs and TF binding and show the utility of employing an exponential activation function in the first layer feature extractors. Specifically, measures of model interpretability are improved relative to state-of-the-art methods and there are no effects on test set accuracy. Interestingly, the learned functions with the exponential tend to be less noisy and more robust to hyper-parameter selections. A discussion of deep learning for TF binding applications completes the dissertation.

Contents

Listing of figures

For my brother Greg.

Acknowledgments

I would like to acknowledge the following National Institute of Health funding opportunities for making the completion of this dissertation possible: NIH NCI F99/K00 Pre-doctoral to Post-doctoral Transition Award, NIH Training Grant T32 CA09337, and NIH Training Grant T32 GM074897. I would like to thank members of the Harvard University Department of Biostatistics and the Dana-Farber Department of Data Sciences for their help with computational-related services, completing program-related duties, and in applying for funding opportunities. I would like to thank my advisor, Dr. Rafael Irizarry, for his academic and career-related advice, statistical and computational insights, and for serving as an outstanding mentor throughout my degree. I would like to thank members of my dissertation committee, Dr. Giovanni Parmigiani, Dr. Danielle Braun, and Dr.Peter K. Koo. I would like to thank other academic mentors throughout my training including Dr. Giacomo De Giorgi and Dr. Giljae Lee. I would like to thank my parents, Patti and Dan, for their continued support over years and my brother, Chris, for pushing me to be my best. I would like to thank my partner, Christine, and lastly, I would like to thank my many friends and colleagues over the years.

1

Reassessing pharmacogenomic cell sensitivity with multi-level statistical models

Pharmacogenomic experiments allow for the systematic testing of drugs, at varying dosage concentrations, to study how genomic markers correlate with cell sensitivity to treatment. We formulate a hierarchical mixture model to estimate the drug-specific mixture distributions for estimating cell sensitivity and for assessing drug effect type (broad versus targeted effect). We motivate two formulations: 1) fit independently *within* dataset and 2) fit jointly *across* dataset. Case studies are provided to assess pairwise agreements for cell lines/drugs within the intersection of two datasets and confirm moderate pairwise agreement between many publicly-available pharmacogenomic datasets. An analysis is presented for the drug Crizotinib and identifies high estimated posterior drug sensitivity for cells harboring EML4-ALK or NPM1-ALK gene fusions, as well as significantly down-regulated cell matrix pathways associated with Crizotinib sensitivity.

1.1 Introduction

Pharmacogenomic studies offer insight into which genomic markers predict drug response and hold potential for developing effective personalized cancer treatment regimes [36,10,70,77,145,12,24]. The approach relies on quantifying the response of cell lines to variable dosage concentrations of drugs of interest. The cell line's relative viability is recorded at each concentration for each cell-line/drug combination and a dose response curve is produced (supplementary figure S1). Genomic measurements are also obtained for each cell line and used to learn molecular signatures correlated with drug response to discover novel biomarkers. For this final part of the analysis, the dose curves are summarized into one number with statistics such as the area-above-the-curve (AAC), the half maximal effective concentration (EC50) or the half maximal inhibitory concentration (IC50). To identify biomarkers, this summary statistic is defined as the outcome and association tests or machine learning algorithms are applied to search for predictive genomic measurements. The quality of these studies are therefore entirely dependent on the quality of the dose curve summaries used as outcomes in these downstream analysis.

In a 2013 paper, the quality of these data were questioned due to the fact that dose curve summaries for the same cell line/drug combinations measured by two independent studies, the Cancer

Cell Line Encyclopedia (CCLE) dataset[10] and the Genomics of Drug Sensitivity in Cancer (GDSC) dataset[36], exhibited low Spearman correlations[46]. However, several follow up studies[60,22,37,111,47,113,100,112] pointed out that Spearman correlation was not an appropriate measure of agreement for these dose curve summaries. For example Geeleher et al.[37] argued that Spearman correlation does not reflect the true level of concordance for highly-targeted drugs because data from the relatively few cell lines sensitive to the compound, such as the case with nilotinib, appeared as outliers when compared to the majority of resistant cell lines[37]. Thus the random measurement error associated with the resistant cell lines will dominate any biological signal. This debate led several groups to propose dichotomizing the data into *sensitive* and *resistant* cell lines for each drug and computing binary measures of agreement between these assigned labels[22,113]. For example, Consortium et al.[22] binarized the sensitivities using the waterfall method (see [10,46] for details) and found improved agreement upon calculating Cohen's Kappa statistic relative to the Spearman correlation[126,22]. Alternative binarization approaches include denoting sensitive cell lines with a hard cutoff (i.e. AAC > 0.2, [113]), assigning sensitive cell lines as those with sensitivity greater than the 66th percentile, assigning resistant to those less than 33rd percentile, and discarding the observations in the middle tercile[47], and fitting two-component mixture distributions[47,60]. Alternative measures of agreement assessed by Safikhani et al.[113] which account for the binarization of the data include Matthew's correlation[89], Cramer's V[25], and Informedness[99] .

Discretizing cell line sensitivities and computing a binary metric of concordance highlighted greater levels of agreement than originally reported for highly-targeted drugs regardless of method and metric employed[113] and motivated the use of dichotomization for downstream analysis[30,151]. However, limitations hinder the universal applicability of each method across pharmacogenomic datasets for several reasons. First, determining a hard threshold to call sensitive cell lines is a difficult manual procedure with no clear and evident cutoff both within study as well as across study (supplementary figure S3). Second, discarding the middle tercile of observations reduces the sample size by 33% and decreases statistical power for biomarker discovery. Finally, it ignores the fact that while most drugs appear to affect only a few cell-lines, what we refer to as a *targeted* effects, some drugs appear to have *broad* effects (supplementary figure S3) making dichotomization inappropriate for downstream analysis. While a strict labelling of drugs as broad effect or targeted effect may be achieved using domain expertise or ad-hoc thresholding[113], the notion of drug *targetedness* is not a well defined concept which necessarily justifies a strict dichotomization; a continuum of drug targetedness is reasonable and may better model the complexity of effect type. Indeed, drugs exhibiting modest degrees of targetedness, such as the drug PD-0332991, may explain why the distribution of AAC appears slightly targeted in the CCLE dataset but less so in the GDSC dataset (supplementary figure S3).

Fitting two-component mixture distributions has been proposed as a data-driven way to binarize cell lines into sensitivity and resistant[60,47]. However, such a procedure is not applicable to the broad effect drugs whose sensitivity distributions are continuous and unimodal. Here, we extend and improve on this idea by proposing a multi-level mixture model in which the first level estimates drug type, broad effect or sensitive, with two-component mixture distributions and the second models the response of the cell-line conditioned on the drug type modeled in the first level. Specifically, two-component mixture distributions are estimated only for targeted drugs in order to classify cells into sensitive or resistant, whereas single-component distributions are estimated for broad effect drugs. Fitting this model permits us to fully described all cell-type/drug combinations without manual annotations and supports the notion that some cell type distributions are continuous while others are binary. A further advantage of our approach is that the probabilistic approach permits us to combine data from different studies in a statistically rigorous way. We note that several other large-scale pharmacogenomic datasets are publicly-available in addition to CCLE and GDSC[70,145,12,24]. We fit our model to all 1,381 distinct cell lines and 733 distinct drugs available in five cell line viability datasets and estimate agreement across common cell lines and drugs. We then use the results of our fitted model to detect biomarkers that are not found with previous approaches. Specifically, a comparison of biomarkers associated with the ALK-inhibitor crizotinib, a highly-targeted drug used for treating non small cell lung carcinoma (NSCLC), identifies significantly down-regulated cell matrix pathways drivers of dose sensitivity.

1.2 Results

Correlation is not an appropriate association measure for targeted drugs

Bivariate correlation measures between the reported dose curve summaries have been used to assess pairwise dataset concordance for cell line-drug pairs assayed in more than one study[46,24,107]. When comparing the CCLE and GDSC datasets, both Pearson and Spearman correlation measures suggest strong agreement for some drugs but moderate to low for others (supplementary figure S4)a. We observe similar results for comparisons between other studies (supplementary figures S10, S11). We confirm previously published observation that these different results are explained by some drugs having broad effects and others being targeted[27,113,60,47]. A targeted drug that, for example, inhibits a specific pathway will be effective only against cells which up-regulate that pathway. This will result in two collections of cells, sensitive and resistant to that drug, and thus the distribution of the quantified effect (AAC, for example) will be a mixture of two components (figure 1.1b). Because correlation is a summary statistic defined for bivariate normal variables, in general, it does not provide a useful summary of association for data with an underlying dichotomous nature (supplementary figure S2). Measures that explicitly model the underlying two-component generating distributions, such as the odds ratio, result in consistent assessments of agreement and demonstrate consistent measurements from these studies (supplementary figure S4b).

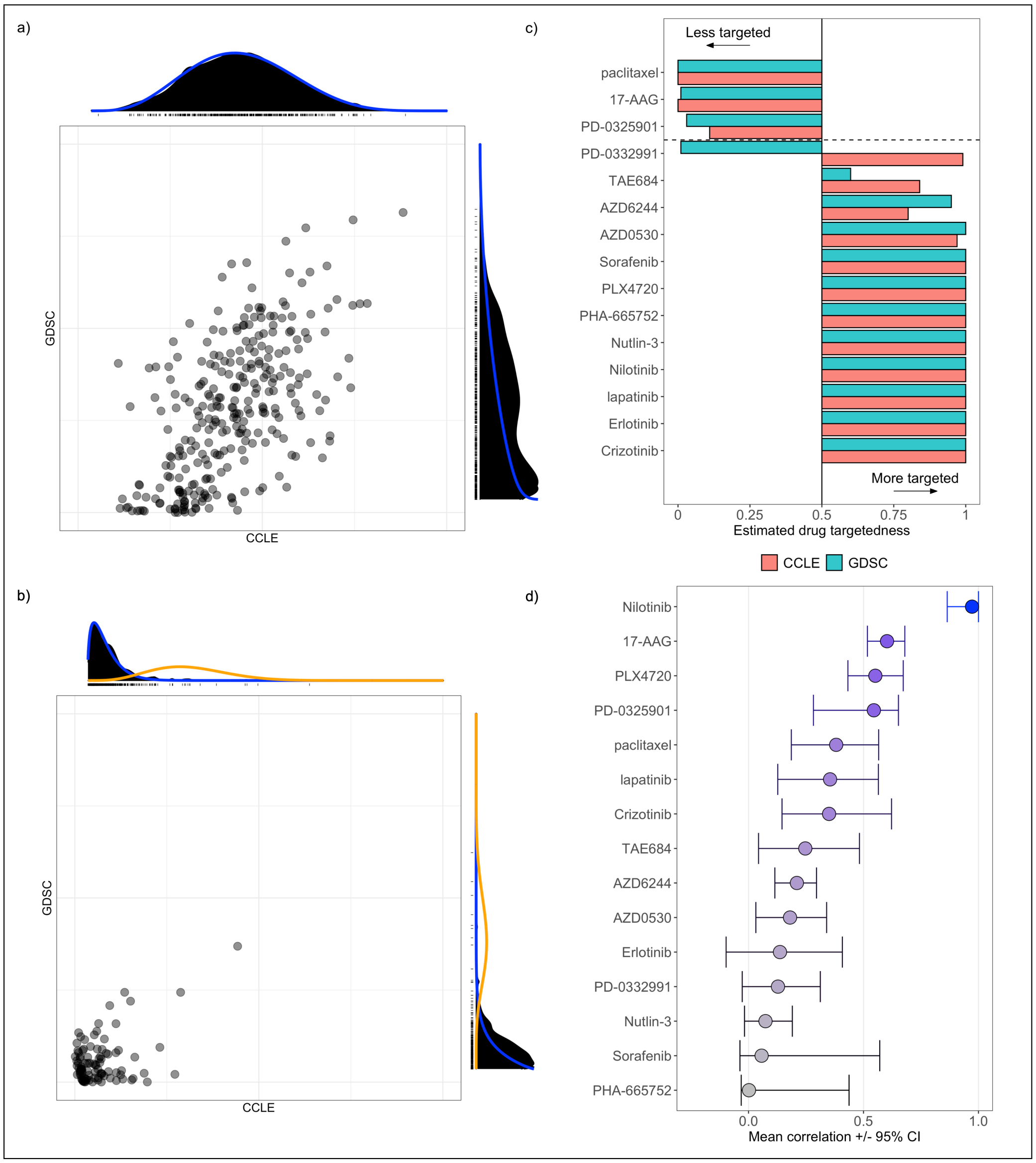


Figure 1.1: Estimated posterior distributions of cell sensitivity for CCLE (X-axis) and GDSC (Y-axis) for broad effect drug (**a** 17-AAG) and targeted drug (**b** Crizotinib). **c)** Estimated posterior probability of drug targetedness for CCLE (salmon) and GDSC (green) for drugs tested in common between the two studies. Solid line represents an estimated 50% probability of the drug being targeted. 14 out of the 15 drugs exhibit strong agreement between drug type. **d)** Mean correlation +/- 95% confidence intervals based on 1000 samplings from the estimated posterior distributions for CCLE and GDSC.

Multi-level model provides interpretable parameterization

To better quantify the effect of drugs on cell-type we developed a two-level mixture model in which the first level accounts for drug targetedness, broad in effect or targeted, and the second level models cell sensitivity conditioned upon drug type. We modeled targeted drug sensitivity with a two-component mixture distribution. If we condition on the drug being targeted, the effect of the drug can be quantified with the posterior probabilities of being sensitive. If we condition on the drug being broad effect, the effect can be quantified with continuous measures such as the Z-score or an empirical cumulative distribution function. Note that this model can be applied to each dataset separately or jointly to all datasets. Applying the model *within-dataset* is useful for validation by calculating pairwise study agreements of drugs and cell lines. Applying the model *across-datasets* is useful to estimate a unified measure of cell sensitivity for each cell-drug combination. We used the latter approach to estimate biomarker associations. Below we provide a summary of how the model provides a useful quantification with further details in the Methods Section.

For any cell lines i, drug j, and pharmacogenomic dataset k we denote the observed dose curve summary statistic with $Y_{i,j,k}$. For the results presented in this section we use AAC values standardized to be between 0 and 1. We define the dichotomous latent variable W_j to be 0 if the drug has a broad effect and 1 if it is a targeted drug. We define $\rho \equiv \Pr(W_j = 1)$ as the proportion of drugs that are targeted. For targeted drugs, when $W_j = 1$, we define another latent variable $Z_{i,j}$ to be 1 if cell line i is sensitive to drug j and 0 otherwise. We next define

$$\pi_j \equiv \Pr(Z_{i,j} = 1)$$

as the proportion of cell lines sensitive to drug j, define $p^t(\pi_j) \equiv p(\pi_j \mid W_j = 1)$ which we assume to be beta distributed with parameters a^t and b^t. Because dichotomizing is not appropriate when $W_j = 0$, for convenience, we assume $\pi_j = 1$ and $Z_{i,j} = 1$ for all i, j when $W_j = 0$.

We then model the bimodal behaviour of the observed summaries Y_i, j, k for targeted drugs with the following mixture distribution

$$z_{i,j} p^s_{j,k}(Y_{i,j,k}) + (1 - z_{i,j}) p^r_{j,k}(Y_{i,j,k})$$

with $p^s_{j,k}$ the distribution density of the sensitive component for drug j in dataset k and $p^r_{j,k}$ the distribution density of the resistant component. We assume both p^s and p^r are beta distributions with with parameters $c^s_{j,k}$ and $d^s_{j,k}$, and $c^b_{j,k}$ and $d^b_{j,k}$, respectively. Finally, we assume that for broad effect drugs, $Y_{i,j,k}$ follows a beta distribution $p^b_{j,k}$ with parameters $c^b_{j,k}$ and $d^b_{j,k}$. We estimate ρ, $c^b_{j,k}$, $d^b_{j,k}$, $c^r_{j,k}$, $d^r_{j,k}$, $c^s_{j,k}$, $d^s_{j,k}$ using maximum likelihood estimation.

Model-based measures of drug targetedness and cell sensitivity

With these assumptions and estimates in place we can now compute quantities useful for downstream analyses. Specifically, for targeted drugs we can compute the posterior probability of cell line i being sensitive to drug j, $\Pr(Z_{i,j} = 1 \mid Y_{1,j,k}, \ldots, Y_{I,j,k}, W_j = 1)$. Note that we can compute this posterior probability for each study k and use this to assess agreement, or compute one quantity using data from all studies. For broad effect drugs we compute the cumulative probability

$$\Pr(Y_{i,j,k} \leq y \mid W_j = 0) = \int_0^y p^b_{j,k}(y \mid c^b_{j,k}, d^b_{j,k}) dy$$

as it will remove study effects such as the location shifts (supplementary figure S3a; paclitaxel and 17-AAG, supplementary figure 1.5a).

For the above calculations we need to decide if a drug is targeted or not. To guide this decision we compute the posterior probability of a drug being targeted given the observed data. Specifically, for each drug j and study k we can compute the posterior distribution $\Pr(W_j = 1 \mid Y_{1,j,k}, \ldots, Y_{I,j,k})$

with I the total number of cell lines. We use this estimate to measure drug targetedness (figure 1.1c).

Across-study agreement is high

We reassessed three pharmacogenomic dataset comparisons[46,24,107]. We begin with the oft-compared Cancer Cell Line Encyclopedia (CCLE) and Genomics of Drug Sensitivity in Cancer (GDSC) datasets to validate the model[10,36]. We considered the entirety of each dataset and fit models independently to each, making use of all drugs tested across all cell lines. We then restricted the pairwise concordance analysis to those cell line/drug pairs tested in common in both experiments. Statistical assessments of the different dose curve summary statistics find that AAC has better properties than EC50 or IC50[60,47], and is the quantification used here, when available. Dose curve summaries were computed based on common dosage concentration ranges when possible. After model fitting, the estimated posterior quantities of interest and distributional parameters, such as the posterior probability of cell sensitivity and the posterior probability of drug targetedness, were extracted and used for the assessments. Confidence intervals were computed based on Monte Carlo samplings from the estimated posterior distributions. Additional details are found in the Methods section.

Agreement varies in CCLE and GDSC despite high drug type agreement

Model-based sensitivities were estimated for the CCLE and GDSC datasets based on the recomputed AACs available in the PharmacoGx R package[131]. The estimated posterior probabilities of drug targetedness were consistent for 14 of the 15 drugs tested in common between the CCLE and GDSC studies (figure 1.1c, solid vertical line indicates 50% probability of being targeted). Only the drug PD-0332991 was estimated more targeted in CCLE but more broad effect in GDSC. Comparing the original distributions of AAC for all cell lines assayed for PD-0332991 in CCLE and GDSC (supplementary figure S3a, sixth drug from the top) highlights the disconnect; sensitivities do indeed appear targeted in CCLE but much less so in GDSC. We assigned drugs to broad effect or targeted effect by computing the product between the studies of the estimated posterior drug type quantities and selecting the maximum. In the cases of 17-AAG, paclitaxel, and PD-0325901 this resulted in drug assignment to broad effect (supplementary figures S4,S5; broad effect drugs denoted in yellow). For all other drugs the estimated posterior probability of the drug being targeted in both studies was larger (supplementary figures S4,S5; targeted drugs denoted in purple). These data-driven drug types match manual annotations for 14 of 15 drugs tested in common between CCLE and GDSC, and for 22 of 24 drugs tested in CCLE[113,60].

We then measured the agreement between the two datasets through Monte Carlo sampling from the estimated posterior distributions having conditioned on the observed AAC values (figure 1.1d, mean correlation and 95% confidence intervals based on 1000 samplings, see Methods for details). All drugs estimated to be more broad effect in nature (17-AAG, paclitaxel, PD-0325901) exhibit moderate-to-high agreement, as do several of the more targeted drugs nilotinib, PLX4720, lapatinib, crizotinib; mean correlation > 0.25). Estimated agreement is moderate but significantly greater than 0 ($0 \notin 95\%$ CI) for the drugs TAE684, AZD6244, AZD0530. For the remaining five drugs the mean estimated correlation is non-negative however the estimates of uncertainty suggest low-to-moderate agreement at best. This often appears to be due to a lack of cell lines sampled in common from the sensitive mixture components (supplementary figure S7,S8, tick marks on mixture distributions represent full set of cell lines tested in each dataset whereas points represent cell lines sampled in common).

Maximum *a posteriori* (MAP) estimates of agreement are improved for nine of the twelve targeted drugs in comparison to a naïve binarization denoting sensitive cells as those with measured AAC>0.2 (supplementary figure S5a, purple points). Moderate-to-high agreement is again confirmed for four highly targeted drugs (nilotinib, PLX4720, lapatinib, and crizotinib) and three broad effect drugs (supplementary figure S5b, yellow points. Pearson correlation on the X-axis computed using the raw AAC values and on the Y-axis computed using the estimated posterior cumulative distribution functions). Several of the drugs exhibit high targetedness and high agreement (crizotinib, nilotinib, PLX4720) using Matthew's correlation coefficient (MCC) as the metric for assessing agreement (supplementary figure S5a) attain much lower values of estimated agreement when using Pearson correlation as the metric instead (supplementary figure S5b, Y-axis). Additionally, the Pearson correlations calculated from the *raw* AAC values are notably higher (i.e. the points fall significantly below the diagonal) highlighting the effect of modelling a targeted drug as a broad effect drug; namely the influence of assuming the data arise from a single continuous distribution when in fact the data generating process is a mixture of two distributions.

Three additional pharmacogenomic datasets are available for download in the PharmacoGx package[131]. These are from 1) the Genentech Cell Line Screening Initiative (gCSI)[70], 2) the Institute for Molecular Medicine Finland (FIMM)[145], and 3) the Cancer Therapeutics Response Portal (CTRPv2)[12,108,117]. Additionally, we replace the GDSC dataset with the newer GDSC1000 dataset and then estimate model parameters independently in each of the five datasets. Reasonable and somewhat consistent levels of agreement are found for each pairwise comparison between any of the five datasets (supplementary figure S14). Correlations based on the modeling procedure as opposed to Spearman rank correlation computed using the raw AAC values are predictably improved for many of the targeted drugs present in at least two of datasets. The gCSI dataset is the most discordant of the five, attaining the lowest levels of agreement with the CTRPv2 dataset, GDSC1000 dataset, and FIMM dataset, respectively. While approximately 75% of estimated drug types are consistent across many pairwise dataset comparisons (supplementary figure S15, just over 50% are equal between gCSI and CTRPv2, again highlighting relatively lower agreement between these datasets.

A lack of sensitive cells influences reported agreement in PRISM

Corsello et al.[24] developed a public resource called the Cancer Dependency Map containing dose sensitivities for 4,518 drugs tested across 578 cell lines by measuring relative barcode abundance with the PRISM molecular barcoding and multiplexed screening method. A validation of the method was performed by comparing the sensitivities obtained from the PRISM multiplexed cell line profiling with the sensitivities obtained from the CTRP and GDSC studies[24]. Dose curve summaries (measured by AAC) were computed based over common dosage ranges and only cell lines and drugs tested in all *three* experiments were included. Corsello et al.[24] report moderate levels of agreement which are broadly consistent across pairwise comparison (CTRP and GDSC, CTRP and PRISM, GDSC and PRISM) based on Pearson correlations computed across common cell lines for each drug (supplementary figure S12a X-axis). Interestingly, they also find a large number of drugs target selective subsets of cell lines and the selectivity is at times predictable based on molecular features, while other drugs with highly unimodal activity distributions are less predictable. Such conclusions support the notion of a drug type continuum and motivate the use of the multi-level model previously described, with agreement measures taking into consideration the underlying two-component data distributions.

We repeated the comparative analysis described by Corsello et al.[24] and estimated drug type and cell type using the multi-level model fit only to those cell line/drug pairs present in all three datasets. MC sampling from the estimated posterior distributions was used to obtain 95% confidence intervals for measuring the pairwise agreement (figure 1.2a). Mean estimated agreements between sampled cell type are significantly lower between the GDSC and PRISM datasets than the GDSC and CTRP datsets, however CTRP and PRISM agree nearly as well CTRP-GDSC (figure 1.2b). The varying levels of agreement are more pronounced by calculating the correlation between the sampled drug types for each of the 1000 MC samplings (figure 1.2c); Matthew's correlation coefficient computed between the sampled drug types is significantly higher in the CTRP and GDSC comparison, with a mean correlation of about 0.72, whereas the mean correlation between drug types is 0.6 in the CTRP-PRISM comparison and even lower for GDSC-PRISM at 0.58. As noted by Corsello et al. [24], there is a clear trend between the correlations in that relative levels of agreement for many drugs are consistent across pairwise comparison. Indeed, many drugs exhibiting moderate-to-high agreement found in the CCLE-GDSC comparison also attain comparable levels herein (e.g. paclitaxel, crizotinib, PLX4720), as do drugs with lower agreement (nutlin-3, erlotinib). However this is not true universally, as drugs like lapatinib attain low agreement in both figure 1.2a and [24] but high agreement in figures 1.1d and 1.3f. The reason for this is a lack of sensitive cells present in the 3-way intersection and a result of the highly targeted nature of the drugs. When considering the full datasets and refitting the models, then calculating the agreement between cell lines present in *any two* datasets, we observe more reasonable levels of agreement for lapatinib (supplementary figure S13).

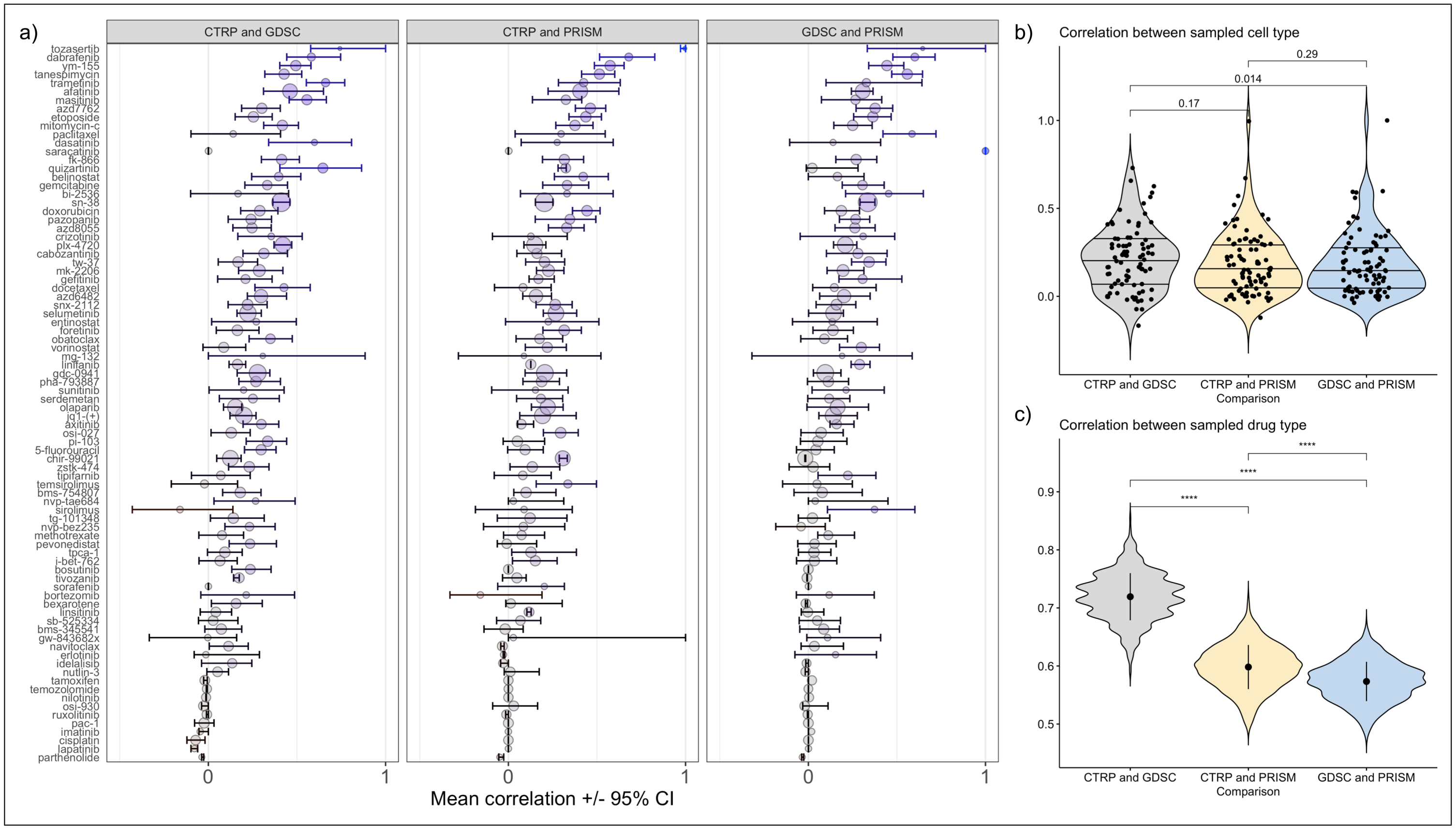


Figure 1.2: **a)** Mean estimated agreement +/- 95% confidence intervals per drug present in CTRP, GDSC, and PRISM confirm moderate and consistent agreement. Panels represent pairwise comparisons for AAC values calculated over common dosage concentrations for all cell lines/drugs tested in common between all three studies (AAC values provided by[24]). **b)** Estimated agreement is significantly higher for the CTRP-GDSC pairwise comparison than for either including the PRISM dataset. Each point represents the mean estimated correlation based on the MC drawings from the estimated posterior distribution. Horizontal bars represent 25th, 50th, and 75th percentiles. **c)** Distributions of drug type correlation based on 1000 MC samplings from the estimated posterior distributions indicates significantly higher agreement between drug types for the CTRP-GDSC comparison. See supplementary methods for additional details about calculation.

MAP measures of agreement further validate moderate but consistent concordance (figure S12a Y-axis). Several drugs with the lowest concordance attain more reasonable levels of agreement when adjusting for the targeted nature of the drugs. Notably, the lowest Spearman correlations in all three pairwise comparisons (figure S12b, right-hand panel outlier points) improve and confirm consistent agreement between the datasets. Collectively there is no evident decrease in agreement based upon the estimated posterior sensitivities (little-to-no change in P-values as provided above the box plots). Additionally, a large proportion (>0.75) of the drugs tested in common exhibit similar degrees of drug targetedness for all pairwise comparisons (figure S12c).

Targeted drugs are more reproducible than broad effect drugs in CellMinerCDB

The CellMinerCDB is a web-based resource useful for unifying the richest cancer cell line datasets and identifying pharmacogenomic determinants and signatures of drug response[85,107]. Specifically, it integrates the NCI-60, GDSC, and CTRP datasets, with sensitivity measured by the IC50 or GI50 metrics for NCI-60 and GDSC but AAC for CTRP. Regardless of metric, however, Rajapakse et al. [107] find certain drugs exhibit specific activity for a small collection of cells (i.e. targeted) or appear broadly active in mechanism, and this is consistent across dataset[107]. Estimated posterior distributions corroborate this observation; for instance the estimated posterior probability of targeted for the drug dabrafenib is greater than 0.90 (figure 1.3a) and the estimated two-component mixture distributions seem to reasonably model the underlying data generating process regardless of dose response summary metric used (figure 1.3b, dashed lines represent $\geq$ 50% probability of membership to the orange sensitive component). Similar findings are supported for broad effect drugs such as topotecan (figure 1.3c).

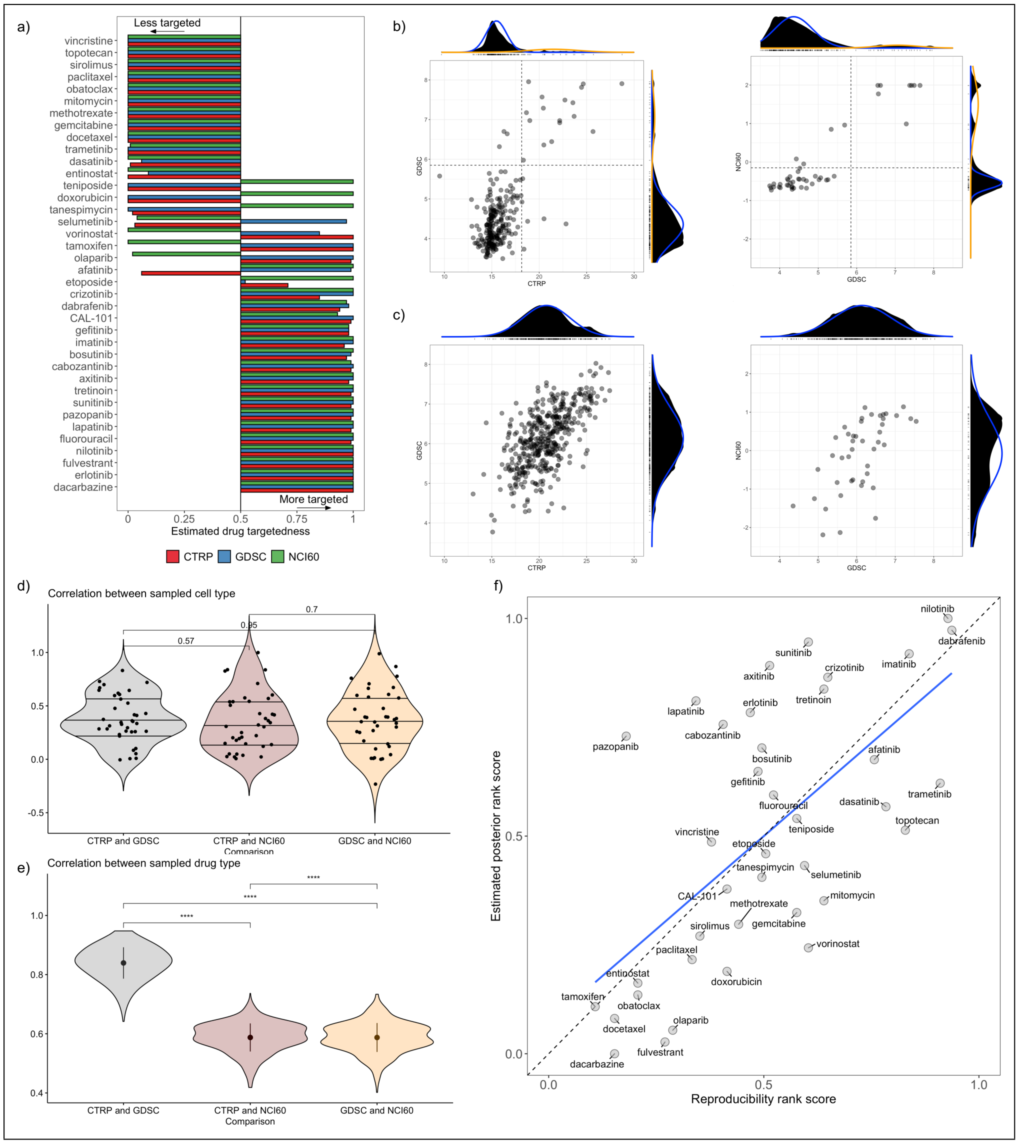


Figure 1.3: a) Estimated posterior probability of drug targetedness for CTRP (red), GDSC (blue) and NCI-60 (green) for drugs tested in common between the three studies. Solid line represents an estimated 50% probability of the drug being targeted. 29 of the 38 drugs exhibit strong agreement between drug type. **b)** Estimated posterior component distributions for the targeted drug dabrafenib. Dashed lines represent equal probability over membership in either component. **c)** Estimated posterior distributions for the broad effect drug topotecan. Stong agreement is evident. **d)** Mean estimated agreement per drug calculated over 1000 MC samplings of the estimated posterior distributions indicates no significant differences in agreement between the three datasets. **e)** Distributions of drug type correlation from samplings of the estimated posterior distributions indicates significantly higher agreement between drug types for the CTRP-GDSC comparison. **f)** Reproducibility rank scores (X-axis, see[107] for a description) are highly correlated with rank scores based on partial correlations of the estimated posterior sensitivities while adjusting for dataset comparison. Many highly targeted drugs which exhibit moderate reproducibility based upon the X-axis measure attain higher levels of reproducibility upon explicitly modeling the two-component data generating process.

Mean estimated agreement, per drug and per pairwise comparison, was calculated based on 1000 MC samples drawn from the estimated posterior distributions for those cell lines present in both datasets considered in the comparison (figure 1.3d). All three datasets agree to a similar extent with one another, and overall these agreements are slightly improved relative to the PRISM dataset comparisons (figure 1.2b). It is important to note that the CTRP-GDSC comparison presented in figure 1.2 only considers cell lines present in all three datasets (CTRP, GDSC, PRISM) whereas the CTRP-GDSC comparison presented in figure 1.3d contains any cell lines present in then CTRP-GDSC intersection. The drug-type agreement is again higher between CTRP and GDSC than in comparisons between NCI-60 and either of the two (figure 1.2ae).

The reproducibility rank score introduced by Rajapakse et al. [107] is an average across the three pairwise comparisons of the ranks of the q-values obtained by calculating the Pearson correlation per drug between sensitivity measures for cell lines present in each pairwise comparison (figure 1.3f X-axis). This score was used to aggregate correlations across pairwise comparison, thus finding that strong activity correlations are not limited to just select targeted drugs, such as protein kinase inhibitors, but that some broad effect drug sensitivities (such as topotecan) are reproducible as well. We calculated a comparable rank score based on the estimated posterior sensitivities by calculating the partial correlation between all sensitivities conditional on dataset comparison. We then ordered these correlations and assigned ranks (figure 1.3f Y-axis). The Pearson correlation between the two rank scores is strikingly high ($\rho = 0.65, p < 2e^{-5}$) . Additionally, broad effect drugs found to be reproducible by Rajapakse et al. [107], such as topotecan and trametinib, still attain moderate-to-high levels of reproducibility under the estimated posterior rank score however the fifteen most reproducible drugs are all drugs with very high estimated posterior targetedness. This latter result suggests highly targeted compounds represent the most reproducible class of drug but broad effect drugs may still be reproducible across dataset.

Model-based classifications improve downstream analysis

CCLE and GDSC contain independently-measured molecular covariate information (RNA-seq, mutation status, and copy-number variants) for each cell which may be used to ascertain biomarkers significantly associated with drug response. We used this information to perform a differential analysis relating the RNA-seq expression levels with the drug sensitivity metric (AAC) for each of the 15 drugs tested in common between the studies. LIMMA[110] was used to obtain test statistics estimating the association between each marker with estimated posterior drug sensitivity. In the case of drugs estimated to be more targeted in nature we rounded estimated sensitivities such that cells with an estimated posterior probability of belonging to the sensitive component greater than 0.5 were denoted as sensitive ($Y_i = 1$), else resistant ($Y_i = 0$, see Supplementary Methods for more information). We then computed the Pearson correlation between the estimated effect sizes output from LIMMA across the two studies, resulting in a single measure of correlation per drug (figure 1.4a Y-axis). We compared the correlations computed using the estimated posterior sensitivities with the correlations attained using the original AAC values (raw AAC for broad effect drugs as described by[60] or binarized AAC using a threshold of 0.2 as described by[113]. We then repeated this procedure for both the mutation and the CNV feature sets.

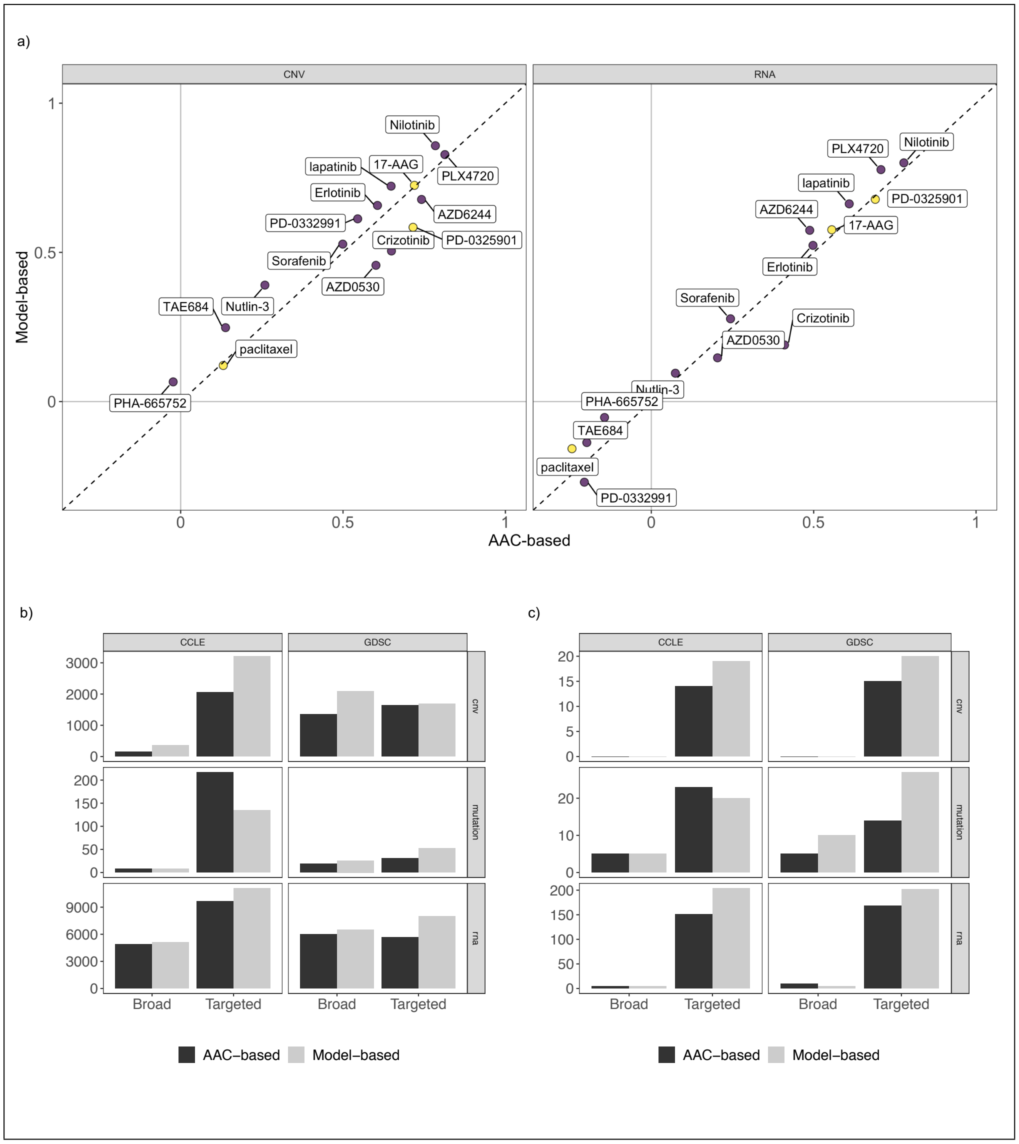


Figure 1.4: a) Pearson correlation between effect size estimates of differential gene expression (RNA) and copy number variant (CNV) for all drugs tested in common between CCLE and GDSC. Effect sizes are estimated using LIMMA with the outcome variable representing raw AAC (continuous in the case of broad effect drugs, binarized using the AAC> 0.2 threshold for targeted drugs, X-axis) or the model-based estimated posterior probability of sensitivity. Coloring represents broad effect drug type (yellow) or targeted (purple). 11 of the 15 drugs exhibit increased correlation due to the model-based metric. **b)** Counts of significant feature associations (Benjamini-Hochberg adjusted P-value $< .05$) between different feature sets. Counts are summed over all drugs falling within that drug-type (broad effect or targeted), and indicate a similar number of significant associations between method (AAc-based versus model-based). **c)** Counts of significant feature associations, as in **b**, however restricted to only those biomarkers with previously-annotated associations for each drug. In nearly all feature types, the model-based outcome measure of sensitivity results in a greater number of significant associations with *known* biomarkers, suggesting the model-based sensitivity metric sharpens biological signal relative to the raw AAC-based outcome measures.

The Pearson correlations calculated between the estimated test statistics are increased for 11 of 15 drugs when assessing RNA association using the model-based measure of drug sensitivity (estimated posterior probability targeted, figure 1.4a right-hand panel, Y-axis) versus using the raw AAC (figure 1.4a X-axis yellow points) or a binarized AAC with a threshold of AAC>0.2 (figure 1.4a X-axis purple points). 11 drugs also exhibit improved consistency for copy number variant associations (figure 1.4a left-hand panel), and only crizotinib and AZD0530 show decreased consistency for both RNA and CNV associations. We investigate crizotinib biomarkers more fully in later results.

P-values were obtained from LIMMA and features with estimated false discovery rate (FDR) less than 0.05 were denoted as significant (figure 1.4b). Benjamini-Hochberg was used to control for multiple hypothesis testing[13]. Counts of significant biomarkers by drug type (targeted or broad effect, bar plot shading), by feature type (CNV, RNA-seq, mutation; rows in the faceting), and by study (CCLE and GDSC, columns in the faceting) are shown. Notably the counts of significant biomarkers are not universally larger when using the estimated posterior sensitivities as the outcome measure compared with using the AAC-based values (light versus dark colored bars, respectively). We next used curations from MolecularMatch Trials, Cancer Genome Interpreter, and Clinical Interpretation of Variants in Cancer to annotate *known* biomarkers[31,135,41]; namely those features with previously found associations with each drug under consideration (figure 1.4c). Interestingly, biomarkers with significant associations found when using the estimated posterior sensitivities as the outcome variable tend to include more of the *known* biomarkers than when using the original AAC as the outcome, suggesting the model-based sensitivity metrics capture at least as much biological signal as the raw AAC-based outcome measures.

Cell matrix pathways highlight molecular complexity driving crizotinib sensitivity

Many pharmacogenomic datasets are publicly available and contain varying degrees of overlap between cells and drugs tested in common (figure S9). Cells tested in more than one experiment with the same targeted drug and within the same concentration ranges effectively define replicate samplings, with the true underlying cell sensitivity (Z_{ij}) a random variable for which we now have multiple measurements. Using the model formulation previously described, we again estimate the latent sensitivity for all cells and all drugs per study, regardless of the number of datasets it may be available in, but now relax the notation such that a common latent variable is estimated across all studies for those cell/drug pairs tested in more than one study. Distributions are fit within study to account for experiment effects (see Methods section for additional details). AACs are computed over common dosage concentrations.

We highlight this approach by focusing on a tyrosine kinase inhibitor Crizotinib, which was the only drug with high estimated posterior targetedness and was also present in each of the five datasets (CCLE, GDSC1000, FIMM, gCSI, CTRPv2, figure 1.5a estimated mixture component distributions, crizotinib estimated posterior probability targeted $>$0.9). We only consider the five datasets with AACs calculated over common dosage ranges as provided by PharmacoGx[131]. The relationship between AAC and the model-estimated posterior sensitivity for cell lines which were present in only a single dataset follow a sigmoidal curve (figure 1.5b) whereas the impact of estimating posterior sensitivity jointly for cell lines present in multiple datasets is to pull values away from that curve to a unifying value per cell line across dataset (figure 1.5b Y-axis). Gene fusions (EML4-ALK or NPM1-ALK) were curated from[97].

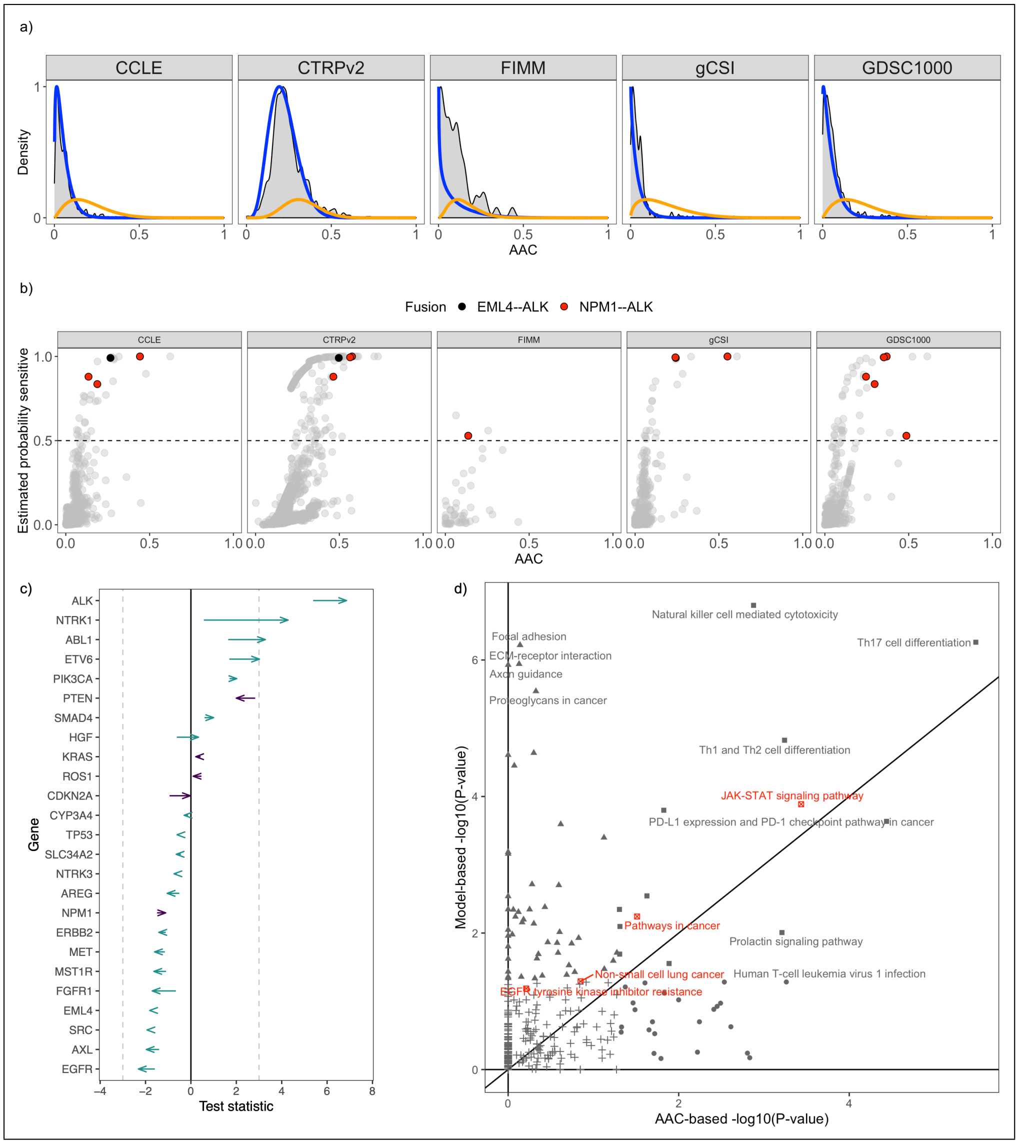


Figure 1.5: a) Estimated component distributions (resistant:blue, sensitive:orange) fit using the joint model for the drug Crizotinib. **b)** Estimated posterior cell sensitivity (Y-axis) given the observed AAC (X-axis) for all cell lines tested. Color denotes whether the cell harbors a known gene fusion through which Crizotinib targets for therpeutic intervention (EML4-ALK: black, NPM1-ALK: red, None:grey). Fusions are curated from [97] supplementary table 2. **c)** Estimated test statistic (X-axis) measuring the association between gene expression for genes with previously-annotated association with Crizotinib sensitivity. Drug sensitivity is measured with the raw AAC (binarized at AAC> 0.2, tail of arrow) and using the estimated posterior probability of sensitivity from the joint model (estimated posterior > 0.5, head of arrow). Coloring indicates whether the test statistics are more extreme using the model fit (lighter blue) or more extreme using the raw AAC (darker blue). Dashed lines represent a statistical significance threshold of $p < .05$. **d)** $-\log_{10}$(P-value) from performing a KEGG pathway analysis using either binarized raw AAC values (X-axis) or the model-based estimated posterior probability of sensitivity (Y-axis). Shape indicates statistical significance at the $p < .05$ level and red coloring indicates pathways with mechanistic KEGG annotations to Crizotinib sensitivity.

Crizotinib is a small-molecule inhibitor of receptor tyrosine kinases and has proven an effective therapy for inducing remission in cases of anaplastic lymphoma kinase (ALK) rearranged non-small cell lung carcinomas (NSCLC)[95,114]. ALK-rearranged NSCLCs represent only a small subset of total NSCLC cases yet the consistent efficacy documented in ALK-rearranged NSCLC patients led to FDA approval of crizotinib as the first ever ALK inhibitor in 2011[64]. The EML4-ALK gene fusion is the most common ALK rearrangement in NSCLC and results in constitutive kinase activity[52,115,142,51]. A similar increase in the oncogenic potential of ALK has been identified for the NPM1-ALK fusion[35]. Evidence supports the high dependence of ALK-driven lung cancers on this ALK fusion, however resistance may be acquired during treatment, perhaps due to a loss of epithelial differentiation[132,143].

Cell lines harboring either a NPM1-ALK or EML4-ALK gene fusion exhibit at least 50% estimated posterior sensitivity to Crizotinib (figure 1.5b). In two datasets, CCLE and FIMM, using the naïve cutoff of AAC>0.2 to denote sensitive cells would result in calling three of the cell lines with the NPM1-ALK fusion as resistant. On the other hand, the joint posterior estimated probabilities of sensitivity are greater than 0.5 for these same cell lines. All other cell lines harboring one of the two fusions exhibit large estimated sensitivity under the model.

We next assessed how the associations with known biomarkers changed due to utilizing the new sensitivity metric. Biomarkers with previously-documented associations with crizotinib sensitivity were curated from MolecularMatch Trials, Cancer Genome Interpreter, and Clinical Interpretation of Variants in Cancer[31,135,41]. The estimated test statistics from performing LIMMA for 20 of these 25 known biomarkers increase in statistical significance when measuring the association between gene expression variability and drug sensitivity as defined with the multi-level model or the AAC raw value (binarized at AAC>0.2 for targeted drugs); the tail of the arrow denotes the estimated test statistic had sensitivity been measured by AAC whereas the head of the arrow denotes the test statistic based upon the modeling procedure (figure 1.5c). The estimated test statistic increases in absolute magnitude for 20 of the 25 biomarkers, with multiple-testing adjusted statistical significance of 0.05 attained for three of these biomarkers (NTRK1, ABL1, EVT6) which would not have been realized under the AAC-based approach. Regardless of sensitivity metric, the largest estimated association is with ALK expression, as one might expect.

A KEGG pathway analysis (figure 1.5d) was performed to identify up- and down-regulated pathways significantly associated with crizotinib sensitivity[63]. Target KEGG pathways are denoted in red text (e.g. JAK-STAT signaling pathway) and for each of the four pathways an increase in statistical significance due to the model-based sensitivity metric is observed ($-\log_{10}$(P-value) based on AAC-based sensitivity: X-axis, model-based: Y-axis). T helper cells have a relatively well characterized role in NSCLC tumor development and progression[90], and pathways related to T helper cells are significantly enriched regardless of sensitivity metric. More importantly is the cluster of down-regulated cell matrix pathways, pathways which would not be found to be significantly associated with response under the alternative approach. Of these, focal adhesion kinase 1 has recently been identified as a relevant target for inhibiting neurofibromatosis type 2-associated malignancies through the repurposing of crizotinib[138]. Somatic mutations in axon guidance pathway genes have also been implicated for their carcinogenic potential in pancreatic cancer[14], with crizotinib found effective for one pancreatic cancer case harboring a DCTN1-ALK fusion[121]. Additionally, Wei et al.[143] find mutations in genes associated with epithelial-mesenchymal transition (EMT)-related pathways, such as proteoglycans in cancer and ECM-receptor interaction, to confer a mechanism for crizotinib resistance[143]. Crizotinib is a multi-targeted tyrosine kinase inhibitor, originally developed as a MET inhibitor prior to clinical trials noting benefit for ALK-positive NSCLC cases[95]. The mechanisms for developing resistance to crizotinib as well as potential for drug repurposing are little understood and the KEGG pathways associated with the model-based sensitivity metric capture this molecular complexity driving drug response.

1.3 Discussion

Despite early concerns of discordance, pharmacogenomic datasets are largely in agreement once conditioning upon drug type. The difficulty, however, lies in the classification of drugs into effect type and the subsequent binarization of cell lines into sensitive and resistant components for those drugs classified as targeted. While manual annotation with domain expertise is indeed possible, and perhaps optimal, the increasingly large number of drugs tested and the rigorous determination of thresholds for binarization of cell lines makes this a difficult and infeasible task. Indeed the quality of any downstream analyses, such as that of learning biomarkers associated with drug response, are entirely dependent on the quality of the dose curve summaries used as the outcome in the analysis.

We extend and improve upon past methods for modeling dose curve summary measures by proposing a multi-level mixture model in which the first level estimates drug type, broad effect or sensitive, with two-component mixture distributions and the second models the response of the cell line conditioned on the drug type modeled in the first level. Specifically, two-component mixture distributions are estimated only for targeted drugs in order to classify cells into sensitive or resistant, whereas single-component distributions are estimated for broad effect drugs. Fitting this model permits us to fully described all cell-type/drug combinations without manual annotations and supports the notion that some cell type distributions are continuous while others are binary. A further advantage of our approach is that the probabilistic approach permits us to combine data from different studies in a statistically rigorous way. We note that several other large-scale pharmacogenomic datasets are publicly-available in addition to CCLE and GDSC[70,145,12,24]. We fit our model to all 1,381 distinct cell lines and 733 distinct drugs available in five cell line viability datasets and find reasonable agreement across common cell lines and drugs. We then use the results of our fitted model to detect biomarkers that are not found with previous approaches. Specifically, a comparison of biomarkers associated with the ALK-inhibitor crizotinib, a highly-targeted drug used for treating non small cell lung carcinoma (NSCLC), identifies significantly down-regulated cell matrix pathways as drivers of dose sensitivity.

Developments in pharmacogenomics continue to progress at a rapid pace since the pioneering NCI-60 panel. Advances include applications to biotherapeutics,[87], testing of 3D patient-derived organoid models[91,39], and in how to bridge the gap between laboratory findings and clinical application[62,74]. Additionally, a greater understanding of technical artifacts which confound the analysis, such as the cell line growth rate, has elucidated how intra- and inter-laboratory factors affect reproducibility[93] and led to the development of dose curve summary statistics which explicitly adjust for the experimental biases[44,45? ,42]. While our multi-level model is adaptable to these new measures and *in vitro* models, the assumption of two classes of effect type may no longer be valid as new classes of drug molecules are added. Determining the suitable number of first-level mixture components requires domain expertise and may vary across datasets. Furthermore, there is no rule to determine which datasets to include in the analysis. Different studies test drugs at different dosages and over different concentration ranges, with some collecting measurements on dose curve confounders and others not; Failing to discard a highly disconcordant dataset or compute sensitivities over common ranges will undoubtedly affect downstream analyses. While our interpretation of cell line sensitivity as the estimated probability of membership to the sensitive mixture component is agnostic of dose curve summary measure, it remains unclear if comparisons should even be made between datasets which report different dose curve summaries, let alone combined in a joint analysis. Interesting avenues for follow up include measuring the effect of the model-based sensitivity estimates on predictive performance and feature selection[5,75] and how agreements calculated by statistical frameworks such as the Alternating Imputation and Correction Method (AICM)[53] or copulas[106] change.

Implementation

Posterior sensitivities may be estimated using an R package[104] available at `https://github.com/mPloenzke/PharmacoMixtuR`. Code to reproduce the analysis and figures herein is available at `https://github.com/mPloenzke/PharmacoMixtuR_scripts`. A shiny application[19] is available at `https://mploenzke.shinyapps.io/correlation_app/` which may be used to view the relationship between various correlation measures when the random variables being correlated arise from bivariate two-component mixture distributions.

1.4 Methods

Measuring association

A fundamental assumption underlying Pearson correlation is a linear relationship between the two continuous random variables, X_1 and X_2, with the conditional distribution $P(X_2 \mid X_1 = x_1)$ bivariate normal (supplementary figure S2a); indeed it is this assumption of joint normality which underpins inferential procedures in the ordinary least squares (OLS) estimator. For instance, consider a normal random variable $Y \sim N(\mu_Y, \sigma_Y^2)$. If $X_{1,i} = Y_i + \varepsilon_i$ and $X_{2,i} = Y_i + \varepsilon_i$ with $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, then $P(X_2 \mid X_1 = x_1) \sim N(\mu_Y, \sigma_Y^2 + \sigma_\varepsilon^2)$ and the Pearson correlation is simply the signal-to-total variation ratio $\frac{\sigma_Y^2}{\sigma_Y^2+\sigma_\varepsilon^2}$. This is the optimal measure in the class of linear unbiased estimators[18] and thus in such cases of bivariate normality Pearson correlation is the optimal measure to assess agreement. The Spearman correlation will yield predictably similar results in this case regardless of the signal-to-noise ratio due to a valid monotonicity assumption.

On the other hand, consider the case when the random variables X_1 and X_2 do not follow a single continuous distribution but instead a two-component mixture-of-normals distribution with $Z \sim \text{Bernoulli}(p)$ specifying the second mixture membership probability (supplementary figure S2c; mixture distribution of a single normal distribution (black) mixed with a small, second component (red). Four signal-to-noise regimes pictured with the signal measured by the distance between the generating mixture component means and the noise as the sum of the component variances). Under such a formulation the conditional distributions $P(X_1 \mid Z = z)$ and $P(X_2 \mid Z = z)$ are themselves normal however the distribution of $P(X_2 \mid X_1 = x_1)$ is no longer bivariate normal and is confounded by Z. Thus the bivariate normality assumption is no longer valid and a measure of association which takes into account the underlying binary nature of the data is warranted (supplementary figure S2bd; simply thresholding values at one-half the signal (mean of red mixture) suffices for high agreement even under low signal-to-noise regimes. Pearson correlation (derived analytically) and Spearman correlation (smoothed estimate across five simulation repetitions) remain predictably low). The use of a 2×2 contingency table (or logistic regression model) follows to highlight that the correct measure of association in this case is in terms of an odds ratio (perfect agreement is expressed as an infinite odds ratio) or the Matthew's correlation coefficient (i.e. an equivalent to Pearson correlation when the underlying variables follow binomial distributions as opposed to gaussians). Further, given the binarization, the interpretation of such a measure as the probability the latent random variables are equal (i.e. both X_1 and X_2 arose from the same mixture component) is a more intuitive notion of association than the alternative Pearson measure expressing the expected standardized change in X_2 given a change in X_1, which is influenced by the signal of the second mixture component (supplementary figure S2b; slope of blue lines is non-zero despite zero intra-cluster covariance). Subsequent reporting of Spearman or Pearson correlations will result in different assessments of agreement based on the signal-to-noise regime because the monotonicity assumption is invalid (supplementary figure S2d).

Different drugs exhibit larger or smaller mean sensitivity and spread due to their mode of action, and thus it follows to report differing measures of drug sensitivity concordance based upon the nature of the drug under consideration (supplementary figure S3a; y-axis ordering by median AAC places targeted drugs near the bottom and broad effect drugs near the top). Indeed one expects few cells to respond to targeted, targeted drugs and thus a binarization of the continuous AAC value into sensitive or resistant calls follows logically (supplementary figure S3b; dashed lines indicate sensitivity threshold: AAC$>$ 0.2). On the other hand, cells treated by broadly-active drugs, such as cytotoxic drugs, all tend to exhibit some degree of response (supplementary figure S3b; broad effect drugs paclitaxel, 17-AAG, and PD-0325901), suggesting a continuous measure of cell sensitivity and within-dataset agreement (Pearson correlation) be reported. The difficulty in such an approach, however, lies in determining both the drug-specific thresholds to discretize the cell sensitivity measures (sensitive versus resistant) as well as the method for classifying drugs into effect types (broad versus targeted effect), both of which affect downstream biomarker inference.

Model formulations

To better quantify the effect of drugs on cell type we developed a two-level mixture model in which the first level accounts for drug targetedness, broad in effect or targeted, and the second level models cell sensitivity conditioned upon drug type. We modeled targeted drug sensitivity with a two-component mixture distribution. If we condition on the drug being targeted, the effect of the drug can be quantified with the posterior probabilities of being sensitive. If we condition on the drug being broad effect, the effect can be quantified with continuous measures such as the Z-score or an empirical cumulative distribution function. Note that this model can be applied to each dataset separately or jointly to all datasets. Applying the model *within-dataset* is useful for validation by calculating pairwise study agreements of drugs and cell lines. Applying the model *across-datasets* is useful to estimate a unified measure of cell sensitivity for each cell-drug combination. Below we detail how the modeling procedure.

For any cell lines i, drug j, and pharmacogenomic dataset k we denote the observed dose curve summary statistic with $Y_{i,j,k}$. The dose curve summary measure is assumed to be the area-above-the-curve (AAC) normalized to a range of 0-1 in the following formulation. In case studies in the text in which AAC is not available we model IC50 as the dose curve summary and use normal component distributions as opposed to the second-level beta distributions described below. We opt for AAC based on the statistical assessment of dose curve summary measures provided by Jang et al.[60]. Three reported advantages of AAC are 1) AAC summarizes the entirety of the dose response curve as opposed to at a single dosage concentration, 2) AAC is finite and bounded for all tested observations, and 3) ACC capture differences in dose response curves which EC50 and IC50 do not[60,47]. When growth rate information is available, the growth-rate adjusted AAC proposed by Hafner et al.[44] may be normalized to a range of 0-1 and modeled similarly to the AAC.

Level 1: Modeling drug type

We define the dichotomous latent variable W_j to be 0 if the drug has a broad effect and 1 if it is a targeted drug. We define $\rho \equiv \Pr(W_j = 1)$ as the proportion of drugs that are targeted. For targeted drugs, when $W_j = 1$, we define another latent variable $Z_{i,j}$ to be 1 if cell line i is sensitive to drug j and 0 otherwise. We next define

$$\pi_j \equiv \Pr(Z_{i,j} = 1)$$

as the proportion of cell lines sensitive to drug j and assume that for targeted drugs π_j has distribution $p^t(\pi_j) \equiv p(\pi_j \mid W_j = 1)$ which we assume to be beta with parameters a^t and b^t. The values of π_j for true targeted drugs will be low given the small proportion of cells which are targeted by the drug. Values of π_j for broad effect drugs modeled as targeted drugs will be much larger.

Because dichotomizing is not appropriate when $W_j = 0$, for convenience, we assume $\pi_j = 1$ and $Z_{i,j} = 1$ for all i, j when $W_j = 0$.

Level 2: Modeling cell type

We then model the bimodal behaviour of the observed summaries for targeted drugs (figure 1.1b; targeted drug crizotinib) with the following mixture distribution

$$p(Y_{i,j,k} \mid Z_{i,j} = z_{i,j}, W_j = 1) = z_{i,j} p^s_{j,k}(Y_{i,j,k}) + (1 - z_{i,j}) p^r_{j,k}(Y_{i,j,k})$$

with $p^s_{j,k}$ the distribution density of the sensitive component for drug j in dataset k and $p^r_{j,k}$ the distribution density of the resistant component. We assume both p^s and p^r are beta distributions with with parameters $c^s_{j,k}$ and $d^s_{j,k}$, and $c^b_{j,k}$ and $d^b_{j,k}$, respectively. These component distributions are replaced with normal distributions when modeling IC50 or EC50 as the dose curve summary statistic.

Finally, we assume that for broad effect drugs, $Y_{i,j,k}$ follows a beta distribution $p^b_{j,k}$ with parameters $c^b_{j,k}$ and $d^b_{j,k}$. This distribution is also replaced with a normal distribution when modeling IC50 or EC50. We estimate ρ, $c^b_{j,k}$, $d^b_{j,k}$, $c^r_{j,k}$, $d^r_{j,k}$, $c^s_{j,k}$, $d^s_{j,k}$ using maximum likelihood estimation.

Model-based measures of drug targetedness and cell sensitivity

With these assumptions and estimates in place we can now compute quantities useful for downstream analyses. Specifically, for targeted drugs we can compute the posterior probability of cell line i being sensitive to drug j, $\Pr(Z_{i,j} = 1 \mid Y_{1,j,k}, \ldots, Y_{I,j,k}, W_j = 1)$. Note that we can compute this posterior probability for each study k and use this to assess agreement, or compute one quantity using data from all studies. For broad effect drugs we compute the cumulative probability

$$\Pr(Y_{i,j,k} \leq y \mid W_j = 0) = \int_0^y p^b_{j,k}(y \mid c^b_{j,k}, d^b_{j,k}) dy$$

as it will remove study effects such as the location shifts (supplementary figure S3a; paclitaxel and 17-AAG, supplementary figure 1.5a).

For the above calculations we need to decide if a drug is targeted or not. To guide this decision we compute the posterior probability of a drug being targeted given the observed data. Specifically, for each drug j and study k we can compute the posterior distribution $\Pr(W_j = 1 \mid Y_{1,j,k}, \ldots, Y_{I,j,k})$ with I the total number of cell lines. We use this estimate to measure drug targetedness (figure 1.1c).

Data-driven model initialization

The W_j are referred to as drug types and the $Z_{i,j}$ as cell types. We initialize the first-level of the model using the median and MAD (median absolute deviation) (supplementary figure S6a,b; CCLE and GDSC datasets, respectively). Drugs are initialized to targeted (blue points) if their median AAC is below the overall median AAC within study (vertical dashed line), and their MAD is below the overall study MAD (horizontal dashed line). All drugs and cell lines in a given dataset are included during this procedure as well as during model fitting. Drugs not satisfying both conditions are initialized to broad effect (red points). In practice we find the 60th percentile as opposed to the median results in a slightly more concordant initialization when comparing to the drug types described in[113] (14 drug types concordant of the 15 tested in both CCLE and GDSC) and[60] (22 of 24 drug types concordant of the 24 tested in CCLE). We therefore report results based upon the 60th quantile. The empirical proportion of drugs initialized to the targeted component based upon this procedure is interpreted as the prior probability a drug is targeted.

Cell type is initialized by assigning cells to sensitive if the observed AAC value is greater than the median AAC value across dataset, else resistant. The proportion of cells initialized to sensitive, π_j, is computed and used as the empirical prior probability the cell is sensitive to drug j.

We observe a bimodal distribution for π_j initialized in this manner for the CCLE and GDSC datasets (supplementary figure S6c). One component of the distributions have a mode around 0.15

implying around 15% of cell lines are sensitive to drugs of this type, namely targeted compounds. The second component distribution is flatter with a mode near 1, following the interpretation that cell lines treated by broad effect drugs exhibit a continuous degree of sensitivity. Recall we assume $\pi_j = 1$ for all broad effect drugs and the π_j depicted in supplementary figure S6c represent the π_j if the drug were to be modeled as a targeted drug type.

Estimating agreement

We obtain estimates for $Z_{i,j}$ and W_j along with the distributional parameters using the expectation-maximization algorithm and then use these quantities to compute the posterior agreement between any two studies. There are two quantities we are interested in: 1) the posterior cell agreement (is the same cell sensitive to the same drug in both experiments?), 2) the posterior drug-type agreement (is the same drug estimated to be the same drug type in both experiments?). We calculate these agreements in two manners: 1) maximum *a posteriori* (MAP) estimates, 2) Monte Carlo sampling.

The MAP estimate of drug type agreement is calculated by rounding the estimated posterior probability of drug targetedness for drug *j* in study *k* and calculating the correlation between the cells of the contingency table summing the counts of broad drugs ($W_{j,(k)} = 0, W_{j,(l)} = 0$) and targeted drugs ($W_{j,(k)} = 1, W_{j,(l)} = 1$) for the two datasets *k* and *l*. Note we add the dataset subscripts, (k) and (l), to denote these values were estimated only using dataset *k* or *l*, respectively. To generate confidence intervals associated with drug type agreement, we perform samplings from the estimated posterior drug type distributions and during each sampling calculate the correlation between cells of the resultant contingency table.

Cell agreement is always measured with Pearson correlation however the underlying data distribution differs conditional upon drug type. For targeted drug *j* with fits obtained only using study *k* the posterior distribution is

$$p(Z_{i,j,(k)} \mid Y^k_{i,j,k} = y, W_{j,(k)} = 1) \sim \text{Bernoulli}(\pi_{j,(k)})$$

Then defining $Z_{\cdot,j,(k)} \equiv \sum_i^{n_{j,k}} Z_{i,j,(k)}$ it follows that $Z_{\cdot,j,(k)} \sim \text{Binomial}(n, \pi_{j,(k)})$. Calculating the Pearson correlation for drug j in datasets k and l then amounts to computing

$$r = \frac{\text{Cov}(Z_{\cdot,j,(k)}, Z_{\cdot,j,(l)})}{\sqrt{\text{Var}(Z_{\cdot,j,(k)})}\sqrt{\text{Var}(Z_{\cdot,j,(l)})}} = \frac{E(Z_{\cdot,j,(k)}Z_{\cdot,j,(l)}) - E(Z_{\cdot,j,(k)})E(Z_{\cdot,j,(l)})}{\sqrt{\text{Var}(Z_{\cdot,j,(k)})}\sqrt{\text{Var}(Z_{\cdot,j,(l)})}}$$

In the case of two random variables following binomial distributions Pearson correlation may be calculated from a two-by-two contingency table with cells tallying the counts of the binomial realizations. That is for $W_1 \sim \text{Bin}(n, p_1)$ and $W_2 \sim \text{Bin}(n, p_2)$

$$r = \frac{\sum_{i=1}^n w_{i,1}w_{i,2} - n\bar{w_1}\bar{w_2}}{\sqrt{\bar{w_1}(1-\bar{w_1})}\sqrt{\bar{w_2}(1-\bar{w_2})}}$$

This metric is often referred to as Matthew's correlation coefficient[89] and is interpreted in terms of prediction quality (i.e. $\sum_{i=1}^n w_{i,1}w_{i,2}$ represents the count of true positives). Its utility has been highlighted when assessing pharmacogenomic agreement for targeted drugs as described in[47,113]. In these cases, MCC is calculated by binarizing AAC using a threshold such as AAC > 0.2, and has improved many of the agreements initially reported for targeted drugs. A similar approach is used to compute the MAP agreement for targeted drugs: the cell probability of sensitive is binarized by rounding the estimated posterior probabilities. To obtain the confidence intervals, we perform MC sampling with each iteration consisting of a drug type sampling of $W_{j,(k)}$ followed by a cell type sampling from $P(Z_{i,j,(k)} \mid Y_{i,j,k} = y, W_{j,(k)} = 1)$ for all $W_{j,(k)} = 1$. The process is repeated for $W_{j,(l)}$ and $P(Z_{i,j,(l)}$. Correlation is computed based on the contingency table generated from these counts. On the other hand, when the sampled drug type is broad effect, $W_{j,(k)} = 0$, the posterior distribution is $P(Y_{i,j,k} \mid W_{j,(k)} = 0) \sim \text{Beta}(c^b_{j,k}, d^b_{j,k})$ and we calculate Pearson correlation under a normality approximation. Mean correlation and confidence intervals are calculated over all MC samplings.

Cross-dataset model

The cross-dataset model combines datasets using several sources of sensitivity data to provide one estimate of cell sensitivity to treatment regardless of the number of times it has been tested (i.e. the number of datasets which contain this specific cell-drug pair). This is achieved by computing the expectation across all K studies during the E step of the EM algorithm. To control for batch effects we allow for the maximization step to proceed within study such that posterior sensitivity distributions are study specific (figure 1.5a). Similarly, the latent variable denoting cell i is sensitive to drug j ($Z_{i,j}$) is shared across all datasets resulting in a single estimate of cell sensitivity (figure 1.5b). Notably, estimates for cell lines tested in a single study are unaffected by cell lines tested in different studies, however estimates for cell lines tested in multiple studies are pulled towards a cell-specific latent variable. First-level drug type initialization is performed using the median and 60th percentile absolute deviation however these computations are performed by pooling all available data, regardless of study, and making global drug assignments.

Likelihood specification

Denote K the set of experiments under consideration, J the full set of drugs tested in all studies $k \in K$, and $I_{j,k}$ the set of cell lines tested with drug j in study k. The full Bayesian hierarchical model contains the following stages:

$$\begin{aligned}
&\text{Stage I: } p(Y_{i,j,k} \mid Z_{i,j} = 1, \pi_j = \pi, W_j = 1, \rho) \sim \text{Beta}(c^t_{j,k}, d^t_{j,k}) \\
&\qquad p(Y_{i,j,k} \mid Z_{i,j} = 0, \pi_j = \pi, W_j = 1, \rho) \sim \text{Beta}(c^r_{j,k}, d^r_{j,k}) \\
&\qquad p(Y_{i,j,k} \mid W_j = 1, \rho) \sim \text{Beta}(c^b_{j,k}, d^b_{j,k}) \\
&\text{Stage II: } p(Z_{i,j} \mid \pi_j = \pi, W_j = 1, \rho) \sim \text{Bernoulli}(\pi_j) \\
&\text{Stage III: } p(\pi_j \mid W_j = 1, \rho) \sim \text{Beta}(a^t, b^t) \\
&\text{Stage IV: } p(W_j \mid \rho) \sim \text{Bernoulli}(\rho)
\end{aligned}$$

2

Interpretable convolution methods for learning genomic sequence motifs

First-layer filters employed in convolutional neural networks tend to learn, or extract, spatial features from the data. Within their application to genomic sequence data, these learned features are often visualized and interpreted by converting them to sequence logos; an information-based representation of the consensus nucleotide motif. The process to obtain such motifs, however, is done through post-training procedures which often discard the filter weights themselves and instead rely upon finding those sequences maximally correlated with the given filter. Moreover, the filters collectively learn motifs with high redundancy, often simply shifted representations of the same sequence. We propose a schema to learn sequence motifs directly through weight constraints and transformations such that the individual weights comprising the filter are directly interpretable as either position weight matrices (PWMs) or information gain matrices (IGMs). We additionally leverage regularization to encourage learning highly-representative motifs with low inter-filter redundancy. Through learning PWMs and IGMs directly we present preliminary results showcasing how our method is capable of incorporating previously-annotated database motifs along with learning motifs *de novo* and then outline a pipeline for how these tools may be used jointly in a data application.

2.1 Introduction

Applications of deep learning methods have become ubiquitous over recent years due primarily to excellent predictive accuracy and user-friendly implementations. One such application has been to nucleotide sequence data, namely data arising in the field of genomics, in which the convolutional neural network (CNN) has enjoyed particular success. The convolutional layers composing a CNN work by extracting and scoring local patches of the input data by computing the cross-correlation between all nucleotide subsequences in the observation and each filter. These feature scores are then passed through any number of subsequent weightings (so-called dense or fully-connected layers) and used to output a final predictive value or values, as in the case of a multi-dimensional output. For example, one of the earliest CNNs trained on genomic data, DeepSea, predicted with high accuracy a 919-dimensional output array with each entry representing the presence/absence of a specific chromatin feature[154]. DeepBind, developed near the same time as DeepSea, further demonstrated the utility of training CNNs on genomic data by showcasing how the first-layer convolutional filters tend to learn relevant sequence *motifs*[6]. This latter finding highlighted, within the application to genomic data, the potential for illuminating the black box that deep models are typically considered; namely it sparked interest in developing computational methods for both incorporating known biological structure into the models[125,4,40,82] as well as interpreting the learned model knowledge[78,123,94].

Much progress has been made to improve predictive accuracy since these pioneering manuscripts however the process proposed by[6] to infer sequence motifs from convolutional filters remains largely unchanged. Specifically, each trained filter is convolved over input test set observations to produce a vector of activation values per filter per observation. High scoring activation values above some threshold are identified and the subsequence giving rise to each value is extracted. All extracted subsequences are stacked together per filter and used to compute a *position frequency matrix* (PFM). The PFM for filter j is a $4 \times L_j$ matrix in which the rows represent nucleotide (A, C, G, T) and columns represent position. L_j is generally on the order of 8-18bp. The columns may be subsequently normalized by their sum to yield a *position probability matrix* (PPM), and then converted into a *position weight matrix* (PWM) by computing, for each element ω_n in the PPM, $log_2(\omega_n) - log_2(b_n)$ where b_n represents the background probability for the nucleotide n. The PWM is often visualized as the so-called sequence logo[116], which is computed by multiplying each entry in the PPM by the column-wise sum of the expected self-information gain (i.e. the Hadamard product of the PPM and PWM). Sequence-logo motifs constructed and visualized in this manner are shown in the bottom rows of Fig. 2.1A-D. We refer to the matrix of values denoting the heights of the nucleotides as the *information gain matrix* (IGM) of the PWM. A worked example converting a PFM to an IGM is provided in the supplementary materials.

Under the standard CNN framework there are no restrictions on the values of the weights comprising each convolutional filter to any range. In addition, the visualization procedure requires the analyst to select the threshold for extracting high activation values and is dependent upon the input observations themselves. As illustrated in panels A and B of Fig. 2.1 however, filters tend to learn redundant and highly-correlated features. Given the non-identifiability of the many weights in a typical CNN it is no wonder why the learned features are so correlated. One recently-proposed technique specifically developed to remedy the issue of correlated filters is to include all possible circular shifts of the filter when convolving it with the sequence [15]. Such a procedure increases the computation time and memory footprint as every convolutional operation now requires all possible spins of the filter and also requires input observations to interpret what has been learned. An alternative approach somewhat abandons the notion of filter interpretability as a sequence motif and instead takes a reverse approach via back-propagating the activation values, in effect addressing which sequences are most important for model classification for a given observation relative to some reference observation [123]. A third approach involves solving a reformulated optimization problem which seeks to find the single consensus sequence maximally activating the model [78,94]. None of these techniques simultaneously address the issues of redundancy and interpretability and, moreover, they require input observations or a post-hoc optimization procedure to infer the learned motif.

We propose to directly learn the sequence motifs such that interpretation of the convolutional filters is not reliant upon test set observations and the weights comprising the filter are directly interpretable as information gain or position weights, both of which may be easily visualized as sequence logos. We simultaneously address the issue of filter redundancy along with interpretability by incorporating weight constraints and regularization techniques. The weight constraints limit the range of the individual filter weights to restrict their values as to be directly interpretable as IGMs or PWMs while the regularization scheme encourages learning non-redundant motifs. Under such a framework previously-annotated database motifs either in the form of PWMs or IGMs, such as those available from JASPAR [67], may be used to initialize the convolutional filters in the model and subsequently held constant or updated during training. In section 2 we provide a brief introduction to the notation that will be used before detailing the method through a toy simulation. Section 3 showcases results for a more realistic simulation study as well as a data example using ChIP-seq peaks from the ENCODE Consortium[23]. Section 4 concludes with a brief discussion.

2.2 Materials and methods

Here we introduce notation and motivate the methodology through a simple simulation study. Consider a set of $N = 30K$ nucleotide sequences X_n where each sequence is of length $I_n = 200$ and composed of bases A, C, G, T drawn from some genome background probabilities (e.g. $[.3, .2, .2, .3]$). Randomly label half of the N sequences $Y_n = 1$ and the remaining $Y_n = 0$. For 98% of the positively-labeled cases, insert the sequence *GGGGGGG* at position $i \in 1 : I_n$ with i drawn uniformly at random. Conversely, insert the sequence *CCCCCCC* into the negatively-labeled cases with a similar uniform-location probability. We wish to train a binary classifier to predict the associated label $Y_n \in \{0, 1\}$ for a given sequence X_n. Of course, under this framework, perfect model accuracy would be obtained if an oracle could encode a binary feature denoting the presence/absence of *GGGGGGG* in each sequence (or similarly, the *CCCCCCC*). The *discovery* and *representation* of such a sequence, however, is what interests us. In other words, can our model directly learn the predictive subsequences without defining features *a priori* or requiring post-hoc interpretation procedures?

We utilize techniques from the deep learning literature to form such a feature finder. Specifically, we consider the convolution operator employed in convolutional deep neural network (CNN) architectures[80]. Consider a set of J convolutional filters where the j^{th} convolutional operator computes the inner product between a weight matrix Ω^j and an observation matrix X_n at each position sliding along user-specified dimensions. These convolutional filters, or patches, are particularly suited for learning local spatial relationships and when employed in deep learning are stacked together, potentially in the hundreds within a single layer, from which the activations produced by each convolutional filter are fed as input into subsequent deeper layers. In genomics applications each first-layer filter Ω^j is a $4 \times L_j$ matrix of weights $\omega^j_{k,l}$ for $k \in \{A, C, G, T\}$ and $l \in 1 : L_j$ convolved upon a one-hot encoding of the input sequence X_n. That is, each X_n is transformed from a nucleotide string of length I_n into a binary matrix of size $4 \times I_n$ with rows corresponding to each base and column i denoting the presence/absence of a nucleotide at position $i \in 1 : I_n$. Generally some sort of pooling operation is performed such that either the maximum (max-pooling) or average (average pooling) is selected within a small window, effectively reducing the parameter space and alleviating observational noise. We may write the model explicitly under the logistic link function with max-pooling performed over the entire input sequence as $\mathcal{G}(X_n) = P(Y_n = 1|X_n = x) = \frac{1}{1+e^{-\beta_0 - \sum_{j=1}^{J} \beta_1^j \zeta^j}}$ where $\zeta^j = \max\limits_{i \in 1:I_n} g(x_i * \Omega^j)$ indicates a max-pooled convolution operation. The convolutional operation itself is explicitly defined as $g(x_i * \Omega^j) = g(\sum_{l=1}^{L_j} \sum_{k \in \{A,C,G,T\}} \omega^j_{k,l} 1_{x_{i+l-1}=k} + \beta^j_0)$ with $g(\cdot)$ representing the sigmoidal activation function in our experiments. We note it is these Ω^j matrices which contain the weights which collectively capture the sequence motifs and are often visualized as sequence logos through the methods described in the introduction.

For this first simulation study, we arbitrarily set $J = 4$ and $L_j = 12$ for all j. We restrict our model to simply the maximum value of the convolutional operator per filter as this represents the best match, or similarity score, between the motif and the sequence, and is also readily interpretable while maintaining parsimony. The four activation values produced from the four filters may be thought of as constituting the input features, or design matrix, to a logistic regression formulation. Any additional predictors may of course be included as well. It should be noted that all subsequent formulations may be extended to any number of layers as the model described is equivalently a single convolutional layer and single connected layer CNN (i.e. a *shallow* CNN). This vanilla CNN provides the baseline comparison and is depicted in panel A of Fig. 2.1. Of particular importance is to note that the weights, $\omega^j_{k,l}$, (middle row of sequence logos within panel A) are unconstrained. Thus we also provide the sequence-logo motifs calculated as described in the introduction in the bottom row of panel A with the threshold set at $0.75 \times \max(\text{activation})$ per filter. The background nucleotide probabilities used when calculating, displaying, or learning any PWMS/IGMs presented herein are taken to be uniform (i.e. $b_k = .25, k \in \{A, C, G, T\}$). While our method and software implementation allow for non-uniform probabilities, motifs such as those downloaded from JASPAR[67] are generally calculated and visualized against a uniform background. We opt to follow suit and note that learning motifs as IGMs against a uniform background is the simplest and quickest option in our implementation. Plots of the weights are also readily interpretable as sequence logos. Details are provided in the supplementary materials.

The top row box plots depict the test set activation differences for each filter broken down by true label ($Y \in \{0, 1\}$, left versus right, respectively), as well as the associated β^j_1 coefficients in red. These β^j_1 may be interpreted as effect size estimates for each motif. Not striking is the observation that the sign of the β^j_1 coefficients associated with filters $j = 1, 2, 3$ is negative while it is positive for filter 4. The sequence logos indicate that, as expected, the strings of cytosine nucleotides are highly predictive for negative sequences while the string of guanine nucleotides is highly predictive for positive sequences.

Our first contribution is illustrated in panel B of Fig. 2.1: we constrain the model weights during the training procedure to encourage motif interpretability. Specifically, we restrict the individual filter weights $\omega^j_{k,l} \geq 0$ and their associated offsets $\beta^j_0 \leq 0$, and additionally, re-scale the weights column-wise to maintain a valid information-theoretic relationship peri-training. The constraint on the offset weights β^j_0 for each filter to be strictly non-positive is incorporated to improve the interpretation of the filter activations: consider that the minimum activation value, ζ^j_n for observation n and filter j, attainable without a bias offset β^j_0 and under the sigmoidal activation would be

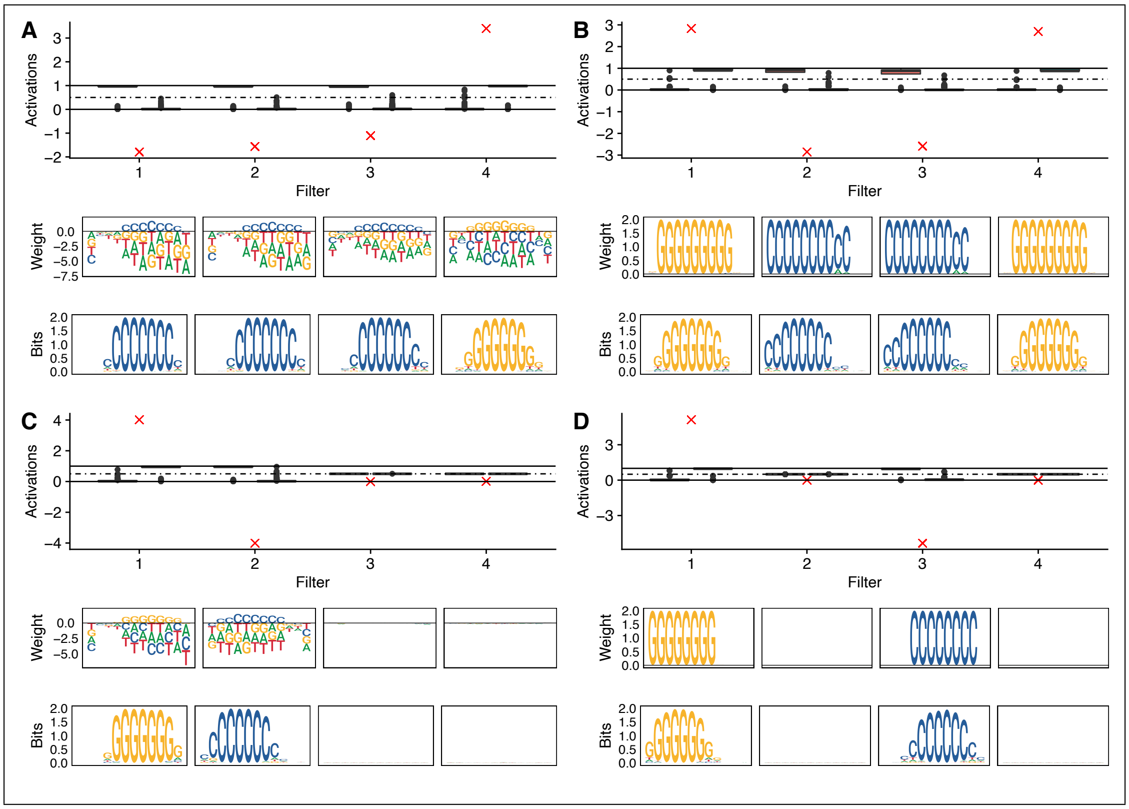


Figure 2.1: Effect of weight constraints and regularization on learning motifs. Box plots show first-layer convolutional filter activations post sigmoid transformation by class label. $Y = 0$ sequences containing the $CCCCCCCC$ motif achieve large activations with the C-motif filters (filters 1, 2, 3), $Y = 1$ sequences containing the $GGGGGGGG$ motif achieve large activations with the G-motif filter (filter 4). Red $\times$'s indicate the associated β_1^j coefficients (effect size estimates). **A.** Unconstrained filters (middle row) within an unregularized model (top row) learn redundant sequence motifs and require test set observations for motif interpretation (bottom row). **B.** Unregularized filters constrained to represent valid IGMs do not require test set observations and are directly interpretable as sequence-logo motifs. Filter redundancy remains. **C.** Filter regularization discourages learning redundant features. **D.** Constrained filters within a regularized model learn distinct sequence motifs directly with no need for post-hoc interpretation procedures.

1/2. Quite simply the addition of a negative offset allows the value of ξ_n^j to decrease to zero. The middle row of Fig. 2.1B highlights the utility of the weight constraints by plotting the weights directly; no input test set observations or post-hoc optimization procedures were required. The filters maintain the strong class-discrimination as evident in the top row box plots, however there appears significant redundancy as the same 10-mer motifs were learned by two filters each. Thus our second contribution, also aimed at encouraging model interpretability, is to regularize the weights during the training procedure. We utilize the sparse group lasso penalty with each filter defined as a group[127]. L1 regularization (i.e. the so-called *lasso* penalty[137]) on the filter weights pushes non-informative weights to zero and may be interpreted as encouraging a KL divergence of 0 between the observed distribution and the background probabilities. We consider the sum of all weights in a filter as a measure of total motif information gain and these are the values regularized via the group lasso penalty which enourages irrelevent motifs to contain zero information gain and discourages correlated motifs. We detail these regularization schemes and interpretations in the supplementary materials.

Panel C of Fig. 2.1 shows the results of such a regularization scheme applied to the vanilla CNN in panel A. As desired, two of the filters now contain zero information gain and their associated effect estimates β_1^j and offsets β_0^j (not pictured) are also zero. These filters may be discarded without impacting model performance. Finally, panel D shows the results of utilizing both the constraints and the regularization scheme. We see the weights perfectly recapitulate the inserted 8-mer sequences and in fact illustrate the motif more clearly than the approach based on test set observations (bottom row). We note that all models achieved near-perfect predictive accuracy (98%) and were trained for five epochs. Parameter tuning was performed to identify suitable values for the regularization penalties. While we only present the results for learning IGMs, learning PWMs is also possible and included in our software implementation. The reader is encouraged to view the supplement for a brief discussion on the implications of learning IGMs or PWMs.

2.3 Results

Simulation Study

The toy example previously described is useful for illustration but it represents an unrealistic situation in which nearly all observations contain an optimal representative motif. We therefore consider a second and more challenging simulation study and base the methodology on that laid out in [123]. Specifically, we utilize the simdna package[76] to generate 100K nucleotide sequences of length 200bp sampling from motifs with less degenerate distributions. We sampled from three motifs: MYC, CTCF, and IRF[*], where positively-labelled sequences ($Y = 1$) contain 0-3 occurrences of each motif. 100K negatively-labelled sequences ($Y = 0$) were generated from random genome background with 40% GC-content. Additionally, 10% of the observations were shuffled between positive and negative classes to increase the difficulty of the learning procedure. The top row of Fig. 2.2B shows the target sequence logos for the three sampled motifs (*MYC_known1, CTCF_known1, IRF_known1*) embedded in the positive cases. These motifs are the subsequences we wish to learn.

Multi-dimensional output model: A regularized and constrained CNN (as described in the previous section) was trained via stochastic gradient descent for twenty epochs with the learning rate initially set at 0.02. $J = 8$ first-layer convolutional filters were randomly initialized following a uniform distribution on the interval $(0, 0.5)$. Logistic loss plus the regularization terms was minimized over the three motif classes. Thus the target output vector Y_n for observation X_n is a 1×3 binary array with each entry indicating the presence/absence of motif j. Sequences containing no motif instances (i.e. purely random background) are labelled $Y_n = [0\ 0\ 0]$ whereas a sequence containing, for example, the motifs MYC and CTCF but not IRF would be labelled $Y_n = [1\ 1\ 0]$. Under such a formulation, β_1^j is no longer a $1 \times J$ vector but a $3 \times J$ matrix with rows corresponding to entries

[*] The JASPAR naming convention denotes the IRF motif (which we sampled from) as the IRF2 motif. We maintain this distinction throughout the text, i.e. when initializing filters with JASPAR motifs the name IRF2 is used (Supplementary Fig.S18) yet when learning IRF *de novo*, the IRF label is used.

in the target array Y. Fig. 2.2 panel A plots the fitted β_1^j estimates (Y-axis) against the mean activation difference between test set observations containing *any* motif occurrences (i.e. any entry in the 1×3 binary array is greater than zero) and test set observations containing *no* motif occurrences (all entries exactly zero). Faceting corresponds to rows in the β_1^j matrix such that the heading CTCF represents the CTCF target class, the IRF heading represents the IRF target class, etc. It is indeed reassuring that a single filter exhibits both the largest mean activation difference and the largest effect size within a facet, and whence visualized as a sequence logo (panel B bottom row) this filter recapitulates the desired target motif (panel B top row). Five of the eight filters have associated β_1^j coefficients equal to zero across all facets as well as zero activation difference. These filters may be removed from the model without impacting predictive accuracy and, as evident in Supplementary Fig. S16, are zero information gain motifs, thus indicating the effectiveness of the regularization. Only the three filters with non-zero information gain, effect size, and mean activation difference need to be retained in the model. The weights composing these filters are shown in panel B of Fig. 2.2 with associated Q-values from running the Tomtom motif matching tool[43]. In all three cases the most significant Q-value is the desired target motif (Supplementary Fig. S17). We label the points in panel A with the most significant Tomtom match and note that due to the construction of the simulation (sequences may contain two or even three different motifs), the mean activation difference is non-zero for two motifs in each facet however the effect size estimate is zero. Thus the two off-target *de novo* learned motifs are uninformative for prediction of a given target class, however approximately one-third of sequences may of course contain either (or both) of the motifs.

Single-dimensional output model: While often considered in the literature the multiple-output model previously described is of little use in practice; rarely would such labels exist denoting the presence/absence of each motif. Indeed it is these motifs which we wish to learn and thus a single-output model is of more practical use. Such a model formulation might arise from, for example, a ChIP-seq experiment in which sequences extended from called peaks would be labelled as $Y =$

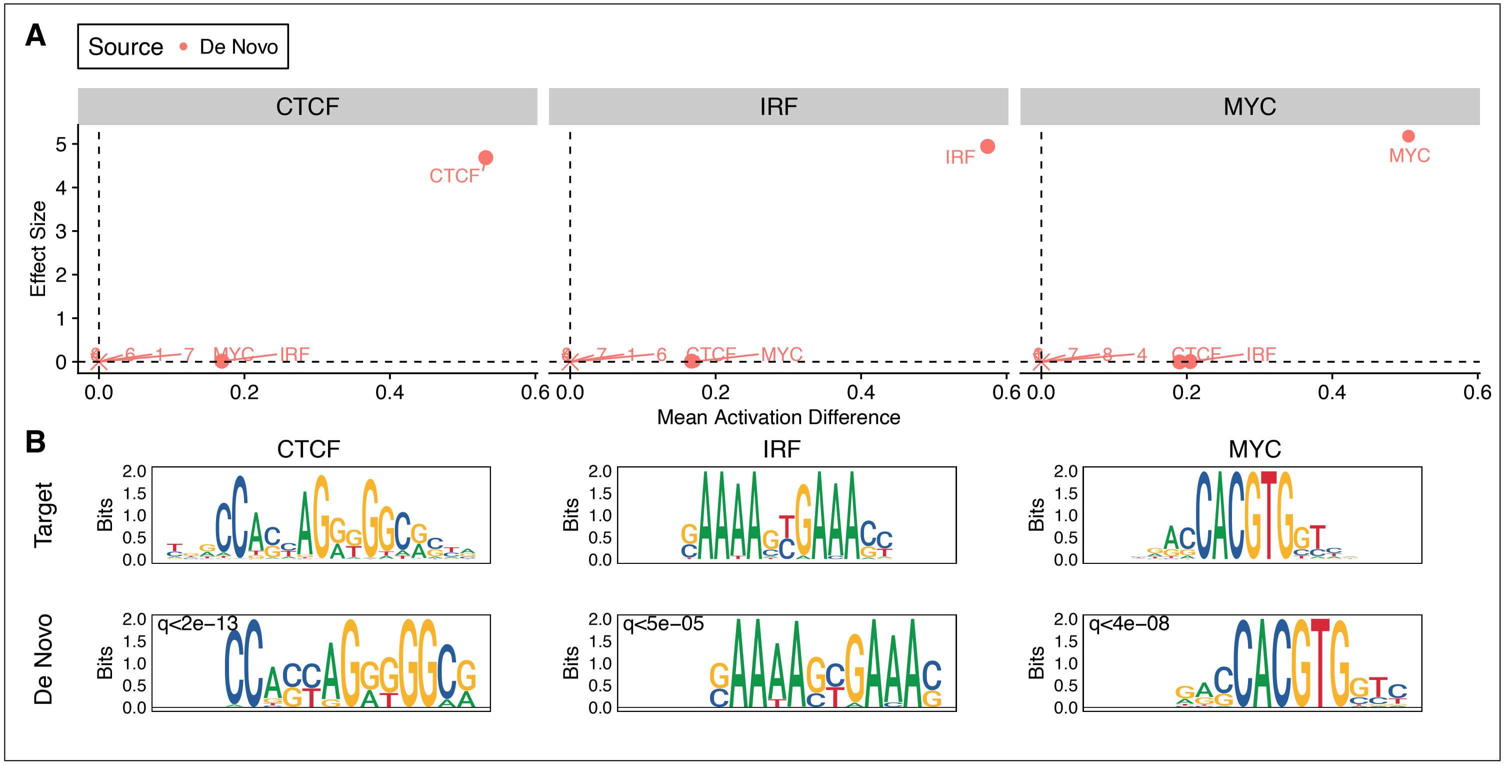


Figure 2.2: *De novo* motif simulation results. Positive cases contain 0-3 insertions of target motifs: MYC, CTCF, IRF (top row panel B) whereas negative cases are sampled from random background. All first-layer convolutional filter weights are initialized randomly and motifs are learned *de novo*. **A.** Filter motif effect size (Y-axis) against mean activation difference between class labels (X-axis). × indicates zero information gain motifs and the associated numeric label indicates the filter number (see Supplementary Fig. S16. **B.** Target sequence-logo motifs (top row), Learned *de novo* motifs (bottom row). Q-values reported from running Tomtom motif comparison tool.[43].

1 while sequences of equivalent length would be drawn from random genome background and labelled as $Y = 0$. The analytic goal of such a formulation would be, again, discovering which motifs (perhaps even beyond those ChIP-ed for) are abundant in the peaks versus the background. For this reason we collapse the 1×3 target vector into a single value denoting the presence of *any* motif ($Y_n = 1$) versus the absence of *all* motifs ($Y_n = 0$) and highlight a use case for our method. We show how one might initialize filters based on annotated motifs from a database and also learn any extra motifs *de novo*.

We consider two models to highlight this use case: Model 1 initializes filters based on the 579 previously-annotated motifs found in the 2018 CORE vertebrates non-redundant JASPAR database and holds these filter weights fixed throughout training[67]. Such a use case might arise when one does not have *a priori* knowledge of which motifs may be present in the sequences and wishes to estimate the prevalence of previously-annotated motifs. Model 2, on the other hand, initializes two filters with the JASPAR MYC and CTCF motifs and tackles the issue of discovering a motif which is present in the data but not the motif database (in this case, the IRF/IRF2 motif). To achieve this we simply remove the IRF2 motif from the filter initialization and try to learn it *de novo*. We initialize two filters uniformly at random on the interval $(0, .5)$ and learn the motif directly. Such a use case might arise during a specific TF ChIP-seq experiment when one believes several previously-annotated motifs may be present but also wishes to learn unannotated motifs *de novo*. In the first model we impose sparse regularization via the L1-norm on the β_1^j coefficients to encourage those motifs with little-to-no abundance to exhibit an exactly zero effect size ($\beta_1^j = 0$) while in the second model we impose the regularization strategy outlined in the previous section to discourage redundancy.

Supplementary Fig. S18 panel A plots the estimated effect size (β_1^j) against the mean activation difference between classes in the test set for each JASPAR-initialized convolutional filter (post-sigmoid transformation). It is evident that, of the 579 JASPAR-initialized filters, only four exhibit

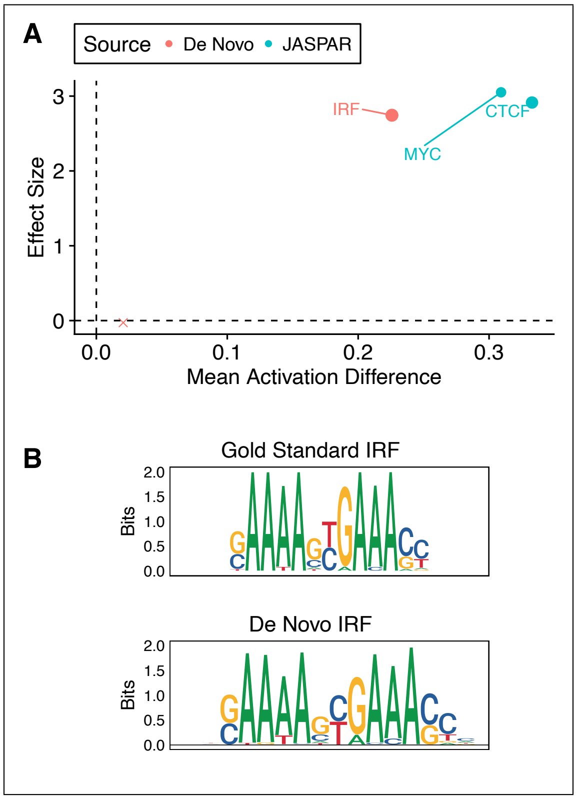


Figure 2.3: Simulation model 2. The two first-layer filters initialized to JASPAR-annotated CTCF and MYC motifs, however the IRF motif must be learned *de novo*. **A.** Filter motif effect size against mean activation difference between class labels by motif source indicates equivalent magnitudes of effect size for both the JASPAR filters and the *de novo* filters. The red cross indicates a low-information gain *de novo* filter that may discarded without affecting model performance. **B.** Gold standard IRF motif (top) exhibits high similarity with *de novo* IRF motif (bottom).

an effect size greater than zero and three of these are the true motifs used in the data simulation (CTCF, IRF/IRF2, MYC). Panel B shows that the fourth, MAX::MYC, is nearly identical to MYC and one may include only a single instance of these during the initialization procedure. We conclude that 575 of the 579 convolutional filters (motifs) may be discarded with no effect on model performance. Similarly, Fig. 2.3 showcases the results for Model 2, in which the IRF motif has been removed. Of note is the large activation difference and estimated effect size for the *de novo* learned IRF motif (Panel A). As desired, the estimated effect size is of an equivalent magnitude as the JASPAR-initialized motifs. Comparing the gold standard embedded motif (left-hand sequence logo of panel B) with the *de novo* IRF motif it is evident how similar these motifs are and how one might utilize our tool for learning motifs *de novo*. We note that, in addition to simply initializing and fixing filters with JASPAR-annotated motifs, one need not hold these fixed during training and may instead choose to update the individual filter weights (motif position-probabilities) to both improve model fit and compare the updated motif with the original motif. We leave this for future work.

Latent variable interpretation: Under the sigmoidal activation function applied to the convolution output (activations) from Model 2, each ξ_n^j is interpretable as the probability that sequence n contains motif j, i.e. $P(\xi_n^j = 1|X_n)$. We sought to assess the accuracy of this latent variable interpretation in Fig. 2.4A-C on a held out test set in which we did not randomly shuffle 10% of the observations between classes. We find the ξ_n^j representations are extremely accurate, achieving $\geq 98\%$ accuracy and an area-under-the-precision-recall curve of ≥ 0.99 for all three motifs (MYC, CTCF, IRF). Pooling information across all motifs slightly diminishes performance, especially for the sequences containing only a single unique motif (combined-model accuracy for sequences containing a single motif ranges from $0.617 - 0.816$ yet accuracy is perfect for all sequences containing two or more unique motifs). One should take care to note the imbalance in the individual motif comparisons relative to the balanced dataset (1:4 versus 1:1 positive:negative cases, respectively).

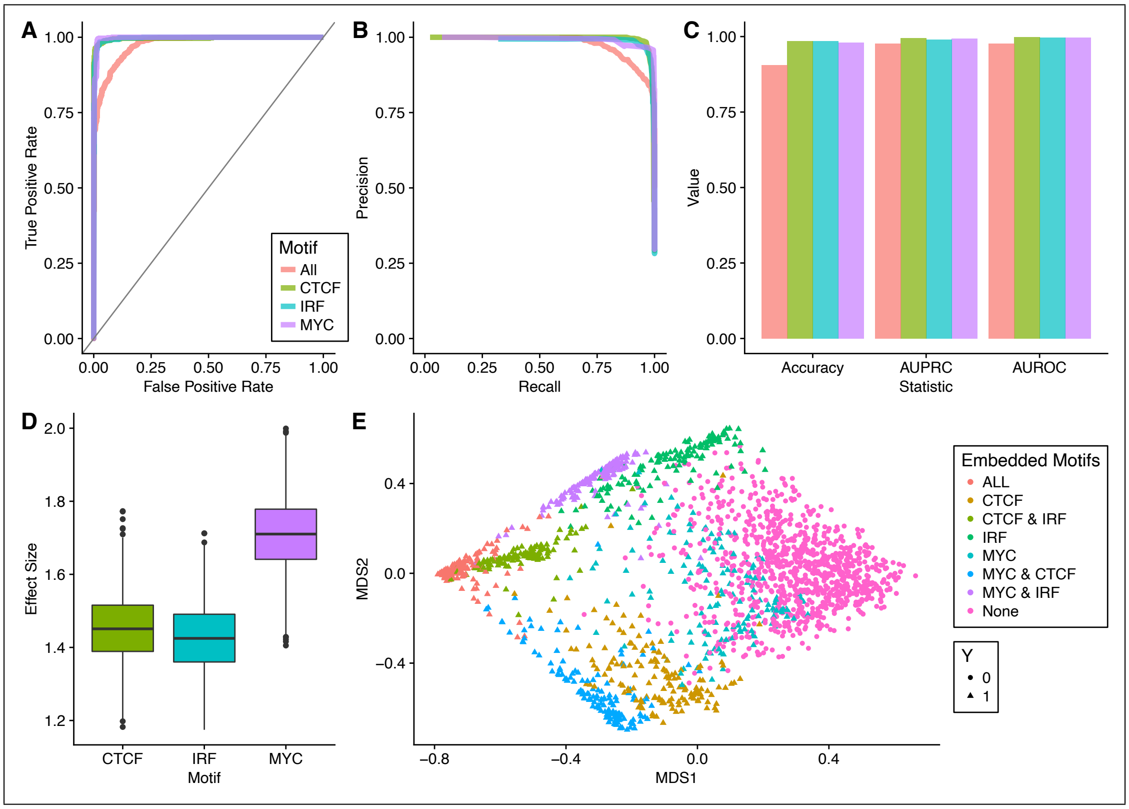


Figure 2.4: Model-based evaluation of simulated motif presence. **A-C.** ROC curve, precision-recall curve, and prediction statistics evaluating correctness of calling individual motifs present within test set sequences based on max-pooled filter activation values. The *All* label simply combines the three filters in a logistic regression model to evaluate presence of *any* motif versus absence of *all* motifs. Dashed lines represent using the true, sampled-from motif whereas solid lines represent the *de novo*-learned motifs. **D.** Effect size estimates for each motif based on Monte Carlo realizations treating the max-pooled filter activation values as Bernoulli random variables. **E.** Multi-dimensional scaling using the max-pooled filter activation values as features.

Additionally, in panel D, we explored the possibility of using Monte Carlo Bernoulli draws with probability based on the individual ξ_n^j, and using the realizations to fit a logistic GLM. We find all coefficients are statistically significant at the $\alpha = .05$ level across all simulations ($m = 1000$), however the fitted effect sizes underestimate the true effect as fixed by the simulation, presumably due to the collinearity of the predictors. Panel E plots two dimensions from multi-dimensional scaling (MDS) performed on the activation values from the retained filters (three in total), highlighting a separation between both the class labels ($Y = 0/1$) and, to a lesser degree, the separation within the positive cases due to the underlying embedded motifs.

Data Application

We applied our method to *in vivo* transcription factoring binding data from the ENCODE Consortium[23]. Specifically, following the protocol described in[125] with slight modifications, we downloaded CTCF TF ChIP-seq data for the Gm12878 cell line. 100bp windows were extended from the called peaks for the positive cases while negative cases were obtained by removing from all DNase peaks in Gm12878 the top 150K relaxed TF peaks called by SPP at a 90% FDR[79]. Peaks originating from chromosome 1 were utilized for the training set and peaks originating from chromosome 2 were utilized for the validation set. We down-sampled the negative cases (genome background) during the training procedure however did not down-sample cases in the validation set.

Fig. 2.5A showcases the utility of our approach: we begin by initializing the filters with all possible JASPAR motifs. We denote this as the shotgun approach as many of the motifs miss the mark (e.g. left-hand panel of A, the vast majority of motifs exhibit both 0 mean activation difference and 0 effect size). We discard all filters from the model which do not have both an estimated effect size greater than 0.01. In the case of this analysis, five motifs were retained and indeed it is reassuring to see both the CTCF motif and the CTCF reverse complement (CTCF_RC) as retained. We train this model for 10 epochs to obtain fitted values for all filters' β_0^j and β_1^j, and then initialize a *de novo* model with these five filters (and their associated β) fixed, but also eight filters randomly initialized. These latter filters will be used to learn the *de novo* motifs. We train this model for 30 epochs, this time utilizing both the regularization tactics and the weight constraints, and report the estimated effect size and mean activation difference for the *de novo* motifs in the middle panel. We calculate the information gain of each motif as the sum of all weights in the filter and provide this value as the size of the associated point. We find six filters to contain zero information gain and thus we discard these filters. Our final model then makes use of the five JASPAR filters and the two *de novo* filters, and we train this model for another 30 epochs to obtain values for each β_0^j and β_1^j, as well as the overall offset β_0. The right-hand panel of Fig. 2.5A illustrates both the high information gain of the *de novo* motifs, as well as the larger effect size and mean activation difference. Visualizing all the motifs in Fig. 2.5B sheds light on what the *de novo* filters have learned: namely slightly altered representations of the CTCF and CTCF RC motifs. In fact, we find the effect of the leading G to be amplified in the *de novo* 2 motif relative to the CTCF known motif, and, correspondingly, the trailing C in the RC to be amplified. This suggests the subsequence GCGC is more abundant than expected by the CTCF JASPAR motifs. Similarly we note the deletion of a rather uninformative position in the motif (position 18 in CTCF_RC and position 2 in CTCF). We finally provide test set accuracy statistics and illustrate via MDS the class-separability of sequences using these seven activation values as features.

2.4 Discussion

Our proposed model directly learns sequence motifs such that interpretation of the convolutional filters is not reliant upon test set observations and the weights comprising the filter are directly interpretable as information gain measures or position weight (log-odds) measures. We address the issue of filter redundancy along with interpretability by incorporating weight constraints and regulariza-

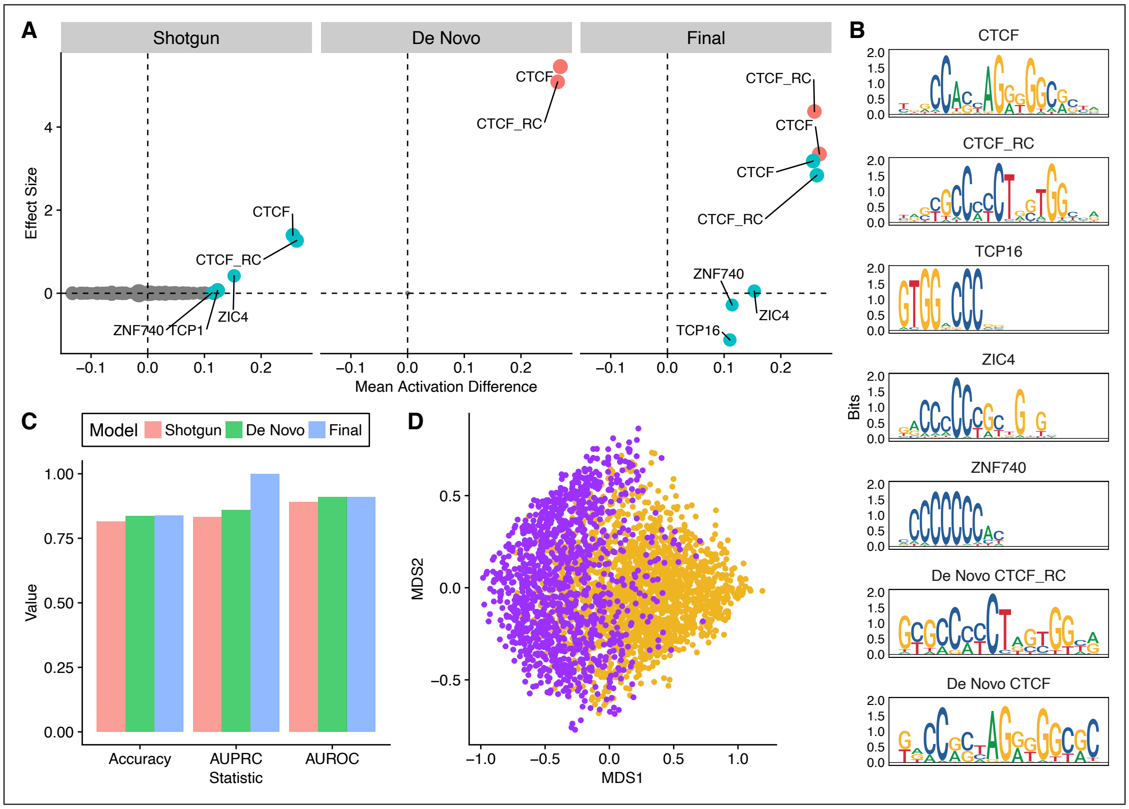


Figure 2.5: ENCODE pipeline results. **A.** Motif effect size against mean activation difference between class labels. The *Pipeline* model includes all 2018 JASPAR-annotated motifs (core vertebrates, non-redundant). Any motifs with associated effect size $< \ .01$ are discarded and subsequently held fixed while eight *de novo* motifs are learned (center panel). The *Final* model discards uninformative *de novo* motifs and refits effect size estimates with filter weights fixed. **B.** Collection of motifs selected from *Final* model plotted as sequence-logos. **C.** Prediction statistics evaluating classification based on model. **D.** Multi-dimensional scaling using the max-pooled filter activation values from the *Final* as features, colored by label (purple: $Y = 1$, yellow: $Y = 0$).

tion techniques. The weight constraints limit the range of the individual filter weights to restrict their values to be directly interpretable as IGMs/PWMs while the regularization scheme encourages learning non-redundant and high-relevance motifs.

To the authors knowledge this is the first method capable of incorporating previously-annotated sequence motifs in a CNN framework, similar to [101] but with the ability to learn motifs *de novo*. Notably the method achieves this by leveraging IGMs/PWMs as convolutional filters and ensuring the *de novo* motifs are valid information gain/position weight measures. Interestingly, other motif measurement systems such as the position probability matrix (PPM) may also be used as convolutional filters although several changes must be made. First, the regularization scheme must be adjusted. In the case of the PPM, filter weights would need to be regularized around their expected background frequencies (e.g. 0.25). Further, all weights would require a column-wise sum to unity, thus all weights would need to be initialized under such a condition and enforced throughout the training procedure. Additionally, any low-information gain motifs would be those filters with all weights centered at their background frequencies, and thus summing all weights within the motif would not constitute a measure of information gain since all sums would be the same, regardless of the amount of information gain contained. One would likely transform the PPM into an IGM in order to quantify importance as measured by the KL divergence. The same holds for using the PWM, as the negative values associated with the low abundance nucleotide positions would overwhelm the sum calculation.

Like any regularization method, parameter tuning is essential and as the number of parameters to tune increases, so does the difficulty in finding suitable values. This issue, however, is ubiquitous with deep learning techniques and does not affect our method any more than usual. Furthermore, as our primary concern is more with learning discriminatory features and less with predictive accuracy, we find parameter tuning to act as a sort of interpretability sieve; under stricter regularization only the most discriminatory features will appear at the cost of predictive accuracy while under laxer regularization predictive accuracy may improve at the cost of lesser filter interpretability. Indeed this trade-off epitomizes the divide between traditional statistical techniques and machine learning methods, however once discriminatory motifs/features have been learned one may refit a more complicated (i.e. deeper) model to attain improved predictive accuracy. One may even desire to learn motifs *de novo* as IGMs, and then refit the model using PWMs given the one-to-one correspondence between the two.

The proposed methods may be useful for several interesting genomics applications; namely any application requiring the need to learn differential sequence motifs. Examples include DNase-seq and ATAC-seq footprinting. We leverage deep learning infrastructures to provide a more succinct set of features and abandon the traditional machine learning paradigm stating that higher accuracy is paramount. We instead focus on a simple modelling framework which provides end-to-end interpretation throughout. Our methods are implemented in an R package available at `https://github.com/mPloenzke/learnMotifs` which relies upon Keras[21] for all of the heavy lifting.

3

Improving CNN interpretability with exponential activations

Deep convolutional neural networks (CNNs) trained on regulatory genomic sequences tend to learn distributed representations of sequence motifs across many first layer filters. This makes it challenging to decipher which features are biologically meaningful. Here we introduce the exponential activation that – when applied to first layer filters – leads to more interpretable representations of motifs, both visually and quantitatively, compared to rectified linear units. We demonstrate this on synthetic DNA sequences which have ground truth with various convolutional networks, and then show that this phenomenon holds on *in vivo* DNA sequences.

3.1 Introduction

Convolutional neural networks (CNNs) applied to genomic sequence data have become increasingly popular in recent years[6,66,154], demonstrating state-of-the-art accuracy on a wide variety of regulatory genomics prediction tasks, including transcription factor binding and chromatin accessibility. Their success has been attributed to the ability to learn features directly from the training data in a distributed manner[80]. These learned features are, in some cases, suggested to correspond to biologically-relevant sequence motifs, particularly in first convolutional layer filters[6,66].

An understanding of what a trained model has learned is then possible through attribution scores, which can be attained with perturbation methods[6,154] and saliency maps/gradient techniques[128,123,71]. However, the resultant attribution maps tend to be difficult to interpret, requiring downstream analysis to obtain more interpretable features, such as sequence motifs, by averaging clusters of attribution scores[124]. The factors that influence the quality of attribution scores – such as the CNN architecture, regularization, and training procedure – are not well characterized. There is no guarantee that attribution methods will reveal features that are biologically interpretable for a given CNN, even if it is capable of a high classification performance.

An alternative approach is to design CNNs such that their filters directly learn more interpretable features[73,98]. In this manner, minimal posthoc analysis is required to obtain representations of "salient" features, such as sequence motifs. For instance, pre-convolution weight transformations that model the first layer filters as position weight matrices (PWMs) may be used to learn sequence motifs through the weights[98]. Another CNN design choice employs a large max-pool window size after the first layer, which obfuscates the spatial ordering of partial features, preventing deeper layers from heirarchically assembling them into whole feature representations[73]. Hence, the CNN's first layer filters must learn whole features, because it only has one opportunity to do so.

One drawback to current design principles of CNNs with interpretable filters is that they tend to be limited to shallower networks. Depth of a network significantly increases its expressivity[105], which enables it to learn a wider repertoire of features. In regulatory genomics, deeper networks have found greater success at classification performance. In practice, deeper CNNs are generally harder to train and are more susceptible to performance variations with different hyperparameter settings.

One consideration for the interpretability of a CNN's filters that has not been thoroughly explored in genomics is the activation function. Rectified linear units (ReLUs) are the most commonly employed activations in genomics. In computer vision, neurons activated with a rectified polynomial, which has a close relationship to dense associative memories[?] , were shown to learn representations of numbers when applied to the MNIST dataset. This activation breaks common sense because it is unbounded and hence can diverge relatively quickly.

A divergent activation is intriguing from a signal processing perspective because it can force the network to regulate its weights such that the activity of a neuron does not blow up. For instance, if background signals are propagated through, then the rest of the network has to suppress this amplified noise in order to make accurate classification. We suspect that the network would instead opt for a simpler strategy of suppressing background signals prior to activation, thereby only propagating discriminatory signals. One drawback of the rectified polynomial, however, is that it is unclear how to select the order of the polynomial thus introducing another hyperparameter to tune.

Building upon these previous studies, we introduce a novel application of an exponential activation function. We perform systematic experiments on synthetic data that recapitulates a multi-class classification task to compare how activations of first layer filters affect representation learning of sequence motifs. We find that an exponential activation applied only to the first layer filters consistently learn whole motif representations, irrespective of the network's depth and design. On the other hand, motif representations for CNNs that employ ReLU activations in the first layer predictively depend on CNN design. We then show that these results generalize to *in vivo* sequences.

3.2 Materials and methods

Data. We analyzed a dataset from[73], which consists of synthetic DNA embedded with known transcription factor (TF) motifs to recapitulate a multi-class classification task of identifying transcription factor binding motifs. Specifically, synthetic sequences, each 200 nucleotides long and composed of random DNA, were implanted with 1 to 5 known TF motifs, randomly selected with replacement from a pool of 12 motifs. This dataset makes a simplifying assumption that the only important pattern for a given binding event is the presence of a PWM-like motif in a sequence. Since we have ground truth for all of the relevant TF motifs, and also where they are embedded in each sequence, we can test the efficacy of the representations learned by a trained CNN.

Models. We used two CNNs, namely CNN-50 and CNN-2[73], to learn "local" representations (whole motifs) and "distributed" representations (partial motifs), respectively. Both networks take as input a 1-dimensional one-hot-encoded sequence with 4 channels, one for each nt (A, C, G, T), and have a fully-connected (dense) output layer with 12 neurons that use sigmoid activations. The hidden layers for each model are:

1. CNN-2
    1. convolution (30 filters, size 19, stride 1)
       max-pooling (size 2, stride 2)
    2. convolution (128 filters, size 5, stride 1, ReLU)
       max-pooling (size 50, stride 50)
    3. fully-connected layer (512 units, ReLU)

2. CNN-50
    1. convolution (30 filters, size 19, stride 1)
       max-pooling (size 50, stride 50)
    2. convolution (128 filters, size 5, stride 1, ReLU)
       max-pooling (size 2, stride 2)
    3. fully-connected layer (512 units, ReLU)
3. CNN-deep
    1. convolution (30 filters, size 19, stride 1)
    2. convolution (48 filters, size 9, stride 1, ReLU)
       max-pooling (size 3, stride 3)
    3. convolution (96 filters, size 6, stride 1, ReLU)
       max-pooling (size 4, stride 4)
    4. convolution (128 filters, size 4, stride 1, ReLU)
       max-pooling (size 3, stride 3)
    5. fully-connected layer (512 units, ReLU)

All models incorporate batch normalization[57] in each hidden layer; dropout[133] with probabilities corresponding to layer1 0.1, layer2 0.1, layer3 0.5 for CNN-2 and CNN-50; and layer1 0.1, layer2 0.2, layer3 0.3, layer4 0.4, layer5 0.5 for DistNet; and $L2$-regularization on all parameters in the network with a strength equal to 1e-6.

Training. We uniformly trained each model by minimizing the binary cross-entropy loss function with mini-batch stochastic gradient descent (100 sequences) for 100 epochs. We updated the parameters with Adam using default settings[68]. All reported performance metrics are drawn from the test set using the model parameters which yielded the lowest loss on the validation set. Each model was trained 5 times with different random initializations according to[48].

Visualization of convolutional filters. To visualize first layer filters, we scanned each filter across every sequence in the test set. Sequences whose maximum activation was less than a cutoff of 50% of the maximum possible activation achievable for that filter were removed. A subsequence the size of the filter is taken about the max activation for each remaining sequence and assembled into an alignment. Subsequences that are shorter than the filter size due to their max activation being too close to the ends of the sequence were also discarded. A position frequency matrix was then created from the alignment and converted to a sequence logo.

Quantitative motif comparison. The interpretability of each filter was assessed using the Tomtom motif comparison search tool[43] to determine statistically significant matches to the 2016 JASPAR vertebrates database[?]. Since the ground truth motifs are available for our synthetic dataset, we can test whether the CNNs have captured *relevant* motifs.

3.3 Results

To test the extent that activation functions influence representation learning by first layer filters, we trained various CNNs, namely CNN-2, CNN-50, and CNN-deep, on the synthetic dataset with 5 different initializations and used the average area under the precision recall curve (auPR) to compare performance and quantify the ability to learn sequence motifs using Tomtom[43]. For each network, we compared ReLU and exponential activations only on the first layer, while employing ReLU activations for the other hidden layers.

Analyzing synthetic sequences CNNs trained on the synthetic dataset show no significant differences in the auPR on held-out test sequences across models and across activations (Table 3.1). A visual comparison of the representations learned by first layer filters show that CNN-2 and CNN-deep do not learn sequence motifs well when employing ReLU activations (Figure 3.1). This is expected because deeper layers are able to build hierarchical representations from partial motif features for these networks. Indeed less than 1% of the filters match ground truth motifs according to a Tomtom motif comparison search across 5 independent trials for each network. Nevertheless, about 60% of the filters of CNN-2 and CNN-deep have a statistically significant match to some motif in

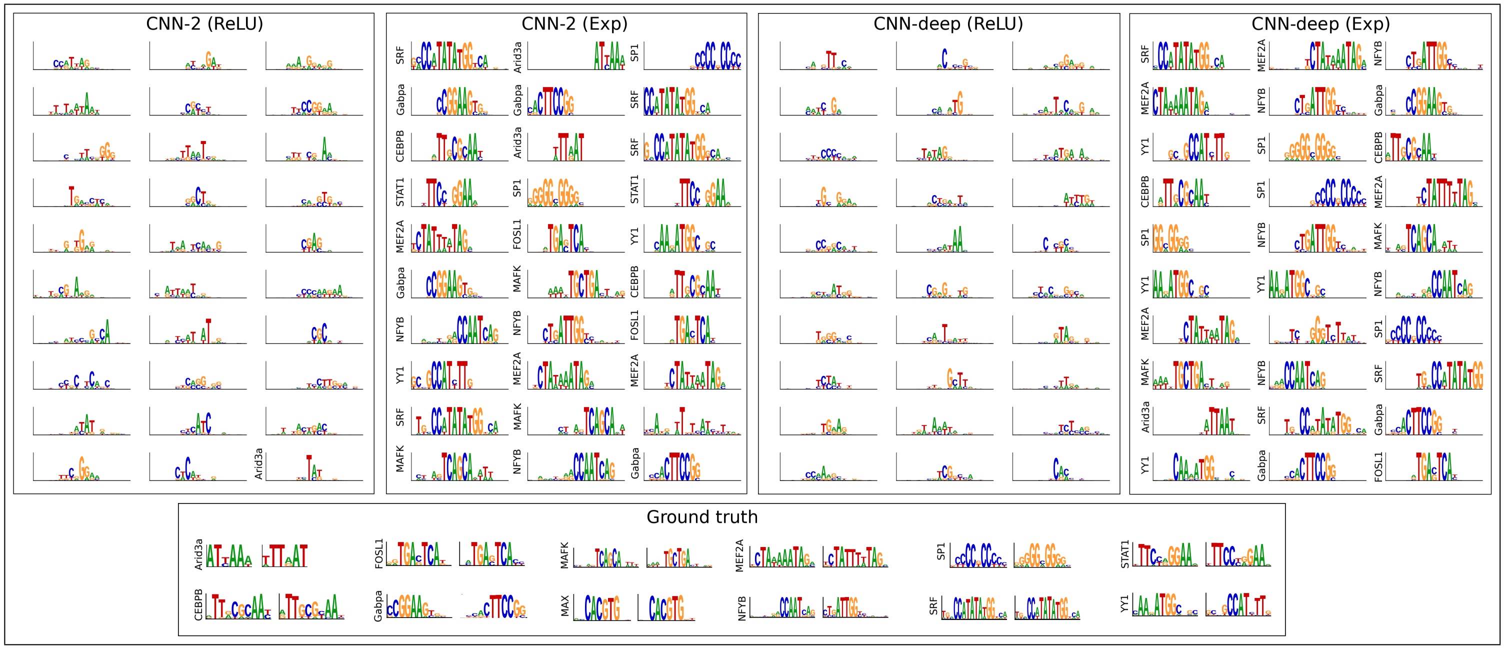


Figure 3.1: Representations learned from synthetic sequences. Sequence logos of the first convolutional layer filters are shown for (from left to right): CNN-2 with ReLU activations, CNN-2 with exponential activations, CNN-deep with ReLU activations, and CNN-deep with exponential activations. The sequence logo of the ground truth motifs and its reverse complement for each ground truth motif is shown at the bottom. The y-axis label on select filters represent a statistically significant match to a ground truth motif.

the JASPAR database, even though most of these matches are not relevant. As expected, larger max-pooling is required to yield interpretable filters for CNNs with ReLU activations[73]. Indeed 92% of CNN-50's filters match ground truth motifs.

Strikingly, the convolutional filters for CNN-2 and CNN-deep, which were unable to learn motifs with ReLU activations, visually seem to capture many ground truth motifs when switching to an exponential activation (Figure 3.1). Quantification by Tomtom confirms that greater that 90% of the filters match ground truth motifs. This demonstrates that exponential activations provide interpretable filters for CNNs, irrespective of max-pooling size.

Functions with positive second derivatives, such as the exponential function, produce increasingly larger values for increasing input values, and as such may be referred to as divergent functions. Undoubtedly, activation values attain much larger values under such a transformation than under a convergent transformation such as the sigmoid function or a linear transformation such as the ReLU (in the positive domain). Despite this signal amplification in terms of magnitude, however, a noise dampening effect is observed as the noisy activation values which the filters propagate through the model, namely false positive motif scans, are all but eliminated when employing the exponential activation function (Figure 3.2C; vertical blue line signifies the location of the embedded MAX motif, vertical red line signifies FOSL1). On the other hand, many high activation values arising from false positive motif scans are evident when employing the ReLU (Figure 3.2B; the filter learning the MAX motif produced only the seventh largest activation value) or the standard PWM scan Figure 3.2A). This phenomenon of sharpened signal was observed when employing other divergent activation functions, such as a third order polynomial, and may be related to the gains in interpretability, however this remains an open question at the time.

ANALYZING *IN VIVO* SEQUENCES To test whether the same representation learning principles generalize to *in vivo* sequences, we modified the DeepSea dataset[154] to include only *in vivo*

			Synthetic			In vivo	
		motif match	motif match		motif match	motif match	
	model	auPR	(JASPAR)	(Relevant)	auPR	(JASPAR)	(Relevant)
ReLU	CNN-2	0.877±0.001	0.607±0.013	0.007±0.013	0.608±0.010	0.656±0.042	0.056±0.042
	CNN-50	0.865±0.033	0.993±0.013	0.920±0.058	0.575±0.005	0.900±0.000	0.733±0.047
	CNN-deep	0.873±0.047	0.600±0.000	0.000±0.000	0.639±0.004	0.611±0.016	0.011±0.016
Exponential	CNN-2	0.884±0.018	0.987±0.016	0.960±0.013	0.620±0.001	0.922±0.016	0.778±0.016
	CNN-50	0.885±0.005	1.000±0.000	0.913±0.034	0.597±0.012	0.933±0.027	0.656±0.031
	CNN-deep	0.835±0.044	0.953±0.016	0.940±0.039	0.630±0.014	0.900±0.027	0.722±0.016

Table 3.1: Performance comparison. This table shows the average area under the precision-recall curve (auPR) across the 12 TF classes, average percent match between the first layer filters and the entire JASPAR vertebrates database (JASPAR), and the average percent match to any ground truth TF motif (Relevant) for different CNNs. The errors represent the standard deviation across 5 independent trials.

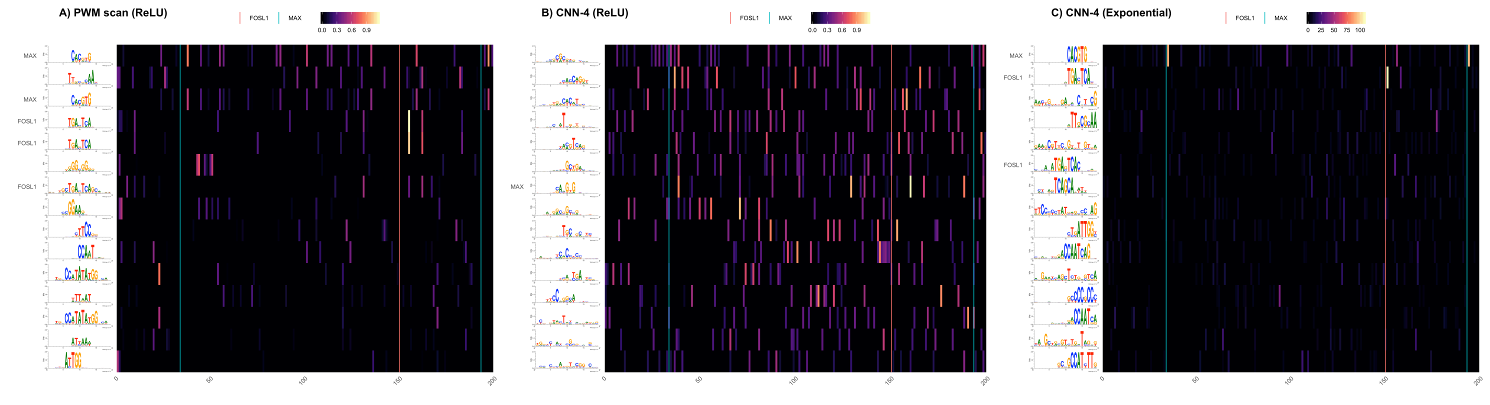


Figure 3.2: First layer filter activation heatmaps for a single sequence containing the MAX and FOSL1 motifs. Three models (panels) shown, all of which utilize the same architecture other than first layer convolutional activation function (PWM scan, ReLU, and Exponential). The 15 first layer filters with the largest activations and their associated sequence logo are depicted along the rows, and the cell value is the activation of the filter with the input sequence. Filters significantly matching annotated JASPAR motifs used in the simulation are indicated with text beside the logo.

sequences that have a peak called for at least one of 12 ChIP-seq experiments, each of which correspond to a TF in the synthetic dataset (see Supplemental Table S1 in[73]). The truncated-DeepSea dataset is similar to the synthetic dataset, except that the input sequences now have a size of 1,000 nt in contrast to the 200 nt sequences in the synthetic dataset.

We trained each CNN on the *in vivo* dataset following the same protocol as the synthetic dataset. Similarily, a qualitative comparison of the first layer filters show that employing exponential activations consistently leads to more interpretable filters that visually matches known motifs (Fig. 3.3). By employing the Tomtom motif comparison search tool, we quantified the percentage of statistically significant hits between the first layer filters against the JASPAR database (see Table 3.1). Indeed, a higher fraction of the filters of CNNs that employ exponential activations have a statistically significant match to known motifs. On the other hand, CNNs that employ ReLU activations are more sensitive to their network design with CNN-50 being the only network that learns motifs well, yielding a percent match of 90%. We note that the performance drop for *in vivo* sequences is expected as they are more complicated, *i.e.* many filters find a GATA motif. We envision that adding more filters in the first layer can help address some of this discrepancy.

3.4 Conclusion

A major goal is to interpret learned representations of CNNs so that we can gain insights into the underlying biology. Deep CNNs, however, tend to learn distributed representations of sequence motifs that are not necessarily human interpretable. Although attribution methods can identify features that lead to decision making, their scores tend to be noisy and difficult to interpret. We show that an exponential activation is a powerful approach to encourage first layer filters to learn sequence motifs. We believe that if applied to deeper layers, it could also improve interpretability in deeper layers to potentially capture motif-motif interactions. Moving forward, one promising

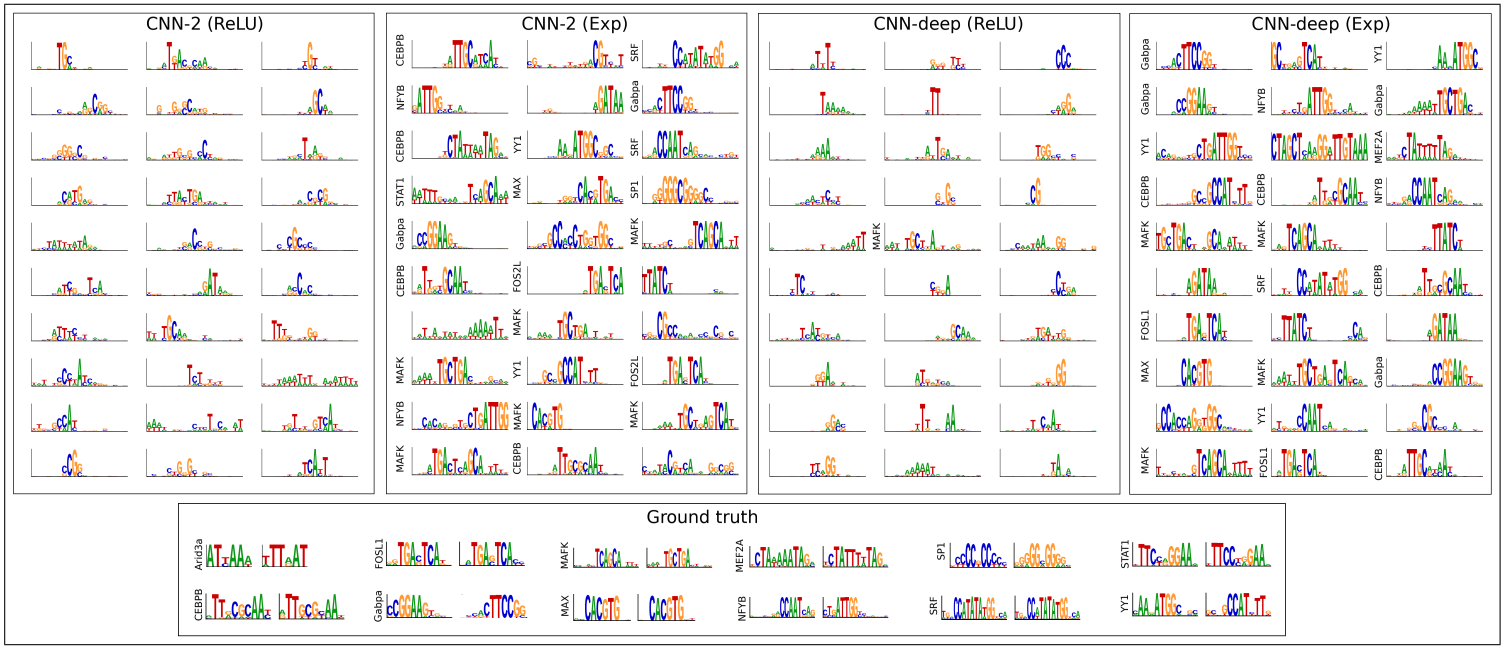


Figure 3.3: Representations learned for *in vivo* sequences. Sequence logos for first convolutional layer filters are shown for (from left to right): CNN-2 with ReLU activations, CNN-2 with exponential activations, CNN-deep with ReLU activations, and CNN-deep with exponential activations. The sequence logo of the ground truth motifs and its reverse complement for each transcription factor is shown at the bottom. The y-axis label on select filters represent a statistically significant match to a ground truth motif.

avenue is to combine attribution methods with CNNs that employ exponential activations so that noisy attribution scores can be aided with the interpretable first layer filters.

4

Deep learning for inferring transcription factor binding sites

Deep learning is a powerful tool for predicting transcription factor binding sites from DNA sequence. Despite their high predictive accuracy, there are no guarantees that a high-performing deep learning model will learn causal sequence-function relationships. Thus a move beyond performance comparisons on benchmark datasets is needed. Interpreting model predictions is a powerful approach to identify which features drive performance gains and ideally provide insight into the underlying biological mechanisms. Here we highlight timely advances in deep learning for genomics, with a focus on inferring transcription factors binding sites. We describe recent applications, model architectures, and advances in *local* and *global* model interpretability methods, then conclude with a discussion on future research directions.

4.1 Introduction

Deep learning is a machine learning paradigm that is represented as a multi-layer, *i.e.* deep, neural network, composed of layers that enable hierarchical representations to be learned automatically from the data through training on one or more tasks. The popularity of deep learning in -omics applications has exploded in recent years[32]. One major reason for this rise is the democratization of deep learning code through high-level APIs, such as Pytorch[96] and Tensorflow[1], which make it possible to seamlessly build and train deep neural networks (DNNs) on graphical processing units in just a few lines of code. Another reason is the big data boom in genomics, enabled by high-throughput experiments and next generation sequencing[69]. Deep learning is thriving in this big data regime and its applications are extending to many areas in genomics[6,154,66,140,153,58,16]. Here, we highlight timely advances in applications for deep learning in genomics, with a focus on inferring transcription factors binding sites. We highlight recent applications and advances in model interpretability and then conclude with a discussion on future research directions.

4.2 Modeling sequence-function relationships with deep learning

The computational task for inferring TF binding sites from DNA sequence is framed as a single-class or multi-class binary classification problem (for an overview, see Fig. 4.1a). The 2017 ENCODE-

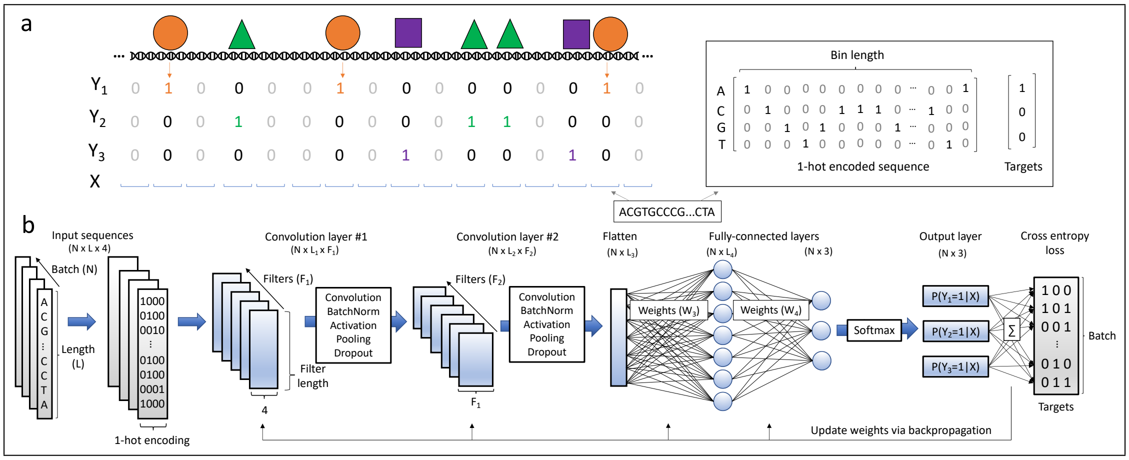


Figure 4.1: Overview of TF binding site prediction task. a) Transcription factors bind to regions of the genome based on sequence specificities and modulate various biological functions. ChIP-seq experiments enrich for short DNA sequences that are interacting with the TF under investigation. The resultant DNA sequences (so-called reads) are aligned to a reference genome and a peak calling tool is employed to find read distributions that are statistically significant compared to background levels. Upon binning the full genome into bins of length L, it is possible to then associate each bin with a binary label denoting the presence ($Y_i = 1$) or absence ($Y_i = 0$) of TF i based on sufficient overlap between the peaks and the bin. The DNA within each bin is represented by a 1-hot encoded matrix and the associated label vectors are used to train a model as a single-class or multi-class supervised learning task. b) Convolutional neural networks are powerful methods to learn sequence-function relationships directly from DNA sequence. A CNN is comprised of a number of first layer filters (F_1) which learn features directly from the N input sequences by computing the cross-correlation between each set of filter weights and the 1-hot encoded sequence. The resultant scans, so-called feature maps, intuitively represent the match between each pattern being learned in a given filter and the input sequence. The feature map then undergoes a series of functional (e.g. batch normalization, non-linear activation) and spatial transformations (e.g. pooling) resulting in a truncated length (L_1). This tensor is then fed into deeper convolutional layers which discriminate higher-order relationships between the learned features. Two convolutional blocks are depicted however this feed-forward process may be repeated any number of times, after which a flattening operation is utilized to reshape the tensor into a $N \times L_3$ matrix. Fully-connected layers perform additional matrix multiplications and ultimately output a probability of class membership for each target. Loss is calculated between the predicted values and the targets, and the weights are updated with a learning rule that uses backpropagation to calculate gradients throughout network.

DREAM challenge exemplifies this task, as competitors were ranked on their ability to accurately predict *in vivo* TF binding on held out test cells and TFs (`https://www.synapse.org/#!Synapse:syn6131484`). The processed data consists of DNA sequences (as a one-hot representation) that are input to the model and corresponding binary labels (peak or no peak). Convolutional neural networks (CNNs) are particularly adept at modeling regulatory genomic sequences (see Fig. 4.1b for details of CNNs). A more detailed review of the computational task and CNNs can be found in Ref.[8]. The primary focus of the following sections will be in the context of CNNs, however many of the techniques described, (e.g. interpretation) are extendable to other classes of DNNs. Moreover, these methods extend naturally to other data modalities that describe sequence-function relationships, such as inferring chromatin accessibility sites and RNA-protein interaction sites.

4.2.1 Recent advances in DNN architectures

There have been many advances in DNN architectures over recent years, primarily driven by applications in computer vision and natural language processing (NLP), that have been slowly ported into genomics, including hybrid models, such as CNN-recurrent neural networks (RNNs)[102,120,103], dilated convolutions[148], residual connections[49], dense connections[54], and (self-)attention[?] .

Network modules Dilated convolutions are interesting because they provide a mechanism for considering a large sequence context, with receptive fields as large as 10kb without pooling[58,65,9]. Dilated convolutions can be combined with other network modules such as residual blocks[58,9] or dense connections[65], both of which foster gradient flow to lower layers. Notably, dilated residual modules were a key component of Alphafold[119], the top protein folding method in the CASP13 free modeling competition.

Attention An interesting direction that is worth serious exploration is attention[129,20,141]. Attention provides an intrinsically interpretable mechanism to place focus on regions-of-interest in the inputs. Albeit, recent evidence suggests that attention is not strongly related to explainability[59]. There are many types of attention mechanisms. State-of-the-art language models in NLP employ a multi-head self-attention, also referred to as a scaled-dot-product attention, which are key components of transformer networks like BERT[29] and XLNet[146]. Recently, Ullah et al. demonstrated how self-attention can be employed to extract associations between TFs that reside in accessible chromatin sites[141].

4.2.2 Incorporating biophysical priors

The salient features in domains such as computer vision or NLP (where most deep learning progress is taking place) are different from genomics, particularly for TF binding, which consists of primary and alternative protein binding sites, cooperative and competitive binding factors, and sequence context (e.g. DNA shape features, GC-content, nucleosome positioning, accessibility and chromatin structure)[56]. In genomics, low-level sequence features, such as motifs, are of particular interest, whereas in images, higher-level features of objects are generally more important. In TF binding prediction tasks, incorporation of biophysical features may provide additional gains in performance. For instance, the top scoring teams[?,?] in the ENCODE-DREAM challenge report increases in predictive performance through the inclusion of manually-crafted chromatin accessibility features (median gains on the area under the precision-recall curve of 0.252 and 0.0504, respectively). Thus an emerging trend is to design DNNs with biophysical priors, making them more suitable to model genomic features, including reverse-compliment equivariance and parameters that capture biophysical properties.

Reverse-compliment equivariance Reverse-compliment (RC) awareness can be achieved via data augmentation with RC sequences, incorporating separate inputs for RC sequences[103], and weight tying[125,11,17], which is more computationally efficient. These domain-motivated models yield improved predictive performance over standard DNNs, with reported gains on the area under the receiver-operating characteristic curve of around 0.02[125]. Reverse-compliment pooling can further reduce the number of parameters[17], albeit introducing a strong prior of motif directional invariance. These strategies are particularly important when analyzing data generated via single-stranded sequencing. To enforce positional invariance of a motif within a filter, circular filters have been shown to be effective[15].

Biophysical parameters Recasting traditional physics-based models as a neural network is an active area of research[28,136,83]. Tareen & Kinney recently showed that biophysical models of TF binding can be represented as a neural network[136], where edges represent meaningful biophysical quantities, such as free energies. In parallel,[83] has also demonstrated how DNNs can be designed with strong biophysical priors. These networks are highly-constrained, but provide interpretable biophysical parameters. They offer starting points which can be embellished upon with machine learning tricks-of-the-trade using deep learning frameworks[96,1].

4.3 Model interpretability is key to moving forward

Biological experiments are noisy but often treated as ground truth for both training and testing. Improved predictions on unvalidated experimental benchmark datasets may not necessarily serve as a reliable way of comparing model performance (Fig. 4.2a). Interpreting models can therefore help to elucidate whether a DNN has learned new biology not captured by previous methods or has gained an advantage by learning correlated features that are indirectly related, such as technical biases of an experiment. Since binary classification tasks require discrimination of sequences between the

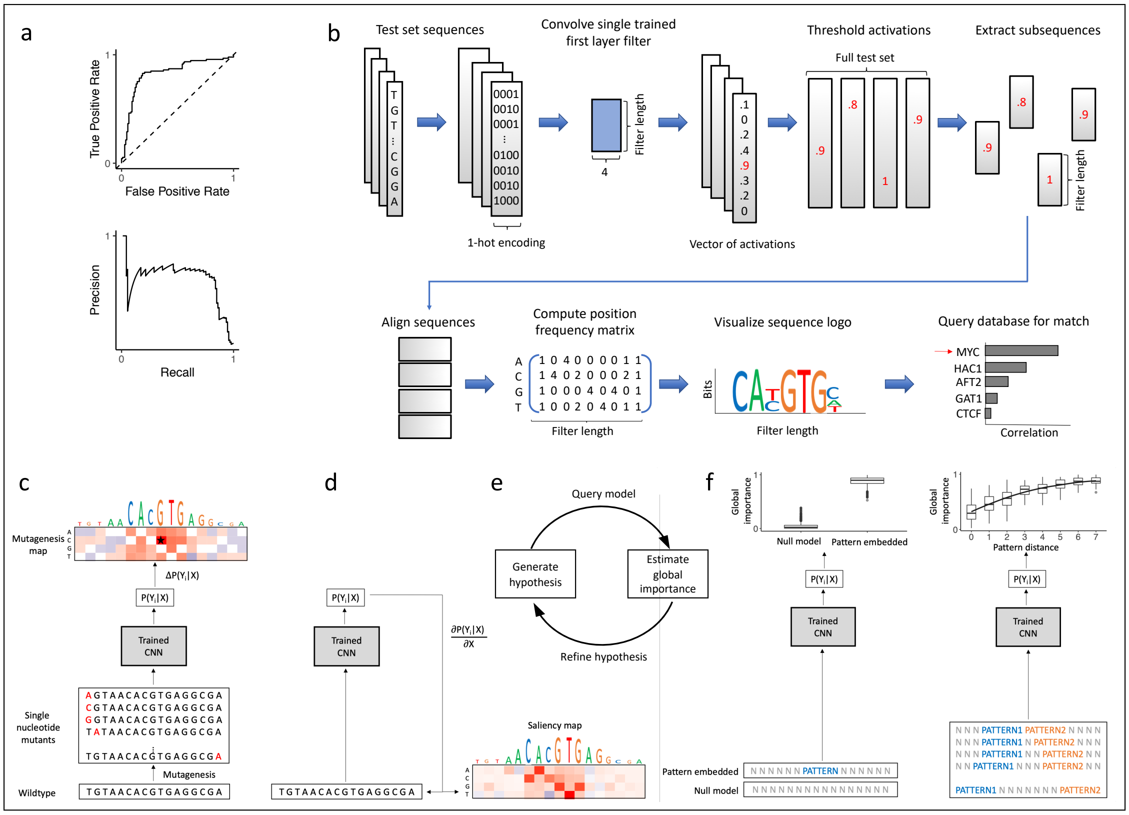


Figure 4.2: Overview of model evaluation and interpretability. a) Model performance is assessed using the receiver-operating characteristic curve (top) or precision-recall curve (bottom). b) Visualizing CNN filters helps to understand learned representations. This can be achieved by scanning each filter across test set sequences, extracting subsequences (the length of the filter) centered on sufficiently large activations (above some threshold), aligning the subsequences, from which a position frequency matrix can be constructed and visualized as a sequence logo. Motif comparison search tools, such as Tomtom, can compare motif similarity against a database of previously-annotated motifs. c) *In silico* mutagenesis provides a single-nucleotide resolution map consisting of an importance score for each nucleotide variant at each position by calculating the difference in predicted values between a given wildtype sequence and new sequences with all possible single nucleotide variants. d) Gradient-based attribution methods analogously provide a single-nucleotide resolution map by calculating the derivative of the output (or logits) of a given class with respect to the inputs. e) A CNN can be used to generate and refine biological hypotheses by querying the model with a set of carefully chosen sequence models and estimating the global importance. f) Given a representative null background model (light gray N nucleotides) the global importance of a pattern (left panel) or spacing between patterns (right panel) may be estimated by querying the trained CNN with a sufficiently-large corpus of randomized, null sequences, each with an instance containing the feature as well as a matched instance without the feature. Such a method allows practitioners to quantitatively test a variety of biological hypotheses while controlling for unwanted confounders.

positive and negative class, interpretability can also help to diagnose whether the DNN has learned poor features that directly result from a poor choice of negative sequences. In genomics, the main approaches to interpret a CNN are through visualizing convolutional filters[6,66], attribution methods[128,150,123], and more recently *in silico* experiments[9,71].

4.3.1 Filter visualization

First layer filters can be directly visualized as sequence logos via activation-based alignments (Fig. 4.2b). This representation makes it possible to compare filter representations against known databases of motifs, such as JASPAR[34], using Tomtom[43], a motif comparison search tool. Filter visualization has been a popular interpretability approach to support that a CNN has learned meaningful biology[6,66,16,8,102,26,50,88]. There are many drawbacks to filter interpretation, including the challenge in quantifying the importance of the feature and how to relate the features to model prediction. Due to the complex dependencies with other filters within and across layers, off-the-shelf CNNs may not necessarily learn complete motif representations in first layer filters. Representations learned by CNNs are strongly influenced by many factors, including inductive biases provided by architectural constraints[73,98], activation functions[72], and training procedure[55]. Hence, filter analysis should only be employed when a model is explicitly trained to learn interpretable motif representations. A more thorough discussion of the benefits and drawbacks to visualizing first layer filters can be found in[73,98].

4.3.2 Attribution methods

In genomics, attribution methods – such as *in silico* mutagenesis[154,66], saliency maps[128], integrated gradients[134], DeepLift[123], and DeepSHAP[86] – provide a single-nucleotide resolution map consisting of an importance score for each nucleotide variant at each position that are directly linked to predictions (Figs. 4.2, c-d). In practice, attribution methods have been utilized to validate that a model has learned representations that resemble known motifs in TF binding[65,9], chromatin accessibility[6,154,66], RNA-protein interactions[38]. There are other interpretability methods that have been developed for genomics, including maximum entropy-based sampling[33] and occlusion experiments[9,150], as well as many other methods that have not yet been thoroughly explored in genomics[150,109,118?].

Limitations Attribution methods are *local* interpretability methods that provide feature importance of individual nucleotides for a single sequence. Hence many attribution maps have to be observed on an individual basis to deduce what features the network has learned *globally* at a population-level. This can be challenging, because attribution methods tend to produce noisy representations with spurious importance scores for seemingly arbitrary nucleotides. TF-MoDISco aims to simplify this process by clustering attribution scores[124]. Even still, attribution methods are unable to quantify the effect that a whole putative motif (not just one nucleotide) has on model predictions. Ongoing research is exploring to what extent we can trust attribution methods[3,2,130?].

Second-order interactions The previously described attribution methods are first-order interpretability methods, revealing the independent contribution of single nucleotide variants in a sequence. There has been growing interest in uncovering interactions between two nucleotide positions, including second-order *in silico* mutagenesis[71], integrated Hessians[61], self-attention networks[141], filter visualization in deeper layers[88], and other gradient-based methods[16,40,84].

4.3.3 Global importance analysis

Global importance analysis (GIA) provides a framework to quantify the effect size of such putative motifs as well as the ability to map specific functions learned by a DNN[?]. GIA performs *in silico* experiments where synthetic sequences are designed with embedded hypothesis patterns while the other positions are randomized by sampling a null sequence model (Fig. 4.2f). By averaging the predictions of these synthetic sequences, GIA quantifies the average effect of the embedded patterns while marginalizing out the contributions of the other positions. Important to this approach is an appropriate null sequence model that minimizes distributional shift between the synthetic sequences and the experimental data. Prior knowledge is critical to determine the null model. For instance, Koo et al. employed GIA to find that the number of motifs, spacing between motifs, relative positions, and aspects of RNA secondary structure were significant learned features in their DNN[71]. More recently, Avsec et al. employed GIA to understand motif syntax, including cooperative associations and positional periodicity[9]. We envision GIA will play a critical role in testing hypotheses of what DNNs have learned, moving beyond speculation from observing putative features in attribution maps and individual filters.

4.4 Conclusion

The timely advances in deep learning and genomics have made research at this intersection progress at a rapid pace. Improvements to architecture and interpretability have been key to the synergy. Yet there are many pressing avenues that are beginning to emerge, including end-to-end models, generative modeling, causal inference, variant effect prediction, and robustness properties.

End-to-end models Framing TF binding as a binary classification task is limiting, because peak calling is noisy and the read distributions themselves can be informative of the underlying biological signals. Recent applications have bypassed the peak calling preprocessing step altogether, directly predicting read distributions from sequence[65,9]. This allows the DNN to learn how to discriminate peaks. Interpreting these so-called end-to-end DNNs may help to isolate biological signals from experimental noise.

Generative modeling In contrast to supervised representation learning, which are informed only through the task they are trained on, unsupervised representations learned with deep generative models, such as generative adversarial networks[?] and variational autoencoders[?], can reveal latent structure of the data on a low dimensional manifold. Deep generative models are an active research area in protein sequence modeling[?][?] but is largely lagging for regulatory genomic sequences. Applications for proteins demonstrate that deep generative models could potentially help to study evolution of sequences across phylogenies[?] and design new sequences with desired properties[?].

Causal inference A fundamental assumption in the field of causal inference is ignorability, for which domain-knowledge is employed to build structural causal graphs which capture relevant data dependencies and explicitly formulate model assumptions to ensure there are no unmeasured confounders. On the other hand, highly-parameterized DNNs which estimate complex functions from rich functional classes run counter to such explicit formulations. A hallmark technique to ensure ignorability is the randomized control trial (RCT). Experiments performed in regulatory genomics, such as massively parallel reporter assays[69], are by design RCTs given a sufficiently large library. While costly, such experiments provide valuable insight into the underlying causal mechanism dictating sequence-function relationships. An alternative to physically performing these experiments is to simulate them *in silico*, namely by performing global importance analysis. To do so, however, requires robust models which accurately learn the functional relationships under consideration. We therefore prioritize the collaboration between bench scientists and computational scientists such that hypotheses generated *in silico* may be validated *in vivo* and a feedback loop may be utilized to develop better models (Fig. 4.2e). DNNs that accurately model the true causal effects are more robust to distribution shifts and improve generalizability[?]. The same may be said when integrating multiple data modalities. For instance, adjusting for confounders such as chromatin accessibility is critical for learning a generalizable function across cell types. Subsequent improved design of models will reduce costs associated with experimental validation, accelerate hypothesis generation and refinement, and provide more accurate discovery of causal biological mechanisms.

Robustness and interpretability By learning sequence-function relationships, a trained DNN can be used to score the effect that disease-associated variants have on the phenotype that it was trained on[6,66,153,58,16,65,152]. This of course assumes that the model has learned an invariant causal representation which is generalizeable beyond the data that it was trained on. Demonstration of out-of-distribution generalization performance has been limiting due to a lack of reliable benchmark datasets with ground truth. In other domains, it has been shown that small, targeted perturbations to the inputs, so-called adversarial examples[?] , generated by an adversary whose sole mission is to trick the classifier, can result in highly unreliable predictions. This has resurrected the field of robust machine learning which focuses on the trustworthiness of model predictions[?] . Counterintuitively, high performing DNNs do not necessarily yield reliable attribution scores[139,7], even in genomics[?] . This raises a red flag that we should not blindly trust model predictions on variant effects just because they generalize well on held-out test data generated from the same distribution, which share the same biases. It has been demonstrated that adversarial training, which incorporates adversarial examples during training, not only leads to improved robustness properties but also improved interpretability[55?] . Although adversarial examples is not a meaningful phenomenon in genomics, their potential for improving the robustness and interpretability properties of DNNs through adversarial training makes them an exciting area of exploration. A thorough evaluation and understanding of how training procedure, incorporation of biophysical priors, and the various advances in DNN architectures all influence model robustness and interpretability is an avenue for future research.

Beyond validation – discovering new biology   Deep learning offers a new paradigm for data analysis in genomics. As powerful function approximators, DNNs can be employed to challenge our underlying assumptions made by traditional (non-deep learning) models. To make meaningful contributions, however, we need to move beyond performance comparisons on benchmark datasets. Through model interpretation, we can identify what novel features drive performance gains. In practice, we believe that a combination of interpretability methods – such as first-order and second-order attribution methods and filter visualization – can collectively help to generate hypotheses of putative features and their syntax. This strategy should compensate for the failures of any individual approach. As a follow up, global importance analysis can be employed to quantify the effect size of putative features and also tease out specific functional relationships of the features, including positional dependence, sequence context, and higher-order interactions. We recommend training various DNNs – ranging from models designed to be highly expressive to models designed to learn interpretable representations – to identify features that are robust across models and initializations. Averaging an ensemble of models is a powerful approach to improve performance and it can also be extended to improve interpretability. Interpreting model predictions is a powerful approach to suggest biological insights and generate hypotheses. The patterns they learn are not proof of biological mechanisms, so any new insights should be followed with experimental validation.

A

Supplement

A.1 Chapter 1: Supplementary Figures

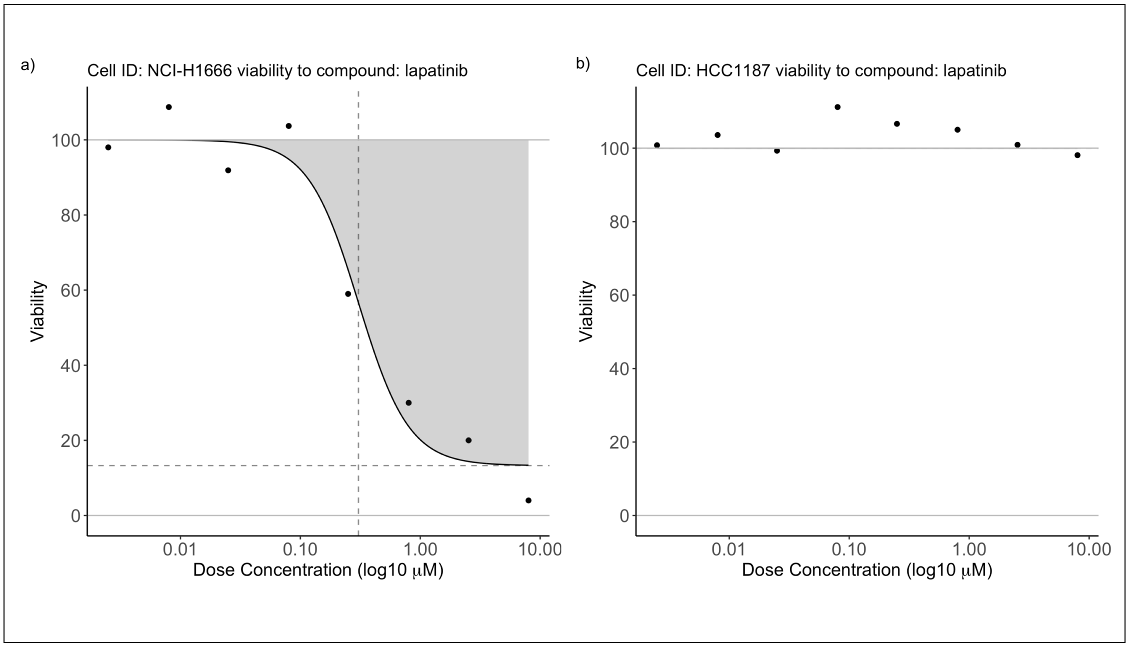


Figure S1: Quantification of cell response to treatment is obtained from a dose response curve in which variable dosage concentrations are administered and the cell relative viability (relative to control wells, viability depicted on Y-axis) is recorded at each dosage concentration (X-axis). Cells exhibiting high sensitivity to a drug attain higher values of area-above-the-curve (**a**, AAC=0.36) while resistant cells maintain high levels of cell relative viability regardless of dose concentration and thus low values of AAC (**b**, AAC=0). Dashed lines indicate estimated EC_{50} (vertical) and E_{∞} (horizontal), two alternative measures of cell sensitivity, however these measures may not exist for resistant dose response curves (i.e. $EC_{50} = \infty$ in **b**).

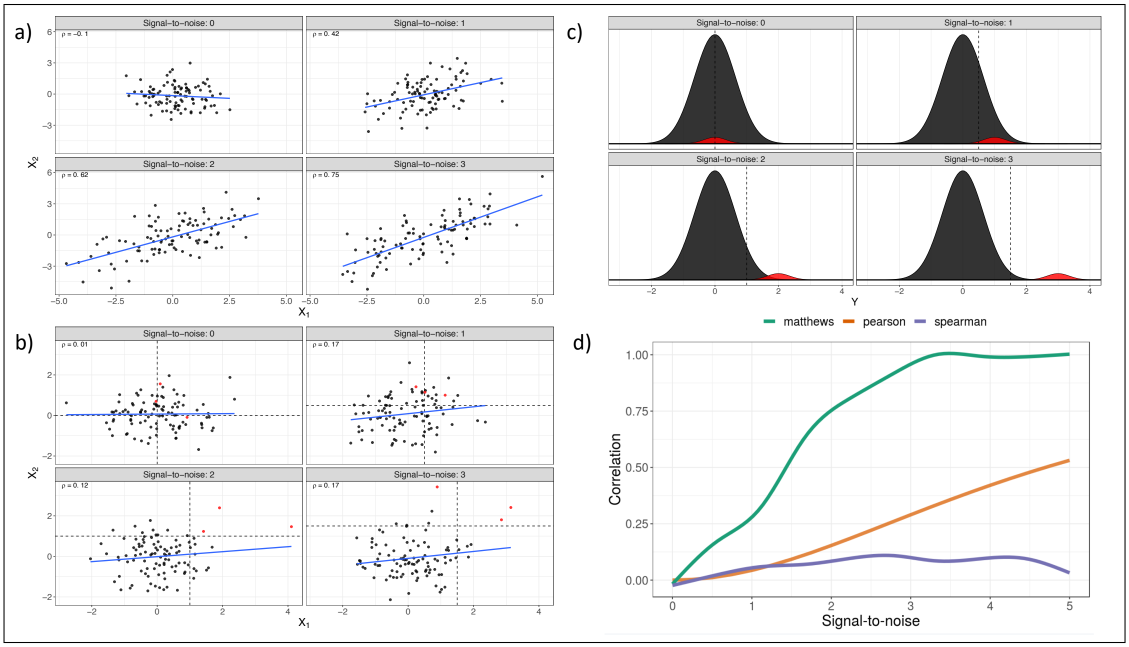


Figure S2: a) Two normally-distributed random variables X_1 and X_2 which follow a bivariate normal distribution. Four signal-to-noise regimes depict the relationship between the signal-to-noise ratio (covariance divided by uncorrelated variance) and the Pearson correlation (covariance divided by total variance). A single simulation repetition is pictured. **b)** Two random variables which instead follow a bivariate two-component mixture of normals. Color indicates mixture membership and dashed lines denote one-half the signal, a naïve method for classifying points to a given mixture and used for calculating Matthew's binary correlation. **c)** The generating mixture distributions under the same four signal-to-noise regimes. Signal is defined as the distance between the means of the distributions and noise is the total uncorrelated variance. Again dashed lines represent one-half the signal. **d)** Three measures of agreement (Pearson correlation, Spearman rank, and Matthew's correlation) for two random variables simulated from the generating distributions in c. Pearson correlation is derived analytically (see Supplementary Methods) while Matthew's correlation and Spearman rank are smoothed across five repetitions.

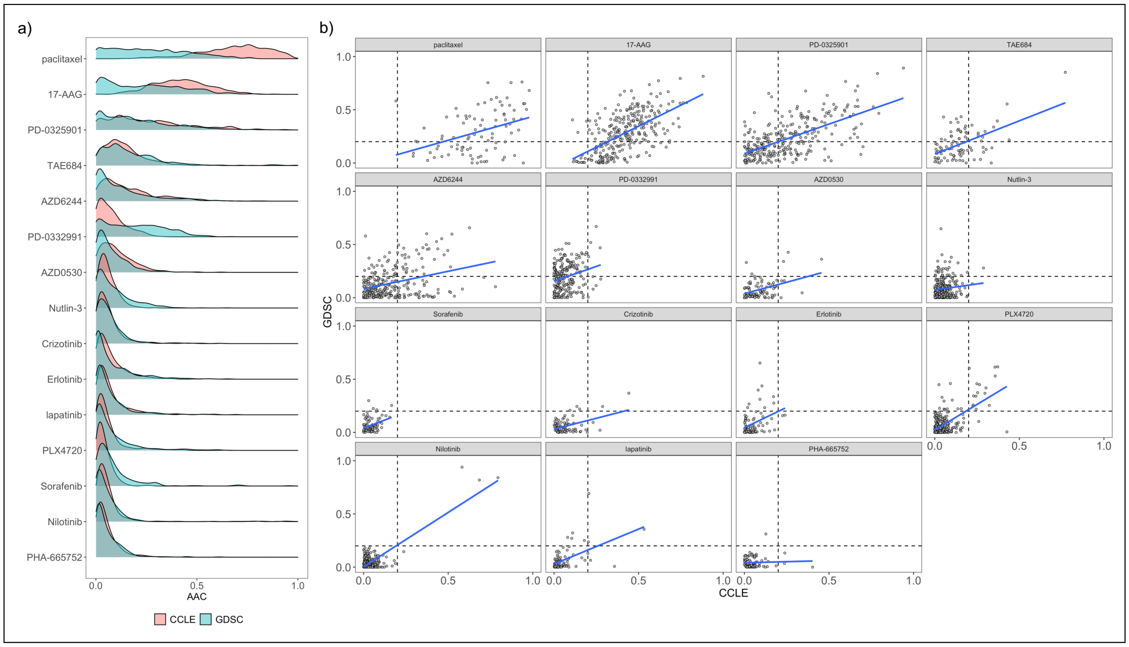


Figure S3: a) Distributions of drug response (AAC) appear either broad (top three drugs) or targeted (bottom twelve drugs) in nature, regardless of study (CCLE: salmon, GDSC: green), suggesting a broad sense of agreement between studies. **b)** AACs computed across common dosage ranges for cell lines tested in both studies (points) and drugs (panels) suggest moderate agreement as measured by Pearson correlation (blue lines) for many drugs. Following[113], AACs may be naively denoted as sensitive if AAC greater than 0.2 (dashed horizontal and vertical lines), else resistant. After such thresholding, binary correlation measures (such as Matthew's correlation coefficient or the log-odds ratio) will more accurately measure agreement given the underlying data generating mechanism consists of resistant and sensitive mixture distributions, as in the case of the targeted drugs (e.g. Nilotinib). We employ a mixture modeling approach to estimate study and drug-specific thresholds since no universal threshold is evident for all targeted drugs across both studies.

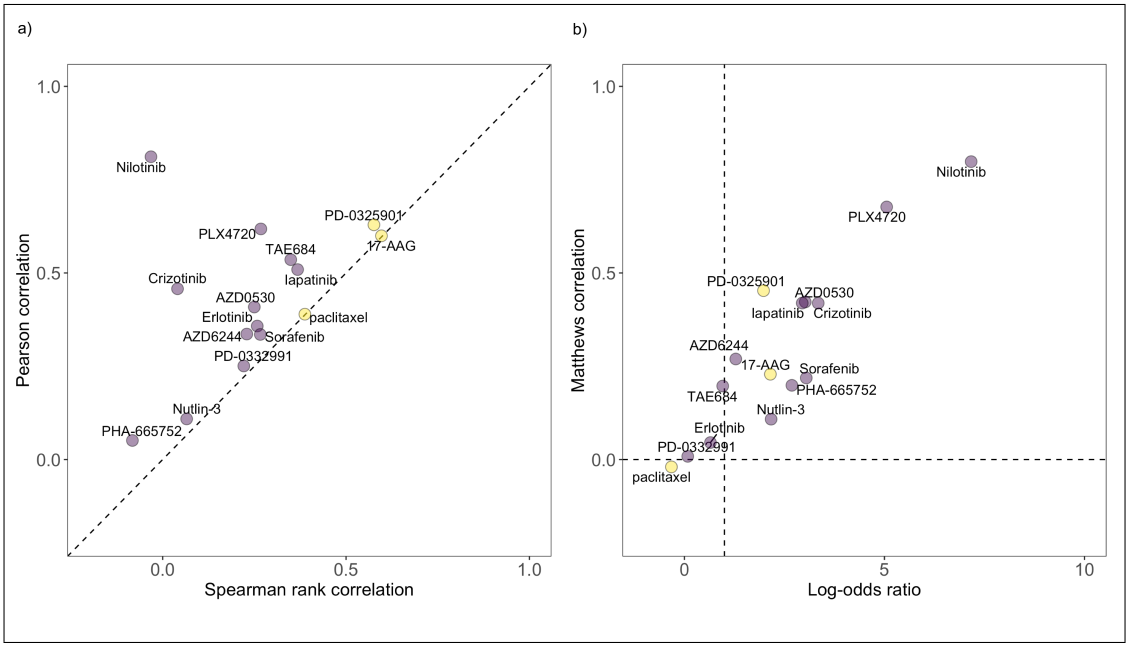


Figure S4: Different association measures suggest different levels of study agreement based upon distributions of cell response (AAC). **a)** Study agreement for broad effect drugs (yellow points) measured with either Pearson correlation or Spearman rank will yield concordant assessments due to a valid univariate monotonicity assumption. In these cases reasonably-high study agreement is observed. On the other hand, when drug mechanisms of action are more targeted in nature, one expects a small proportion of sensitive cells to respond and many others to exhibit resistance. In these cases (purple points) the data generating distributions are comprised of a sensitive component and a resistant component, and thus Spearman rank correlation and Pearson correlation will yield differing conclusions of concordance due to a violation of their assumptions. An association measure which takes into account the underlying binary nature of the data is more appropriate and yields consistent conclusions of agreement (**b**). The odds ratio (log transformed on the X-axis) or Matthew's correlation coefficient are two such binary measures which may be used for targeted drugs.

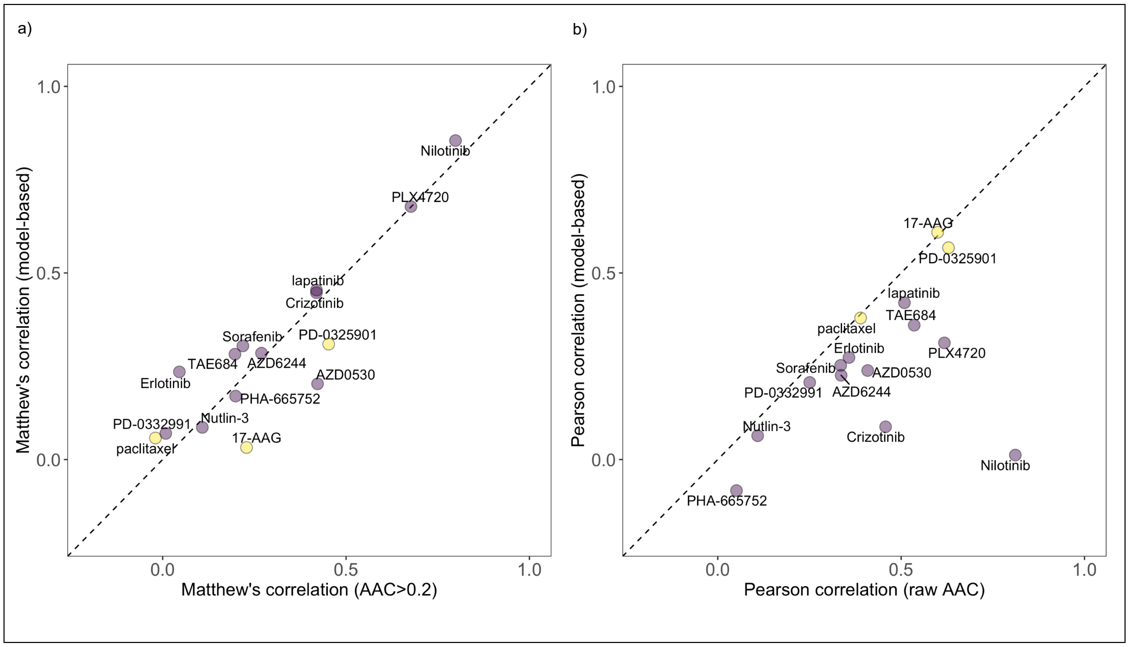


Figure S5: a) Binary agreement as measured with Matthew's correlation coefficient is improved for 9 of 12 targeted drugs (purple points) which were tested in common between the CCLE and GDSC studies. AAC is binarized into resistant and sensitive components using a threshold of AAC$>$0.2 (X-axis) or an estimated posterior probability of sensitivity$>$0.5 (Y-axis) based upon the model fits. Yellow points are broad effect drugs and thus measuring agreement with MCC is inappropriate. On the other hand, agreements as measured by Pearson's correlation coefficient are little changed for broad effect drugs (**b**).

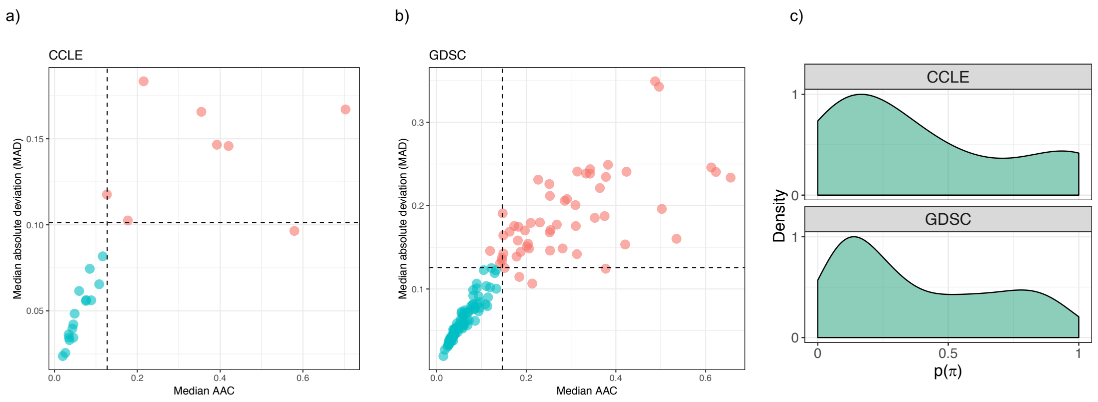


Figure S6: The median AAC computed per drug (X-axis) along with the median absolute deviation (Y-axis) may be used to reasonably classify drugs into drug type (broad effect: red, targeted: blue) for both the **a)** CCLE and **b)** GDSC studies. Dashed lines represent the 60^{th} percentile and are used to initialize the model fit for the expectation-maximization algorithm. **c)** Both CCLE and GDSC exhibit bimodality in the drug-type priors based upon initialization using the median AAC and median absolute deviation results.

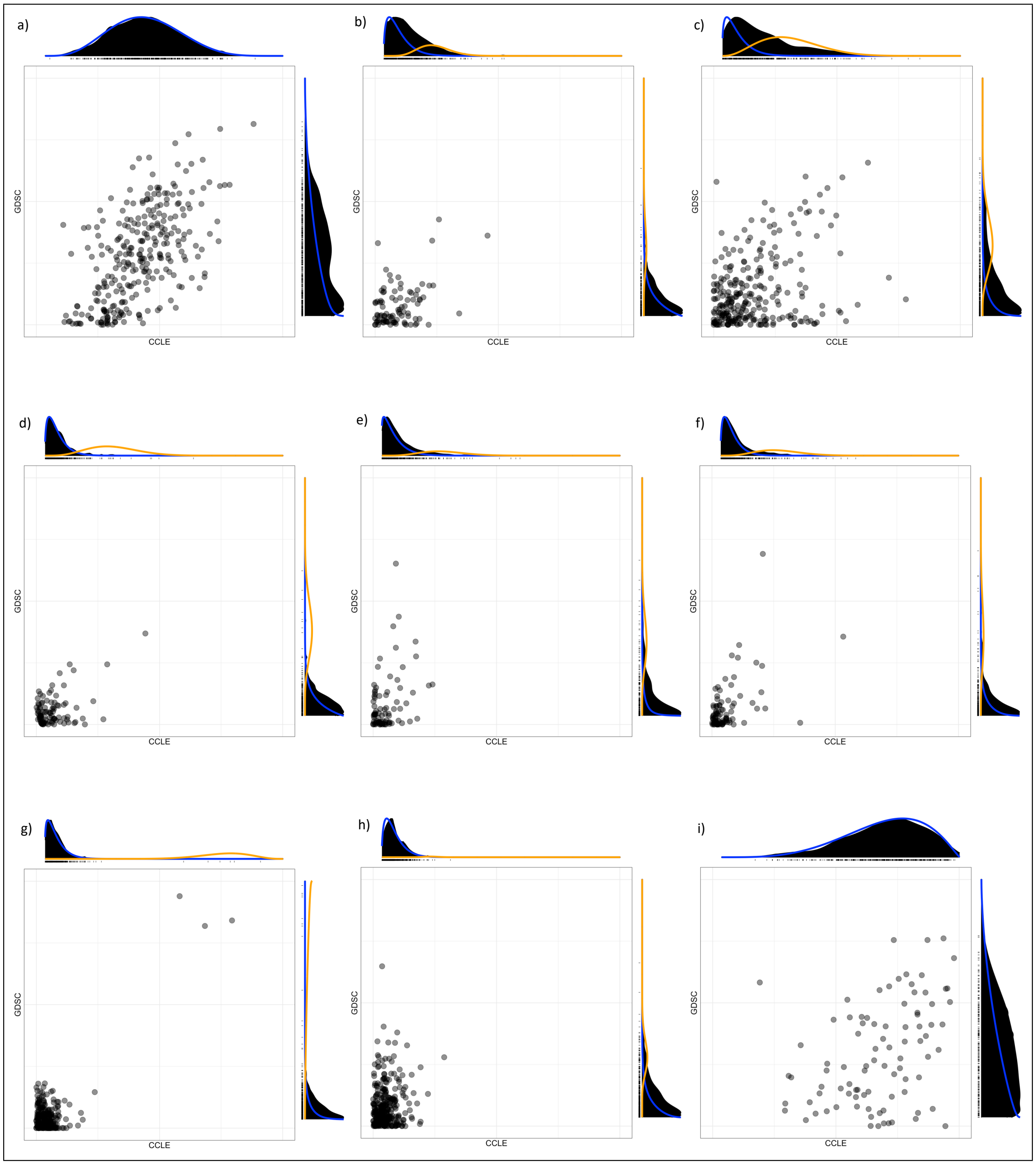


Figure S7: Estimated posterior distributions of cell sensitivity for CCLE (X-axis) and GDSC (Y-axis) for drug and cell lines present in both datasets. Distributions are fit using all available cell lines (tick marks provided alongside densities). **a)** 17-AAG, **b)** AZD0530, **c)** AZD6244, **d)** Crizotinib, **e)** Erlotinib, **f)** Lapatinib, **g)** Nilotinib, **h)** Nutlin-3, **i)** Paclitaxel, **j)** PD0325901, **k)** PD0332991, **l)** PHA665752, **m)** PLX4720, **n)** Sorafenib, **o)** TAE684 (continued on following figure).

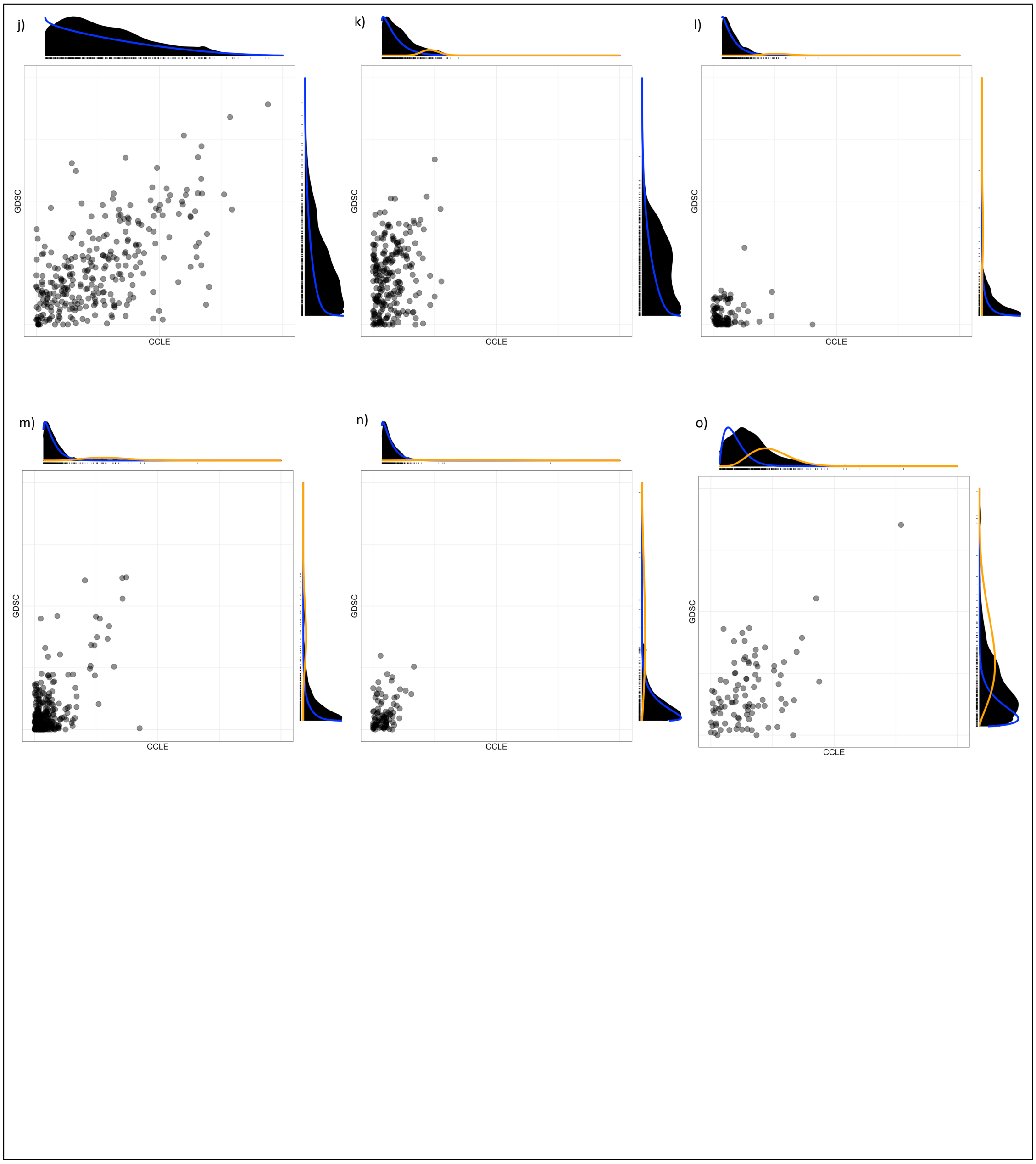


Figure S8: Estimated posterior distributions continued.

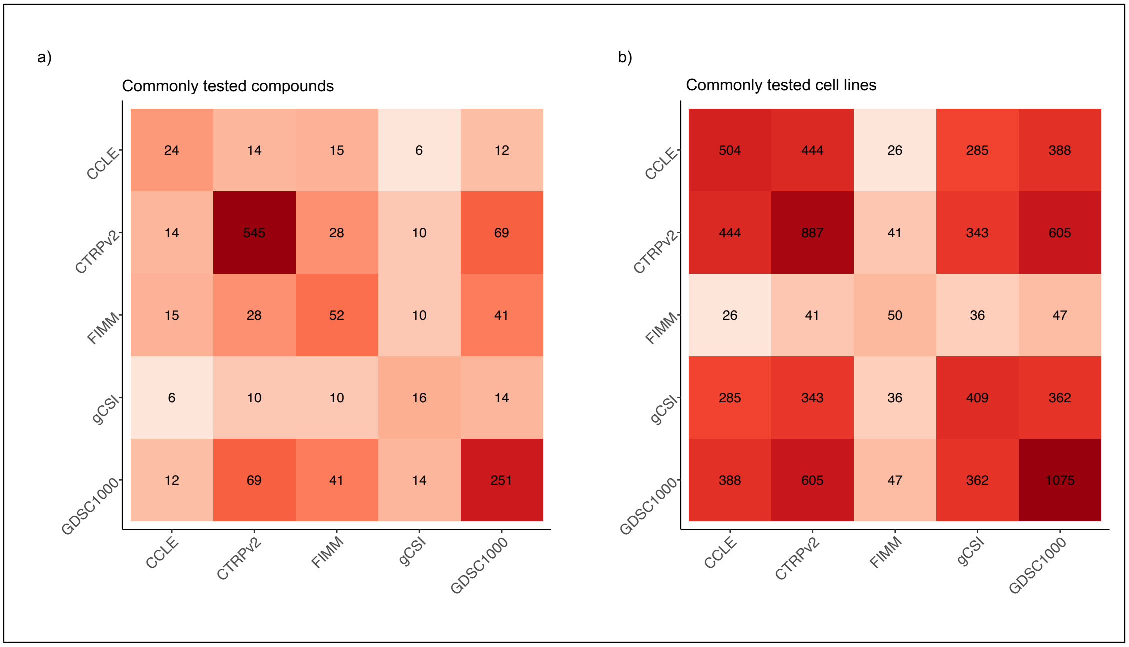


Figure S9: Heatmaps indicating the number of drugs (**a**) and cell lines (**b**) tested in common between any two studies (CCLE, GDSC1000, CTRPv2, FIMM, gCSI).

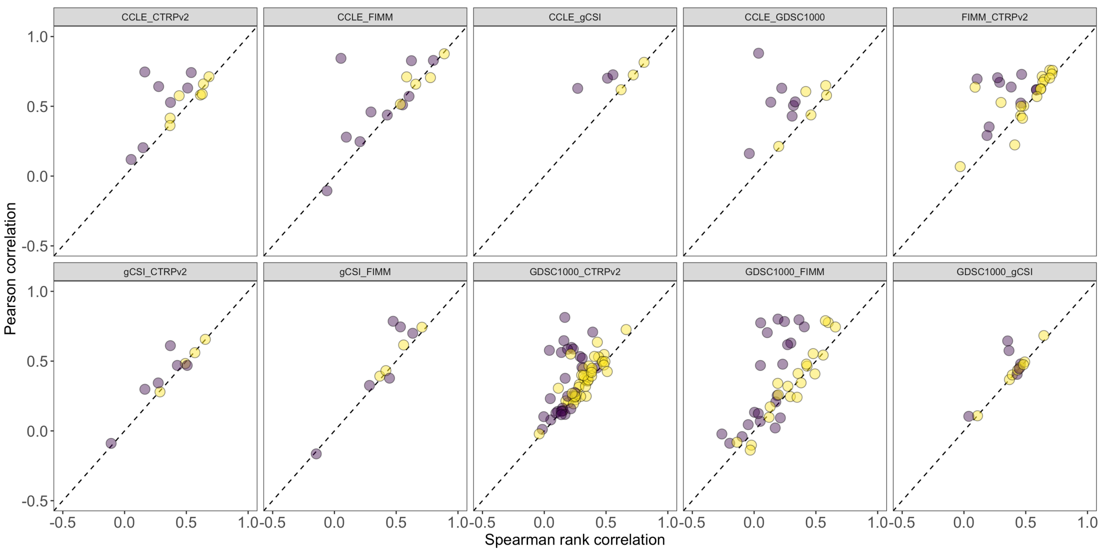


Figure S10: Spearman rank correlation (X-axis) and Pearson correlation (Y-axis) between the raw AAC values for all drugs and cell lines tested in common between any two studies (CCLE, GDSC1000, CTRPv2, FIMM, gCSI). Yellow coloring represents drugs estimated to be broad effect, purple coloring represents drugs estimated to be targeted.

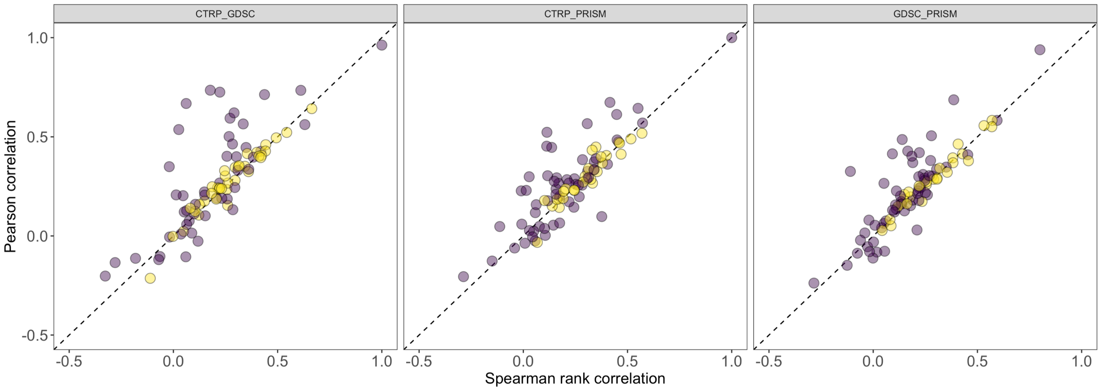


Figure S11: Spearman rank correlation (X-axis) and Pearson correlation (Y-axis) between the raw AAC values for drugs and cell lines tested in common between all three studies (CTRP, GDSC, PRISM). Yellow coloring represents drugs estimated to be broad effect, purple coloring represents drugs estimated to be targeted.

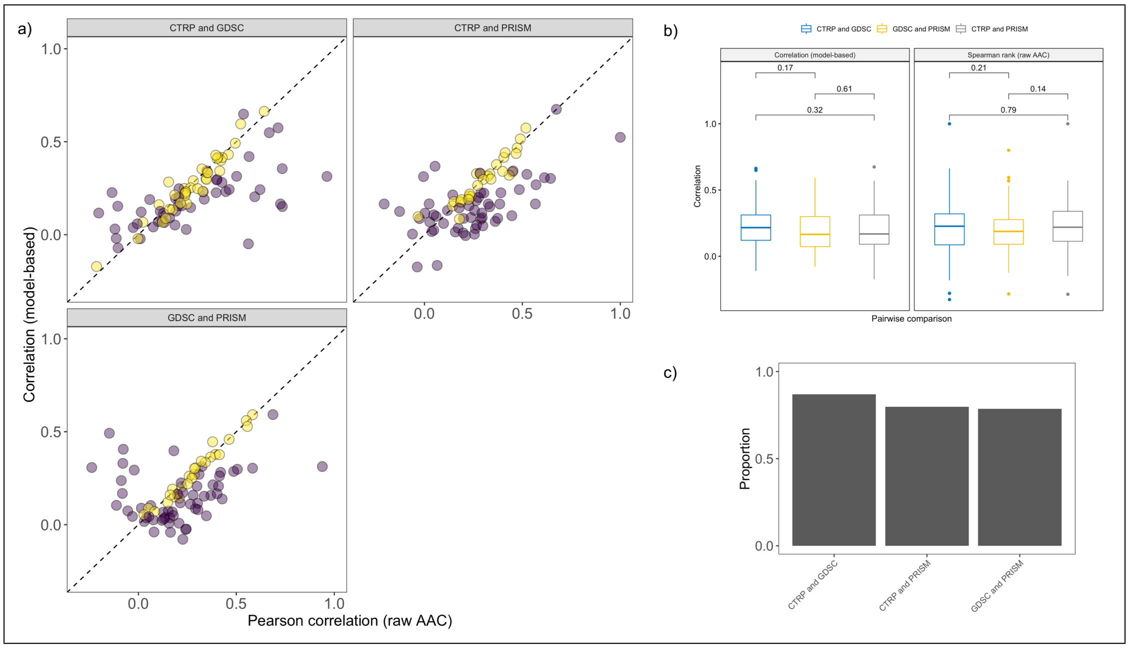


Figure S12: a) Pearson correlation between raw AAC values (X-axis) for all drugs and cell lines tested in common between all three studies (CTRP, GDSC, and PRISM). The Y-axis represents either the Pearson correlation (yellow points, broad effect drugs) or the Matthew's correlation coefficient (purple points, targeted drugs). Panels represent pairwise comparisons for AAC values calculated over common dosage concentrations for all cell lines/drugs tested in common between all three studies (AAC values provided by [24]). Coloring indicates whether the drug was estimated to be targeted (purple, estimated posterior probability of drug targetedness > posterior probability of drug being broad effect; yellow). Targeted drugs exhibit improved concordance when considering an appropriate measure of agreement (Matthew's correlation coefficient, Y-axis purple points), and unchanged concordance for broad effect drugs (Spearman rank correlation for yellow points). **b)** Boxplots of Spearman rank correlation (right panel) for the three pairwise comparisons using the AAC values provided by [24] highlight several drugs with negative correlation. Reporting a binary measure of agreement improves the low correlations as assumptions are violated when the underlying data distribution is composed of a mixture of distributions (targeted drugs) rather than a single univariate distribution (broad effect drugs). Reported correlation in the left-hand panel is Spearman correlation for broad effect drugs and Matthew's correlation coefficient for targeted drugs. There is no evident decrease in overall agreement based upon the estimated posterior sensitivities (little-to-no change in P-values as provided above the boxplots). **c)** Proportion of drugs with estimated posterior drug type (broad effect or targeted) equal between the pairwise comparisons. A large proportion (>0.75) of the drugs tested in common exhibit similar degrees of drug targetedness for all pairwise comparisons.

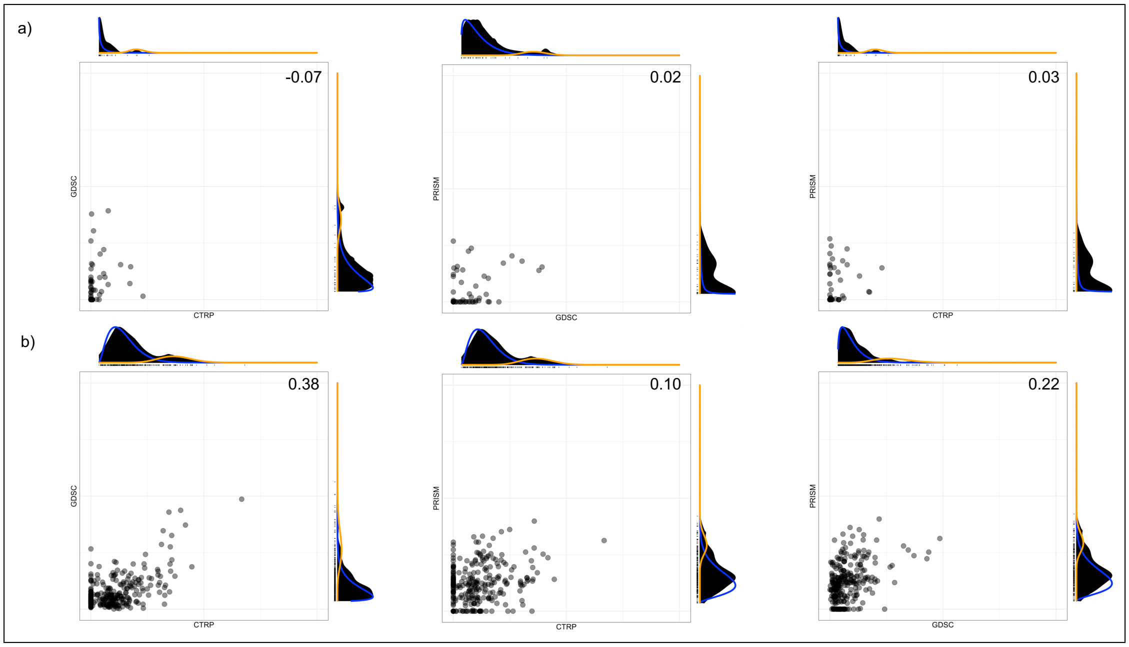


Figure S13: **a)** Estimated posterior distributions for the drug lapatinib when fitting the model using only drugs and cell lines present in each of PRISM, CTRP, and GDSC (**a**) as opposed to considering the datasets in their entirety (**b**). Each point represents the AAC value for that dataset and the number in the upper corner of each plot is the mean correlation calculated based on 1000 MC samplings from the estimated posterior distributions. The number of cells sensitive to lapatinib is very low due to the highly targeted nature of the drug, and essentially zero by restricting the analysis to only those cell lines present in the 3-way intersection. This results in low levels of estimated agreement. On the other hand, agreement is improved when considering the full datasets as there are a sufficient number of sensitive cells. We thank[24] for kindly sharing the data necessary for **a** but were unable to replicate the values using the full dataset, hence there is no one-to-one correspondence between points in **a** and **b**.

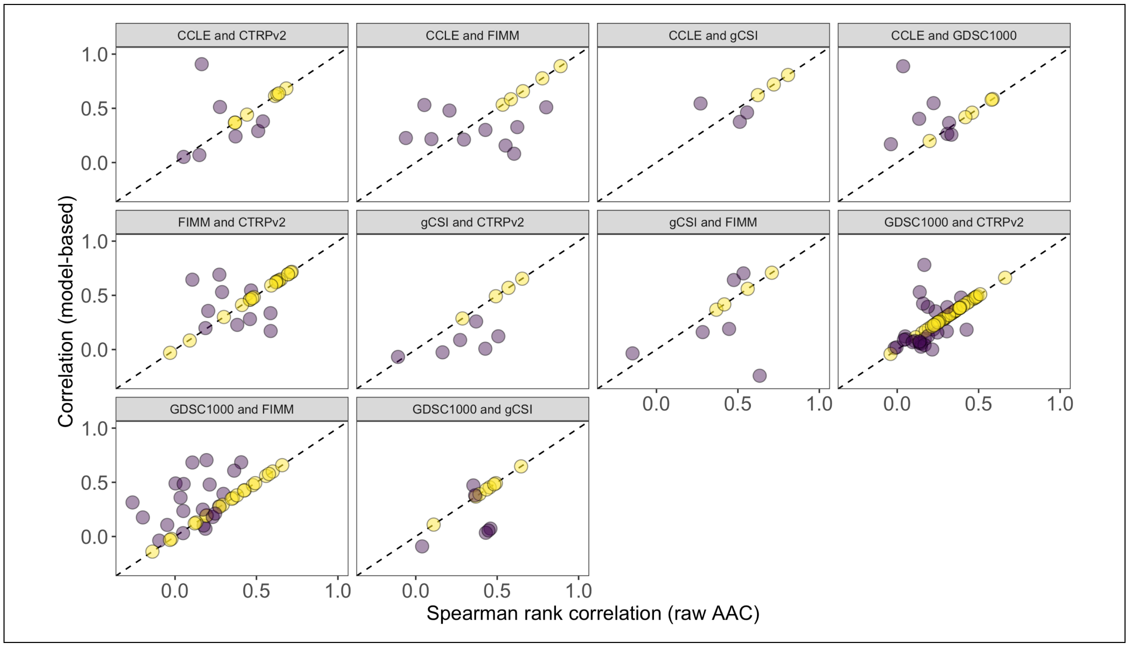


Figure S14: Spearman rank correlation between the raw AAC values (X-axis) for all drugs and cell lines tested in common between any two studies (CCLE, GDSC1000, CTRPv2, FIMM, gCSI). The Y-axis represents either the Spearman rank correlation (yellow points, broad effect drugs) or the Matthew's correlation coefficient (purple points, targeted drugs). Panels represent pairwise comparisons for AAC values calculated over common dosage concentrations for all cell lines/drugs tested in common between the two studies. Coloring indicates whether the drug was estimated to be targeted (purple, estimated posterior probability of drug targetedness $>$ 0.5 for both studies) or broad effect (yellow). Targeted drugs exhibit improved concordance when considering an appropriate measure of agreement (Matthew's correlation coefficient, Y-axis purple points), and unchanged concordance for broad effect drugs (Spearman rank correlation for yellow points).

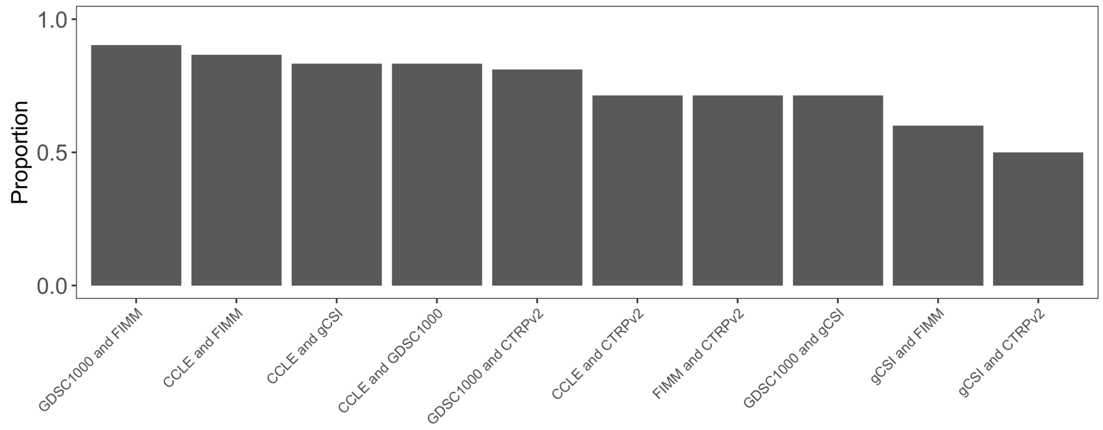


Figure S15: Proportion of drugs tested in common with estimated posterior drug type (broad effect or targeted) equal between any two studies (CCLE, GDSC1000, CTRPv2, FIMM, gCSI). Over 90% of the 41 drugs tested in common between the GDSC1000 and FIMM studies exhibit agreement between estimated posterior drug type, whereas 5 out of 10 drugs tested in common between gCSI and CTRPv2 agree.

A.2 Chapter 2: Supplementary Figures

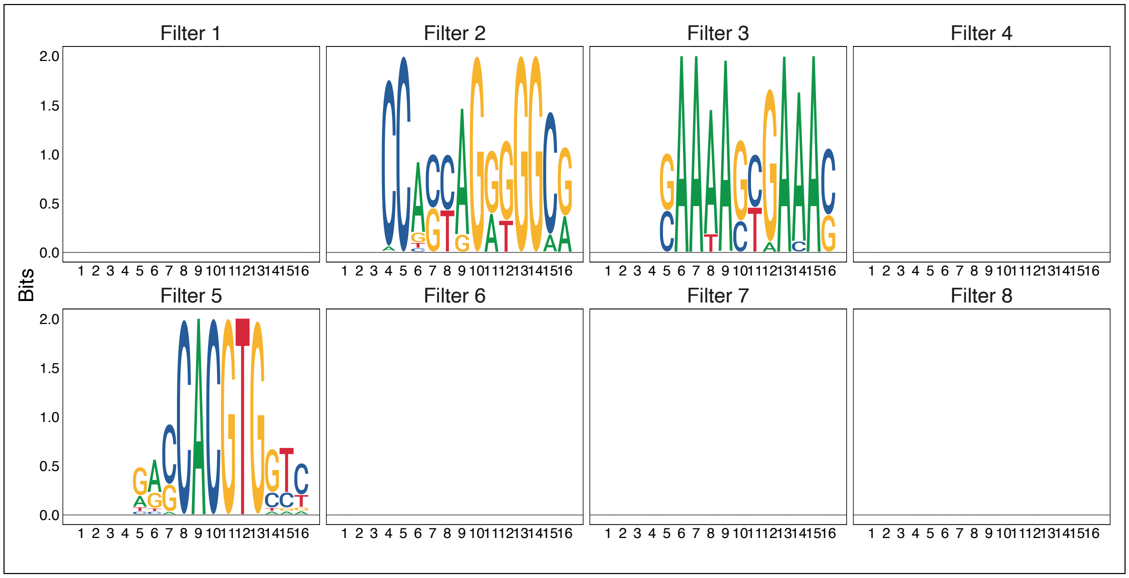
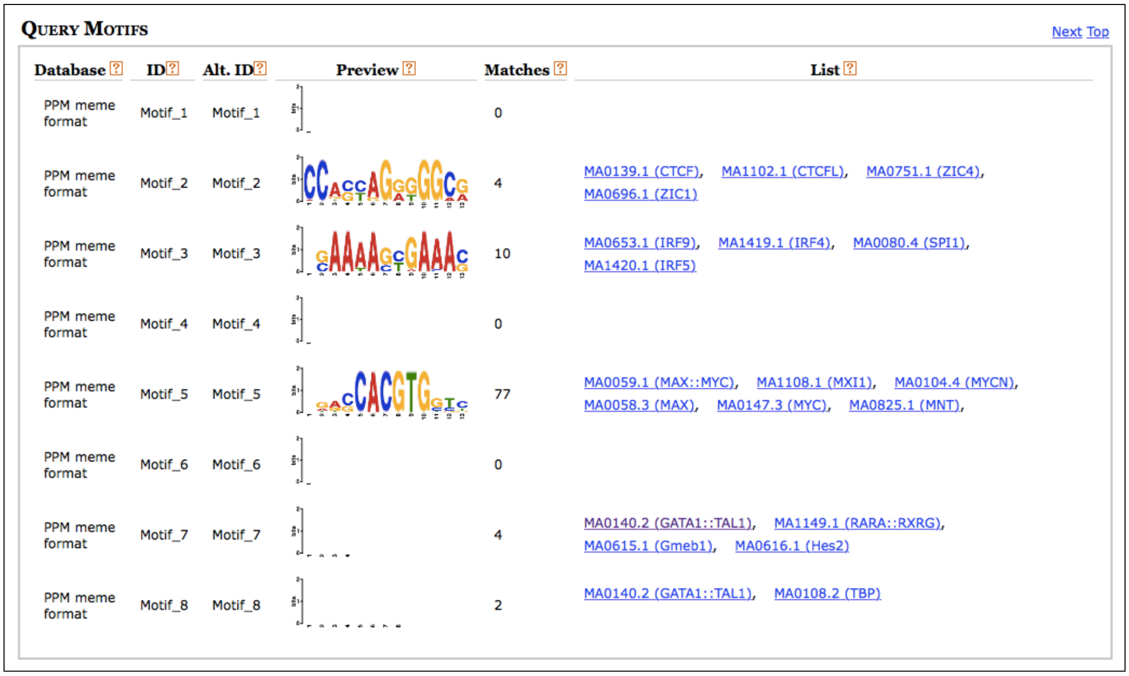


Figure S16: All *de novo* motif simulation filters. All first-layer convolutional filter weights trained on the CTCF-IRF-MYC simulation dataset. Many weights are regularized to zero and thus do not contribute to the model performance.



Query Motifs

Next Top

Database	ID	Alt. ID	Preview	Matches	List
PPM meme format	Motif_1	Motif_1		0	
PPM meme format	Motif_2	Motif_2		4	MA0139.1 (CTCF), MA1102.1 (CTCFL), MA0751.1 (ZIC4), MA0696.1 (ZIC1)
PPM meme format	Motif_3	Motif_3		10	MA0653.1 (IRF9), MA1419.1 (IRF4), MA0080.4 (SPI1), MA1420.1 (IRF5)
PPM meme format	Motif_4	Motif_4		0	
PPM meme format	Motif_5	Motif_5		77	MA0059.1 (MAX::MYC), MA1108.1 (MXI1), MA0104.4 (MYCN), MA0058.3 (MAX), MA0147.3 (MYC), MA0825.1 (MNT),
PPM meme format	Motif_6	Motif_6		0	
PPM meme format	Motif_7	Motif_7		4	MA0140.2 (GATA1::TAL1), MA1149.1 (RARA::RXRG), MA0615.1 (Gmeb1), MA0616.1 (Hes2)
PPM meme format	Motif_8	Motif_8		2	MA0140.2 (GATA1::TAL1), MA0108.2 (TBP)

Figure S17: *de novo* motif simulation Tomtom query. Results from running the Tomtom motif similarity tools[43] querying learned motifs against the 2018 JASPAR Core Non-vertebrates database[67].

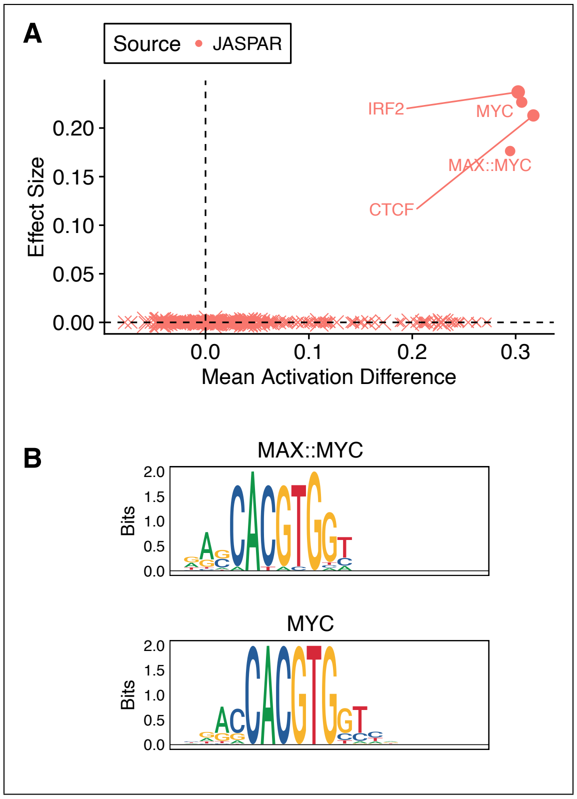
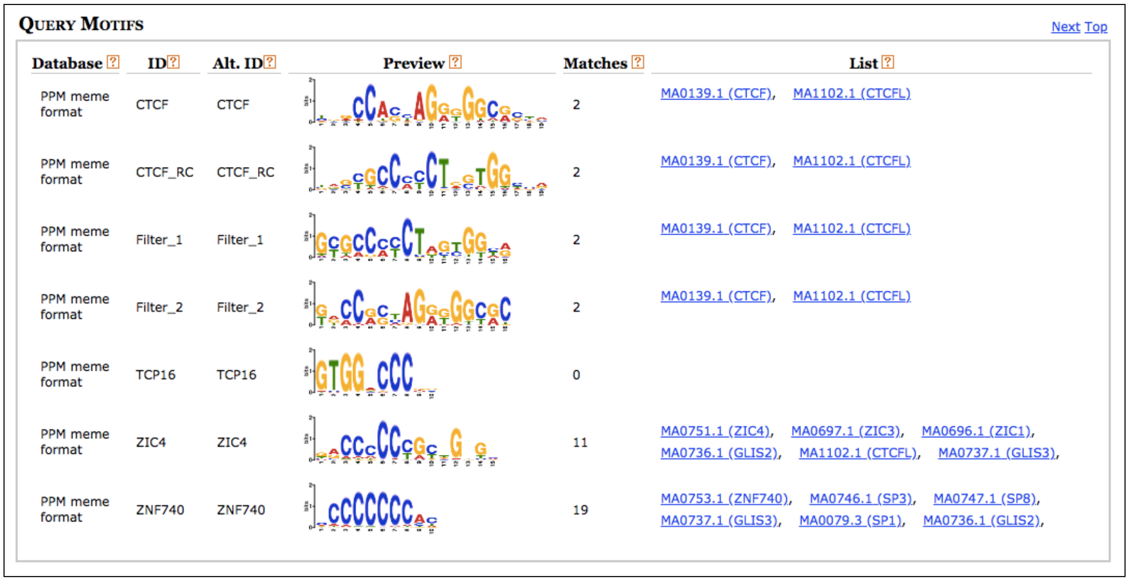


Figure S18: Simulation model 1. All first-layer convolutional filters initialized to 2018 JASPAR Core Non-vertebrates database[67]. All but four motifs exhibit zero effect size and thus may be discarded. Of the four non-zero effect-size motifs, MAX::MYC and MYC are nearly identical and one of the pair could be removed.



Query Motifs

Next Top

Database	ID	Alt. ID	Preview	Matches	List
PPM meme format	CTCF	CTCF		2	MA0139.1 (CTCF), MA1102.1 (CTCFL)
PPM meme format	CTCF_RC	CTCF_RC		2	MA0139.1 (CTCF), MA1102.1 (CTCFL)
PPM meme format	Filter_1	Filter_1		2	MA0139.1 (CTCF), MA1102.1 (CTCFL)
PPM meme format	Filter_2	Filter_2		2	MA0139.1 (CTCF), MA1102.1 (CTCFL)
PPM meme format	TCP16	TCP16		0	
PPM meme format	ZIC4	ZIC4		11	MA0751.1 (ZIC4), MA0697.1 (ZIC3), MA0696.1 (ZIC1), MA0736.1 (GLIS2), MA1102.1 (CTCFL), MA0737.1 (GLIS3),
PPM meme format	ZNF740	ZNF740		19	MA0753.1 (ZNF740), MA0746.1 (SP3), MA0747.1 (SP8), MA0737.1 (GLIS3), MA0079.3 (SP1), MA0736.1 (GLIS2),

Figure S19: ENCODE Gm12878 CTCF TF ChIP-seq data application Tomtom query. Results from running the Tomtom motif similarity tools[43] querying learned motifs against the 2018 JASPAR Core Non-vertebrates database[67].

A.3 Chapter 2: Supplementary Methods

From nucleotide sequences to information gain

A position frequency matrix (PFM) tabulates the frequency of each nucleotide at each position in a set of reads. For example, consider 100 sequences each of length six letters composed of the nucleotides $\{A, C, G, T\}$. A single example sequence could be $ACCTAG$. Under a uniform distribution, one would expect 25 of each nucleotide at each position and the resulting PFM would be:

$$\begin{bmatrix} 25 & 25 & 25 & 25 & 25 & 25 \\ 25 & 25 & 25 & 25 & 25 & 25 \\ 25 & 25 & 25 & 25 & 25 & 25 \\ 25 & 25 & 25 & 25 & 25 & 25 \end{bmatrix}$$

The top-left entry corresponds to the count of A nucleotides in the first position. Dividing by the column sums yields a position probability matrix (PPM), of which each column defines a multinomially-distributed random variable. Denote this r.v. P_c for column c and for the example consider the PFM and resulting PPM below:

$$PFM = \begin{bmatrix} 39 & 4 & 1 & 10 & 25 & 30 \\ 61 & 30 & 94 & 8 & 25 & 20 \\ 0 & 29 & 2 & 70 & 25 & 20 \\ 0 & 37 & 3 & 12 & 25 & 30 \end{bmatrix}$$

$$PPM = \begin{bmatrix} .39 & .04 & .01 & .1 & .25 & .3 \\ .61 & .3 & .94 & .08 & .25 & .2 \\ 0 & .29 & .02 & .7 & .25 & .2 \\ 0 & .37 & .03 & .12 & .25 & .3 \end{bmatrix}$$

If one observed counts as tallied in column three of the PFM, then one might expect to reject a frequentist null hypothesis of uniform nucleotide probability ($P_3 \sim \text{multinomial}(1, .25, .25, .25, .25)$). A likelihood ratio test could be used to perform such inference with the log-likelihood of observing the data under the null taking the form[18]:

$$l(q_A, q_C, q_G, q_T | c_A, c_C, c_G, c_T) = \frac{1}{100} \log 100! - \frac{1}{100} \sum_n \log c_n! + \sum_n \frac{c_n}{100} \log q_n$$

where q_n denotes the null probabilities (0.25) for nucleotide n and c_n denotes the observed count (in the PFM). Following the derivation provided by[122] and making use Stirling's approximation for large N ($\log N! \approx N \log N - N$) gives us:

$$\begin{aligned} l(q_A, q_C, q_G, q_T | c_A, c_C, c_G, c_T) &= \frac{1}{N}(N \log N - N) - \frac{1}{N} \sum_n (c_n \log c_n - c_n) + \sum_n \frac{c_n}{N} \log q_n \\ &= \log N - \sum_n \frac{c_n}{N} \log c_n + \sum_n \frac{c_n}{N} \log q_n \\ &= -\sum_n \frac{c_n}{N} \log \frac{c_n}{N} + \sum_n \frac{c_n}{N} \log q_n \\ &= -\sum_n p_n \log p_n + \sum_n p_n \log q_n \\ &= -\text{KL}(p||q) \end{aligned}$$

Here N denotes the total number of sequences observed (100 in the example) and p_n denotes the observed PPM values for nucleotide $n \in \{A, C, G, T\}$ calculated from the PFM. The last line follows by the definition of Kullback–Leibler (KL) divergence, a measure which has many interpretations including the negative observed log-likelihood, the relative entropy or information content, and in the bayesian paradigm the information gain from using the posterior probabilities p_n relative to q_n. Rearranging the last line reveals how the this quantity takes into account the background distribution:

$$\mathrm{KL}(p||q) = \sum_n p_n \log \frac{p_n}{q_n}$$

This is a weighted sum of the log-odds ($\log \frac{p_n}{q_n}$), terms which define the position weight matrix (PWM). Under a uniform background, the equation above simplifies to $2 - \sum_n p_n \log p_n$ and in our example gives the following PWM:

$$PWM = \begin{bmatrix} 0.64 & -2.64 & -4.64 & -1.32 & 0 & 0.26 \\ 1.29 & 0.26 & 1.91 & -1.64 & 0 & -0.32 \\ -\infty & 0.21 & -3.64 & 1.48 & 0 & -0.32 \\ -\infty & 0.56 & -3.05 & -1.05 & 0 & 0.26 \end{bmatrix}$$

Each entry is upper bounded by 2 due to the choice of the uniform background but the entries are not lower bounded. The absolute magnitudes of the column sums indicate how different the probabilities are and are also unbounded. Pseudo-counts may be added to the PFM or PPM to remedy the $-\infty$ terms. When done so by adding 0.5 to the zero entries in the PFM results in the following PWM and KL divergence:

$$PWM = \begin{bmatrix} 0.64 & -2.64 & -4.64 & -1.32 & 0 & 0.26 \\ 1.29 & 0.26 & 1.91 & -1.64 & 0 & -0.32 \\ -5.64 & 0.21 & -3.64 & 1.48 & 0 & -0.32 \\ -5.64 & 0.56 & -3.05 & -1.05 & 0 & 0.26 \end{bmatrix}$$

$$KL = \begin{bmatrix} 0.98 & 0.244 & 1.58 & 0.65 & 0 & 0.03 \end{bmatrix}$$

As expected the KL divergence is zero in the fifth column, small in the sixth column, and largest in the third and first columns. In the bayesian paradigm, the information gained by using the posterior probabilities depicted in the PPM relative to a uniform prior would be largest in the third column and zero in the fifth. One may lastly obtain the weights shown in the sequence logo plots[116] by multiplying each value in the PPM by its corresponding column in the KL divergence vector. We term this matrix the information gain matrix (IGM). Many other names would suffice including the sequence logo matrix.

$$IGM = \begin{bmatrix} 0.38 & 0.01 & 0.02 & 0.06 & 0 & 0.01 \\ 0.60 & 0.07 & 1.49 & 0.05 & 0 & 0.01 \\ 0.00 & 0.07 & 0.03 & 0.45 & 0 & 0.01 \\ 0.00 & 0.09 & 0.05 & 0.08 & 0 & 0.01 \end{bmatrix}$$

Of course the IGM and PWM contain the same information and are simply rescalings; one may start with a PWM and calculate an IGM and vice versa. Indeed one may transform a PWM/IGM based on background probabilities q_n into a PWM/IGM based on background probabilities r_n simply by backing out the p_n and recalculating the quantity of interest.

Model formulation

We develop our model based on the convolution operator employed in convolutional deep neural network (CNN) architectures[80] considering the PWMs and IGMs as the filters. Since each filter is itself directly interpretable as a sequence motif, we may initialize (or fix) the filter values with those of previously-annotated database motifs directly. We may also attempt to learn the motifs *de novo* given the weights maintain an equivalent interpretation.

Consider a set of J convolutional filters where the j^{th} convolution operator computes the inner product between a weight matrix Ω^j and an observation matrix X_n at each position sliding along user-specified dimensions. These convolutional filters, or patches, are particularly suited for learning local spatial relationships and when employed in deep learning are often stacked together in the hundreds within a single layer, from which the activations produced by each convolutional filter at each position are fed as input into subsequent deeper layers. The choice to use these filters to learn sequence motifs is motivated by the work of[154,6,66]. In these genomics applications each filter Ω^j is a $4 \times L_j$ matrix of weights $\omega^j_{k,l}$ for $k \in \{A, C, G, T\}$ and $l \in 1 : L_j$ convolved upon a one-hot encoding of the input sequence X_n. That is, each X_n is transformed from a nucleotide string of length I_n into a binary matrix of size $4 \times I_n$ with rows corresponding to each base and column i denoting the presence/absence of a nucleotide at position $i \in 1 : I_n$.

We write the model explicitly under the logistic link function (i.e. the sigmoidal activation function), due to binary classification objective as follows:

$$\mathcal{G}(X_n) = P(Y_n = 1 | X_n = x) = \frac{1}{1 + e^{-\beta_0 - \sum_{j=1}^{J} \beta_1^j g(\zeta^j)}}$$

where ζ^j indicates a max-pooled convolution transformation for filter j at location i:

$$\zeta^j = \max_{i \in 1:I_n}(x_i * \Omega^j)$$

and $g(\cdot)$ is some activation function (e.g. linear). The convolution operation is explicitly defined for observation $X_n = x$ at position i as:

$$x_i * \Omega^j = \sum_{l=1}^{L_j} \sum_{k \in \{A,C,G,T\}} \omega^j_{k,l} 1_{x_{i+l-1}=k} + \beta^j_0$$

We again note that it is these Ω^j matrices which contain the weights which collectively capture the sequence motifs. We introduce this notation to motivate the merger of the convolutional filter as a linear predictor: consider the case when $J = 1$ and the weights are all fixed such that $\omega^j_{k,l} = 1$ for $k = C, l \in 1 : L_j$, and 0 else. This trivial filter does nothing more than compute the count of C nucleotides within a L_j-bp sliding window for each observation and assigns the maximum sigmoidal-transformed value as a feature to be fed into a logistic regression model. Interestingly it does this by computing the similarity (cross-correlation) between each test sequence with the all-C motif. There is nothing 'deep' about this model and there are only two parameters to fit, β_0 and β_1, either via an iterative maximum-likelihood approach[92] or gradient descent. Moreover, the interpretations of these values directly correspond to the standard statistical interpretation of regression coefficients.

We now introduce the random variable $Z^j_n \sim \text{Ber}(\pi^j_n)$ to denote the presence ($Z^j_n = 1$) of motif j in sequence X_n. Z^j_n is unobserved and we wish to estimate it via:

$$P(Z^j_n = 1|X_n = x) = g(\zeta^j) = \frac{1}{1 + e^{-\zeta^j}}$$

where ζ^j denotes the max-pooled convolution operator from above. Note that Z^j_n is computed simply by transforming the max-pooled convolution operation to be in $[0, 1]$ via the sigmoid function. Recalling that the outcome variable Y_n is itself a random Bernoulli variable with probability p_n we may condition upon the hidden variables Z_n^j to rewrite equation **??** as:

$$P(Y_n = 1 | X_n = x, Z_n^j = z^j) = \frac{1}{1 + e^{-\beta_0 + \sum_{j=1}^{J} \beta_1^j z^j}}$$

Thus Y_n is simply modeled via a transformation of a linear combination of hidden variables denoting the presence or absence of a given motif based on its maximum subsequence similarity. Of course there is no reason beyond parsimony and interpretability for the need to reduce the j^{th} feature map to a single maximum value (versus, for example, the sum of all elements, or the average within the first half of the sequence and a separate term for the average in the second half). We also note that one may encode any subsequent values from the j^{th} filter as further hidden states sharing the same motif filter to assess the additive impact of motif occurrences in this nested modeling framework. Finally, we note that one may consider a different distribution for the Z_n^j, such as a Poisson to model to the count of a given motif or a linear link to model intensity, however we leave this for future work. We fit all model parameters via stochastic gradient descent in Keras[21] with a batch size of 64.

Encouraging interpretability

Upon successful model training there are three interpretable quantities of interest in our model: 1) The set of filters Ω^j representing motifs, 2) their associated β_1^j model coefficients of estimated effect sizes, and 3) the hidden variables Z_n^j denoting the presence/absence of motif j in sequence n. We present this simple model in the text to encourage the interpretability of each model layer within the CNN terminology. One may opt for a more complicated (deeper) model while still maintaining the interpretation of filters as motifs. We detail the weight constraints and peri-training transformations below, along with some practical considerations for implementation.

IGMs as filters

When representing the motifs (convolutional filters) as information gain matrices we restrict the individual filter weights $\omega^j_{k,l} \geq 0$ and their associated offsets $\beta^j_0 \leq 0$. Additionally, the weights for a given filter at a given position must be valid information gain measures. To achieve this we restrict the column-wise sum to be less than or equal to 2 under uniform background and rescale the weights column-wise to maintain a valid information-theoretic relationship during training. This latter step is accomplished by rescaling the weights from information gain to probabilities by dividing each weight by its column-wise sum and subsequently converting back to information gain by multiplying each weight in the PPM by the column-wise sum of the expected self-information gain. A psuedo-count of 0.05 is added to entries whose column sum is less than 0.1 when converting to the PPM to control for cases in which a single, small weight in the column is non-zero and thus occupies a position-probability of 1. We perform this rescaling at the end of each training epoch. The constraint on the offset weights β^j_0 for each filter to be strictly non-positive is incorporated to improve the interpretation of the filter activations: consider that the minimum activation value, ζ^j_n for observation n and filter j, could take without a bias offset β^j_0 and under the sigmoidal activation would be $1/2$. Quite simply the addition of a negative offset helps decrease the value of ζ^j_n to 0 for certain n.

Under such a scheme we find that the learned filters may be interpreted as information gain and are directly comparable to previously-annotated database motifs in the form of IGMs. For this reason we consider the sum over the filter as a measure of information gain of the motif, which may equivalently be interpreted as the KL divergence. In addition, the associated β^j_1 model coefficients are interpreted as the estimated effect size of the motif. Under the model described above, and within the context of the applications considered, the β^j_1 estimates translate to log-odds ratios. In experiments in which the negatively-labelled cases represent purely random genome background, we also constrain these β_1^j to be strictly non-negative. While the inclusion of such a constraint is surely debatable per the application at hand, when the task is to discover enriched motifs in the positively-labelled class, such as in the MYC-CTCF-IRF simulation, the constraint is justifiable as one does not expect to discover depleted motifs. Surely any motifs with negative effect sizes would, by construction of the simulation, be due to over-fitting or spurious learned features. For this reason we include the $\beta_1^j \geq 0$ constraint in the data application, but do not in the first simulation presented with the *C*-motifs and *G*-motifs. In the latter case no such constraint is warranted.

PWMs as filters

When representing the motifs as position wieght matrices an altered weight constraint scheme is used. We no longer require the non-negativity constraint on the $\omega_{k,l}^j$ nor the β_0^j negativity constraint. We limit the upper bound of the individual weights to be less than or equal to 2 (again for the case of uniform background) and rescale the weights column-wise to maintain a valid distributional relationship during training. This latter step is accomplished by rescaling the weights from log-odds to probabilities by adding $log_2(b_n)$ to each weight and raising two to this power. We subsequently convert the calculated probability back to a position weight by computing the log-odds. Again a psuedo-count of 0.05 is added to the zero entries to avoid values of negative infinity. We perform this rescaling at the end of each training epoch as in the IGM case. Under such a scheme the regularization is again interpretable, this time encouraging small and non-zero log-odds to shrink to zero. The filter-level regularization again discourages redundancy but this time does so on the log-odds scale instead of the information gain scale. Lastly, it is worth noting that backing the probabilities out during the weight rescaling performed for the PWM requires the addition operation and the power operation, while in the case of the IGM, the calculation simply requires the division by the column sum.

Practical considerations

All sequence motifs have been represented as information gain matrices (IGMs) in this manuscript however this need not be the case and our software implementation provides support for initializing and/or learning motifs as PWM representations. As previously described, there is a one-to-one correspondence between the two measures some notes should be made on the implications between choosing one representation over the other. Notably the use of a PWM tends to spread out the activation values due to the negatively-unboundedness of the log-odds computed on multinomial probabilities. This leads to an asymmetry in how the sparsity in interpreted since extremely small position probabilities will never attain zero probability ($-\infty$ position weight). The same is not true for the IGM representation as low position probabilities indeed attain zero. In other words, in the case of the IGM the sparsity encourages position probabilities to not only attain exactly 0.25 but to also attain smaller values. In the case of the PWM the sparsity exclusively encourages position probabilities of 0.25. Another consideration is that interpretation and weight visualization is a bit more nuanced when utilizing PWMs because the weights may be both positive and negative. Whereas added weight constraints were used in the case of the IGM to encourage interpretability, the same is not justifiable for the PWM. Additionally, in light of the large influence of increasingly-negative position weights, one may opt to abandon the latent variable interpretation presented herein and instead use a ReLU activation function to alleviate this. Undoubtedly it is a decision best determined by the problem at hand and the goals of the analysis.

A uniform background assumption simplifies the training procedure because the maximum value in the weight constraints is simply two across the board. Under a different background probabilities assumption, we do not include a maximum constraint while performing gradient descent, and instead recommend rescaling the weights after each batch (as opposed to epoch). This procedure is computationally more expensive and one may instead train all weights under the uniform background assumption and rescale them after the fact.

Regularizing redundancy

Despite encouraging interpretable learning through weight constraints, the filters tend to learn correlated and often redundant features, with this being related to the hyper-parameter J determining the number of such motifs to learn *de novo*. We remedy this issue through regularization, specifically by incorporating a sparse group lasso penalty into the formulation to encourage learning highly predictive and distinct motifs[127]. Our regularization scheme may be expressed as:

$$\min_{\Omega,\beta\in\mathbb{R}}\left(R_{\mathcal{G}} + \lambda_1 R_{\mathcal{F}} + \lambda_2 R + \lambda_3 R + \lambda_4 \sum_{j=1}^{J} |\beta_1^j|\right)$$

$$R_{\mathcal{G}} = \sum_{n=1}^{N} -Y_n \left[\log\left(\frac{1}{1+e^{-\mathcal{G}(X_n)}}\right)\right] - (1-Y_n)\left[\log\left(\frac{e^{-\mathcal{G}(X_n)}}{1+e^{-\mathcal{G}(X_n)}}\right)\right]$$

$$R_{\mathcal{F}} = \sum_{j=1}^{J} \sqrt{4 \times L_j}||\omega^j_{\cdot\cdot}||_2 = \sum_{j=1}^{J} \sqrt{4 \times L_j}\sqrt{\sum_{l=1}^{L_j} \sum_{k\in\{A,C,G,T\}} (\omega^j_{k,l})^2}$$

$$R = \sum_{j=1}^{J} \sum_{l=1}^{L_j} \sum_{k\in\{A,C,G,T\}} |\omega^j_{k,l}|$$

$$R = \sum_{j=1}^{J} \sum_{l=1}^{L_j} \sum_{k\in\{A,C,G,T\}} \frac{|\omega^j_{k,l}|}{\rho_l}$$

Model training proceeds by trying to find the weights which minimize this sum. In other words, gradient descent is performed to minimize this sum. $R_{\mathcal{G}}$ simply denotes the standard logistic loss function and the λ parameters dictate the trade-off between minimizing this loss at the cost of each regularization penalty, respectively. $R_{\mathcal{F}}$ encourages filter-level group sparsity with the L2/L1 norm[149], R encourages nucleotide-level sparsity with the L1 norm, and R encourages motifs to form near the center of the filter with a location-specific penalty ρ_l. Recall in the genomic data applications we consider each filter is a $4 \times L_j$ matrix, with L_j assuming values around 8-16 depending on the specific problem. We wish to discourage learning shifted versions of the same motif and so we set the vector ρ to be a concatenation of a sequence of decreasing values beginning at λ_3 and ending at 0 of length of equal to $L_j/2$, concatted with a reversed version of the same sequence (i.e. increasing values from 0 to λ_3). Thus sparsity is more strongly encouraged at the outer positions of the filter than towards the middle, in turn discouraging redundant and shifted versions of the same motif. Finally, we penalize the L1 norm of the β_1^j and include this regularization penalty in many of the models considered throughout the text. This penalty simply pushes effect size estimates to 0 and is often employed in CNNs. While group sparsity has surely been implemented in the context of image analysis (for example, [144,81,147], this is the first instance the authors are aware of using this regularization framework on genomic data. The regularization scheme in total pushes many individual weights to zero and discourages spurious motifs by pushing entire filters to zero except for those attaining suitably large KL divergence (or log-odds in the case of the PWM).

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv*, 1603.04467.

[2] Adebayo, J., Gilmer, J., Goodfellow, I., & Kim, B. (2018a). Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307*.

[3] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018b). Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems* (pp. 9505–9515).

[4] Alexandari, A. M., Shrikumar, A., & Kundaje, A. (2017). Separable fully connected layers improve deep learning models for genomics. *BioRxiv*, (pp. 146431).

[5] Ali, M. & Aittokallio, T. (2019). Machine learning and feature selection for drug response prediction in precision oncology applications. *Biophysical reviews*, 11(1), 31–39.

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

[7] Alvarez-Melis, D. & Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv*, 1806.08049.

[8] Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology*, 12(7), 878.

[9] Avsec, Z., Weilert, M., Shrikumar, A., Alexandari, A., Krueger, S., Dalal, K., , Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., & Zeitlinger, J. (2019). Deep learning at base-resolution reveals motif syntax of the cis-regulatory code. *bioRxiv*, 737981.

[10] Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., Wilson, C. J., Lehár, J., Kryukov, G. V., Sonkin, D., et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391), 603–607.

[11] Bartoszewicz, J. M., Seidel, A., Rentzsch, R., & Renard, B. Y. (2020). Deepac: predicting pathogenic potential of novel dna with reverse-complement neural networks. *Bioinformatics*, 36(1), 81–89.

[12] Basu, A., Bodycombe, N. E., Cheah, J. H., Price, E. V., Liu, K., Schaefer, G. I., Ebright, R. Y., Stewart, M. L., Ito, D., Wang, S., et al. (2013). An interactive resource to identify cancer genetic and lineage dependencies targeted by small molecules. *Cell*, 154(5), 1151–1161.

[13] Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1), 289–300.

[14] Biankin, A. V., Waddell, N., Kassahn, K. S., Gingras, M.-C., Muthuswamy, L. B., Johns, A. L., Miller, D. K., Wilson, P. J., Patch, A.-M., Wu, J., et al. (2012). Pancreatic cancer genomes reveal aberrations in axon guidance pathway genes. *Nature*, 491(7424), 399–405.

[15] Blum, C. & Kollmann, M. (2019). Neural networks with circular filters enable data efficient inference of sequence motifs. *Bioinformatics*, 35(20), 3937–3943.

[16] Bogard, N., Linder, J., Rosenberg, A. B., & Seelig, G. (2019). A deep neural network for predicting and engineering alternative polyadenylation. *Cell*, 178(1), 91–106.

[17] Brown, R. C. & Lunter, G. (2019). An equivariant bayesian convolutional network predicts recombination hotspots and accurately resolves binding motifs. *Bioinformatics*, 35(13), 2177–2184.

[18] Casella, G. & Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.

[19] Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2019). *shiny: Web Application Framework for R*. R package version 1.3.2.

[20] Chen, Chen, J. H. X. S. H. Y. J. A. B. & Cheng, J. (2019). Interpretable attention model in transcription factor binding site prediction with deep neural networks. *bioRxiv*, 648691.

[21] Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras.

[22] Consortium, C. C. L. E., of Drug Sensitivity in Cancer Consortium, G., et al. (2015). Pharmacogenomic agreement between two cancer cell line data sets. *Nature*, 528(7580), 84.

[23] Consortium, E. P. et al. (2012). An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414), 57.

[24] Corsello, S. M., Nagari, R. T., Spangler, R. D., Rossen, J., Kocak, M., Bryan, J. G., Humeidi, R., Peck, D., Wu, X., Tang, A. A., et al. (2020). Discovering the anticancer potential of non-oncology drugs by systematic viability profiling. *Nature Cancer*, 1(2), 235–248.

[25] Cramir, H. (1946). Mathematical methods of statistics. *Princeton U. Press, Princeton*, (pp. 500).

[26] Cuperus, J., Groves, B., Kuchina, A., Rosenberg, A., Jojic, N., Fields, S., & Seelig, G. (2017). Deep learning of the regulatory grammar of yeast 5′ untranslated regions from 500,000 random sequences. *Genome research*, 27(12), 2015–2024.

[27] Dančík, V., Carrel, H., Bodycombe, N. E., Seiler, K. P., Fomina-Yadlin, D., Kubicek, S. T., Hartwell, K., Shamji, A. F., Wagner, B. K., & Clemons, P. A. (2014). Connecting small molecules with similar assay performance profiles leads to new biological hypotheses. *Journal of biomolecular screening*, 19(5), 771–781.

[28] Dauparas, J., Wang, H., Swartz, A., Koo, P., Nitzan, M., & Ovchinnikov, S. (2019). Unified framework for modeling multivariate distributions in biological sequences. *arXiv:1906.02598*.

[29] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[30] Ding, M. Q., Chen, L., Cooper, G. F., Young, J. D., & Lu, X. (2018). Precision oncology beyond targeted therapy: Combining omics data with machine learning matches the majority of cancer cells to effective therapeutics. *Molecular Cancer Research*, 16(2), 269–278.

[31] Duren, R., Neeley, S., Welsh, J. W., Tackes, N., Li, X. S., & Davis, C. F. (2017). Molecularmatch lab integrates knowledgebases for collaborative clinical interpretation of variation in cancer. *Cancer Genetics*, 214, 45.

[32] Eraslan, G., Avsec, Z., Gagneur, J., & Theis, F. (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, (pp.1~).

[33] Finnegan, A. & Song, J. (2017). Maximum entropy methods for extracting the learned features of deep neural networks. *PLoS Computational Biology*, 13(10), e1005836.

[34] Fornes, O., Castro-Mondragon, J. A., Khan, A., van der Lee, R., Zhang, X., Richmond, P. A., Modi, B. P., Correard, S., Gheorghe, M., Baranašić, D., et al. (2020). Jaspar 2020: update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 48(D1), D87–D92.

[35] Fujimoto, J., Shiota, M., Iwahara, T., Seki, N., Satoh, H., Mori, S., & Yamamoto, T. (1996). Characterization of the transforming activity of p80, a hyperphosphorylated protein in a ki-1 lymphoma cell line with chromosomal translocation t (2; 5). *Proceedings of the National Academy of Sciences*, 93(9), 4181–4186.

[36] Garnett, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., Greninger, P., Thompson, I. R., Luo, X., Soares, J., et al. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391), 570–575.

[37] Geeleher, P., Gamazon, E. R., Seoighe, C., Cox, N. J., & Huang, R. S. (2016). Consistency in large pharmacogenomic studies. *Nature*, 540(7631), E1–E2.

[38] Ghanbari, M. & Ohler, U. (2019). Deep neural networks for interpreting rna binding protein target preferences. *bioRxiv*.

[39] Grassi, L., Alfonsi, R., Francescangeli, F., Signore, M., De Angelis, M. L., Addario, A., Costantini, M., Flex, E., Ciolfi, A., Pizzi, S., et al. (2019). Organoids as a new model for improving regenerative medicine and cancer personalized therapy in renal diseases. *Cell death & disease*, 10(3), 1–15.

[40] Greenside, P., S. T. F. P. & Kundaje, A. (2018). Discovering epistatic feature interactions from neural network models of regulatory dna sequences. *Bioinformatics*, 34(117), i629–i637.

[41] Griffith, M., Spies, N. C., Krysiak, K., McMichael, J. F., Coffman, A. C., Danos, A. M., Ainscough, B. J., Ramirez, C. A., Rieke, D. T., Kujan, L., et al. (2017). Civic is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer. *Nature genetics*, 49(2), 170.

[42] Gupta, A., Gautam, P., Wennerberg, K., & Aittokallio, T. (2020). A normalized drug response metric improves accuracy and consistency of anticancer drug sensitivity quantification in cell-based screening. *Communications biology*, 3(1), 1–12.

[43] Gupta, S., Stamatoyannopoulos, J. A., Bailey, T. L., & Noble, W. S. (2007). Quantifying similarity between motifs. *Genome Biology*, 8(2).

[44] Hafner, M., Niepel, M., Chung, M., & Sorger, P. K. (2016). Growth rate inhibition metrics correct for confounders in measuring sensitivity to cancer drugs. *Nature methods*, 13(6), 521.

[45] Hafner, M., Niepel, M., & Sorger, P. K. (2017). Alternative drug sensitivity metrics improve preclinical cancer pharmacogenomics. *Nature biotechnology*, 35(6), 500–502.

[46] Haibe-Kains, B., El-Hachem, N., Birkbak, N. J., Jin, A. C., Beck, A. H., Aerts, H. J., & Quackenbush, J. (2013). Inconsistency in large pharmacogenomic studies. *Nature*, 504(7480), 389–393.

[47] Haverty, P. M., Lin, E., Tan, J., Yu, Y., Lam, B., Lianoglou, S., Neve, R. M., Martin, S., Settleman, J., Yauch, R. L., et al. (2016). Reproducible pharmacogenomic profiling of cancer cell line panels. *Nature*, 533(7603), 333–337.

[48] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1026–1034).

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

[50] Hoffman, G. E., Bendl, J., Girdhar, K., Schadt, E. E., & Roussos, P. (2019). Functional interpretation of genetic variants using deep learning predicts impact on chromatin accessibility and histone modification. *Nucleic acids research*, 47(20), 10597–10611.

[51] Holla, V. R., Elamin, Y. Y., Bailey, A. M., Johnson, A. M., Litzenburger, B. C., Khotskaya, Y. B., Sanchez, N. S., Zeng, J., Shufean, M. A., Shaw, K. R., et al. (2017). Alk: a tyrosine kinase target for cancer therapy. *Molecular Case Studies*, 3(1), a001115.

[52] Horn, L. & Pao, W. (2009). Eml4-alk: honing in on a new target in non–small-cell lung cancer. *Journal of clinical oncology: official journal of the American Society of Clinical Oncology*, 27(26), 4232.

[53] Hu, Z. T., Ye, Y., Newbury, P. A., Huang, H., & Chen, B. (2019). Aicm: A genuine framework for correcting inconsistency between large pharmacogenomics datasets. In *PSB* (pp. 248–259).: World Scientific.

[54] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. (2017). Densely connected convolutional networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4700–4708).

[55] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems* (pp. 125–136).

[56] Inukai, S., Kock, K. H., & Bulyk, M. L. (2017). Transcription factor–dna binding: beyond binding site motifs. *Current opinion in genetics & development*, 43, 110–119.

[57] Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv*, 1502.03167.

[58] Jaganathan, K., Panagiotopoulou, S. K., McRae, J. F., Darbandi, S. F., Knowles, D., Li, Y. I., Kosmicki, J. A., Arbelaez, J., Cui, W., Schwartz, G. B., et al. (2019). Predicting splicing from primary sequence with deep learning. *Cell*, 176(3), 535–548.

[59] Jain, S. & Wallace, B. (2019). Attention is not explanation. *arXiv*, (1902.10186).

[60] Jang, I. S., Neto, E. C., Guinney, J., Friend, S. H., & Margolin, A. A. (2014). Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data. In *Biocomputing 2014* (pp. 63–74). World Scientific.

[61] Janizek, J.D., S. P. & Lee, S. (2020). Explaining explanations: Axiomatic feature interactions for deep networks. *arXiv*, (2002.04138).

[62] Kalamara, A., Tobalina, L., & Saez-Rodriguez, J. (2018). How to find the right drug for each patient? advances and challenges in pharmacogenomics. *Current opinion in systems biology*, 10, 53–62.

[63] Kanehisa, M. & Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1), 27–30.

[64] Kazandjian, D., Blumenthal, G. M., Chen, H.-Y., He, K., Patel, M., Justice, R., Keegan, P., & Pazdur, R. (2014). Fda approval summary: crizotinib for the treatment of metastatic non-small cell lung cancer with anaplastic lymphoma kinase rearrangements. *The oncologist*, 19(10), e5.

[65] Kelley, D. R., Reshef, Y. A., Bileschi, M., Belanger, D., McLean, C. Y., & & Snoek, J. (2018). Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5), 739–750.

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

[67] Khan, A., Fornes, O., Stigliani, A., Gheorghe, M., Castro-Mondragon, J. A., van der Lee, R., Bessy, A., Cheneby, J., Kulkarni, S. R., Tan, G., et al. (2017). Jaspar 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic acids research*, 46(D1), D260–D266.

[68] Kingma, D. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*, 1412.6980.

[69] Kinney, J. B. & McCandlish, D. M. (2019). Massively parallel assays and quantitative sequence–function relationships. *Annual Review of Genomics and Human Genetics*.

[70] Klijn, C., Durinck, S., Stawiski, E. W., Haverty, P. M., Jiang, Z., Liu, H., Degenhardt, J., Mayba, O., Gnad, F., Liu, J., et al. (2015). A comprehensive transcriptional portrait of human cancer cell lines. *Nature biotechnology*, 33(3), 306.

[71] Koo, P., Anand, P., Paul, S., & Eddy, S. (2018). Inferring sequence-structure preferences of rna-binding proteins with convolutional residual networks. *bioRxiv*, (418459).

[72] Koo, P. & Ploenzke, M. (2019). Improving convolutional network interpretability with exponential activations. *bioRxiv*, (650804).

[73] Koo, P. K. & Eddy, S. R. (2018). Representation learning of genomic sequence motifs with convolutional neural networks. *BioRxiv*, (362756).

[74] Krebs, K. & Milani, L. (2019). Translating pharmacogenomics into clinical decisions: do not let the perfect be the enemy of the good. *Human genomics*, 13(1), 39.

[75] Kurilov, R., Haibe-Kains, B., & Brors, B. (2020). Assessment of modelling strategies for drug response prediction in cell lines and xenografts. *Scientific reports*, 10(1), 1–11.

[76] Lab, K. (2016–). simdna: simulated datasets of dna. [Online; accessed 2018-04-08].

[77] Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., et al. (2006). The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *science*, 313(5795), 1929–1935.

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

[79] Landt, S. G., Marinov, G. K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., Bernstein, B. E., Bickel, P., Brown, J. B., Cayting, P., et al. (2012). Chip-seq guidelines and practices of the encode and modencode consortia. *Genome research*, 22(9), 1813–1831.

[80] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[81] Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016). Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

[82] Liu, B., Hussami, N., Shrikumar, A., Shimko, T., Bhate, S., Longwell, S., Montgomery, S., & Kundaje, A. (2019). A multi-modal neural network for learning cis and trans regulation of stress response in yeast. *arXiv preprint arXiv:1908.09426*.

[83] Liu, Yi, K. B. & Reinitz, J. (2019). Fully interpretable deep learning model of transcriptional control. *BioRxiv*, (655639).

[84] Liu, G., Z. H. & Gifford, D. (2019). Visualizing complex feature interactions and feature sharing in genomic deep neural networks. *BMC Bioinformatics*, 20(1), 1–14.

[85] Luna, A., Rajapakse, V. N., Sousa, F. G., Gao, J., Schultz, N., Varma, S., Reinhold, W., Sander, C., & Pommier, Y. (2016). rcellminer: exploring molecular profiles and drug response of the nci-60 cell lines in r. *Bioinformatics*, 32(8), 1272–1274.

[86] Lundberg, S. & Lee, S. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 4765-4774.

[87] Mahajan, P. B. (2016). Recent advances in application of pharmacogenomics for biotherapeutics. *The AAPS journal*, 18(3), 605–611.

[88] Maslova, A., Ramirez, R., Ma, K., Schmutz, H., Wang, C., Fox, C., Ng, B., Benoist, C., & Mostafavi, S. (2019). Learning immune cell differentiation. *bioRxiv*.

[89] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2), 442–451.

[90] Misra, P. & Singh, S. (2019). Role of cytokines in combinatorial immunotherapeutics of non-small cell lung cancer through systems perspective. *Cancer medicine*, 8(5), 1976–1995.

[91] Nagle, P. W., Plukker, J. T. M., Muijs, C. T., van Luijk, P., & Coppes, R. P. (2018). Patient-derived tumor organoids for prediction of cancer treatment response. In *Seminars in cancer biology*, volume 53 (pp. 258–264).: Elsevier.

[92] Ng, S.-K. & McLachlan, G. J. (2004). Using the em algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification. *IEEE transactions on neural networks*, 15(3), 738–749.

[93] Niepel, M., Hafner, M., Mills, C. E., Subramanian, K., Williams, E. H., Chung, M., Gaudio, B., Barrette, A. M., Stern, A. D., Hu, B., et al. (2019). A multi-center study on the reproducibility of drug-response assays in mammalian cell lines. *Cell systems*, 9(1), 35–48.

[94] Onimaru, K., N. O. & Kuraku, S. (2018). A regulatory-sequence classifier with a neural network for genomic information processing. *bioRxiv*, (355974).

[95] Ou, S.-H. I. (2011). Crizotinib: a novel and first-in-class multitargeted tyrosine kinase inhibitor for the treatment of anaplastic lymphoma kinase rearranged non-small cell lung cancer and beyond. *Drug design, development and therapy*, 5, 471.

[96] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc.

[97] Picco, G., Chen, E. D., Alonso, L. G., Behan, F. M., Gonçalves, E., Bignell, G., Matchan, A., Fu, B., Banerjee, R., Anderson, E., et al. (2019). Functional linkage of gene fusions to cancer cell fitness assessed by pharmacological and crispr-cas9 screening. *Nature communications*, 10(1), 1–12.

[98] Ploenzke, M. & Irizarry, R. (2018). Interpretable convolution methods for learning genomic sequence motifs. *bioRxiv*, 411934.

[99] Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

[100] Pozdeyev, N., Yoo, M., Mackie, R., Schweppe, R. E., Tan, A. C., & Haugen, B. R. (2016). Integrating heterogeneous drug sensitivity data from cancer pharmacogenomic studies. *Oncotarget*, 7(32), 51619.

[101] Quang, D., Guan, Y., & Parker, S. C. (2018). Yamda: thousandfold speedup of em-based motif discovery using deep learning libraries and gpu. *Bioinformatics*, 1, 3.

[102] Quang, D. & Xie, X. (2016). Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic Acids Research*, 44(11), 107.

[103] Quang, D. & Xie, X. (2019). Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, 166, 40–47.

[104] R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[105] Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., & Sohl-Dickstein, J. (2016). On the expressive power of deep neural networks. *arXiv preprint arXiv*, 1606.05336.

[106] Rahman, R., Dhruba, S. R., Matlock, K., De-Niz, C., Ghosh, S., & Pal, R. (2019). Evaluating the consistency of large-scale pharmacogenomic studies. *Briefings in Bioinformatics*, 20(5), 1734–1753.

[107] Rajapakse, V. N., Luna, A., Yamade, M., Loman, L., Varma, S., Sunshine, M., Iorio, F., Sousa, F. G., Elloumi, F., Aladjem, M. I., et al. (2018). Cellminercdb for integrative cross-database genomics and pharmacogenomics analyses of cancer cell lines. *iScience*, 10, 247–264.

[108] Rees, M. G., Seashore-Ludlow, B., Cheah, J. H., Adams, D. J., Price, E. V., Gill, S., Javaid, S., Coletti, M. E., Jones, V. L., Bodycombe, N. E., et al. (2016). Correlating chemical sensitivity and basal gene expression reveals mechanism of action. *Nature chemical biology*, 12(2), 109.

[109] Ribeiro, M. T., Singh, S., & Guestrin., C. (2016). Why should i trust you?: Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*.

[110] Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, 43(7), e47–e47.

[111] Safikhani, Z., El-Hachem, N., Quevedo, R., Smirnov, P., Goldenberg, A., Birkbak, N. J., Mason, C., Hatzis, C., Shi, L., Aerts, H. J., et al. (2016a). Assessment of pharmacogenomic agreement. *F1000Research*, 5.

[112] Safikhani, Z., El-Hachem, N., Smirnov, P., Freeman, M., Goldenberg, A., Birkbak, N. J., Beck, A. H., Aerts, H. J., Quackenbush, J., & Haibe-Kains, B. (2016b). Safikhani et al. reply. *Nature*, 540(7631), E2–E4.

[113] Safikhani, Z., Smirnov, P., Freeman, M., El-Hachem, N., She, A., Rene, Q., Goldenberg, A., Birkbak, N. J., Hatzis, C., Shi, L., et al. (2016c). Revisiting inconsistency in large pharmacogenomic studies. *F1000Research*, 5.

[114] Sahu, A., Prabhash, K., Noronha, V., Joshi, A., & Desai, S. (2013). Crizotinib: A comprehensive review. *South Asian journal of cancer*, 2(2), 91.

[115] Sasaki, T., Rodig, S. J., Chirieac, L. R., & Jänne, P. A. (2010). The biology and treatment of eml4-alk non-small cell lung cancer. *European journal of cancer*, 46(10), 1773–1780.

[116] Schneider, T. D. & Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, 18(20), 6097–6100.

[117] Seashore-Ludlow, B., Rees, M. G., Cheah, J. H., Cokol, M., Price, E. V., Coletti, M. E., Jones, V., Bodycombe, N. E., Soule, C. K., Gould, J., et al. (2015). Harnessing connectivity in a large-scale small-molecule sensitivity dataset. *Cancer discovery*, 5(11), 1210–1223.

[118] Selvaraju, R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *In Proceedings of the IEEE International Conference on Computer Vision*, (pp. 618–626).

[119] Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A., et al. (2019). Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13). *Proteins: Structure, Function, and Bioinformatics*, 87(12), 1141–1148.

[120] Shen, Zhen, W. B. & Huang, D.-S. (2018). Recurrent neural network for predicting transcription factor binding sites. *Scientific Reports*, 8(1).

[121] Shimada, Y., Kohno, T., Ueno, H., Ino, Y., Hayashi, H., Nakaoku, T., Sakamoto, Y., Kondo, S., Morizane, C., Shimada, K., et al. (2017). An oncogenic alk fusion and an rras mutation in kras mutation-negative pancreatic ductal adenocarcinoma. *The oncologist*, 22(2), 158.

[122] Shlens, J. (2014). Notes on kullback-leibler divergence and likelihood. *arXiv preprint arXiv:1404.2000*.

[123] Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. *In Proceedings of the 34th International Conference on Machine Learning*, 70, 3145–3153.

[124] Shrikumar, A., Tian, K., Shcherbina, A., Avsec, Z., Banerjee, A., Sharmin, M., Nair, S., & Kundaje, A. (2018). Tf-modisco v0. 4.4. 2-alpha. *arXiv*, (pp. 1811.00416).

[125] Shrikumar, A., G. P. & Kundaje, A. (2017). Reverse-complement parameter sharing improves deep learning models for genomics. *bioRxiv*, (103663).

[126] Sim, J. & Wright, C. C. (2005). The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3), 257–268.

[127] Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2), 231–245.

[128] Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv*, 1312.6034.

[129] Singh, Ritambhara, J. L. A. S. & Qi, Y. (2017). Attend and predict: Understanding gene regulation by selective attention on chromatin. *Advances in neural information processing systems*, (pp. 6785–6795).

[130] Sixt, L., G. M. & Landgraf, T. (2019). When explanations lie: Why modified bp attribution fails. *arXiv*, (1912.09818).

[131] Smirnov, P., Safikhani, Z., El-Hachem, N., Wang, D., She, A., Olsen, C., Freeman, M., Selby, H., Gendoo, D. M., Grossmann, P., et al. (2016). Pharmacogx: an r package for analysis of large pharmacogenomic datasets. *Bioinformatics*, 32(8), 1244–1246.

[132] Soda, M., Takada, S., Takeuchi, K., Choi, Y. L., Enomoto, M., Ueno, T., Haruta, H., Hamada, T., Yamashita, Y., Ishikawa, Y., et al. (2008). A mouse model for eml4-alk-positive lung cancer. *Proceedings of the National Academy of Sciences*, 105(50), 19893–19897.

[133] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.

[134] Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *In Proceedings of the 34th International Conference on Machine Learning*, 70, 3319–3328.

[135] Tamborero, D., Rubio-Perez, C., Deu-Pons, J., Schroeder, M. P., Vivancos, A., Rovira, A., Tusquets, I., Albanell, J., Rodon, J., Tabernero, J., et al. (2018). Cancer genome interpreter annotates the biological and clinical relevance of tumor alterations. *Genome medicine*, 10(1), 25.

[136] Tareen, A., . K. J. B. (2019). Biophysical models of cis-regulation as interpretable neural networks. *arXiv*, :2001.03560.

[137] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, (pp. 267–288).

[138] Troutman, S., Moleirinho, S., Kota, S., Nettles, K., Fallahi, M., Johnson, G. L., & Kissil, J. L. (2016). Crizotinib inhibits nf2-associated schwannoma through inhibition of focal adhesion kinase 1. *Oncotarget*, 7(34), 54515.

[139] Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv*, 1805.12152.

[140] Tunney, R., McGlincy, N. J., Graham, M. E., Naddaf, N., Pachter, L., & Lareau, L. F. (2018). Accurate design of translational output by a neural network model of ribosome distribution. *Nature structural & molecular biology*, 25(7), 577–582.

[141] Ullah, F. & Ben-Hur, A. (2020). A self-attention model for inferring cooperativity between regulatory features. *BioRxiv*.

[142] Wang, D., Li, D., Qin, G., Zhang, W., Ouyang, J., Zhang, M., & Xie, L. (2015). The structural characterization of tumor fusion genes and proteins. *Computational and mathematical methods in medicine*, 2015.

[143] Wei, J., Van der Wekken, A. J., Saber, A., Terpstra, M. M., Schuuring, E., Timens, W., Hiltermann, T. J. N., Groen, H. J., Van den Berg, A., & Kok, K. (2018). Mutations in emt-related genes in alk positive crizotinib resistant non-small cell lung cancers. *Cancers*, 10(1), 10.

[144] Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems* (pp. 2074–2082).

[145] Yadav, B., Pemovska, T., Szwajda, A., Kulesskiy, E., Kontro, M., Karjalainen, R., Majumder, M. M., Malani, D., Murumägi, A., Knowles, J., et al. (2014). Quantitative scoring of differential drug sensitivity for individually optimized anticancer therapies. *Scientific reports*, 4, 5193.

[146] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5754–5764).

[147] Yoon, J. & Hwang, S. J. (2017). Combined group and exclusive sparsity for deep neural networks. In *International Conference on Machine Learning* (pp. 3958–3966).

[148] Yu, F. & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

[149] Yuan, M. & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49–67.

[150] Zeiler, M. D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision* (pp. 818–833).: Springer.

[151] Zhao, Z., Li, K., Toumazou, C., & Kalofonou, M. (2019). A computational model for anticancer drug sensitivity prediction. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (pp. 1–4).: IEEE.

[152] Zhou, J., Park, C., Theesfeld, C., Wong, A., Yuan, Y., Scheckel, C., Fak, J., Funk, J., Yao, K., Tajima, Y., & Packer, A. (2019). Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk. *Nature Genetics*, 51(6), 973.

[153] Zhou, J., Theesfeld, C. L., Yao, K., Chen, K. M., Wong, A. K., & Troyanskaya, O. G. (2018). Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8), 1171–1179.

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

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The template that was used to format this thesis has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu.