

UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

Leveraging Chemical Structures and Molecular Information with Interpretable Deep Learning

Permalink

<https://escholarship.org/uc/item/38d5h7tt>

Author

Anastopoulos, Ioannis

Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

**LEVERAGING CHEMICAL STRUCTURES AND MOLECULAR
INFORMATION WITH INTERPRETABLE DEEP LEARNING**

A thesis submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY
in
BIOINFORMATICS AND BIOENGINEERING

by

Ioannis Nikolaos Anastopoulos

March 2022

The Dissertation of I. Anastopoulos
is approved:

Professor David Haussler, Chair

Professor Joshua Stuart

Professor David Draper

Professor Olena Vaske

Professor Angela Brooks

Peter Biehl
Vice Provost and Dean of Graduate Studies

Copyright © by
Ioannis N. Anastopoulos
2021

Contents

Abstract	iv
Acknowledgements	vi
Chapter 1 : Introduction	1
1.1 Drug Response Prediction In the Era of Deep Learning	1
1.2 Embedding Molecular Structure in a Data-Driven Manner	3
Chapter 2 : Reconciling the Differences Between Cell Lines and Primary Tumors Coupled with Drug Chemical Information to Improve Drug Response Prediction	6
2.1 Chapter Introduction	6
2.2 Generalizing Drug Response Prediction to Patients	7
2.3 Regularized Drug Response Prediction in Cancer Patients using Deep Learning	9
Chapter 3 : Biological Pathway Informed Deep Learning Recapitulates Known Biology	34
3.1 Chapter Introduction	34
3.2 Biological Network-Inspired Interpretable Variational Autoencoder	35
Chapter 4 : Leveraging Chemical Structure Information to Identify Methylation Status of Nanopore Reads	45
4.1 Chapter Introduction	45
4.2 Towards Inferring Nanopore Sequencing Ionic Currents from Nucleotide Chemical Structures	46
Chapter 5 : Future Directions and Other Works	56
5.1 Future Direction: Drug Response in Tumor Patients	56
5.2 Future Direction: Molecular Structure Embedding in Nanopore Sequencing	56
5.3 Other work: Internship at Coral Genomics Inc.	57

Abstract

Ioannis Anastopoulos

Leveraging Chemical Structures And Molecular Information With Interpretable Deep Learning

As biological data become more readily available and convoluted, equally involved methods are needed to predict and understand outcomes in biological systems. Classical machine learning methods are not well suited for prediction tasks that need to integrate heterogeneous sources of information to predict the target variable. Deep learning is capable of integrating such disparate inputs with impressive results.

Here, I present my work in integrating cancer cell line transcriptomic information with the chemical structure information of the perturbation they were treated with (Chapter 2). This work leverages recent developments in deep learning for aligning domains (here cell lines and patients) in a data-driven way and advanced featurization of molecules. I show that by integration of these methods predicting drug response in patients is improved compared to more conventional methods. This model can be used to identify therapies for a patient by using only transcriptomic and chemical information. I further present my work on applying chemical featurization on nanopore sequences to *de novo* model nucleotide modifications (Chapter 3). Given the polynomial nature of possible modifications, producing gold standard data to identify such events is a daunting task. I show that knowledge learned on chemical features in the canonical (un-modified) context can be transferred to identify nucleotide modifications with a high degree of accuracy.

Finally, in Chapter 4 I present a collaborative work on developing an interpretable deep learning model for identifying the activation of biological pathways following application of a

perturbagen. We show that by guiding the the flow of information through the neural network we can extract more biologically meaningful information following perturbation compared to more classical methods such as Gene Set Enrichment Analysis (GSEA). This model can be used to circumvent costly and time consuming experiments to inform of the pathways being altered during application of a pertubagent is any biological system.

Acknowledgements

I gratefully acknowledge the support of my advisors Dr. Joshua Stuart and Dr. Olena Vaske.

I thank the members of the Stuart Lab for being outstanding fellow researchers and friends.

I especially want to thank Dr. Hongxu Ding whose ambition inspired me to aim high.

I thank my parents and my grandparents, who are the real heroes here. I am grateful for their encouragement and support all these years.

Chapter 1 : Introduction

1.1 Drug Response Prediction In the Era of Deep Learning

Precision medicine aims to tailor treatment to each patient's genome by obtaining a clear understanding of the underlying cause of the disease. Once the cause is identified, and the basic biology contributing to the disease is well established, treatments targeting specific genes and pathways can be developed benefiting patients that display a similar phenotype [1].

Nevertheless, developing such treatments is a long and expensive process [2]. The stages of drug development include the following:

1. Drug discovery. This involved the identification of molecules that can bind to a the target in question
2. Preclinical research. Successful molecules from stage 1 are tested on disease models, such as tumor-derived cell lines or patient derived xenografts (PDX).
3. Clinical research. When a drug passes through stage 2 it moves on to clinical trials that assess efficacy and toxicity in patients.
4. Approval and marketing. The drug gets approved by the Food and Drug Administration (FDA) and is ready to be sold on the market.

It can take anywhere between 10 - 20 years, for a drug to pass through the above stages. The cost to develop a New Molecular Entity (NME; a small molecule compound) or New Biological Entity (NBE; an antibody, protein, gene therapy, or other biological medicine) can reach \$2.6 billion or more [3, 4, 5, 2]. The pharmaceutical industry's business model cannot sustain adequate innovation of new compounds without a significant increase in RD productivity. A big part of this issue are the diminishing numbers of novel drugs that are approved by the FDA. For example in 2007 only 17 NMEs were approved - the lowest number of approved novel drugs since 1983 [6].

Given the decline in RD productivity [7], and the diminishing market exclusivity for recently launched new medicines coupled with the huge loss of revenues owing to generic competition over the next decade [8], it is evident that the current state of drug development and marketing needs rethinking and new tools to expedite RD and reduce costs.

However, computational approaches and various prediction algorithms can help to reduce time, risks and save resources [9]. Techniques developed in deep learning have already shown promising results within the past 5 years. While many non-deep learning methods have been developed [10] none are yet optimal, given their inability to perform high-throughput screening, and to incorporate a variety of protein classes. On top of that, several studies now show that deep learning is an important approach to consider.

Drug–target interaction prediction is one of the most important tasks in developing a drug. Targets (proteins) often have one or more binding sites with substrates or regulatory molecules; these can be used for building prediction models. However, including other protein sites could bring bias into the analysis. The ability of pairwise input neural network (PINN) to accept two vectors with features obtained both from protein sequences and target profiles was used by Wang et al. to compute target–ligand interaction [11]. This advantage of NNs resulted in a better accuracy than other representative target–ligand interaction prediction methods. More recently Zhavoronkov et al. developed an deep learning approach for de novo drug design named GENTRL (developed generative tensorial reinforcement learning) using graph convolution networks and reinforcement learning. The algorithm allowed for designing, synthesizing, and experimentally testing potent six DDR1 kinase inhibitors in just 46 days, demonstrating the potential of this approach to provide rapid and effective molecular design [2].

A frequent cause for removal of a drug from the development or production pipeline is the risk of toxicity. Prediction of hepatotoxicity with computational approaches could help to avoid likely hepatotoxic drugs [9]. Xu et al. showed that it is possible using deep learning to predict

compound toxicity with raw chemical structure without requiring a complex encoding process [12]. Using CNNs, it is also possible to predict properties such as epoxidation, which means high reactivity and possible toxicity; this was first implemented by Hughes et al. by using simplified molecular input line entry specification (SMILES) format data of epoxidized molecules and hydroxides molecules, as a negative control [13]. Unintended side effects is another major cause for halting development of a new compound. This is especially true when patients with complex diseases or co-existing conditions are treated with a cocktail of medicines. While the use of drug combinations has been shown to be beneficial to such patients [14, 15] the risk for side effects induced by drug-drug interactions is quite high. Zitnik et al. showed that Decagon, a deep learning approach using graph convolutional networks to model drug-drug interactions, outperforms other approaches by a significant margin in predicting a large number of different side effects [16].

1.2 Embedding Molecular Structure in a Data-Driven Manner

Encoding the molecular structure and predicting molecular properties from it has been of significant interest in recent years [17]. Prediction of properties include modification status of molecules, solubility, membrane permeability, drug-target interaction, side effect prediction, transcriptome alteration prediction, etc. The value of creating a very accurate molecular embedding is that high throughput screening can become less laborious and time consuming. Models developed in predicting the aforementioned chemical properties can help in identifying lead molecules in terms of their predicted performance in membrane permeability, drug response and assist in whether molecules should proceed in the later stages of development [18].

The following have been common approaches in the field for extracting molecular features:

1. Sparse bit vectors with indices corresponding to the presence or absence of molecular substructures called fingerprints. Traditional fingerprints (which use a limited set of

predefined substructures) have developed into extended connectivity fingerprints (ECFPs), which are able to incorporate more predefined substructures [19]. ECFPs incorporate all substructures up to a user-selected radius size into the molecular representation, similar to Morgan fingerprints. Multiple substructures are represented by the same fingerprint index, however these representations are very static as the importance of any one substructure over another is not encoded in the vector.

2. Determination of empirical solute descriptors. Abraham relationships are a representative example of those. Solute and solvents are described by five empirical parameters measures chemical functionalities. The descriptor is thereof is then written as a linear combination of solute–solvent interactions.
3. Concatenation of many known molecular descriptors, which may include calculation of electronic descriptors or the use of existing property-estimation models. Software packages including Dragon[20] and CDK[21] among others can produce thousands od such descriptors.
4. SMILES strings or InChI strings, which are two different methods to encode molecules as character sequences. These can be used as as model inputs to character based models, or molecular structure models [22, 23].

Although classical machine learning models have been applied on commonly extracted molecular features, the high dimensional of these embeddings lead these models to downgrade features that have low effect on the final prediction. Furthermore, the chemical descriptors described above are very static in nature. Creating a more flexible molecular representation can lead to improvements in chemical property prediction, and in turn to the machine suggesting more and more viable lead molecules. ECFPs are still being extensively used in a variety of prediction tasks. However, it is possible that multiple important substructures are described by the same ECFP index, leading to collisions. In addition, it is also possible that many of the indices are not relevant to the prediction task. It is clear, therefore, that a more flexible molecular

representation is needed. Further, models that can learn this representation while learning the prediction target can lead to superior prediction performance, aiding in high throughput screening of molecules.

Techniques developed in deep learning allow models to learn general rules of chemical structure while learning the target variable. Specifically, recent studies have shown that graph convolutional networks (GCNs) perform better than the state of the art in node embedding, classification, and graph visualization. Graphs are essential tools to capture and model complicated relationships among data. In a variety of graph applications, including protein-protein interaction networks, social media, and citation networks, analyzing graph data plays an important role in various data mining tasks including node or graph classification. In this thesis I have applied graph embeddings learned from GCNs to learn general rules for a) drug response prediction in cancer, and b) for identifying the modification status of nucleotide bases from nanopore sequencing.

Kipf & Welling showed that a GCN autoencoder (GAE) performs better against the baseline graph embedding algorithms[24] in link prediction In the citation network datasets—Citeseer, Cora and Pubmed [25]. Further, the same group showed that a variational version of their graph encoder (VGAE) performed significantly better than their previously developed GAE on the same task [26]. Lastly, Pan et al. developed an adversarial version of graph autoencoders: an adversarially regularized graph autoencoder (ARGA) and an adversarially regularized variational graph autoencoder (ARVGA). Both of these models showed superior performance against GAE, and VGAE in link predictions on the same datasets. For the node clustering task, the authors showed that both ARGA, and ARVGA achieve dramatic improvement against not only GAE and VGAE, but also compared to other baselines across all metrics used: accuracy, non-mutual information, F1score, Precision, average rand index (ARI) [27].

Chapter 2 : Reconciling the Differences Between Cell Lines and Primary Tumors Coupled with Drug Chemical Information to Improve Drug Response Prediction

2.1 Chapter Introduction

Cell lines have long served as models to study molecular mechanisms of cancer. Cell lines harbor many of the mutations and transcriptomic alterations that are present in the primary tumor from which they were cultured. As pharmacogenomics models, cell lines offer the advantages of being easily grown, relatively inexpensive, and amenable to high-throughput genomic and pharmacokinetic testing. Data generated from cell lines can then be used to link cellular drug response to genomic features, where the ultimate goal is to build predictive signatures of patient outcome [28]. In colorectal cancer, cell lines have been used extensively to study mechanism of disease and biomarkers. Mouradov et al. showed that COAD/READ cell lines exhibit similarities in hypermutation profiles, particularly in DNA mismatch repair, chromosomal remodeling genes (e.g. ARD1A) and histone methylation genes to the colorectal cancers in TCGA [29]. Further, Daemen et al. used least-squared support vector machine (LS-SVM) to predict drug response across 90 compounds in breast cancer cell lines. They found that mutation frequencies of crucial genes was the same in cell lines as in their primary tumors, and that their gene panel for Tamoxifen predicted a significantly improved relapse-free survival for patients predicted to be tamoxifen-sensitive [30].

Furthermore, large datasets have been generated to characterize pharmacogenomic profiles of tumor-derived cell lines. One of the first these datasets was the NCI-60, which consisted of 60

cancer cell lines screened across several drugs [31]. Genomic features were also characterized for the same cell lines. The genomic and response data were collectively displayed on CellMiner [32]. Targeted study of a panel of breast cancer cell lines have led to insights into the pathways and process directly affected by anticancer compounds [33].

Additional pharmacogenomics datasets such as the Connectivity Map [34], Genomics of Drug Sensitivity in Cancer [35], the Cancer Cell Line Encyclopedia [36], the Cancer Therapeutics Response Portal [37], and the Cancer Target Discovery and Development Project (<https://ocg.cancer.gov/programs/ctd2>) have expanded the numbers of cell lines, drugs, and cancer types. The most recent version of the CCLE dataset includes the same tumor-derived cell lines sequenced with next generation sequencing, providing an up-to-date genomic profiling of this valuable resource [38]. Table 1 in this study [28] shows the number of tumor derived cell lines collected across all groups. These studies have led to advances in our understanding of cellular response to drugs and have provided the necessary data to develop prediction algorithms that aim to match the response with genomic features (The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease)

2.2 Generalizing Drug Response Prediction to Patients

Despite the large contribution that cell lines have had in our understanding of cancer biology and development, there are disadvantages associated with them. One of the main criticisms is that the environment the tumor-derived cell lines are developed in is very different from the environment that gave rise to the primary tumor [39]. Another is that while cell lines carry many of the aberrations that are represented in the original tumor, many of the passenger mutations (low frequency) are not maintained leading to a homogeneous population of cells that does not necessarily resemble the heterogeneity that has been observed in primary tumors [39]. Therefore, there is an unmet need to efficiently and systematically transform cell line sequencing

data rendering them more similar to that of their primary tumor, while maintaining hot-spot aberrations.

While precision medicine has benefited from a large increase in RNA-seq being carried out in the clinic, and from large consortia such as The Cancer Genome Atlas (TCGA), and the International Cancer Genome Consortium (ICGC), drug response data of large patient cohorts have not become readily available. Because of the fundamental differences between the in vitro and in vivo setting, translation from pharmacogenomics features derived from cells to the prediction of drug response in adult and childhood cancers has not yet been fully realized [40]

The majority of prior art in deep learning and drug design has focused on drug target interactions and drug-drug interactions, showing serious promise for reducing labor, cost, and time in the initial stages of drug development. However, there is still a need to test these molecules in model organisms, such as tumor-derived cell lines and mice, which also takes time and is not cost-effective. Using similar approaches in deep learning as discussed above, prediction of drug response in model organisms as well as further generalization to patients is possible. Due to the inherent differences between in vitro and in vivo biological systems, a translation of pharmacogenomics features derived from cells to the prediction of drug response of tumors is not fully realized. Prior work in this area has made use of the previous generation of CCLE data (sequenced with Affymetrix arrays) expression data and SMILES (Simplified Molecular-Input Line-Entry System) strings [41] to predict drug response in cell lines and aid in cancer-drug re-purposing without applying the model to patients. Chiu et al. used mutation and expression information from the newly sequenced (RNA-seq) CCLE cell lines to develop a deep learning framework for the prediction of drug response named DeepDR [42]. Although the performance of the model in cell lines is moderate (RMSE achieved 1.4 compared to 0.98 achieved by CDRscan), the model showed promise when applied to select patient cohorts, such as HER2(+) BRCA patients predicted to be significantly more sensitive to Tamoxifen. EGFR(+) NSCLC patients

were predicted to be significantly more sensitive to EGFR inhibitors Gefitinib and Afatinib than EGFR(-) patients, however Erlotinib was not shown to be sensitive in this patient cohort. Nevertheless, the model focuses primarily on adult patients and not on pediatric patients.

The following work aims at bridging the discipline of domain adaptation, to incorporate patient information regularization during training of a deep learning framework, and leveraging chemical structure information to improve drug response prediction in patients. I led this study and I was responsible for the analysis and writing. The contributions of others are as follows: Lucas Seninge contributed to the analysis and interpretation of results, and Hongxu Ding contributed to the refinement of the paper and the analysis.

This manuscript has been submitted to *International Journal of Environmental Research and Public Health*, on November 2021.

Embedding Drug Chemical Information with Patient-Cell Line Domain Adaptation to Predict Clinical Response

Ioannis Anastopoulos^{1,2,*}, Lucas Seninge^{1,2}, Hongxu Ding^{1,2,*}, Joshua Stuart^{1,2,*}

¹Department of Biomolecular Engineering, UC Santa Cruz, Santa Cruz, California, USA.

²UC Santa Cruz Genomics Institute, Santa Cruz, California, USA.

*Correspondence should be addressed to I.A. (ianastop@ucsc.edu), H.D. (hding16@ucsc.edu) or J.S. (jstuart@ucsc.edu).

ABSTRACT

In silico modeling of patient clinical drug response (CDR) promises to revolutionize personalized cancer treatment. State-of-the-art CDR predictions are usually based on cancer cell line drug perturbation profiles. However, prediction performance is limited due to the inherent differences between cancer cell lines and primary tumors. In addition, current computational models generally do not leverage chemical information of drugs. Here we develop the Prediction with Adaptation and Chemical Embedding (PACE) dual convergence deep learning framework that **a**) integrates gene expression along with drug chemical structures, and **b**) is adapted in an unsupervised fashion to primary tumor gene expression. We show that PACE outperforms state-of-the-art linear regularized method TG-LASSO in recapitulating drug efficacy (9/12 VS 3/12 drugs with available clinical outcomes).

GLOSSARY: *GCN*, Graph Convolutional Network. *MorganFP*, Morgan Fingerprint. *SMILES*, Simplified Molecular Input Line Entry System for annotating chemical structures using character strings. *ML/DL*, machine learning/deep learning, *CDR*, Clinician Drug Response. *CDI* Cell-line-Drug-IC₅₀. *EM* Expression Module. *DM* Drug Module. *PM* Prediction Module. *OOD* Out of Distribution.

INTRODUCTION

Precision medicine promises to revolutionize cancer treatment by improving clinical drug response (CDR) prediction. CDR prediction could be greatly facilitated by cutting-edge high-throughput sequencing technologies, which provide comprehensive and individualized omics profiles. Based on these omics profiles, several CDR prediction approaches have been proposed. For instance, TG-LASSO¹ integrates tissue-of-origin information with gene expression profiles for CDR prediction. DeepDR², on the other hand, predicts CDR from mutation and expression profiles.

However, as another crucial component for CDR prediction, the chemical properties of drugs have been under-utilized. Although the traditional Morgan Fingerprint (MorganFP) molecular representation³ has been used to integrate drug chemical information for CDR prediction, it cannot adaptively learn alternative representations of drug chemical properties, as it is a static representation of the molecule and does not dynamically extract features for the desired prediction task. For example, CDRscan uses MorganFP to represent key molecular

substructures with an explicitly defined featurization. A limitation of this specific methodology is its inability to adaptively learn alternative representations that may be beneficial to the particular task in hand⁴. DrugCell also uses MorganFP to represent drugs along with a Visible Neural Network (VNN) embedded in Gene Ontology (GO) terms, which provides interpretable results⁵, however it has the same limitation as CDRScan in terms of adaptively learning chemical features.

The Graph Convolutional Network (GCN)⁶ representation emerges as an attractive alternative for encoding drug chemical properties. GCN adaptively learns chemical information by generalizing the convolution operation from a grid of pixels to a graph, where each node can have a variable number of neighbors. GCNs have been used to explore drug-target interactions and side effect predictions - the two most important factors for developing a new drug. For instance, Decagon uses GCN to predict potential side effects of a drug⁷. DeepDrug is another such example that predicts drug-drug interactions⁸. Such methodological advances provide novel insights in incorporating drug chemical properties during CDR prediction.

The majority of CDR prediction algorithms are trained with cancer cell line drug preturbation profiles. Cell lines have long served as models to study molecular mechanisms of cancer, because they maintain valuable molecular information of the primary tumors from which they were derived. Cell lines offer the advantages of being easily grown, relatively inexpensive, and amenable to high-throughput assays. Data generated from cell lines can then be used to link cellular drug response to molecular features, where the ultimate goal is to build predictive signatures of patient outcomes⁹. Various models have been developed to predict patient CDR from the molecular profiles of cell lines¹⁰⁻¹². However, these models only show limited success in certain drugs¹³⁻¹⁴. Therefore, developing a model based on cell line molecular features to predict CDR in patients for most drugs remains challenging¹⁵. One major difficulty for such cross-domain CDR prediction is the prominent differences between cell lines and primary tumors¹⁶⁻²⁰. Recent advances in domain adaptation aim at aligning domains to tackle domain alignment problems, such as batch effect correction to reconcile differences across laboratories and studies²¹. Mean Maximum Discrepancy (MMD)²² has shown promising results in aligning domains in an unsupervised manner²³. Such a technique could be used to align cell lines and patient tumors in developing drug response models that are more clinically focused.

Inspired by the advanced GCN-based drug chemical information encoding, as well as the MMD-based domain adaptation, we develop a model for drug response Prediction with Adaptation and Chemical Embedding (PACE, Figure 1A). The PACE deep learning framework predicts drug efficacy by integrating compound chemical information extracted using GCN layers, with gene expression profiles encoded using fully connected neural network layers. Specifically, to make PACE more compatible for patient CDR predictions, when training the framework, we applied the MMD domain adaptation technique²⁴⁻²⁶ to implicitly align cell lines with reference patient samples of the same tissue of origins. Thus, the model does not assume that cell line and patient samples are drawn from the same distribution. We demonstrated the superior performance of PACE in recapitulating drug efficacy compared to the state-of-the-art linear regularized method TG-LASSO (9/12 VS 3/12 drugs with available clinical outcomes).

RESULTS

Overview of PACE

The PACE deep learning framework consists of three modules: Expression Module (EM), Drug Module (DM), and Prediction Module (PM). As shown in Figure 1A, the EM is composed of fully connected layers and learns highly informative features from gene expression vectors. The DM is composed of a GCN and learns highly informative features for each atom from the graph representation of chemical compounds. Atom-level features are then aggregated to represent information about the compound as a whole (see METHODS). Given the success GCNs have had in computational chemistry and biology applications²⁸⁻³⁰, we posited that the DM could learn a general graph embedding that would extend to drugs unseen during training. The PM is composed of a fully connected layer and takes the information learned from the EM and DM as input to predict $\log(\text{IC}_{50})$. The model was trained with “CDI tuples” -- Cell line gene expression, Drug SMILES, IC_{50} -- indicating the contribution of the cell line transcriptomic landscape and the drug chemical property to the cell line-drug IC_{50} value. Specifically, we included cell line gene expression profiles from the Cancer Cell Line Encyclopedia (CCLE) project³¹, and the cell line-drug IC_{50} values from the Genomics of Drug Sensitivity in Cancer (GDSC) project³².

Our goal is to extrapolate drug response from cell lines to patients. Hence, the model needs to embed the cell line training data to an out-of-distribution (OOD) embedding space representing patient samples. Inspired by²⁴, and recent advances in the field of domain adaptation²⁷, we used maximum mean discrepancy (MMD) to adapt the latent distribution produced by the EM so that cell lines could be aligned to patients. We excluded cell lines whose tissue-of-origins are not shared with TCGA reference samples when training PACE. We provide the samples from CCLE and TCGA with matching tumor types in Supplementary Tables 1 and 2. By this means, we collected 531 remaining cell lines treated across 310 drugs, amounting to 142,351 total CDI pairs. During the MMD domain adaptation step of the training process, each cell line was paired with a random TCGA reference sample of the same tissue of origin, in order to create a general enough adaptation of EM's latent space. We included in total 7,702 TCGA patient samples as MMD references.

To test the efficacy of adapting the EM with patient gene expression via MMD, we constructed a non-adapted version of PACE for comparison purposes. In addition, we also compared our model to one in which the DM uses MorganFP, representing a more conventional molecular encoding. Altogether, we created three alternative models closely related to PACE -- PACE-Morgan, noPACE, and noPACE-Morgan (see Table 1).

Table 1. All alternative models used and their specifications

Name	EM	DM
PACE	Adapted	GCN
PACE-Morgan	Adapted	MorganFP
noPACE	Not adapted	GCN
noPACE-Morgan	Not adapted	MorganFP

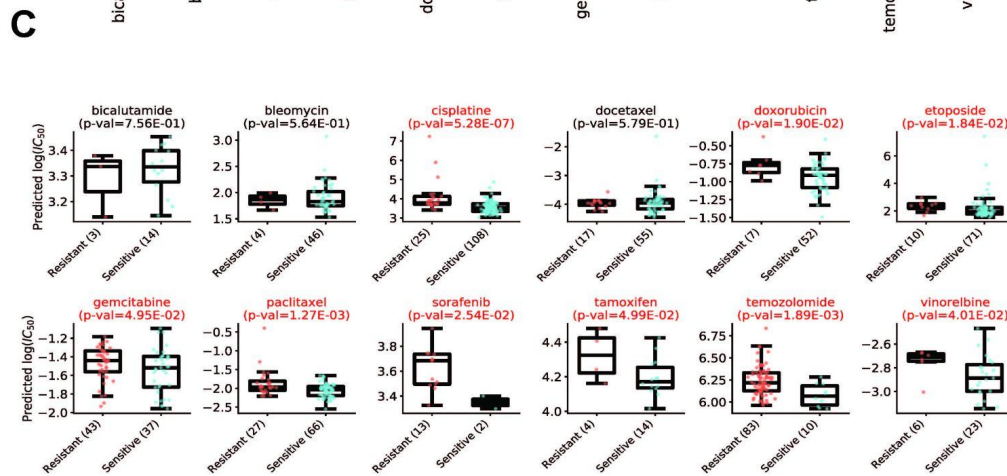
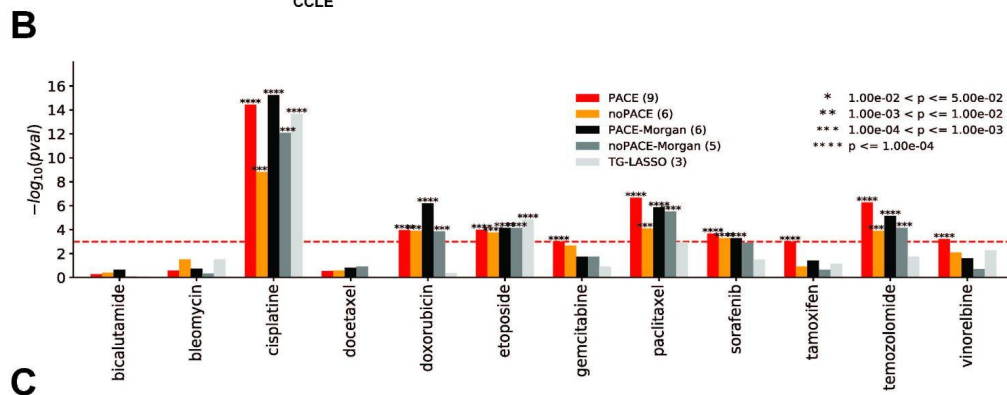
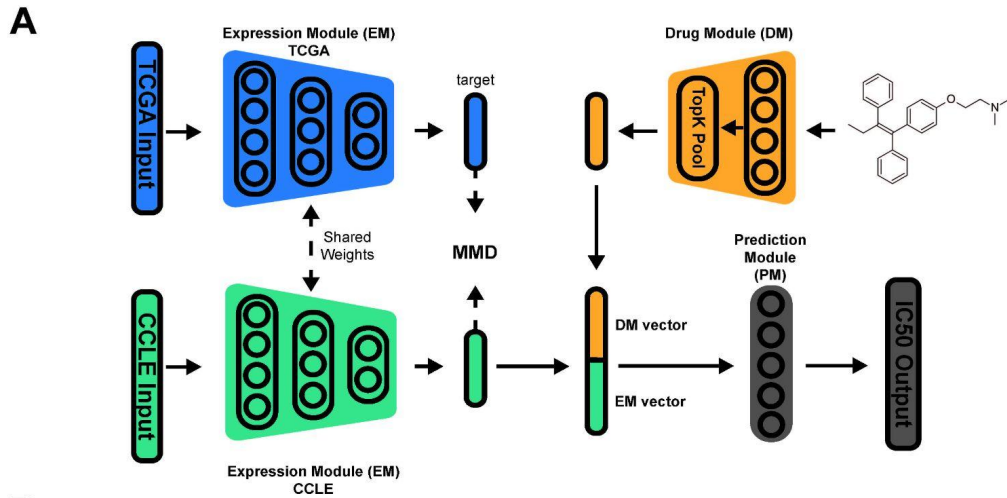
We compared the drug-level predictive performance (per-drug across-cell line Spearman Rho against ground truth) achieved by our proposed PACE model to all the other PACE-variations (Supplementary Figure 1A). We found that PACE does better compared to PACE-Morgan for the majority of the drugs (points above the diagonal), suggesting that in general drug chemical structures are better encoded by GCN. Meanwhile, the non-adapted noPACE and noPACE-Morgan achieve a slightly higher correlation (points below the diagonal) as opposed to PACE and PACE-Morgan, respectively. Such slightly compromised IC_{50} predictions for PACE and PACE-Morgan are expected, as MMD was introduced as an additional regularization term. Moreover, such a prediction performance decrease is negligible: besides slightly compromised drug-level predictive performance, when comparing across all 142,351 CDI pairs, the PACE model and the alternatives achieved comparable results (Supplementary Figure 1B). This suggests that MMD adaptation preserves the prediction performance of drug response in cell lines while yielding superior performance in the patient setting (Figure 1B).

In addition to PACE derivatives, we further compared PACE to the state-of-the-art TG-LASSO¹ model, which is a linear regularized method for CDR prediction. To evaluate all of the models in the patient setting, we followed the same evaluation presented in the TG-LASSO study¹. Specifically, we used the same curated CDR dataset consisting of 531 patients treated across 24 drugs labeled with the type of response indicated for each patient. Some drugs are used in multiple tumor types (e.g. cisplatin) whereas others are used in a restricted setting (e.g. bicalutamide in prostate cancer, tamoxifen in breast cancer, bleomycin in testicular cancer, see Supplementary Table 3 for details). The majority of patients in this dataset (70%) were treated with a single drug, while the rest were given two or more. Patients with stable diseases and clinical progressive diseases were labeled as resistant (R), whereas those with partial or complete response were labeled as sensitive (S). Following the same data filtering steps as TG-LASSO and further retaining samples for which we had expression information, 506 patients across 12 drugs remained. To measure the performance of the methods, we asked if their predicted $\log(IC_{50})$ drug response (a continuous measure) correlated to the drug response labels (R/S) (a categorical measure) in the CDR dataset (see METHODS). Specifically, a one sided Mann-Whitney U test was used to determine whether the predicted $\log(IC_{50})$ for the resistant (R) patients is significantly larger than that of sensitive (S) patients.

As shown in Figure 1B, sensitive patients across more drugs compared to all other models that lack such adaptation. The combination of patient information adaptation with MMD and GCN for drug embedding had better correlation to patient response than all the alternative

methods examined. Specifically, PACE showed significant discrimination between resistant and sensitive patients ($p < 0.05$) for nine out of the twelve drugs compared to six by noPACE (Figure 1B). Similarly, PACE-Morgan predicted six drugs, compared to five by noPACE-Morgan (Supplementary Figure 2). Please note that all PACE-variants could outperform TG-LASSO, which could only recapitulate the efficacy of three drugs.

Taken together, these results suggest that the combination of patient adaptation via MMD and a combination of the chemical embedding learned from GCN produced a highly informative model that can be extended to the patient setting.



Cell line diversity is crucially important for patient CDR prediction

Next, we asked if gene expression information or drug information has a bigger impact in predicting drug response in patients. To this end, we created two different dropout experiments -- one where all the CDI pairs for a *cell line* were withheld, and another in which all the CDI pairs for a *drug* were withheld. For the cell line dropout experiment, we created training sets with 20%, 40%, 60%, and 80% of the total cell lines (531). For the drug dropout experiment, we removed the 12 drugs presented in the CDR dataset, and created training sets that included 20%, 40%, 60%, and 80% of the remaining 298 drugs. For each group in cell line and drug dropout experiments, the PACE model was trained ten independent times. The ten independently trained models were then applied to the patient CDR dataset, and the average predicted $\log(\text{IC}_{50})$ was computed. The Mann-Whitney U test was used to evaluate the discrimination between labeled resistant and sensitive patients (see METHODS).

As summarized in Supplementary Figure 3A and B, lack of gene expression information had a bigger impact compared to the lack of drug information across all drugs in our CDR dataset, which is suggested by the higher prediction performance variance. This is likely caused by the vast difference in complexity and variance between the gene expression profiles and the compound structures. The robustness (measured by the variance of the p-value across 10 fold cross validation) of the model suffers more with 20% of the cell lines included in training compared to the same percentage of drugs included in training (Supplementary Figure 3B). Addition of more cell lines in the training set drastically improves robustness of the model as shown by the decreasing variance of the p-value across all 10 folds, indicative of the crucial role expression information plays in predicting drug response (Supplementary Figure 3A). This result also suggests that the GCN needs a small amount of drug chemical structures in the training set to be able to generalize well to new drugs not seen in the training set. To further demonstrate that the drug chemical structure diversity has been saturated, we removed all the 12 drugs in the CDR dataset during training the PACE model. As shown in Supplementary Figure 3C, compared to the full PACE performance, the efficacy of only tamoxifen and vinorelbine were no longer recapitulated. Please note that the significance level drop of the two drugs are marginal, suggesting that new drugs could be generalized with only a small amount of training drug chemical structures.

PACE predictions recapitulate knowledge on targeted therapy

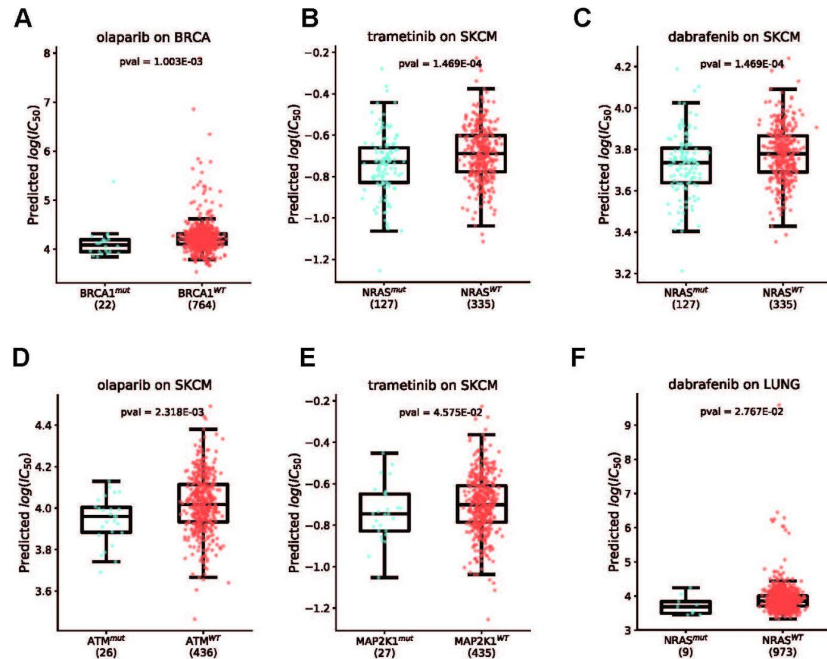
Most of the drugs we tested can be classified as chemotherapy agents, with the exception of sorafenib (VEGFR inhibitor), tamoxifen (*ESR1* inhibitor) and androgen receptor (AR) inhibitor bicalutamide. To assess the performance on other targeted agents for well characterized cohorts, we carried out an *in silico* analysis on drugs with known biomarkers of response. We used mutations as biomarkers of response for the TCGA cohorts where expression and mutation information were available. For the cohorts where these were not available we used expression as a biomarker of response to confirm that the model learns biologically meaningful information. The idea here is that as a target gene's expression increases, the drug's predicted IC_{50} should decrease accordingly, indicating an increase in sensitivity.

We collected mutation information from TCGA breast cancer (BRCA), melanoma (SKCM) samples and lung adenocarcinoma (LUAD) and lung squamous cell carcinoma cohorts (combined and abbreviated as LUNG). We used mutation information as a biomarker of sensitivity. We tested trametinib, olaparib, and dabrafenib on all of the aforementioned cohorts. Trametinib is a MEK inhibitor and used to treat SKCM, and dabrafenib is a *BRAF* inhibitor also used in SKCM. Olaparib is a PARP1 inhibitor and used to treat *BRCA1*- or *BRCA2*-mutated breast and ovarian (OV) cancers. Next, we examined the correlation between a drug's predicted $\log(IC_{50})$ and its target gene's expression (after Z-score transformation of the gene expression values). We specifically examined OV in this way due to the fact that we could not collect sufficient OV samples with predicted *BRCA1* or *BRCA2* mutations, and thus could not use *BRCA1/2* mutation as a biomarker for OV and olaparib.

As expected, *BRCA1* mutant samples were predicted to be more significantly sensitive to olaparib compared to *BRCA1* WT samples (Figure 2A), *NRAS* and *MAP2K1* SKCM mutants were predicted to be significantly more sensitive to trametinib compared to SKCM WT samples (Figure 2B/E). *NRAS* SKCM mutants were additionally predicted to be more sensitive to dabrafenib compared to *NRAS* WT samples (Figure 2C). Olaparib has been previously shown to be effective in *ATM* mutated BRCA patients. Although our model was not able to predict this association correctly, SKCM *ATM* mutated samples were predicted to be sensitive to olaparib (Figure 2D). Lastly, LUNG cancer *NRAS* mutated samples were predicted to be more sensitive to dabrafenib (Figure 2F).

Studies have pointed to *PARP* expression as a promising biomarker of olaparib response³³. When we examined the correlation between *PARP1* z-score and the predicted $\log(IC_{50})$ per disease, OV had a significantly negative Spearman Rho ($\rho=-0.48$) (Supplementary Figure 4A). Testicular cancer (TGCT) showed the strongest negative correlation between *PARP1* expression and predicted IC_{50} for olaparib, which has recently been in clinical trials in combination with chemotherapy for TGCT³⁴.

We further examined the predicted response of lapatinib and the correlation with the expression of its target genes, *EGFR* and *ERBB2*. *In-vitro* studies have previously shown that lapatinib inhibits cell proliferation and migration of breast cancer cell lines expressing different levels of *EGFR* and *ERBB2*, and that cells overexpressing *ERBB2* were more sensitive³⁵. Interestingly, our model predicted *EGFR* expression as a stronger biomarker (Supplementary Figure 4B) in BRCA patients compared to *ERBB2* expression (Supplementary Figure 4C). Taken together, these results suggest that our model can recapitulate the relationship of well characterized drugs with the appropriate biomarkers, and their applicability in equally well-characterized cohorts.



DISCUSSION

In this study, we presented PACE, a new deep learning framework that uses both a graph convolutional network (GCN) as a general encoding for drug information together with patient information to aid in out-of-dataset prediction. During training, the method aligns cell-line and patient gene expression domains using implicit tissue-driven adaptation together with drug information to derive highly informative features for drug response prediction.

We showed that adapting cell lines to tumor gene expression space with maximum mean discrepancy (MMD) preserves performance in cell lines while improving the prediction of clinical drug response (CDR) in patients regardless of the drug encoding used. We found that GCN's embedding extends to drugs that have not been seen in training. These results suggest that a combination of implicit tissue-driven adaptation and a highly flexible drug encoding lead to improved prediction performance in patient samples. Interestingly, we note that the drug dropout experiments revealed that only a random 20% (60) of drugs are needed to yield robust generalization performance. On the other hand, the cell line dropout experiments showed that a lack of cell line diversity during training greatly impairs generalization of drug response in patients. Such discoveries shed light on the design of training datasets when extending PACE to larger-scale patient CDR prediction tasks. To test whether PACE could be used for real-world applications, we examined our model on some of the well known targeted therapeutics for melanoma, breast cancer, and lung cancer. We found that PACE was able to predict *MEK2* mutant melanomas as significantly more sensitive to trametinib, a MEK inhibitor, compared to the WT cohort. Similarly, *BRCA1* mutants in breast cancer were significantly more sensitive to olaparib, a first-line treatment to patients with such a mutation, compared to the WT cohort.

Ideally, PACE should be trained on patient samples rather than surrogates such as cell lines. However, at this time, an adequate amount of patient data is lacking for any particular drug of interest as most patients receive the standard of care based on the tissue of origin. For example, as shown in Supplementary Table 3 all of the testicular germ cell tumors (TGCT) patients were treated with bleomycin, whereas there were no TGCT cell lines treated with the same drug. In the coming years, single cell sequencing should improve the performance of predictive models. Promising results have been published by the MIX-seq study in which the sequencing of cell lines before and after drug treatment has detailed the heterogeneity in response across individual cancer cells³⁶. Together with single-cell sequencing, human-derived xenografts and 3D human organoids should complement cell line studies to add needed realism for model training; e.g. by including contributions from the microenvironment³⁷⁻³⁹.

PACE can be extended to incorporate additional diverse biological data as it becomes available. As expected, we found that accuracy depended heavily on the presence of an appropriate cell line of a matching tissue type in the training data. Beyond extending the training data to cover more cell lines, which will increase the diversity of patients to which the method can be applied, other data types may also provide a boost in performance. For example, the current work focuses on gene expression and does not consider genomic alterations, such as mutations and structural variants, and the vulnerabilities that these may introduce. In particular, the CDR prediction on bicalutamide could be better with adding mutation information to account for Antiandrogen withdrawal syndrome (AAWS) caused by mutations in the AR, which is known to confer resistance^{40,41}. Genetic dependency data generated from the ACHILLES project⁴² for example are now available for many of the same cell lines that our model was trained on. In theory, incorporating synthetic lethality prediction into the model should improve drug response prediction, as drugs that target synthetically lethal pairs should have a substantial impact on drug response. Incorporating protein level information could also lead to improvements in performance as many of the drugs target specific proteins whose expression may or may not be correlated with the gene's RNA. The ongoing CPTAC project⁴³ is systematically quantifying protein levels and phosphorylation states in cancer patients from TCGA. In addition, it has been shown that proteome-level characterization of cell lines can aid in drug response prediction⁴⁴. It is therefore evident that addition of proteomic data to our model could have a significant impact on the prediction of drug response. Furthermore, the PACE framework could be repurposed to adapt any source distribution to any target distribution. For instance, xenograft space could be adapted to a patient space. We leave such explorations to future studies.

Increasing the interpretability of PACE would be of great value. It would be very informative to developers of new drugs if they could predict the pathways affected by administration of a new treatment. Recent advances in developing more interpretable biological models^{45,46} should help models like ours in providing generalizable and interpretable results. Lastly, the GCN of our model uses only atomic features for drug encoding. Other types of GCN, such as GINEConv⁴⁷, are more expressive and use both atomic- and bond-level features, which could potentially create an even more generalizable drug embedding. We leave the exploration of the most appropriate GCN for this task and the inclusion of an interpretable EM to future work.

METHODS

Overall Framework. Our model is an adapted dual convergence architecture that integrates gene expression information with drug structure aimed at generalizing clinical drug response (CDR) prediction in patients. It consists of three modules: Expression Module (EM), Drug Module (DM) and Prediction Module (PM). Highly informative representations of gene expression and drug structure are generated by the EM and DM, respectively. These representations are jointly passed to the PM where the $\log(\text{IC}_{50})$ prediction is made. The model takes as input a cell line expression vector (\mathbf{x}_c), a primary tumor expression vector (\mathbf{x}_t), and the compound that was applied on the cell line. The way the compound is presented as input to the model is explained in the **Morgan Fingerprint (MorganFP) Representation of Drugs** and **Graph Representation of Drugs** sections.

Expression Module (EM). The EM consists of 2 fully connected layers of 1024, and 100 nodes with Rectified Linear Unit (ReLU) activation. BatchNormalization and Dropout of 0.35 are applied on each layer. During training, the EM produces latent representations for both cell line and primary tumors via weight sharing as follows:

$$f_{EM}(\mathbf{x}_c; \theta_{EM}) = z_c, \text{ and}$$

$$f_{EM}(\mathbf{x}_t; \theta_{EM}) = z_t,$$

where z_c and z_t represent the latent vectors of the cell line and primary tumor, respectively.

Inspired by the field of domain adaptation, and driven by the need to generalize drug response prediction to patients, we used a domain alignment method called Mean Maximum Discrepancy (MMD)²². Specifically, the model tries to align z_c to z_t with the goal of making the cell line latent space more similar to the primary tumor latent space by minimizing the following loss:

$$L_{MMD}(z_c, z_t) = k(z_c, z_c) + k(z_t, z_t) - k(z_t, z_c) - k(z_c, z_t),$$

where k denotes the universal Gaussian kernel. Each z_c and z_t represent the same tissue-of-origin during training. Thereby, the model implicitly aligns cell lines and primary tumors in a tissue driven manner.

MorganFP Representation of Drugs. We used the python library RDKit to generate Simplified Molecular Input Line Entry System (SMILES) strings, which describe the structure of a molecule using a single line of text, and compute MorganFP for each molecule in our datasets⁴⁸. SMILES strings are simple string annotations that describe the structure of the molecule. MorganFP is part of the Extended-Connectivity Fingerprints (ECFPs) family and are generated using the

Morgan algorithm^{3,49}. These fingerprints represent molecular structures and the presence of substructures by means of circular atom neighborhoods (bond radius). In this study we used radius 2 and constructed a 2048 long bit vector for each molecule. A radius of 2 takes into account neighbors up to two atoms away when constructing the bit vector (fingerprint) of the molecule.

Graph Representation of Drugs. We used RDKit to generate SMILES strings for each drug. Next, we represented the SMILES string for each compound $\{c_j\} \in \mathcal{C}$ as a graph

$G = \{V, X\}$, where $V = \{v_j\}$ represents the set of nodes (nodes here are atoms on the molecule). An adjacency matrix A represents the topological structure of each molecule with $A_{i,j} = 1$ denoting a bond between two atoms, otherwise $A_{i,j} = 0$. $x_i \in X$ indicates the vector of features for each atom v_i on the compound. The features (189 in total) used for each compound can be found in Table 2.

Table 2. Description of Atomic Features

Atom feature	Size	Description
Atom symbol	19	[As, B, Br, C, Cl, F, Hg, I, K, N, Na, O, P, Pt, S, Sb, Se, V, Zn] (one-hot)
Atomic Number	119	Atomic number of each atom (one-hot)
Chirality type	4	[UNSPECIFIED, R, S, OTHER]
Degree	11	Number of covalent bonds (one-hot)
Formal Charge	12	Electrical charge (one-hot)
Hydrogens	9	Number of connected hydrogens (one-hot)
Radical Electrons	5	Number of radical electrons (one-hot)
Hybridization	8	[UNSPECIFIED, sp, sp2, sp3, sp3d, sp3d2, OTHER] (one-hot)
Aromatic	2	Atom is an aromatic ring (one-hot)

Total	189	Total number of features
-------	-----	--------------------------

Drug Module (DM). The DM of the model aims at extracting highly informative features from each molecule. This is done via either the MorganFP representation of the molecule, or the graph representation of the molecule. For the former, the DM consists of one fully connected layer, ReLU, BatchNormalization and Dropout. For the latter, we used the python library PyTorch Geometric to produce data-driven molecular features using GCN⁵⁰. In particular, we used the GCN architecture from⁵¹. That architecture learns substructures of a given graph, and relationships between graphs, which is crucial in this study as we aim to generate a general embedding space for structurally diverse molecules presented in the drug response dataset. This type of GCN falls under the spatial GCN category, which can generalize the learned embedding to heterogeneous graphs⁵². We used one layer, followed by a pooling layer, which aggregates highly informative nodes on the molecular graph⁵³. The DM consists of one layer due to the small average size of the molecules (34 nodes). ReLU, BatchNormalization and Dropout were applied here as well.

The purpose of the GCN is to map each $v_i \in V$ to low dimensional vectors $z_i \in R$. The formal mapping is as follows:

$$f_{DM}(A, X; \theta_{DM}) \rightarrow Z,$$

with $Z \in R^{n \times d}$ for compound C_j , where n is the number of atoms, and d is the dimension of the latent space produced by the GCN.

Furthermore, to obtain a latent representation Z_d for graph C_j , we computed both average and maximal features across Z and concatenated them with the following operation:

$$z_d = \left(\frac{1}{n} \sum_{i=1}^n Z_{i,j} \parallel \max_{1 \leq i \leq n} Z_{i,j} \right)$$

, where $z_d \in R^{n \times 2d}$ and \parallel denotes the concatenation operator. The dimensionality is doubled due to concatenation of both average and maximal features for each graph.

Prediction Module (PM). The PM of the model consists of one fully connected layer, and aims at predicting $\log(\text{IC}_{50})$ using highly informative features derived from the EM and DM. As such, the operation carried out by the PM is the following:

$$f_{PM} = (z_c \| z_d; \theta_{PM})$$

Our model updates the weights of EM, DM, and PM by minimizing the mean squared error (MSE),

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

where N is the number of samples,

between the observed and predicted $\log(\text{IC}_{50})$, denoted by y and \hat{y} , respectively, and L_{MMD} .

Hence, the overall loss minimized by PACE is:

$$L_{PACE} = L_{MSE} + \lambda \cdot L_{MMD}$$

where λ controls the tradeoff between the goals of aligning the cell line latent space with the primary tumor latent space, and achieving an accurate predicted $\log(\text{IC}_{50})$.

Training Procedure and Tuning. Our model was implemented in Python with the PyTorch API⁵⁴ using the Adam optimizer⁵⁵ for gradient descent optimization. The training was allowed to proceed for a maximum of 200 epochs. To control for overfitting EarlyStopping was used to monitor the training loss for overfitting. Training was terminated after 10 epochs if the training loss was not further minimized after 10 consecutive epochs, with a delta of 0.05. Dropout was applied on a random 35% of nodes to further prevent overfitting. We used the Adam⁵⁵ optimizer for gradient descent optimization with a learning rate of 1E-4. Given the stochasticity of the training procedure, and that we wanted to achieve considerable robustness with our model when predicting CDR of patients, we repeated the training 10 independent times.

Due to the computational expense of training, the number of layers for the DM and PM were fixed to one, and the number of layers of the EM were fixed to 2. We experimented with the λ , and with the number of drug nodes for the DM, as well as the number of layers of the EM.

We found that $\lambda = 0.01$ and 200 drug nodes were the best parameters for distinguishing sensitive from resistant patients in the CDR dataset

CDR prediction in TCGA patients. We obtained the clinical drug response (CDR) of 531 TCGA patients across 24 drugs from this study ¹. Following the same filtering steps as Huang et al. resulted in 12 drugs. Finally, after filtering for patients for which we had gene expression information resulted in 506 patients. Patients with “clinical progressive disease” or “stable disease” were labeled as resistant (**R**). Those with “partial response” or “complete response” were labeled sensitive (**S**). These are categorical variables, whereas our model predicts $\log(IC_{50})$ which is a continuous variable. To test how well our model can be extended to OOD samples, we grouped the predicted $\log(IC_{50})$ of each patient in the corresponding R or S bin. Then, we tested if the predicted $\log(IC_{50})$ of the R patients was significantly larger than that of the S patients by performing a one-sided nonparametric Mann Whitney U test. A summary of the number of R and S patients for each drug is shown in Table 3.

Table 3. Number of Resistant and Sensitive Patients in TCGA CDR dataset

Drug	Num Resistance	Num Sensitive	N of cell lines in training	Mode of Action
bicalutamide	3	14	525	Androgen receptor antagonist
bleomycin	4	46	470	DNA synthesis inhibitor
cisplatin	25	108	524	DNA synthesis inhibitor
docetaxel	17	55	524	Tubulin polymerization inhibitor
doxorubicin	7	52	479	Topoisomerase inhibitor
etoposide	10	71	484	Topoisomerase inhibitor
gemcitabine	43	37	509	Ribonucleotide reductase inhibitor
paclitaxel	27	66	470	Tubulin polymerization inhibitor
sorafenib	13	2	470	FLT3 inhibitor

tamoxifen	4	14	520	ESR1 inhibitor
temozolomide	83	10	520	DNA alkylating agent
vinorelbine	6	23	513	Tubulin polymerization inhibitor

Drug/cell line exclusion experiment. For dropout analysis, we created random train splits in a 10-fold cross validation. After training on each fold 10 independent times, we tested the generalizability potential of our model in the CDR dataset for each fold, thereby producing 10 p-values (see **CDR prediction in TCGA patients**). For drug-centered dropout analysis, we created train sets by first removing all 12 CDR drugs (bicalutamide, bleomycin, cisplatin, docetaxel, doxorubicin, etoposide, gemcitabine, paclitaxel, sorafenib, tamoxifen, temozolomide, and vinorelbine), and then retaining random 20%, 40%, 60%, and 80% of the remaining 298 drugs (310 drugs in total). Similar to the drug-centered dropout analysis, the cell line-centered dropout was carried out in a similar manner without removing the 12 CDR drugs.

AUTHOR CONTRIBUTIONS

I.A. conceived the idea, performed deep learning framework modeling, optimization and analysis. L.S. contributed in developing ideas on how to align cell lines and patients. H.D., J.S. supervised the project. All authors prepared the manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

CODE AVAILABILITY

The package and API for PACE is available at <https://github.com/ioannisa92/PACE>. The code and data to train and deploy the model of this manuscript are available on the github.

DATA AVAILABILITY

Expression Datasets. We downloaded gene expression data of 1376 cell lines of the Cancer Cell Line Encyclopedia (CCLE) project, along with their metadata³¹, and 10,536 TCGA pan-cancer tumors from the DepMap project⁵⁶ and UCSC Xena browser⁵⁷, respectively. All expression values were represented as $\log_2(\text{TPM}+1)$, where TPM denoted transcripts per million reads of each gene in each sample. The gene space was intersected resulting in 31,501 common genes.

Drug Response Datasets. We downloaded release 8.1 of the GDSC project containing drug response measured by the half maximal inhibitory concentration (IC_{50}) from the DepMap project, which has harmonized cell lines and drug names^{32,58}. In total 974 cell lines tested across 398 drugs are included in this dataset, amounting to 387,626 cell line-drug- IC_{50} pairs (CDI pairs). After intersecting for cell lines included in the CCLE RNA-Seq compendium, selecting drugs for which we could obtain SMILES string, removing CDI pairs representing combination therapies and pairs with missing values for either drug name or IC_{50} , 692 cell lines tested on 310 drugs remained, amounting to 185,186 CDI pairs. All IC_{50} values were transformed to log scale $\log_{10}(IC_{50})$. After selecting for cell lines that represent the same tissue of origin as the TCGA dataset (25 tumor types), 531 cell lines tested on 310 drugs amounting to 142,351 CDI pairs.

FIGURE LEGENDS

Figure 1. Deep learning model architecture and performance comparison of PACE. (A) Graphic overview of the proposed deep learning framework PACE. Expression Module (EM) extracts informative features for the input expression vectors for both CCLE and TCGA via shared weights. These two compact expression representations are compared with each other via Mean Maximum Discrepancy (MMD) to diminish the distance between them, thereby aligning the two representations. The Drug Module (DM) encodes the molecule and pools the most informative nodes (atoms) to also create a compact representation. Finally, the CCLE expression representation and the drug representation are concatenated together and passed to the Prediction Module (PM) that makes the final $\log(IC_{50})$ prediction for each CDI pair. (B) PACE was compared to non-adapted alternatives as well as alternatives using MorganFP for molecule encoding, against the state-of-the-art TG-LASSO linear method. Bar plots showcasing p-value, corresponding to the one-sided Mann-Whitney U test, determined by averaging 10 independent predictions made by each model. (C) Box plots reflecting the distribution of estimated $\log(IC_{50})$ values using PACE for resistant or sensitive patients. The p-values here also correspond to a one-sided Mann-Whitney U test.

Figure 2. Functional Analysis. (A) Predicted $\log(IC_{50})$ for *BRCA1* mutant and wild-type (WT) breast cancer (BRCA) samples *in-silico* treated with olaparib. P-value corresponds to the one-sided Mann Whitney U test discriminating between mutant and WT predicted $\log(IC_{50})$. (B) Predicted $\log(IC_{50})$ for *NRAS* mutant and wild-type (WT) melanoma (SKCM) samples *in-silico* treated with trametinib. (C) Predicted $\log(IC_{50})$ for *NRAS* mutant and wild-type (WT) melanoma (SKCM) samples *in-silico* treated with dabrafenib. (D) Predicted $\log(IC_{50})$ for *ATM* mutant and wild-type (WT) melanoma (SKCM) samples *in-silico* treated with olaparib. (E) Predicted $\log(IC_{50})$ for *MAP2K1* mutant and wild-type (WT) melanoma (SKCM) samples *in-silico* treated with trametinib. (F) Predicted $\log(IC_{50})$ for *NRAS* mutant and wild-type (WT) LUSC/LUAD (LUNG) samples *in-silico* treated with dabrafenib.

BIBLIOGRAPHY

1. Huang, E. W., Bhope, A., Lim, J., Sinha, S. & Emad, A. Tissue-guided LASSO for prediction of clinical drug response using preclinical samples. *PLoS Comput. Biol.* **16**, e1007607 (2020).
2. Chiu, Y.-C. *et al.* Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC Med. Genomics* **12**, 18–18 (2019).
3. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
4. Mayr, A., Klambauer, G., Unterthiner, T. & Hochreiter, S. DeepTox: Toxicity Prediction using Deep Learning. *Front. Environ. Sci. Eng. China* **3**, 80 (2016).
5. Kuenzi, B. M. *et al.* Predicting Drug Response and Synergy Using a Deep Learning Model of Human Cancer Cells. *Cancer Cell* **38**, 672–684.e6 (2020).
6. Kipf, T. N. & Welling, M. Variational Graph Auto-Encoders. *arXiv [stat.ML]* (2016).
7. Zitnik, M., Agrawal, M. & Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34**, i457–i466 (2018).
8. Cao, X., Fan, R. & Zeng, W. DeepDrug: A general graph-based deep learning framework for drug relation prediction. *biorxiv* (2020).
9. Goodspeed, A., Heiser, L. M., Gray, J. W. & Costello, J. C. Tumor-Derived Cell Lines as Molecular Models of Cancer Pharmacogenomics. *Mol. Cancer Res.* **14**, 3–13 (2016).

10. Costello, J. C. *et al.* A community effort to assess and improve drug sensitivity prediction algorithms. *Nat. Biotechnol.* **32**, 1202–1212 (2014).
11. Jiang, P., Sellers, W. R. & Liu, X. S. Big Data Approaches for Modeling Response and Resistance to Cancer Drugs. *Annu Rev Biomed Data Sci* **1**, 1–27 (2018).
12. Ali, M. & Aittokallio, T. Machine learning and feature selection for drug response prediction in precision oncology applications. *Biophys. Rev.* (2018)
doi:10.1007/s12551-018-0446-z.
13. Geeleher, P., Cox, N. J. & Huang, R. S. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome Biol.* **15**, R47 (2014).
14. Falgreen, S. *et al.* Predicting response to multidrug regimens in cancer patients using cell line experiments and regularised regression models. *BMC Cancer* **15**, 235 (2015).
15. Geeleher, P. *et al.* Discovering novel pharmacogenomic biomarkers by imputing drug response in cancer patients from large genomics studies. *Genome Res.* **27**, 1743–1751 (2017).
16. Yu, K. *et al.* Comprehensive transcriptomic analysis of cell lines as models of primary tumors across 22 tumor types. *Nature Communications* vol. 10 (2019).
17. Domcke, S., Sinha, R., Levine, D. A., Sander, C. & Schultz, N. Evaluating cell lines as tumour models by comparison of genomic profiles. *Nat. Commun.* **4**, 2126 (2013).
18. Chen, B., Sirota, M., Fan-Minogue, H., Hadley, D. & Butte, A. J. Relating hepatocellular carcinoma tumor samples and cell lines using gene expression data

- in translational research. *BMC Medical Genomics* vol. 8 (2015).
19. Jiang, G. *et al.* Comprehensive comparison of molecular portraits between cell lines and tumors in breast cancer. *BMC Genomics* **17 Suppl 7**, 525 (2016).
 20. Vincent, K. M., Findlay, S. D. & Postovit, L. M. Assessing breast cancer cell lines as tumour models by comparison of mRNA expression profiles. *Breast Cancer Res.* **17**, 114 (2015).
 21. Ge, S., Wang, H., Alavi, A., Xing, E. & Bar-Joseph, Z. Supervised Adversarial Alignment of Single-Cell RNA-seq Data. doi:10.1101/2020.01.06.896621.
 22. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **13**, 723–773 (2012).
 23. Louizos, C., Swersky, K., Li, Y., Welling, M. & Zemel, R. The Variational Fair Autoencoder. *arXiv [stat.ML]* (2015).
 24. Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional out-of-distribution generation for unpaired data using transfer VAE. *Bioinformatics* **36**, i610–i617 (2020).
 25. Long, M., Cao, Y., Wang, J. & Jordan, M. Learning Transferable Features with Deep Adaptation Networks. in *Proceedings of the 32nd International Conference on Machine Learning* (eds. Bach, F. & Blei, D.) vol. 37 97–105 (PMLR, 2015).
 26. Zhang, X., Yu, F. X., Chang, S.-F. & Wang, S. Deep Transfer Network: Unsupervised Domain Adaptation. *arXiv [cs.CV]* (2015).
 27. Farahani, A., Voghoei, S., Rasheed, K. & Arabnia, H. R. A Brief Review of Domain Adaptation. *arXiv [cs.LG]* (2020).
 28. Pham, T.-H., Qiu, Y., Zeng, J., Xie, L. & Zhang, P. A deep learning framework for

high-throughput mechanism-driven phenotype compound screening.

doi:10.1101/2020.07.19.211235.

29. Stokes, J. M. *et al.* A Deep Learning Approach to Antibiotic Discovery. *Cell* **181**, 475–483 (2020).
30. Keshavarzi Arshadi, A., Salem, M., Collins, J., Yuan, J. S. & Chakrabarti, D. DeepMalaria: Artificial Intelligence Driven Discovery of Potent Antiplasmodials. *Front. Pharmacol.* **10**, 1526 (2019).
31. Ghandi, M. *et al.* Next-generation characterization of the Cancer Cell Line Encyclopedia. *Nature* **569**, 503–508 (2019).
32. Iorio, F. *et al.* A Landscape of Pharmacogenomic Interactions in Cancer. *Cell* **166**, 740–754 (2016).
33. Oplustilova, L. *et al.* Evaluation of candidate biomarkers to predict cancer cell sensitivity or resistance to PARP-1 inhibitor treatment. *Cell Cycle* **11**, 3837–3850 (2012).
34. Yi, M. *et al.* Advances and perspectives of PARP inhibitors. *Exp. Hematol. Oncol.* **8**, 29 (2019).
35. Gril, B. *et al.* Effect of lapatinib on the outgrowth of metastatic breast cancer cells to the brain. *J. Natl. Cancer Inst.* **100**, 1092–1103 (2008).
36. McFarland, J. M. *et al.* Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nat. Commun.* **11**, 4296 (2020).
37. Chen, J. *et al.* Single-cell transcriptome analysis identifies distinct cell types and niche signaling in a primary gastric organoid model. *Scientific Reports* vol. 9 (2019).

38. Krieger, T. G. *et al.* Single-cell analysis of patient-derived PDAC organoids reveals cell state heterogeneity and a conserved developmental hierarchy. *bioRxiv* 2020.08.23.263160 (2021) doi:10.1101/2020.08.23.263160.
39. Chen, K.-Y. *et al.* Single-Cell Transcriptomics Reveals Heterogeneity and Drug Response of Human Colorectal Cancer Organoids. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2018**, 2378–2381 (2018).
40. Hara, T. *et al.* Novel mutations of androgen receptor: a possible mechanism of bicalutamide withdrawal syndrome. *Cancer Res.* **63**, 149–153 (2003).
41. Lorente, D. *et al.* Switching and withdrawing hormonal agents for castration-resistant prostate cancer. *Nat. Rev. Urol.* **12**, 37–47 (2015).
42. McFarland, J. M. *et al.* Improved estimation of cancer dependencies from large-scale RNAi screens using model-based normalization and data integration. *Nat. Commun.* **9**, 4610 (2018).
43. Rodriguez, H., Zenklusen, J. C., Staudt, L. M., Doroshow, J. H. & Lowy, D. R. The next horizon in precision oncology: Proteogenomics to inform cancer diagnosis and treatment. *Cell* **184**, 1661–1670 (2021).
44. Frejno, M. *et al.* Proteome activity landscapes of tumor cell lines determine drug responses. *Nat. Commun.* **11**, 3639 (2020).
45. Seninge, L., Anastopoulos, I., Ding, H. & Stuart, J. Biological network-inspired interpretable variational autoencoder. *Cold Spring Harbor Laboratory* 2020.12.17.423310 (2020) doi:10.1101/2020.12.17.423310.
46. Ma, J. *et al.* Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* **15**, 290–298 (2018).

47. Hu, W. *et al.* Strategies for Pre-training Graph Neural Networks. *arXiv [cs.LG]* (2019).
48. Landrum, G. & Others. RDKit: Open-source cheminformatics. (2006).
49. Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **5**, 107–113 (1965).
50. Fey, M. & Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv [cs.LG]* (2019).
51. Morris, C. *et al.* Weisfeiler and leman go neural: Higher-order graph neural networks. in *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 33 4602–4609 (2019).
52. Such, F. P. *et al.* Robust Spatial Filtering With Graph Convolutional Neural Networks. *IEEE Journal of Selected Topics in Signal Processing* vol. 11 884–896 (2017).
53. Gao, H. & Ji, S. Graph U-Nets. in *Proceedings of the 36th International Conference on Machine Learning* (eds. Chaudhuri, K. & Salakhutdinov, R.) vol. 97 2083–2092 (PMLR, 2019).
54. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv [cs.LG]* (2019).
55. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2014).
56. Tsherniak, A. *et al.* Defining a Cancer Dependency Map. *Cell* **170**, 564–576.e16 (2017).

57. Goldman, M. J. *et al.* Visualizing and interpreting cancer genomics data via the Xena platform. *Nat. Biotechnol.* **38**, 675–678 (2020).
58. Picco, G. *et al.* Functional linkage of gene fusions to cancer cell fitness assessed by pharmacological and CRISPR-Cas9 screening. *Nat. Commun.* **10**, 2198 (2019).

Chapter 3 : Biological Pathway Informed Deep Learning

Recapitulates Known Biology

3.1 Chapter Introduction

Although cancer cell lines have played critical role in our understanding of cancer biology and how tumors might respond to various treatments, cell lines do not adequately represent their tumor type. Chen et al compared hepatocellular carcinoma primary tumor samples to cell lines using gene expression data and showed that only about half of the cell lines sufficient resemble their primary tumors [43]. Vincent et al performed similar analysis integrating use gene expression data for breast cancer cell lines and primary tumors and reported similar results with Chen et al [44]. With the revised CCLE compendium there is a unique opportunity to study the differences and commonalities between the two cohorts. The compendium now contains over 1,000 cell lines across 36 tumor types sequenced with next generation RNA-seq [38]. However, the differences between cell lines and primary tumors are still apparent. Yu et al identified important differences between cell lines and tumors. Namely, using the top 5k most variable genes and GSEA, the authors were able to identify cell cycle related pathways upregulated in cell lines, and a moderate upregulation of immune system related pathways in primary tumors. The upregulation of cell cycle pathways in CCLE, is potentially reflective of the culturing conditions [45]. These differences are important, however, to improve our understanding of disease biology, and identify pathways that contribute to sensitivity or resistance to treatment, we need to identify common pathways between cell lines and tumors. This is important, because pathways provide a more holistic understanding of biology than genes. One of the shortcoming of established methods, such as GSEA, is that the transcriptome has to be restricted to the most important genes. Deep learning methods are not constrained by this, and may identify new and novel pathways. In addition, the authors attempt to remedy the issue of poor translatability between

the two cohorts by creating a new cancer cell line compendium called the TCGA110-CL. Its comprised of the top 5 most correlated cell lines for each of the 22 common tumor types between CCLE and TCGA.

Deep learning models usually act as black boxes learning a mapping from input to output. However, it is possible to embed an interpretable structure in the hidden layers of the model. Ma et al used extensive knowledge of cell biology to create a Visible Neural Network (VNN) named DCell. The hidden layers of DCell are hierarchical and resemble the inner workings of a eukaryotic cell, namely the budding yeast *Saccharomyces cerevisiae*. The model was trained so several million genotypes to predict cell growth. The authors showed that the model could make accurate phenotypic predictions for cell growth, and that it can outperform an equivalent fully connected artificial neural network (ANN) [46]. By embedding the structure of the model with prior knowledge of biological systems not only enables competitive performance, but also transparent biological interpretation of the predictions the model makes.

The following work aims at developing a variational autoencoder (VAE) that is not a black box. It can be interpreted to identify the biological pathways responsible for transcriptomic alteration following perturbation. The model takes as input un-stimulated (control) cell lines and predicts their perturbed transcriptomic profile. The decoding of the model is embedded with prior biological pathways which are connected to the respective genes that they comprise. The bottleneck space can then give us the most active pathways following perturbation.

The manuscript has been published at *Nature Communications*, 2021.

3.2 Biological Network-Inspired Interpretable Variational Autoencoder

VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics

Lucas Seninge¹, Ioannis Anastopoulos ¹, Hongxu Ding ¹✉ & Joshua Stuart ¹✉

Deep learning architectures such as variational autoencoders have revolutionized the analysis of transcriptomics data. However, the latent space of these variational autoencoders offers little to no interpretability. To provide further biological insights, we introduce a novel sparse Variational Autoencoder architecture, VEGA (VAE Enhanced by Gene Annotations), whose decoder wiring mirrors user-provided gene modules, providing direct interpretability to the latent variables. We demonstrate the performance of VEGA in diverse biological contexts using pathways, gene regulatory networks and cell type identities as the gene modules that define its latent space. VEGA successfully recapitulates the mechanism of cellular-specific response to treatments, the status of master regulators as well as jointly revealing the cell type and cellular state identity in developing cells. We envision the approach could serve as an explanatory biological model for development and drug treatment experiments.

¹Department of Biomolecular Engineering and Genomics Institute, University of California, Santa Cruz, CA, USA. ✉email: hding16@ucsc.edu; jstuart@ucsc.edu

Recent advances in single-cell RNA sequencing (scRNA-Seq) technologies have enabled the characterization of cellular states at an unprecedented scale and resolution¹. Among the many widely-used frameworks for analyzing complex transcriptomic patterns in single cells, artificial neural networks (ANNs) such as autoencoders (AEs)² have emerged as powerful tools. AEs are neural networks that transform an input dataset into a decoded representation while minimizing the information loss³. The diversity in their architectural design makes AEs suitable to tackle various important challenges of scRNA-Seq analysis, such as dimensionality reduction⁴, clustering⁵, and data denoising⁶.

More recently, deep generative models such as variational autoencoders⁷ (VAEs) have proven to be extremely useful for the probabilistic modeling of single-cell transcriptomes, such as scVI and scGen^{8–10}. While standard AEs learn to reconstruct an input dataset, deep generative architectures explicitly model and learn the true data distribution, which allows a broader set of queries to be addressed. While deep generative models have shown impressive performance for their dedicated modeling tasks, they often lack interpretability thus cannot offer a biologically meaningful latent representation of transcriptomes. For example, latent perturbation vectors extracted with scGen cannot be directly related to gene module variations¹⁰.

Integration of prior knowledge about gene modules to aid interpretability has already been successfully applied to transcriptomics data. DCell¹¹ is a deep neural network integrating the hierarchical information about the molecular subsystems involved in cellular processes to guide supervised learning tasks, such as predicting growth in yeast. Such a model yields an informative biological interpretation of predictions by investigating the activation of the different subsystems embedded in the model's architecture. However, this model only works in a supervised learning setting where the goal is to predict a phenotypic outcome. On the other hand, f-scLVM¹² is a Bayesian hierarchical model with explicit prior biological knowledge specification to infer the activity of latent factors as a priori characterized gene modules. While this approach enables the modeling of single-cell transcriptomes in an interpretable manner, the computational cost of the inference algorithm, as well as the absence of inference for out-of-sample data, make the development of more efficient approaches highly desirable.

Here we propose VEGA (VAE enhanced by gene annotations), a VAE with a sparse linear decoder informed by biological networks. VEGA offers an interpretable latent space to represent various biological information, e.g., the status of biological pathways or the activity of transcriptional regulators. Specifically, the scope of VEGA is twofold, (1) encoding data over an interpretable latent space and (2) inferring gene module activities for out-of-sample data.

Results

Architectural design of VEGA. To create a readily interpretable VAE, we propose a novel architecture we refer to as VEGA (VAE enhanced by gene annotations) where the decoder (generative part) connections of the neural network are guided by gene module membership as recorded in gene annotation databases (e.g., Gene Ontology, PANTHER, MolSigDB, or Reactome) (Fig. 1a). In many standard VAE implementations, the information bottleneck of the encoder-decoder architecture often represents latent variables modeled as a multivariate normal distribution. Despite providing highly informative representations of the input data, VAE latent variables are in general hard to interpret. Svensson et al.¹³ proposed using a linear decoder which directly connects latent variables to genes, providing

interpretability similar to that offered by standard factor models such as PCA. Although providing valuable insights, such an approach requires further statistical enrichment tests on the weights of the decoder to infer biological processes contributing to the single-cell expression dataset.

In contrast to previous approaches, VEGA implements a sparse architecture that explicitly reflects knowledge about gene regulation. In the service of biological pathways, genes work together in gene modules, regulated by common transcription factors that often produce correlated expression. Thus, if a given scRNA-Seq dataset X reflects the patterns of known gene modules, then it is possible for a VAE to learn a compact representation of the data by incorporating those modules as latent variables Z . VAEs use multiple layers to approximate the latent variable distribution and produce a low dimensional, nonlinear representation of the original feature data. Importantly, the first and last layers directly connect to the input or predicted features and so can be fashioned to depict intuitive groupings. Standard VAEs use a fully connected layer for both the encoding first layer and the decoding final layer (SFig. 1aiv). Instead, VEGA uses a gene membership mask M to select a subset of trainable weights in the decoder layer that are determined by a given set of gene modules (see Methods). The mask is applied to the weights that connect to the predicted output features to yield an interpretation of the latent variable layer where each latent variable is viewed as a specific gene module, henceforth referred to as a gene module variable (GMV). Specifically, the generative part of VEGA (decoder) maintains a link from a GMV to an output gene only if this gene is annotated to be a member of this specific gene module. The two main advantages of this design are (1) the latent variables are directly interpretable as the activity of biological modules and (2) the flexibility in the gene module specification allows it to generalize to different biological abstractions (such as pathways, gene regulatory networks (GRNs), or even cell types) and can be taken from any of several curated databases of gene sets (such as MSigDB¹⁴, Reactome pathways¹⁵, inferred GRNs¹⁶). Additionally, VEGA incorporates information about covariates such as technical replicates in its latent space. This can be used to alleviate batch effects, as it has been demonstrated in previous deep generative models for single-cell data⁹ (Fig. 1a and SFig. 2)

Note that it is possible to implement gene module sparseness in the encoder half of the neural network (inference part), in addition to (or in place of) the decoder half (generative part), which gives three possible VAE architectures that we considered for single-cell RNA-seq analysis (SFig. 1ai–iii). As expected, we found that the GMV-guided designs resulted in decent although slightly worse performance compared to the full architecture (SFig. 1c). Among these options, we chose the sparse decoding architecture over the others for its improved separation of known cellular states and types in the Kang et al. PBMC data¹⁷ (SFig. 1b). Intuitively, using a deep encoder maintains a full VAE's inference capacity to capture a potentially complex latent space while together with a sparse decoder approximates the posterior distribution of GMV activities $p(Z|X)$ to provide interpretation over gene modules. Additionally, we found that VEGA benefits from having a trainable, sparse decoder to adequately capture the biological signal of a dataset compared to simpler pathway transformations (SFig. 3).

Recapitulating biological information over an interpretable latent space. We asked if VEGA could recapitulate the status of biological pathways by applying it to a published and well-studied peripheral blood mononuclear cells (PBMCs) dataset stimulated with the chemokine interferon- β ¹⁷ (Methods). We first found that VEGA is able to capture cell types and stimulation status using the Reactome collection of processes and pathways¹⁵ in the GMV decoding layer

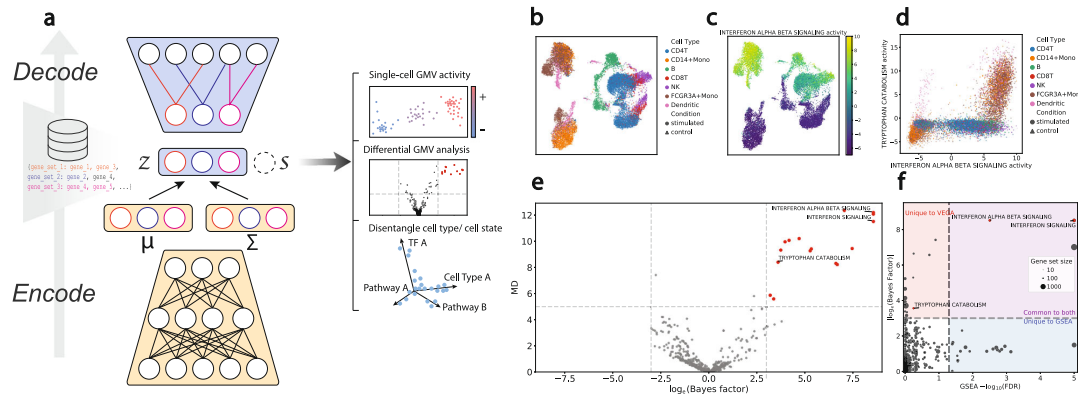


Fig. 1 Designing a novel VAE architecture with interpretable latent space. **a** Overview of the VEGA model. Composed of a deep nonlinear encoder (μ , Σ) and a masked linear decoder, VEGA represents single-cell transcriptomics data into a lower-dimensional interpretable latent space z that approximates a set of user-supplied gene modules (GMV). Additionally, VEGA can integrate batch information as another variable s to condition its generative process on batch labels. **b** UMAP embedding of the latent space of VEGA retains the biological signal of the Kang et al. PBMCs dataset¹⁷. **c** Inferred interferon- α /beta signaling pathway activity segregates stimulated cells from the control population. **d** Bivariate GMV plot showing the ability of the model to recover the tryptophan catabolism activity, an innate (Dendritic cells, FCGR3A+monocytes, CD14+monocytes) immune cell-specific response to the perturbation. **e** Volcano plot showing differentially active GMVs between stimulated and control innate immune cells. The red dots indicate GMVs with $|\log_e(\text{Bayes Factor})| > 3$ and a mean absolute difference (MD) in the latent space of at least 5. **f** Comparison of VEGA Bayes Factor with GSEA $-\log_{10}(\text{FDR})$. The size of the dots indicates the gene set size. The red, blue, and purple quadrants correspond respectively to significant hits unique to our model, unique to GSEA, and common to both.

(Fig. 1b). Specifically, we found that the interferon- α/β signaling GMV activity segregates stimulated and naive cells, confirming the ability of VEGA to capture pathway activity in its latent space (Fig. 1c, d). We further examined other known biological pathways involved in interferon-induced immune cell activation and found cell-type-specific activation of certain cellular processes. For example, tryptophan catabolism response to interferon separates innate immune cells (Dendritic cells, FCGR3A+monocytes, and CD14+monocytes) from adaptive immune cells (NK cells, T-cell CD8, T-cell CD4, and B cells) (Fig. 1d), as previously investigated^{18,19}. Together, these results suggest that VEGA's GMVs reflect the expected major biological pathways in PBMCs and therefore may be useful for other datasets to project cells into an interpretable space, allowing investigation of cell-type-specific patterns at the cellular process level.

We next asked whether the differential activities of the GMVs accurately contrast pathway states as a function of a specific, experimentally controlled context.

For this purpose, we propose a similar Bayesian hypothesis testing procedure as introduced by Lopez et al.⁹ to study the difference in GMV activities. As VEGA models the posterior distribution of each GMV, we can formulate mutually exclusive hypotheses similar to differential gene expression tests (i.e., GMVs are activated at different levels). We can approximate the posterior probability of these hypotheses through Monte Carlo sampling of VEGA's latent variable distribution. The ratio of hypothesis probabilities corresponds to the Bayes Factor²⁰ (BF, see Methods).

When applied to innate immune cells in the stimulated vs control groups of the Kang et al.¹⁷ dataset, the BF analysis found GMVs that correspond to pathways expected to be activated in the stimulated groups (interferon signaling, tryptophan catabolism; $|\log_e(\text{BF})| > 3$, Fig. 1e). We compared the GMV BFs with the false discovery rate (FDR) values of the standard GSEA toolkit (Methods, Fig. 1f). While both methods found the expected activation of the interferon- α/β signaling pathway GMV in the stimulated groups, GSEA missed the tryptophan catabolism

activation in innate immune cells (Fig. 1f). Overall, VEGA seems more robust than GSEA to gene set size bias (Fig. 1f and SFig. 4), suggesting it may emphasize more context-relevant pathways. Additionally, the differential GMV activity test can be applied in a cell-type-specific fashion (similar to one-vs-rest differential gene expression analyses). We found that such a procedure yields informative results in terms of cell type-specific biological processes activated independently of perturbation status (SFig. 5 and Supplementary Data 1).

Large-scale investigation of biological responses to drug treatments in cell lines.

Next, we investigated whether VEGA could detect patterns of drug responses in large-scale experiments over cancer cell lines, such as the data introduced in recent experimental protocols like MIX-Seq²¹. To this end, we gathered single-cell data for 97 cancer cell lines under five different conditions: 24 h DMSO treatment (control), 24 h Trametinib treatment (MEK inhibitor), 24 h Dabrafenib treatment (Mutated BRAF inhibitor), 24 h Navitoclax treatment (Bcl-2 inhibitor), and 24 h BRD3379 treatment (tool compound with unknown mode of action, MoA) (Methods). We trained one model for each different drug treatment (four models in total) by combining the drug treatment dataset and the control group (DMSO dataset), initializing the GMVs of VEGA with the hallmark gene sets from MSigDB²² to focus on core cellular processes. Overall, each model was able to separate cell lines and treatment conditions in the GMV space (Fig. 2a, and SFig. 6). For Trametinib notably, the important change in G2M checkpoint GMV activity (decrease in the treated condition) agrees with the expected MoA of a MEK inhibitor^{23,24} (Fig. 2b). Next, we sought to investigate whether we could recapitulate the pattern of biological responses between control and treated conditions for each cell line/drug treatment pair. For each pair, we computed GMV BFs to approximate differential pathway activities between the two conditions. The resulting heatmap can be used to understand and interpret patterns of response over all experimental conditions (Fig. 2c). As found when visually investigating the low dimensional

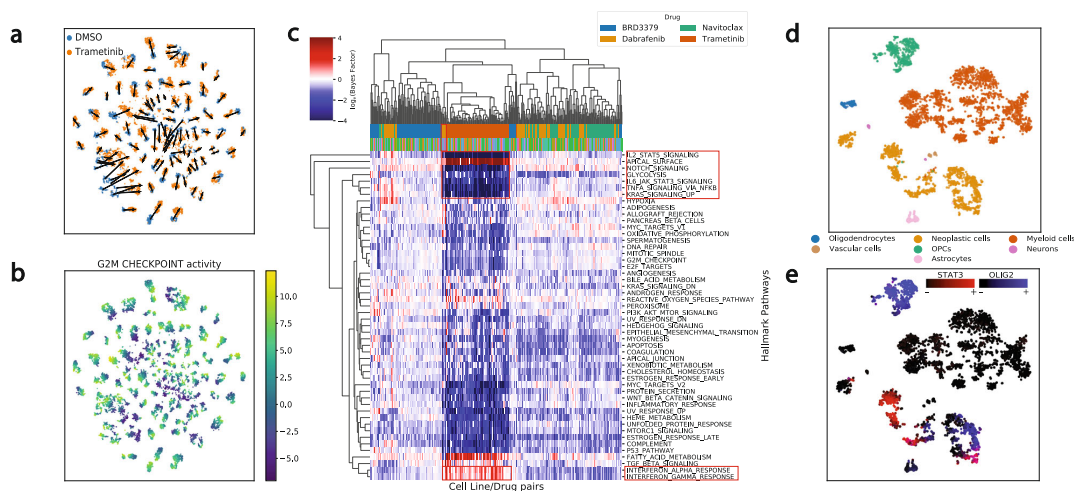


Fig. 2 The flexibility in the latent space specification sheds light on the activity of core cellular processes and transcription factors. **a** tSNE embedding of the latent space of VEGA for the MIX-Seq data²¹. The color indicates the treatment condition, and the arrow indicates the median shift in coordinates of each cell line between the two conditions. **b** Inferred G2M checkpoint activity of each cells, showing a decreased activity in the treated condition, as expected from the MoA of Trametinib. **c** Heatmap with hierarchical clustering showing the average $\log_e(\text{Bayes Factor})$ of each pathway for each cell line/drug treatment pair (test between DMSO and treatment condition). Each row corresponds to a hallmark gene set and each column to a different cell line/drug pair. The first row of color indicates the drug, and the second row of color indicates the tissue identity (Tissue legend available in SFig. 2). Highlighted cell lines correspond to BRAF-mutant melanoma. Highlighted activities correspond to Trametinib-specific responses. **d** tSNE embedding of the latent space of the model for the glioblastoma dataset²⁹, colored by cell type or **e** Inferred activity of the master regulators STAT3 and OLIG2.

embedding of each dataset (Fig. 2a and SFig. 6a–c), Trametinib resulted in the strongest transcriptional response of all studied drugs. Notably, the Trametinib-specific interferon- α and interferon- γ response was correctly recapitulated in VEGA’s latent space, consistent with previous experimental work²⁵ and the findings reported by the original MIX-Seq authors²¹. Furthermore, we found that Dabrafenib-treated BRAF-mutant melanoma cell lines exhibited larger $[\log_e(\text{BF})]$ than other Dabrafenib-treated cell lines (average $[\log_e(\text{BF})]$ of 0.763 vs 0.668 for other cell lines), clustering with the Trametinib-treated cell lines as reported in the MIX-Seq study (Fig. 2c and SFig. 6d). Overall, the results presented here agree with the previous gene set analysis results on this dataset, and demonstrate VEGA’s GMVs can recapitulate patterns of drug response in large-scale experiments.

Gene regulatory analysis of glioblastoma reveals stratification of neoplastic cells. As previously mentioned, one of VEGA’s strengths is the flexibility in the specification of the GMV connectivity, as any gene module can be used in the decoder. Transcription factors often exert tight regulation of gene expression in many biological contexts²⁶. Analyzing the activity of transcriptional regulators is important in understanding biological states like cell types or diseases, as dysregulation in their activity can have a dramatic impact on gene expression programs and phenotypes^{27,28}. To this end, we investigated whether using master transcriptional regulators as the GMVs could help understand the underlying GRNs in the context of a single-cell glioblastoma (GBM) dataset²⁹. We used the GBM ARACNe¹⁶ network reported in Carro et al.²⁸ to guide the structural design of our model. Specifically, VEGA’s GMVs were set to the reported transcription factors and the connectivity matrix \mathbf{M} , defining the GMVs decoding architecture, was created from the set of predicted target genes of each transcription factor. After training, we found that the pre-annotated cell types were well-separated in the latent space (Fig. 2d). We examined the activity of STAT3 and

OLIG2, two well-known master regulators of the mesenchymal (MES) and proneural (PN) GBM subtypes, respectively. We confirmed that their GMV activity was largely anticorrelated in neoplastic cells (Fig. 2e). Additionally, OLIG2, a known master regulator of oligodendrocyte differentiation³⁰, was inferred as active in oligodendrocyte precursor cells (OPCs). These results demonstrate that VEGA is able to home-in on the relevant transcriptional regulators when the decoder wiring is extended to model known factor-to-target relationships.

Combining cell type and cellular state representations refines cortical organoid development analysis. A great challenge of modern cellular biology is to identify and define cell types and cellular states, at the level of individual cells, in order to systematically study homeostasis and disease development under a common vocabulary. In a typical single-cell study, a few “marker sets” will be known, each containing a list of genes having expected expression patterns for some of the cell types of interest. Leveraging such marker sets often provides clues and helps orient data analysis. We asked whether the information recorded in such marker sets could be used in VEGA to produce a disentangled representation of cell types and cellular states. To this end, we added a GMV z_t , with appropriate entries in \mathbf{M} , for each latent cell type t in addition to the Reactome pathway GMVs already in VEGA’s model.

We applied VEGA to a dataset of cells assayed during the early development of cortical organoids from Field et al.³¹, including all of the major cell types defined in the study as GMVs (Fig. 3a). After training, we found that the activity of each marker set GMV was able to correctly segregate its corresponding cell type as annotated by the original authors (Fig. 3b–d). Moreover, in a one-vs-rest differential GMV analysis setting for each cell type population, the activity of the corresponding marker set GMV showed significant enrichment ($[\log_e(\text{BF})] > 3$), which suggests using GMV BFs could help annotate the cell types of unknown clusters (Fig. 3e). We further noted that the

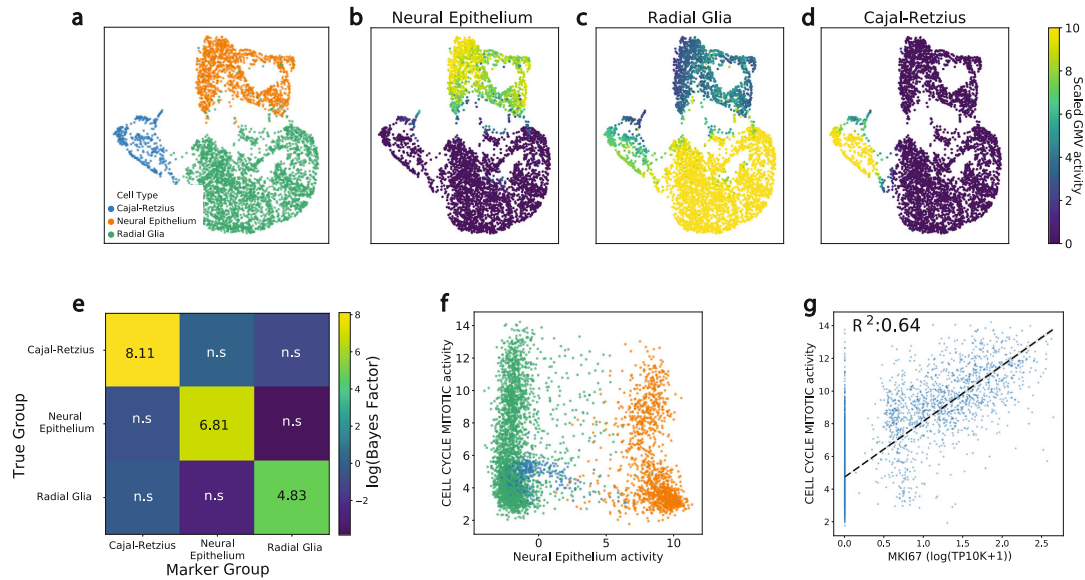


Fig. 3 Disentangling cellular states and cell types in the early development of cortical organoids. **a** UMAP embedding of the latent space of our model for the week 2 cortical organoid dataset³¹. The cell type annotation corresponds to the original paper annotation. **b, c, d** The inferred activity of each cell type GMVs (as defined by marker genes) correctly identifies the three main subpopulations of cells. **e** One-vs-rest differential GMV analysis of each cell type population provides a statistical significance for each cell type signature. The significance threshold for positive enrichment was set to $\log_e(\text{BF}) > 3$. **f** Identification of dividing and quiescent subpopulations of neural progenitors using pathway and cell-type activity projection. **g** CELL_CYCLE_MITOTIC pathway activity correctly identifies dividing cells as reported by its correlation with MKI67 gene expression (an external canonical marker of dividing cells).

most differentially activated GMVs were coherent in the context of early brain development (SFig. 7 and Supplementary Data 2). To study whether VEGA could separate cell type identity from cellular states such as dividing vs quiescent cell populations, we projected the dataset into two components: (1) the cell type GMV representing the neural epithelium marker set (a type of early brain progenitor) and (2) the cell state GMV representing the cell cycle mitotic pathway activity (Fig. 3f). As discussed previously, the activity of the neural epithelium GMV separated the neural epithelium cells from the rest of the dataset, while the activity of the cell cycle mitotic pathway GMV separated quiescent from actively dividing cells in the two progenitors populations (radial glia cells and neural epithelium). To validate that the cells identified as dividing were proliferating, we studied the correlation between the cell cycle mitotic pathway GMV activity and the expression of the MKI67 gene, a canonical marker of proliferation (external validator not present in the cell cycle mitotic pathway set) (Fig. 3g). Overall, the expression of MKI67 correlates well with the inferred activity of the cell cycle mitotic pathway GMV ($R^2 = 0.64$). Together, these results demonstrate VEGA's potential use to jointly infer cell type and state for different populations of cells, as combining different sources of information (pathways, master regulators, and cell type markers) in the latent space can shed light on different aspects of the identity of a single-cell.

Generalization of the inference process to out-of-sample data.

We next asked whether VEGA could generalize to correctly infer an interpretable latent representation of data unseen at the time of training (out-of-sample data). To this end, we evaluated VEGA in two settings. In the first case, we measured the biological generalization of VEGA's inference by holding out (cell type, condition) pairs during training. Specifically, we investigated whether the inferred GMV activities for held-out cells were conveying the same biological information as to when this

population is seen at the time of training. To this end, we removed one cell type of the stimulated condition during training, and then inferred the GMV activities for that held-out population (out-of-sample) and compared them to the GMV activities learned from the fully trained model. The experiment was conducted using the Kang et al.¹⁷ PBMC dataset. In the second case, we estimated the “technical generalization” of VEGA's inference by training on one dataset (study A) and then evaluating on a second dataset (study B) that contains only control cells. We used the Kang et al.¹⁷ PBMC dataset as study A and the Zheng et al.³² dataset as study B.

For the biological generalization test, we first checked that the distribution of the interferon- α/β signaling pathway GMV activity in the out-of-sample stimulated CD4 T cells matched the inferred activity in the in-sample CD4 T cells (Fig. 4a). To perform a more systematic comparison of the inferred latent space between out-of-sample and in-sample cells, we used the differential BF procedure (Methods) between (1) stimulated in-sample cells and control cells for a given cell type (model trained with the whole dataset) and (2) stimulated out-of-sample cells and control cells for the same cell type (model trained with one cell type/condition pair left out), and checked the amount of overlaps in the top 50 differentially activated GMVs (Fig. 4b). The results suggested consistency between the in-sample and out-of-sample differentially activated GMVs, with an average 72% overlap. To further evaluate the capacity of data reconstruction, we measured the R^2 between the original and decoded data in the in-sample and out-of-sample settings (Fig. 4c). We found that the R^2 decreases only marginally in the out-of-sample setting, confirming the ability of the model to generalize to unseen data produced in a similar experimental setting.

For the technical generalization test, we again checked that the interferon- α/β signaling pathway GMV activity distribution of study B encoded control CD4 T cells matched that of study A

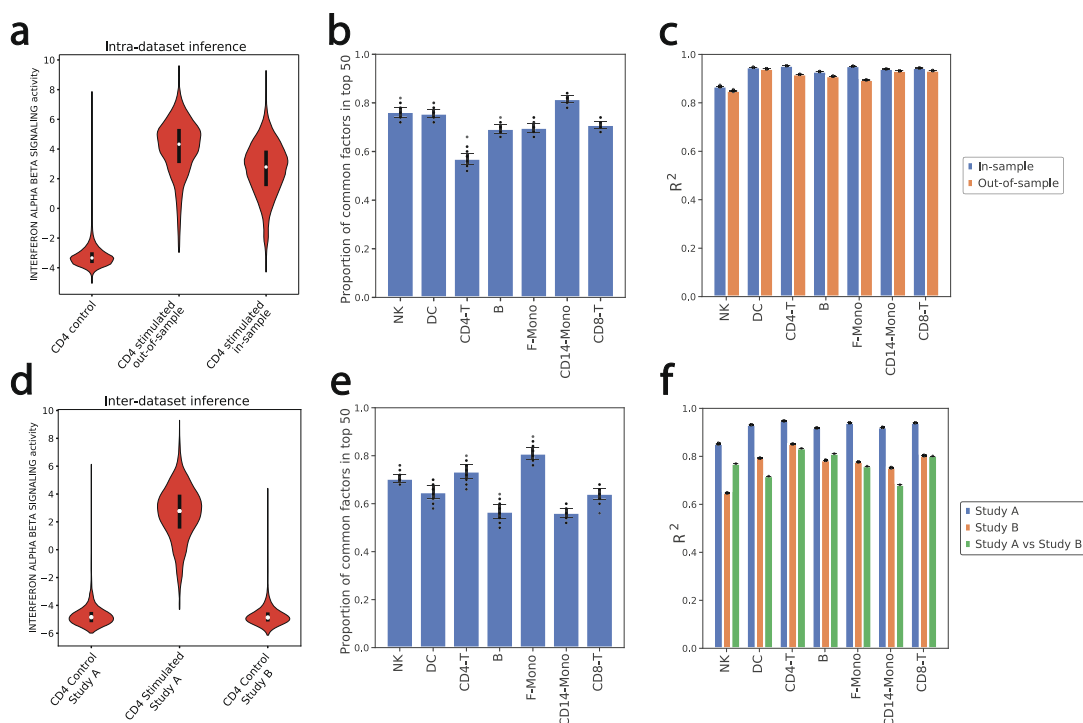


Fig. 4 Generalization of VEGA architecture to out-of-sample data. **a** Violin plot ($n=10,000$ randomly sampled cells per condition) representing the distribution of the interferon- α/β pathway activity in control CD4-T cells, stimulated CD4 T cells unseen at the time of training (out-of-sample), and stimulated CD4-T cells when included in the training procedure (in-sample). Boxes inside the violins represent the median of the distribution bounded by the first and third quartile. Violin limits correspond to data extrema. **b** Proportion of overlap in the top 50 differentially activated GMVs in the in-sample and out-of-sample settings with stimulated vs control differential procedures for the seven main cell types in the study. Data were presented as mean values \pm standard deviation over 100 random samplings. **c** R^2 between the mean expression of real and reconstructed cells in the in-sample and out-of-sample settings for the seven main cell types of the study. Data were presented as mean values \pm standard deviation over 100 random samplings. **d** Violin plot ($n=2000$ randomly sampled cells per condition) of distribution of the interferon- α/β pathway activity in control CD4-T cells of study A (Kang et al.¹⁷), stimulated CD4-T cells of study A and control CD4-T cells of study B (Zheng et al.³²). Boxes inside the violins represent the median of the distribution bounded by the first and third quartile. Violin limits correspond to data extrema. **e** Proportion of overlap in the top 50 differentially activated GMVs of each cell type with one-vs-rest differential procedures for the control cells of the seven main PBMC cell types. Data were presented as mean values \pm standard deviation over 100 random samplings. **f** R^2 between the mean expression of real and reconstructed cells of study A (Study A), mean expression of real and reconstructed cells of study B (Study B), and mean expression of real cells of study A and real cells of study B (Study A vs Study B). Data were presented as mean values \pm standard deviation over 100 random samplings.

control CD4 T cells (Fig. 4d). We also investigated whether the top 50 differential GMVs of each cell type in a “one-vs-rest” differential setting for the control cells of study A overlapped with a similar procedure performed on the control cells of study B (Fig. 4e). We found that on average 67% of the top 50 differential GMVs for study A overlap with those of study B, showing that the model can generalize across studies unseen at the time of training. We then asked whether the model can use the inferred latent space to accurately reconstruct the original expression profiles of both studies. We found that the R^2 between original and reconstructed cells of study B, although lower than those for study A, improves upon the baseline correlation between the expression profiles of study A vs study B for most of the cell types (Fig. 4f).

Discussion

In this study, we introduced VEGA, a novel VAE architecture with a decoder inspired by known biology to infer the activity of various gene modules at the level of individual cells. By encoding single-cell

transcriptomics data into an interpretable latent space specified a priori, our method provides a fast and efficient way of analyzing the activity of various biological abstractions in different contexts. In contrast, previous approaches used a posteriori interpretations of the latent variables to infer modules. VEGA’s flexibility in the specification of the latent space paves the way for analyzing the activity of biological modules such as pathways, transcriptional regulators, and cell type-specific modules. We illustrated how VEGA could be used to simultaneously investigate both cell type and cell state of cell subpopulations, in both control and experimentally perturbed conditions. Additionally, the weights of decoder connections provide direct interpretability of the relationship between the latent variables and the original features. For example, the decoder’s weights could be used to contrast interaction confidence in inferred GRNs or to rank genes by their importance in a certain biological module in a data-driven way. We further note that it was possible to modify VEGA’s architecture, following the same rationale as widely-used scVI⁹ and linear scVI¹³, such that it could handle count data in place of normalized expression profiles (SFig. 8).

The clear limitations of the current architecture resides in the sparse, single-layer decoder of the model. In fact, such an architectural design prevents the further improvement of generalizability and robustness. As a consequence, the generative capacity of VEGA is limited. For example, while VEGA theoretically could be used for interpretable response prediction using latent vector arithmetics in a similar fashion to scGen¹⁰, VEGA's limited generative capacity sacrifices predictive performance for biological interpretability of the latent space. We believe advanced insights in network biology, e.g., multi-layer GRNs that can describe regulatory machinery more comprehensively, could alleviate these limitations. This would open the possibility to perform targeted, in-silico activation, and repression of biological programs on specific cell populations to study its effect on development or disease progression. On the other hand, hard-coded connections of the linear decoder do not leave any room for correcting prior knowledge about gene modules when the context requires it, as is the case in other latent variable models such as f-scLVM¹². In fact, prior biological knowledge obtained from existing databases like MSigDB can be incomplete or not context-specific, as additional unannotated genes can play an important role in certain gene modules. In parallel to our work on VEGA, Rybakov et al.³³ introduced a regularization procedure to incorporate prior knowledge from gene annotation databases via a penalty term on the weights of the linear decoder. We demonstrated that VEGA performs comparatively to their interpretable autoencoder (SFig. 9), and that their approach is complementary to the unique attributes of VEGA and can be used to recover missing gene-GMV links in a data-driven fashion (SFig. 10).

In summary, we found VEGA useful for understanding the response of specific cell type populations to different perturbations, providing interpretable insights on biological module activity. The variational aspect of VEGA provides an advantage for addressing queries about samples, or sample groups, that are not possible with a regular AE. We illustrated how the latent multivariate Gaussian distribution of the VAE, which approximates the posterior probability of every GMV, enables a new kind of differential test to be performed. The BF reflects the likelihood of how active a gene module is in one condition compared to another, providing a straightforward method to perform differential activity analysis using the RNA-Seq data similar to the approach described by Lopez et al.⁹. Other types of queries are possible, for example, to automate the annotation of unsupervised clusters or modules that dynamically change across the branches of an inferred cellular trajectory. We envision VEGA could also be useful to prioritize drugs based on pathway expression in cancer, as studying the response of specific cell populations may inform drug sensitivity and resistance. Integrating drug response prediction models with such explanatory models could benefit designing novel therapeutic strategies.

Methods

The VEGA architecture. VEGA is a deep generative VAE that aims at maximizing the likelihood of a single-cell dataset X under a generative process^{7,10} described as:

$$p(X|\theta) = \int p(X|Z, \theta)p(Z|\theta)dZ, \quad (1)$$

with θ being the learnable parameters of a neural network. VEGA uses a set of latent variables Z that explicitly represent sets of genes (gene modules), such as pathways, GRNs, or cell type marker sets. To enforce the VAE to interpret a dataset from the viewpoint of a set of gene modules, VEGA's decoder part is made up of a single, masked, linear layer. Specifically, the connection of this layer, between latent node $z^{(j)}$ and gene features, are specified using a binary mask M in which M_{ij} is true if gene i is a member of gene module j and false otherwise. We refer to each latent variable $z^{(j)}$ as a GMV since each provides a view of the data constrained to the subset of genes for a distinct gene module j . During training, gradients associated with masked (false) weights are "zeroed out" such that backpropagation only applies to weights originating from a user-supplied given gene set. Additionally, the weights of the decoder are constrained

to be nonnegative ($w \geq 0$) to maintain interpretability as to the directionality of gene module activity.

Having explicitly specified the connections between genes and latent variables in the decoder of VEGA (generative part), we incentivize that the latent space represents a biological module activity interpretation of the data. We choose to model the GMVs as a multivariate normal distribution, parametrized by our inference network with learnable parameters ϕ . As such, the distribution of the Z latent variables can be expressed as:

$$q(Z|X, \phi) = \mathcal{N}(\mu_\phi(X), \Sigma_\phi(X)) \quad (2)$$

This choice of variational distribution is common and has proven to work well in previous single-cell studies^{9,10}. Following similar standard VAE implementations^{7,10}, the objective to be maximized during training is the evidence of lower bound (ELBO):

$$\mathcal{L}(X) = \mathbb{E}_{q(Z|X, \phi)} [\log p(X|Z, \theta)] - KL(q(Z|X, \phi) || p(Z|\theta)) \quad (3)$$

where the expectation over the variational distribution can be approximated using Monte Carlo integration over a minibatch of data, and the Kullback-Leibler divergence term has a closed-form solution as we set the prior to:

$$p(Z|\theta) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (4)$$

The reparametrization trick⁷ is used when sampling VEGA's variational distribution to allow standard backpropagation to be applied when training the model.

To retain information of genes that are not present in our pre-annotated biological networks, we add additional fully connected nodes to the latent space of our model. This has two effects: (1) it allows VEGA to model the expression of unannotated genes, which could be crucial for a good reconstruction of the data during training, and (2) it can help capture additional variance of the data that is unexplained by the provided gene modules, considerably improving the training of the model. The number of additional fully connected nodes can be determined based on a trade-off between model performances and the loss of information encoded by pre-annotated GMV nodes. As a rule of thumb, we recommend picking 16 or fewer extra FC nodes to preserve the biological signals encoded by GMV nodes (SFig. 11).

Additionally, the diagonal covariance prior used in the latent space modeling discourages GMVs from being correlated. Thus, the VAE may be forced to choose an arbitrary gene set among many equally informative but overlapping sets and could fail to reveal a key annotation. To address this issue, we add a dropout layer to the latent space of the model. This has been shown to force the VAE to preserve redundancy between latent variables³⁴, which is applicable when the gene annotation database used to initialize VEGA's latent space contains overlapping gene sets (SFig. 12).

Finally, batch information or other categorical covariates can be encoded via extra nodes in the latent space, conditioning the generative process of VEGA on this additional covariate information (SFig. 2).

Measuring differential GMVs activity of the latent space with Bayes Factor (BF)

The difference in the activity of genes and/or pathways is often of interest when contrasting two different groups of cells. To this end, we draw inspiration from the Bayesian differential gene expression procedure introduced in Lopez et al.⁹ and propose a similar differential GMV analysis procedure. We follow a similar notation as Lopez et al. For a given GMV k , a pair of cells (x_a, x_b) and their respective group ID (s_a, s_b) (e.g., two different treatment conditions), our two mutually exclusive hypotheses are:

$$\mathcal{H}_0^k := \mathbb{E}_s [z_a^k] > \mathbb{E}_s [z_b^k] \text{ vs. } \mathcal{H}_1^k := \mathbb{E}_s [z_a^k] \leq \mathbb{E}_s [z_b^k] \quad (5)$$

This can intuitively be seen as testing whether a cell has a higher mean GMV activation than another, the expectation representing empirical frequency. We evaluate the most probable hypothesis by studying the log-Bayes factor K defined as:

$$K = \log_e \frac{p(\mathcal{H}_0^k | x_a, x_b)}{p(\mathcal{H}_1^k | x_a, x_b)} \quad (6)$$

Here, the sign of K tells us which hypothesis is more likely, and the magnitude of K encodes a significance level. Having access to the conditional posterior distribution $q(Z|X)$ over the GMVs activation (the encoding part of VEGA), we can approximate each hypothesis' probability distribution as:

$$p(\mathcal{H}_0^k | x_a, x_b) \approx \sum_s p(s) \int \int_{\text{sup}(z_a), \text{sup}(z_b)} p(z_a^k > z_b^k) dq(z_a^k) dq(z_b^k | x_b) \quad (7)$$

where $p(s)$ is the relative abundance of cells in group s , and the integrals are approximated with direct Monte Carlo sampling.

Similarly to Lopez et al.⁹, assuming cells are independent, we can compute the average Bayes factor across many cell pairs randomly sampled from each group respectively. This helps us decide whether a GMV is activated at a higher frequency in one group or the other. Through the paper, we consider GMVs to be significantly differentially activated if the absolute value of K is greater than 3 (equivalent to an odds ratio of ≈ 20)^{9,20}.

Datasets and preprocessing

Kang et al. dataset. The Kang et al.¹⁷ dataset consisted of two groups of PBMCs, one control and one stimulated with interferon- β . We chose to use the same preprocessing steps as described by scGen authors¹⁰, using the Scanpy package³⁵. Briefly, cells were annotated using the maximum correlation to one of the eight original cell type clusters identified, using an average of the top 20 cluster genes. Megakaryocytes were removed due to uncertainty about their annotation. Then data were filtered to remove cells with less than 500 genes expressed and genes expressed in five or less cells, using the `scanpy.pp.filter_genes()` and `scanpy.pp.filter_cells()` functions. Count per cells were then normalized and log-transformed using the `scanpy.pp.normalize_per_cell()` and `scanpy.pp.log1p()` functions, and we selected the top 6998 highly variable genes with `scanpy.pp.highly_variable_genes()`, resulting in a final dataset of 18,868 cells. Raw data is available at [GSE96583](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE96583). We used the same preprocessing functions for the rest of the datasets unless specified otherwise.

Zheng et al. dataset. The Zheng et al.³² dataset consists of 3K PBMCs from a healthy donor. After filtering the cells, the count per cells were normalized and log-transformed. We then subset the genes to use the same 6998 genes of the Kang et al. PBMC dataset. The final dataset has 2623 cells and 6998 genes. Raw data are available at <https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k>.

MIX-seq dataset. The MIX-seq²¹ datasets were obtained from <https://figshare.com/s/139f64b495dea9d88c70>, and we used the data from experiment 3 to have enough cells to carry a smooth training of our model. For the five available datasets (97 cell lines treated with respectively DMSO, Trametinib, Dabrafenib, Navitoclax, and BRD3379), we removed cells with 200 or less expressed genes, and genes expressed in less than three cells. We then normalized the number of counts per cell, and log-transformed the data. Finally, each dataset that was a drug treatment experiment was combined with a copy of the control dataset (DMSO treatment), and we extracted the top 5000 highly variable genes. This resulted in final datasets of size (16,732 cells and 4999 genes) for the Trametinib+DMSO data, (16,942 cells and 5000 genes) for the Dabrafenib+DMSO data, (14,507 cells and 5000 genes) for the Navitoclax+DMSO data, and (15,304 cells and 5000 genes) for the BRD3379+DMSO data.

Darmanis et al. dataset. The raw GBM data from Darmanis et al.²⁹ were obtained from <http://www.gbmseq.org/> and preprocessed as followed: we removed cells with 200 or less expressed genes, and genes expressed in three or less cells. Count per cells were normalized and data were then log-transformed. Finally, we restricted the transcriptome to the top 6999 highly variable genes. The final dataset had a total of 3566 cells. Raw data is available at [GSE84465](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE84465).

Field et al. dataset. The cortical organoid data from Field et al.³¹ was processed similarly to the GBM dataset. After normalization and highly variable genes selection, the dataset had a total of 4378 cells, with 6999 genes. Raw data is available at [GSE106245](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE106245).

Shekhar et al. dataset. The mouse retina dataset from Shekhar et al.³⁶ was processed as described (see <https://github.com/broadinstitute/BipolarCell2016>). Briefly, we removed cells with more than 10% mitochondrial transcripts. Then, cells with less than 500 genes were removed, and genes expressed in less than 30 cells and with less than 60 transcripts across all cells were removed. To be able to use human versions of gene modules from the Reactome database, we performed one-to-one ortholog mapping of mouse transcripts to human transcripts using BioMart from the Ensembl project³⁷. Genes without human orthologs were removed. We saved a version of the dataset with the raw count data for the selected genes/cells, and further processed the data by normalizing and log-transforming the libraries. Finally, we restricted the transcriptome to the top 4000 highly variable genes. The same highly variable genes were used to subset the raw QC count matrix. The final datasets (for both count and log-normalized versions) had a total of 27,499 cells, coming from two technical batches. We used the annotation with 15 cell types from the original authors. Raw data is available at [GSE81904](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE81904).

Choice of gene annotations for the latent space of VEGA. When initializing the latent space of our model, we chose to use pre-annotated gene sets from the Molecular Signature Database (MSigDB, at <https://www.gsea-msigdb.org/gsea/msigdb/collections.jsp#C2>)¹⁴. In particular, we chose to use the hallmark gene sets annotation (50 gene sets) or the Reactome database (674 gene sets). Reactome was used for the stimulated PBMCs analysis, and MSigDB's Hallmark gene sets were used in the MIX-Seq analysis part of this study. For the gene regulatory network analysis of GBM cells, we derived an ARACNe^{16,38} network from bulk RNA-Seq samples of GBM. Specifically, this network was obtained from a previously published paper³⁹ and repurposed for the study of GBM single-cell transcriptomics profiles.

For the cell type marker genes in the cortical organoid analysis, we contacted the authors to obtain relevant genes used in annotating those cell types. The GMT file including these marker genes can be found along with the reproducibility code at <https://github.com/LucasESBS/vega-reproducibility>.

Dimensionality reduction for visualization. For visualizing datasets, we used the UMAP algorithm⁴⁰ as implemented in the Scanpy³⁵ python package, using `scanpy.pp.neighbors()` for the k-NN computation with `n_neighbors=15`, and `scanpy.tl.umap()` for the actual dimensionality reduction. We used default parameters except for the `min_dist` parameter that we set to 0.5. We also used tSNE⁴¹ implemented as `sklearn.manifold.TSNE()` in the sklearn python package⁴², with default parameters.

Comparison with GSEA. We ran Gene Set Enrichment Analysis <https://www.zotero.org/google-docs?grfpAv14> (GSEA) using the `prerank` function from the gseapy package in Python. Briefly, we calculated differential expression scores for each gene between the control and treatment group using a Wilcoxon rank-sum test, as implemented in the `scanpy.tl.rank_genes_groups()` functionality of the Scanpy package <https://www.zotero.org/google-docs?fkYtI735>. We ranked genes according to their test statistics, and ran GSEA using the gseapy package function `gseapy.prerank()` with the following settings: a minimum gene set size `min_size=5`, a maximum gene set size `max_size=1000`, and a number of permutations `permutation_num=1000`. We ranked gene sets according to their FDR and considered significant hits when `FDR ≤ 0.05`. When the FDR returned by GSEA was equal to 0, we replaced it with `1e-5` (to avoid math error when taking the logarithm).

Batch correction comparison. To assess batch information integration in VEGA's latent space, we compared the average silhouette scores on batch labels from the Shekhar et al. retina dataset of (1) PCA with 50 principal components (computed using `scanpy.tl.pca()` function), (2) linear scVI¹³ as implemented in the `scvi-tools` package ran on the count version of the dataset with following parameters: AnnData object setup with `batch_key=Batch`, model initialized with `n_hidden=800`, `n_layers=2`, `dropout_rate=0.2`, `n_latent=677`, training performed with `max_epochs=300`, `early_stopping=True`, `lr=5e-4`, `train_size=0.8`, `early_stopping_patience=20`, and (3) VEGA with following parameters: AnnData object setup with `batch_key=atc`, model initialized using the REACTOME pathway database with three extra FC nodes to initialize the latent space and the same training hyperparameters as linear scVI.

Evaluation metrics. Silhouette scores were calculated to evaluate the separation of cell types and states in the latent space of our model. We used Euclidean distance in the latent space to compute the silhouette coefficient of each cell i defined as :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (8)$$

where $a(i)$ and $b(i)$ are respectively the mean intra-cluster distance and the mean nearest-cluster distance for cell i . We used either the stimulation or cell type labels from Kang et al.¹⁷ to assess the biological relevance of the latent space of our model. The sklearn package¹⁷ `silhouette_score()` implementation was used for computation. For computing correlations throughout the paper, we used the function `numpy.corrcoef()` from the Numpy package⁴³.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

All of the datasets analyzed in this manuscript are publicly available. Please see the section Datasets and preprocessing of Methods for details. These datasets are also downloadable at <https://github.com/LucasESBS/vega-reproducibility>.

Code availability

The package and API for VEGA is available at https://github.com/LucasESBS/vega/tree/vega_dev44. The code and data to reproduce the results of this manuscript is available at <https://github.com/LucasESBS/vega-reproducibility>.

Received: 11 January 2021; Accepted: 13 September 2021;

Published online: 28 September 2021

References

- Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C. & Teichmann, S. A. The technology and biology of single-cell RNA sequencing. *Mol. Cell* **58**, 610–620 (2015).
- Hinton, G. E. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).
- Baldi, P. Autoencoders, unsupervised learning, and deep architectures. *Proc. Mach. Learn. Res.* **27**, 37–49 (2012).
- Wang, D. & Gu, J. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. *Genomics, Proteomics Bioinformatics* **16**, 320–331 (2018).

5. Geddes, T. A. et al. Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis. *BMC Bioinformatics* **20**, 660 (2019).
6. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
7. Kingma, D.P. & Welling, M. Auto-encoding variational Bayes. Preprint at *arXiv:1312.6114 [cs, stat]* (2014).
8. Way, G. P. & Greene, C. S. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pac. Symp. Biocomput.* **23**, 80–91 (2018).
9. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
10. Lotfollahi, M., Wolf, A. F. & Theis, F. J. scGen predicts single-cell perturbation responses. *Nat. Methods* **16**, 715–721 (2019).
11. Ma, J. et al. Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods* **15**, 290–298 (2018).
12. Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C. & Stegle, O. fscLVM: scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol.* **18**, 212 (2017).
13. Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics* **36**, 3418–3421 (2020).
14. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA* **102**, 15545–15550 (2005).
15. Jassal, B. et al. The reactome pathway knowledgebase. *Nucleic Acids Res.* **48**, D498–D503 (2020).
16. Margolin, A. A. et al. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7**, S7 (2006).
17. Kang, H. M. et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36**(January), 89–94 (2018). Number: 1 Publisher: Nature Publishing Group.
18. Mellor, A. L., Lemos, H. & Huang, L. Indoleamine 2,3-dioxygenase and tolerance: where are we now? *Front. Immunol.* **8**, 1360 (2017).
19. Sorgdrager, F. J. H., Naudé, P. J. W., Kema, I. P., Nollen, E. A. & De Deyn, P. P. Tryptophan metabolism in inflammaging: from biomarker to therapeutic target. *Front. Immunol.* **10**, 2565 (2019).
20. Kass, R. E. & Raftery, A. E. Bayes factors. *J. Am. Stat. Assoc.* **90**, 773–795 (1995).
21. McFarland, J. M. et al. Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nat. Commun.* **11**, 4296 (2020).
22. Liberzon, A. et al. The molecular signatures database hallmark gene set collection. *Cell Syst.* **1**, 417–425 (2015).
23. Kurata, K. et al. Growth arrest by activated BRAF and MEK inhibition in human anaplastic thyroid cancer cells. *Int. J. Oncol.* **49**, 2303–2308 (2016).
24. Joshi, M., Rice, S. J., Liu, X., Miller, B. & Belani, C. P. Trametinib with or without vemurafenib in BRAF mutated non-small cell lung cancer. *PLoS ONE* **10**, e0118210 (2015).
25. Lulli, D., Carbone, M. L. & Pastore, S. The MEK inhibitors trametinib and cobimetinib induce a type I interferon response in human keratinocytes. *Int. J. Mol. Sci.* **18**, 2227 (2017).
26. Spitz, F. & Furlong, E. E. M. Transcription factors: from enhancer binding to developmental control. *Nat. Rev. Genet.* **13**, 613–626 (2012).
27. Lee, T. I. & Young, R. A. Transcriptional regulation and its misregulation in disease. *Cell* **152**, 1237–1251 (2013).
28. Carro, M. S. et al. The transcriptional network for mesenchymal transformation of brain tumours. *Nature* **463**, 318–325 (2010).
29. Darmanis, S. et al. Single-cell RNA-seq analysis of infiltrating neoplastic cells at the migrating front of human glioblastoma. *Cell Reports* **21**, 1399–1410 (2017).
30. Lu, R. Q. et al. Common developmental requirement for olig function indicates a motor neuron/oligodendrocyte connection. *Cell* **109**, 75–86 (2002).
31. Field, A. R. et al. Structurally conserved primate lncRNAs are transiently expressed during human cortical differentiation and influence cell-type-specific genes. *Stem Cell Reports* **12**, 245–257 (2019).
32. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 1–12 (2017).
33. Rybakov, S., Lotfollahi, M., Theis, F. J. & Wolf, A. F. Learning interpretable latent autoencoder representations with annotations of feature sets. Preprint at *bioRxiv* <https://doi.org/10.1101/2020.12.02.401182> (2020).
34. Yeung, S., Kannan, A., Dauphin, Y. & Fei-Fei, L. Tackling over-pruning in variational autoencoders. Preprint at *arXiv: 1706.03643* (2017).
35. Wolf, A. F., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
36. Shekhar, K. et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell* **166**, 1308–1323.e30 (2016).
37. Howe, K. L. et al. Ensembl 2021. *Nucleic Acids Res.* **49**, D884–D891 (2021).
38. Lachmann, A., Giorgi, F. M., Lopez, G. & Califano, A. ARACNe-AP: gene network reverse engineering through adaptive partitioning inference of mutual information. *Bioinformatics* **32**, 2233–2235 (2016).
39. Ding, H. et al. Quantitative assessment of protein activity in orphan tissues and single cells using the metaVIPER algorithm. *Nat. Commun.* **9**, 1471 (2018).
40. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. Preprint at *arXiv: 1802.03426* (2020).
41. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
42. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
43. Harris, C. R. et al. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
44. Seninge, L., Anastopoulos, I., Ding, H. & Stuart, J. VEGA is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *vega*. <https://doi.org/10.5281/zenodo.5338892> (2021).

Acknowledgements

L.S. was supported by the Schmidt Futures Foundation SF 857 and by the National Institute Of Mental Health of the National Institutes of Health award R01MH120295. J.M.S. was supported by a grant 5R01GM109031 from the NIGMS. J.S. and H.D. were supported by a grant from the Chan-Zuckerberg Initiative's Human Cell Atlas portals project. H.D. was supported by a gift from Seagate Technology. J.S. was supported by grant GC1R-06673-C from the California Institute for Regenerative Medicine's Center of Excellence for Stem Cell Genomics. The authors would like to thank Dr. David Haussler and Dr. Sofie Salama for their support. L.S. would also like to thank David Parks for the useful feedback during the early development of the method. We also would like to thank Dr. Maximilian Haeussler for the feedback on the manuscript.

Author contributions

L.S. and I.A. conceived the idea. L.S. implemented the method and gathered the data. L.S. and I.A. performed the analysis. H.D. and J.S. supervised the research. All authors contributed to the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-021-26017-0>.

Correspondence and requests for materials should be addressed to Hongxu Ding or Joshua Stuart.

Peer review information *Nature Communications* thanks Gokcen Eraslan and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021

Chapter 4 : Leveraging Chemical Structure Information to Identify Methylation Status of Nanopore Reads

4.1 Chapter Introduction

Nanopore sequencing has made possible sequencing of a single strand of unamplified genetic material [47]. During sequencing, long strands of DNA or RNA block the pores embedded in a bilayer. Current that flows across the membrane is disrupted as nucleotide bases pass through it. The characteristic trace that the chemical structure of each base produces can be used to not only infer the identity of the nucleotide base, but also its potential modification status.

One of the major modifications the community is interested in is nucleotide methylation status. The current 'gold standard' method for profiling methylation (specifically cytosine methylation) in DNA has been bisulfite sequencing [48]. Such treatment is very harsh on the genetic material and leads to excessive DNA fragmentation. In addition, traditional short read sequencing only identifies short-range patterns of methylation [49]. As such, long-range nanopore sequencing is uniquely positioned to discover patterns of methylation that have not been previously characterized. Previous work has shown that 5-mC can be distinguished from cytosine by careful analysis of the electrical current signals measured by nanopore-based sequencing devices [50, 51].

To date, most modification prediction algorithms are based on kmer models. However, such learning strategies struggle to generalize knowledge between related kmers. Moreover, such approaches necessarily represent base modifications as distinct, unrelated characters. The upshot being that such kmer character-based models require extensive training data and are unable to predict the impact of a chemical modification *de novo*. Given that the number of possible kmers increases polynomially with the number of modifications being modeled, it is extremely

challenging to generate sufficient control data for such models, especially considering that more than 50 and 160 nucleotide modifications have been verified in DNA and RNA respectively [52, 53].

It is therefore clear that there is a need for models to learn general chemical structure rules in specific chemical contexts that can then be extended to nucleotide modifications that have not been seen by the model. As described in Chapter 2 graph convolutions have been successful in a number of fields, ranging from social media to protein-protein interaction networks and drug-target interaction prediction. In addition, I showed that such networks can generalize chemical information on drugs that have not been included in training.

The following work aims at applying similar graph convolutional networks presented in Chapter 2.3 to embed the structure of nucleotides from nanopore reads, and associating them to their distinctive pA value produced when they traverse the bilayer. This chemical embedding is then used to predict modification on DNA and RNA bases in a *de novo* manner. Specifically, we show that the model can predict 5mC and 2mG modifications, having only seen (trained on) canonical DNA.

The manuscript has been published in *Nature Communications*, 2021






ARTICLE



<https://doi.org/10.1038/s41467-021-26929-x>

OPEN

Towards inferring nanopore sequencing ionic currents from nucleotide chemical structures

Hongxu Ding ^{1,2,3}✉, Ioannis Anastopoulos ^{1,2,3}, Andrew D. Bailey IV ^{1,2,3}, Joshua Stuart ^{1,2}✉ & Benedict Paten ^{1,2}✉

The characteristic ionic currents of nucleotide kmers are commonly used in analyzing nanopore sequencing readouts. We present a graph convolutional network-based deep learning framework for predicting kmer characteristic ionic currents from corresponding chemical structures. We show such a framework can generalize the chemical information of the 5-methyl group from thymine to cytosine by correctly predicting 5-methylcytosine-containing DNA 6mers, thus shedding light on the de novo detection of nucleotide modifications.

¹Department of Biomolecular Engineering, UC Santa Cruz, Santa Cruz, CA, USA. ²UC Santa Cruz Genomics Institute, Santa Cruz, CA, USA. ³These authors contributed equally: Hongxu Ding, Ioannis Anastopoulos, Andrew D. Bailey IV. ✉email: hding16@ucsc.edu; jstuart@ucsc.edu; bpaten@ucsc.edu

During nanopore sequencing, consecutive nucleotide sequence kmers block the pores sequentially, producing ionic currents¹. Chemical modifications on nucleotides additionally alter the ionic currents measured during nanopore sequencing^{2–22}. The characteristic ionic currents of kmers, which are represented in kmer models, are used in interpreting nucleotide modifications^{2,3,7,23}. Up to now, 29^{2–11} and 30^{12–22} modifications have been successfully characterized in DNA and RNA, respectively.

To date, most modification analysis algorithms are based on kmer models^{2,3,7,23}. However, such learning strategies struggle to generalize knowledge between related kmers. For example, our previous hierarchical Dirichlet process approach could be structured to learn associations between kmers with specific shared properties, e.g., by numbers of pyrimidine bases, but could not generally learn relationships between arbitrary chemical similarities². Moreover, such approaches necessarily represent base modifications as distinct, unrelated characters. The upshot is that such kmer character-based models require extensive training data and are unable to de novo predict the impact of a chemical modification. Given that the number of possible kmers increases polynomially with the number of modifications being modeled, it is extremely challenging to generate sufficient control data for such models, especially considering that more than 50 and 160 nucleotide modifications have been verified in DNA and RNA, respectively^{24,25}.

To start to tackle this problem, we propose a graph convolutional network (GCN)-based deep learning framework^{26,27} for predicting kmer characteristic ionic currents from corresponding kmer chemical structures. We confirm that the proposed framework is able to represent individual kmer chemical modules, such as the phosphate group and the sugar backbone, as well as the nucleobase methyl and amine groups. We further demonstrate that this framework can infer full kmer models even when the training data does not include all possible kmers. This opens up the possibility of modeling kmers that are under-represented in control datasets. We also show that the framework can generalize the 5-methyl group in thymine to cytosine, thereby accurately predicting the characteristic ionic currents of 5-methylcytosine (5mC)-containing DNA 6mers. Such generalization of chemical information is a reason for optimism about the potential for de novo detection of nucleotide modifications.

Results

Architecture of the deep learning framework. Our deep learning framework consists of three groups of layers, including GCN layers, convolutional neural network (CNN) layers, and one fully connected neural network (NN) layer. As shown in Fig. 1A, the kmer chemical structures are first represented as graphs, with atoms as nodes and covalent bonds as edges. The atom chemical properties are then assigned as node attributes. Based on such graphs, GCN layers extract one chemical feature vector for every atom, by visiting its immediate graph neighbors. By this means, after several GCN layers, atom feature vectors will contain chemical information for all atoms within a certain graph distance. Specifically, this distance equals the number of GCN layers applied. Considering the small encoding distance of each layer of a GCN, to improve the encoding efficiency of the framework, CNN layers are then applied to summarize relatively long-range chemical information above the GCN layers. The output matrices of the final CNN layer are then “flattened” as feature vectors. Such feature vectors are then passed to the final fully connected NN layer to summarize kmer-level information and finally predict the kmer characteristic ionic currents (see “Methods”). For DNA and RNA, the corresponding best-performing architecture

in hyperparameter tuning was selected for downstream analysis (see “Methods”).

Kmer-level generalization. We first confirmed that the proposed framework can accurately predict characteristic ionic currents of kmers from their chemical structures. To do so, we performed a downsample analysis on the canonical DNA 6mer model provided by Oxford Nanopore Technologies (ONT, see “Methods”), by randomly partitioning canonical DNA 6mers with various train-test splits. For each train-test split group, we performed 5-fold cross-validation and used root mean square error (RMSE) and Pearson’s correlation (r) to quantify the goodness of fit (see “Methods”). As shown in Fig. 1B, Supplementary Fig. 1, and Supplementary Table 1, the performance stabilized as more than 40% of DNA 6mers were included in the training. Specifically, for DNA 6mers only used in the test, average RMSE and Pearson’s correlation reached 1 and 0.995, respectively. Such a result indicated on average 40% of randomly selected DNA 6mers contain sufficient information to recapitulate the full DNA 6mer model.

We next explored how specific kmer training subsets influence the ionic current predictions. Specifically, we trained the framework using either the DNA 6mers that (a) do not contain a given nucleotide (base dropout), (b) do not specify a nucleotide at a given position (position dropout), or (c) that are combined from different base dropouts (for instance, using the union of A-dropout and T-dropout kmers, such that kmers containing both A and T would be excluded, but not kmers containing either A or T, noted as A–T model combination, see “Methods” and Supplementary Note 1 for details). This latter combination analysis simulates the situation in which we have knowledge about two modifications independently, but must guess at the effect of their combination. For each group in (a–c), 50 independent repeats were performed, and goodness of fit was used to evaluate the performance. As shown in Fig. 1B and Supplementary Fig. 1, base and position dropouts significantly decreased the prediction power. Moreover, dropouts in third and fourth positions contributed the most to the decrease in prediction power, followed by the second and fifth positions, consistent with prior observations²⁸. Model combinations, on the other hand, in general, had a minor influence on the prediction power.

The above-mentioned analyses together suggest, once properly trained with sufficient and diverse 6mers, the kmer-level generalizability of the framework. To further validate and extend our framework, we performed all the above-mentioned analyses using RNA, switching to using 5mers instead of 6mers to match the available training data. Considering the significantly smaller amount of training data (1/4th the number of distinct RNA 5mers vs DNA 6mers), the prediction power of the RNA architecture is compromised. However, once trained with a similar number of kmers, the RNA architecture yielded comparable prediction power. For instance, the RNA 0.95–0.05 (972 training kmers) and DNA 0.25–0.75 (1024 training kmers) train-test splits yielded comparable performance on test data. Such a result suggests the validity of our proposed architecture (see “Methods,” Supplementary Fig. 2, and Supplementary Note 2 for details).

Such kmer-level generalizability could facilitate nucleotide modification detection by greatly reducing the required control data to generate reliable full modification-containing kmer models. As a proof of concept, we trained the DNA deep learning architecture with all canonical 6mers plus {1%, 5%, 10%, 30%, 50%, 70%, 90%} of randomly selected 5mC-containing 6mers (“modification imputation” analysis). The characteristic ionic current signals of such 5mC-containing DNA 6mers were

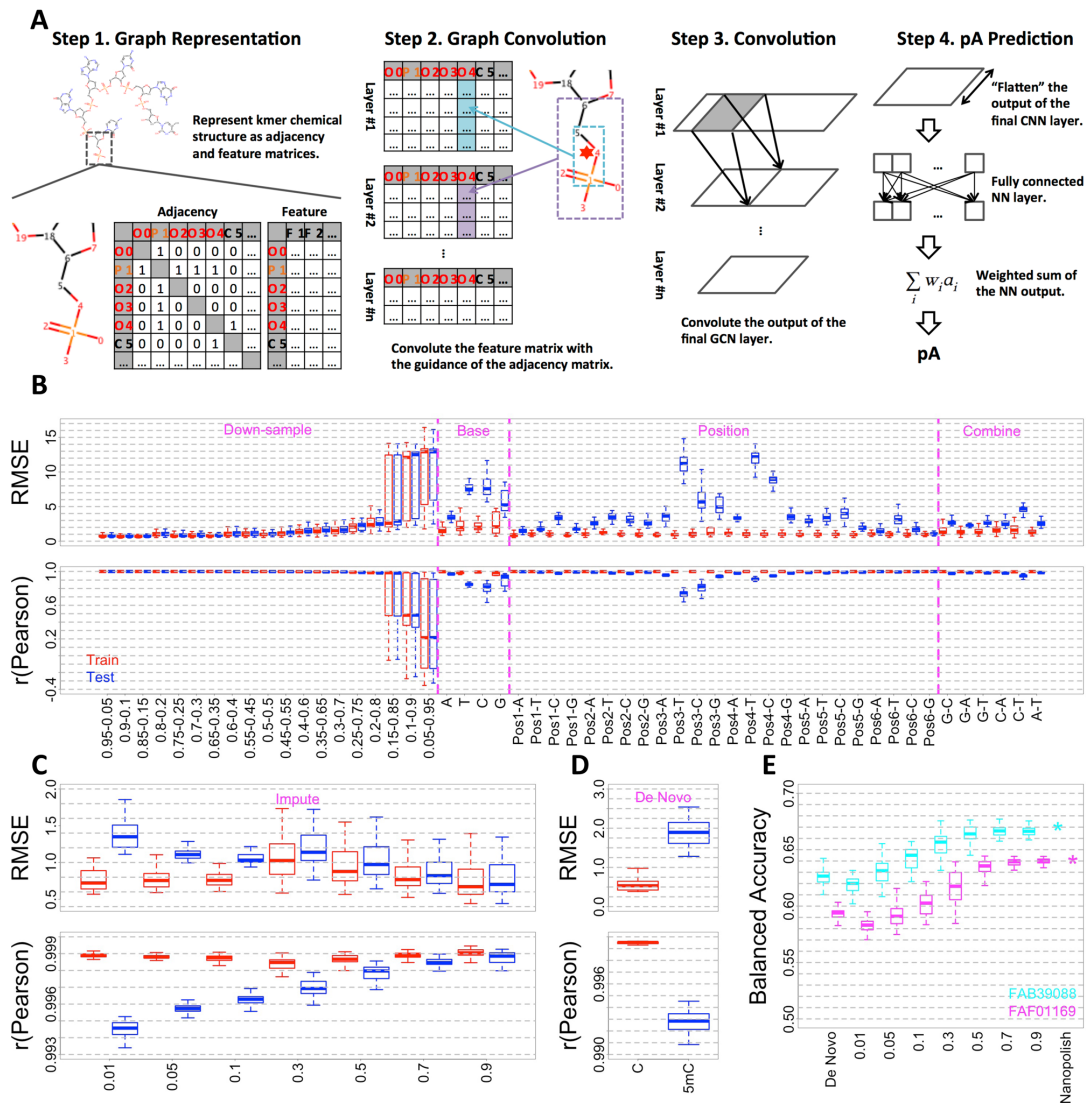


Fig. 1 Predicting kmer characteristic ionic currents from chemical structures. **A** Graphic overview of the proposed deep learning framework for DNA analysis. **B** Goodness of fit of DNA canonical random downsample, base-dropout, position-dropout, and model combination analyses. Specifically, “downsample” denotes the random dropout experiment, where we create random train-test splits. “Base” denotes base-dropout experiment, where we drop DNA 6mers that contain a specific base in any given position during training. “Position” denotes positional base-dropout experiment, where we drop DNA 6mers that contain a specific base in a given position during training. As for “combine,” we drop DNA 6mers that contain both of the specified bases during training. **C** Goodness of fit of 5mC-containing DNA 6mer imputation analysis. **D** Goodness of fit of de novo 5mC-containing DNA 6mer prediction. **C** and 5mC refer to the goodness of fit of canonical DNA 6mers and 5mC-containing DNA 6mers, respectively. In **B–D**, Train (red) and Test (blue) refer to the goodness of fit of the training and test DNA 6mers, respectively. **E** Predictive accuracy of C/5mC status quantified by balanced accuracy. Nanopolish, predictive analysis with the nanopolish model as baseline control. De Novo, predictive analysis with 5mC-containing DNA 6mer models described in **(D)**, which were predicted from canonical training. 0.01–0.9, predictive analysis with different imputation 5mC-containing DNA 6mer models as described in **(C)**. FAB39088 (cyan) and FAF01164 (purple) refer to two independent NA12878 cell line native genomic DNA nanopore sequencing datasets. Throughout **(B–E)**, the median, minimum/maximum (excluding outliers), and first/third quartile values were shown by the boxplots.

obtained from the nanopolish model as reported in refs. 3,23. For each training group, 50 independent repeats were performed (see “Methods”). As shown in Fig. 1C and Supplementary Fig. 3, decent goodness of fit could be obtained when as few as 5% of 5mC-containing DNA 6mers were used as training data.

Specifically, for test DNA 6mers, the average RMSE and Pearson’s correlation reached 1.2 and 0.995, respectively. Furthermore, models trained with the knowledge of 50% 5mC-containing DNA 6mers performed about as well as models trained with 90%.

Chemical group-level generalization in DNA 5mC de novo prediction. We noted that performance of the model on held out 5mC kmers trained with just 1% of 5mC kmers was better than chance. This raised the question of if chemical group-level information was being usefully generalized among nucleotides by our framework, potentially allowing the 5mC to be predicted de novo, without ever having been seen by the model. As a chemical derivative of cytosine, 5mC contains an additional methyl group at the fifth position (5-methyl) of the pyrimidine ring. This 5-methyl group is shared between 5mC and thymine. We thus hypothesized that 5mC can be generalized by combining the pyrimidine ring from cytosine and 5-methyl group from thymine. As a proof of concept, we trained the framework with all canonical DNA 6mers to make de novo predictions on 5mC-containing DNA 6mers. Similar to previous analyses, 50 independent repeats were performed, and the prediction power was first quantified by goodness of fit against the above-mentioned nanopolish model. As shown in Fig. 1D and Supplementary Fig. 3, although goodness of fit of 5mC-containing DNA 6mers was significantly worse than the canonical counterparts, decent performance could still be obtained (average RMSE and Pearson's correlation reached 1.8 and 0.993, respectively). We also compared the goodness of fit between canonical and 5mC-containing DNA 6mers, and as shown in Supplementary Fig. 4, a positive correlation trend could be observed. Such a result confirmed that no overfitting was introduced during architecture training with canonical DNA 6mers, and further suggested 5-methyl generalization.

Human genome C/5mC-status predictive analysis. We next performed “predictive analysis” to test whether the DNA 6mer models inferred by our deep learning framework could be used to correctly predict DNA C/5mC status at a per-read, per-site resolution from ionic currents (“predictive accuracy,” see “Methods”). C/5mC sites to be predicted were confirmed by bisulfite sequencing (see “Methods”). We also quantified the predictive accuracy with the above-mentioned nanopolish model as a baseline control (see “Methods”). As shown in Fig. 1E, average predictive accuracy, quantified by balanced accuracy (BA), became comparable with baseline control with 50% of imputed 5mC-containing 6mers. Taken together, these results confirmed the kmer-level generalizability of our framework, as well as suggesting that reliable modification-containing kmer models can be built with significantly less control data once facilitated by our methodology. Such a result confirmed the successful 5-methyl generalization. More confusion matrix-based prediction evaluations can be found in Supplementary Fig. 5.

The encoding of chemical structures. To better understand how chemical structures were encoded, we visualized DNA 6mer atom similarity matrices. Specifically, we trained the proposed framework with all canonical DNA 6mers. We then calculated and visualized the Pearson's correlations of the feature vectors derived by the final GCN layer as atom-level similarities. As shown in Supplementary Fig. 6, we visualized ten randomly chosen canonical DNA 6mers. Taking CGACGT as an example, as shown in Fig. 2A, C, atoms were in general aggregated by chemical contexts. For instance, as shown in (A), for the first cytidine monophosphate in CGACGT, atoms #0–4 were tightly clustered with average $r > 0.9$, recapitulating the phosphate group. Atoms #5–8 and #17–18 are also clustered with average $r > 0.9$, denoting the deoxyribose backbone. Among cytosine atoms #9–16, #9 nitrogen atom connected the nucleobase to the deoxyribose backbone, atoms #10–11 denoted the C=O group, and atoms #12–16 composed the C=C–C=N conjugation system and the covalently bonded amine group. Similarly, atoms in other

nucleotides can also be clustered into phosphate groups, deoxyribose backbones and nucleobases. Within the nucleobases, chemical modules including chemical groups and conjugation systems can further be dissected. Such a phosphate-deoxyribose-nucleobase pattern repeated and constituted DNA 6mers.

We also examined the inter-nucleotide similarities of different components. As shown in Fig. 2A, C, in general high similarities (average $r > 0.9$) were observed among phosphates, as well as deoxyriboses from different nucleotides. Meanwhile, chemical modules sharing similar structures, e.g., the conjugation systems of adenines, cytosines, and guanines were more similar to each other. On the other hand, low similarities (average $r < 0.5$) were observed between chemical modules with distinct structures, e.g., the cytosine C=O group and the thymine methyl group. Taken together, these results suggest that the GCN layers in the proposed framework can effectively capture features interpretable as individual chemical modules.

We further visualized the atom-level similarity matrices of 5mC-containing DNA 6mers, aiming to understand the generalization of methyl group among thymine and 5mC. We thus trained our deep learning framework with all canonical DNA 6mers, calculated the Pearson's correlations of the feature vectors derived by the final GCN layer, and further visualized such atom-level similarity matrices of ten randomly selected 5mC-containing DNA 6mers (Supplementary Fig. 7). Taking GT(5mC)AGA as an example (Fig. 2D, F), the phosphate-deoxyribose-nucleobase repetitive pattern was recapitulated. Within nucleobases, high similarities (average $r > 0.9$) were again observed among chemical modules with similar structures. Specifically, strong similarities (average $r > 0.9$) were observed between thymine (#38) and 5mC (#58) methyl groups (Me). In addition, such methyl groups were uniquely encoded as they were less correlated with any other DNA 6mer chemical modules (average $r < 0.5$). We also quantified the atom-level similarity between GT(5mC)AGA and corresponding canonical counterpart GTCAGA. As shown in Supplementary Fig. 8, strong similarities (average $r > 0.9$) were observed between GT(5mC)AGA and GTCAGA thymine methyl groups, as well as the 5mC-methyl groups from GT(5mC)AGA and thymine methyl groups from GTCAGA. These observations together suggested the successful chemical information generalization. Noticeably, the methyl groups were encoded with the pyrimidine backbone C=C modules. Such a result suggests that the GCN encoding is driven by chemical context, which further implies when generalizing one specific chemical group among different nucleotides, the corresponding chemical contexts in which such chemical group resides should be the same.

Finally, we projected kmer atom feature vectors into the tSNE space, in order to summarize the atom-level similarity matrices further providing a global visualization of kmer atoms. As shown in Fig. 2B, E, atoms under the same chemical context clustered together, e.g., phosphate group phosphate atoms (#1, #20, #42, #63, #82, and #104 in B and #1, #23, #43, #63, #84, and #106 in F) and deoxyribose ring oxygen atoms (#7, #26, #48, #69, #88, and #110 in B and #7, #29, #49, #69, #90, and #112 in E), as well as NH₂ group nitrogen atoms (#14, #35, #55, #76, and #97 in B and #16, #56, #76, #99, and #119 in E). Specifically, as shown in E, in 5mC-containing DNA 6mer GT(5mC)AGA, T-methyl group carbon atom #38 and 5mC-methyl group carbon atom #58 clustered together, along with pyrimidine backbone C=C module atoms #37 and #39 in T, as well as #57 and #59 in 5mC. Taken together, these results confirm that GCN could properly encode chemical structures based on the corresponding chemical contexts.

Analyzing the 2mG site in *Escherichia coli* 16S ribosomal RNA (rRNA). Our deep learning framework could potentially shed

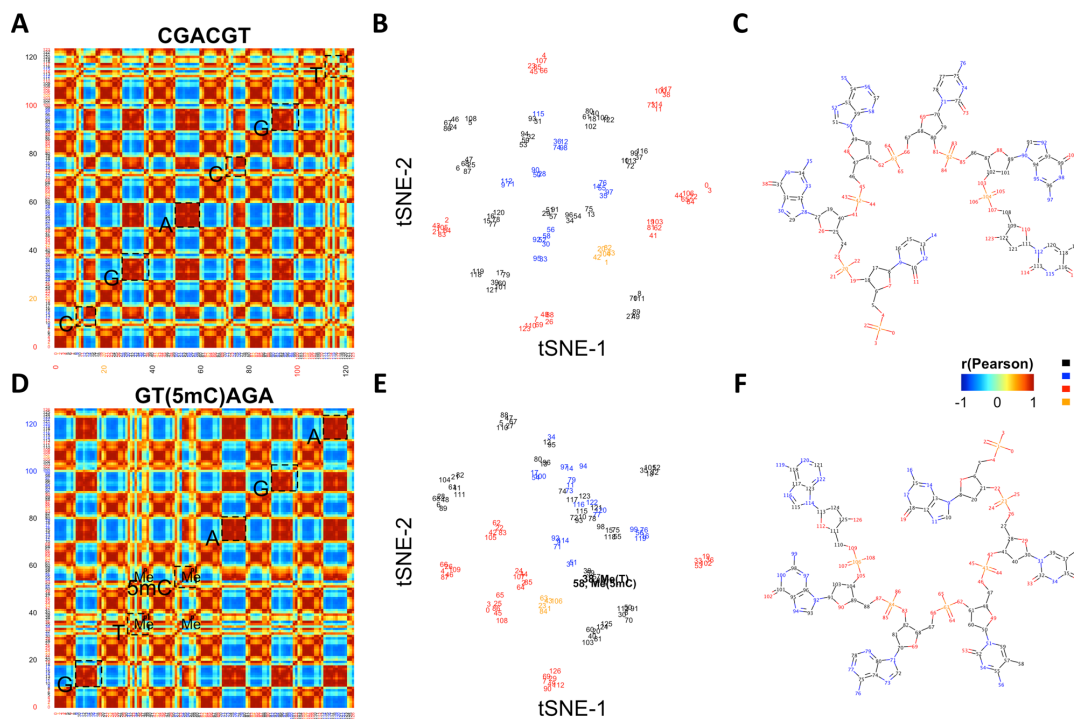


Fig. 2 Visualizing the encoding of chemical structures. **A–C** Atom similarity matrix, tSNE visualization, and chemical structure of the example canonical DNA 6mer CGACGT. In **A**, **B**, atoms were numbered and colored based on the chemical structure in **(C)**. Carbon, nitrogen, oxygen, and phosphorus were colored as black, blue, red, and orange, respectively. Specifically, in **A**, nucleobases were highlighted by dashed boxes. **D–F** Atom similarity matrix, tSNE visualization, and chemical structure of the example 5mC-containing DNA 6mer GT(5mC)AGA. In **D**, **E**, atoms were numbered and colored based on the chemical structure in **(F)**. Carbon, nitrogen, oxygen, and phosphorus were colored as black, blue, red, and orange, respectively. Specifically, in **D**, **E**, methyl group carbon atoms (#38 in T and #58 in 5mC) were highlighted.

light on previously understudied, less prevalent nucleotide modifications. As a proof of concept, we analyzed 2mG, which can be represented as the purine ring in guanine with the N2-methyl group in 6mA. Specifically, we generated an RNA 5mer model using canonical and 6mA-containing kmers (see “Methods”). We then predicted the characteristic ionic current signals of 2mG-containing RNA 5mers (see “Methods”). To test our predictions, we analyzed nanopore sequencing reads of *E. coli* 16S rRNA transcript J01859.1, which contains an annotated 2mG at position 1206 (see “Methods”). As shown in Supplementary Fig. 9, our predictions recapitulated the characteristic ionic current signals of 2mG-containing and pairing canonical RNA 5mers (see “Methods”). Moreover, we confirmed that such predicted characteristic ionic current signals could be used to correctly determine the G/2mG modification status (see “Methods”).

Discussion

We propose a GCN-based deep learning framework for associating kmer chemical structures with corresponding characteristic ionic currents. We show that such a framework can recapitulate full kmer models from partial training data, thus greatly facilitating modification analysis by reducing the amount of required control data. Specifically, for cases where a small proportion of random kmers are under-represented in control data, we can apply the same principle as the downsample analysis to learn around these training deficiencies. For cases where comprehensive control datasets are available only for single

modifications, we could apply model combination (as we showed for individual nucleotides) to model kmers containing multiple modifications simultaneously.

We further demonstrated that our framework can represent novel modifications by generalizing encoded chemical groups between nucleotides, thus shedding light on de novo modification detection. However, the current model is not without its limitations. For example, the proposed framework encodes chemical groups, e.g., the methyl groups in thymine and 5mC, as well as the amine groups in cytosine, guanine, and adenine, with covalently bonded “backbone atoms,” showing a strong chemical context-specificity (Fig. 2 and Supplementary Figs. 6 and 7). Thus, the current framework cannot properly handle “stacked” chemical groups. For instance, the methylamine group in N6-methyladenine (6mA) cannot be correctly encoded by simply stacking methyl with an amine. As shown in Supplementary Fig. 10, substituting A with 6mA was predicted to decrease characteristic ionic currents, which is the opposite of a previous study⁶. Therefore, the extensibility of the framework is currently limited. To overcome such a limitation, controlled nanopore sequencing profiles of diverse nucleotide modifications are needed, in addition to the modeling of other chemical interactions.

Deep learning-based approaches have emerged as powerful tools for detecting nucleotide modifications from nanopore sequencing readouts. Compared to kmer model-based counterparts, deep learning-based approaches are reported to have better accuracy and less computational resource consumption^{5,8}.

Recently, ONT released the megalodon algorithm (<https://github.com/nanoporetech/megalodon>), which can drastically increase the accuracy for 5mC identification (Supplementary Fig. 5, see “Methods”). Thus, one potential future extension of the paper would be using the learned models as components of a larger, recurrent deep NN.

Another potential future direction would be generalizing the proposed framework to handle both DNA and RNA kmers. Due to different translocation speeds, the nanopore sequencing ionic currents of DNA and RNA are not directly comparable²⁹. Therefore, advanced deep learning frameworks, which can take both kmer chemical structures and nanopore sequencing experimental setups, are needed. Considering DNA and RNA share several noncanonical nucleobases, e.g., inosine (I)³⁰, we might combine the ribose in RNA and I in DNA to reconstruct I-containing RNA 5mers, and vice versa for I-containing DNA 6mers. By this means, required RNA control nanopore sequencing reads, which are usually challenging to obtain, can be largely compensated. Meanwhile, such generalization would largely diversify the chemical contexts that can be represented, further facilitating the de novo modification analysis.

Methods

Graph representation of kmer chemical structures. Following the workflow described in ref. 26, kmer chemical structures were first described by SMILES (Simplified Molecular Input Line Entry System) strings, which were assembled by concatenating SMILES strings of individual nucleotides, as summarized in Table 1. Each nucleotide base can be described by several SMILES strings. The SMILES strings presented in Table 1 were selected due to the ease of combining them into complete kmers. Based on the information provided by ONT, as well as a previous study²⁸, DNA and RNA are represented by 6mer and 5mer, respectively. An “O” was then added to the end of each concatenation to represent the residual unbonded hydroxyl group on the sugar backbone.

We then represent the SMILES string of each kmer as a graph noted as $G(A, X)$. Specifically, the topology (atom order is determined by SMILES string) of each kmer chemical structure was represented by an adjacency matrix A , with A_{ij} equals 1 iff the i th and j th atoms were covalently bonded. Meanwhile, for every atom in A , the corresponding chemical properties were represented by feature matrix X , with X_i recording the chemical property vector for the i th atom. Atom chemical properties included in the study were summarized in Table 2.

Therefore, the GCN has encoded as input a chemical feature matrix X with the guide of chemical topology matrix A , representing kmer chemical structures. Notably, for convenient GCN implementation, the size of A and X is kept constant. Due to the variable number of atoms across kmers, A and X were thereby padded with zeros based on the largest kmers. Specifically, the A matrix was padded at the end of its rows and columns, with $\dim(A)$ is $\{133, 133\}$ and $\{116, 116\}$ for DNA and RNA, respectively. While the X matrix was padded at the end of its rows, with $\dim(X)$ is $\{133, 8\}$ and $\{116, 8\}$ for DNA and RNA, respectively. Note that the kmer representation is guided by the nonzero elements (covalent bonds) in A , thus such padding will not affect the GCN encoding.

Architecture of the deep learning framework. The GCN layers of our framework were built based on the procedure described by ref. 26. Fast approximate convolutions on G were used to create a graph-based NN $f(X, A)$, following the

propagation rule:

$$H^{(l+1)} = \sigma(\tilde{U}^{\dagger} \tilde{A} \tilde{U}^{\dagger} H^{(l)} W^{(l)})$$

$\sigma(\bullet)$ is the activation function applied to each layer. Here, the activation function used was the exponential linear unit (ELU). $\tilde{U}_{ij} = \sum_j A_{ij}$ is the degree matrix for each atom in the graph. $\tilde{A} = A + I$ adds self edges to each of the atoms. The $\tilde{U}^{\dagger} \tilde{A} \tilde{U}^{\dagger}$ transformation prevents changes in the scale of the feature vectors²⁶ and constructs filters for the averaging of neighboring node features. H and W denote the output (activation vectors) and weights of each GCN layer, respectively. The corresponding superscript represents the layer index. $H^0 = X$; however, subsequent H represents the GCN-derived features.

The intuition of the graph convolution process is described as follows. For every kmer, chemical properties of atoms, together with their covalently bonded neighbors, will be convoluted with the guidance of G . Such graph convolution yields an activation matrix H , following the aforementioned propagation rule. H is an atom-by-feature matrix, with dimensions $\{133, N\}$ and $\{116, N\}$ for each of the DNA and RNA kmers, respectively. Here, N equals the number of nodes of the GCN layer, which determines the number of features to be derived. The selection rule for N is described in the following section. As more GCN layers are stacked, the graph convolution process is repeated. The H matrix will thus contain chemical information of all atoms within a certain graph distance, which equals the number of GCN layers applied. By this means, “chemical modules” composed of several atoms linked by covalent bonds are encoded.

Considering the small encoding distance of a GCN, for a better encoding efficiency we wanted additional layers that can quickly summarize atom information. We thus applied standard 1-D CNN layers with rectified linear unit activation right after the GCN layers. Average Pooling³¹ was applied on the output of each 1-D CNN layer. Average Pooling takes the average of each 2×2 patch of the CNN output matrix. Specifically, the output dimension of the first CNN layer equals $\{133 - K + 1, N'\}$ and $\{116 - K + 1, N'\}$ for DNA and RNA kmers, respectively. Here, K is the CNN kernel size and N' is the node number of the final GCN layer. Output dimensions of subsequent CNN layers equals $\{m - K + 1 - 2 + 1, n - 2 + 1\}$, where $\{m, n\}$ denotes the output dimension of the previous layer and 2 denotes the Average Pooling patch size. The output from the final 1-D CNN layer, after Average Pooling, was passed to a Flatten layer, which converts the final 1-D CNN output matrix to a 1-D feature vector in a row-wise fashion. The NN layer then takes the flattened vector as input, thereby summarizing information about the entire kmer and producing a highly informative representation. Elements of the NN layer output vector are linearly combined as the final pA value.

Training procedure. Our framework was trained with the Keras³² framework (2.3.1) with TensorFlow backend using the Adam³³ optimizer for gradient descent optimization. The framework was allowed to train for a maximum of 500 epochs. To control for overfitting, EarlyStopping³⁴ was used by monitoring the increase in validation loss. Early termination of training was reached if the validation loss was increasing for ten consecutive epochs, indicating that the framework had reached maximum convergence. A mean-squared error was used as the loss function during the training process. Meanwhile, a 10% random dropout was applied after each layer, to further prevent overfitting³⁵. In the following experiments, the exact same training routine was used.

Hyperparameter tuning. In order to determine the optimal architecture, we performed a hyperparameter grid search. The search involved the hyperparameters shown in Table 3.

We used the following scaling factor to determine the number of nodes in each GCN/CNN layer of our framework:

$$n = 16 \times 2^{(l-1)},$$

Table 1 SMILES strings of individual nucleotides.

Nucleotide	SMILES string
A (DNA)	OP(=O)(O)OCC1OC(N3C=NC2=C(N)N=CN=C23)CC1
T (DNA)	OP(=O)(O)OCC1OC(N2C(=O)NC(=O)C(C)=C2)CC1
C (DNA)	OP(=O)(O)OCC1OC(N2C(=O)N=C(N)C=C2)CC1
G (DNA)	OP(=O)(O)OCC1OC(N2C=NC3=C2N=C(N)NC3=O)CC1
5mC (DNA)	OP(=O)(O)OCC1OC(N2C(=O)N=C(N)C(C)=C2)CC1
6mA (DNA)	OP(=O)(O)OCC1OC(N3C=NC2=C(NC)N=CN=C23)CC1
A (RNA)	OP(=O)(O)OCC1OC(N3C=NC2=C(N)N=CN=C23)C(O)C1
U (RNA)	OP(=O)(O)OCC1OC(N2C(=O)NC(=O)C=C2)C(O)C1
C (RNA)	OP(=O)(O)OCC1OC(N2C(=O)N=C(N)C=C2)C(O)C1
G (RNA)	OP(=O)(O)OCC1OC(N2C=NC3=C2N=C(N)NC3=O)C(O)C1
6mA (RNA)	OP(=O)(O)OCC1OC(N3C=NC2=C(NC)N=CN=C23)C(O)C1
2mG (RNA)	OP(=O)(O)OCC1OC(N2C=NC3=C2N=C(NC)NC3=O)C(O)C1

Table 2 Atom chemical properties included in the study.

Feature	Description
Carbon	1 if the atom is carbon, 0 otherwise (boolean)
Nitrogen	1 if the atom is nitrogen, 0 otherwise (boolean)
Oxygen	1 if the atom is oxygen, 0 otherwise (boolean)
Phosphorus	1 if the atom is phosphorus, 0 otherwise (boolean)
Atom degree	Total number of covalent bonds around an atom (integer)
Implicit valence	It equals the valence of the atom minus the valence calculated from the bond connections (integer)
Number of hydrogens	Total count of hydrogens (integer)
Aromaticity	1 if atom in an aromatic ring, 0 otherwise (boolean)

Table 3 Hyperparameters searched in the study.

Parameters	Space searched	ATGC DNA	AUGC RNA	A(6mA)UGC RNA
The number of GCN layers	{2, 3, 4, 5, 6}	4	4	6
The number of CNN layers	{2, 3, 4, 5, 6}	3	5	6
The kernel size for the CNN layers	{2, 4, 10, 20}	10	10	10
The number of nodes in the dense (NN) layer	{32, 128, 512, 2048, 8192}	8192	8192	8192

where l is the layer index of the GCN, CNN, and NN layer groups. For instance, the number of GCN layers determined to yield the best performance for DNA was 4. The number of nodes for each GCN layer was therefore 128, 64, 32, and 16. The same logic was applied to all other layer groups.

We performed 10-fold cross-validation for each hyperparameter combination. The combination that produced the lowest average RMSE across all folds was adopted as the optimal architecture. The optimal framework for DNA analysis (ATGC DNA) has four GCN layers and three CNN layers with a kernel size of 10 and 8192 nodes in the NN layer. The optimal framework for canonical RNA analysis (AUGC RNA) has four GCN layers and five CNN layers with a kernel size of 10 and 8192 in the NN layer. The optimal framework for modified RNA analysis (A(6mA)UGC RNA) has six GCN layers and six CNN layers with a kernel size of 10 and 8192 in the NN layer.

Downsample, base-dropout, position-dropout, and combination analysis. For downsample analysis, we performed random train-test splits in 5% intervals, noted as 0.95–0.05, etc. For base-dropout analysis, we created training sets by removing certain bases. Such train-test split creates 729/4096 (18%) training kmers and 3367/4096 (82%) test kmers for DNA and 243/1024 (24%) training kmers and 781/1024 (76%) test kmers for RNA. It is important to note that everytime a base is dropped from the training set, it is retained in the test set. Similar to base dropout, the position dropout adds one more dimension, which is the position of the nucleotide base. For a given position dropout, the testing kmers are all kmers with the dropout nucleotide covering the target position, and the training kmers are the remaining kmers. Such position dropout creates 3072/4096 (75%) training kmers and 1024/4096 (25%) test kmers for DNA and 768/1024 (24%) training kmers and 256/1024 (25%) test kmers for RNA. It is important to note that bases dropped in a specific position in the training appear in the same position in testing. For combination analysis, we trained the framework by combining any of the two base-dropout kmer sets. For instance, all G- and C-dropout DNA 6mers were noted as G-C. Such analysis creates 1394/4096 (34%) training kmers and 2702/4096 (66%) test kmers for DNA and 454/1024 (44%) training kmers and 570/1024 (56%) test kmers for RNA. For each above-mentioned train-test split, in order to perform statistical analyses, we produced 50 independently trained frameworks for each experiment. Specifically, we performed 50-fold cross-validation in the downsample analysis, considering for each fold the train kmers were randomly selected. As for other analyses, we performed 50 independent repeats using the same training kmer sets. The variability among repeats came from the stochasticity of the training process. To confirm the robustness of our architecture, we further performed two independent replicates (Run-1 and Run-2) of 50.

Predicting modification-containing kmers. For the 5mC imputation experiment, the framework was trained on all 4096 {A, T, C, G} DNA 6mers plus {1%, 5%, 10%, 30%, 50%, 70%, 90%} of randomly selected 5mC-containing DNA 6mers, following the training process as described above. In order to perform statistical analyses, we produced 50 independently trained frameworks (50 independent repeats) for each category, with a total of two independent replicates (Run-1 and Run-2) of 50. Such frameworks were then applied on all 15,625 possible {A, T, C, G, 5mC} DNA 6mers.

For the chemical group-level generalization experiment, the framework was trained on all 4096 {A, T, C, G} DNA 6mers following the training process as described above. In order to perform statistical analyses, we produced 50 independently trained frameworks (50 independent repeats), with a total of two

independent replicates (Run-1 and Run-2) of 50. Such frameworks were then applied on all 15,625 possible DNA 6mers, including those composed of {A, T, C, G, 5mC} and {A, T, C, G, 6mA}.

For the 2mG prediction experiment, the framework was trained by the union of {A, U, C, G} and {6mA, U, G, C} RNA 5mers (GSE124309 model, in total 1805 RNA 5mers), which were reported in ref. ¹³, following the training process as described above. In order to perform statistical analyses, we produced 50 independently trained frameworks (50 independent repeats). Such frameworks were then applied on all 7776 possible {A, 6mA, U, G, 2mG, C} RNA 5mers.

Human genome C/5mC-status predictive analysis

Overview. To test whether the predicted {A, T, G, C, 5mC} DNA 6mer models can be used to correctly interpret C/5mC status from nanopore readouts, we performed predictive analysis by using signalAlign to make per-read per-base predictions². For a given reference position, signalAlign can produce posterior probabilities for all possible bases based on a provided kmer model. Thus, for DNA 6mer models generated as described in “predicting modification-containing kmers,” the empirical nanopolish^{3,23} model obtained as described in “kmer models,” we allowed signalAlign to predict between C and 5mC. Considering no significant goodness-of-fit differences were observed between Run-1 and 2, only models generated in Run-1 were used here. All predictive analyses performed in this paper were within the human NA12878 cell line.

Selecting prediction sites. The prediction sites were selected among the entire human genome. To avoid artifacts caused by ambiguous genomic DNA modification status, we only focused on confident 5mC sites and canonical genomic regions in our analysis. Besides 5mC, other modifications exist in genomic DNA. Considering extremely low fractions of other modifications, e.g., only ~0.05% are modified as 6mAs in the human genome³⁶, we define “non-5mC” sites as “canonical regions” during predictive analysis. Among these canonical regions, we used the Poisson process with lambda equals 50 to randomly select genomic sites for signalAlign to predict. Such selected sites were at least 12 nucleotides apart, avoiding potential interference by the neighbors. We thus obtained confident 5mC and C sites for signalAlign prediction.

The genomic DNA C/5mC status was determined by analyzing two independent NA12878 cell line bisulfite sequencing datasets³⁷. A C site was determined as confidently methylated if, for both bisulfite sequencing datasets, 95% of reads were methylated with at least 10x coverage. On the other hand, a C site was considered confidently unmethylated if, for both bisulfite sequencing datasets, at most 1% of reads were methylated with at least 10x coverage. Such analysis covered 3367/3367 canonical C-containing DNA 6mers and 3950/6144 single-5mC-containing DNA 6mers.

Selecting nanopore sequencing reads. We then ran signalAlign with reads reported in the nanopore consortium NA12878 cell line native genomic DNA datasets³⁸ covering the above-mentioned prediction sites. Considering the computational complexity of signalAlign, we performed the following filtering steps to use the fewest reads to cover the most kmers. First, we calculated read-level kmer coverage. For example, the center 5mC site of DNA read CAGAT(5mC)ACAGA was selected for signalAlign prediction. 6mers CAGAT(5mC), AGAT(5mC)A, GAT(5mC)AC, AT(5mC)ACA, T(5mC)ACAG, and (5mC)ACAGA span such 5mC site and therefore are considered as being covered. Based on such read-level kmer coverage, we iteratively selected reads that covered the least frequently

covered kmers. Thus, building a read set that covers as many kmers as possible as often as possible with the fewest number of reads. We included two biological replicates of NA12878 cell line native genomic DNA-sequencing experiments (FAB39088 and FAF01169) in the C/5mC predictive analysis. For such analysis, our final FAB39088 set contained 1706 reads, which covered 2625/3367 C-only DNA 6mers with an average 61.52× coverage as negative control and 3105/3950 possible single-5mC DNA 6mers with an average 5.01× coverage. The final FAF01169 set contained 1396 reads, which covered 2610/3367 C-only DNA 6mers with an average 63.26× coverage as negative control and 3140/3950 single-5mC DNA 6mers with an average 4.76× coverage. Combining the two sets, in total 2792/3367 C-only DNA 6mers were covered with an average 58.49× coverage and 3481/3950 single-5mC DNA 6mers were covered with an average 4.38× coverage.

Performing signalAlign prediction. Based on the selected prediction sites and nanopore sequencing reads as described above, per-read per-site predictive analysis was performed by signalAlign. The signalAlign analysis was performed with default parameters, except for internal read-level quality filtering. Such quality filtering removes reads with poor kmer-to-ionic current correspondence. During signalAlign analysis, kmer-to-ionic current correspondence probability matrices (event tables) are first generated. Based on such event tables, signalAlign will remove reads with low average probabilities ($<10^{-5}$). In addition, reads with >50 consecutive ionic current signals that cannot be corresponded to kmers (probability equals 0) will be discarded. Considering that the event table generation is based on the provided kmer model, after the above-mentioned default quality filtering, the number of remaining reads varies when different kmer models are supplied during predictive analysis. To ensure the statistical soundness, we deactivate the default quality filtering, such that reads to be analyzed by different supplied kmer models will be the same.

Performing megalodon prediction. We also performed predictive analysis using the deep learning-based modification calling algorithm megalodon (<https://github.com/nanoporetech/megalodon>) as an additional baseline control. The megalodon (version 2.3.1) analysis was performed with tags “<fast5>--outputs mod_mappings mods --reference <reference>--processes 1 --overwrite --guppy-server-path guppy_basecall_server --output-directory <output dir>--guppy-time-out 1000 --guppy-concurrent-reads 1 --guppy-params'--num_callers 7--cpu_threads_per_caller 10--chunks_per_runner 100.”

Considering the extraordinary performance of megalodon (Supplementary Fig. 5), we further used megalodon predictions as additional ground truth for the C/5mC status for every nanopore sequencing read at every prediction site. Please see Supplementary Note 3 for more information.

Quantifying predictive accuracy. signalAlign quantifies the probability of being C or 5mC for every prediction. We used probability threshold 0.7 to ensure only confident predictions were included in predictive accuracy quantification. Together with the megalodon 5mC calling results, we further created confusion matrices (2×2 for 5mC predictive analysis with 5mC as “positive” class and C as “negative” class) to quantify predictive accuracy. Specifically, we calculated the true-positive rate, true-negative rate, positive predictive value, negative predictive value, F1 score (F1), and BA as predictive accuracy quantifications. BA was presented in Fig. 1E as representative quantification and the full predictive performance can be found in Supplementary Fig. 5.

Escherichia coli 16S rRNA 2mG-site analysis

Ionic current signal distributions. We first downloaded the nanopore sequencing fast5 reads of *E. coli* 16S rRNA nanopore sequencing reads reported in ref. 14. We then performed nanopore extract analysis^{3,23} to retrieve the fastq records, with tags “-v -r -q -t template.” The fastq records were then aligned using minimap2 (2.16-r922)³⁹ with flags “-ax map-ont,” further sorted and indexed by samtools (1.12)⁴⁰. Per-read event tables were generated using nanopore eventalign with flag “-scale-events,” by taking fast5 reads, alignment files, and retrieved fastq records as described above. The yielded event tables contain RNA 5mer sequences and corresponding ionic current signals. We then quantified the distributions of RNA 5mer ionic current signals.

Predictive analysis. We also performed predictive analysis for the {A, 6mA, T, G, 2mG, C} RNA 5mer model described in “predicting modification-containing kmers.” Specifically, we tested whether the predicted RNA 5mer model could be used to correctly identify the position of 1206 2mG site, as well as three nearby G sites (positions 851, 1221, and 1386) in *E. coli* 16S rRNA (see <https://www.ncbi.nlm.nih.gov/nucleotide/J01859> for details). We thus ran signalAlign with nanopore sequencing reads reported in ref. 14, following the same steps as described in “human genome C/5mC-status predictive analysis.” We also used probability threshold 0.7 to select confident predictions.

Kmer models. Canonical DNA 6mer and RNA 5mer models are available at: https://github.com/nanoporetech/kmer_models. The nanopore 5mC-containing DNA 6mer model is available at: <https://github.com/nanoporetech/nanopolish/tree/master/etc/r9-models>. The GSE124309 model, which contains the union of

{A, U, C, G} and {6mA, U, G, C} RNA 5mers, was constructed by the following steps. We first downloaded the nanopore sequencing fast5 reads of modified and non-modified “curlcake constructs” replicate 1 with GEO accession code GSE124309¹³. We then performed nanopore extract analysis^{3,23} to retrieve the fastq records, with tags “-v -r -q -t template.” The fastq records were then aligned using minimap2 (2.16-r922)³⁹ with flags “-ax map-ont,” further sorted and indexed by samtools (1.12)⁴⁰. Per-read event tables were generated using nanopore eventalign (0.11.1) with flag “-scale-events,” by taking fast5 reads, alignment files, and retrieved fastq records as described above. The yielded event tables contain RNA 5mer sequences and corresponding ionic current signals. For every RNA 5mer, we averaged ionic current signals of all instances recorded in the event tables to build the GSE124309 model. Please note that for more recent nanopore sequencing chemistries, e.g., R10 where ONT kmer models are no longer available, empirical kmer models could be trained instead as above-mentioned. Please see Supplementary Note 4 for details.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The FAB39088 and FAF01169 NA12878 cell line native genomic DNA nanopore sequencing datasets were downloaded from <https://github.com/nanopore-wgs-consortium/NA12878/blob/master/Genome.md>. The two independent NA12878 bisulfite datasets were downloaded from <https://www.encodeproject.org/experiments/ENCSTR890UQO/>. The *E. coli* 16S rRNA nanopore sequencing dataset was reported by Smith et al.¹⁴. The nanopore sequencing dataset used to construct the GSE124309 model is available at GEO under the accession code GSE124309¹³.

Code availability

Codes for constructing, training, and running the deep learning framework are available at https://github.com/iaonisa92/Nanopore_modification_inference⁴¹. Codes for nanopore sequencing data analysis are available at https://github.com/adbailey4/functional_model_analysis⁴². Specifically, we adapted the original nanopore (0.11.1) for our analysis. The adapted nanopore is available at <https://github.com/adbailey4/nanopolish>⁴³. Codes for reproducing all figures are available upon request to the corresponding authors.

Received: 15 July 2021; Accepted: 19 October 2021;

Published online: 11 November 2021

References

- Deamer, D., Akeson, M. & Branton, D. Three decades of nanopore sequencing. *Nat. Biotechnol.* **34**, 518 (2016).
- Rand, A. C. et al. Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods* **14**, 411 (2017).
- Simpson, J. T. et al. Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods* **14**, 407 (2017).
- Liu, Q., Georgieva, D. C., Egli, D. & Wang, K. NanoMod: a computational tool to detect DNA modifications using Nanopore long-read sequencing data. *BMC Genomics* **20**, 31–42 (2019).
- Liu, Q. et al. Detection of DNA base modifications by deep recurrent neural network on Oxford Nanopore sequencing data. *Nat. Commun.* **10**, 2449 (2019).
- McIntyre, A. B. et al. Single-molecule sequencing detection of N 6-methyladenine in microbial reference materials. *Nat. Commun.* **10**, 1–11 (2019).
- Mueller, C. A. et al. Capturing the dynamics of genome replication on individual ultra-long nanopore sequence reads. *Nat. Methods* **16**, 429 (2019).
- Ni, P. et al. DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics* **35**, 4586–4595 (2019).
- Georgieva, D., Liu, Q., Wang, K. & Egli, D. Detection of base analogs incorporated during DNA replication by nanopore sequencing. *Nucleic Acids Res.* **48**, e88–e88 (2020).
- Kot, W. et al. Detection of preQ0 deazaguanine modifications in bacteriophage CAJAN DNA using Nanopore sequencing reveals same hypermodification at two distinct DNA motifs. *Nucleic Acids Res.* **48**, 10383–10396 (2020).
- Nookaew, I. et al. Detection and discrimination of DNA adducts differing in size, regiochemistry, and functional group by Nanopore sequencing. *Chem. Res. Toxicol.* **33**, 2944–2952 (2020).
- Leger, A. et al. RNA modifications detection by comparative Nanopore direct RNA sequencing. Preprint at *bioRxiv* <https://doi.org/10.1101/843136> (2019).

13. Liu, H. et al. Accurate detection of m⁶A RNA modifications in native RNA sequences. *Nat. Commun.* **10**, 1–9 (2019).
14. Smith, A. M., Jain, M., Mulrone, L., Garalde, D. R. & Akeson, M. Reading canonical and modified nucleobases in 16S ribosomal RNA using nanopore native RNA sequencing. *PLoS ONE* **14**, e0216709 (2019).
15. Viehweger, A. et al. Direct RNA nanopore sequencing of full-length coronavirus genomes provides novel insights into structural variants and enables modification analysis. *Genome Res.* **29**, 1545–1554 (2019).
16. Workman, R. E. et al. Nanopore native RNA sequencing of a human poly (A) transcriptome. *Nat. Methods* **16**, 1297–1305 (2019).
17. Lorenz, D. A., Sathe, S., Einstein, J. M. & Yeo, G. W. Direct RNA sequencing enables m⁶A detection in endogenous transcript isoforms at base-specific resolution. *RNA* **26**, 19–28 (2020).
18. Maier, K. C., Gressel, S., Cramer, P. & Schwalb, B. Native molecule sequencing by nano-ID reveals synthesis and stability of RNA isoforms. *Genome Res.* **30**, 1332–1344 (2020).
19. Parker, M. T. et al. Nanopore direct RNA sequencing maps the complexity of *Arabidopsis* mRNA processing and m⁶A modification. *Elife* **9**, e49658 (2020).
20. Stephenson, W. et al. Direct detection of RNA modifications and structure using single molecule nanopore sequencing. Preprint at *bioRxiv* <https://doi.org/10.1101/2020.05.31.126763> (2020).
21. Aw, J. G. A. et al. Determination of isoform-specific RNA structure with nanopore long reads. *Nat. Biotechnol.* **39**, 336–346 (2021).
22. Gao, Y. et al. Quantitative profiling of N⁶-methyladenosine at single-base resolution in stem-differentiating xylem of *Populus trichocarpa* using Nanopore direct RNA sequencing. *Genome Biol.* **22**, 1–17 (2021).
23. Loman, N. J., Quick, J. & Simpson, J. T. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat. Methods* **12**, 733 (2015).
24. Sood, A. J., Viner, C. & Hoffman, M. M. DNAmoD: the DNA modification database. *J. Cheminform.* **11**, 1–10 (2019).
25. Boccaletto, P. et al. MODOMICS: a database of RNA modification pathways. 2017 update. *Nucleic Acids Res.* **46**, D303–D307 (2018).
26. Duvenaud, D. K. et al. in *Advances in Neural Information Processing Systems* 2224–2232 (2015).
27. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. Preprint at *arXiv* <https://arxiv.org/abs/1609.02907> (2016).
28. Ding, H., Bailey, A. D., Jain, M., Olsen, H. & Paten, B. Gaussian mixture model-based unsupervised nucleotide modification number detection using Nanopore sequencing readouts. *Bioinformatics* **36**, 4928–4934 (2020).
29. Derrington, I. M. et al. Nanopore DNA sequencing with MspA. *Proc. Natl Acad. Sci. USA* **107**, 16060–16065 (2010).
30. Alseth, I., Dalhus, B. & Bjørås, M. Inosine in DNA and RNA. *Curr. Opin. Genet. Dev.* **26**, 116–123 (2014).
31. LeCun, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**, 541–551 (1989).
32. Chollet, F. et al. Keras. *GitHub*. Retrieved from <https://github.com/fchollet/keras> (2015).
33. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at *arXiv* <https://arxiv.org/abs/1412.6980> (2014).
34. Yao, Y., Rosasco, L. & Caponnetto, A. On early stopping in gradient descent learning. *Constr. Approx.* **26**, 289–315 (2007).
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
36. Xiao, C. L. et al. N⁶-methyladenine DNA modification in the human genome. *Mol. Cell* **71**, 306–318 (2018).
37. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**, 57–74 (2012).
38. Jain, M. et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338 (2018).
39. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
40. Li, H. et al. The sequence alignment/map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
41. Ding, H., Anastopoulos, I., Bailey, A. D., Stuart, J. & Paten, B. Towards inferring nanopore sequencing ionic currents from nucleotide chemical structures. *Zenodo* <https://doi.org/10.5281/zenodo.5574151> (2021).
42. Ding, H., Anastopoulos, I., Bailey, A. D., Stuart, J. & Paten, B. Towards inferring nanopore sequencing ionic currents from nucleotide chemical structures. *Zenodo* <https://doi.org/10.5281/zenodo.5571020> (2021).
43. Ding, H., Anastopoulos, I., Bailey, A. D., Stuart, J. & Paten, B. Towards inferring nanopore sequencing ionic currents from nucleotide chemical structures. *Zenodo* <https://doi.org/10.5281/zenodo.5571031> (2021).

Acknowledgements

Research reported in this publication was supported by the National Institutes of Health under Award Numbers R01-HG010053-02, U01HG010961, U41HG010972, R01HG010485, 2U41HG007234, 5U54HG007990, 5T32HG008345-04, and U01HL137183. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors would thank Jordan Eizenga, Dr. Jonas Sibbesen, Dr. Mark Akeson, and Dr. Miten Jain for critical insight and help with drafting the manuscript.

Author contributions

H.D. conceived the idea. I.A. performed deep learning framework modeling, optimization, and analysis. A.D.B. and H.D. performed the nanopore sequencing data analysis. H.D., J.S., and B.P. supervised the project. All authors prepared the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-021-26929-x>.

Correspondence and requests for materials should be addressed to Hongxu Ding, Joshua Stuart or Benedict Paten.

Peer review information *Nature Communications* thanks the anonymous reviewers for their contribution to the peer review of this work. Peer reviewer reports are available.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021

Chapter 5 : Future Directions and Other Works

5.1 Future Direction: Drug Response in Tumor Patients

The current framework presented in Chapter 2.3 is a novel way to integrate patient information of the same tissue of origin with cell lines. In addition the framework is able to integrate the chemical information of every molecule in the training. I have shown that including such information during training not only improves the accuracy of the discrimination between resistant and sensitive patients, but also the number of drugs the model can have an accurate prediction in. This means the model can be extended to more drugs.

However, the model lacks in two main areas: a) interpretability, and b) not including bond information of each molecule. For a) there is an opportunity to combine the model presented in Chapter 4.2 with the Expression Module of the model presented in Chapter 2.3. This joining will allow the model to be interpretable at a pathway level. Furthermore, due to the flexibility VEGA provides cell-states, transcription factors, or a protein-protein interaction network can be used as input to its latent space. Thus, the drug response prediction could be interpreted in a variety of ways. As for b) the current model only includes atom level features. There is an opportunity to use a more expressive graph convolutional network, such as the GIN. This network can incorporate bond level features along with atomic level features. The hope is that by included a more expressive embedding space for each molecule, the model will be able to be extended to even more drugs.

5.2 Future Direction: Molecular Structure Embedding in Nanopore Sequencing

Similarly to what is described in the section above, the model developed for identifying *de novo* the methylation status of nucleotides from nanopore reads only relies on atomic level

features. As such, a more expressive graph convolutional network is needed so that the model can be potentially extended to modifications beyond 5mC and 2mG.

At the same time, enough high quality nanopore data do not exist yet. Therefore, the evaluation of the model on more modifications is currently a daunting task, if not impossible.

5.3 Other work: Internship at Coral Genomics Inc.

During my internship at Coral Genomics Inc. I developed a model that is able to handle multiple drugs per patient. The goal was to use all the drugs each patient has been prescribed and develop a model which, coupled with demographic or genetic information, can predict the probability of hospitalization or death. FDA has created the FDA Adverse Event Reporting System (FAERS) which collects information on records of adverse event reports, medication error reports and product quality complaints resulting in adverse events that were submitted to the FDA. This dataset includes details on demographics, drugs, indications, outcomes, reactions, sources and therapies for the related events.

I used this information for each patient by creating a model that can encode the variable number of drugs each patient is on, along with demographics data. I experimented with traditional molecular representations, such as Morgan fingerprints, and compared the performance of the model with graph convolution. Graph convolutions provided a superior encoding to the static ones that Morgan fingerprints provide.

While the focus of this study was incorporating chemical structure into adverse event prediction there are other avenues that future investigations could take to improve on this work. For example, there is much attention on how socioeconomic factors might also influence the quality of the prediction such models make [54]. In our work, we attempted to capture ethnic influences using a PCA characterization of major DNA variants. PCA was also used to capture some of the clinical information. The PCA reduction provides (hopefully) a non-redundant set of features


for learning. Although such data are sparse in the FAERS database, it would be interesting to encode additional socioeconomic-related variables. Then one could imagine training models with and without these new variables to measure the amount of predictive information they carry. A joint encoding of DNA, clinical, and socio-economic features could allow for correlations among them to be accounted as well.

The following manuscript has been published on the International Journal of Environmental Research and Public Health, on March 2021.



Article

Multi-Drug Featurization and Deep Learning Improve Patient-Specific Predictions of Adverse Events

Ioannis N. Anastopoulos^{1,2}, Chloe K. Herczeg², Kasey N. Davis² and Atray C. Dixit^{2,*} 

¹ Biomolecular Engineering, University of California, Santa Cruz, CA 95064, USA; ianastop@ucsc.edu

² Coral Genomics, Inc., 953 Indiana St., San Francisco, CA 94107, USA; chloe@coralgenomics.com (C.K.H.); kasey@coralgenomics.com (K.N.D.)

* Correspondence: atray@coralgenomics.com

Abstract: While the clinical approval process is able to filter out medications whose utility does not offset their adverse drug reaction profile in humans, it is not well suited to characterizing lower frequency issues and idiosyncratic multi-drug interactions that can happen in real world diverse patient populations. With a growing abundance of real-world evidence databases containing hundreds of thousands of patient records, it is now feasible to build machine learning models that incorporate individual patient information to provide personalized adverse event predictions. In this study, we build models that integrate patient specific demographic, clinical, and genetic features (when available) with drug structure to predict adverse drug reactions. We develop an extensible graph convolutional approach to be able to integrate molecular effects from the variable number of medications a typical patient may be taking. Our model outperforms standard machine learning methods at the tasks of predicting hospitalization and death in the UK Biobank dataset yielding an R^2 of 0.37 and an AUC of 0.90, respectively. We believe our model has potential for evaluating new therapeutic compounds for individualized toxicities in real world diverse populations. It can also be used to prioritize medications when there are multiple options being considered for treatment.

Keywords: adverse events; real world evidence; neural networks; graph convolution; FDA FAERS; UK Biobank



Citation: Anastopoulos, I.N.; Herczeg, C.K.; Davis, K.N.; Dixit, A.C. Multi-Drug Featurization and Deep Learning Improve Patient-Specific Predictions of Adverse Events. *Int. J. Environ. Res. Public Health* **2021**, *18*, 2600. <https://doi.org/10.3390/ijerph18052600>

Academic Editor: Paul B. Tchounwou

Received: 12 January 2021

Accepted: 3 March 2021

Published: 5 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Clinical trials are used to determine the efficacy and toxicity of medications in humans. Although effective in elucidating various acute responses to the pharmaceutical compound in question, clinical trials are inherently limited by representation bias, size, and duration [1,2]. Current methods of toxicity and drug testing are unable to predict adverse drug reaction (ADR) across diverse populations under conditions of chronic exposure [3,4]. Such ADRs are a significant global health issue that affects millions of people each year with and accounting for an estimated 17% of hospital readmissions [5–8].

One response to this inherent short-coming in predicting and preventing ADRs has been the Tox21 Program. Through collaborative efforts, the U.S. National Institute of Health (NIH), Federal Drug Administration (FDA), and Environmental Protection Agency (EPA) have come together to help promote the evolution of toxicological testing and achieve specific goals that would increase both acute and predictive testing capacities [9]. To increase the ability to understand toxicity effects via data-driven predictions, the program outlines key objectives that address current limitations in identifying rare idiosyncratic responses, characterizing non-genotoxic potential carcinogens, gaining further insight into Adverse Outcome Pathways for risk assessment, and other gaps in testing technology [9].

To fuel the large-scale studies geared towards advancing toxicological and predictive technology, scientists utilize centralized real-world evidence (RWE) databases that contain individual level records of adverse events and associated patient features. Such sources, include the FDA Adverse Event Reporting System (FAERS) dataset which has been

standardizing post-market adverse event reports for over six years and the UK Biobank (UKBB), which links a large set of clinical variables (including medications) longitudinally to genetic information [5,10]. These databases compile relevant information with the intent to improve public health through innovation and discovery [11].

Machine learning may be able to fill the gaps outlined in the Tox21 program by creating models that are predictive, scalable, cost-effective, and adaptable. More specifically, computational methods that have been adapted to biomedical applications, such as toxicity testing, drug responses, and drug discovery, include approaches that leverage Morgan Fingerprints (Morgan FP), Graph Convolutional Networks (GCNs), and Neural Networks (NNs) for Deep Learning applications [12,13].

Such methods utilize structural characteristics of molecules as inputs to computational programs, which in turn, informs in vivo response predictions. Morgan FP, for example, represent key molecular substructures using an explicitly defined featurization. However, a limitation of this specific methodology is its inability to adaptively learn alternative representations that may be more adapted to a particular task [13].

Differing from traditional fingerprint representations, GCNs represent atoms, molecular connectivity, and bond characteristics in a graph-based format. The relationships between neighboring atom-level features that are most informative for a particular task can be learned as the network updates the weights connecting each of the graph convolutional layers [13]. Since featurization and task prediction happen simultaneously in GCN based models, they have significantly higher model complexity and typically require a large dataset in order to outperform traditional machine learning approaches based on traditional fingerprint representations. Alternatively, a pre-training strategy can be used where the model is first trained on a related task (on which a large dataset is available) and then subsequently fine-tuned [14].

Integrative approaches are required to not only enable accurate predictions, but also to address the need for real world applicability. In clinical practice, it is common for those with chronic disease to be on a regimen of multiple medications, which has a positive correlation with the occurrence of ADRs [15,16]. Not only are multiple medications required for patients with comorbidities, but a single illness may also commonly be treated with more than one medication [17,18]. The Center for Disease Control reported that from 2013–2016, in the U.S. alone, 24% of the population reported using three or more medications, whereas 12.6% reported using five or more within the month preceding the survey [15].

While efforts have been made to create machine learning approaches and corresponding databases that capture new ADRs or drug-drug interactions [19], they are limited in their ability to generalize across larger sets. The computational costs and data required to model these higher order interactions scales exponentially with the number of drug interactions (i.e., n , n^2 , n^3 , for single drug, drug-drug, and drug triplet interactions respectively) and as such no existing method can flexibly learn interactions across all medications for patients on multiple medications. Additionally, incorporating other risk factors into the model such as demographic and clinical data can be important in obtaining the most accurate predictions and disentangling confounding risks [2,20,21].

In this paper, we discuss an integrative, precision medicine approach to multi-drug adverse event prediction. The strategy utilized seeks to fill translational gaps in current predictive methodologies.

2. Materials and Methods

2.1. Drug Name to Chemical Structure

Pubchempy (an open source Github repository) was used to convert drug names or active ingredient to isomeric SMILES representation. Simple text filtering for case and punctuation was performed on the drug name input.

2.2. Chemical Structure Featurization of Single Molecules

In our GCN, SMILES structures are featurized using functionality from RDKit (an open source Github repository). Specifically, atom-level features consisting of a one-hot encoding if the atom is either C, N, O, F, P, S, Cl, Br, I, the atomic number, a one-hot coding of chirality, the atoms degree (number of neighboring atoms), formal charge, number of hydrogens, number of radical electrons, a one-hot encoding of hybridization, whether it is or is not aromatic, and whether it is or is not in a ring. Bond-level features including, conjugated status, a one hot encoding of bond type (single, double, triple, aromatic), and a one hot encoding of whether the bond is stereoisomeric. We chose to concatenate the bond-level features to the atom-level features by summing over all bonds directly connected to each atom. This resulted in a feature vector of length 42 for each atom. Finally, we constructed the connectivity matrix between atoms and loaded these features into a Pytorch Geometric Data object.

For cases in which a linear model was to be used as a comparator to the neural network architecture, SMILES structures were featurized into binary feature vectors of length 2048 using Morgan FP with radius 2 using the python package RDKit.

2.3. Extension to Multi-Drug Framework

For our GCN, the atom-level connectivity matrices for each molecule were connected in a block diagonal manner with atom-level and bond-level features being adjoined directly.

For use in the linear model comparisons when patients were taking multiple medications, the maximum value of each element of the Morgan fingerprint across all medications was used as the corresponding featurization for the linear model (again resulting a vector of length 2048).

2.4. Linear/Logistic Regression

For continuous variables, such as predicting hospitalization in the UKBB dataset, a linear regression with an L^2 norm penalty (Ridge regression) was used as a comparator model (sklearn's Ridge module with default parameters).

For discrete variables, such as predicting death or outcome labels in the FAERS dataset a comparable model using sklearn's logistic regression (with default parameters) was used.

2.5. Neural Network/Graph Convolutional Neural Network

For non-drug features, a simple neural network was constructed with the following form:

1. Linear layer transforming the feature vector into a hidden dimension (100 in our model)
2. Rectified linear unit (ReLU) transform
3. Batch normalization
4. Fully connected linear layer transforming hidden dimension to hidden dimension
5. ReLU transform
6. Linear layer transforming hidden dimension to target dimension

For medication associated features, the following architecture was used based roughly on [14,22]:

1. GINConv (graph isomorphism) layer feature vector into a hidden dimension (100 in our model)
 - a. This model performs uses a small neural network to map input atom-features to the output dimension taking into account neighboring atoms
2. Rectified linear unit (ReLU) transform
3. Batch normalization
4. Four additional layers as in 1–3 above

Medication features are aggregated using pooling operators including Set2Set [23], global_max_pool, and global_mean_pool, available in the PyTorch Geometric library.

For the combined model, the outputs of the architectures described above are concatenated for the relevant feature subset.

2.6. Model Evaluation

In all cases, 5-fold cross validation is performed to evaluate model performance.

2.7. Feature Attribution/Importance

To evaluate the importance of individual features within the combined neural network architecture the Integrated Gradients method [24] within the Captum library for PyTorch was used. The sum of all gradients for each feature across patients is used as an estimate of feature importance.

2.8. UK Biobank

UK Biobank (UKBB) contains deep genetic and phenotype information on approximately 500,000 individuals from across the United Kingdom who were aged 40 to 69 at recruitment [10]. Our data were resourced under Application Number 5424. It is available to researchers pending confirmation of their institutional affiliations by their approval committee and payment of any applicable fees.

Medication and health supplements data (Data Field: 20,003) were coded using 6745 categories (Data coding 4) which were mapped to their corresponding active ingredient follow steps similar to those in [11]. This active ingredient was used to obtain SMILES strings and featurization as described above.

Clinical features for each patient were extracted from ICD10 codes (Data Field: 41,202). PCA was performed across all patients to reduce the dimensionality and the scores were extracted as a representation of the “clinical status” of each patient.

A summary of the key subsets of the UKBB dataset that we reference in this work is described in Table 1.

Table 1. Summary of key parameters of UKBB dataset.

Feature	Value
Number of patients selected after filtering	291,560
Average number of medications per patient	3.2
Demographics	Age, Sex, Weight, Height, BMI, and the number of drugs the patient is taking
DNA	Scores from first five genetic PCA components from UKBB—Data-Field 22,009 [10]
Clinical	Scores from first 10 PCA components of ICD10 codes—Data-Field 41,202 (see description above)
Drug structure	For linear model, the maximum of the Morgan Fingerprint is used to featurized multi-drug features, for the GCN, the featurization is flexibly learned during model training
Hospitalization	Log10 (hospitalizations documented + 1) —Data-Field 41,235
Death	Based on Data-Field 40,000

2.9. FDA FAERS

We captured a total of 8,224,912 unique cases in the FAERS database spanning the years between 2014–2020 (through Q3 2020). The data may be readily accessed through the FDA’s online portal. The data were further filtered by the reported role the drug played in the adverse event report, which is characterized by the physician. We only selected drugs that were characterized as primary suspect drugs, secondary suspect drugs,

or suspected interacting drugs (PS, SS, or I in the DRUG file), meaning they may have played an important role in the adverse event. Finally, we performed filtering to remove potentially duplicated entries for cases in which the same combination of sex, weight, age, and medications appeared more than once.

A summary of the key subsets of the FAERS dataset that we reference in this work is described in Table 2.

Table 2. Summary of key parameters of FAERS dataset.

Feature	Value
Number of cases selected (after filtering)	143,412
Average number of medications per case	1.5
Demographics	Age, Sex, Weight, Reporting country
Clinical	Individual presence or absence for the top 200 indications for which drugs were prescribed in the entire FAERS database
Drug structure	For the GCN, the featurization is flexibly learned during model training

3. Results

We sought to construct a machine learning framework that could incorporate vast (but often disparate and filled with missing data elements) RWE databases to predict adverse events (Figure 1). We imposed the requirement that the model be able to flexibly model patients who are on multiple medications without being explicitly constrained to pairwise drug-drug interactions or those previously described.

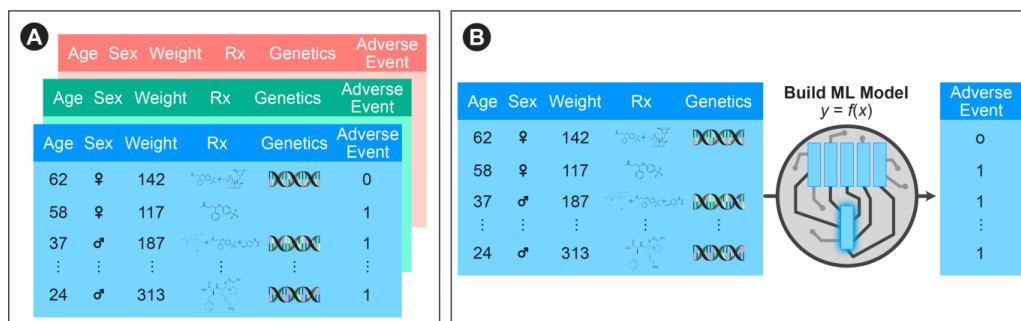


Figure 1. Overview of our approach: (A) Integration of multiple real world evidence databases including demographic, medication, and genetic information; (B) A machine learning model to predict adverse events is constructed.

In order to model the variable number of medications that any patient may be taking, we leverage the graph-based featurization of molecules that GCNs can learn (Figure 2). A single chemical can be represented by the connectivity between atoms, atom-level features, and bond-level features. By concatenating atom level connectivity matrices in a block diagonal format, and simply concatenating atom and bond-level features, multiple molecules can be featurized together. This concatenation represents a collection of disjoint subgraphs. Since there are no connections between different molecules in the connectivity matrix, the GCN operations will not incorporate information across molecules. However, subsequent fully connected operations can learn from the collection of extracted features. With a sufficiently large training dataset, this architecture is able to learn new chemical features and interactions between them that predict multi-drug properties including adverse events.

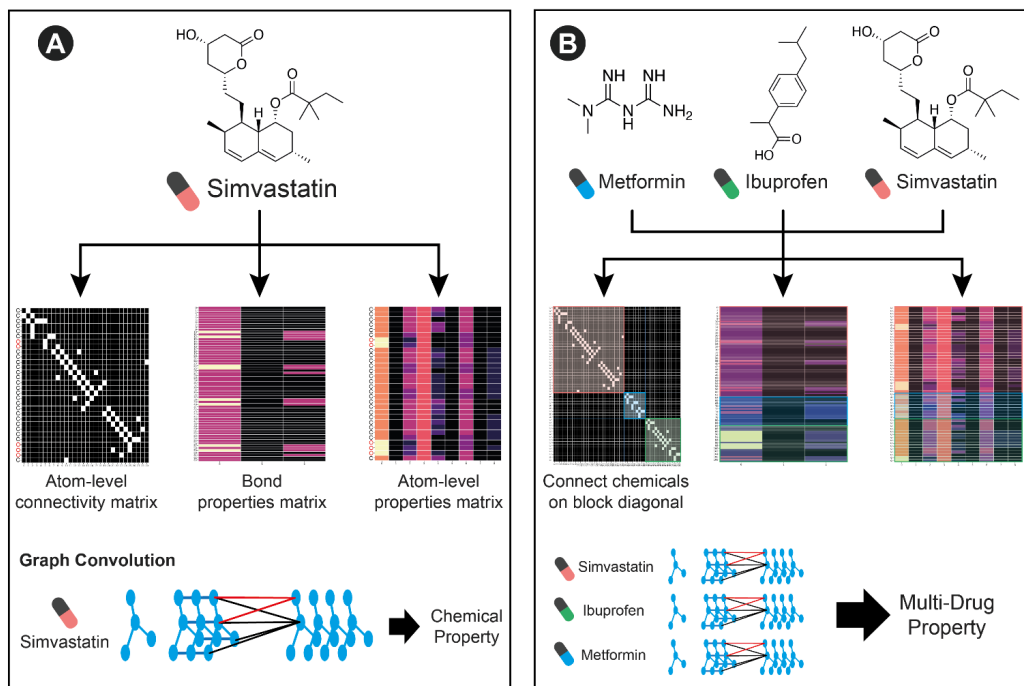


Figure 2. An overview of the multi-drug GCN architecture: (A) A standard GCN applied to a chemical structure creates bond and atom-level features, and an atom-level connectivity matrix to describe the molecule. Graph convolutions are performed to learn new feature representations that learn local structures that can be used to predict chemical properties (B) Our multi-drug GCN architecture concatenates the bond and atom-level features and creates a block diagonal connectivity matrix that represents the set of molecules an individual is taking. In a generalization of the single molecule GCN, the multi-drug GCN aggregates information from local structures across all molecules to predict multi-drug properties. We highlight the featurization of an example patient currently taking simvastatin (red pill), ibuprofen (green pill), and metformin (blue pill).

In order to flexibly model other available individually predictive features (such as age, sex, weight, and genetics), we create separate compact neural networks that learn representations of these features. Finally, the learned representations across each small neural network and the GCN can be combined in a final set of neural network layers to predict the patient-level variable of interest.

3.1. Predicting Adverse Events in the UK Biobank Dataset

In order to test the performance of our framework, we applied it to the UK Biobank dataset on two separate tasks: predicting the number of hospitalizations a patient experienced and predicting whether an individual has died. We were particularly interested in characterizing the relative importance of each of the features and any nonlinear interactions between features. As such we create separate models that contained each of the individual features as well as a combined model containing all features. To benchmark the performance of our approach, we contrast the neural network performance with that of a linear model (which would have limited ability to discern interactions between feature sets) (Figure 3A).

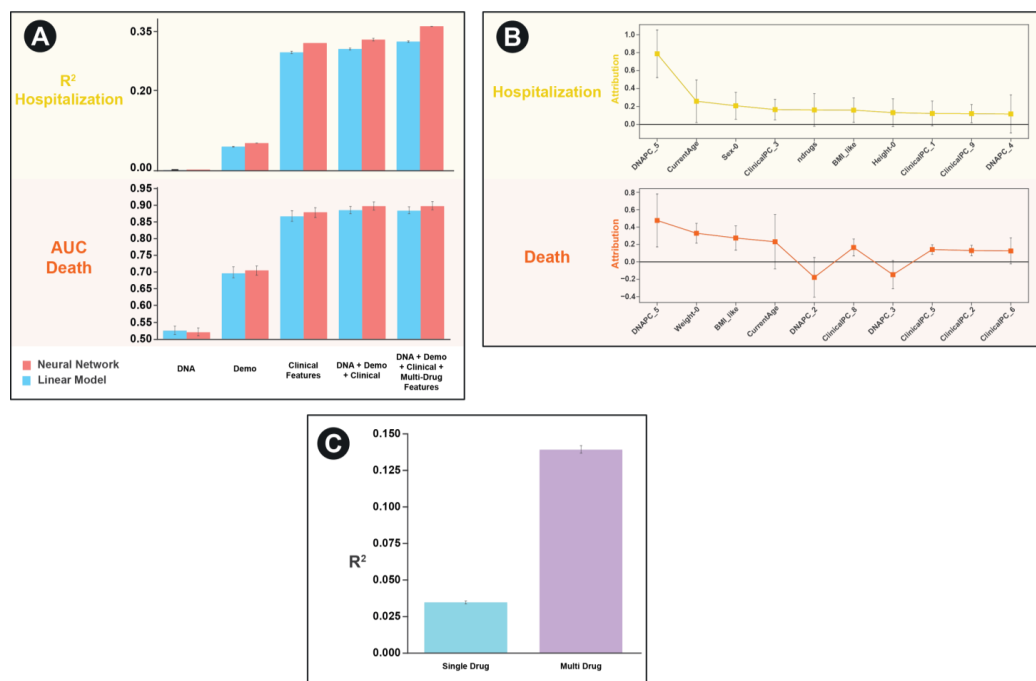


Figure 3. Predictive utility of various features and model architectures for predicting adverse events in the UKBB dataset: (A) Results for hospitalization (top) and death (bottom). Red bar corresponds to neural network architecture, blue bar corresponds to linear model. Y-axis is an R^2 measure of model performance on predicting \log_{10} (hospitalization + 1) (top) and AUC for predicting death (bottom). X-axis contains various combinations of features used in the model. Error bars correspond to 95% confidence interval derived from bootstrapping on 5-fold cross-validation (each fold contains 58,312 patients). (B) Feature weights (attributions) for each the top 10 most important non-drug structure features in the integrated model for predicting hospitalization in the UKBB dataset (top) and death (bottom). Error bars correspond to ± 1 s.d. (C) Bar plot comparing results of using single drug features to using multi-drug features alone for predicting hospitalization. Error bars correspond to 95% confidence interval derived from bootstrapping on 5-fold cross-validation (each fold contains 42,114 patients).

For both the task of predicting hospitalization and death in the UKBB, we found that clinical features (based on PCA scores of ICD10 codes) to be the most predictive individual feature set. For hospitalization, but not death, we find that the neural network architectures significantly outperformed the simple linear model for every feature set except the genetic principal components by themselves. Similarly, for hospitalization, but not death, we find a combined model including multi-drug GCN features significantly improves the predictive performance of the model compared to one without those features (R^2 0.364 vs. 0.331, $p = 0.00004$, Figure 3A).

As a final evaluation of feature importance in the combined model, we use the Integrated Gradients approach to assess contributions of the non-drug features [24]. Surprisingly, we find that despite the fact that the both DNA and demographic features had relatively low predictive performance individually, they were amongst the most predictive features in the combined model (Figure 3B).

To highlight the improvements made possible by our multi-drug framework compared to a single-drug framework, we reevaluated performance on the task of predicting hospitalization for the subset of patients who are on 2 or more medications. We compare the performance of a model that only considers one randomly selected drug per patient to

a model that considers all drugs (Figure 3C). We find a highly significant improvement in model performance (R^2 of 0.035 vs. 0.14, $p < 10^{-10}$).

3.2. Predicting Adverse Events in the FDA FAERS Dataset

We next sought to apply our framework to the FDA FAERS dataset. It contains a larger volume of data and a more targeted set of adverse event labels. Specifically, we attempted to predict the outcomes codes using a similar set of features to those available in the UK Biobank (except for DNA/genetic features which are not available in FAERS).

With the exception of congenital abnormality, which can be significantly predicted with demographic information such as age, the best single feature set for predicting the majority of outcomes was drug structure specific features (Figure 4A). For most categories, an integrated model of demographic, clinical, and drug structure significantly outperformed any of the individual feature set models. These categories included hospitalization ($p < 10^{-5}$) and death ($p < 10^{-5}$).

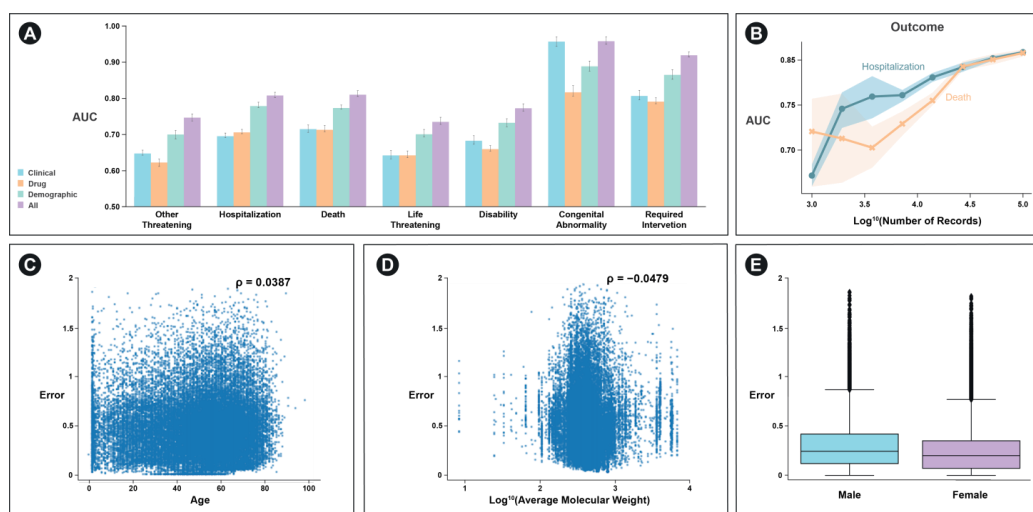


Figure 4. Performance comparisons on the FAERS dataset. (A) Predictive utility of various features and model architectures for predicting adverse events in the FAERS dataset. X-axis labels correspond to adverse event categories for a particular case. Y-axis is the AUC at predicting each of the labels. Colors correspond to various feature subsets tested. Error bars correspond to 95% confidence interval derived from bootstrapping on 5-fold cross-validation (each fold contains 28,682 records). (B) Power analysis demonstrating improvement in performance as a function of the number of patient records examined. Blue corresponds to hospitalization model performance and orange corresponds to performance of model predicting death. X-axis is \log_{10} (number of records) Y-axis is AUC. Shaded error region corresponds to 95% confidence interval derived from bootstrapping on 5-fold cross-validation in a subsampled dataset corresponding to the X-axis location. (C) Plot demonstrating relationship between model error across all outcomes and age, (D) average molecular weight of drugs patient is taking, and (E) patient sex.

We examined the extent to which model performance would be expected to improve through the incorporation of additional data and find that the model would continue to improve for both the prediction of hospitalization and death (Figure 4B). Additionally, we examine the extent to which the model performs better or worse as a function of the covariates we used. We find weak, but significant relationships between age, average molecular weight, and sex and model error ($p < 10^{-6}$) (Figure 4C–E).

4. Discussion

The two datasets analyzed in this paper have contrasting strengths and weaknesses. The UK Biobank includes deep genetic and phenotype information to compare the relative predictive performance of a wide range of well annotated features, but is generally limited to adults 40 years and older in the United Kingdom. It does not have a well-curated a collection of adverse event labels and, as such, surrogate labels such as hospitalization or death were used in this paper. In contrast FDA FAERS collection, is solely focused on adverse events with detailed event labels, however, the database contains selection bias against patients who did not suffer any adverse events on a medication. As such, a predictor built on FDA FAERS will overestimate the likelihood of a particular patient in the general population on a particular combination of medications having an adverse event without performing an additional calibration for how widely those particular medications are prescribed.

There are several promising directions for expanding upon and improving the approach described in this paper. These range from feature expansion for the representation of each atom (including aspects such as drug route of administration and dosage) in each chemical to optimization of the model architecture. One particularly promising area is the incorporation of convolutions which incorporate bond features and the spatial relationships between atoms [25,26].

The Integrated Gradients approach that we used can also be used to increase model interpretability on drug features. Specifically, the relative importance of particular chemical motifs (and interactions between motifs across medications) that drive the prediction a particular individual to experience an adverse event can be visualized [24].

We also note several limitations of the work we present here. Modification of model architecture would likely be required to incorporate and model the impact of biologic therapies. Additionally, for the FAERS dataset, we filtered to around 3% of the overall dataset, this limited dataset may have reduced the ability of the GCN approach to learn improved featurizations. As such, we could either use less strict filtering or pre-train the GCN using other datasets such as the Tox21 Data Challenge or UKBB data sets.

Despite the limited performance of genetic features as standalone predictors of ADRs, we were encouraged by the feature importance of several genetic principal components in the combined model to predict hospitalization in the UKBB dataset. As such we explored using a more comprehensive genetic feature set and developed a companion manuscript, which describes a more thorough variant level prediction of the genetic basis of ADRs across the millions of genetic variants present in the UKBB.

Finally, we highlight two specific examples to illustrate situations in which our model performs poorly and when it performs well. In the first example, we describe the case of a 30-year-old female on Nexplanon who experienced a hospitalization and related life-threatening event that our model failed to predict. We find multiple similar cases of patients on Nexplanon or Nuvaring (implantable birth control medications that the model performed poorly on (there are 659 such cases in our FAERS dataset, and we find they have a 32% higher error than other cases, $p < 10^{-4}$). We hypothesize that this is due to the route of administration not being a component of our model (i.e., pill, infusion, eluting implantable device, etc). In our second example, we highlight the case of a 35-year-old female taking multiple medications who is likely immunocompromised on medications for multiple infections and HIV antiretrovirals whose four adverse events were predicted almost perfectly (difference between actual adverse event outcomes and predicted probabilities was 0.82 out of 7).

5. Conclusions

In this work, we compare the relative predictive utility of demographic, genetic, clinical, multiple drug structures, and the integration of these features to predict adverse outcomes in real world evidence databases including the UKBB and FAERS dataset (Table 3).

Table 3. Summary and support of key findings in each of the two datasets examined.

Attribute	UKBB	FAERS
Neural network outperforms linear model for individual features	For hospitalization across all features except genetic principal components, but not death	For most categories except congenital abnormality, and disability and most models except those only involving clinical features
Combined multi-drug model improves performance relative to other feature sets	For hospitalization, but not death	For both hospitalization and death
Most important single feature	Clinical ICD10 features	Multi-drug features

In the UKBB, we find that in many cases the incorporation of a neural network framework significantly improved predictive performance relative to a standard linear model suggesting the presence of nonlinear interactions between features. We also find that in an integrated model of all features, which outperformed and of the single feature models for hospitalization, demographic and genetic features had significant weights despite not having strong individual level performance.

Similarly, in the FAERS dataset we find that a combined model of demographic, clinical, and multi-drug feature sets is able to outperform any individual feature set for key outcomes like hospitalization and death. This result suggests a role for personalized medicine approaches to predictive toxicology that incorporate patient specific and multi-drug structure features into joint models.

As part of this work we outline and implement a multi-drug GCN framework that is able to flexibly incorporate the variable numbers of medications that real-world patient populations are taking. Built on a deep neural network architecture and deployed on GPU frameworks, it has the potential to rapidly learn complex interactions from growing databases of real-world evidence.

Overall, we believe that these methods will facilitate more accurate predictive personalized toxicology efforts in the future.

Author Contributions: Conceptualization, methodology, software, data analysis, and visualization I.N.A. and A.C.D.; writing—original draft preparation, I.N.A., C.K.H. and A.C.D.; writing—review and editing, A.C.D.; visualization, A.C.D.; supervision, A.C.D. and K.N.D.; project administration, A.C.D. and K.N.D.; clinical significance review, C.K.H. and K.N.D.; funding acquisition, A.C.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by internal funding and the NIH SBIR program, grant number 1R44ES032515-01.

Institutional Review Board Statement: Not applicable. No new human data was collected for this project. Anonymized data was obtained as per UK Biobank policies.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of the UK Biobank dataset and it is available through an application process. The FDA FAERS dataset is publicly available. This data can be found here: <https://www.fda.gov/drugs/questions-and-answers-fdas-adverse-event-reporting-system-faers/fda-adverse-event-reporting-system-faers-latest-quarterly-data-files> (accessed on 14 December 2020).

Acknowledgments: This research has been conducted using the UK Biobank Resourced under Application Number 54241. We thank Tyler Yath for his early conceptualization efforts predicting adverse events from chemical structures using image convolutions.

Conflicts of Interest: A.D. and K.D. are equity holders in Coral Genomics, Inc.

References

- Clark, L.T.; Watkins, L.; Piña, I.L.; Elmer, M.; Akinboboye, O.; Millicent, G.; Jamerson, B.; McCullough, C.; Pierre, C.; Polis, A.B.; et al. Increasing Diversity in Clinical Trials: Overcoming Critical Barriers. *Curr. Probl. Cardiol.* **2019**, *44*, 148–172. [CrossRef]
- Mak, W.W.S.; Law, R.W.; Alvidrez, J.; Pérez-Stable, E.J. Gender and Ethnic Diversity in NIMH-Funded Clinical Trials: Review of a Decade of Published Research. *Adm. Policy Ment. Heal. Ment. Heal. Serv. Res.* **2007**, *34*, 497–503. [CrossRef]
- Ramamoorthy, A.; Pacanowski, M.; Bull, J.; Zhang, L. Racial/Ethnic Differences in Drug Disposition and Response: Review of Recently Approved Drugs. *Clin. Pharmacol. Ther.* **2015**, *97*, 263–273. [CrossRef] [PubMed]
- Ksenia, J.G.; Carvalhoc, N.R.; Chipmand, K.J.; Denslowe, N.D.; Halder, M.; Murphy, C.A.; Roelofs, D.; Rolaki, A.; Schirmer, K.; Watanabek, K.H. Development and Application of the Adverse Outcome Pathway Framework for Understanding and Predicting Chronic Toxicity: I. Challenges and Research Needs in Ecotoxicology. *Chemosphere* **2015**, *120*, 764–777.
- Ngufor, C.; Wojtusiak, J.; Pathak, J. A Systematic Prediction of Adverse Drug Reactions Using Pre-Clinical Drug Characteristics and Spontaneous Reports. *Int. Conf. Healthc. Inform.* **2015**. [CrossRef]
- Center for Disease Control and Prevention, Adverse Drug Events in Adults. Available online: https://www.cdc.gov/medicationsafety/adult_adversedrugevents.html (accessed on 14 December 2020).
- Willson, N.M.; Greer, C.L.; Weeks, D.L. Medication Regimen Complexity and Hospital Readmission for an Adverse Drug Event. *Ann. Pharmacother.* **2014**, *48*, 26–32. [CrossRef] [PubMed]
- White, J.T.; Arakelian, A.; Rho, J.P. Counting the Costs of Drug-Related Adverse Events. *Pharmacoeconomics* **1999**, *15*, 445–458. [CrossRef]
- Administration, U.F.D. FDA Predictive Toxicology Roadmap. Available online: <https://www.fda.gov/media/109634/download> (accessed on 14 December 2020).
- Bycroft, C.; Freeman, C.; Petkova, D.; Band, G.; Lloyd, T.E.; Sharp, K.; Motyer, A.; Vukcevic, D.; Delaneau, O.; O’Connell, J.; et al. The UK Biobank Resource with Deep Phenotyping and Genomic Data. *Nature* **2018**, *562*, 203–209. [CrossRef]
- Wu, Y.; Byrne, E.M.; Zheng, Z.; Kemper, K.E.; Yengo, L.; Mallett, A.J.; Yang, J.; Visscher, P.M.; Genome, W.N.R. Wide Association Study of Medication-Use and Associated Disease in the UK Biobank. *Nat. Commun.* **2019**, *10*, 1891. [CrossRef]
- Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction Using Deep Learning. *Front. Environ. Sci.* **2016**, *3*, 80. [CrossRef]
- Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving Beyond Fingerprints. *J. Comput. Aided Mol. Des.* **2016**, *30*, 595–608. [CrossRef] [PubMed]
- Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for Pre-Training Graph Neural Networks. *arXiv* **2019**, arXiv:1905.12265v3. Available online: <https://arxiv.org/abs/1905.12265> (accessed on 14 December 2020).
- Therapeutic Drug Use. Center for Disease Control. 2020. Available online: <https://www.cdc.gov/nchs/fastats/drug-use-therapeutic.htm> (accessed on 14 December 2020).
- Sommer, J.; Seeling, A.; Rupprecht, H. Adverse Drug Events in Patients with Chronic Kidney Disease Associated with Multiple Drug Interactions and Polypharmacy. *Drugs Aging* **2020**, *37*, 359–372. [CrossRef] [PubMed]
- Libby, A.M.; Fish, D.N.; Hosokawa, P.W.; Linnebur, S.A.; Metz, R.K.; Nair, K.V.; Saseen, J.J.; Griend, J.P.V.; Vu, S.P.; Hirsch, J.D. Patient-Level Medication Regimen Complexity Across Populations with Chronic Disease. *Clin. Ther.* **2013**, *35*, 385–398. [CrossRef] [PubMed]
- Sun, J.A.; Li, S.; Zhang, A.C.; Jensen, K.T.; Lindahl-Jacobsen, R.; Eisenberg, M.L. Parental Comorbidity and Medication Use in the USA: A Panel Study of 785 000 Live Births. *Hum. Reprod.* **2020**, *35*, 669–675. [CrossRef] [PubMed]
- Tatonetti, P.N.; Ye, P.P.; Daneshjou, R.; Altman, R.B. Data-Driven Prediction of Drug Effects and Interactions. *Sci. Transl. Med.* **2012**, *4*. [CrossRef]
- Alomar, M.J. Factors Affecting the Development of Adverse Drug Reactions (Review Article). *Saudi Pharm. J.* **2014**, *22*, 83–94. [CrossRef]
- Hajjar, E.R.; Hanlon, J.T.; Artz, M.B.; Lindblad, C.I.; Pieper, C.F.; Sloane, R.J.; Ruby, C.M.; Schmader, K.E. Adverse Drug Reaction Risk Factors in Older Outpatients. *Am. J. Geriatr. Pharmacother.* **2003**, *1*, 82–89. [CrossRef]
- Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? *arXiv* **2018**, arXiv:1810.00826. Available online: <https://arxiv.org/abs/1810.00826> (accessed on 14 December 2020).
- Vinyals, O.; Bengio, S.; Kudlur, M. Order Matters: Sequence to Sequence for Sets. *arXiv* **2015**, arXiv:1511.06391. Available online: <https://arxiv.org/abs/1511.06391> (accessed on 14 December 2020).
- Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017. Available online: <http://proceedings.mlr.press/v70/sundararajan17a.html> (accessed on 14 December 2020).
- Simonovsky, M. Komodakis Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3693–3702.
- Feinberg, E.N.; Sur, D.; Wu, Z.; Husic, B.E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V.S. PotentialNet for Molecular Property Prediction. *ACS Cent. Sci.* **2018**, *4*, 1520–1530. [CrossRef] [PubMed]

References

- [1] Sarah A Dugger, Adam Platt, and David B Goldstein. Drug development in the era of precision medicine. *Nat. Rev. Drug Discov.*, 17(3):183–196, March 2018.
- [2] Alex Zhavoronkov, Yan A Ivanenkov, Alex Aliper, Mark S Veselov, Vladimir A Aladinskiy, Anastasiya V Aladinskaya, Victor A Terentiev, Daniil A Polykovskiy, Maksim D Kuznetsov, Arip Asadulaev, Yury Volkov, Artem Zholus, Rim R Shayakhmetov, Alexander Zhebrak, Lidiya I Minaeva, Bogdan A Zagribelnyy, Lennart H Lee, Richard Soll, David Madge, Li Xing, Tao Guo, and Alán Aspuru-Guzik. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nat. Biotechnol.*, 37(9):1038–1040, September 2019.
- [3] Richard C Mohs and Nigel H Greig. Drug discovery and development: Role of basic biological research. *Alzheimers. Dement.*, 3(4):651–657, November 2017.
- [4] Steven M Paul, Daniel S Mytelka, Christopher T Dunwiddie, Charles C Persinger, Bernard H Munos, Stacy R Lindborg, and Aaron L Schacht. How to improve R&D productivity: the pharmaceutical industry’s grand challenge. *Nat. Rev. Drug Discov.*, 9(3):203–214, March 2010.
- [5] Jerry Avorn. The \$2.6 billion pill — methodologic and policy considerations, 2015.
- [6] Clayton E Friedman, Quan Nguyen, Samuel W Lukowski, Abbigail Helfer, Han Sheng Chiu, Jason Miklas, Shiri Levy, Shengbao Suo, Jing-Dong Jackie Han, Pierre Osteil, Guangdun Peng, Naihe Jing, Greg J Baillie, Anne Senabouth, Angelika N Christ, Timothy J Bruxner, Charles E Murry, Emily S Wong, Jun Ding, Yuliang Wang, James Hudson, Hannele Ruohola-Baker, Ziv Bar-Joseph, Patrick P L Tam, Joseph E Powell, and Nathan J Palpant. Single-Cell transcriptomic analysis of cardiac differentiation from human PSCs reveals HOPX-Dependent cardiomyocyte maturation. *Cell Stem Cell*, 23(4):586–598.e8, October 2018.

- [7] Helena Viita, Agnieszka Pacholska, Farizan Ahmad, Johanna Tietäväinen, Jonne Naarala, Anna Hyvärinen, Thomas Wirth, and Seppo Ylä-Herttuala. 15-lipoxygenase-1 induces lipid peroxidation and apoptosis, and improves survival in rat malignant glioma. *In Vivo*, 26(1):1–8, January 2012.
- [8] Z Lindgardt, M Reeves, and J Wallenstein. Waking the giant: business model innovation in the drug industry. *IN VIVO-NEW SERIES-*, 26(6):54, 2008.
- [9] Polina Mamoshina, Armando Vieira, Evgeny Putin, and Alex Zhavoronkov. Applications of deep learning in biomedicine. *Mol. Pharm.*, 13(5):1445–1454, May 2016.
- [10] Markus Schirle and Jeremy L Jenkins. Identifying compound efficacy targets in phenotypic drug discovery, 2016.
- [11] Jingshu Wang, Divyansh Agarwal, Mo Huang, Gang Hu, Zilu Zhou, Vincent B Conley, Hugh MacMullan, and Nancy R Zhang. Transfer learning in single-cell transcriptomics improves data denoising and pattern discovery. November 2018.
- [12] Youjun Xu, Ziwei Dai, Fangjin Chen, Shuaishi Gao, Jianfeng Pei, and Luhua Lai. Deep learning for Drug-Induced liver injury. *J. Chem. Inf. Model.*, 55(10):2085–2093, October 2015.
- [13] Tyler B Hughes, Grover P Miller, and S Joshua Swamidass. Modeling epoxidation of drug-like molecules with a deep machine learning network. *ACS Cent Sci*, 1(4):168–180, July 2015.
- [14] Daniel C Liebler and F Peter Guengerich. Elucidating mechanisms of drug-induced toxicity. *Nat. Rev. Drug Discov.*, 4(5):410–420, May 2005.
- [15] Nicholas P Tatonetti, Patrick P Ye, Roxana Daneshjou, and Russ B Altman. Data-driven prediction of drug effects and interactions. *Sci. Transl. Med.*, 4(125):125ra31, March 2012.

- [16] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, July 2018.
- [17] Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, Viviana Consonni, Victor E Kuz'min, Richard Cramer, Romualdo Benigni, Chihae Yang, James Rathman, Lothar Terfloth, Johann Gasteiger, Ann Richard, and Alexander Tropsha. QSAR modeling: where have you been? where are you going to? *J. Med. Chem.*, 57(12):4977–5010, June 2014.
- [18] Erik Gawehn, Jan A Hiss, and Gisbert Schneider. Deep learning in drug discovery. *Mol. Inform.*, 35(1):3–14, January 2016.
- [19] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, May 2010.
- [20] Andrea Mauri, Viviana Consonni, Manuela Pavan, and Roberto Todeschini. Dragon software: An easy approach to molecular descriptor calculations. *Match*, 56(2):237–248, 2006.
- [21] Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The chemistry development kit (CDK): An Open-Source java library for chemo- and bioinformatics, 2003.
- [22] A A Toropov, A P Toropova, and E Benfenati. QSPR modeling of octanol water partition coefficient of platinum complexes by InChI-based optimal descriptors. *J. Math. Chem.*, 46(4):1060–1073, November 2009.
- [23] D-S Cao, J-C Zhao, Y-N Yang, C-X Zhao, J Yan, S Liu, Q-N Hu, Q-S Xu, and Y-Z Liang. In silico toxicity prediction by support vector machine and SMILES representation-based string kernel. *SAR QSAR Environ. Res.*, 23(1-2):141–153, January 2012.

- [24] Thomas N Kipf and Max Welling. Semi-Supervised classification with graph convolutional networks. September 2016.
- [25] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.
- [26] Thomas N Kipf and Max Welling. Variational graph Auto-Encoders. November 2016.
- [27] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. February 2018.
- [28] Andrew Goodspeed, Laura M Heiser, Joe W Gray, and James C Costello. Tumor-Derived cell lines as molecular models of cancer pharmacogenomics. *Mol. Cancer Res.*, 14(1):3–13, January 2016.
- [29] D Mouradov, C Sloggett, R N Jorissen, C G Love, S Li, A W Burgess, D Arango, R L Strausberg, D Buchanan, S Wormald, L O’Connor, J L Wilding, D Bicknell, I P M Tomlinson, W F Bodmer, J M Mariadason, and O M Sieber. Colorectal cancer cell lines are representative models of the main molecular subtypes of primary cancer, 2014.
- [30] Anneleen Daemen, Obi L Griffith, Laura M Heiser, Nicholas J Wang, Oana M Enache, Zachary Sanborn, Francois Pepin, Steffen Durinck, James E Korkola, Malachi Griffith, Joe S Hur, Nam Huh, Jongsuk Chung, Leslie Cope, Mary Jo Fackler, Christopher Umbricht, Saraswati Sukumar, Pankaj Seth, Vikas P Sukhatme, Lakshmi R Jakkula, Yiling Lu, Gordon B Mills, Raymond J Cho, Eric A Collisson, Laura J van’t Veer, Paul T Spellman, and Joe W Gray. Modeling precision treatment of breast cancer. *Genome Biol.*, 14(10):R110, 2013.
- [31] Robert H Shoemaker. The NCI60 human tumour cell line anticancer drug screen. *Nat. Rev. Cancer*, 6(10):813–823, October 2006.

- [32] Uma T Shankavaram, Sudhir Varma, David Kane, Margot Sunshine, Krishna K Chary, William C Reinhold, Yves Pommier, and John N Weinstein. CellMiner: a relational database and query tool for the NCI-60 cancer cell lines. *BMC Genomics*, 10:277, June 2009.
- [33] Laura M Heiser, Anguraj Sadanandam, Wen-Lin Kuo, Stephen C Benz, Theodore C Goldstein, Sam Ng, William J Gibb, Nicholas J Wang, Safiyah Ziyad, Frances Tong, Nora Bayani, Zhi Hu, Jessica I Billig, Andrea Dueregger, Sophia Lewis, Lakshmi Jakkula, James E Korkola, Steffen Durinck, François Pepin, Yinghui Guan, Elizabeth Purdom, Pierre Neuvial, Henrik Bengtsson, Kenneth W Wood, Peter G Smith, Lyubomir T Vassilev, Bryan T Hennessy, Joel Greshock, Kurtis E Bachman, Mary Ann Hardwicke, John W Park, Laurence J Marton, Denise M Wolf, Eric A Collisson, Richard M Neve, Gordon B Mills, Terence P Speed, Heidi S Feiler, Richard F Wooster, David Haussler, Joshua M Stuart, Joe W Gray, and Paul T Spellman. Subtype and pathway specific responses to anticancer compounds in breast cancer. *Proc. Natl. Acad. Sci. U. S. A.*, 109(8):2724–2729, February 2012.
- [34] Justin Lamb, Emily D Crawford, David Peck, Joshua W Modell, Irene C Blat, Matthew J Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N Ross, Michael Reich, Haley Hieronymus, Guo Wei, Scott A Armstrong, Stephen J Haggarty, Paul A Clemons, Ru Wei, Steven A Carr, Eric S Lander, and Todd R Golub. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–1935, September 2006.
- [35] Wanjuan Yang, Jorge Soares, Patricia Greninger, Elena J Edelman, Howard Lightfoot, Simon Forbes, Nidhi Bindal, Dave Beare, James A Smith, I Richard Thompson, Sridhar Ramaswamy, P Andrew Futreal, Daniel A Haber, Michael R Stratton, Cyril Benes, Ultan McDermott, and Mathew J Garnett. Genomics of drug sensitivity in cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res.*, 41(Database

issue):D955–61, January 2013.

- [36] Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, Anupama Reddy, Manway Liu, Lauren Murray, Michael F Berger, John E Monahan, Paula Morais, Jodi Meltzer, Adam Korejwa, Judit Jané-Valbuena, Felipa A Mapa, Joseph Thibault, Eva Bric-Furlong, Pichai Raman, Aaron Shipway, Ingo H Engels, Jill Cheng, Guoying K Yu, Jianjun Yu, Peter Aspesi, Jr, Melanie de Silva, Kalpana Jagtap, Michael D Jones, Li Wang, Charles Hatton, Emanuele Palesscandolo, Supriya Gupta, Scott Mahan, Carrie Sougnez, Robert C Onofrio, Ted Liefeld, Laura MacConaill, Wendy Winckler, Michael Reich, Nanxin Li, Jill P Mesirov, Stacey B Gabriel, Gad Getz, Kristin Ardlie, Vivien Chan, Vic E Myer, Barbara L Weber, Jeff Porter, Markus Warmuth, Peter Finan, Jennifer L Harris, Matthew Meyerson, Todd R Golub, Michael P Morrissey, William R Sellers, Robert Schlegel, and Levi A Garraway. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, March 2012.
- [37] Amrita Basu, Nicole E Bodycombe, Jaime H Cheah, Edmund V Price, Ke Liu, Giannina I Schaefer, Richard Y Ebright, Michelle L Stewart, Daisuke Ito, Stephanie Wang, Abigail L Bracha, Ted Liefeld, Mathias Wawer, Joshua C Gilbert, Andrew J Wilson, Nicolas Stransky, Gregory V Kryukov, Vlado Dancik, Jordi Barretina, Levi A Garraway, C Suk-Yee Hon, Benito Munoz, Joshua A Bittker, Brent R Stockwell, Dineo Khabele, Andrew M Stern, Paul A Clemons, Alykhan F Shamji, and Stuart L Schreiber. An interactive resource to identify cancer genetic and lineage dependencies targeted by small molecules. *Cell*, 154(5):1151–1161, August 2013.
- [38] Mahmoud Ghandi, Franklin W Huang, Judit Jané-Valbuena, Gregory V Kryukov, Christopher C Lo, E Robert McDonald, 3rd, Jordi Barretina, Ellen T Gelfand, Craig M Bielski, Haoxin Li, Kevin Hu, Alexander Y Andreev-Drakhlin, Jaegil Kim, Julian M Hess, Brian J

Haas, François Aguet, Barbara A Weir, Michael V Rothberg, Brenton R Paoletta, Michael S Lawrence, Rehan Akbani, Yiling Lu, Hong L Tiv, Prafulla C Gokhale, Antoine de Weck, Ali Amin Mansour, Coyin Oh, Juliann Shih, Kevin Hadi, Yanay Rosen, Jonathan Bistline, Kavitha Venkatesan, Anupama Reddy, Dmitriy Sonkin, Manway Liu, Joseph Lehar, Joshua M Korn, Dale A Porter, Michael D Jones, Javad Golji, Giordano Caponigro, Jordan E Taylor, Caitlin M Dunning, Amanda L Creech, Allison C Warren, James M McFarland, Mahdi Zamanighomi, Audrey Kauffmann, Nicolas Stransky, Marcin Imielinski, Yosef E Maruvka, Andrew D Cherniack, Aviad Tsherniak, Francisca Vazquez, Jacob D Jaffe, Andrew A Lane, David M Weinstock, Cory M Johannessen, Michael P Morrissey, Frank Stegmeier, Robert Schlegel, William C Hahn, Gad Getz, Gordon B Mills, Jesse S Boehm, Todd R Golub, Levi A Garraway, and William R Sellers. Next-generation characterization of the cancer cell line encyclopedia. *Nature*, 569(7757):503–508, May 2019.

- [39] Daniela Ferreira, Filomena Adegas, and Raquel Chaves. The importance of cancer cell lines as in vitro models in cancer methylome analysis and anticancer drugs testing. In Cesar Lopez-Camarillo, editor, *Oncogenomics and Cancer Proteomics - Novel Approaches in Biomarkers Discovery and Therapeutic Targets in Cancer*. InTech, March 2013.
- [40] Michael P Menden, Dennis Wang, Mike J Mason, Bence Szalai, Krishna C Bulusu, Yuanfang Guan, Thomas Yu, Jaewoo Kang, Minji Jeon, Russ Wolfinger, Tin Nguyen, Mikhail Zaslavskiy, AstraZeneca-Sanger Drug Combination DREAM Consortium, In Sock Jang, Zara Ghazoui, Mehmet Eren Ahsen, Robert Vogel, Elias Chaibub Neto, Thea Norman, Eric K Y Tang, Mathew J Garnett, Giovanni Y Di Veroli, Stephen Fawell, Gustavo Stolovitzky, Justin Guinney, Jonathan R Dry, and Julio Saez-Rodriguez. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nat. Commun.*, 10(1):2674, June 2019.
- [41] Yoosup Chang, Hyejin Park, Hyun-Jin Yang, Seungju Lee, Kwee-Yum Lee, Tae Soon Kim,

- Jongsun Jung, and Jae-Min Shin. Cancer drug response profile scan (CDRscan): A deep learning model that predicts drug effectiveness from cancer genomic signature. *Sci. Rep.*, 8(1):8857, June 2018.
- [42] Yu-Chiao Chiu, Hung-I Harry Chen, Tinghe Zhang, Songyao Zhang, Aparna Gorthi, Li-Ju Wang, Yufei Huang, and Yidong Chen. Predicting drug response of tumors from integrated genomic profiles by deep neural networks.
- [43] Bin Chen, Marina Sirota, Hua Fan-Minogue, Dexter Hadley, and Atul J Butte. Relating hepatocellular carcinoma tumor samples and cell lines using gene expression data in translational research, 2015.
- [44] Krista Marie Vincent, Scott D Findlay, and Lynne Marie Postovit. Assessing breast cancer cell lines as tumour models by comparison of mRNA expression profiles. *Breast Cancer Res.*, 17:114, August 2015.
- [45] K Yu, B Chen, D Aran, J Charalel, C Yau, D M Wolf, L J van 't Veer, A J Butte, T Goldstein, and M Sirota. Comprehensive transcriptomic analysis of cell lines as models of primary tumors across 22 tumor types, 2019.
- [46] Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nat. Methods*, 15(4):290–298, April 2018.
- [47] David Deamer, Mark Akeson, and Daniel Branton. Three decades of nanopore sequencing. *Nat. Biotechnol.*, 34(5):518–524, May 2016.
- [48] Fumihito Miura, Yusuke Enomoto, Ryo Dairiki, and Takashi Ito. Amplification-free whole-genome bisulfite sequencing by post-bisulfite adaptor tagging. *Nucleic Acids Res.*, 40(17):e136, September 2012.

- [49] Gilad Landan, Netta Mendelson Cohen, Zohar Mukamel, Amir Bar, Alina Molchadsky, Ran Brosh, Shirley Horn-Saban, Daniela Amann Zalcenstein, Naomi Goldfinger, Adi Zundelevich, Einav Nili Gal-Yam, Varda Rotter, and Amos Tanay. Epigenetic polymorphism and the stochastic formation of differentially methylated regions in normal and cancerous tissues. *Nat. Genet.*, 44(11):1207–1214, November 2012.
- [50] Andrew H Laszlo, Ian M Derrington, Henry Brinkerhoff, Kyle W Langford, Ian C Nova, Jenny Mae Samson, Joshua J Bartlett, Mikhail Pavlenok, and Jens H Gundlach. Detection and mapping of 5-methylcytosine and 5-hydroxymethylcytosine with nanopore MspA. *Proc. Natl. Acad. Sci. U. S. A.*, 110(47):18904–18909, November 2013.
- [51] Jacob Schreiber, Zachary L Wescoe, Robin Abu-Shumays, John T Vivian, Baldandorj Baatar, Kevin Karplus, and Mark Akeson. Error rates for nanopore discrimination among cytosine, methylcytosine, and hydroxymethylcytosine along individual DNA strands. *Proc. Natl. Acad. Sci. U. S. A.*, 110(47):18910–18915, November 2013.
- [52] Ankur Jai Sood, Coby Viner, and Michael M Hoffman. DNAmoD: the DNA modification database. *J. Cheminform.*, 11(1):30, April 2019.
- [53] Pietro Boccaletto, Magdalena A Machnicka, Elzbieta Purta, Pawel Piatkowski, Blazej Baginski, Tomasz K Wirecki, Valérie de Crécy-Lagard, Robert Ross, Patrick A Limbach, Annika Kotter, Mark Helm, and Janusz M Bujnicki. MODOMICS: a database of RNA modification pathways. 2017 update. *Nucleic Acids Res.*, 46(D1):D303–D307, January 2018.
- [54] Giorgio Sirugo, Sarah A Tishkoff, and Scott M Williams. The quagmire of race, genetic ancestry, and health disparities. *J. Clin. Invest.*, 131(11), June 2021.