Signals in the Cells: Multimodal and Contextualized Machine Learning Foundations for Therapeutics

Alejandro Velez-Arce*
Department of Biomedical Informatics
Harvard Medical School
Boston, MA 02115
alejandro_velez-arce@hms.harvard.edu

Xiang Lin
Department of Biomedical Informatics
Harvard Medical School
Boston, MA 02115
xiang_lin@hms.harvard.edu

Michelle M. Li
Department of Biomedical Informatics
Harvard Medical School
Boston, MA 02115
michelleli@g.harvard.edu

Kexin Huang
Department of Computer Science
Stanford School of Engineering
Stanford, CA 94305
kexinh@stanford.edu

Wenhao Gao
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
whgao@mit.edu

Tianfan Fu
Department of Computational Science
Rensselaer Polytechnic Institute
Troy, NY 12180
futianfan@gmail.com

Bradley L. Pentelute
Department of Chemistry
Massachusetts Institute of Technology
Cambridge, MA 02139
blp@mit.edu

Manolis Kellis
Broad Institute of MIT and Harvard
Computer Science and Artificial Intelligence Laboratory, MIT
Electrical Engineering and Computer Science Department
Massachusetts Institute of Technology
Cambridge, MA 02139
manoli@mit.edu

Marinka Zitnik*
Broad Institute of MIT and Harvard
Harvard Data Science Initiative
Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University
Department of Biomedical Informatics
Harvard Medical School
Boston, MA 02215
marinka@hms.harvard.edu

Abstract

*Co-corresponding Author

AI for New Drug Modalities at NeurIPS 2024.

Drug discovery AI datasets and benchmarks have not traditionally included single-cell analysis biomarkers. While benchmarking efforts in single-cell analysis have recently released collections of single-cell tasks, they have yet to comprehensively release datasets, models, and benchmarks that integrate a broad range of therapeutic discovery tasks with cell-type-specific biomarkers. Therapeutics Commons (TDC-2) presents datasets, tools, models, and benchmarks integrating cell-type-specific contextual features with ML tasks across therapeutics. We present four tasks for contextual learning at single-cell resolution: drug-target nomination, genetic perturbation response prediction, chemical perturbation response prediction, and protein-peptide interaction prediction. We introduce datasets, models, and benchmarks for these four tasks. Finally, we detail the advancements and challenges in machine learning and biology that drove the implementation of TDC-2 and how they are reflected in its architecture, datasets and benchmarks, and foundation model tooling.

1 Introduction

Single-cell genomics has enabled the study of cellular processes with remarkable resolution, offering insights into cellular heterogeneity and dynamics [1]. Progress in data generation and computational methods designed for single-cell analysis [2] has facilitated machine learning models that incorporate cell-type-specific data across various therapeutic areas [3, 4]. Despite these advances, there remains a need for comprehensive datasets, benchmarks, and tools that integrate single-cell analysis with diverse therapeutic approaches.

Out-of-distribution (OOD) generalization [5] and incorporation of novel tools [6] and modalities [4, 7] remain as challenges for biomedical machine learning models across a broad range of tasks. Models capable of accurate OOD predictions promise to expand to the vast molecular space, whose size is estimated at 10^{60} potential drug-like molecules [8], yet less than 10^5 of those are FDA-approved drugs [9], suggesting the potential for advanced computational methods to navigate the molecular space and help find, generate, and optimize candidate drugs. Further, handling multimodal data is essential for building foundation models that accurately capture the complex interactions within biological systems [10], which is vital for understanding disease mechanisms and discovering effective treatments. These challenges are compounded by the lack of unified datasets organized across stages of drug discovery. Therapeutics Data Commons [11, 12] addresses these challenges by providing a unified platform that consolidates therapeutic datasets and benchmarks. Still, benchmarks tailored to measuring the effectiveness of models at OOD predictions are rare for several key biological tasks [13]. Most dataset and benchmark providers also struggle to evaluate models using longitudinal data [14] and real-world evidence [15] due to challenges in continual data collection [16].

Public benchmarks and competitions that measure performance using state-of-the-art methods against standardized criteria have a strong track record of accelerating innovation in algorithm development in therapeutic science [17]. However, machine learning researchers face several challenges in this domain, including: (1) a lack of domain knowledge regarding essential tasks in the field, (2) the absence of standard benchmarks for different methods due to their varying implementations [17], and (3) the high cost of implementing complex data pre-processing pipelines for each task [11]. Despite the progress made over the last five years in developing datasets and benchmarks for machine learning methods in therapeutics [18, 12], decentralized and standardized benchmarks are still needed in single-cell therapeutics. Similarly, recent advancements in single-cell analysis tools, datasets, and benchmarks [1, 2, 19] have yet to address therapeutic tasks.

Further, there is demand for retrieval tooling that can integrate into emerging tool-based LLMs [20, 21]. [21] shows that integrating retrieval APIs with LLMs mitigates the issue of hallucination, commonly encountered when prompting LLMs directly. They also discuss the challenges of supporting a web scale collection of millions of changing APIs. With the advent of public petabyte-scale public-API-accessible databases in biomedical informatics [2], it is imperative to build systems providing unified retrieval across biomedical modalities and large-scale, changing, data sources.

Present work. The Commons 2.0, a.k.a. TDC-2, introduces a multi-modal retrieval API implementing a novel API-first-dataset architecture. The API-first-dataset architecture supports retrieval of continually updated heterogenous data sources (i.e., knowledge graphs [22], embedding models [19],

petabyte-scale RNA data [2], etc.) across biomedical modalities providing abstractions for complex data processing pipelines [11], data lineage [23], and versioning [24]. This lays the foundation for improving the stability of biomedical AI workflows with continuous data updates [25]. We have developed these resources building on the Therapeutic Data Commons platform, and leverage them to present four novel ML tasks with fine-grained biological contexts: single-cell drug-target identification [4], single-cell chemical/genetic perturbation response predictions [26, 27], and a cell-type-specific protein-peptide interaction task [28, 15] (Section 3.3.1). The models, datasets, and benchmarks composing these tasks address cell-type-specific molecule ML modeling [4], evaluation of contextual AI models [29, 4], heuristics for generating negative samples in peptidomimetics [30], and OOD generalization in single-cell perturbation response prediction [31, 32]. Overall, the contributions are (summarized in Table 5):

1. TDC-2 is the first platform to integrate single-cell analysis with multimodal machine learning in drug discovery via four contextual AI tasks along with corresponding models, datasets, and benchmarks (Section 3).
2. TDC-2 formalizes contextualized metrics for evaluating contextual AI models on therapeutic tasks. This allows for evaluating models' abilities in identifying the most predictive cell type contexts (Section 4).
3. TDC-2 introduces an API-first-dataset architecture (Section 6.3.1) for augmenting therapeutic models with retrieval APIs, improving workflow stability with continual data collection.
4. An unprecedented collection of heterogeneous data sources is unified under the API-first-dataset architecture (section 6.3.2). These include: a petabyte-scale single-cell RNA data atlas [2], a framework for biomedical knowledge graphs [22], and a collection of retrieval APIs integrated with complex data processing workflows (Section 6.3.2; see listing 2 for example usage).

2 Related Work

Machine learning datasets and benchmarks in therapeutics. Therapeutics Data Commons (TDC) was the first unifying platform providing systematic access and evaluation for machine learning across the entire range of therapeutics [11]. TDC included 66 AI-ready datasets and 22 learning tasks, spanning the discovery and development of safe and effective medicines. TDC also provided an ecosystem of tools and community resources, including 33 data functions and types of meaningful data splits, 23 strategies for systematic model evaluation, 17 molecule generation oracles, and 29 public leaderboards. TDC-2 augments the biomedical modalities covered by TDC data, tasks, and benchmarks to lay the foundations for building and evaluating foundation models. We expanded the biomedical modalities covered by TDC, introduced single-cell resolution to various modalities, introduced access to model embeddings, introduced machine learning model retrieval APIs for inference and fine-tuning, and incorporated contextualized metrics into model evaluation to lay the foundations for building and evaluating single-cell therapeutic foundation models. We further developed an API-first dataset design (Section 6.3.1) unifying modalities [33] for augmenting LLMs with biomedical experimental data retrieval [6]. TDC-2 distinguishes itself from related datasets [34, 35], benchmarks [17, 18, 36, 37], model development frameworks [38, 39, 40], and therapeutic initiatives [2] in its integration of single-cell analysis with multimodal machine learning in drug discovery via four contextual AI tasks and retrieval APIs for multimodal datasets and models.

Therapeutic and single-cell foundation models. Foundation models trained on TDC, and a subset of tasks in TDC-2 (Sections 7.2.3, 7.2.4, and 7.2.5), have been shown to generalize across several therapeutic tasks [41]. Additionally, parallel efforts in training foundation models on large single-cell atlases have shown a potential to advance cell type annotation and matching of healthy-disease cells to study cellular signatures of disease [3, 42, 43, 44]. TDC-2 bridges these independent and parallel efforts by providing formal definitions of therapeutic tasks, datasets, and benchmarks at single-cell resolution in order to provide precise therapeutic predictions incorporating cell type contexts [29, 4].

Augmenting biomedical AI workflows with multi-modal tooling. Recent advancements in tool-based large language models (LLMs) showcase the potential of augmenting these systems to call external functions and APIs [20, 21]. By integrating chemistry tooling, LLMs can be augmented with novel capabilities across chemical tasks [6]. LLM multi-agent frameworks have also succeeded at manipulating collections of tools for the automatic processing and execution of single-cell analysis tasks [45]. The API-first [46, 47, 48] approach adopted by TDC-2's multimodal retrieval API,

dubbed the API-first dataset, integrates expert-designed tools with continually updated data to support grounding of biomedical AI workflows.

3 Results

TDC-2 introduces four tasks with fine-grained biological contexts: contextualized drug-target identification, single-cell chemical/genetic perturbation response prediction, and cell-type-specific protein-peptide binding interaction prediction, which introduce antigen-processing-pathway-specific, cell-type-specific, peptide-specific, and patient-specific biological contexts. Benchmarks for drug target nomination, genetic perturbation response prediction and chemical perturbation response prediction, all at single-cell resolution, were computed with corresponding leaderboards introduced on the TDC website. In addition, a benchmark and leaderboard was introduced for the TCR-Epitope binding interaction task for peptide design at single-cell resolution for T-cell receptors.

Context-specific metrics. In real-world machine learning applications, data subsets can correspond to critical outcomes. In therapeutics, there is evidence that the effects of drugs can vary depending on the type of cell they are targeting and where specific proteins are acting [49]. We build on the "slice" abstraction [50] to measure model performance at critical biological subsets. Context-specific metrics are defined to measure model performance at critical biological slices, with our benchmarks focused on measuring cell-type-specific model performance. In the case of benchmarks for perturbation response prediction and protein-peptide binding affinity, the studies are limited to a particular cell line, however our definition for context-specific metrics lays the foundations for building models which can generalize across cell lines and make context-aware predictions [29, 50]. For single-cell drug-target nomination, we measure model performance at top-performing cell types. See Section 7.2.6 for definitions.

3.1 `TDC.scDTI`: Contextualized Drug-Target Identification

Motivation. Single-cell data have enabled the study of gene expression and function at the level of individual cells across healthy and disease states [51, 2, 29]. To facilitate biological discoveries using single-cell data, machine-learning models have been developed to capture the complex, cell-type-specific behavior of genes [3, 44, 42, 4]. In addition to providing the single-cell measurements and foundation models, TDC-2 supports the development of contextual AI models to nominate therapeutic targets in a cell type-specific manner [4]. We introduce a benchmark dataset, model, and leaderboard for context-specific therapeutic target prioritization, encouraging the innovation of model architectures (e.g., to incorporate new modalities, such as protein structure and sequences [52, 53, 54, 55, 56], genetic perturbation data [57, 58, 59, 60], disease-specific single-cell atlases [61, 62, 63], and protein networks [64, 65, 66]). TDC-2's release of `TDC.scDTI` is a step in standardizing benchmarks for more comprehensive assessments of context-specific model performance.

Dataset and benchmark. We use curated therapeutic target labels from the Open Targets Platform [67] for rheumatoid arthritis (RA) and inflammatory bowel disease (IBD) [4] (section 7.1.1). We benchmark PINNACLE [4]—trained on cell type specific protein-protein interaction networks—and a graph attention neural network (GAT) [68]—trained on a context-free reference protein-protein interaction network—on the curated therapeutic targets dataset. As expected, PINNACLE underperforms when evaluated on context-agnostic metrics (Table 1) and drastically outperforms GAT when evaluated on context-specific metrics (Appendix Table 1). To our knowledge, TDC-2 provides the first benchmark for context-specific learning [29]. TDC-2's contribution helps standardize the evaluation of single-cell ML models for drug target identification and other single-cell tasks [42, 3, 4, 44].

3.2 `TDC.PerturbOutcome`: Perturbation-Response Prediction

Motivation. Understanding and predicting transcriptional responses to genetic or chemical perturbations provides insights into cellular adaptation and response mechanisms. Such predictions can advance therapeutic strategies, as they enable researchers to anticipate how cells will react to targeted interventions, potentially guiding more effective treatments. Models that have shown promise at this task [26, 27] are limited to either genetic or chemical perturbations without being able to generalize to the other. Approaches that can generalize across chemical and genetic perturbations [31, 32] may be unable to generalize to unseen perturbations without modification.

Table 1: **Cell-type specific target nomination for two therapeutic areas, rheumatoid arthritis (RA) and inflammatory bowel diseases (IBD).** Cell-type specific context metrics (definitions in Section 7.2.6): AP@5 Top-20 CT - average precision at $k = 5$ for the 20 best-performing cell types (CT); AUROC Top-1 CT - AUROC for top-performing cell type; AUROC Top-10 CT and AUROC Top-20 CT - weighted average AUROC for top-10 and top-20 performing cell types, respectively, each weighted by the number of samples in each cell type; AP@5/AUROC CF - context-free AP@5/AUROC integrated across all cell types. Shown are results from models run on ten independent seeds. N/A - not applicable.

Model	AP@5 Top-20 CT	AUROC Top-1 CT	AUROC Top-10 CT	AUROC Top-20 CT	AP@5 CF	AUROC CF
PINNACLE (RA)	$0.913_{\pm 0.059}$	$0.765_{\pm 0.054}$	$0.676_{\pm 0.017}$	$0.647_{\pm 0.014}$	$0.226_{\pm 0.023}$	$0.510_{\pm 0.005}$
GAT (RA)	N/A	N/A	N/A	N/A	$0.220_{\pm 0.013}$	$0.580_{\pm 0.010}$
PINNACLE (IBD)	$0.873_{\pm 0.069}$	$0.935_{\pm 0.067}$	$0.799_{\pm 0.017}$	$0.752_{\pm 0.011}$	$0.198_{\pm 0.013}$	$0.500_{\pm 0.010}$
GAT (IBD)	N/A	N/A	N/A	N/A	$0.200_{\pm 0.023}$	$0.640_{\pm 0.017}$

Dataset and benchmark. We used the scPerturb [69] datasets to benchmark the generalizability of perturbation-response prediction models across seen/unseen perturbations and cell lines. We benchmark models in genetic and chemical perturbations using metrics measuring intra/inter-cell line and seen/unseen perturbation generalizability. We provide results measuring unseen perturbation generalizability for Gene Perturbation Response Prediction using the scPerturb gene datasets (Norman K562, Replogle K562, Replogle RPE1) [70, 71] with results shown in Table 2. For Chemical Perturbation Prediction, we evaluated chemCPA utilizing cold splits on perturbation type and showed a significant decrease in performance for 3 of 4 perturbations evaluated (3). We have also included Biolord [31] and scGen [72] for comprehensive benchmarking on the well-explored perturbation response prediction on seen perturbation types problem. These tests were run on sciPlex2 [73].

3.2.1 Genetic Perturbation Response Prediction

We use scPerturb gene datasets (Norman K562, Replogle K562, Replogle RPE1) [70, 71]. In the case of single-gene perturbations, we assessed the models based on the perturbation of experimentally perturbed genes not included in the training data. We used data from two genetic perturbation screens, with 1,543 perturbations for RPE-1 (retinal pigment epithelium) cells and 1,092 for K-562 cells, each involving over 170,000 cells. These screens utilized the Perturb-seq assay, which combines pooled screening with single-cell RNA sequencing to analyze the entire transcriptome for each cell. We trained GEARS separately on each dataset. In addition to an existing deep learning-based model (CPA), we also developed a baseline model (no perturbation), assuming that gene expression remains unchanged after perturbation.

We evaluated the models' performance by calculating the mean squared error between the predicted gene expression after perturbation and the actual post-perturbation expression for the held-out set. Based on the highest absolute differential expression upon perturbation, the top 20 most differentially expressed genes were selected.

3.2.2 Chemical Perturbation Response Prediction

The dataset consists of four drug-based perturbations from sciPlex2 [73, 69] (BMS, Dex, Nutlin, SAHA). sciPlex2 contains alveolar basal epithelial cells from the A549 (lung adenocarcinoma), K562 (chronic myelogenous leukemia), and MCF7 (mammary adenocarcinoma) tissues. Results are shown in Table 3. Our experiments rely on the coefficient of determination (R^2) as the primary performance measure. We calculate this score by comparing actual measurements with counterfactual predictions for all genes. Assessing all genes is essential to evaluating the decoder's overall performance and understanding the background context. Still, it is also beneficial to determine performance based on top differentially expressed genes [27]. The baseline used discards all perturbation information, adequately measuring the improvement resulting by the models' drug encoding [27]. ChemCPA's performance dropped by an average of 15% across the four perturbations. The maximum drop was 34%. Code for intra/inter cell-line benchmarks for chemical (drug) and genetic (CRISPR) perturbations is in Appendix 7.3.5 and Appendix 7.3.5, respectively. Using this code, users can evaluate models of their choice on the benchmark and submit them to the TDC-2 leaderboards for this task (Appendix 7.3.5).

Table 2: **Unseen genetic perturbation response prediction.** We evaluate GEARS across the top 20 differentially expressed genes, based on the highest absolute differential expression upon perturbation, for MSE (MSE@20DEG). Gene expression was measured in log normalized counts. In single-cell analysis, a standard procedure is to normalize the counts within each cell so that they sum to a specific value (usually the median sum across all cells in the dataset) and then to log transform the values using the natural logarithm [26]. For both normalization and ranking genes by differential expression, we utilized Scanpy [74]. We used the sc.tl.rank_genes_groups() function with default parameters in Scanpy, which employs a t-test to estimate scores. This function provides a z-score for each gene and ranks genes based on the absolute values of the score. Genes showing a significant level of dropout were not included in this metric.

Dataset	Tissue	Cell Line	Method	MSE@20DEG
Norman K562	K562	lymphoblast	no-perturb	0.341±0.001
Norman K562	K562	lymphoblast	CPA	0.230±0.008
Norman K562	K562	lymphoblast	GEARS	0.176±0.003
Replogle 562	K562	lymphoblast	no-perturb	0.126±0.000
Replogle 562	K562	lymphoblast	CPA	0.126±0.000
Replogle 562	K562	lymphoblast	GEARS	0.109±0.004
Replogle RPE1	RPE-1	epithelial	no-perturb	0.164±0.000
Replogle RPE1	RPE-1	epithelial	CPA	0.162±0.001
Replogle RPE1	RPE-1	epithelial	GEARS	0.110±0.003

Table 3: **Unseen chemical perturbation response prediction.** We have evaluated chemCPA utilizing cold splits on perturbation type and show a significant decrease in performance for 3 of 4 perturbations evaluated. We have also included Biolord [31] and scGen [72] for comparison. The dataset used consists of four chemical (drug) perturbations from sciPlex2 [69] (BMS, Dex, Nutlin, SAHA). sciPlex2 contains alveolar basal epithelial cells from the A549 (lung adenocarcinoma), K562 (chronic myelogenous leukemia), and MCF7 (mammary adenocarcinoma) tissues. Our experiments rely on the coefficient of determination (R^2) as the primary performance measure.

Drug	Method	R^2 (seen perturbations)	R^2 (unseen perturbations)
BMS	Baseline	0.620±0.044	N/A
Dex	Baseline	0.603±0.053	N/A
Nutlin	Baseline	0.628±0.036	N/A
SAHA	Baseline	0.617±0.027	N/A
BMS	Biolord	0.939±0.022	N/A
Dex	Biolord	0.942±0.028	N/A
Nutlin	Biolord	0.928±0.026	N/A
SAHA	Biolord	0.980±0.005	N/A
BMS	ChemCPA	0.943±0.006	0.906±0.006
Dex	ChemCPA	0.882±0.014	0.540±0.013
Nutlin	ChemCPA	0.925±0.010	0.835±0.009
SAHA	ChemCPA	0.825±0.026	0.690±0.021
BMS	scGen	0.903±0.030	N/A
Dex	scGen	0.944±0.018	N/A
Nutlin	scGen	0.891±0.032	N/A
SAHA	scGen	0.948±0.034	N/A

3.3 `TDC.ProteinPeptide`: Contextualized Protein-Peptide Interaction Prediction

Motivation. Evaluating protein-peptide binding prediction models requires standardized benchmarks, presenting challenges in assessing and validating model performance across different studies [13]. Despite the availability of several benchmarks for protein-protein interactions, this is not the case for protein-peptide interactions. The renowned multi-task benchmark for Protein sEquence undERstanding (PEER) [18] and MoleculeNet [35] both lack support for a protein-peptide interaction prediction task. Furthermore, protein-peptide binding mechanisms vary wildly by cellular and biological context [75, 76, 77, 78]. Current models, as such, tend to be restricted to one task instance (i.e., T Cell Receptor (TCR) and Peptide-MHC Complex or B Cell Receptor (BCR) and Antigen Peptide binding) and do not span protein-peptide interactions [79, 80, 81, 7, 82, 83, 84].

Table 4: **TCR-epitope binding interaction binary classification performance.** All models perform poorly under realistic but challenging RN and ET experimental setups. The best-performing model in RN is AVIB-TCR, with an average of 0.576 (AUROC). The best-performing model in ET is MIX-TPI, with an average of 0.700 (AUROC). For NA, 4 of 6 models achieve near-perfect AUROC.

Methods	Experimental setup	ACC	F1	AUROC	AUPRC
AVIB-TCR	RN	0.570±0.028	0.468±0.086	0.576±0.049	0.605±0.044
MIX-TPI	RN	0.539±0.039	0.408±0.122	0.558±0.028	0.597±0.049
Net-TCR2	RN	0.528±0.050	0.354±0.036	0.551±0.042	0.554±0.075
PanPep	RN	0.507±0.028	0.473±0.039	0.535±0.021	0.579±0.040
TEINet	RN	0.459±0.036	0.619±0.036	0.535±0.029	0.581±0.043
TITAN	RN	0.476±0.063	0.338±0.111	0.502±0.066	0.523±0.055
AVIB-TCR	ET	0.611±0.012	0.553±0.020	0.683±0.010	0.815±0.006
MIX-TPI	ET	0.652±0.009	0.523±0.035	0.703±0.016	0.825±0.014
Net-TCR2	ET	0.621±0.027	0.522±0.020	0.674±0.017	0.810±0.016
PanPep	ET	0.556±0.009	0.506±0.011	0.638±0.009	0.753±0.009
TEINet	ET	0.356±0.008	0.512±0.010	0.571±0.009	0.646±0.011
TITAN	ET	0.670±0.013	0.492±0.048	0.624±0.021	0.733±0.018
AVIB-TCR	NA	0.636±0.062	0.197±0.169	0.944±0.021	0.949±0.023
MIX-TPI	NA	0.952±0.029	0.937±0.040	0.992±0.002	0.995±0.001
Net-TCR2	NA	0.655±0.051	0.274±0.123	0.973±0.009	0.985±0.005
PanPep	NA	0.419±0.011	0.352±0.006	0.611±0.014	0.499±0.031
TEINet	NA	0.413±0.023	0.582±0.023	0.973±0.011	0.981±0.006
TITAN	NA	0.695±0.050	0.404±0.141	0.629±0.053	0.661±0.040

We define and evaluate a subtask for TCR-Epitope binding interaction prediction applying contextual AI (Section 7.2.3) to the T Cell cell line.

3.3.1 TCR-Epitope (Peptide-MHC Complex) Interaction Prediction

The critical challenge in TCR-Epitope (Peptide-MHC Complex) interaction prediction lies in creating a model that can effectively generalize to unseen TCRs and epitopes [85]. While TCR-H [86] and TEINet [87] have shown improved performance on prediction for known epitopes, by incorporating advanced features like attention mechanisms and transfer learning, the performance considerably drops for unseen epitopes [88, 89]. Another challenge in TCR-Epitope interaction prediction lies in the choice of heuristic for generating negative samples, with non-binders often underrepresented or biased in curated datasets, leading to inaccurate predictions when generalized [30].

Datasets and Benchmarks. TDC-2 establishes a curated dataset and benchmark within its single-cell protein-peptide binding affinity prediction task to measure model generalizability to unseen TCRs and epitopes and model sensitivity to the selection of negative data points. Benchmarking datasets use three types of heuristics for generating negative samples: random shuffling of epitope and TCR sequences (RN), experimental negatives (NA), and pairing external TCR sequences with epitope sequences (ET). We harness data from the TC-hard dataset [90] for the first two types and PanPep [91] for the third type. Both datasets use hard [90] splits, ensuring that epitopes in the testing set are not present in the training set. Our results (Table 4) show the lack of a reasonable heuristic for generating negative samples, with model performance evaluation shown to be unsatisfactory. For two heuristics, all models perform poorly. The best-performing model in ET is MIX-TPI, with roughly 0.70 AUROC. The best-performing model in RN is AVIB-TCR, with approximately 0.576 AUROC. For NA, 4 of 6 models perform near-perfectly as measured on AUROC. Models benchmarked include AVIB-TCR [28], MIX-TPI [92], Net-TCR2 [93], PanPep [85], TEINet [87], and TITAN [89].

4 Discussion

TDC-2 makes technical strides in unifying a challenging body of work under a representative architecture (Section 6.3.1) and a collection of datasets (Section 7.1), benchmarks (Section 3), and foundation model tools (Section 6.3.2). The presented models and benchmarks illustrate the challenges of developing machine learning methods in therapeutics capable of generalizing to out-of-distribution samples across varying biological contexts.

Identifying most predictive biological contexts and cell types. There is evidence that the effects of drugs can vary depending on the type of cell they are targeting and where specific proteins are acting [49]. Therefore, it is essential to identify the most predictive cell-type contexts by evaluating therapeutic machine-learning tasks with cell-type-specific metrics. This approach can help determine the cell types that play crucial and distinct roles in the disease pathogenesis of conditions like rheumatoid arthritis (RA) and inflammatory bowel diseases (IBD). In our study, we used cell-type-specific metrics to compare the performance of [4] and [68] models for the TDC.scDTI task (Table 1). The results showed that PINNACLE protein representations outperformed the GAT model for RA and IBD diseases across the top 1, 10, and 20 (most predictive) cell types.

Generalizing genetic perturbation response prediction models to out-of-distribution samples. Gene editing is a promising tool for diseases that cannot be effectively treated or managed with alternative therapeutic modalities. For instance, the FDA recently approved gene editing to modify T-cells for treating patients with acute lymphoblastic leukemia [94]. However, many disease-causing genetic variants result from insertions and deletions, making it crucial to accurately predict gene editing outcomes to ensure effectiveness and minimize off-target effects. Understanding how cells respond to genetic changes is essential for this. However, the vast number of potential genetic changes makes it difficult to study all possibilities experimentally [95]. Although there have been improvements in predicting the outcomes of genetic changes, applying these predictions to new genetic changes and cell types when predicting gene expression responses remains challenging [26]. We assessed the performance of various models in predicting gene responses to single and multiple genetic perturbations using single-cell RNA-sequencing data from genetic screens. We found that recently developed models perform well when tested on a single cell line; however, they struggle to generalize to cell lines not encountered during training. This gap presents valuable opportunities for algorithmic innovation.

Generalizing chemical perturbation response prediction models to out-of-distribution samples. Using high-throughput techniques such as cell hashing has made it easier to conduct single-cell RNA sequencing in multi-sample experiments at a low cost. However, these methods require expensive library preparation and are not easily scalable to many perturbations. This becomes incredibly challenging when studying the effects of combination therapies or genetic perturbations, where experimental screening of all possible combinations becomes impractical. While projects like the Human Cell Atlas aim to comprehensively map cellular states across tissues, creating a similar atlas for the effects of perturbations on gene expression is impossible due to the vast number of possibilities. Therefore, it is crucial to develop computational tools to guide the exploration of the perturbation space and identify promising candidate combination therapies in high-throughput screenings. A successful computational method for navigating the perturbation space should be able to predict cell behavior when subject to novel combinations of perturbations that were only measured separately in the original experiment. In our study, we benchmarked Biolord [31], scGen [72], and ChemCPA [27] on chemical perturbation response prediction and showed significant improvement over the baseline method. Furthermore, we demonstrate a considerable drop in performance for ChemCPA when generalizing to unseen perturbations. As Biolord and scGen cannot generalize to unseen perturbations without modification, our study highlights the need for developing models that can more effectively generalize to unseen combinations of chemical perturbations.

Defining negative samples for out-of-distribution prediction of protein-peptide binding affinity. Studying T-cell receptors (TCRs) has become crucial in cancer immunotherapy and human infectious disease research [96]. TCRs can detect processed peptides within infected or abnormal cells. Recent studies have focused on predicting TCR-peptide/-pMHC binding using machine or deep learning methods [89, 79]. Many of these studies use data from the Immune Epitope Database (IEDB) [97], VDJdb [98], and McPAS-TCR [99], which predominantly contain CDR3-beta data and lack information on CDR3-alpha. While these methods perform well on test sets from the same source as the training set, they struggle with out-of-distribution samples [91]. This study evaluates cutting-edge methods for out-of-distribution predictions by splitting datasets through a hard split [28]. Additionally, the study reveals a significant sensitivity in model performance to the choice of heuristic for generating negative samples [30], emphasizing the need for further dataset curation and the evaluation of out-of-distribution samples. The datasets have been made available in TDC-2 containing CDR3-beta and CDR3-alpha sequences (Section 7.1).

Limitations and societal considerations. Open-source datasets and benchmark providers like [17], [35], and [11, 12] play a role in advancing AI by enabling accessible and standardized evaluation methods. However, their limitations and potential negative societal impacts include the risk of biased

or incomplete data, which may lead to inaccurate or non-representative AI models. Additionally, the open accessibility of such datasets could lead to misuse, including unethical applications or the proliferation of AI models that reinforce existing biases in medicine. Moreover, reliance on standardized benchmarks may discourage innovation and lead to the over-fitting of models to specific datasets, potentially limiting their generalization in real-world scenarios. Last, evaluating deep learning models for genetic perturbation tasks requires reconsideration in light of recent findings that question their effectiveness for this problem. A recent study revealed that deep learning models do not consistently outperform simpler linear models across various benchmarks [100]. TDC-2's datasets, benchmarks, metrics, and foundation model tooling lay the foundation for more thorough study, development, and evaluation of models in this space.

5 Conclusion

TDC-2 introduces an API-first-dataset architecture that supports retrieving continually updated heterogeneous data sources. The architecture provides abstractions for complex data processing pipelines [11], data lineage [23], and versioning [24]. It augments the stability of emerging biomedical AI workflows, based on advancements from [20, 21, 101], with continuous data updates [25]. It does so via the development of a multi-modal data and model retrieval API leveraging the Model-View-Controller [33, 102, 103] paradigm to introduce data views [104] and a domain-specific-language [105] (Section 6.3.1).

The Commons 2.0 (TDC-2) presents a collection of datasets, tools, models, and benchmarks integrating cell-type-specific contextual features with ML tasks across the range of therapeutics. TDC-2 drastically expands the modalities previously available on TDC [11, 12]. TDC-2 supports a far larger set of data modalities and ML tasks than other dataset collections [2] and benchmarks [17, 38, 18, 35]. Modalities in TDC-2 include but are not limited to: single-cell gene expression atlases [2, 51], chemical and genetic perturbations [69], clinical trial data [14], peptide sequence data [90, 91], peptidomimetics protein-peptide interaction data from AS-MS spectroscopy [15, 106], novel 3D structural protein data [107, 108, 109], and cell-type-specific protein-protein interaction networks at single-cell resolution [4]. TDC-2 introduces ML tasks taking on open challenges, including the inferential gap in precision medicine [110, 111] and evaluation on longitudinal data (equation 20), model generalization across cell lines [26, 31] and single-cell perturbations [27] that were not encountered during model training, and evaluation of models across a broad range of diverse biological contexts [4, 30].

TDC-2 is a platform that quantitatively defines open challenges in single-cell therapeutics, determines the current state-of-the-art solutions, promotes method development to improve these solutions, and monitors progress toward these goals. TDC-2 enables broader accessibility for scientists to contribute to advancing the field of single-cell therapeutics. TDC-2 will shift the perspective on method selection and evaluation for therapeutics discovery and machine learning scientists, supporting a transition towards higher standards for methods in contextual AI for therapeutics.

6 Appendix

This technical appendix, along with supplementary in section 7, provides a detailed overview of the design, tasks, and benchmarks introduced by TDC-2.

All code and documentation can be found in the TDC-2 Github repository. The URL ishttps://github.com/mims-harvard/TDC/tree/main. In addition, our website contains all datasets and licenses and further documentation https://tdcommons.ai/.

6.1 Data Availability

The website contains all informaton on all datasets discussed in this manuscript under their corresponding tasks. These are: TDC.scDTI (https://tdcommons.ai/multi_pred_tasks/scdti/), TDC.PerturbOutcome (https://tdcommons.ai/multi_pred_tasks/counterfactual/), TDC.TCREpitope (https://tdcommons.ai/multi_pred_tasks/tcrepitope/), TDC.TrialOutcome (https://tdcommons.ai/multi_pred_tasks/trialoutcome/), TDC.SBDD (https://tdcommons.ai/generation_tasks/sbdd/). In addition, all TDC datasets are made available via the harvard dataverse https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/21LKWG . Instructions for accessing datasets via the TDC Python API can be found in section 7.1.

6.2 Code Availability

All code and documentation can be found in our Github repo. The URL is https://github.com/mims-harvard/TDC/tree/main.

6.3 Methods

6.3.1 API-First Design and Model-View-Controller

TDC-2 drastically expands dataset retrieval capabilities available in TDC-1 beyond those of other leading benchmarks. Leading benchmarks, like MoleculeNet [35] and TorchDrug [17] have traditionally provided dataloaders to access file dumps. TDC-2 introduces API-integrated multimodal data-views [33, 112, 104]. To do so, the software architecture of TDC-2 was redesigned using the Model-View-Controller (MVC) design pattern [103, 102]. The MVC architecture separates the model (data logic), view (UI logic), and controller (input logic), which allows for the integration of heterogeneous data sources and ensures consistency in data views [33]. The MVC pattern supports the integration of multiple data modalities by using data mappings and views [104]. The MVC-enabled-multimodal retrieval API is powered by TDC-2's Resource Model (Section 6.3.2).

TDC DataLoader (*Model*). As per the TDC-1 specification, this component queries the underlying data source to provide raw or processed data to upstream function calls. We augmented this component beyond TDC-1 functionality to allow for querying datasets introduced in TDC-2, such as the CZ CellXGene.

TDC meaningful data splits and multimodal data processing (*View*). As per the TDC-1 specification, this component implements data splits to evaluate model generalizability to out-of-distribution samples and data processing functions for multiple modalities. We augmented this component to act on data views [33] specified by TDC-2's controller.

TDC-2 Domain-Specific Language (*Controller*). TDC-2 develops an Application-Embedded Domain-Specific Data Definition Programming Language facilitating the integration of multiple modalities by generating data views from a mapping of multiple datasets and functions for transformations, integration, and multimodal enhancements, while mantaining a high level of abstraction [105] for the Resource framework. We include examples developing multimodal datasets leveraging this MVC DSL in listing 2.

6.3.2 Resource Model

The Commons introduces a redesign of TDC-1's dataset layer into a new data model dubbed the TDC-2 resource, which has been developed under the MVC paradigm to integrate multiple modalities into the API-first model of TDC-2.

CZ CellXGene with single cell biology datasets. CZ CellXGene [2] is a open-source platform for analysis of single-cell RNA sequencing data. We leverage the CZ CellXGene to develop a TDC-2 Resource Model for constructing large-scale single-cell datasets that maps gene expression profiles of individual cells across tissues, healthy and disease states. TDC-2 leverages the SOMA (Stack of Matrices, Annotated) API, adopts TileDB-SOMA [113] for modeling sets of 2D annotated matrices with measurements of features across observations, and enables memory-efficient querying of single-cell modalities (i.e., scRNA-seq, snRNA-seq), across healthy and diseased samples, with tabular annotations of cells, samples, and patients the samples come from.

We develop a remote procedure call (RPC) API taking the string name (e.g., listing 3 in section 7.3.1) of the desired reference dataset as specified in the CellXGene [2]. The remote procedure call for fetching data is specified as a Python generator expression, allowing the user to iterate over the constructed single-cell atlas without loading it into memory [114]. Specifying the RPC as a Python generator expression allows us to make use of memory-efficient querying as provided by TileDB [113]. The single cell datasets can be integrated with therapeutics ML workflows in TDC-2 by using tools such as PyTorch's IterableDataset module [115].

Knowledge graph, external APIs, and model hub. We have developed a framework for biomedical knowledge graphs to enhance multimodality of dataset retrieval via TDC-2's Resource Model. Our system leverages PrimeKG to integrate 20 high-quality resources to describe 17,080 diseases with 4,050,249 relationships [22]. Our framework also extends to external APIs, with data views currently leveraging BioPython [116], for obtaining nucleotide sequence information for a given non-coding RNA ID from NCBI [116], and The Uniprot Consortium's RESTful GET API [117] for obtaining amino acid sequences. In addition we've developed the framework to allow access to embedding models under diverse biological contexts via the TDC-2 Model Hub. Examples using these components are in sections 7.3.2 and 7.3.3.

6.4 Experiments

6.4.1 TDC.TrialOutcome

TDC-2 introduces a model framework, task definition, dataset, and benchmark for the Clinical Outcome Prediction task tailored to precision medicine. The framework and definition aim to assess clinical trials systematically and comprehensively by predicting various endpoints for patient subpopulations. Our benchmark uses the Trial Outcome Prediction (TOP) dataset [14]. TOP consists of 17,538 clinical trials with 13,880 small-molecule drugs and 5,335 diseases. We include the task formulation (section 7.2.4), dataset details 7.1.6, and benchmark (section 7.3.5).

Dataset and benchmark. Our benchmark uses the Trial Outcome Prediction (TOP) dataset [14]. TOP consists of 17,538 clinical trials with 13,880 small-molecule drugs and 5,335 diseases. Out of these trials, 9,999 (57.0%) succeeded (i.e., meeting primary endpoints), and 7,539 (43.0%) failed. Out of these trials, 1,787 were in Phase I testing (toxicity and side effects), 6,102 in Phase II (efficacy), and 4,576 in Phase III (effectiveness compared to current standards). We perform a temporal split for benchmarking. The train/validation and test are time-split by the date January 1, 2014, i.e., the start dates of the test set are after January 1, 2014, while the completion dates of the train/validation set are before January 1, 2014. Here, the HINT model [14], is benchmarked against COMPOSE [118] and DeepEnroll [119] models. Results are shown in table 6.

6.4.2 Reproducing TDC-2 Benchmarks

Here, we include the instructions for replicating TDC-2 benchmarks, the total amount of computing, and the type of resources used. Code and data details are in sections 7.3.5 and 7.3.4.

TDC.scDTI. For benchmarking across ten seeds and another model benchmark, see Section 7.3.5. For pre-training, the best hyperparameters are as follows: the dimension of the nodes' feature matrix = 1024, dimension of the output layer = 16, lambda = 0:1, learning rate for link prediction task = 0:01, learning rate for protein's cell type classification task = 0:1, number of attention heads = 8, weight decay rate = 0:00001, dropout rate = 0:6, and normalization layers are `layernorm` and `batchnorm`. For pre-training, models are trained on a single NVIDIA Tesla V100-SXM2-16GB GPU. Hyperparameters are used for fine-tuning, as per the Github documentation linked in section 7.3.5. Models are trained on a single NVIDIA Tesla M40 GPU. The relevant function calls are documented in Section 7.3.5.

Listing 1: Command line invocations to reproduce the TDC.scDTI benchmark results for [4]. These can also be found by following links in section 7.3.5 or directly in `https://github.com/mims-harvard/PINNACLE/tree/main/finetune_pinnacle`.

```
# Rheumatoid Arthritis (EFO_0000685)
python train.py \
    --task_name=EFO_0000685 \
    --embeddings_dir=../data/pinnacle_embeds/ \
    --positive_proteins_prefix ../data/therapeutic_target_task/
        ,! positive_proteins_EFO_0000685 \
    --negative_proteins_prefix ../data/therapeutic_target_task/
        ,! negative_proteins_EFO_0000685 \
    --data_split_path ../data/therapeutic_target_task/data_split_EFO_0000685
        ,!  \
    --actn=relu \
    --dropout=0.2 \
    --hidden_dim_1=32 \
    --hidden_dim_2=8 \
    --lr=0.01 \
    --norm=bn \
    --order=dn \
    --wd=0.001 \
    --random_state 1 \
    --num_epoch=2000

# Inflammatory bowel disease (EFO_0003767)
python train.py \
    --task_name=EFO_0003767 \
    --embeddings_dir=../data/pinnacle_embeds/ \
    --positive_proteins_prefix ../data/therapeutic_target_task/
        ,! positive_proteins_EFO_0003767 \
    --negative_proteins_prefix ../data/therapeutic_target_task/
        ,! negative_proteins_EFO_0003767 \
    --data_split_path ../data/therapeutic_target_task/data_split_EFO_0003767
        ,!  \
    --actn=relu \
    --dropout=0.4 \
    --hidden_dim_1=32 \
    --hidden_dim_2=8 \
    --lr=0.001 \
    --norm=ln \
    --order=nd \
    --wd=0.0001 \
    --random_state 1 \
    --num_epoch=2000
```

TDC.PerturbOutcome - Genetic. All benchmarked methods follow the training procedure described in [26]. Specifically, we use the simulation data split to mimic the real-world use case of genetic perturbation machine learning models. For the Norman double combination perturbation dataset, we withhold perturbations that are either both unseen, one unseen, or both seen in the test set. For the Replogle K562 and RPE1 single perturbation datasets, we split the data by single genes and test on unseen single-gene perturbations. The hyperparameters used were optimal after optimization as reported in [26]. Each model run was executed on an internal high-performance cluster with an Ubuntu 16.04 operating system, using a single Nvidia Quadro RTX 8000 48GB GPU. The code to reproduce the experiment is available at `https://github.com/mims-harvard/TDC/tree/main/examples/multi_pred/geneperturb`.

TDC.PerturbOutcome - Chemical. Benchmark results can be reproduced with code in section 7.3.5. Default settings were used from each model's GitHub repository, and the experiments were run using Nvidia A100.

TDC.TCREpitope. The models for TCR-epitope binding prediction were run on a single A100. We prepared the input data files in the format (most in CSV files) according to the official tutorials. Unknown amino acid letters were replaced by X or removed according to the method requirements. If CDR3A and CDR3B are available, the models will be trained on both unless they can only accept one TCR sequence as input (such as TITAN). If CDR3A is unavailable (ET data), all the models will be trained in the beta-only module. We kept the default parameters to run all the methods. For running TITAN, we transferred the amino acid sequences of epitopes to the SMILE sequences as the inputs. To keep the unseen scenario, we used a zero-shot module of PanPep in the tests of all the data settings. The code for reproducing our benchmark results is in table 7.3.5.

6.5 Tables

Table 5: **Comparison of TDC-2 with other datasets, benchmarks, and ML platforms in therapeutics** TDC-2 distinguishes itself from related datasets [34, 35], benchmarks [17, 18, 36, 37], model development frameworks [38, 39, 40], and therapeutic initiatives [2] in its integration of single-cell analysis with multimodal machine learning in drug discovery via four contextual AI tasks and retrieval APIs for multimodal datasets and models. The API-first [46, 47, 48] approach adopted by TDC-2's multimodal retrieval API, dubbed the API-first dataset, enables development of large language models invoking experimental biomedical data retrieval APIs based on therapeutic queries. TDC-2 integrates expert-designed tools into this unified LLM-friendly API to foster scientific advancement by bridging the gap between experimental and computational therapeutic science. OP - Open Problems.

Feature	TDC	TorchDrug	MoleculeNet	OP	scPerturb	CELLXGENE	TDC-2
Single-cell Drug-Target Identification Task	✗	✗	✗	✗	✗	✗	✓
CRISPR-Cell Perturbation Response Prediction	✗	✗	✗	✗	✓	✗	✓
Drug-Cell Perturbation Response Prediction	✗	✗	✗	✗	✓	✗	✓
Single-cell Protein-Peptide Binding Interaction Prediction	✗	✗	✗	✗	✗	✗	✓
Structure-Based Drug Design	✗	✓	✓	✗	✗	✗	✓
Clinical Trial Outcome Prediction	✗	✗	✗	✗	✗	✗	✓
Knowledge Graphs	✗	✓	✗	✗	✗	✗	✓
Python API	✓	✓	✗	✓	✗	✓	✓
Single-cell Gene Expression Atlases	✗	✗	✗	✓	✓	✓	✓
External API data retrieval	✗	✗	✗	✗	✗	✗	✓
Data Views	✗	✗	✗	✗	✗	✗	✓
Foundation Model Embeding Retrieval	✗	✓	✗	✗	✗	✓	✓

Table 6: Clinical Trial Outcome Prediction task benchmark model results on the TOP dataset [14], described in section 6.4.1.

Model	Phase 1 AUPRC	Phase 2 AUPRC	Phase 3 AUPRC	Indication Level AUPRC
HINT	0.772	0.607	0.623	0.703
COMPOSE	0.665	0.532	0.545	0.624
DeepEnroll	0.701	0.580	0.590	0.655

6.6 Figures

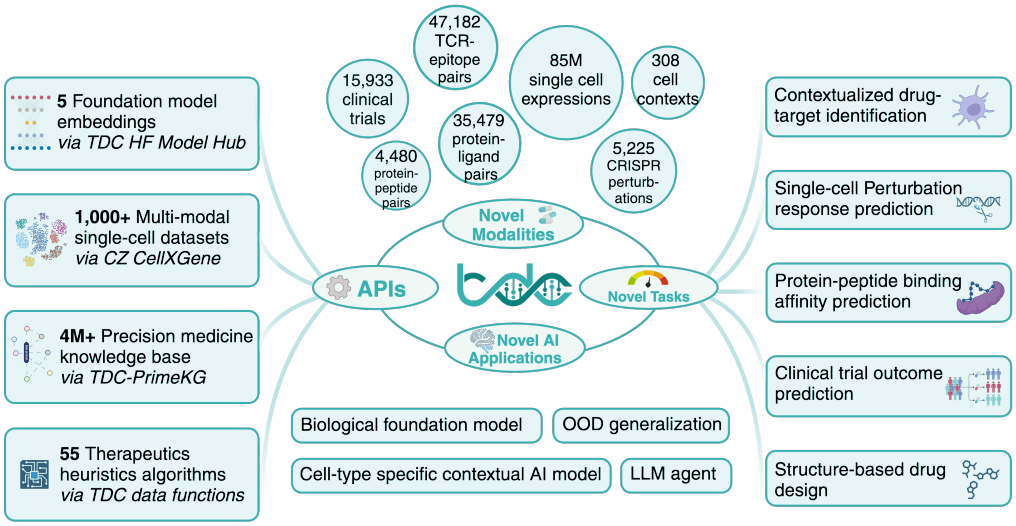

Figure 1: **Overview of TDC-2.** TDC-2 introduces a multimodal retrieval API powering ML-task-driven [11] datasets [69, 4, 67, 107, 91, 90, 14, 15, 109, 108]and benchmarks spanning 10+ new modalities and 5 state-of-the-art machine learning tasks (section 7.2), including 4 contextual AI tasks: TDC.scDTI (section 3.1), single-cell genetic perturbation response prediction (section 3.2.1), single-cell chemical perturbation response prediction (section 3.2.2), and single-cell protein-peptide interaction prediction (section 3.3). Model benchmarks highlighting biomedical AI challenges in OOD Generalization [26, 27, 120, 14] and evaluation [4, 30] of cell-type-specific contextual AI models are introduced.

7 Supplementary

7.1 Datasets

The website `tdcommons.ai` contains all datasets discussed in this manuscript under their corresponding tasks. These are: TDC.scDTI (`https://tdcommons.ai/multi_pred_tasks/scdti/`), TDC.PerturbOutcome (`https://tdcommons.ai/multi_pred_tasks/counterfactual/`), TDC.TCREpitope (`https://tdcommons.ai/multi_pred_tasks/tcrepitope/`), TDC.TrialOutcome (`https://tdcommons.ai/multi_pred_tasks/trialoutcome/`), TDC.SBDD (`https://tdcommons.ai/generation_tasks/sbdd/`). In addition, all TDC datasets are made available via the harvard dataverse `https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/21LKWG` . Here we include dataset curation details and code for accessing all datasets for the introduced tasks.

7.1.1 (Li, Michelle, et al.) Dataset

To curate target information for a therapeutic area, we examine the drugs indicated for the therapeutic area of interest and its descendants. The two therapeutic areas examined are rheumatoid arthritis (RA) and inflammatory bowel disease. For rheumatoid arthritis, we collected therapeutic data (i.e., targets of drugs indicated for the therapeutic area) from OpenTargets for rheumatoid arthritis (EFO 0000685), ankylosing spondylitis (EFO 0003898), and psoriatic arthritis (EFO 0003778). For inflammatory bowel disease, we collected therapeutic data for ulcerative colitis (EFO 0000729), collagenous colitis (EFO 1001293), colitis (EFO 0003872), proctitis (EFO 0005628), Crohn's colitis (EFO 0005622), lymphocytic colitis (EFO 1001294), Crohn's disease (EFO 0000384), microscopic colitis (EFO 1001295), inflammatory bowel disease (EFO 0003767), appendicitis (EFO 0007149), ulcerative proctosigmoiditis (EFO 1001223), and small bowel Crohn's disease (EFO 0005629).

We define positive examples (i.e., where the label y = 1) as proteins targeted by drugs that have at least completed phase 2 of clinical trials for treating a specific therapeutic area. As such, a protein is a promising candidate if a compound that targets the protein is safe for humans and effective for treating the disease. We retain positive training examples activated in at least one cell type-specific protein interaction network.

We define negative examples (i.e., where the label y = 0) as druggable proteins that do not have any known association with the therapeutic area of interest according to Open Targets. A protein is deemed druggable if targeted by at least one existing drug. We extract drugs and their nominal targets from Drugbank. We retain negative training examples activated in at least one cell type-specific protein interaction network.

Dataset statistics. The final number of positive (negative) samples for RA and IBD were 152 (1,465) and 114 (1,377), respectively. In [4], this dataset was augmented to include 156 cell types.

Dataset split. Cold Split: We split the dataset such that about 80% of the proteins are in the training set, about 10% of the proteins are in the validation set, and about 10% of the proteins are in the test set. The data splits are consistent for each cell type context to avoid data leakage.

References. [4]

Dataset license. CC BY 4.0

Code Sample

The dataset and splits are currently available on TDC Harvard Dataverse. In addition, you may obtain the protein splits used in [4] via the following code.

```
from tdc.resource.dataloader import DataLoader
data = DataLoader(name="opentargets_dti")
splits = data.get_split()
```

7.1.2 scPerturb Dataset

The scPerturb dataset is a comprehensive collection of single-cell perturbation data harmonized to facilitate the development and benchmarking of computational methods in systems biology. It includes various types of molecular readouts, such as transcriptomics, proteomics, and epigenomics. scPerturb is a harmonized dataset that compiles single-cell perturbation-response data. This dataset is designed to support the development and validation of computational tools by providing a consistent and comprehensive resource. The data includes responses to various genetic and chemical perturbations, crucial for understanding cellular mechanisms and developing therapeutic strategies. Data from different sources are uniformly pre-processed to ensure consistency. Rigorous quality control measures are applied to maintain high data quality. Features across different datasets are standardized for easy comparison and integration.

Dataset statistics. 44 publicly available single-cell perturbation-response datasets. Most datasets have, on average, approximately 3000 genes measured per cell. 100,000+ perturbations.

Dataset split. Cold Split and **Random Split** defined on cell lines and perturbation types.

References. [69]

Dataset license. CC BY 4.0

Code Sample

```
from tdc.multi_pred.perturboutcome import PerturbOutcome
from pandas import DataFrame
test_loader = PerturbOutcome(
    name="scperturb_drug_AissaBenevolenskaya2021")
testdf = test_loader.get_data()
```

7.1.3 TCHard Dataset

The TChard dataset is designed for TCR-peptide/-pMHC binding prediction. It includes over 500,000 samples from sources such as IEDB, VDJdb, McPAS-TCR, and the NetTCR-2.0 repository. The dataset is utilized to investigate how state-of-the-art deep learning models generalize to unseen peptides, ensuring that test samples include peptides not found in the training set. This approach highlights the challenges deep learning methods face in robustly predicting TCR recognition of peptides not previously encountered in training data.

Dataset statistics. 500,000 samples

Dataset split. Cold Split referred to as "Hard" split in [90].

References. [90]

Dataset license. Non-Commercial Use

Code Sample

```
from tdc.resource.dataloader import DataLoader
data = DataLoader(name="tchard")
self.split = data.get_split()
```

7.1.4 PanPep Dataset

PanPep is a framework constructed in three levels for predicting the peptide and TCR binding recognition. We have provided the trained meta learner and external memory, and users can choose different settings based on their data available scenarios: Few known TCRs for a peptide: few-shot setting; No known TCRs for a peptide: zero-shot setting; plenty of known TCRs for a peptide: majority setting. More information is available in the Github repo .

Dataset statistics. Data from multiple studies involving millions of TCR sequences.

Dataset split. Cold Split referred to as "Hard" split in [91].

References. [91]

Dataset license. GPL-3.0

Code Sample

```
from tdc.resource.dataloader import DataLoader
data = DataLoader(name="panpep")
self.split = data.get_split()
```

7.1.5 (Ye X et al) Dataset

Affinity selection-mass spectrometry data of discovered ligands against single biomolecular targets (MDM2, ACE2, 12ca5) from the Pentelute Lab of MIT This dataset contains affinity selection-mass spectrometry data of discovered ligands against single biomolecular targets. Several AS-MS-discovered ligands were taken forward for experimental validation to determine the binding affinity (KD) as measured by biolayer interferometry (BLI) to the listed target protein. If listed as a "putative binder," AS-MS alone was used to isolate the ligands to the target, with KD < 1 uM required and often observed in orthogonal assays, though there is some (< 50%) chance that the ligand is nonspecific. Most of the ligands are putative binders, with 4446 total provided. For those characterized by BLI (only 34 total), the average KD is 266 ± 44 nM; the median KD is 9.4 nM.

Dataset statistics. 34 positive ligands, 4446 putative binders, and three proteins

Dataset Split. Stratified Split and **N/A Split**: We provide stratified 10/90 split on train/test as well as "test set only" split.

References. [106, 15]

Dataset license. CC BY 4.0

Code Sample

```
from tdc.multi_pred import ProteinPeptide
data = ProteinPeptide(name="brown_mdm2_ace2_12ca5")
data.get_split()
```

7.1.6 TOP Dataset

TOP [14] consists of 17,538 clinical trials with 13,880 small-molecule drugs and 5,335 diseases. Out of these trials, 9,999 (57.0%) succeeded (i.e., meeting primary endpoints), and 7,539 (43.0%) failed. For each clinical trial, we produce the following four data items: (1) drug molecule information, including Simplified Molecular Input Line Entry System (SMILES) strings and molecular graphs for the drug candidates used in the trials; (2) disease information including ICD-10 codes (disease code), disease description, and disease hierarchy in terms of CCS codes (https://www.hcup-us.ahrq.gov/toolssoftware/ccs10/ccs10.jsp); (3) trial eligibility criteria are in unstructured natural language and contain inclusion and exclusion criteria; and (4) trial outcome information includes a binary indicator of trial success (1) or failure (0), trial phase, start and end date, sponsor, and trial size (i.e., number of participants).

Dataset statistics. Phase I: 2,402 trials / Phase II: 7,790 trials / Phase III: 5,741 trials.

Dataset split. Temporal Split as defined in [14] and Section 6.4.1.

References. [14]

Dataset license. Non-Commercial Use

Code Sample

```
from tdc.multi_pred import TrialOutcome
data = TrialOutcome(name = 'phase1') # 'phase2' / 'phase3'
split = data.get_split()
```

7.1.7 PDBBind Dataset

PDBBind is a comprehensive database extracted from PDB with experimentally measured binding affinity data for protein-ligand complexes. PDBBind does not allow the dataset to be re-distributed in any format. Thus, we could not host it on the TDC server. However, we provide an alternative route since significant processing is required to prepare the dataset ML. The user only needs to register at http://www.pdbbind.org.cn/, download the raw dataset, and then provide the local path. TDC will then automatically detect the path and transform it into an ML-ready format for the TDC data loader.

Dataset statistics. 19,445 protein-ligand pairs

Dataset split. Random Split

References. [107]

Dataset license. See note in the description on the TDC website.

Code Sample

```
from tdc.generation import SBDD
data = SBDD(name='PDBBind', path='./pdbbind')
split = data.get_split()
```

DUD-E Dataset

DUD-E provides a directory of valuable decoys for protein-ligand docking.

Dataset statistics. 22,886 active compounds and affinities against 102 targets. DUD-E does not support pocket extraction as protein and ligand are not aligned.

Dataset split. Random Split

References. [109]

Dataset license. Not specified

Code Sample

```
from tdc.generation import SBDD
data = SBDD(name='dude')
split = data.get_split()
```

7.1.8 scPDB Dataset

scPDB is processed from PDB for structure-based drug design that identifies suitable binding sites for protein-ligand docking.

Dataset statistics. 16,034 protein-ligand pairs over 4,782 proteins and 6,326 ligands

Dataset split. Random Split

References. [108]

Dataset license. Not specified

Code Sample

```
from tdc.generation import SBDD
data = SBDD(name='scPDB')
split = data.get_split()
```

7.2 Equations

7.2.1 TDC.scDTI: Contextualized Drug-Target Nomination (Identification)

TDC-2 introduces TDC.scDTI task. The predictive, non-generative task is formalized as learning an estimator for a disease-specific function f of a target protein and cell type outputting whether the candidate protein t is a therapeutic target in that cell type c:

$$y = f(t; c): \tag{1}$$

Target candidate set. The target candidate set includes proteins, nucleic acids, or other molecules drugs can interact with, producing a therapeutic effect or causing a biological response. The target candidate set is constrained to proteins relevant to the disease being treated. It is denoted by:

$$\mathsf{T} = \{t_1; \ldots; t_{N_t}\}; \tag{2}$$

where $t_1; \ldots; t_{N_t}$ are N_t target candidates for the drugs treating the disease. Information modeled for target candidates can include interaction, structural, and sequence information.

Biological context set. The biological context set includes the cell-type-specific contexts in which the target candidate set operates. This set is denoted as:

$$\mathsf{C} = \{c_1; \ldots; c_{N_c}\}; \tag{3}$$

where $c_1; \ldots; c_{N_c}$ are N_c biological contexts on which drug-target interactions are being evaluated. Information modeled for cell-type-specific biological contexts can include gene expression and tissue hierarchy. The set is constrained to disease-specific cell types and tissues.

Drug-target identification. Drug-Target Identification is a binary label $y \in \{1; 0\}$, where $y = 1$ indicates the protein is a candidate therapeutic target. At the same time, 0 means the protein is not such a target.

The goal is to train a model f for predicting the probability $\hat{y} \in [0; 1]$ that a protein is a candidate therapeutic target in a specific cell type. The model learns an estimator for a disease-specific function of a protein target $t \in \mathsf{T}$ and a cell-type-specific biological context $c \in \mathsf{C}$ as input, and the model is tasked to predict:

$$\hat{y} = f\ (t \in \mathsf{T}; c \in \mathsf{C}): \tag{4}$$

7.2.2 `TDC.PerturbOutcome`: Perturbation-Response Problem Formulation

TDC-2 introduces Perturbation-Response prediction task. The predictive, non-generative task is formalized as learning an estimator for a function of the cell-type-specific gene expression response to a chemical or genetic perturbation, taking a perturbation $p \in \mathsf{P}$, a pre-perturbation gene expression profile from the control set $e_0 \in \mathsf{E}_0$, and the biological context $c \in \mathsf{C}$ under which the gene expression response to the perturbation is being measured:

$$y = f(p; e_0; c): \tag{5}$$

We center our definition on regression for the cell-type-specific gene expression vector in response to a chemical or genetic perturbation.

Perturbation set. The perturbation set includes genetic and chemical perturbations. It is denoted by:

$$\mathsf{P} = \{p_1; \ldots; p_{N_p}\}; \tag{6}$$

where $t_p; \ldots; p_{N_p}$ are N_p evaluated perturbations. Information modeled for genetic perturbations can include the type of perturbation (i.e., knockout, knockdown, overexpression) and target gene(s) of the perturbation. Information modeled for chemical perturbations can include chemical structure (i.e., SMILES, InChl) and concentration and duration of treatment.

Control set. The control set includes the unperturbed gene expression profiles. This set is denoted as:

$$\mathsf{E}_0 = \{e_{0_1}; \ldots; e_{N_{e_0}}\}; \tag{7}$$

where $e_{0_1}; \ldots; e_{N_{e_0}}$ are N_{e_0} unperturbed gene expression profile vectors. Information models for gene expression profiles can include raw or normalized gene expression counts, transcriptomic profiles, and isoform-specific expression levels.

Biological context set. The biological context set includes the cell-type-specific contexts under which the perturbed gene expression profile is measured. It is denoted by:

$$\mathsf{C} = \{c_1; \ldots; c_{N_c}\}; \tag{8}$$

where $c_1; \ldots; c_{N_c}$ are the N_c biological contexts under which perturbations are being evaluated. Information modeled for biological contexts can include cell type or tissue type and experimental conditions [69] as well as epigenetic markers [121, 122].

Perturbation-response readouts. Perturbation-Response is a gene expression vector e_1, where e_{1i} denotes the expression of the i-th gene in the vector. It is the outcome of applying a perturbation, $p_i \in \mathsf{P}$, within a biological context, $c_j \in \mathsf{C}$, to a cell with a measured control gene expression vector, $e_{0_k} \in \mathsf{E}_0$.

The Perturbation-Response Prediction learning task is to learn a regression model f estimating the perturbation-response gene expression vector $\hat{e_1}$ for a perturbation applied in a cell-type-specific biological context to a control:

$$\hat{e_1} = f\ (p \in \mathsf{P}; e_0 \in \mathsf{E}_0; c \in \mathsf{C}): \tag{9}$$

7.2.3 `TDC.ProteinPeptide`: Protein-Peptide Interaction Prediction Problem Formulation

TDC-2 introduces the Protein-Peptide Binding Affinity prediction task. The predictive, non-generative task is to learn a model estimating a function of a protein, peptide, antigen processing pathway, biological context, and interaction features. It outputs a binding affinity value (e.g., dissociation constant Kd, Gibbs free energy G) or binary label indicating strong or weak binding. The binary label can also include additional biomarkers, such as allowing for a positive label if and only if the binding interaction is specific [15, 123, 124]. To account for additional biomarkers beyond binding affinity value, our task is specified with a binary label.

Protein set. The protein set includes target proteins. It is denoted by:

$$\mathsf{P} = \{p_1; \ldots; p_{N_p}\}; \tag{10}$$

where $p_1; \ldots; p_{N_p}$ are N_p target proteins. Information modeled for proteins can include sequence, structural, or post-translational modification data.

Peptide set. The control set includes the peptide candidates. This set is denoted as:

$$\mathsf{S} = \{s_1; \ldots; s_{N_s}\}; \tag{11}$$

where $s_1, \ldots, s_{N_s}$ are N_s candidate peptides. Information modeled for candidate peptides can include sequence, structural, and physicochemical data.

Antigen processing pathway set. The antigen processing pathway set includes antigen processing pathway profile information about prior steps in the biological antigen presentation pathway processes. It is denoted by:

$$\mathsf{A} = \{a_1, \ldots, a_{N_a}\}, \tag{12}$$

where $a_1, \ldots, a_{N_a}$ are the N_a antigen processing pathway profiles modeled. Information modeled in a profile can include proteasomal cleavage sites [125], classification into viral, bacterial, and self-protein sources and endogenous vs exogenous processing pathway [126, 127, 84, 128], and target/receptor-specific pathway attributes such as transporter associated with antigen processing (TAP) affinity [129], and endosomal/lysosomal processing efficiency [130].

Interaction set. It contains the interaction feature profiles. The set is denoted by:

$$\mathsf{I} = \{i_1, \ldots, i_{N_i}\}, \tag{13}$$

where $i_1, \ldots, i_{N_i}$ are the N_i interaction feature profiles. Information modeled in an interaction feature profile can include contact maps [131, 132, 133, 134], distance maps [132, 135], electrostatic interactions [131], and hydrogen bonds [131].

Cell-type-specific biological context set. It contains the interaction feature profiles. The set is denoted by:

$$\mathsf{C} = \{c_1, \ldots, c_{N_c}\}, \tag{14}$$

where $c_1, \ldots, c_{N_c}$ are the N_c cell-type-specific biological contexts under which the protein-peptide interaction is being evaluated. Information modeled in the cell-type-specific biological context can include transcriptomic and proteomic data. We note, however, that, to our knowledge, single-cell transcriptomic and proteomic data has yet to be used in protein-peptide binding affinity prediction, outlining a promising avenue of research in developing machine learning models for peptide-based therapeutics.

Protein-peptide interaction. It is a binary label, $y \in \{1, 0\}$, where $y = 1$ indicates a protein-peptide pair met the target biomarkers and $y = 0$ indicates the pair did not meet the target biomarkers.

The Protein-Peptide Interaction Prediction learning task is to learn a binary classification model f estimating the probability, $\hat{y}$, of a protein-peptide interaction meeting specific biomarkers:

$$\hat{y} = f\ (p \in \mathsf{P}, s \in \mathsf{S}, a \in \mathsf{A}, i \in \mathsf{I}, c \in \mathsf{C}). \tag{15}$$

7.2.4 Clinical Trial Outcome Prediction Problem Formulation

The Clinical Trial Outcome Prediction task is formulated as a binary classification problem, where the machine learning model predicts whether a clinical trial will have a positive or negative outcome. It is a function that takes patient data, trial design, treatment characteristics, disease, and macro variables as inputs and outputs a trial outcome prediction, a binary indicator of trial success (1) or failure (0).

Patient set. The patient set includes one or multiple patient sub-populations, with the extreme case representing personalization. It is denoted as follows:

$$\mathsf{P} = \{p_1, \ldots, p_{N_p}\}, \tag{16}$$

where $p_1, \ldots, p_{N_p}$ are N_p patient sub-populations in this trial. The TOP benchmark [14] dataset represents patient data as part of the trial eligibility criteria. Patient data can include demographics [136, 137, 138, 139, 140], baseline health metrics [139, 140, 141], and medical history [136, 137, 138, 139, 140].

Trial design set. The trial design set includes this clinical trial's design profiles. It is denoted as:

$$\mathsf{D} = \{d_1, \ldots, d_{N_d}\}, \tag{17}$$

where $d_1, \ldots, d_{N_d}$ are N_d eligible trial design profiles for this clinical trial. Trial design profiles can model information including phase of the trial [14], number of participants, duration of the trial, trial eligibility criteria [14], and randomization and blinding methods [142, 143, 144].

Treatment set. The treatment set includes the candidate treatments for the trial. It is denoted as:

$$\mathsf{T} = \{t_1, \ldots, t_{N_t}\}, \tag{18}$$

where $t_1, \ldots, t_{N_t}$ are N_t candidate treatments for the clinical trial. The information modeled for treatments can include type of treatment (drug [14, 145], device [146, 147, 148], procedure [149, 150, 151, 152, 153]), dosage and administration route [142, 141, 154], mechanism of action [155, 156, 157], pre-clinical and early-phase trial results [156, 141, 158, 159].

Macro context set. The macro context set contains the configurations of macro variables relevant to the clinical trial. It is denoted as:

$$\mathsf{C} = \{c_1, \ldots, c_{N_c}\}, \tag{19}$$

where $c_1, \ldots, c_{N_c}$ are N_c configurations containing the values for macro variables relevant to the trial, which can include geography [160, 156, 159, 161] and regulatory considerations [156, 160].

Trial outcomes. The trial outcome is a binary label $y \in \{1, 0\}$, where $y = 1$ indicates the trial met their primary endpoints, while 0 means failing to meet with the primary endpoints.

The learning task is to learn a model f for predicting the trial success probability $\hat{y}$, where $\hat{y} \in [0, 1]$:

$$\hat{y} = f\ (p \in \mathsf{P}, d \in \mathsf{D}, t \in \mathsf{T}, c \in \mathsf{C}). \tag{20}$$

7.2.5 Structure-Based Drug Design Problem Formulation

Structure-based Drug Design aims to generate diverse, novel molecules with high binding affinity to protein pockets (3D structures) and desirable chemical properties. These properties are measured by oracle functions. A machine learning task first learns the molecular characteristics given specific protein pockets from a large set of protein-ligand pair data. Then, from the learned conditional distribution, we can sample novel candidates.

Target candidate set. The target candidate set includes proteins, nucleic acids, or other biomolecules drugs can interact with, producing a therapeutic effect or causing a biological response. It is denoted by:

$$\mathsf{T} = \{t_1, \ldots, t_{N_t}\}, \tag{21}$$

where $t_1, \ldots, t_{N_t}$ are N_t target candidates for the evaluated set of drugs. Information modeled for target candidates can include interaction, structural, and sequence information.

Ligand candidate set. The ligand drug candidate set includes the drug molecules being tested for a particular therapeutic effect or biological response. It is denoted by:

$$\mathsf{L} = \{l_1, \ldots, l_{N_l}\}, \tag{22}$$

where $l_1, \ldots, l_{N_l}$ are the N_l ligand/drug molecules being evaluated. Drug modeling can include molecular structure, often represented in formats such as SMILES (Simplified Molecular Input Line Entry System) or InChI (International Chemical Identifier) [162], physicochemical properties like hydrophobicity and molecular weight [80], and molecular descriptors and fingerprints [163].

Scoring function. The scoring function, denoted by S, evaluates the binding affinity of ligand $l \in \mathsf{L}$ to protein target $t \in \mathsf{T}$.

Drug-likeness function. Function representing the drug-likeness of ligand $l \in \mathsf{L}$, including properties like solubility, stability, and toxicity.

The generative learning task is to generate the ligand $l \in \mathsf{L}$ maximizing binding affinity, S , and drug-likeness, f . Given a loss function, $\text{Loss}(S(t, l), f(l))$, for $t \in \mathsf{T}$ and $l \in \mathsf{L}$, the first step is to learn a model M s.t.,

$$M = \text{argmin}\ [\text{Loss}(S\ (t, l), f\ (l))]. \tag{23}$$

This is followed by the ligand optimization step, which optimizes the ligand for maximum binding affinity and drug-likeness given the trained model. A ligand optimization function, F, such as addition or multiplication, is used for the optimization:

$$l^* = \text{argmax}_{l \in \mathsf{L}}[F(S\ (t \in \mathsf{T}, l), f\ (l))]. \tag{24}$$

An example formulation would be as follows:

$$l^* = \text{argmax}_{l \in \mathsf{L}}[S\ (t \in \mathsf{T}, l) \quad f\ (l)]. \tag{25}$$

7.2.6 Context-Specific Metrics

Context-specific metrics are defined to measure model performance at critical biological slices, with our benchmarks focused on measuring cell-type-specific model performance. For single-cell drug-target nomination, we measure model performance at top-performing cell types. The metrics chosen were: APR@5 Top-20 CT - average precision and recall at $k = 5$ for the 20 best-performing cell types (CT); AUROC Top-1 CT - AUROC for top-performing cell type; AUROC Top-10 CT and AUROC Top-20 CT - weighted average AUROC for top-10 and top-20 performing cell types, respectively, each weighted by the number of samples in each cell type. Formally, we define context-specific APR@5 and AUROC below.

Context-specific AUROC

To calculate the **AUROC for the top K performing cell types**, we first need to determine which cell types achieve the highest AUROC scores. After selecting the top-performing cell types, we weigh each top-performing cell type's AUROC score by the number of samples in that cell type.

We denote:

$$\mathsf{D} = f(x_i; y_i; c_i)g; \quad 8i \; 2 \; \mathsf{S} \tag{26}$$

Here, D denotes the dataset where x_i denotes the feature vector, y_i is the true label, and c_i is the cell type for sample i from S. We further denote C, the set of unique cell types. Then, the AUROC for a specific cell type, $AUROC_c$, is computed as:

$$AUROC_c = AUROC(D_c) \tag{27}$$

Here, $D_c = f(x_i; y_i) j c_i = cg$ is the subset of the dataset for cell type c and $AUROC(D_c)$ represents the AUROC score computed over this subset. Once these are computed, values can be sorted in descending order to select the top X cell type with highest AUROC value.

$$C_K = fc_1; c_2; : : : ; c_K g \quad s:t: \quad AUROC_{c_i} \quad AUROC_{c_j}; 8i \quad K; j > K \tag{28}$$

The weighted AUROC for the top K cell types is given by weighting each cell type's AUROC by the proportion of its samples relative to the total samples in the top K cell types.

$$AUROC_{TopK} = \frac{\sum_{c \in C_K} AUROC_c \quad j D_c j}{\sum_{c \in C_K} j D_c j} \tag{29}$$

This measure represents a balance between representation and performance of the cell types.

Context-specific Average Precision at rank R (AP@R)

In our study, we let $R = 5$ and compute **AP@5 for the top K performing cell types**. We denote dataset and samples as above.

$$\mathsf{D} = f(x_i; y_i; c_i)g; \quad 8i \; 2 \; \mathsf{S} \tag{30}$$

Here, D denotes the dataset where x_i denotes the feature vector, y_i is the true label, and c_i is the cell type for sample i from S. We further denote C, the set of unique cell types. The samples of each cell type, $D_c = (x_i; y_i) j c_i = c$, can be sorted based on the score output by the model for said sample $f(x_i)$, with average precision at rank type computed accordingly.

$$D_c^5 = fx_1; : : : ; x_5 g \quad s:t: \quad f(x_i) \quad f(x_j); 8i \quad 5; j > 5; c_i = c; c_j = c \tag{31}$$

$$AP@5_c = AP(fy_1; : : : ; y_5 g; f f(x_1); : : : ; f(x_5)g); \quad x_i \; 2 \; D_c^5 \tag{32}$$

The average precision at rank k at Top X cell types can then be defined as:

$$C_K = fc_1; c_2; : : : ; c_k g \quad s:t: \quad AP@5_{c_i} \quad AP@5_{c_j}; 8i \quad K; j > K \tag{33}$$

$$AP@5_{TopK} = mean(fAP@5_{c_i} g; \quad 8c_i \; 2 \; C_K) \tag{34}$$

AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight. Some key advantages of using AP@K include robustness to (1) varied numbers of protein targets activated across cell type-specific protein interaction networks and (2) varied sizes of cell type-specific protein interaction networks [4]. We compute AP using the scikit package as specified in `https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.average_precision_score.html`.

7.3 Algorithms, Program codes and Listings

We provide code samples for the components described in sections 6.3.1 and 6.3.2.

Listing 2: The above configuration augments a protein-peptide dataset with an additional modality, amino acid sequence, and invokes numerous data processing functions tailored to the specific needs of the underlying dataset. Added information for this demonstration can be found at: `https://colab.research.google.com/drive/13MYlg5tWpywWbKYsJQXafKAlVF2hz-sP?usp=sharing`. There are more complex workflows implemented for current TDC dataviews and all such views leveraging the DSL can be found in the repo at `https://github.com/mims-harvard/TDC/blob/main/tdc/dataset_configs/config_map.py`

```
from .config import DatasetConfig
from ..feature_generators.protein_feature_generator import
    ,! ProteinFeatureGenerator


class BrownProteinPeptideConfig(DatasetConfig):
    """Configuration for the brown-protein-peptide datasets"""

    def __init__(self):
        super(BrownProteinPeptideConfig, self).__init__(
            dataset_name="brown_mdm2_ace2_12ca5",
            data_processing_class=ProteinFeatureGenerator,
            functions_to_run=[
                "autofill_identifier", "create_range", "
                    ,! insert_protein_sequence"
            ],
            args_for_functions=[{
                "autofill_column": "Name",
                "key_column": "Sequence",
            }, {
                "column": "KD␣(nM)",
                "keys": ["Putative␣binder"],
                "subs": [0]
            }, {
                "gene_column": "Protein␣Target"
            }],
            var_map={
                "X1": "Sequence",
                "X2": "protein_or_rna_sequence",
                "ID1": "Name",
                "ID2": "Protein␣Target",
            },
        )
```

7.3.1 TDC-2 Multimodal Single-Cell Retrieval API

We focus on the use case of an ML researcher who wishes to train a model on a large-scale single-cell atlas. In particular, researchers would be familiar with and have trained models on traditional single-cell datasets such as Tabula Sapiens [51]. Their interest is to scale a model by training it on a more extensive single-cell atlas based on this reference dataset. We build such an API. Specifically, given a reference dataset available in CellXGene Discover [2], we allow the user to perform a memory-efficient query using TileDB-SOMA to expand the reference dataset to include cell entries with non-zero readouts for any of the genes present in the reference dataset. This allows users to build large-scale single-cell atlases on familiar reference datasets. The example below illustrates how a user may construct a large-scale atlas with Tabula Sapiens as the reference dataset. Other use cases include augmenting datasets using knowledge graphs and cell-type-specific biomedical contexts. These capabilities are all powered by the Model-View-Controller framework (section 6.3.1).

Listing 3: The example below illustrates how a user may construct a large-scale atlas with Tabula Sapiens as the reference dataset using the TDC-2 CELLXGENE API.

```
from tdc.multi_pred.single_cell import CellXGene
dataloader = CellXGene(name="Tabula␣Sapiens␣-␣All␣Cells")
gen = dataloader.get_data(
    value_filter="tissue␣==␣'brain'␣and␣sex␣==␣'male'"
)
df = next(gen)
```

Listing 4: In addition to our TDC-2 DataLoader API implementation for the CellXGene RPC API, we provide a simplified wrapper over the CellXGene Census Discovery API, which allows users to perform remote procedure calls to fetch Cell Census data in more machine-learning-friendly formats like Pandas and Scipy. We also maintain support for the AnnData format. Users can query Cell Census counts as well as metadata using this API. The code sample below illustrates such usage.

```
from tdc.resource import cellxgene_census

# initialize Census Resource and query filters
resource = cellxgene_census.CensusResource()
cell_value_filter = "tissue␣==␣'brain'␣and␣sex␣==␣'male'"
cell_column_names = ["assay", "cell_type", "tissue"]

# Obtaining cell metadata from the cellxgene census in pandas format
obsdf = resource.get_cell_metadata(
    value_filter=cell_value_filter,
    column_names=cell_column_names,
    fmt="pandas")
```

7.3.2 PrimeKG Knowledge Graph

PrimeKG supports drug-disease prediction by including an abundance of 'indications,' 'contradictions', and 'off-label use' edges, which are usually missing in other knowledge graphs. We accompany PrimeKG's graph structure with text descriptions of clinical guidelines for drugs and diseases to enable multimodal analyses [22]. The code below depict example use cases of the TDC-2 PrimeKG API. Demonstrations are additionally available in `https://colab.research.google.com/drive/1kYH8nt3nW7tXYBPNcfYuDbWxGTqOEnWg?usp=sharing`.

Listing 5: We illustrate here example utilities for retrieving drug-target-disease associations using the TDC-2 PrimeKG API

```
from tdc.resource import PrimeKG

pkg = PrimeKG()
pkgdf = pkg.get_data()

def get_all_drug_evidence(disease):
    """given a disease, retrieve all drugs interacting with proteins
        ,! relevant to disease"""
    prots = pkgdf[(pkgdf["relation"] == "disease_protein") & (pkgdf["x_name"
        ,! ] == disease)]["y_name"].unique()
    drugs = pkgdf[(pkgdf["relation"] == "drug_protein") & (pkgdf["y_name"].
        ,! isin(prots))]
    relations = drugs["display_relation"].unique()
    out = {}
    for rel in relations:
        out[rel] = drugs[drugs["display_relation"] == rel]["x_name"].unique
            ,! ()
    return out
```

```
def get_all_associated_targets(disease):
    return pkgdf[(pkgdf["relation"] == "disease_protein") & (pkgdf["x_name"]
        ,!  == disease)][["y_name", "display_relation"]]

def get_disease_disease_associations(disease):
    return pkgdf[(pkgdf["relation"] == "disease_disease") & (pkgdf["x_name"]
        ,!  == disease)][["y_name", "display_relation"]]

def get_labels_from_evidence(disease):
    diseases = get_disease_disease_associations(disease)["y_name"]
    out = set()
    for d in diseases:
        targets = get_all_associated_targets(d)["y_name"].unique()
        out.update(targets)
    return list(out)

def all_diseases_by_keyword(kw):
    return pkgdf[(pkgdf["relation"] == "disease_protein") & (pkgdf["x_name"
        ,! ].str.contains(kw, case=False, na=False))]["x_name"].unique()

if __name__ == "__main__":
    x = all_diseases_by_keyword("autism")
    [get_all_drug_evidence(d) for d in x]
    [get_all_associated_targets(d) for d in x]
    [get_disease_disease_associations(d) for d in x]
    print([get_labels_from_evidence(d) for d in x])
```

Listing 6: Here we illustrate combinig the TDC-2 PrimeKG API with the networkx module to retrieve drug repositioning opportunities.

```
import networkx as nx
from tdc.resource import PrimeKG

# Load the PrimeKG data
kg = PrimeKG()
data = kg.get_data()
data = data[data["relation"].str.contains("drug")]

# Create a graph from the knowledge graph data
G = nx.from_pandas_edgelist(data, 'x_id', 'y_name', edge_attr='relation')

# Example function to find repositioning opportunities for a given drug
def find_repositioning_opportunities(drug):
    neighbors = list(G.neighbors(drug))
    diseases = [node for node in neighbors if G[drug][node]['relation'] == '
        ,! drug_protein']
    return diseases

# Find repositioning opportunities for a specific drug
drug_name = 'DB00945'
repositioning_opportunities = find_repositioning_opportunities(drug_name)
```

7.3.3 TDC-2 Model Server

The introduced model server is composed of the TDC-2 Model Hub and a set of utilities and endpoints for facilitating model inference and fine-tuning. TDC-2 introduces The Commons' HuggingFace Model Hub. It is a resource with pre-trained models, including geometric deep learning models, large language models, and other contextualized multimodal models for therapeutic tasks. The models can be fine-tuned using datasets in TDC-2 and be used for downstream tasks such as

implementations of multi-agent collaborative schemes [164] (i.e., expert consultants) our predictive therapeutic tasks [3, 19]. The model hub details and available models can be found at `https://huggingface.co/tdc`.

Listing 7: The below illustrates the basic functionality of the model hub to download a model and perform inference on a precdictive task as well as fine-tune the model

```
from tdc import tdc_hf_interface
tdc_hf = tdc_hf_interface("BBB_Martins-AttentiveFP")
# load deeppurpose model from this repo
dp_model = tdc_hf.load_deeppurpose('./data')
tdc_hf.predict_deeppurpose(dp_model, ['YOUR SMILES STRING'])
# fine-tune
dp_model.train(train, val, test) # for some defined splits
```

Listing 8: The below illustrates using the tdc model hub to download a foundation model [3]

```
from tdc import tdc_hf_interface
from transformers import BertModel
geneformer = tdc_hf_interface("Geneformer")
model = geneformer.load()
assert isinstance(model, BertModel), type(model)
```

Listing 9: Beyond downloading a foundation model [3], the model server facilitates model inference across a range of datasets. Below an example integrating the TDC-2 CellXGene API with the model server.

```
from tdc.resource import cellxgene_census
from tdc.model_server.tokenizers.geneformer import GeneformerTokenizer
from tdc import tdc_hf_interface
import torch

# query the CELLXGENE census
adata = self.resource.get_anndata(
var_value_filter=
"feature_id in ['ENSG00000161798', 'ENSG00000188229']",
obs_value_filter=
"sex == 'female' and cell_type in ['microglial cell', 'neuron']",
column_names={
    "obs": [
        "assay", "cell_type", "tissue", "tissue_general",
        "suspension_type", "disease"
    ]
},
)

# tokenize gene expression vectors
tokenizer = GeneformerTokenizer()
x = tokenizer.tokenize_cell_vectors(adata,
                                    ensembl_id="feature_id",
                                    ncounts="n_measured_vars")

cells, _ = x

# load the model
geneformer = tdc_hf_interface("Geneformer")
model = geneformer.load()

"""
Custom pre-processing code can include padding and attention mask
    ,! definitions.
```

```
"""

input_tensor = torch.tensor(cells)
out = []
for batch in input_tensor:
    # build an attention mask
    attention_mask = torch.tensor(
        [[x[0] != 0, x[1] != 0] for x in batch])
    # run batched inference
    out.append(model(batch, attention_mask=attention_mask))
```

7.3.4 Running TDC-2 Benchmarks

We provide code for replicating all introduced benchmarks and testing other model performance on all TDC-2 tasks. We include here snippets for all introduced benchmarks.

Listing 10: The below code illustrates how to retrieve the train, test, and val splits used for the TDC.scDTI benchmark

```
from tdc.benchmark_group import scdti_group
group = scdti_group.SCDTIGroup()
train_val = group.get_train_valid_split()
tst = group.get_test()["test"]
# train your model and test on the test set
group.evaluate(preds)
```

Listing 11: The below code illustrates how to retrieve the train, test, and val splits used for the TDC.PerturbOutcome chemical perturbation benchmark

```
from tdc.benchmark_group import counterfactual_group
group = counterfactual_group.CounterfactualGroup()
train, val = group.get_train_valid_split(remove_unseen=False)
test = group.get_test()
# train your model and test on the test set
group.evaluate(preds)
```

Listing 12: The below code illustrates how to retrieve the train, test, and val splits used for the TDC.PerturbOutcome genetic perturbation benchmark

```
from tdc.benchmark_group import geneperturb_group
group = geneperturb_group.GenePerturbGroup()
train_val = group.get_train_valid_split()
test = group.get_test()
# train your model and test on the test set
group.evaluate(preds)
```

Listing 13: The below code illustrates how to retrieve the train, test, and val splits used for the TDC.TCREpitope benchmark

```
from tdc.benchmark_group.tcrepitope_group import TCREpitopeGroup
group = TCREpitopeGroup()
train_val = group.get_train_valid_split()
test = group.get_test()
# train your model and test on the test set
group.evaluate(preds)
```

7.3.5 External Links - Reproducibility

Here, we include pointers to external resources to reproduce the results reported in this manuscript.

Model Benchmarking

TDC-2 Benchmarking Tooling Code for Chemical Perturbations, https://github.com/mims-harvard/TDC/blob/main/tdc/benchmark_group/counterfactual_group.py

TDC-2 Benchmarking Tooling Code for CRISPR-based Perturbations, https://github.com/mims-harvard/TDC/blob/main/tdc/benchmark_group/geneperturb_group.py

Reproducing Benchmark Results for Clinical Trial Outcome Prediction, https://github.com/futianfan/clinical-trial-outcome-prediction

Evaluating Cell-Type-Specific Context Metrics for PINNACLE Across 10 Seeds, https://colab.research.google.com/drive/1gjZIfmF2Gmz3Nqm1uGP79l0AmsPAvj_5?usp=sharing

Evaluating Cell-Type-Specific Context Metrics for PINNACLE Across 10 Seeds. Outputs Referenced in PINNACLE [4] and its reproducibility documentation, https://drive.google.com/drive/folders/1QX05afMekucbtjl_07ZxZhgnKVH3OXMk?usp=sharing

Code for reproducing PINNACLE results [4], https://github.com/mims-harvard/PINNACLE/tree/main/evaluate

Reproducing TCR-Epitope results. Code for Benchmarking models in Section 3.3.1. A bash script for each negative sampling method is included for each TCR-Epitope model, https://drive.google.com/drive/folders/1O7G_h_06VDABM6U_Xt7otXPazK0XTAG9?usp=sharing

Reproducing Chemical Perturbation results. Code for Benchmarking models in Section 3.2.2 chemical perturbation section. A run_chemical_sc.py Python script is included for each model. Default settings were used from each model's GitHub repository, https://drive.google.com/drive/folders/1RlBnRPmWFRQ6M_1EQ_FMwFb1Y8IjXoyC?usp=sharing

Leaderboards

TDC.PerturbOutcome Leaderboard, https://tdcommons.ai/benchmark/counterfactual_group/overview/

TDC.ProteinPeptide Leaderboard, https://tdcommons.ai/benchmark/proteinpeptide_group/overview/

TDC.scDTI Leaderboard, https://tdcommons.ai/benchmark/scdti_group/overview/

Acknowledgments and Disclosure of Funding

We thank Zitnik Lab, Pentelute Lab, and Kellis Lab members for their constructive input on the manuscript and contributions to TDC-2 datasets, benchmarks, and experiments.

We gratefully acknowledge the support of NIH R01-HD108794, NSF CAREER 2339524, US DoD FA8702-15-D-0001, awards from Harvard Data Science Initiative, Amazon Faculty Research, Google Research Scholar Program, AstraZeneca Research, Roche Alliance with Distinguished Scientists, Sanofi iDEA-iTECH Award, Pfizer Research, Chan Zuckerberg Initiative, John and Virginia Kaneb Fellowship award at Harvard Medical School, Aligning Science Across Parkinson's (ASAP) Initiative, Biswas Computational Biology Initiative in partnership with the Milken Institute, Harvard Medical School Dean's Innovation Awards for the Use of Artificial Intelligence, and Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. M.M.L. is supported by T32HG002295 from the National Human Genome Research Institute and a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

References

[1] Malte D. Luecken, Scott Gigante, Daniel B. Burkhardt, Robrecht Cannoodt, Daniel C. Strobl, Nikolay S. Markov, Luke Zappia, Giovanni Palla, Wesley Lewis, Daniel Dimitrov, Michael E. Vinyard, D.S. Magruder, Alma Andersson, Emma Dann, Qian Qin, Dominik J. Otto, Michal Klein, Olga Borisovna Botvinnik, Louise Deconinck, Kai Waldrant, Open Problems Jamboree Members, Jonathan M. Bloom, Angela Oliveira Pisco, Julio Saez-Rodriguez, Drausin Wulsin, Luca Pinello, Yvan Saeys, Fabian J. Theis, and Smita Krishnaswamy. Defining and benchmarking open problems in single-cell analysis. *Research Square Preprint*, 2023.

[2] CZI Single-Cell Biology, et al. Cz cellxgene discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. bioRxiv Preprint, 2023.

[3] Christina Theodoris, Ling Xiao, Anant Chopra, Mark Chaffin, Zeina Sayed, Matthew Hill, Helene Mantineo, Elizabeth Brydon, Zexian Zeng, Shirley Liu, and Patrick Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618:1–9, 05 2023.

[4] Michelle M Li, Yepeng Huang, Marissa Sumathipala, Man Qing Liang, Alberto Valdeolivas, Ashwin N Ananthakrishnan, Daniel Marbach, and Marinka Zitnik. Contextualizing protein representations using deep learning on protein networks and single-cell data. *bioRxiv*, 2023.

[5] Yasha Ektefaie, Andrew Shen, Daria Bykova, Maximillian Marin, Marinka Zitnik, and Maha Farhat. Evaluating generalizability of artificial intelligence models for molecular datasets. *bioRxiv*, 2024.

[6] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Phillipe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 2024.

[7] Phil Bradley. Structure-based prediction of t cell receptor:peptide-mhc interactions. *eLife*, 12, 2022.

[8] Jean-Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48(3):722–730, 2015.

[9] Michael S Kinch, Zachary Kraft, and Tyler Schwartz. 2023 in review: Fda approvals of new medicines. *Drug discovery today*, page 103966, 2024.

[10] Juan Jose Garau-Luis, Patrick Bordes, Liam Gonzalez, Masa Roller, Bernardo P. de Almeida, Lorenz Hexemer, Christopher Blum, Stefan Laurent, Jan Grzegorzewski, Maren Lang, Thomas Pierrot, and Guillaume Richard. Multi-modal transfer learning between biological foundation models, 2024.

[11] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development, 2021.

[12] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Artificial intelligence foundation for therapeutic science. *Nature chemical biology*, 18(10):1033–1036, 2022.

[13] Sandra Romero-Molina, Yasser B. Ruiz-Blanco, Joel Mieres-Perez, M. Harms, J. Münch, M. Ehrmann, and E. Sánchez-García. Ppi-affinity: A web tool for the prediction and optimization of protein–peptide and protein–protein binding affinity. *Journal of Proteome Research*, 21:1829 – 1841, 2022.

[14] Tianfan Fu, Kexin Huang, Cao Xiao, Lucas M. Glass, and Jimeng Sun. Hint: Hierarchical interaction network for clinical-trial-outcome predictions. *Patterns*, 3(4):100445, Feb 2022. eCollection 2022 Apr 8.

[15] Unsupervised machine learning leads to an abiotic picomolar peptide ligand. May 2023. License CC BY-NC-ND 4.0.

[16] Che Ngufor, H. Houten, B. Caffo, N. Shah, and R. McCoy. Mixed effect machine learning: A framework for predicting longitudinal change in hemoglobin a1c. *Journal of Biomedical Informatics*, 89:56–67, 2019.

[17] Yanbin Zhu, Xin Ouyang, Peilin Li, Quan Jin, Jun Su, Lirong Zheng, and Li Ye. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *Journal of Chemical Information and Modeling*, 62(9):2204–2212, 2022.

[18] Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. Peer: A comprehensive and multi-task benchmark for protein sequence understanding, 2022. Accepted by NeurIPS 2022 Dataset and Benchmark Track. arXiv v2: source code released; arXiv v1: release all benchmark results.

[19] Chan Zuckerberg Initiative. Embedding metrics in the december 2023 lts, 2023. Accessed: 2024-09-14.

[20] Timo Schick, Helmut Schmid, and Hinrich Schütze. Toolformer: Language models can teach themselves to use tools. In *Proceedings of the 2023 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2023.

[21] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis. *ArXiv*, abs/2305.15334, 2023.

[22] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023.

[23] Raphael Thiago, Renan Souza, L. Azevedo, E. Soares, Rodrigo Santos, Wallas Santos, Max De Bayser, M. Cardoso, M. Moreno, and Renato Cerqueira. Managing data lineage of og machine learning models: The sweet spot for shale use case, 2020.

[24] Tom van der Weide, Dimitris Papadopoulos, Oleg Smirnov, Michal Zielinski, and Tim van Kasteren. Versioning for end-to-end machine learning pipelines. In *Proceedings of the 1st Workshop on Data Management for End-to-End Machine Learning*, DEEM'17, New York, NY, USA, 2017. Association for Computing Machinery.

[25] Huiting Liu, Avinesh P.V.S, Siddharth Patwardhan, Peter Grasch, and Sachin Agarwal. Model stability with continuous data updates, 2022.

[26] Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, Aug 2023. Open access.

[27] L. Hetzel, S. Böhm, N. Kilbertus, S. Günnemann, M. Lotfollahi, and F. Theis. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *arXiv*, abs/2204.13545, 2022.

[28] Filippo Grazioli, Pierre Machart, Anja Mösch, Kai Li, L. Castorina, N. Pfeifer, and Martin Renqiang Min. Attentive variational information bottleneck for tcr–peptide interaction prediction. *Bioinformatics*, 39, 2022.

[29] Jin Joo Kwon, Jie Pan, Gabriela Gonzalez, William C. Hahn, and Marinka Zitnik. On knowing a gene: A distributional hypothesis of gene function. *Cell Systems*, 2024.

[30] Ha Young Kim, Sungsik Kim, Woong-Yang Park, and Dongsup Kim. Tspred: a robust prediction framework for tcr-epitope interactions based on an ensemble deep learning approach using paired chain tcr sequence data. *bioRxiv*, 2023.

[31] Z. Piran, Niv Cohen, Yedid Hoshen, and M. Nitzan. Biological representation disentanglement of single-cell data. *bioRxiv*, 2023.

[32] Hengshi Yu and Joshua D. Welch. Perturbnet predicts single-cell responses to unseen chemical and genetic perturbations. *bioRxiv*, 2022.

[33] Prathamesh P. Churi, Sharad Wagh, Deepa Kalelkar, and M. Kalelkar. Model-view-controller pattern in bi dashboards: Designing best practices. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 2082–2086, 2016.

[34] Miquel Duran-Frigola, Eduardo Pauls, Oriol Guitart-Pla, Martino Bertoni, Víctor Alcalde, David Amat, Teresa Juan-Blanco, and Patrick Aloy. Chemicalcheckr: Extending the small molecule similarity principle to all levels of biology. *Nature Biotechnology*, 38:1087–1096, 2020.

[35] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2018.

[36] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, J. Canny, P. Abbeel, and Yun S. Song. Evaluating protein transfer learning with tape. *bioRxiv*, 2019.

[37] Malte D Luecken, Scott Gigante, Daniel B Burkhardt, Robrecht Cannoodt, Daniel C Strobl, Nikolay S Markov, Luke Zappia, Giovanni Palla, Wesley Lewis, Daniel Dimitrov, et al. Defining and benchmarking open problems in single-cell analysis. *Research Square*, 2024.

[38] Xiaoxiao Li, Shujie Wang, Jian Zhang, and Rui Zhang. Torchprotein: A deep learning library for protein sequence and structure modeling. *Bioinformatics*, 38(6):1743–1745, 2022.

[39] Benedek Rozemberczki, Charles Tapley Hoyt, Alexandra Gogleva, Piotr Grabowski, Klas Karis, Andrej Lamov, Andrey Nikolov, Sebastian Nilsson, Massimiliano Ughetto, Yu Wang, Tyler Derr, and Benjamin M. Gyori. Chemicalx: A deep learning library for drug pair scoring. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[40] Kexin Huang, Tianfan Fu, Lucas Glass, Marinka Zitnik, Cao Xiao, and Jimeng Sun. Deeppurpose: a deep learning library for drug–target interaction prediction. *Bioinformatics*, 36(22):5545–5547, 2020.

[41] Juan Manuel Zambrano Chaves, Eric Wang, Tao Tu, Eeshit Dhaval Vaishnav, Byron Lee, S. Sara Mahdavi, Christopher Semturs, David Fleet, Vivek Natarajan, and Shekoofeh Azizi. Tx-llm: A large language model for therapeutics. *arXiv preprint arXiv:2406.06316*, 2024.

[42] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

[43] Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, pages 1–11, 2024.

[44] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, pages 1–11, 2024.

[45] Yihang Xiao, Jinyi Liu, Yan Zheng, Xiaohan Xie, Jianye Hao, Mingzhi Li, Ruitao Wang, Fei Ni, Yuxiao Li, Jintian Luo, et al. Cellagent: An llm-driven multi-agent framework for automated single-cell data analysis. *bioRxiv*, 2024. Preprint.

[46] Nicole Beaulieu, Sergiu Dascalu, and Emily Hand. Api integrator: A ui design and code automation application supporting api-first design. In *Proceedings of the 9th International Conference on Applied Computing & Information Technology*, 2022.

[47] Martin Reddy. *API Design for C++*. Elsevier, 2011.

[48] M. Lipchanskyi and O. O. Iliashenko. code first design first api (comparison of code first and design first approaches in api development). *Science and Education a New Dimension. Natural and Technical Sciences*, 4:51–54, 2020.

[49] Qian Zhang, Felix X. Yu, Yanlin Wu, et al. Novel gene therapy for rheumatoid arthritis with single local injection: adeno-associated virus-mediated delivery of a20/tnfaip3. *Military Medical Research*, 9(1):34, 2022.

[50] Vincent S. Chen, Sen Wu, Zhenzhen Weng, Alexander Ratner, and Christopher Ré. Slice-based learning: A programming model for residual learning in critical data slices, 2020.

[51] Robert C. Jones, Jim Karkanias, Mark Krasnow, Angela Pisco, Stephen Quake, Julia Salzman, Nir Yosef, Bryan Bulthaup, Patrick Brown, William Harper, Marisa Hemenez, Ramalingam Ponnusamy, Ahmad Salehi, Bhavani A. Sanagavarapu, Eileen Spallino, Ksenia A. Aaron, Waldo Concepcion, Jennifer Gardner, Brian Kelly, Nicole Neidlinger, Zifa Wang, Sheela Crasta, Saroja Kolluru, Maurizio Morri, Serena Y. Tan, Katherine Travaglini, Chenling A. Xu, Mar Alcántara-Hernández, Natalia Almanzar, Jane Antony, Benjamin Beyersdorf, Deviana Burhan, Kruti Calcuttawala, Matthew M. Carter, Charles K. F. Chan, Charles A. Chang, Stephen Chang, Andrea Colville, Rebecca Culver, Ivana Cvijovic, Gaetano D'Amato, Camille Ezran, Francisco X. Galdos, Andre Gillich, William Goodyer, Yuxuan Hang, Alyssa Hayashi, Shahin Houshdaran, Xianxi Huang, Jeremy Irwin, SoRi Jang, Julia Vallve Juanico, Aaron M. Kershner, Soochi Kim, Bence Kiss, Winson Kong, Maya E. Kumar, Andrew Kuo, Rebecca Leylek, Baoxiang Li, Gabriel B. Loeb, Wan-Jin Lu, Sruthi Mantri, Maxim Markovic, Patrick L. McAlpine, Antoine de Morrée, Khedidja Mrouj, Shravani Mukherjee, Tyler Muser, Patrick Neuhöfer, Tam D. Nguyen, Kim Perez, Ragini Phansalkar, Natasha Puluca, Zhen Qi, Poorvi Rao, Hayley M. Raquer-McKay, Nicole Schaum, Bronwyn Scott, Bobak Seddighzadeh, Jonathan Segal, Sushmita Sen, Shaheen S. Sikandar, Stephanie Spencer, Lauren Steffes, Vishwanath Subramaniam, Aditi Swarup, Michael Swift, William W. Van Treuren, Emily Trimm, Stefan Veizades, Swathi Vijayakumar, Kevin C. Vo, Samantha Vorperian, Wanxin Wang, Hannah N. Weinstein, Juliane Winkler, Timothy Wu, Jamie Xie, Andrew Yung, Yue Zhang, Andrea Detweiler, Honey E. Mekonen, Norma Neff, Robert Sit, Michelle Tan, Jiacheng Yan, Gregory Bean, V. Charu, Erna Forgó, Barbara A. Martin, Michael Ozawa, Oscar Silva, Andrea Toland, Venkata N. P. Vemuri, Shaked Afik, Kyle Awayan, Oleg Botvinnik, Adam Byrne, Michelle Chen, Roozbeh Dehghannasiri, Adam Gayoso, Alejandro A. Granados, Qiqing Li, Gita Mahmoudabadi, Alexandra McGeever, Jaclyn Olivieri, Madeline Park, Nitin Ravikumar, Geoffrey M. Stanley, Wei Tan, Alexander J. Tarashansky, Rohan Vanheusden, Peter L. Wang, Sheng Wang, Galen Xing, Rebecca Culver, Les Dethlefsen, Po-yi Ho, Shixuan Liu, Jordan Maltzman, Ryan Metzger, Koki Sasagawa, Rahul Sinha, Hanbing Song, Bruce Wang, Steven Artandi, Philip Beachy, Michael Clarke, Linda Giudice, Fred Huang, Kerwyn C. Huang, Juliana Idoyaga, Seung K. Kim, Mark Krasnow, Connie Kuo, Patricia Nguyn, Thomas Rando, Kavitha Red-Horse, Jeremy Reiter, David Relman, Justin Sonnenburg, Albert Wu, Sean M. Wu, and Tony Wyss-Coray. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376, 2022.

[52] Y. Li, Xiao-zhang Liu, Zhuhong You, Liping Li, Jianping Guo, and Zheng Wang. A computational approach for predicting drug–target interactions from protein sequence and drug substructure fingerprint information. *International Journal of Intelligent Systems*, 36:593 – 609, 2020.

[53] Yang-Ming Li, Yu-An Huang, Zhuhong You, Liping Li, and Zheng Wang. Drug-target interaction prediction based on drug fingerprint information and protein sequence. *Molecules*, 24, 2019.

[54] Ingoo Lee, Jongsoo Keum, and Hojung Nam. Deepconv-dti: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Computational Biology*, 15, 2018.

[55] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, Sep 2018.

[56] Fan-Rong Meng, Zhu-Hong You, Xing Chen, Yong Zhou, and Ji-Yong An. Prediction of drug–target interaction networks from the integration of protein sequences and drug chemical structures. *Molecules*, 22(7), 2017.

[57] Yanrong Ji, Rama K. Mishra, and R. Davuluri. In silico analysis of alternative splicing on drug-target gene interactions. *Scientific Reports*, 10, 2020.

[58] Mohamed A. Ghadie, L. Lambourne, M. Vidal, and Yu Xia. Domain-based prediction of the human isoform interactome provides insights into the functional impact of alternative splicing. *PLoS Computational Biology*, 13, 2017.

[59] Jie Zeng, Guoxian Yu, Jun Wang, Maozu Guo, and Xiangliang Zhang. Dmil-iii: Isoform-isoform interaction prediction using deep multi-instance learning method. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 171–176, 2019.

[60] Jun Wang, Long Zhang, An Zeng, Dawen Xia, Jiantao Yu, and Guoxian Yu. Deepiii: Predicting isoform-isoform interactions by deep neural networks and data fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19:2177–2187, 2021.

[61] Konstantin Carlberg, M. Korotkova, L. Larsson, A. Catrina, Patrik L. Ståhl, and V. Malmström. Exploring inflammatory signatures in arthritic joint biopsies with spatial transcriptomics. *Scientific Reports*, 9, 2019.

[62] B. Kuenzi, Jisoo Park, Samson H. Fong, Kyle S. Sanchez, John Lee, J. Kreisberg, Jianzhu Ma, and T. Ideker. Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer cell*, 2020.

[63] H. Julkunen, A. Cichońska, Prson Gautam, S. Szedmák, Jane Douat, T. Pahikkala, T. Aittokallio, and Juho Rousu. Leveraging multi-way interactions for systematic prediction of pre-clinical drug combination effects. *Nature Communications*, 11, 2020.

[64] L. Parca, G. Pepe, M. Pietrosanto, G. Galvan, Leonardo Galli, Antonio Palmeri, M. Sciandrone, F. Ferrè, G. Ausiello, and M. Helmer-Citterich. Modeling cancer drug response through drug-specific informative genes. *Scientific Reports*, 9, 2019.

[65] Shilu Zhang, Saptarshi Pyne, Stefan J. Pietrzak, S. Halberg, S. McCalla, Alireza F. Siahpirani, Rupa Sridharan, and Sushmita Roy. Inference of cell type-specific gene regulatory networks on cell lineages from single cell omic datasets. *Nature Communications*, 14, 2023.

[66] Chirag Gupta, Jielin Xu, Ting Jin, Saniya Khullar, Xiaoyu Liu, Sayali Alatkar, F. Cheng, and Daifeng Wang. Single-cell network biology characterizes cell type gene regulation for drug repurposing and phenotype prediction in alzheimer's disease. *PLoS Computational Biology*, 18, 2022.

[67] Open Targets. Open targets platform: Ra and ibd disease drug targets, 2023. Accessed: 2024-05-21.

[68] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, P. Lio', and Yoshua Bengio. Graph attention networks. *ArXiv*, abs/1710.10903, 2017.

[69] Stefan Peidli, Tessa D. Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J. Schumacher, Jake P. Taylor-King, Debora S. Marks, Augustin Luna, Nils Blüthgen, and Chris Sander. scperturb: harmonized single-cell perturbation data. *Nature Methods*, 21:531–540, Jan 2024.

[70] Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

[71] Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Marco Jost, Thomas M. Norman, Jonathan S. Weissman, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. Open access.

[72] M. Lotfollahi, F.A. Wolf, and Fabian J. Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16:715–721, 2019.

[73] Sanjay R. Srivatsan, José L. McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A. Pliner, Dana L. Jackson, Riza M. Daza, Cole Trapnell, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2019.

[74] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19:1–5, 2018.

[75] Charles A Janeway, Paul Travers, Mark Walport, and Mark J Shlomchik. *Immunobiology: The Immune System in Health and Disease*. Garland Science, 2001.

[76] Kenneth Murphy and Casey Weaver. *Janeway's Immunobiology*. Garland Science, 2016.

[77] Hidde L. Ploegh. Antigen processing and presentation. *Nature*, 353(6342):125–130, 1998.

[78] David G Schatz and Peter C Swanson. V(d)j recombination: mechanisms of initiation. *Annual Review of Genetics*, 45:167–202, 2011.

[79] Ido Springer, Hanan Besser, Nitzan Tickotsky-Moskovitz, Shlomo Dvorkin, and Yoram Louzoun. Prediction of specific tcr-peptide binding from large dictionaries of tcr-peptide pairs. *Frontiers in Immunology*, 11, 2019.

[80] Ziqi Chen, Martin Renqiang Min, and Xia Ning. Ranking-based convolutional neural network models for peptide-mhc binding prediction. *ArXiv*, abs/2012.02840, 2020.

[81] Zhonghao Liu, Jing Jin, Yuxin Cui, Zheng Xiong, Alireza Nasiri, Yong Zhao, and Jianjun Hu. Deepseqpanii: an interpretable recurrent neural network model with attention mechanism for peptide-hla class ii binding prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[82] Xihao Hu and Shirley Liu. Deepbcr: Deep learning framework for cancer-type classification and binding affinity estimation using b cell receptor repertoires. *bioRxiv*, 2019.

[83] Antonio Lupia, Stefania Mimmi, Enzo Iaccino, Domenico Maisano, Federica Moraca, Carmine Talarico, Eugenio Vecchio, Gennaro Fiume, Francesco Ortuso, Giovanna Scala, Isabella Quinto, and Stefano Alcaro. Molecular modelling of epitopes recognized by neoplastic b lymphocytes in chronic lymphocytic leukemia. *European Journal of Medicinal Chemistry*, 111838, 2019.

[84] Shikhar Saxena, Sambhavi Animesh, Michael Fullwood, and Yuguang Mu. Onionmhc: A deep learning model for peptide — hla-a*02:01 binding predictions using both structure and sequence feature sets. *Journal of Micromechanics and Molecular Physics*, 2020.

[85] Pieter Moris, Joey De Pauw, A. Postovskaya, Sofie Gielis, Nicolas De Neuter, Wout Bittremieux, B. Ogunjimi, K. Laukens, and P. Meysman. Current challenges for unseen-epitope tcr interaction prediction and a new perspective derived from image classification. *Briefings in Bioinformatics*, 22, 2020.

[86] R. T, Omar Demerdash, and Jeremy C. Smith. Tcr-h: Machine learning prediction of t-cell receptor epitope binding on unseen datasets. *bioRxiv*, 2023.

[87] Yuepeng Jiang, Miaozhe Huo, and Shuai Cheng Li. Teinet: a deep learning framework for prediction of tcr-epitope binding specificity. *Briefings in bioinformatics*, 2023.

[88] Michael Cai, Seo-Jin Bang, Pengfei Zhang, and Heewook Lee. Atm-tcr: Tcr-epitope binding affinity prediction using a multi-head self-attention model. *Frontiers in Immunology*, 13, 2022.

[89] Anna Weber, Jannis Born, and María Rodríguez Martínez. Titan: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*, 37:i237–i244, 2021.

[90] Filippo Grazioli, Anja Mösch, Pierre Machart, Kai Li, Israa Alqassem, Timothy J O'Donnell, and Martin Renqiang Min. On tcr binding predictors failing to generalize to unseen peptides. *Frontiers in Immunology*, 13:1014256, 2022.

[91] Yicheng Gao, Yuli Gao, and Qi Liu. Pan-peptide meta learning for t-cell receptor–antigen binding recognition. *Nature Machine Intelligence*, 5:236–249, 2023.

[92] Minghao Yang, Zhi-an Huang, Wei Zhou, Junkai Ji, Jun Zhang, Sha He, and Zexuan Zhu. Mix-tpi: a flexible prediction framework for tcr–pmhc interactions based on multimodal representations. *Bioinformatics*, 39, 2023.

[93] Mathias Fynbo Jensen and Morten Nielsen. Nettcr 2.2 - improved tcr specificity predictions by combining pan- and peptide-specific training strategies, loss-scaling and integration of sequence similarity. *bioRxiv*, 2023.

[94] Ryan T. Leenay et al. Large dataset enables prediction of repair after crispr–cas9 editing in primary t cells. *Nature Biotechnology*, 37(9):1034–1037, 2019.

[95] Ken Kamimoto et al. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614:742–751, 2023.

[96] Anne Kunert, Mike van Brakel, Sylvia van Steenbergen-Langeveld, Madelon da Silva, Pierre G. Coulie, Cornelis Lamers, et al. MAGE-C2–specific TCRs combined with epigenetic drug-enhanced antigenicity yield robust and tumor-selective T cell responses. *The Journal of Immunology*, 197:2541–2552, 2016.

[97] Randi Vita, Shuchi Mahajan, Jeffrey A. Overton, Sandeep K. Dhanda, Sara Martini, Jason R. Cantrell, et al. The immune epitope database (iedb): 2018 update. *Nucleic Acids Research*, 47:D339–D343, 2019.

[98] Dmitriy V. Bagaev, Roshan M. A. Vroomans, Jessica Samir, Ulrik Stervbo, Clara Rius, Graham Dolton, et al. Vdjdb in 2019: database extension, new analysis infrastructure and a t-cell receptor motif compendium. *Nucleic Acids Research*, 48:D1057–D1062, 2020.

[99] Niv Tickotsky, Tal Sagiv, Jason Prilusky, Eran Shifrut, and Nir Friedman. Mcpas-tcr: a manually curated catalogue of pathology-associated t cell receptor sequences. *Bioinformatics*, 33:2924–2929, 2017.

[100] Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear methods. *bioRxiv*, 2024.

[101] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv:2308.08155*, 2023.

[102] Vinay Kumar Malik, Shivani Pathak, Kumari Anamika, Amarjit Kaur, and Vimal Kumar. A study of mvc: A software design pattern for web application development on j2ee architecture. *Academia.edu*, 2021.

[103] James Bucanek. *Model-View-Controller Pattern*. 01 2009.

[104] Martin Rammerstorfer and H. Mössenböck. Data mappings in the model-view-controller pattern. pages 121–132, 2003.

[105] Richard Membarth, Oliver Reiche, Frank Hannig, J. Teich, M. Körner, and Wieland Eckert. Hipacc: A domain-specific language and compiler for image processing. *IEEE Transactions on Parallel and Distributed Systems*, 27:210–224, 2016.

[106] X. Ye, Y. C. Lee, Z. P. Gates, Y. Ling, J. C. Mortensen, F. S. Yang, Y. S. Lin, and B. L. Pentelute. Binary combinatorial scanning reveals potent poly-alanine-substituted inhibitors of protein-protein interactions. *Communications Chemistry*, 5(1):128, Oct 2022.

[107] Zhijie Liu, Youyong Li, Lili Han, Jinwen Li, Jianyuan Liu, Zheng Zhao, Wenxuan Nie, Yuchi Liu, and Ruili Wang. Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31(3):405–412, 2015.

[108] Jamal Meslamani, Didier Rognan, and Esther Kellenberger. sc-pdb: a database for identifying variations and multiplicity of 'druggable' binding sites in proteins. *Bioinformatics*, 27(9):1324–1326, 2011.

[109] M. Michael Mysinger, Matteo Carchia, John J. Irwin, and Brian K. Shoichet. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012.

[110] Vishnu H. Murthy, Harlan M. Krumholz, and Catherine M. Gross. Participation in cancer clinical trials: Race-, sex-, and age-based disparities. *JAMA*, 291(22):2720–2726, 2004.

[111] Alice B. Popejoy and Stephanie M. Fullerton. Genomics is failing on diversity. *Nature*, 538(7624):161–164, 2016.

[112] M. V. Emmerik, A. Rappoport, and J. Rossignac. Simplifying interactive design of solid models: A hypertext approach. *The Visual Computer*, 9:239–254, 1993.

[113] Stavros Papadopoulos, Kushal Datta, S. Madden, and T. Mattson. The tiledb array data storage manager. *Proc. VLDB Endow.*, 10:349–360, 2016.

[114] Berker Tasoluk and Zuhal Tanrikulu. The performance comparison of a brute-force password cracking algorithm using regular functions and generator functions in python. *International Journal of Security, Privacy and Trust Management*, 2023.

[115] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martín Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[116] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J L de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[117] The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021.

[118] Junyi Gao, Cao Xiao, Lucas M Glass, and Jimeng Sun. Compose: Cross-modal pseudo-siamese network for patient trial matching. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 803–812, 2020.

[119] Xingyao Zhang, Cao Xiao, Lucas M Glass, and Jimeng Sun. Deepenroll: patient-trial matching with deep embedding and entailment prediction. In *Proceedings of the web conference 2020*, pages 1029–1037, 2020.

[120] Mohammad Lotfollahi, Anna Klimovskaia, Carlo De Donno, Yuge Ji, Ignacio L. Ibarra, F. Alexander Wolf, Nafissa Yakubova, Fabian J. Theis, and David Lopez-Paz. Compositional perturbation autoencoder for single-cell response modeling. *bioRxiv*, 2021.

[121] P. Agrawal, V. Gopalan, and S. Hannenhalli. Predicting gene expression changes upon epigenomic drug treatment. *bioRxiv*, 2023.

[122] Vineeta Nair, Rana Saleh, Sidrah Toor, Rania Z. Taha, Anwar Ahmed, Mohammed Kurer, Khaled A. Murshed, Mohammad Nada, and Ehab Elkord. Epigenetic regulation of immune checkpoints and t cell exhaustion markers in tumor-infiltrating t cells of colorectal cancer patients. *Epigenomics*, 12(17):1481–1492, 2020.

[123] Andrew N. Hoofnagle and Katheryn A. Resing. Proteomics and the analysis of protein phosphorylation. *Current Opinion in Biotechnology*, 12(6):617–622, 2001.

[124] Lloyd M. Smith and Neil L. Kelleher. Proteoform: a single term describing protein complexity. *Nature Methods*, 10(3):186–187, 2013.

[125] Morten Nielsen, Claus Lundegaard, Ole Lund, and Can Keşmir. The role of the proteasome in generating cytotoxic t-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics*, 57:33–41, 2005.

[126] Birkir Reynisson, Bruno Alvarez, S. Paul, Bjoern Peters, and M. Nielsen. Netmhcpan-4.1 and netmhciipan-4.0: improved predictions of mhc antigen presentation by concurrent motif deconvolution and integration of ms mhc eluted ligand data. *Nucleic Acids Research*, 48:W449 – W454, 2020.

[127] Tim O'Donnell, Alex Rubinsteyn, and Uri Laserson. Mhcflurry 2.0: Improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. *Cell Systems*, 11(1):42–48.e7, 2020.

[128] Damien Boulanger, R. C. Eccleston, Andrew Phillips, Peter Coveney, Tim Elliott, and Neil Dalchau. A mechanistic model for predicting cell surface presentation of competing peptides by mhc class i molecules. *Frontiers in Immunology*, 9:1538, 2018.

[129] Manoj Bhasin, Suman Lata, and Gajendra P. S. Raghava. Tappred prediction of tap-binding peptides in antigens. *Methods in Molecular Biology*, 409:381–386, 2007.

[130] Zeynep Koşaloğlu-Yalçın, Juhye Lee, Morten Nielsen, Jason Greenbaum, Stephen Schoenberger, Aaron M. Miller, Y. J. Kim, Alessandro Sette, and Bjoern Peters. Combined assessment of mhc binding and antigen expression improves t cell epitope predictions. *bioRxiv*, 2020.

[131] Songtao Huang and Yanrui Ding. Predicting binding affinity between mhc-i receptor and peptides based on molecular docking and protein-peptide interaction interface characteristics. *Letters in Drug Design Discovery*, 2022.

[132] Adiba Yaseen, Wajid Arshad Abbasi, and Fayyaz ul Amir Afsar Minhas. Protein binding affinity prediction using support vector regression and interfecial features. *2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 194–198, 2018.

[133] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Hui Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2021.

[134] Yuning You and Yang Shen. Cross-modality protein embedding for compound-protein affinity and contact prediction. *bioRxiv*, 2020.

[135] Shuangjia Zheng, Yongjian Li, Sheng Chen, Jun Xu, and Yuedong Yang. Predicting drug–protein interaction using quasi-visual question answering system. *Nature Machine Intelligence*, 2:134–140, 2019.

[136] D. Hong, D. Fort, Lizheng Shi, and E. Price-Haywood. Electronic medical record risk modeling of cardiovascular outcomes among patients with type 2 diabetes. *Diabetes Therapy*, 12:2007 – 2017, 2021.

[137] Haichen Lv, Xiaolei Yang, Bingyi Wang, Shaobo Wang, Xiaoyan Du, Qian Tan, Zhujing Hao, Y. Liu, Jun Yan, and Yunlong Xia. Machine learning–driven models to predict prognostic outcomes in patients hospitalized with heart failure using electronic health records: Retrospective study. *Journal of Medical Internet Research*, 23, 2020.

[138] Subendhu Rongali, A. Rose, D. McManus, Adarsha S. Bajracharya, Alok Kapoor, Edgard Granillo, and Hong Yu. Learning latent space representations to predict patient outcomes: Model development and validation. *Journal of Medical Internet Research*, 22, 2020.

[139] Fatemeh Rahimian, G. Salimi-Khorshidi, A. H. Payberah, J. Tran, R. Ayala Solares, F. Raimondi, M. Nazarzadeh, D. Canoy, and K. Rahimi. Predicting the risk of emergency admission with machine learning: Development and validation using linked electronic health records. *PLoS Medicine*, 15, 2018.

[140] Ji Hwan Park, Han Eol Cho, Jong Hun Kim, M. Wall, Y. Stern, H. Lim, Shinjae Yoo, Hyoung-Seop Kim, and Jiook Cha. Machine learning prediction of incidence of alzheimer's disease using large-scale administrative health data. *NPJ Digital Medicine*, 3, 2020.

[141] Luca Bedon, E. Cecchin, E. Fabbiani, M. Dal Bo, A. Buonadonna, Maurizio Polano, and G. Toffoli. Machine learning application in a phase i clinical trial allows for the identification of clinical-biomolecular markers significantly associated with toxicity. *Clinical Pharmacology Therapeutics*, 111, 2021.

[142] Alexander V. Schperberg, A. Boichard, I. Tsigelny, S. Richard, and R. Kurzrock. Machine learning model to predict oncologic outcomes for drugs in randomized clinical trials. *International Journal of Cancer*, 147:2537–2549, 2020.

[143] Yizhuo Wang, B. Carter, Ziyi Li, and Xuelin Huang. Application of machine learning methods in clinical trials for precision medicine. *JAMIA Open*, 5, 2021.

[144] R. Dai, T. Kannampallil, Jingwen Zhang, N. Lv, Jun Ma, and Chenyang Lu. Multi-task learning for randomized controlled trials. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6:1–23, 2022.

[145] Maria Brbi, Michihiro Yasunaga, Prabhat Agarwal, and Jure Leskovec. Predicting drug outcome of population via clinical knowledge graph. *To be published*, 2024. Preprint.

[146] Matthew M. Kalscheur, R. Kipp, M. Tattersall, Chaoqun Mei, K. Buhr, D. DeMets, M. Field, L. Eckhardt, and C. D. Page. Machine learning algorithm predicts cardiac resynchronization therapy outcomes: Lessons from the companion trial. *Circulation: Arrhythmia and Electrophysiology*, 11:e005499, 2018.

[147] N. Fujima, Y. Shimizu, D. Yoshida, S. Kano, T. Mizumachi, A. Homma, K. Yasuda, R. Onimaru, O. Sakai, K. Kudo, and H. Shirato. Machine learning-based prediction of treatment outcomes using mr imaging-derived quantitative tumor information in patients with sinonasal squamous cell carcinomas: A preliminary study. *Cancers*, 11:800, 2019.

[148] H. van Os, L. A. Ramos, A. Hilbert, Matthijs van Leeuwen, M. V. van Walderveen, N. Kruyt, D. Dippel, E. Steyerberg, I. van der Schaaf, Hester F. Lingsma, W. Schonewille, C. Majoie, S. Olabarriaga, K. Zwinderman, E. Venema, H. Marquering, and M. Wermer. Predicting outcome of endovascular treatment for acute ischemic stroke: Potential value of machine learning algorithms. *Frontiers in Neurology*, 9:784, 2018.

[149] H. Asadi, R. Dowling, B. Yan, and P. Mitchell. Machine learning for outcome prediction of acute ischemic stroke post intra-arterial therapy. *PLoS ONE*, 9:e88225, 2014.

[150] Varun Arvind, Jun S. Kim, E. Oermann, Deepak A Kaji, and Samuel K. Cho. Predicting surgical complications in adult patients undergoing anterior cervical discectomy and fusion using machine learning. *Neurospine*, 15:329–337, 2018.

[151] J. Senders, Patrick C. Staples, A. Karhade, Mark M. Zaki, W. Gormley, M. Broekman, T. Smith, and O. Arnaout. Machine learning and neurosurgical outcome prediction: A systematic review. *World Neurosurgery*, 109:476–486.e1, 2018.

[152] Zahra Jourahmad, J. M. Habibabadi, Houshang Moein, R. Basiratnia, Ali Rahmani Geranqayeh, S. S. Ghidary, and Seyed-Ali Sadegh-Zadeh. Machine learning techniques for predicting the short-term outcome of resective surgery in lesional-drug resistance epilepsy. *ArXiv*, abs/2302.10901, 2023.

[153] Emily J. MacKay, M. D. Stubna, Corey Chivers, Michael Draugelis, William J. Hanson, Nimesh D. Desai, and Peter W. Groeneveld. Application of machine learning approaches to administrative claims data to predict clinical outcomes in medical and surgical patient populations. *PLoS ONE*, 16, 2021.

[154] Erin Bowman, Shyam Banuprakash, Kim-Son Nguyen, and Matthew Marini. Machine learning prediction of progression events in oncology recist 1.1 clinical trials. *Journal of Clinical Oncology*, 2023.

[155] Rosalyn W. Sayaman, Denise M. Wolf, Christina Yau, Julie Wulfkuhle, Emanuel Petricoin, Lamorna Brown-Swigart, Smita M. Asare, Gillian L. Hirst, Laura Sit, Nicholas O'Grady, Diane Hedistian, I-SPY 2 TRIAL Consortium, Laura J. Esserman, Mark A. LaBarge, and Laura J van 't Veer. Application of machine learning to elucidate the biology predicting response in the i-spy 2 neoadjuvant breast cancer trial. *Cancer Research*, 80(4 Suppl), 2020.

[156] F. Beacher, L. Mujica-Parodi, Shreyash Gupta, and Leonardo A. Ancora. Machine learning predicts outcomes of phase iii clinical trials for prostate cancer. *Algorithms*, 14:147, 2021.

[157] K. W. Siah, S. Khozin, Chi Heem Wong, and A. Lo. Machine-learning and stochastic tumor growth models for predicting outcomes in patients with advanced non-small-cell lung cancer. *JCO Clinical Cancer Informatics*, 3:1–11, 2019.

[158] G. Beinse, Virgile Tellier, V. Charvet, E. Deutsch, I. Borget, C. Massard, A. Hollebecque, and L. Verlingue. Prediction of drug approval after phase i clinical trials in oncology: Resolved2. *JCO Clinical Cancer Informatics*, 3:1–10, 2019.

[159] Zifeng Wang, Cao Xiao, and Jimeng Sun. Spot: Sequential predictive modeling of clinical trial outcome with meta-learning. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2023.

[160] Farah E. Shamout, T. Zhu, and D. Clifton. Machine learning for clinical outcome prediction. *IEEE Reviews in Biomedical Engineering*, 14:116–126, 2020.

[161] N. Liu and J. Salinas. Machine learning for predicting outcomes in trauma. *SHOCK*, 48:504–510, 2017.

[162] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. Widedta: prediction of drug-target binding affinity. *arXiv preprint arXiv:1902.04166*, 2019.

[163] Zeng J. Yang J. Zhou J. Niu B. Guan J. Wang, Y. Prediction of drug-target interaction networks from the integration of protein sequences and drug chemical structures. *Molecules*, 24(2):321, 2019.

[164] Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. Empowering biomedical discovery with ai agents. *CellPress*, 187, 2024.