TwinCell: Large Causal Cell Model for Reliable and Interpretable Therapeutic Target Prioritisation

Jean-Baptiste Morlot[1,✉], **Thomaz Dias**[1,†], **Elie Hatem**[1,†], **and Yann Abraham**[1,†]

[1]DeepLife Lab, 8 Rue Jean Antoine de Baïf, Paris 75013, France
[†]These authors contributed equally

Drug discovery is impeded by the difficulty of translating targets from preclinical models to patients. Here, we present TwinCell, a Large Causal Cell Model (LCCM) capable of generalising from *in vitro* cell lines to patient-derived cell types while providing biologically meaningful interpretations of its predictions. Rather than predicting perturbation outcomes, TwinCell focuses on identifying the upstream regulators driving the transition between diseased and healthy states. By integrating single-cell foundation model embeddings with a multiomics interactome, TwinCell constrains predictions to mechanistically plausible signalling routes. To validate this approach, we introduce TwinBench, a novel benchmarking framework to assess virtual cell models generalisation capability while correcting for popularity bias and mode collapse through empirical p-value estimation. When applied to *in clinico* data, TwinCell recovers known drug targets aligned with the biology of the disease and demonstrates generalisation performance on targets unseen during training. This highlights the model's ability to learn mechanistic principles in a biological context. TwinCell represents a significant step toward building reliable and interpretable virtual cells for target identification bridging the gap between high-throughput *in vitro* experiments and clinical insights.

Virtual Cell | World Models | Digital Twin | Single-cell RNA-seq | Foundation Models | Target Identification | Causal Inference | Drug Discovery | Large Scale Causal Cell Model

Correspondence: ***jb.morlot@deeplife.co***

Introduction

Drug discovery remains a high-risk journey: only a small fraction of compounds ultimately reach patients, with estimates on the order of 10% progressing from early clinical trials to approval (1). This attrition is largely driven by the iterative nature of biological hypothesis generation involving time-consuming, costly experimental validations using *in vitro* and in vivo models. As a consequence, teams are routinely forced into high-stakes decisions under substantial uncertainty committing to costly clinical validation before the confidence can compensate for the risks (2).

An analogy can be drawn from complex engineered systems. Historically, aerospace design relied heavily on iterative prototyping to identify failure modes and optimise architectures. However, this approach is limited by the impossibility of exhaustively testing the full parameter space (3). This paradigm shifted with the introduction of digital twins: high-fidelity digital counterparts of physical systems that support design exploration and validation prior to fabrication and deployment (3). Importantly, digital twins do not eliminate the need for physical validation; rather, they compress the expensive experimental iterations toward the end of the optimisation cycle while enabling broad in silico exploration early in development.

An essential value of a digital twin is its causal framework, providing interpretable and testable explanations of its behaviour. To extend this framework to natural systems, including biological ones, one must learn to identify the regulatory mechanisms governing cellular state transitions. Recent developments in world models have shown promising results in this direction. These models are composed of two components: a representation of the system's accessible states and a model of its states' evolution. Once trained, these systems can be used to guide an agent to navigate this latent space and identify the optimal perturbations that drive an initial state toward desired outcomes (4, 5).

Recent developments in machine learning have paved the way for digital twins of biological systems, termed virtual cells. Those models are trained to predict transitions between cellular states that are relevant to common biological questions, such as predicting the effect of a therapeutic intervention. Early approaches such as scGen (6) and CPA (7) use autoencoders to learn a latent space of the cellular state and predict perturbation effects across contexts. Graph-based methods like GEARS (8) leverage knowledge graphs to capture the biological relationships between genes, improving generalisation to unseen perturbations by explicitly modelling gene-gene dependencies. Meanwhile, the growing availability of large single-cell atlases and the success of foundational models in other domains results in the emergence of new large-scale pretrained embeddings such as scGPT (9) and Geneformer (10), which learn biologically meaningful gene representations from millions of cells. Finally, large-scale perturbation datasets including scPerturb (11) and Tahoe-100M (12) now provide the data required to train models that can predict transcriptomic changes across diverse cellular contexts.

The Arc Institute's recent STATE model (13) exemplifies this emerging class of models: trained on multiple large scale perturbation databases, including Tahoe-100M, it predicts transcriptomic changes due to perturbations across diverse contexts by learning to navigate into a pretrained latent space.

Starting from an unperturbed cellular state and a perturbation, the STATE model predicts the perturbed cellular state, enabling downstream exploratory tasks. However, generalising perturbation effects to entirely new cell types or novel perturbations remains a challenge, as recent work demonstrated that linear models can outperform deep learning approaches when extrapolating to unseen contexts (14).

This finding underscores both the need for challenging, biologically meaningful benchmarks and the importance of developing interpretable models that capture causal relationships rather than optimising global transcriptional similarity. Conventional benchmarks relying on metrics such as Pearson correlation or mean squared error can be misleading in high-dimensional transcriptomic spaces: they may reward average-case predictions, fail to detect mode collapse, and overlook intervention-specific signatures (13, 15–17).

In this work, we address a complementary formulation of the perturbation modelling problem by extending the approach introduced in PDGrapher (18). Rather than predicting perturbation outcomes from an initial state and subsequently identifying which perturbation would produce a desired outcome, we frame the task as a perturbation recommendation system: given an initial state and a desired target state, we want to predict the perturbations that most likely drive the transition between them. This formulation aligns directly with the practical needs of drug discovery, where the goal is to identify interventions that reverse disease phenotypes.

We introduce TwinCell, a Large Causal Cell Model (LCCM) for target recommendation and prioritisation. We define an LCCM as a model that computes the probability of candidate targets given observed Differentially Expressed Genes (DEGs) and cell state context, decomposing this probability over signalling paths through a multiomics interactome. TwinCell was trained on *in vitro* perturbation databases, scPerturb (11) and Tahoe-100M (12), and tested on various in therapeutical area including Systemic Lupus Erythematosus (SLE) (19).

In this work, we show that TwinCell recovers established therapeutic targets as top predictions and provides mechanistic insight into disease-relevant pathways. Those mechanistic hypotheses are supported by the underlying causal graph linking predicted targets to disease-associated gene expression changes. We also highlight TwinCell performances against state-of-the-art and baselines methods *in vitro* and across various therapeutic areas.

Overall, TwinCell represents a significant first step toward creating interpretable, mechanistically grounded virtual cells for target identification and drug discovery.

Results

TwinCell: Large Scale Causal Cell Model for interpretable target prioritisation

TwinCell is a large-scale single-cell causal model for target prioritisation, trained to predict perturbation targets from pairs of cell states derived from single-cell RNA-seq data (Figure 1a). The model operates on a multiomics interactome graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes $\mathcal{V}$ represent genes/proteins and edges $\mathcal{E}$ encode transcriptional regulatory and protein–protein interactions. Given two cell states with embeddings $\mathbf{x}_1$ and $\mathbf{x}_2$ obtained from a pre-trained foundation model, TwinCell learns a cell-state-specific edge probability for each interaction:

$$w_{ij} = P(n_i \mid n_j, \mathbf{x}_1, \mathbf{x}_2) \quad \textbf{(1)}$$

representing the conditional probability that node n_i is activated given activation of node n_j in the context defined by the two cell states. These edge probabilities are optimised end-to-end to predict:

$$t^* = \arg\max_{t \in \mathcal{V}} P(t \mid \text{DEGs}, \mathbf{x}_1, \mathbf{x}_2, \mathcal{G}) \quad \textbf{(2)}$$

where t^* is the most likely target responsible for the observed differentially expressed genes DEGs. The message-passing mechanism propagates causal evidence through the interactome, and the learned cell-state-specific edge weights can be extracted to reconstruct the most probable signalling routes connecting targets to differentially expressed genes, enabling users to interpret each prediction alongside its mechanistic reasoning.

TwinCell is trained on *in vitro* perturbation data from cell lines from the scPerturb database (11) (drug perturbations only) and the Tahoe-100M dataset (12). During training, the model receives paired cell states—a control condition and the same cell line subjected to a perturbation—and predict a probability for each target. The model is trained to maximise the probability of the true perturbation target.

After training, TwinCell takes a pair of cell states as input, and return a p returns a causal graph corresponding to the most likely explanation for the observed differentially expressed genes (DEGs), and a score corresponding to the importance of specific genes in this explanation (figure 1a).

TwinBench: Robust benchmarking framework for virtual cell models

We developed TwinBench, a benchmarking framework for target prioritisation that adapts metrics from recommendation systems while correcting for popularity bias (Figure 1b). In conventional recommendation settings, recall and precision are computed at a fixed rank cutoff K, selecting the top-K items and evaluating how many true positives are retrieved. However, out-of-distribution evaluation requires additional care: even when presented with random noise, recommendation models do not predict uniformly at random but instead favour items frequently observed during training—a phenomenon known as popularity bias. To address this, we replace the fixed K threshold with an empirical p-value computed independently for each gene. Specifically, we assess

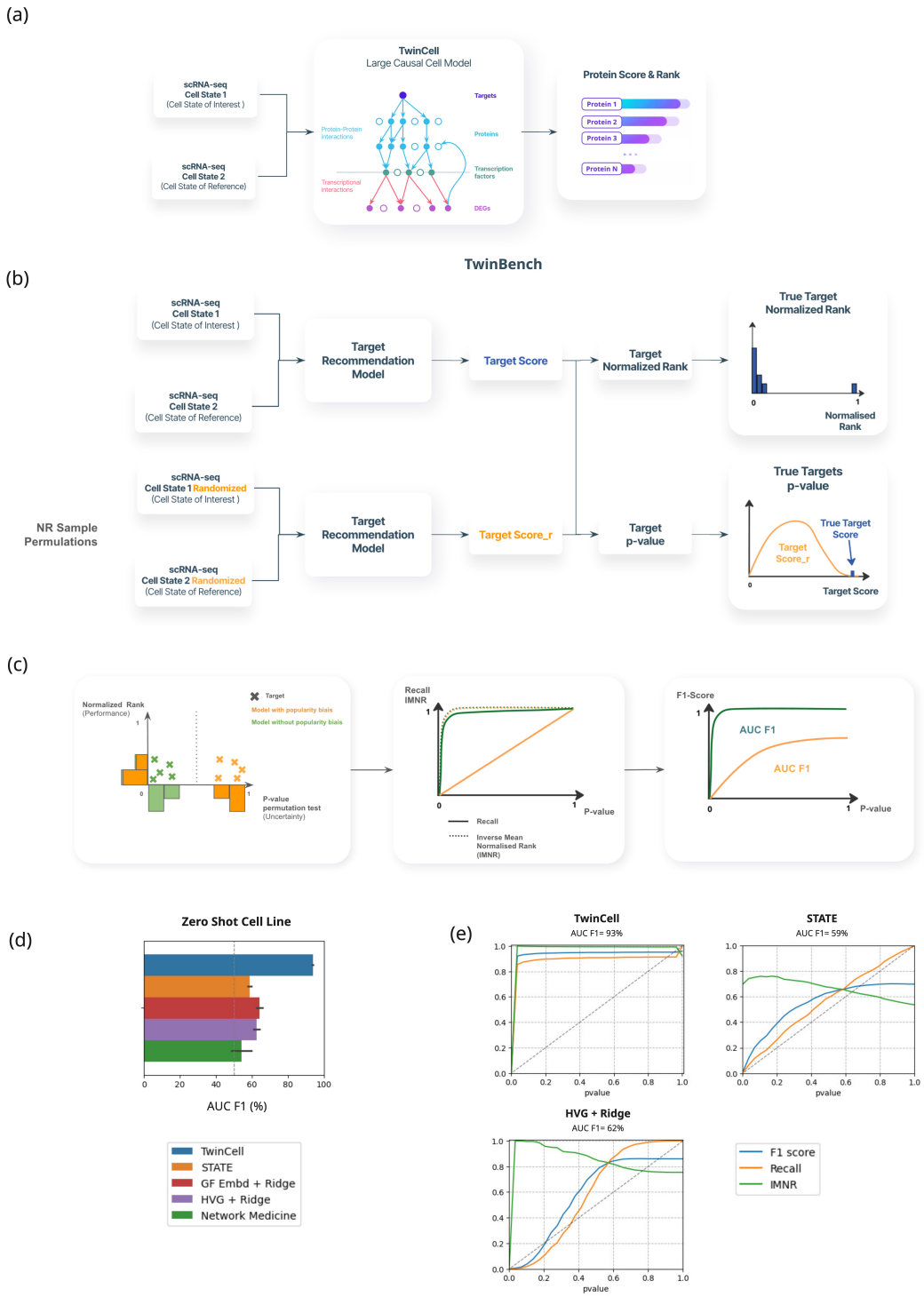

Fig. 1. TwinCell Large Scale Causal Cell Model architecture and TwinBench evaluation framework for virtual cell models. (a) TwinCell model architecture. Given two cell states from scRNA-seq data, TwinCell estimates the probability that each candidate target caused the observed differentially expressed genes (DEGs), using a multiomics interactome graph combining transcriptional regulatory and protein–protein interactions as inductive bias for a causal inference model: $P(\text{Target} \mid \text{DEGs}, \mathbf{x}_1, \mathbf{x}_2)$. The model learns the most probable signalling routes connecting DEGs to the target and provides interpretable mechanistic evidence for each prediction. (b) TwinBench evaluation pipeline for virtual cell models. From two input cell states, a model predicts a score for each target candidate. This list is first used to compute a normalised rank for each protein. To assess popularity bias in recommendation systems, we compute an empirical p-value for each protein, estimating whether its observed score is higher than expected by chance. Specifically, we permute the input data, recompute scores, and calculate the fraction of permutations in which the permuted score exceeds the actual score. This empirical p-value quantifies model uncertainty and corrects for models that predict the same targets regardless of input signal. (c) Empirical p-values can be used to estimate the popularity bias in target prediction. We derive a new metric to assess model performance: the Inverse Mean Normalised Rank (IMNR), defined as $1 -$ mean normalised rank of true targets with p-value $< pv$; recall is the fraction of true targets with p-value $< pv$. Using IMNR as a proxy for precision, we compute the F1-score across different p-value thresholds and derive the AUC under the F1-score curve. (d) TwinBench F1 AUC of multiple models: TwinCell (blue), STATE (orange), FM embedding + ridge regression (red), Highly Variable Genes + ridge regression (purple) and Network Medicine (green). (e) Detailed TwinBench results for INMR (green), Recall (orange) and F1 score (blue) for the TwinCell, STATE and Highly Variable Genes + ridge regression Models

each gene's score against a null distribution obtained by permuting input data, counting how often the permuted score exceeds the observed score. The empirical p-value is defined as:

$$p\text{-value}_j = \frac{1}{N_R} \sum_{i=1}^{N_R} \mathbf{1}\left[\text{score}_{ij}^{\text{random}} \geq \text{score}_j\right], \quad \textbf{(3)}$$

where N_R is the number of random permutations. This formulation establishes a gene-specific significance threshold: targets that consistently achieve high scores regardless of input signal (due to interactome topology or training frequency) will not reach significance, whereas targets whose scores depend on the input differential expression will.

For aggregate evaluation, we extend the AUC Precision–Recall framework by replacing precision with the Inverse Mean Normalised Rank (IMNR). Given a p-value threshold pv, IMNR is defined as $1 - \overline{r}(pv)$, where $\overline{r}(pv)$ denotes the mean normalised rank of true targets with p-value below pv; recall is correspondingly defined as the fraction of true targets reaching this significance level. The AUC of the resulting IMNR–Recall curve is computed across thresholds pv. We employ a hybrid sampling 30 values in with linearly-spaced thresholds in $[0.01, 1.0]$ Figure 1c.

TwinCell outperforms state-of-the-art models on *in vitro* out-of-distribution generalisation scenario

We benchmarked TwinCell against a suite of baselines, including 2 linear models adapted from EASE (20), a simple yet strong baseline for recommendation systems that often outperforms complex deep learning models, Network Medicine (21), a standard approach in drug repurposing relying on network proximity, and the recent STATE model (13), a leading virtual cell architecture. To assess the impact of input representation on linear baseline performance, we evaluated Ridge regression models trained on two distinct feature sets: highly variable genes (HVGs) selected globally across the training set, and Geneformer embeddings representing foundation-model latent spaces Figure 1.

While TwinCell and linear models explicitly predict perturbation targets, Network Medicine and STATE can be configured to rank drugs. Since TwinBench evaluates recommendation quality and penalises popularity bias, it is agnostic to whether the ranking applies to targets or drugs, provided the ground truth is defined accordingly (see Material & Methods).

We compared the performance of these different models using the zero-shot cell line approach used by STATE (13) where 5 cell lines were isolated from the training set. Those were initially selected based on their similarity to other cell lines in the training set, either high or low, to reflect the complexity of the biological space. The AUC of the different models in TwinBench is shown in Figure 1d: compared to STATE, network medicine shows the worst performance (AUC 44%), corresponding to noise. STATE performance is on par with the performance of linear models (AUC 59% for HVG + Ridge), while TwinCell outperforms other methods with an AUC of 93%. To illustrate the importance of the TwinBench metric we generated the Recall, IMNR and F1 score curves for TwinCell, STATE and the HVG + Ridge linear model. Figure 1e illustrates how our evaluation metrics capture popularity bias: across p-value thresholds, TwinCell maintains high IMNR (green curves) and recall (orange curves), whereas the HVG + Ridge baseline exhibits the characteristic Recall pattern of a model relying on learned priors rather than input-specific signal, with high IMNR and recall close to diagonal. Combining those 2 values into the F1 score (blue curves) enables a more direct comparison between the performance of different methods, especially on low p-values, and highlights the improvements brought by the TwinCell model.

TwinCell identifies key pathways associated with the treatment of Systemic Lupus Erythematosus

We tested TwinCell on a target identification task, using a dataset from Perez et al. (19). Blood samples were collected from healthy donors and patients with Systemic Lupus Erythematosus (SLE). After re-annotating the cell types, we focused our analysis on atypical B cells, a subset of B cells known to be associated with SLE (22–29) (Figure 2a). We computed Differentially Expressed Genes (DEGs) comparing atypical B cells from SLE samples to those derived from healthy donors, and used DEGs as input to the model.

Multiple methods use information encoded in interactome graphs, such as centrality or distance to the DEGs, to prioritise potential targets (30). These methods treat each edge as equivalent, irrespective of the underlying biology. These differences are reflected in the TwinCell model after training on perturb-seq data, and captured in the edge weights shown in the causal graph. Using TwinCell's probabilistic framework, we can then infer the most likely routes between the DEGs and any selected target protein. To illustrate this concept, we focused on the JAK/STAT pathway, which is upregulated in B cells associated with SLE (Figure 2b). We extracted the top paths connecting DEGs to IFNAR1 and visualised the results as a graph where nodes represent proteins and genes, and edge thickness corresponds to weights learned during the training. Directionality of the edges is either taken from the input interactome, or inferred by TwinCell which learns a Directed Acyclic Graph (DAG) underlying the causal relationships between observations and targets. This graph illustrates how transcription factors are connected either directly (IRF9, STAT3) or indirectly (STAT2) to the IFNAR1 and IFNAR2 sub-units of the IFN-α receptor and to the DEGs that were detected. We generated an alternative representation where nodes corresponding to members of the JAK/STAT pathway are coloured in orange (Figure 2c), emphasising the connections between the SLE DEGs and the response to IFN-α.

TwinCell can also be used for de novo hypothesis generation. Each protein in the interactome is ranked by the computed TwinCell score corresponding to $P(t \mid \text{DEGs}, \mathbf{x}_1, \mathbf{x}_2)$ for a given target t. The top 20 proteins for atypical B cells in SLE are shown in Table S1. The top ranking target is STAT2, a key signal transduction molecule in the JAK/STAT pathway. The model successfully recovers 2 other members of the pathway among the top candidates, including the Type I

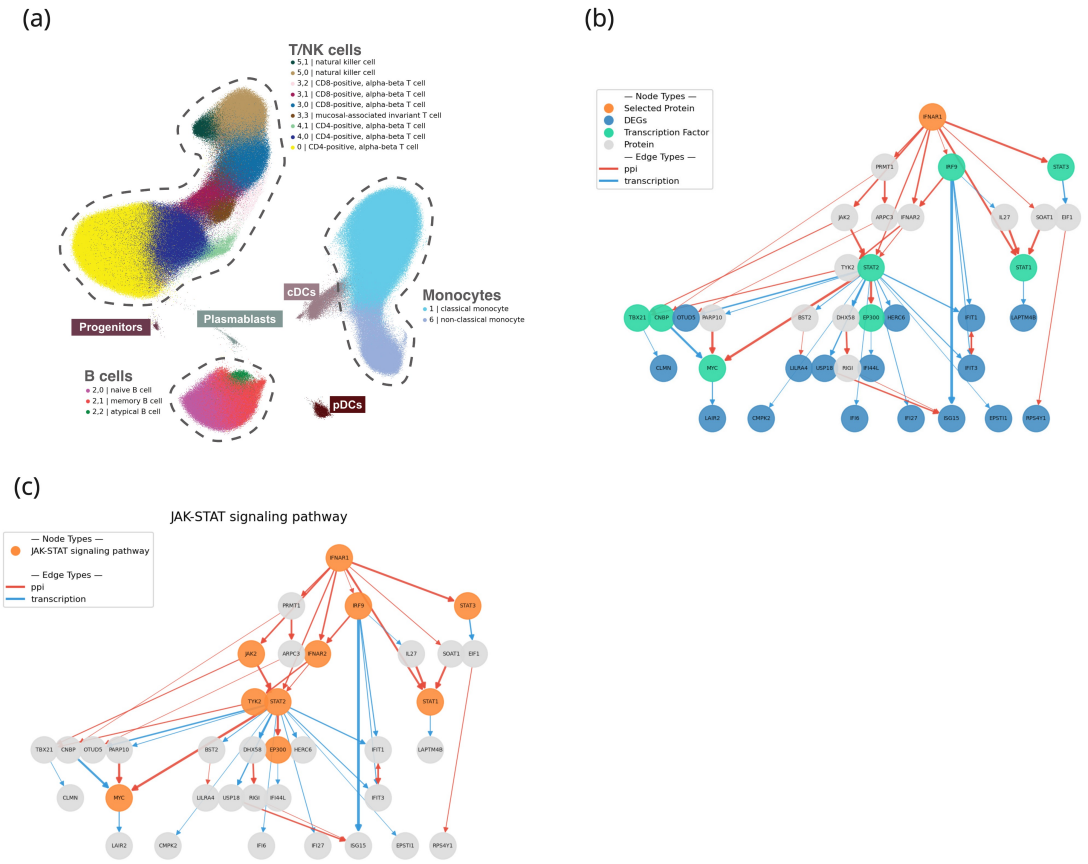

Fig. 2. Causal Analysis on Systemic Lupus Erythematosus B cell subtype(a) UMAP visualisation of the SLE atlas with cell annotation. (b) Causal Graph representing the most likely paths from the DEGs to IFN-α receptor in 2,2 | atypical B cells. The orange node corresponds to IFN-α, green nodes represent Transcription Factors, blue nodes correspond to DEGs that participated the most to IFN-α score. Edges correspond to directed interactions defined either in the interactome or learned during training. Red edges correspond to Protein-Protein Interactions (post-translational), blue edges to Protein-Gene Interactions (transcriptional). The edge thickness corresponds to weights associated by the model, with stronger weights corresponding to the most likely routes. (c) Same as (b) with orange nodes corresponding to proteins associated with the JAK/STAT pathway.

interferon receptor subunits IFNAR2 (rank 2) and IFNAR1 (rank 11). To further refine target prioritisation, the TwinCell score can be integrated with complementary metrics derived from the permutation test—specifically the empirical p-value and z-score—as well as the target's differential expression status and its predicted downstream impact (quantified as the number of DEGs with $\sum_i P(\text{DEG}_i \mid t, \mathbf{x}) > 10^{-3}$). For instance, high-confidence drivers like STAT2 and TLR7 are distinguished by their broad downstream regulatory footprint ($n \geq 14$ affected DEGs), whereas lower-relevance candidates such as CTCF or LIG3 are characterised by limited downstream impact, lower z-scores, or non-significant p-values. This multi-dimensional assessment enables more robust identification of therapeutic candidates.

Next we compared this list to the list of approved drug targets for SLE obtained from OpenTargets, corresponding to compounds in phase III and IV clinical trials (Table S2); 3 proteins were not expressed or detected in atypical B cells thus excluded from this analysis. On top of IFNAR1 (rank #11) we identified other members of the JAK/STAT pathways: TYK2 (rank #147) and the kinases JAK2 and JAK3 (rank #102 and #131). We also identified TLR7 and TLR9 (rank #1 and rank #137). Those targets were identified in the top 1% of all genes present in the interactome, illustrating the ability of TwinCell to enrich for known targets in a cell-type-specific way.

TwinCell outperforms several target identification methods in the retrieval of clinically validated targets in multiple independent diseases

Next, we compared TwinCell performance to other target identification methods. We applied the TwinBench framework on four distinct diseases that were selected because they impact different tissues and are distinct from the cancer-related cell lines that were used during training, including: Ulcerative Colitis (intestine), Systemic Lupus Erythematosus (SLE, blood), Parkinson Disease (brain), and Psoriasis (skin). For each disease, we focused the analysis on the disease-specific approved targets extracted from OpenTargets as described before, then focused on the cell type with the highest

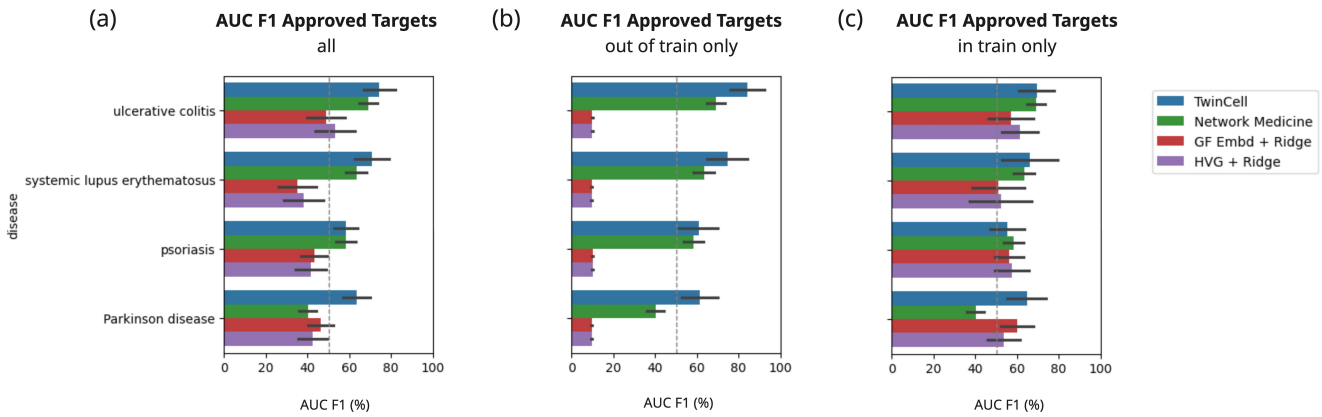

Fig. 3. In clinico scenario results across four therapeutic areas. (a) TwinBench AUC F1 scores were calculated using 4 models for all cellular types across 4 distinct diseases, using approved targets from OpenTargets as ground truth. The following models were compared, TwinCell (blue), Network Medicine (green), and 2 different ridge regression models using Geneformer embeddings (red) and Highly Variable Genes (purple) as input. Dark segments correspond to the 95% confidence interval of the TwinBench F1 score AUC. For each disease the results for the best cell type by TwinCell score are shown. (b) Same as (a) using only targets in the training set for TwinCell and linear methods. (c) same as (a) for targets that are outside the training set. Network medicine values are kept the same in (a) (b) and (c) for reference.

AUC across all models: inflammatory macrophages in Ulcerative Colitis, atypical B cells in SLE, microglial cells in Parkinson Disease, and CD4-positive helper T cells in Psoriasis. Results are shown as bar graphs, where the vertical dotted line corresponds to random performance (Figure 3a).

To assess the models' ability to generalise beyond targets present in the training set, we compared the performance of the different models on three scenarios: All targets (Figure 3a), Out-of-train targets only (Figure 3b), and Train-only targets (Figure 3c). The targets per disease and per scenario are available in Table S3, unfiltered by cell type. We used Network Medicine values as a reference, since the performance of this method is not impacted by the presence or absence of a specific target in a training set. In the All targets scenario (Figure 3a), we observe that TwinCell and Network Medicine consistently outperform the linear methods once the popularity bias is accounted for. For Parkinson disease, TwinCell outperforms other methods with a performance of 60%. In the Out-of-train targets only scenario (Figure 3b), we observe that TwinCell and Network Medicine outperform linear methods by a large margin across the four diseases. This suggests that using an interactome is an important prior to generalise beyond the training set. Finally, in the Train-only targets scenario (Figure 3c), we observe that the performance becomes comparable across all models with higher score for linear models and decreased performances for TwinCell, reflecting the metric's correction for targets that are frequently represented in the training set.

Taken together, these results illustrate TwinCell's ability to recover clinically validated targets across different diseases and to generalise to *de novo* targets absent from the training set, in contrast to other methods that either fail to generalise, or are too computationally expensive for *de novo* target identification.

Discussion

We present TwinCell, a large causal cell model capable of generalising at scale from *in vitro* cell lines to patient-derived cell types while providing biologically meaningful interpretations of its predictions. Rather than predicting perturbation outcomes TwinCell focuses on the inverse problem: given two cell states, identify the upstream regulators driving the transition between them. This framing aligns more directly with the practical needs of drug discovery, where the goal is to identify interventions that reverse disease states. By integrating single-cell foundation model embeddings with a multiomics interactome, our approach constrains predictions to mechanistically plausible signalling routes. When applied to patient-derived data, TwinCell recovers clinically-approved drug targets across multiple diseases, supported by a causal graph linking targets to DEGs. This graph highlights the biological rationale behind each prediction, increasing the trust in the ranked targets and enabling the exploration of alternative hypotheses in different pathways.

The importance of biological priors for generalisation. By comparing network based approaches in clinical settings, we found that incorporating curated biological knowledge as an inductive bias improves generalisation on new targets out of distribution. While recent large-scale perturbation models trained end-to-end on transcriptomic data have shown impressive interpolation capabilities, they often struggle when extrapolating to unseen cell types or perturbations (13, 14). In contrast, TwinCell's use of an interactome graph constrains the solution space to biologically plausible signalling routes, enabling the model to reason about targets by leveraging conserved pathway structures. This suggests that the path to reliable virtual cell models may lie not in ever-larger datasets alone, but high-quality interactomes.

TwinBench: Towards robust evaluation of virtual cell models. An important methodological contribution of this

work is TwinBench, a benchmarking framework that addresses the evaluation challenges inherent to perturbation modelling. Standard metrics such as correlation or mean squared error can be misleading when models exhibit mode collapse or popularity bias, predicting average responses or frequently observed targets regardless of input. By introducing empirical p-values that assess whether predictions depend on the specific input signal, TwinBench provides an approach to discriminate true signal from overfitting training priors. This benchmark will be released as a github package to foster innovation.

Perspectives. This work can be extended in multiple ways: on the benchmarking side, TwinBench will be expanded with increasingly difficult zero-shot scenarios, disease atlases covering additional therapeutic areas, and a wider range of baselines to ensure robust comparison of virtual cell models. On the modelling side, future iterations of TwinCell will integrate higher-quality interactomes, fine-tuned embeddings, larger perturbation datasets, cell-cell communication, and multi-species data.

TwinCell demonstrates competitive generalisation performance across diverse scenarios, ranging from zero-shot *in vitro* cell lines to patient-derived disease states. Furthermore, its ability to generalise to targets absent from the training set highlights its potential for *de novo* target identification. Underlying the predictions, the mechanistic insights provided by the causal graphs provide better confidence and facilitate the exploration of alternative pathway hypotheses. The architecture is highly scalable, generating predictions in seconds and enabling deployment across a broad spectrum of applications, including large scale screening and integration into agentic discovery pipelines.

Looking ahead, we envision a lab-in-the-loop paradigm where TwinCell generates testable hypotheses that are iteratively refined through experimental feedback, thereby building high-confidence causal cell models across various therapeutic areas, positioning virtual cells as translation tools bridging high-throughput *in vitro* experiments and clinical insights. Although virtual cell models have yet to achieve the transformative performance seen in domains such as text and images (14), we believe they are nearing a similar inflection point.

Code availability

To foster academic collaboration, we will soon release an API for academic use, with a waiting list currently available on the DeepLife website, and the TwinBench package through Github.

Acknowledgement

We thank Drs Mayar Ali, Sean Gorman, Nathalie Jaure, Sebastien Legare, Alessandro Romualdi, Christian Tendeng and Moritz Thomas for their critical feedback during the writing of this manuscript. We thank Axel Kuehn and Maxime Gendre for their work on the infrastructure and TwinCell API.

Bibliography

1. Shingo Yamaguchi, Masayuki Kaneko, and Mamoru Narukawa. Approval success rates of drug candidates based on target, action, modality, application, and their combinations. *Clinical and Translational Science*, 14(3):1113–1122, 2021.
2. Jack W. Scannell, Alex Blanckley, Helen Boldon, and Brian Warrington. Diagnosing the decline in pharmaceutical r&d efficiency. *Nature Reviews Drug Discovery*, 11(3):191–200, 2012. doi: 10.1038/nrd3681. Cited for: high-stakes go/no-go decisions in drug discovery.
3. Michael Grieves and John Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems*. Springer, 2017. doi: 10.1007/978-3-319-38756-7_4. Cited for: origin of digital twin concept in aerospace/engineering.
4. David Ha and Jürgen Schmidhuber. World models. *arXiv*, 2018. Cited for: world models concept - learning state representations and dynamics.
5. DeepMind. Genie 2: A large-scale foundation world model, 2024. Cited for: modern world model innovations.
6. Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scgen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019. doi: 10.1038/s41592-019-0494-8. Cited for: VAE-based perturbation prediction.
7. Mohammad Lotfollahi et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):e11517, 2023. doi: 10.15252/msb.202211517. Cited for: CPA model; posterior collapse in perturbation models.
8. Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2024. doi: 10.1038/s41587-023-01905-2. Cited for: GEARS - graph-based perturbation prediction.
9. Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, 21:1470–1480, 2024. doi: 10.1038/s41592-024-02201-0. Cited for: scGPT foundation model.
10. Christina V. Theodoris et al. Transfer learning enables predictions in network biology. *Nature*, 618:616–624, 2023. doi: 10.1038/s41586-023-06139-9. Cited for: Geneformer foundation model.
11. S. Peidli et al. scperturb: information-rich benchmarks for single-cell perturbation prediction. *Nature Methods*, 21:1137–1149, 2024. doi: 10.1038/s41592-024-02203-y. Cited for: scPerturb database - harmonized perturbation data.
12. J. Zhang et al. Tahoe-100m: A giga-scale single-cell perturbation atlas for context-dependent gene function and cellular modeling. *bioRxiv*, 2025. doi: 10.1101/2025.01.01.123456. Cited for: Tahoe-100M - 100M cells, 1100 drugs, 50 cell lines.
13. A. K. Adduri et al. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, 2025. Cited for: STATE model - virtual cell trained on Tahoe-100M; OOD challenges.
14. Constantin Ahlmann-Eltze et al. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear baselines. *Nature Methods*, 22:1657–1661, 2025. doi: 10.1038/s41592-025-02772-6. Cited for: linear models outperforming DL in some OOD contexts.
15. Y. Wu et al. Perturbench: Benchmarking machine learning models for cellular perturbation analysis. *arXiv*, 2024. Cited for: PerturBench; inadequacy of Pearson/MSE for perturbation models.
16. Mathieu Chevalley et al. Causalbench: A large-scale benchmark for network inference from single-cell perturbation data. *arXiv*, 2022. Cited for: CausalBench - ranking-based evaluation.
17. Mohammad Dehghani et al. The benchmark lottery. *arXiv*, 2021. Cited for: importance of meaningful benchmarks.
18. Guadalupe Gonzalez, Xiang Lin, Isuru Herath, Kirill Veselkov, Michael Bronstein, and Marinka Zitnik. Combinatorial prediction of therapeutic perturbations using causally-inspired neural networks. *bioRxiv*, pages 2024–01, 2025. Cited for: PDGrapher - reverse problem framing (desired state to perturbation).
19. Richard K. Perez et al. Single-cell rna-seq reveals cell type-specific molecular and genetic signatures in systemic lupus erythematosus. *Science*, 376(6589):eabf1970, 2022. doi: 10.1126/science.abf1970. Cited for: SLE single-cell atlas (GSE174188).
20. Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pages 3251–3257, 2019. Cited for: EASE baseline; popularity bias = class imbalance.
21. Emre Guney, Jörg Menche, Marc Vidal, and Albert-László Barabási. Network-based in silico drug efficacy screening. *Nature Communications*, 7:10331, 2016. doi: 10.1038/ncomms10331. Cited for: network proximity method for drug-disease associations.
22. Annett M. Jacobi, Marcus Odendahl, Kerstin Reiter, Annegret Bruns, Gerd-Rüdiger Burmester, Andreas Radbruch, Günter Valet, Peter E. Lipsky, and Thomas Dörner. Correlation between circulating CD27high plasma cells and disease activity in patients with systemic lupus erythematosus. *Arthritis and Rheumatism*, 48(5):1332–1342, 2003. doi: 10.1002/art.10949. Cited for: plasma cells and SLE disease activity.
23. Annett M. Jacobi, Kerstin Reiter, Mackenzie Mackay, Cynthia Aranow, Falk Hiepe, Andreas Radbruch, Anja Hansen, Gerd-Rüdiger Burmester, Betty Diamond, Peter E. Lipsky, and Thomas Dörner. Activated memory B cell subsets correlate with disease activity in systemic lupus erythematosus: Delineation by expression of CD27, IgD, and CD95. *Arthritis and Rheumatism*, 58(6):1762–1773, 2008. doi: 10.1002/art.23498. Cited for: B cell subsets and SLE disease activity.
24. Annett M. Jacobi, Henrik Mei, Bimba F. Hoyer, Irfan M. Mumtaz, Kerstin Thiele, Andreas Radbruch, Gerd-Rüdiger Burmester, Falk Hiepe, and Thomas Dörner. HLA-DRhigh/CD27high plasmablasts indicate active disease in patients with systemic lupus erythematosus. *Annals of the Rheumatic Diseases*, 69(1):305–308, 2010. doi: 10.1136/ard.2008.096495. Cited for: plasmablasts as markers of SLE activity.
25. Claudia Wehr, Hermann Eibel, Mohan Masilamani, Harald Illges, Michael Schlesier, Hans-Hartmut Peter, and Klaus Warnatz. A new CD21low B cell population in the peripheral blood of patients with SLE. *Clinical Immunology*, 113(2):161–171, 2004. doi: 10.1016/j.clim.2004.05.010. Cited for: CD21low B cells in SLE.

26. Christopher M. Tipton, Christopher F. Fucile, Jennie Darce, Asiya Chida, Toru Ichikawa, Ignacio Gregoretti, Scott Schiber, John Hom, Scott Jenks, Rebecca J. Feldman, Ramit Mehr, Chungwen Wei, F. Eun-Hyung Lee, Winnie C. Cheung, Alexander F. Rosenberg, and Ignacio Sanz. Diversity, cellular origin and autoreactivity of antibody-secreting cell population expansions in acute systemic lupus erythematosus. *Nature Immunology*, 16(7):755–765, 2015. doi: 10.1038/ni.3175. Cited for: antibody-secreting cells in SLE flares.
27. Manabu Torigoe, Shigeru Iwata, Shingo Nakayamada, Kazuhisa Sakata, Mingzhu Zhang, Gabor Hajber, Maiko Yoshikawa, Achilleas Floudas, Douglas J. Veale, Ursula Fearon, and Yoshiya Tanaka. Metabolic reprogramming commits differentiation of human CD27(+)IgD(+) B cells to plasmablasts or CD27(-)IgD(-) cells. *Journal of Immunology*, 199(2):425–434, 2017. doi: 10.4049/jimmunol.1601908. Cited for: B cell differentiation in SLE.
28. Scott A. Jenks, Kevin S. Cashman, Esther Zumaquero, Urko M. Marigorta, Arezou V. Patel, Xialei Wang, Deepak Tomar, Matthew C. Woodruff, Zaya Simon, Regina Bugrovsky, et al. Distinct effector b cells induced by unregulated toll-like receptor 7 contribute to pathogenic responses in systemic lupus erythematosus. *Immunity*, 49(4):725–739.e6, 2018. doi: 10.1016/j.immuni.2018.08.015. Cited for: atypical/DN2 B cells role in SLE pathogenesis.
29. Shuyi Wang, Jingchuan Wang, Vikas Kumar, Jodi L. Karnell, Bridget Naber, Saurabh Bhattacharya, Saurabh Bhattacharya, Beth Naiman, Rachel Ettinger, and Saurabh Bhattacharya. IL-21 drives expansion and plasma cell differentiation of autoreactive CD11c(hi)T-bet(+) B cells in SLE. *Nature Communications*, 9(1):1758, 2018. doi: 10.1038/s41467-018-03750-7. Cited for: IL-21 and atypical B cells in SLE.
30. Ariele Viacava Follis. Centrality of drug targets in protein networks. *BMC Bioinformatics*, 22:527, 2021. doi: 10.1186/s12859-021-04342-x. Cited for: network centrality methods for drug target identification.
31. B. Rath et al. How do recommendation models amplify popularity bias? an analysis from the spectral perspective. *arXiv*, 2024. Cited for: theoretical bridge between popularity bias and embedding collapse.
32. Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021. Cited for: MLE bias against rare classes; safe prediction concept.
33. Wout Mulder and Tim Baumgartner. Gans for imbalanced data: Problem analysis and a novel solution. *arXiv*, 2021. Cited for: mode collapse as failure to model minority class.
34. L. Hetzel et al. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems*, 35:26711–26722, 2022. Cited for: ChemCPA; metric trap where R2 is misleading.
35. Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Managing popularity bias in recommender systems with personalized re-ranking. *Proceedings of the Thirty-Second International FLAIRS Conference*, pages 413–418, 2019. Cited for: standard accuracy metrics hide tail failures.
36. Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, 2020. doi: 10.18653/v1/2020.acl-main.442. Cited for: accuracy paradox - high accuracy masking conditional logic failures.
37. Jack W. Scannell and Jim Bosley. Predictive validity: The missing link in drug discovery productivity. *Drug Discovery Today*, 27(6):1515–1523, 2022. doi: 10.1016/j.drudis.2022.02.015. Cited for: importance of predictive validity; metrics masking model failures.
38. Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, New York, 1994. Cited for: bootstrap methodology.
39. Anthony C. Davison and David V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, Cambridge, 1997. Cited for: two-level bootstrap.
40. Ewout W. Steyerberg. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. Springer, 2nd edition, 2019. Cited for: per-sample bootstrap in clinical validation.
41. Adarsh Subbaswamy and Suchi Saria. From development to deployment: dataset shift, causality, and shift-stable models in health ai. *Biostatistics*, 21(2):345–352, 2020. Cited for: context-specific model reliability.
42. David Ochoa, Andrew Hercules, Luca Fumis, Ana Hernández-Plaza, Felipe Velez, Miguel Mirasierra, Eliseo Papa, Ian Dunham, and Andreas Mayr. The next-generation open targets platform: reimagined, redesigned, rebuilt. *Nucleic Acids Research*, 51(D1):D1353–D1359, 2023. doi: 10.1093/nar/gkac1046. Cited for: Open Targets Platform data sources.
43. Rose Oughtred, Jennifer Rust, Christie Chang, Bobby-Joe Breitkreutz, Chris Stark, Andrew Willems, Lorrie Boucher, Genie Leung, Nadine Kolas, Frederick Zhang, Sonam Dolma, Jasmin Coulombe-Huntington, Andrew Chatr-Aryamontri, Kara Dolinski, and Mike Tyers. The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. *Protein Science*, 30(1):99–111, 2021. doi: 10.1002/pro.3978. Cited for: BioGRID interaction database.
44. Marija Milacic, Deidre Beavers, Patrick Conley, Chuqiao Gong, Marc Gillespie, Johannes Griss, Robin Haw, Bijay Jassal, Lisa Matthews, Bruce May, Robert Petryszak, Eliot Ragueneau, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Ralf Stephan, Krishna Tiwari, Thawfeek Varusai, Joel Weiser, Adam Wright, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The Reactome pathway knowledgebase 2024. *Nucleic Acids Research*, 52(D1):D649–D658, 2024. doi: 10.1093/nar/gkad1025. Cited for: Reactome pathway database.
45. Sandra Orchard, Mais Ammari, Bruno Aranda, Lionel Breuza, Leonardo Briganti, Fiona Broackes-Carter, Nancy H. Campbell, Gayatri Chavali, Carol Chen, Noemi del Toro, Margaret Duesbury, Marine Dumousseau, Eugenia Galeota, Ursula Hinz, Marta Iannuccelli, Sruthi Jagannathan, Rafael Jimenez, Jyoti Khadake, Astrid Lagreid, Luana Licata, Ruth C. Lovering, Birgit Mber, Sundar Mudali, Prudence Mutowo-Muellenet, Anushya Muruganujan, Charles Orellana, Ana Del Pozo, Andrea Patriarca, Shilpa Peri, Pablo Porras, Sivakumar Raghunath, Sylvie Riber, Bernd Roechert, Joan Segura, Aleksandra Shypitsyna, Nives Skunca, Roberto Solano, Paul D. Thomas, Kalliopi Trachana, Athanasia Tsatsos, Denise Turber, Gamze Uzsoy, Yves Vandenbrouck, Kris Vanhoorelbeke, Marc Vidal, Ale Vullo, Colin Webb, Xavier Watkins, Maria Zeshan, and Henning Hermjakob. The MIntAct project–IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research*, 42(D1):D358–D363, 2014. doi: 10.1093/nar/gkt1115. Cited for: IntAct molecular interaction database.
46. Prisca Lo Surdo, Marta Iannuccelli, Silvia Contino, Arnaldo Ceol, Luana Licata, Gianni Cesareni, and Livia Perfetto. SIGNOR 3.0, the SIGnaling network open resource 3.0: 2022 update. *Nucleic Acids Research*, 51(D1):D483–D489, 2022. doi: 10.1093/nar/gkac883. Cited for: SIGNOR signaling network database.
47. Samuel L. Wolock, Romain Lopez, and Allon M. Klein. Scrublet: Computational identification of cell doublets in single-cell rna-seq data. *Cell Systems*, 8(4):281–291, 2019. doi: 10.1016/j.cels.2018.11.005. Cited for: Scrublet doublet detection.
48. Ilya Korsunsky et al. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0. Cited for: Harmony batch correction.
49. F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0. Cited for: Scanpy preprocessing pipeline.
50. Isaac Virshup et al. anndata: Access and store annotated data matrices. *Journal of Open Source Software*, 6(68):4371, 2024. doi: 10.21105/joss.04371. Cited for: AnnData format.

Material & Methods

TwinCell LCCM Learning Objective

TwinCell is trained to identify the target t^* that maximises the probability given the observed differentially expressed genes DEGs and cell state context $\mathbf{x}$:

$$t^* = \arg\max_t P(\mathrm{t} \mid \mathrm{DEGs}, \mathbf{x}) \tag{4}$$

where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ are the cell state embeddings obtained from Geneformer foundation model (10), t is the target, and DEGs is the differentially expressed genes.

The model computes this probability by considering all possible paths connecting the observed DEGs to candidate targets through the interactome:

$$P(t \mid \mathrm{DEGs}, \mathbf{x}) = \sum_{l=1}^{L_{\max}} \sum_{\substack{\text{paths } p:|p|=l \\ n_0 \in \mathrm{DEGs},\, n_l = t}} \prod_{i=0}^{l-1} P(n_{i+1} \mid n_i, \mathbf{x}) \tag{5}$$

where each path p of length $l \leq L_{\max}$ connects a starting node $n_0 \in$ DEGs to the target $n_l = t$ through successive transitions in the interactome. Here we set $L_{\max} = 5$ as a trade-off between computational cost and model complexity while accounting for the interactome topology with an average shortest path length of 2.98.

The model learns the cell-state-specific transition probabilities:

$$w_{ij} = P(n_i \mid n_j, \mathbf{x}) \tag{6}$$

through end-to-end optimisation to maximise $P(t \mid \mathrm{DEGs}, \mathbf{x})$. Once trained, the model enables probabilistic inference for:

1. **Target prioritisation**: Computing $P(t \mid \mathrm{DEGs}, \mathbf{x})$ for all candidate targets and ranking them.
2. **Causal graph construction**: Identifying the signalling routes that maximise $\prod_{i=0}^{l-1} P(n_{i+1} \mid n_i, \mathbf{x})$ between the DEGs and a selected target, extracting the most probable causal paths.
3. **Downstream effect prediction**: Applying Bayes' rule to invert the learned conditionals, computing $P(\mathrm{DEG}_i \mid \mathrm{t}_j, \mathbf{x})$ to identify DEGs impacted by modulating a given target t_j.

TwinBench Metric Details

Recent work has established a theoretical bridge between popularity bias in recommendation systems, class imbalance in classification, and mode collapse in generative models (20, 31). The unifying concept is that standard training objectives (Cross-Entropy, MSE, or ELBO) optimise for average-case performance. When data is long-tailed, this incentivises models to predict frequent outcomes accurately while ignoring rare but biologically meaningful perturbation effects (32, 33).

In virtual cell modelling, mode collapse manifests when models learn to predict a mean distribution equidistant to all target states, rather than the specific transcriptional response to each perturbation. Such models can still achieve high R^2 or cosine similarity because standard metrics do not account for the similarity expected by chance—and in high-dimensional transcriptomic space, random predictions are not independent but exhibit substantial baseline correlation (7, 34, 35). This allows models with no genuine predictive power to appear performant, masking their ability to learn perturbation-specific causal effects (36, 37).

To address this, TwinBench introduces a statistical correction using empirical p-values via a permutation test. The workflow proceeds as follows:

Step 1: Score Computation and Input Permutation. For each experiment (control vs. perturbed cell states), the model produces a score for every candidate target. To assess whether predictions depend on the specific biological signal rather than learned priors, we perform N_R random permutations of the input differential expression signature. This permutation destroys the gene–gene correlation structure (the specific signal defining the condition) while preserving the marginal distribution of expression values. For each target j, we compute an empirical p-value:

$$p\text{-value}_j = \frac{1 + \sum_{i=1}^{N_R} \mathbf{1}\left[\mathrm{score}_{ij}^{\mathrm{perm}} \geq \mathrm{score}_j\right]}{N_R + 1} \tag{7}$$

If a target receives a high score even under permuted conditions, the model is driven by the target's global popularity rather than the specific causal evidence in the query. Conversely, a significant p-value indicates the model is actively using the input signal to deviate from its background prior.

Step 2: Normalised Rank Computation. Each target receives a normalised rank $r_j \in [0,1]$ based on its score relative to all candidate targets, where $r_j = 0$ indicates the top-ranked target and $r_j = 1$ the lowest.

Step 3: IMNR, Recall, and F1 Score at Each p-value Threshold. For a given p-value threshold pv, we define:

- **Recall**(pv): fraction of true targets with p-value $< pv$
- **IMNR**(pv): Inverse Mean Normalised Rank $= 1 - \overline{r}(pv)$, where $\overline{r}(pv)$ is the mean normalised rank of true targets with p-value $< pv$
- **F1**(pv): harmonic mean of IMNR and Recall:

$$\mathrm{F1}(pv) = \frac{2 \cdot \mathrm{Recall}(pv) \cdot \mathrm{IMNR}(pv)}{\mathrm{Recall}(pv) + \mathrm{IMNR}(pv)} \tag{8}$$

We evaluate across p-value thresholds using hybrid sampling: 30 linearly-spaced values in $[0.01, 1]$.

Step 4: AUC-F1 Computation. The final metric is the Area Under the Curve (AUC) of the F1 score across p-value thresholds. For a random predictor with uniform p-values, recall at threshold pv equals pv, and IMNR equals 0.5 (targets ranked at the 50th percentile on average). Under our pvalue sampling scheme, the theoretical random baseline yields AUC-F1$_{\text{random}} = 45\%$, establishing the performance

floor against which all models are compared. To better interpret the AUC-F1, we normalise it such that the normalised AUC-F1 is between 0 and 1, and 0.5 corresponds to the random baseline:

$$\text{AUC-F1}_{\text{normalised}} = 0.5 + 0.5 \times \frac{\text{AUC-F1} - \text{AUC-F1}_{\text{random}}}{1 - \text{AUC-F1}_{\text{random}}} \quad \textbf{(9)}$$

Aggregation Strategies. The number of ground-truth targets varies across experiments, requiring different aggregation strategies for *in vitro* and *in clinico* evaluation.

In Vitro: Two-Level Bootstrap for Scenario-Level Performance. For *in vitro* benchmarks, we aggregate results across multiple experiments within each scenario (e.g., zero-shot cell line, zero-shot perturbation). Since experiments have heterogeneous numbers of targets, we employ a two-level bootstrap ($B = 300$ iterations):

1. **Within-experiment selection**: For each experiment S_i with m_i targets, uniformly sample one target, yielding tuples (r_i, p_i).
2. **Experiment-level resampling**: Draw n experiments with replacement.
3. **Compute AUC-F1** on the resampled set.

The final estimate is reported as the median with 95% confidence intervals (38, 39). This ensures each experiment contributes equally regardless of target count.

In Clinico: Per-Sample Bootstrap for Disease–Cell Type Performance. For *in clinico* benchmarks, we compute performance separately for each disease–cell type combination to assess translational reliability in specific biological contexts. For each sample S_i independently, we perform $B = 300$ bootstrap iterations:

1. **Within-sample resampling**: Draw m_i targets with replacement from sample S_i.
2. **Compute AUC-F1** on the bootstrap sample.

This yields per-sample AUC estimates with confidence intervals, enabling identification of contexts where the model performs well versus poorly (38, 40, 41).

Adapting TwinBench to Assess Mode Collapse in Generative Models. TwinBench can be readily adapted to evaluate conditional generative models (such as STATE (13)), where the objective is to predict a final cellular state $\mathbf{x}_2$ given an initial state $\mathbf{x}_1$ and a perturbation, *i.e.*, $\hat{\mathbf{x}}_2 = f(\mathbf{x}_1)$. A common failure mode in this setting is "conditional posterior collapse" or "mode collapse", where the model ignores the specific initial condition $\mathbf{x}_1$ and generates a generic response (or simply the identity). To quantify this, we extend the empirical p-value framework. In the context of STATE, the model comprises two components: an embedding function f_{emb} that maps gene expression to a latent space, and a transition function f_{trans} that predicts the perturbed state. Given input $\mathbf{x}_1$ and target $\mathbf{x}_2$, we compute $\mathbf{z}_1 = f_{\text{emb}}(\mathbf{x}_1)$, $\mathbf{z}_2 = f_{\text{emb}}(\mathbf{x}_2)$, and $\hat{\mathbf{z}}_2 = f_{\text{trans}}(\mathbf{z}_1)$. The evaluation metric is the cosine similarity between predicted and true states in latent space: $S = \text{CosSim}(\hat{\mathbf{z}}_2, \mathbf{z}_2)$.
For the permutation test, we randomly permute the features of $\mathbf{x}_1$ to obtain $\tilde{\mathbf{x}}_1^{(k)}$, compute $\tilde{\mathbf{z}}_1^{(k)} = f_{\text{emb}}(\tilde{\mathbf{x}}_1^{(k)})$ and $\hat{\mathbf{z}}_2^{(k)} = f_{\text{trans}}(\tilde{\mathbf{z}}_1^{(k)})$, then calculate $S_k^{\text{perm}} = \text{CosSim}(\hat{\mathbf{z}}_2^{(k)}, \mathbf{z}_2)$. The empirical p-value is:

$$p\text{-value} = \frac{1 + \sum_{k=1}^{N_R} \mathbf{1}\left[S_k^{\text{perm}} \geq S\right]}{N_R + 1} \quad \textbf{(10)}$$

A non-significant p-value implies that the model's prediction accuracy is independent of the specific gene–gene correlation structure in $\mathbf{x}_1$, indicating mode collapse where the model predicts an average response rather than the perturbation-specific effect.

Description of Scenarios in TwinBench

TwinBench is a benchmarking framework specifically designed to assess the capability of Virtual Cell models to generalise out-of-distribution in real-life scenarios. Unlike conventional benchmarks that may allow models to exploit in-distribution patterns, TwinBench enforces strict separation between training and evaluation data to measure true generalisation performance. The framework defines two complementary evaluation scenarios that mirror practical challenges in the drug discovery pipeline:

In Vitro Scenario: Zeros shot cell line. For *in vitro* benchmarking, the data was split by cell line identity, ensuring that cell lines present in the test set were entirely excluded. We reused the same split as in STATE article to ensure consistency (13).

In Clinico Scenario: Therapeutic Target Validation. The *in clinico* scenario evaluates model predictions against clinically validated drug-target-disease associations from the Open Targets Platform (version 25.06) (42), providing a direct assessment of translational relevance. We extracted drug–target–disease associations for compounds that have reached Phase III or Phase IV clinical status. For each indication, we compiled the set of protein targets for all treatments that have successfully completed late-stage clinical trials, providing a gold-standard reference for retrospective validation.

Multiomics Interactome

We used a proprietary interactome comprising 15K nodes and 201K edges, including 126K protein-protein interactions and 75K transcription interactions. Genes and proteins were treated as equivalent nodes. The interactome was built by aggregating and harmonising protein and gene interaction data from close to a hundred databases including BioGRID (43), Reactome (44), and IntAct (45). Only experimentally determined entries were considered, excluding computationally inferred ones. Interactions were restricted to direct physical connections rather than those potentially mediated by unknown intermediary molecules. Recommended filtering for

high confidence was applied for each database when available. We included only interactions found in at least two independent databases, unless they originated from a single database with high standards of manual curation, such as SIGNOR (46).

Linear Ridge Regression as a baseline for recommendation systems

To evaluate the comparative performance of our proposed models, we employ Linear Ridge Regression as a robust baseline Figure S1. In the context of collaborative filtering, we model the recommendation problem as an item-item regression task, where the goal is to learn a weight matrix W that reconstructs the user-item interaction matrix X. Because user-item interaction data is inherently sparse and high-dimensional, standard least squares estimation is prone to overfitting and numerical instability. Ridge regression addresses this by introducing an L_2 regularization term to the objective function:

$$\hat{W} = \underset{W}{\operatorname{argmin}} \left(\|X - XW\|_F^2 + \lambda \|W\|_F^2 \right) \quad \textbf{(11)}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\lambda > 0$ is a hyperparameter controlling the regularization strength. The constraint $diag(W) = 0$ is typically applied to prevent the model from learning identity mappings. Solving this optimisation problem via its closed-form analytical solution allows for computationally efficient training, providing a deterministic and competitive baseline consistent with established linear methods such as EASE (20).

To assess the impact of input representation on linear baseline performance, we evaluated three distinct preprocessing strategies applied to pseudobulk expression profiles:

1. **Geneformer embeddings**: Pseudobulk profiles were passed through the pre-trained Geneformer foundation model to obtain fixed-dimensional latent representations, capturing rich transcriptomic context learned from large-scale single-cell atlases. While Geneformer is typically applied to single-cell data, its rank-based tokenisation scheme motivated our use of pseudobulk profiles as input. By aggregating cells before embedding, we aim to maximise the signal-to-noise ratio of gene rankings, reducing dropout-induced noise present in individual cells. This strategy is also used in TwinCell as a preprocessing step and allows a fair comparison. Future work will explore the differences with embedding individual cells first before averaging the resulting latent vectors.

2. **Highly Variable Genes (HVG)**: The top 5,000 highly variable genes were selected across the training set using standard variance-based filtering. The samples are then pseudobulked and normalised using log1p.

Network Medicine Baseline

Network Medicine relies on computing the topological distance between observed DEGs and drugs defined by their targets within the interactome (21). Unlike the other models that learn to predict targets directly, this method calculates a proximity score (z-score) assessing whether drug targets are closer to the disease module (DEGs) than expected by chance in the interaction graph. Because this approach inherently includes a permutation test to compute its specific z-score and p-value (comparing the observed distance to a distribution of distances generated by random sets of proteins), we used these pre-computed metrics directly within the TwinBench framework.

Omics Dataset Description

In vitro dataset. To train and benchmark our models, we utilised two complementary large-scale single-cell perturbation resources: the *scPerturb* database (11) and the *Tahoe-100M* atlas (12).

The *scPerturb* database is a harmonised collection of single-cell perturbation response data, aggregating 44 diverse datasets primarily consisting of scRNA-seq experiments, standardised into a uniform format to facilitate machine learning applications. The data spans a wide variety of biological contexts, primarily focusing on human cancer cell lines widely used in pharmacological screening, including K562 (myeloid leukemia), MCF7 (breast cancer), A549 (lung carcinoma), and HEK293T (embryonic kidney). Additionally, the database includes datasets derived from primary tissues, such as peripheral blood mononuclear cells (PBMCs) and cortical organoids, ensuring biological diversity in the training distribution. After quality control checks, we decided to focus on the 12 drug datasets involving the application of small molecules, cytokines, or drug combinations.

The *Tahoe-100M* atlas represents a giga-scale single-cell perturbation resource comprising 100 million transcriptomic profiles measuring how each of 1,100 small-molecule perturbations impact cells across 50 cancer cell lines (12). While scPerturb provides perturbation data from multiple independent studies, Tahoe-100M offers a unified, large-scale chemical perturbation screen with controlled experimental design.

In clinico dataset. We built atlases using our internal pipeline on psoriasis, ulcerative colitis and Parkinson disease. Regarding the systemic lupus erythematosus (SLE) atlas, we selected dataset GSE174188 (19), which comprises over 1.2 million cells profiled from 161 SLE patients and 99 healthy controls.

Raw, annotated count matrices were obtained from the cellxgene portal. scRNA-seq data was reprocessed using DeepLife's pipeline, including cell- and gene-level quality control. Briefly, cells were filtered based on total UMI counts (maximum of 8,000), number of detected genes (minimum of 300 and maximum of 2,200), doublet probability estimated using Scrublet (47), and mitochondrial RNA content, with cells exceeding 20% mitochondrial reads excluded as likely stressed or dying cells. Highly variable genes were identified using Scanpy's `pp.highly_variable_genes` function, the top 4,000 highly variable genes were retained for downstream analyses. Dimensionality reduction was performed by principal component analysis (PCA) on the highly variable genes, computing the first 15 principal components

using `pp.pca`. To mitigate technical batch effects across samples, Harmony (48) was applied to the PCA embeddings. A neighbourhood graph was then constructed on the first 50 Harmony-adjusted principal components using `pp.neighbours` with 15 nearest neighbours. For two-dimensional visualisation, the neighbourhood graph was embedded using UMAP via `tl.umap`, with a minimum distance parameter set to 0.5. Cell clustering was performed using the Leiden algorithm on the integrated neighbourhood graph. To improve cellular resolution, selected clusters were iteratively re-clustered at varying resolution parameters, enabling finer subdivision of heterogeneous populations. Cluster identities were assigned through manual annotation based on marker genes reported in the literature, complemented by automated cell type labels generated using Pan-Azimuth. All preprocessing and analyses steps of scRNA-seq data were run in python 3 using Scanpy (49) and anndata (50) unless otherwise stated.

Supplementary Materials

Table S1. Top 20 predicted proteins from TwinCell analysis of SLE atypical B cells ranked by the TwinCell probability score; Score: TwinCell probability; Rank %: percentile rank in the interactome; p-value: permutation test significance; z-score: standardised effect size; is_deg: whether the gene is differentially expressed; n_degs_impacted: number of DEGs predicted to be affected by the target ($\sum_i P(DEG_i \mid t, \mathbf{x}) > 10^{-3}$).

Gene	Score	Rank	Rank (%)	p-value	z-score	is_deg	n_degs
STAT2	2.0e-2	0	0.000	0.010	26.72	No	15
TLR7	1.3e-2	1	0.006	0.020	6.32	No	14
IFNAR2	8.2e-3	2	0.013	0.010	17.89	No	15
DDOST	7.3e-3	3	0.019	0.010	10.62	No	3
NEIL1	7.2e-3	4	0.025	0.010	11.79	Yes	2
RPN1	6.9e-3	5	0.032	0.030	4.71	No	5
RIG-I	6.8e-3	6	0.038	0.010	22.11	No	12
IRF7	6.7e-3	7	0.044	0.010	11.28	No	16
IFIT3	6.7e-3	8	0.051	0.010	21.55	Yes	2
ISG15	6.6e-3	9	0.057	0.010	119.26	Yes	11
XRCC1	6.0e-3	10	0.063	0.010	13.57	No	2
IFNAR1	5.5e-3	11	0.070	0.010	11.01	No	15
ARPC3	5.5e-3	12	0.076	0.020	7.39	No	16
USP18	5.4e-3	13	0.082	0.010	575.95	Yes	10
FBLN2	5.1e-3	14	0.089	0.010	284.95	Yes	2
HERC3	4.9e-3	15	0.095	0.010	48.18	No	5
IFIT1	4.9e-3	16	0.101	0.020	9.81	Yes	2
COL4A5	4.8e-3	17	0.108	0.010	54.46	No	3
CTCF	4.8e-3	18	0.114	0.139	1.28	No	16
LIG3	4.8e-3	19	0.120	0.010	57.66	No	1

Table S2. Open Targets approved drug targets in SLE ranked by TwinCell analysis of atypical B cells. This table shows the ranking of clinically validated SLE targets (from Open Targets, phase III/IV clinical trials) according to TwinCell's target prioritisation score. Rank percentage indicates the position within the full interactome (15,799 proteins).

Gene	Score	Rank	Rank (%)	p-value	z-score	is_deg	n_degs
TLR7	1.3e-2	1	0.006	0.020	7.12	No	14
IFNAR1	5.5e-3	11	0.070	0.010	7.04	No	15
JAK2	1.3e-3	102	0.646	0.059	1.86	No	22
JAK3	1.0e-3	131	0.829	0.020	2.92	No	21
TLR9	9.6e-4	137	0.867	0.059	2.75	No	20
TYK2	8.5e-4	147	0.930	0.040	1.15	No	22
JAK1	6.4e-4	209	1.323	0.446	-0.11	No	22
IMPDH2	5.4e-4	257	1.627	0.089	1.75	No	20
VDR	5.3e-4	265	1.677	0.099	0.23	No	22
IFNG	5.2e-4	267	1.690	0.287	-0.08	No	22
IL2RB	4.0e-4	400	2.532	0.010	12.94	No	20
NR3C1	3.6e-4	443	2.804	0.564	-0.25	No	22
ESR1	2.5e-4	707	4.475	0.851	-0.71	No	22
DHFR	2.5e-4	714	4.519	0.228	0.40	No	22
IL23A	2.2e-4	852	5.393	0.079	0.89	No	22
ESR2	2.2e-4	859	5.437	0.891	-0.76	No	22
PGR	1.6e-4	1219	7.716	0.624	-0.28	No	22
IL12A	1.5e-4	1356	8.583	0.030	2.42	No	22
HSPA8	1.5e-4	1366	8.646	0.356	-0.09	No	22
S1PR1	9.8e-5	2090	13.229	0.743	-0.24	No	22
CD22	7.0e-5	2665	16.868	0.030	2.04	No	22
PTGS2	6.9e-5	2673	16.919	0.812	-0.25	No	22
PTGS1	6.7e-5	2714	17.178	0.614	-0.42	No	11
IL12B	6.0e-5	2905	18.387	0.040	1.86	No	22
IL2RG	3.7e-5	3612	22.862	0.673	-0.45	No	22
IL2RA	3.0e-5	3861	24.438	0.356	0.13	No	22
TNFSF13B	2.5e-5	4057	25.679	0.158	-0.06	No	22
HMGCR	2.2e-5	4200	26.584	0.653	-0.48	No	8
CD40LG	1.3e-5	4790	30.318	0.525	-0.27	No	22
TNFRSF13C	2.8e-6	6018	38.091	0.416	-0.09	No	22
FAAH	2.1e-6	6220	39.370	0.327	-0.22	No	22
MS4A1	1.6e-6	6387	40.427	0.178	-0.03	No	22
TRPV1	6.6e-7	6940	43.927	0.485	-0.16	No	20
CLEC4C	7.7e-10	8950	56.649	0.594	-0.02	No	17

Table S3. Approved drug targets per disease used in *in clinico* benchmarking. Targets are categorised based on their presence in the training set or not.

Disease	Category	Targets
SLE	In train	DHFR, ESR1, ESR2, HMGCR, HRH1, HSPA8, IMPDH2, JAK1, JAK2, JAK3, NR3C1, PGR, PPAT, PTGS1, PTGS2, S1PR1, TLR7, TYK2
	Not in train	CD22, CD40LG, CLEC4C, FAAH, IFNAR1, IFNG, IL12A, IL12B, IL23A, IL2RA, IL2RB, IL2RG, IMPDH1, MS4A1, TLR9, TNFRSF13C, TNFSF13B, TRPV1, VDR
Parkinson	In train	ACHE, ADORA1, ADORA2A, ADORA2B, ADORA3, ADRB1, CACNA1C, CACNA1D, CACNA1F, CACNA1S, CHRM1, CNR1, COMT, DRD1, DRD2, DRD3, DRD4, DRD5, HRH1, HTR1A, HTR2A, HTR2C, MAOB, OPRM1, SCN4A, SCN5A, SLC6A4
	Not in train	BCHE, CHRM2, CHRM3, DDC, GLP1R, GRIA1, GRIA2, GRIA3, GRIA4, GRIN1, GRIN2A, GRIN2B, GRIN2C, GRIN2D, GRIN3A, GRIN3B, HRH3, OPRD1, OPRK1, SCN10A, SCN11A, SCN1A, SCN2A, SCN3A, SCN7A, SCN8A, SCN9A, SLC6A2, SNAP25
Ulcerative Colitis	In train	ALOX5, CD86, CRBN, CUL4A, DDB1, FKBP1A, JAK1, JAK2, JAK3, NCBP1, NCBP2, NR3C1, PDE4A, PDE4B, PDE4C, PDE4D, PPARG, PPAT, PTGS1, PTGS2, RBX1, S1PR1, S1PR4, S1PR5, TYK2, XDH
	Not in train	CD80, IL12A, IL12B, IL23A, ITGA4, ITGB7, MADCAM1, TLR9, TNF, VDR

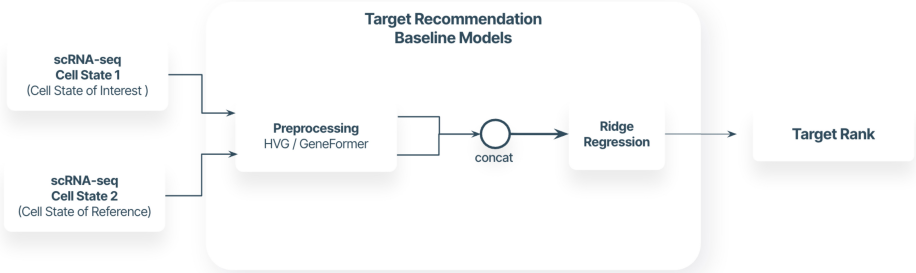

Fig. S1. Linear Ridge Regression as a baseline for recommendation systems. Two input cell states are processed using various feature-extraction methods: selection of highly variable genes (HVGs) and Geneformer embeddings on pseudobulk profiles. The processed features are concatenated and fed into a Ridge regression model trained to predict target scores, adapting the EASE architecture (20) to this domain.