

Maintaining a Living Virtual Cell Benchmark with Item Response Theory

Author One¹, Author Two¹

¹*Northwestern University*

Abstract. We present Virtual Cell Bench, a self-updating benchmark for virtual cell models that works like a standardized exam. Most benchmarks score models with one average number, such as mean squared error or accuracy. However, we argue that some test items (samples) are easy for every model and some are hard for all of them, so the average hides this information from readers. Using item response theory, Virtual Cell Bench gives each item a difficulty and each model an ability, both on one shared scale with error bars. Models earn more ability for solving hard items than easy ones, so true breakthroughs stand out. The current release covers two aspects, representation quality and perturbation response, with 424,478 cell-type items from fourteen annotation panels and 88,164 perturbation items from fifteen screens, about 110 million cells in all. In Virtual Cell Bench, we estimate the difficulty of each test item, which tells us how difficult a test item is for the current research field. To keep Virtual Cell Bench informative over time, we use a pipeline that automatically downweights items that have become easy or stopped discriminating models. Overall, we evaluate [N] virtual cell models and present their ranking based on abilities instead of raw averaged metrics.

1 Introduction

Single-cell sequencing measures the activity of thousands of genes in one cell at a time. A virtual cell model learns to represent gene expressions as latent representations and predict the response after intervention. [Figure 1](#) shows the whole system on one page; the rest of this paper explains each of its parts.

Scoring these models is harder than it looks. The usual score is an average over a fixed set of test cases. This has three problems. First, test cases differ in difficulty, so the average depends on which cases happen to be in the set as much as on the model. The same model scores 0.89 on one of our panels and 0.39 on another, not because it changed but because the test suite did. Second, a fixed test set gets stale. As models improve, the easy samples stop telling models apart. Once the test data is public, it also finds its way into training sets, and a high score may reflect memory rather than skill. Third, an average cannot say why a score is low: whether the cases were impossible, the labels were wrong, or the model was weak.

[Figure 2](#) draws the three problems and what a shared scale changes.

Standardized exams solved the same problems long ago. An exam such as the SAT does not report the fraction of questions answered correctly. It measures the difficulty of every question, measures the ability of every test taker on the same scale, retires questions that no longer discriminate, and writes new ones, while keeping scores comparable across years. The statistical tool behind this is called item response theory.

This paper describes a benchmark for virtual cell models built on that idea. Every test case, which we call an *item*, has a difficulty. Every model has an ability. Both sit on one scale, and both come with an error bar. The scale is defined by a pool of simple methods, such as principal components, that are cheap to run and cannot have memorized the test data. Larger pretrained models are placed on the scale afterwards, so their scores can be read next to the simple methods they are supposed to beat.

The rest of the paper walks through the system from the bottom up. [Section 2](#) explains what a model is asked to do. [Section 3](#) describes the data and how items are built from it. [Section 4](#) explains how models are scored and how difficulty and ability are measured. [Section 5](#) lists the models and reports what the scale shows, including the first placement of seven pretrained foundation models. [Section 6](#) explains how the benchmark is kept alive as models improve.

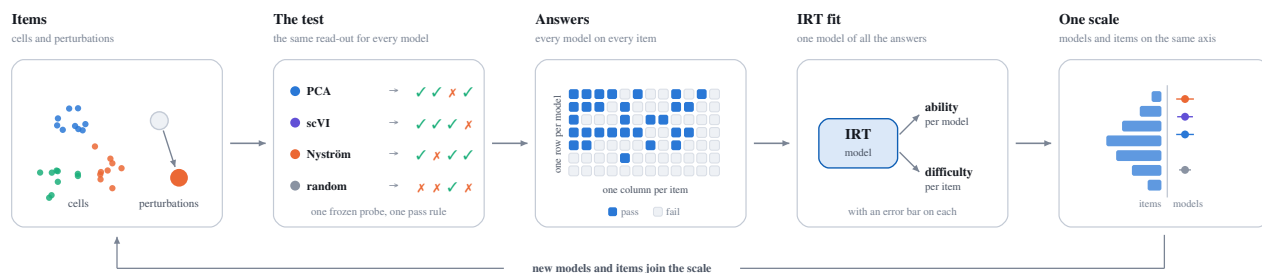


Figure 1 The pipeline. Cells and perturbations are the items. Every model takes the same test. The answers form a pass/fail table with one row per model and one column per item. One fit of the item response model gives every model an ability and every item a difficulty, each with an error bar, on one shared scale. New models and new items join that scale, which is what keeps the benchmark alive.



Figure 2 What an average score cannot do, and what a scale does instead. Top left: the same model scores 0.89 and 0.39 on two panels; on the scale the items spread out by difficulty and the model keeps one ability. Top right: an average moves every time a panel or a model is added, while an ability on an anchored scale keeps its value and its error bar across versions. Bottom left: only items near a model's ability carry information about it, so a few hundred well-placed items can place a model. Bottom right: a low average cannot say why; a difficulty per item can be related to a reason, such as a shallow cell or a noisy label.

2 What a Model Is Asked to Do

The benchmark has two parts. Both use the same format: a model either passes an item or fails it. The two parts test different skills and are scored on separate scales.

Part A: recognizing cell types. Here an item is one cell. The model turns the cell's gene activity into a short list of numbers, which we call its representation or embedding. We then train one simple classifier on the representations of a separate set of training cells, with fixed settings, and ask it to name the type of the test cell. The item is passed when the name matches the type that the data's authors assigned. The classifier is the same for every model. The only thing that differs between models is the representation, so a difference in passes is a difference in representations. We call this classifier *frozen* because nothing about it is tuned per model.

Part B: predicting the effect of a perturbation. Here an item is one experimental condition: one perturbation applied in one context. A perturbation is a gene that was switched off or on, a drug at a dose, or a signaling protein (a cytokine) added to the culture. A context is a cell line, a donor, or a cell type; when a dataset has more than one kind of context, the context is the combination, for example a donor and a cell type. The measured

effect of a condition is the average gene activity of its cells minus the average of matching untreated control cells. The cells inside a condition are repeats of the same measurement, not separate items. A model has to predict this effect for conditions it was not trained on.

A prediction is scored by how well it correlates with the measured effect over the 2,000 most variable genes. The item is passed when that correlation is higher than the one achieved by a fixed reference, which we call the *mean shift*: the average effect of all training conditions in the same context. This reference is chosen on purpose. Recent benchmarks ([authors to fill], 2025; Wu et al., 2025; Ahlmann-Eltze et al., 2025) found that this plain average matches deep models on standard metrics, because a large part of any perturbation’s effect is shared by every perturbation in that context. A pass therefore gives credit only for the part of a prediction that is specific to the perturbation, not for the part that every perturbation shares.

3 Where the Data Comes From

Sources. Part A draws on seven public sources: Tabula Sapiens v2 (The Tabula Sapiens Consortium, 2026); four curated atlases from the CELLxGENE collection, namely the Hao 2021 blood reference (Hao et al., 2021), the adult heart atlas (Litviňuková et al., 2020), the Human Lung Cell Atlas core (Sikkema et al., 2023) and the cross-tissue immune atlas (Domínguez Conde et al., 2022); the Parse 10 million PBMC cytokine atlas (Parse Biosciences, 2026); and scBaseCount (Youngblut et al., 2025), a repository of public single-cell submissions reprocessed in one uniform way with machine-made labels. Part B uses thirteen screens from the scPerturb collection (Peidli et al., 2024), the Tahoe-100M drug atlas (Zhang et al., 2025) and the Parse cytokine atlas; Table 2 lists them. All sources are public under CC0, CC BY or CC BY-NC licences.

Panels for Part A. Each annotation source is turned into one or more *panels* by the same recipe. We first choose the unit that is kept apart for testing: whole donors, whole studies, or whole tissues. Everything from a held-out unit goes to the test side and nothing from it is used for training. This matters. If cells from the same donor were on both sides, a classifier could learn to recognize the donor and use that as a stand-in for the cell type. We call each combination of context, cell type and donor a *group* (a stratum in the tables). Within each group we keep at most 200 training cells and 50 to 100 test cells, and we drop groups with fewer than 20 cells. Test cells whose type never appears on the training side stay in the panel but are not scored, since no classifier could name them.

Table 1 lists the fourteen panels. Eleven hold out donors, which is the usual setting. Three hold out whole studies or tissues, so a cell type must be recognized in a setting the classifier has never seen. The lung atlas and the immune atlas appear twice, once under each split. The scBaseCount panels differ from the curated atlases in one way that matters later: each holds every sample that passes a quality filter, up to 200 per tissue, from many different public submissions, and the labels come from the repository’s own per-cell annotation, so the label vocabulary is not the same from one sample to the next. Two more panels were built and set aside as floors: a Parse panel whose label is the cytokine a cell was exposed to, which no method can answer above a few percent, and a mouse twelve-tissue scBaseCount panel with 109 different label vocabularies, on which 60% of items are passed by nothing. Six mouse single-tissue panels were also built; this release pools human panels only. Table 10 in the appendix breaks the Tabula Sapiens panel down by tissue.

Items for Part B. Every screen is mapped to one common table with the perturbation, its target genes, the kind of perturbation, the context, the dose, the time point and a flag for control cells. Cells with fewer than 500 counts or 200 detected genes are removed. The average activity per condition is then computed in one pass over the raw counts. For Tahoe-100M this pass streams 95.6 million cells from the public files and reduces them to 56,827 conditions, of which 53,972 have at least 50 cells. For Parse it streams 9.7 million cells and yields 13,967 conditions over 90 cytokines, 12 donors and 18 cell types, each with its own untreated control. The median number of cells per item ranges from 74 for Nadig HepG2 to 1,363 for Tahoe-100M. Each item also carries a few descriptors for later analysis: the cell count, the size of the effect (the length of the effect vector), the target gene or the drug’s chemical structure, and the dose.

Table 1 Part A panels. Cell types counts the distinct labels among a panel’s items; none pass is the fraction of items that no model labels correctly; mean difficulty is on the pooled scale of Section 4.3, over informative items. Panels hold out whole donors unless marked otherwise; the held-out units, contexts and groups of every panel are listed in Table 9.

Panel	Cell types	Train cells	Items	None pass	Mean diff.
Tabula Sapiens v2	102	146,800	20,026	11%	-0.57
Hao 2021 PBMC	24	42,441	7,999	2%	-1.55
Adult heart	59	81,001	14,396	8%	-0.10
HLCA core	52	134,615	11,389	4%	-1.24
HLCA core (studies held out)	51	85,476	32,221	2%	-1.19
Cross-tissue immune atlas	41	66,684	20,548	3%	-0.84
Cross-tissue immune atlas (tissues held out)	38	59,565	23,038	5%	-0.97
Parse 10M PBMC	18	45,948	16,811	1%	-1.92
scBaseCount, lung	25	145,543	64,635	7%	-0.93
scBaseCount, skin	15	68,112	28,812	11%	-0.47
scBaseCount, bone marrow	20	104,777	41,262	14%	-0.23
scBaseCount, liver	19	51,432	17,559	15%	-0.17
scBaseCount, blood	14	129,019	46,625	11%	-0.35
scBaseCount, 12 tissues (tissues held out)	22	180,127	79,157	32%	-0.05
Total	299	1,341,540	424,478	12%	

Table 2 Part B items by source. An item is a condition with at least 50 quality-passing cells and a matched control. Contexts are cell lines, culture conditions, or donor and cell type pairs. Tahoe-100M conditions are compound, dose and cell line triples; Parse conditions are cytokine, donor and cell type triples; the Replogle genome-wide screen is the largest genetic source. Cell counts are after quality filtering.

Dataset	Perturbation	Contexts	Perturbations	Items	Cells
Tahoe-100M	compound, 3 doses	50	379	53,972	95.6M
Parse 10M PBMC	cytokine	215	90	13,967	9.6M
Replogle 2022, K562 genome-wide	CRISPRi	1	9,326	9,326	1.97M
sciPlex3, Srivatsan 2020	compound, 4 doses	3	188	2,423	762k
Replogle 2022, K562 essential	CRISPRi	1	1,832	1,832	303k
Nadig 2024, Jurkat	CRISPRi	1	1,815	1,815	245k
Replogle 2022, RPE1	CRISPRi	1	1,611	1,611	225k
Nadig 2024, HepG2	CRISPRi	1	1,061	1,061	104k
McFarland 2020	compound, 7 doses	196	16	866	166k
Frangieh 2021	CRISPR knockout	3	229	669	216k
Norman 2019	CRISPRa, single and double	1	236	236	111k
Tian 2021, neurons	CRISPRi	1	183	183	32k
Adamson 2016	CRISPRi	1	87	87	63k
Tian 2019, iPSC	CRISPRi	1	81	81	29k
Tian 2019, neurons	CRISPRi	1	35	35	8k
Total				88,164	109.5M

4 How Models Are Scored and How the Scale Is Measured

Figure 1 shows the pipeline. Both parts of the benchmark end in the same object: a table of passes and fails with one row per model and one column per item. One fit turns that table into an ability for every model and a difficulty for every item.

4.1 From data to a pass/fail table

1. **Build the items.** Part A: build the panels of Section 3 and record each test cell’s annotated type. Part B: map every screen to the common table, filter the cells, and compute each condition’s effect as the average activity of its cells minus the average of its control cells.
2. **Run every model on every item.** Part A: for each panel, fit the model on the training cells only, compute representations for both sides, train the frozen classifier on the training representations, and let it name the test cells. Part B: split the conditions into five folds; fit the model on four folds and predict the fifth; repeat so that every condition gets exactly one prediction from a model that never saw it.

3. **Decide pass or fail.** Part A: a pass when the predicted type equals the annotated type. Part B: a pass when the prediction’s correlation with the measured effect beats the mean shift’s correlation on the same item. The raw scores are kept too.
4. **Assemble the table.** Rows are models, columns are items. A pair that was never run, for example a method that only applies to screens with several contexts, is left empty and plays no part in the fit. Items from different panels stay distinct even when the same cell appears in two panels under two splits.

Every model faces the same test: the same panels and folds, the same frozen classifier, the same pass rule. Nothing is tuned per model.

4.2 Difficulty and ability

The scale rests on one assumption, borrowed from standardized testing. Each model j has an ability θ_j and each item i has a difficulty b_i , both plain numbers. The chance that the model passes the item depends only on the gap between the two:

$$P(\text{model } j \text{ passes item } i) = \sigma(\theta_j - b_i), \quad \sigma(z) = \frac{1}{1 + e^{-z}}. \quad (1)$$

This is the one-parameter model of item response theory, often called the Rasch model. When ability equals difficulty the chance of a pass is one half. One unit above, it is 73%; two units above, 88%; one unit below, 27%. The unit of the scale is a logit: one extra unit of ability multiplies the odds of passing any item by about 2.7.

The model also says which items are worth asking. An item tells you the most about a model whose ability is close to the item’s difficulty, where the outcome is uncertain. An item that a model almost always passes, or almost always fails, says little about that model. This is why a few hundred well-chosen items can place a model as precisely as a few hundred thousand poorly matched ones, and why items that every model passes carry no information about the current pool.

4.3 Fitting the scale

All abilities and difficulties are estimated at once from the pass/fail table. The fit looks for the values that make the observed passes and fails most likely under [equation \(1\)](#). Two mild constraints are added. Abilities are pulled gently toward zero, and difficulties a little more gently. The second constraint does two jobs. It pins down where zero sits on the scale, which the passes and fails alone cannot do because only the gap between ability and difficulty enters [equation \(1\)](#). It also keeps items that every model passes, or every model fails, at a finite value instead of letting them run off to infinity. Such items are flagged and left out of every summary in this paper. They carry no information about the current pool, though they may once a stronger model arrives.

The fit alternates between the two sides. With difficulties held fixed, every ability is nudged toward its best value; with abilities held fixed, every difficulty is nudged likewise; the two sweeps repeat until nothing moves by more than a millionth. This takes a few hundred sweeps and about eight minutes for the 424,478 Part A items. The same code handles a second, coarser fit for Part A in which an item is a whole group of cells answered k times correctly out of n . Each part of the benchmark is fitted on one pooled scale: all fourteen panels together, and all fifteen screens together. The link between panels is the models, which answer every panel under the same name. Every panel and screen also gets its own fit, whose abilities are comparable within that fit but not across fits.

Every model and every item also receives a standard error and a fit statistic. The standard error says how precisely the ability or difficulty is known; it is small when many informative items were answered. The fit statistic, called outfit, compares a model’s passes and fails with what its ability predicts. A value near 1 means the model behaves as the scale expects. A value well above 2 means the model passes some items its ability says it should fail, and fails some it should pass. Such a model is flagged on the board rather than silently ranked.

Checks. Two checks are run before any reading of the scale. On synthetic data generated from [equation \(1\)](#) with 80 models and 2,000 items, with 30% of the table left empty, the recovered abilities correlate 0.998 with the truth and the difficulties 0.969. On real data, the floors described in [Section 5.1](#), which are random representations, random effects and predicting no change, must come out at the bottom of the scale, and every family in which we vary one setting must come out in the expected order. [Section 5](#) confirms both.

Adding a model without moving the scale. Once a scale exists, a new model is placed on it with every item difficulty held fixed. Only the new model’s ability is estimated. This takes seconds, and no published number changes. When enough new models have been added, everything is refitted together and the scale gets a new version number. Old versions stay available, so a result can always be cited with the version it was measured on. The pretrained models in [Section 5.4](#) were placed this way.

What the numbers mean, and what they do not. Ability is relative to the pool that defined the difficulties. A value of 0.69 for PCA-128D means it passes a typical item of this bank with probability $\sigma(0.69 - b)$. It does not mean the model is good in an absolute sense. Difficulty is likewise relative to the models: an item is hard because these methods fail it. Abilities are comparable across models within a part of the benchmark and, with the caveats of [Section 3](#), across datasets within a part. They are not comparable between Part A and Part B, which are fitted separately.

5 Models and Results

A scale is only as trustworthy as the models that define it. We calibrate it with methods that are cheap, easy to understand, and could not have seen the test data during any kind of pretraining. Within each family we vary one capacity setting, so that abilities should spread out in a known order. We include several families, so that items which one family passes and another fails become visible. Floors pin the bottom of the scale. Pretrained models are placed afterwards. This order matters: it is the only way to read a pretrained model’s ability separately from its exposure to the test data.

5.1 The model pool

[Table 3](#) lists every model. All Part A models see the same input: log-normalized counts on the 2,000 most variable genes of the training panel, standardized per gene. Each is fitted on training cells only and then applied to test cells, so test cells never shape a representation. This rules out UMAP and t-SNE, which have no way to place a new cell. Part B models predict a condition’s effect from two inputs: the effects of the training conditions, and a description of the condition. A gene target is described by a 64-number summary of the gene learned from control cells only. A drug is described by its chemical structure ([Rogers and Hahn, 2010](#)), its dose and its context. Cytokines have neither a target gene nor a structure in their source files, so Parse items carry only indicators of which cytokine and which context; with these inputs the regression models reduce to a cytokine effect plus a context effect, and the nearest-neighbour model reduces to the average effect of the same cytokine in other contexts.

Splits. Part B models are scored out of fold. Perturbations are divided into five folds; each model is trained on four and predicts the fifth, so every condition is predicted by a model that never saw its perturbation. Screens with several contexts are also split over perturbation and context pairs, so that a perturbation seen in some contexts must be predicted in a new one. Part A models are fitted on each panel’s training cells and scored on that panel’s held-out donors, studies or tissues. All models share the same folds, panels, classifier and pass rules.

Fit. Abilities and difficulties are estimated as in [Section 4.3](#). For cells we report both the cell-level fit and the group-level fit; the two order the models identically.

5.2 How hard the items are

[Figure 3](#) shows the fitted difficulties. The 5th to 95th percentiles of the informative items run from -3.5 to 3.2 for cells and from -3.4 to 1.4 for conditions, with standard deviations of 1.9 and 1.5. The abilities of the simple models occupy the shaded band in each panel. 52,890 cells, or 12%, are labelled correctly by no model and 1,247 by every model; 21,733 conditions, or 25%, are passed either by every model or by none, all but 56 of them by none. In Tahoe-100M the fraction passed by none is 28%, in Parse 24%, and in the Tian 2021 neuron screen 61%; among the annotation panels it ranges from 1% for Parse to 32% for the twelve-tissue scBaseCount panel. On the pooled Part A scale the mean difficulty by panel runs from -1.9 for Parse to -0.05 for the twelve-tissue scBaseCount panel; the six scBaseCount panels have means between -0.9 and -0.05 , the eight curated panels between -1.9 and -0.1 . On the pooled Part B scale the mean by dataset runs from -2.6 for Norman to $+0.6$ for Tian 2021 neurons, with Tahoe-100M at -1.0 and Parse at $+0.3$; the spread within a dataset is about 1.5.

Table 3 The model pool. Each row is one family; the setting varied within the family is in the last column. For Part B, d is the number of principal components of the training effects that a model predicts before mapping back to genes, and the kernel width is a multiple of the median distance between training conditions. Learned models are listed last; they are placed on the scale that the simple models define.

Model	What it does	Settings
<i>Part A: representations of a cell, followed by the same frozen classifier</i>		
PCA- k D	top- k principal components	$k \in \{4, 8, 16, 32, 64, 128, 256\}$
Random projection- k D	Gaussian random projection to k dimensions	$k \in \{8, 32, 128\}$, 2 seeds
Nyström- k D	RBF kernel features on PCA-50D	$k \in \{64, 256\}$, kernel width 0.5 or 2
NMF- k D	non-negative matrix factorization	$k \in \{16, 64\}$
ICA- k D	independent components of PCA- k D	$k \in \{16, 64\}$
GMM- k memberships	soft cluster memberships from a Gaussian mixture on PCA-50D	$k \in \{10, 30, 100\}$
Depth only	two numbers, log total counts and log detected genes; a floor	
Random-32D	random numbers; a floor	2 seeds
scVI- k D	variational autoencoder for counts (Lopez et al., 2018), with and without the panel’s context as the batch variable	$k \in \{10, 30\}$
Foundation models	seven pretrained single-cell models, used as fixed cell encoders; Section 5.4	
<i>Part B: prediction of a condition’s effect</i>		
No change	predicts zero effect; a floor	
Random shift	the effect of a random training condition; a floor	2 seeds
Mean shift	the average effect of the training conditions in that context; defines the pass rule	
Mean shift + on-target change	mean shift with the target gene forced down for CRISPRi and knockout or up for CRISPRa	
KNN ($K=k$)	average effect of the k nearest training conditions by description	$k \in \{1, 5, 20\}$
Ridge (d, λ)	ridge regression from description to the top- d components of the effects	$d \in \{10, 50\}$, $\lambda \in \{1, 100\}$
PCA regression- k D	ridge to the top- k components with a fixed penalty; the capacity ladder of this part	$k \in \{2, 5, 10, 20, 50, 100, 200\}$
Gaussian process (kernel width)	Gaussian-process regression with an RBF kernel on the description, to the top-50 components	kernel width 0.5 or 2
Training-fraction ladder	ridge and KNN trained on a random 30% or 10% of the training conditions	2 fractions
Low-rank completion	soft-impute of the perturbation-by-context effect matrix; multi-context screens only	rank $\in \{2, 5, 10\}$

What makes an item hard. Because every item has a number, we can ask what that number tracks. Within the curated atlases, a cell is easier when its type has more training cells: the rank correlation between difficulty and training cells of the same type is -0.35 for Tabula Sapiens, -0.39 for Hao, -0.33 and -0.29 for the two lung panels and -0.25 and -0.12 for the two immune panels. On the single-tissue scBaseCount panels this correlation is between -0.04 and $+0.17$, on the Parse panel $+0.51$, and on the twelve-tissue scBaseCount panel -0.43 . Cells with more reads are easier on every panel, with correlations between -0.02 and -0.38 , most strongly on the scBaseCount skin and lung panels. Across the whole pooled scale the two correlations are 0.00 and -0.19 . Cell type explains between 8% and 50% of the spread of difficulty on a panel, and the full group (type, donor and context) between 42% and 66%, so a good part of what makes a cell hard is which sample it came from, not only which type it is. When the classifier is wrong, its answer is a close relative of the right type (a sibling, parent or child in the Cell Ontology) in only 22% of cases. The hardest types in the curated atlases are sibling immune states, such as activated versus naive CD8 T cells, gamma-delta T cells, innate lymphoid cells and intermediate monocytes, and stromal types that share a lineage. On the scBaseCount panels they are coarse or progenitor labels such as leukocyte, common myeloid progenitor and granulocyte-monocyte progenitor, which points at the labels rather than the biology. For conditions, difficulty rises with the size of the effect in the genetic screens ($+0.16$ in the genome-wide K562 screen and $+0.19$ in RPE1) and falls with it in the drug and cytokine screens (-0.23 for sciPlex3, -0.27 for McFarland and -0.31 for Parse). The raw scores are kept so that a graded fit can be run alongside the pass/fail one.

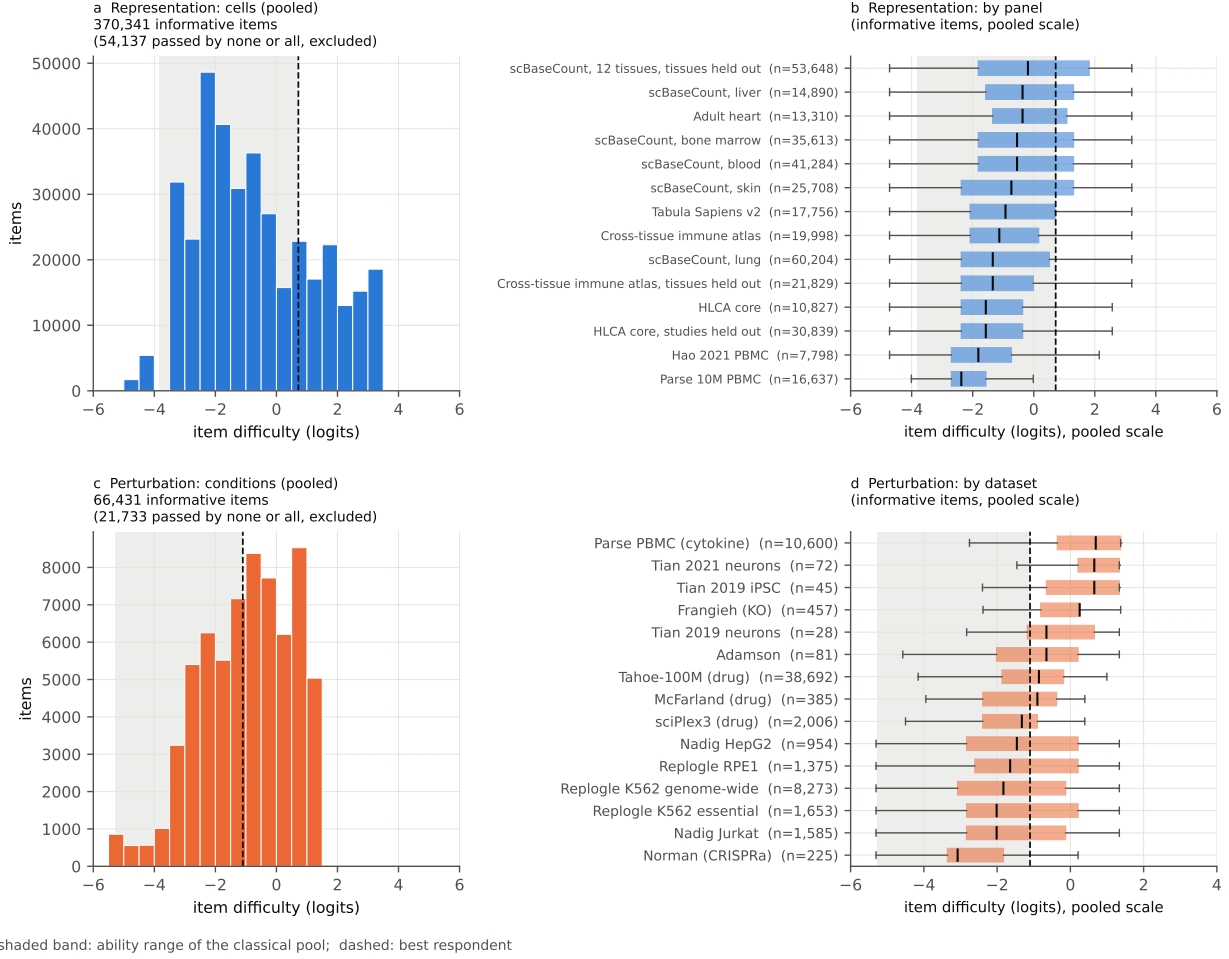


Figure 3 Item difficulty. (a) Part A items, one per cell, on the pooled scale of all fourteen panels. (b) The same items by panel. (c) Part B conditions pooled across datasets and (d) the same conditions by dataset. Boxes span the middle half of the items and whiskers the 5th to 95th percentile. The shaded band is the ability range of the simple models and the dashed line the best of them. An item is informative when at least one model passes it and at least one fails it.

5.3 Abilities of the simple models

Figure 4a gives the Part A abilities on the pooled scale of all fourteen panels. The best simple model is Nyström-256D with kernel width 0.5 at 0.72; tissue-corrected scVI-30D is at 0.72, scVI-30D without batch correction at 0.71 and PCA-128D at 0.69, so the top four lie within 0.03 of one another. PCA-256D, Nyström-64D, ICA-64D and PCA-64D follow at 0.66 to 0.67, then Nyström-256D with width 2 at 0.62 and PCA-32D at 0.53. The next group is Nyström-64D with width 2 at 0.42, NMF-64D at 0.41 and tissue-corrected scVI-10D at 0.37. PCA-16D and ICA-16D are at 0.11, the 128-dimensional random projections at -0.08 , GMM-100 at -0.18 and NMF-16D at -0.37 . Below zero by more than half a unit are PCA-8D (-0.67), GMM-30 (-0.68), the 32-dimensional random projections (-1.30), PCA-4D (-1.50), GMM-10 (-1.88) and the 8-dimensional random projections (-2.70); the depth-only floor is at -3.14 and the random representations at -3.83 . Plain accuracy is 0.618 for the best model and 0.615 for PCA-128D; the eight best models are within one accuracy point of one another (Table 5). Ability rises with dimension in every family: PCA from -1.50 at 4 dimensions to 0.69 at 128, with 256 at 0.67; random projections from -2.70 at 8 to -0.08 at 128, which is 0.8 to 2.0 units below PCA at the same dimension.

Fitted panel by panel, the best model is scVI-30D on five panels: Hao PBMC (3.57), the lung atlas with studies held out (2.82), the immune atlas with tissues held out (2.17), scBaseCount blood (0.53) and scBaseCount bone marrow (0.31). It is Nyström-256D on six: Parse (3.41), the immune atlas with donors held out (2.01), scBaseCount lung (1.44, level with PCA-128D), adult heart (0.63), scBaseCount skin (0.41) and scBaseCount liver (-0.03). It is PCA-128D on the lung atlas with donors held out (2.69) and on Tabula Sapiens (0.98),

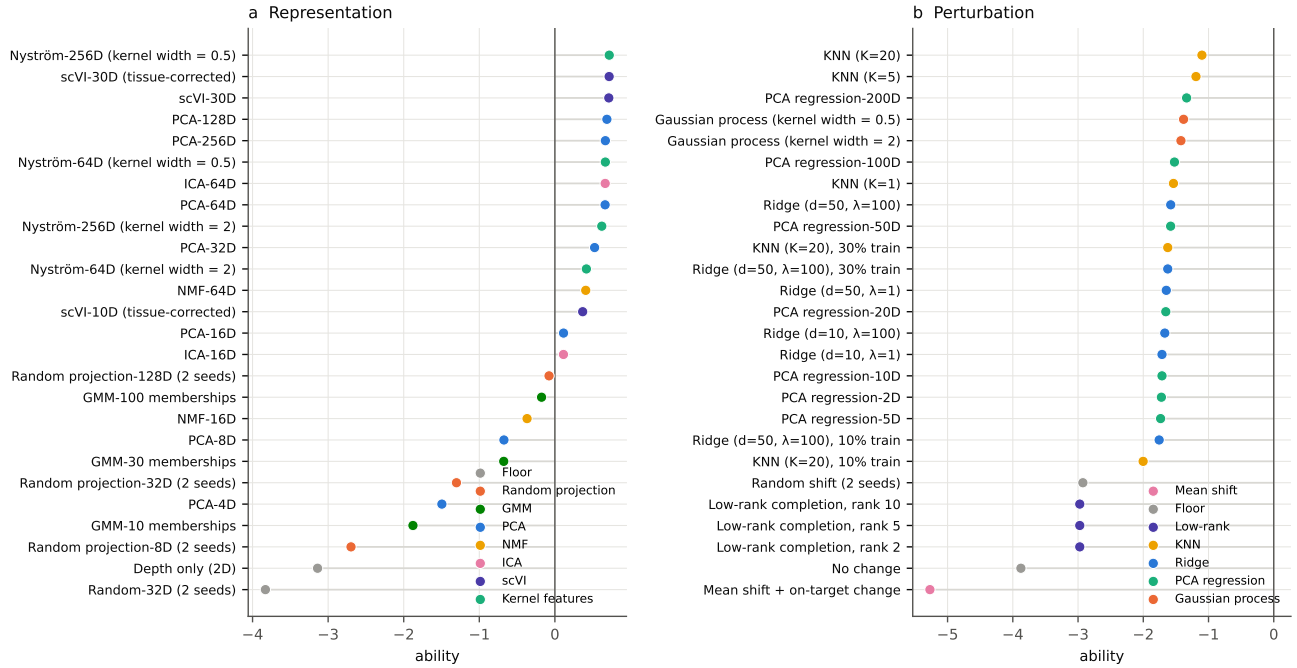


Figure 4 Abilities of the model pool. (a) Part A models on the pooled cell-level scale of all fourteen panels, 424,478 items. (b) Part B models on the pooled scale of all fifteen screens, 88,164 conditions. Colour marks the family; the two seeds of a random model are averaged. The mean shift defines the pass rule and so has no ability of its own. The full tables, with plain accuracy next to ability for Part A, are in [Appendix A](#).

and Nyström-64D on the twelve-tissue scBaseCount panel (-1.15). scVI-30D is below PCA-128D by 0.24 on Tabula Sapiens, by 0.12 on the donor-held-out lung atlas, by 0.18 on the donor-held-out immune atlas, by 0.30 on scBaseCount lung and by 0.13 on the twelve-tissue panel, and above it by 0.39 on scBaseCount blood, 0.26 on bone marrow and 0.08 on skin. On Tabula Sapiens, plain accuracy rises from 0.606 at PCA-32D to 0.654 at PCA-128D while ability rises from 0.55 to 0.98 ([Table 6](#)). The top accuracy per panel is 0.61 to 0.89 on the eight curated panels and 0.39 to 0.70 on the six scBaseCount panels. Between the 60-sample and 200-sample versions of the scBaseCount blood panel, scVI-30D’s accuracy rose from 0.50 to 0.60 and the best simple model’s from 0.46 to 0.56, while on lung, skin and liver the top accuracy changed by less than 0.02.

[Figure 4b](#) gives the Part B abilities on the pooled scale of all fifteen screens in the unseen-perturbation split. The best model is KNN with $K = 20$ at -1.10 , followed by KNN with $K = 5$ at -1.19 , PCA regression with 200 components at -1.34 , the Gaussian processes at -1.38 and -1.42 , PCA regression-100D at -1.52 and KNN with $K = 1$ at -1.54 . The ridge models lie between -1.58 and -1.71 , and the PCA-regression ladder rises from -1.73 at 5 components to -1.34 at 200. The training-fraction ladders are in order: ridge at -1.58 on the full training set, -1.63 at 30% and -1.76 at 10%; KNN ($K = 20$) at -1.10 , -1.63 and -2.00 . Below them are the random shifts at -2.93 , low-rank completion at -2.97 , no change at -3.88 and the mean shift with the on-target change at -5.27 . Ranked by raw correlation instead, the best model in eight of the fifteen screens, Tahoe-100M and Parse among them, is the mean shift or its on-target variant; the exception with the widest margin is Norman, where the Gaussian process reaches 0.69 against 0.49 for the mean shift.

In the unseen-combination split, where a perturbation has been seen in other contexts and the context with other perturbations, the picture differs. On Tahoe-100M, low-rank completion reaches a raw correlation of 0.75 and KNN ($K = 20$) 0.73 against 0.42 for the mean shift, which ranks 18th of 21 models, and 169 of 53,972 conditions are passed by no model. On Parse, PCA regression with 200 components reaches 0.50 against 0.36 for the mean shift and passes 87% of items, with ability 3.74 on that screen; low-rank completion reaches 0.45 and passes 59%; 719 of 13,967 conditions are passed by no model. In Parse’s unseen-cytokine split, KNN ($K = 20$) passes 47% of items and every other model fewer than 17%, none beats the mean shift’s raw correlation of 0.36 on average, and 3,367 conditions are passed by no model.

Table 4 *Pretrained foundation models on the curated Part A panels, next to the best simple models. Each column is one panel’s own fit; abilities are comparable within a column. HLCA d. and s. are the lung atlas with donors and with studies held out; Immune d. and t. the immune atlas with donors and with tissues held out. A dash marks a run still in progress.*

Model	Hao PBMC	Parse PBMC	Heart	HLCA d.	HLCA s.	Immune d.	Immune t.	Tabula S.
TranscriptFormer (sapiens)	+3.44	–	+0.64	+2.49	+1.96	+2.20	+2.34	+0.94
Geneformer V2-316M	+3.09	+3.28	+0.58	+2.65	+1.92	+2.42	+2.55	+0.99
scGPT (whole-human)	+2.43	+2.75	+0.64	+2.67	+2.47	+2.26	+2.30	+0.76
STATE SE-600M	+2.56	+2.70	–0.06	+2.09	+2.33	+1.86	+1.91	+0.69
UCE (4 layer)	+2.32	+2.75	+0.46	+2.43	+2.10	+2.13	+2.11	+0.81
CellPLM 85M	+1.12	+2.27	+0.02	+1.72	+1.95	+1.22	+1.33	+0.48
scPRINT large v1	+0.99	+1.71	–0.49	+1.29	+1.19	+0.71	+0.56	+0.17
scVI-30D	+3.46	+3.34	+0.55	+2.57	+2.80	+1.82	+2.12	+0.74
Nyström-256D (kernel width = 0.5)	+2.97	+3.41	+0.63	+2.61	+2.53	+2.01	+2.14	+0.85
PCA-128D	+2.88	+3.33	+0.57	+2.69	+2.46	+2.00	+2.04	+0.98
Random-32D (seed 0)	–4.17	–3.82	–4.97	–5.00	–4.82	–3.90	–4.49	–4.14

5.4 Pretrained foundation models

Seven pretrained single-cell foundation models were placed on the Part A scale: scGPT (Cui et al., 2024), Geneformer V2 (Theodoris et al., 2023), UCE (Rosen et al., 2023), STATE (Adduri et al., 2025), scPRINT (Kalfon et al., 2025), CellPLM (Wen et al., 2024) and TranscriptFormer (Pearce et al., 2025). Each is used as a fixed encoder: the published weights turn a cell into a representation, and the frozen classifier of Section 2 does the rest, exactly as for PCA. Nothing is fine-tuned. The models were placed with item difficulties held fixed, so no simple model’s ability moved. Table 4 gives their abilities on every curated panel next to the three best simple models; Table 8 in the appendix gives plain accuracy.

On every panel the foundation models land inside the range of the simple pool, between the random floor and the top of the ladder. On Hao PBMC, TranscriptFormer (3.44) sits level with scVI-30D (3.46) and Geneformer (3.09) just above the simple ladder, while STATE, scGPT and UCE (2.32 to 2.56) sit just below PCA-128D (2.88) and CellPLM and scPRINT (1.12, 0.99) two units below it. On the immune atlas, Geneformer, scGPT, TranscriptFormer and UCE (2.11 to 2.55) are above every simple model (2.00 to 2.17) under both splits. On the adult heart panel, where every method is near 0.6, the foundation models are level with PCA. On the lung atlas and on Parse PBMC, the strongest foundation models match the simple ladder and the rest fall below it. In plain accuracy, the best foundation model is within two points of the best simple model on every panel except the heart and lung-study panels, where it trails by one to four points.

The foundation models also behave differently from the simple pool in one way that the scale makes visible. Their outfit values are 1.0 to 3.8, against 0.2 to 0.8 for every simple model. A foundation model therefore passes some cells that its ability says it should fail and fails some that it should pass; its answers follow the difficulty order of the scale less closely. This can mean knowledge beyond the training panel, or noise in the representation, and the pooled scale and the per-item records allow the two to be told apart in later work. One caution applies to all seven. They were pretrained on large public corpora that include the CELLxGENE atlases from which several panels are drawn. We record this exposure for every pretrained model in a contamination register, which is published next to its ability.

5.5 Items that no longer tell models apart

Figure 5 gives the fraction of items that are already uninformative for the evaluated models. An item is too easy at threshold t when every model other than the floors passes it with probability at least t , and too hard when every one fails it with probability at least t . At $t = 0.75$, 0.3% of cell items and 0.1% of conditions are too easy, and 24% and 47% are too hard. At $t = 0.95$ the too-hard fractions are 12.5% and 25%, which are the items no model passes. A model one unit above the best current one would move 45% of the cell items and 9% of the conditions into the too-easy class at $t = 0.95$. This number sets the pace at which new hard items must be added, as Section 6 explains.

The pool has about 30 models per part. The pooled scales carry a spread between datasets of 1.9 units across annotation panels and 3.2 across perturbation screens. The drugs in sciPlex3 and McFarland lack chemical structures in their source files and the cytokines in Parse lack any descriptor, so those screens run in the unseen-

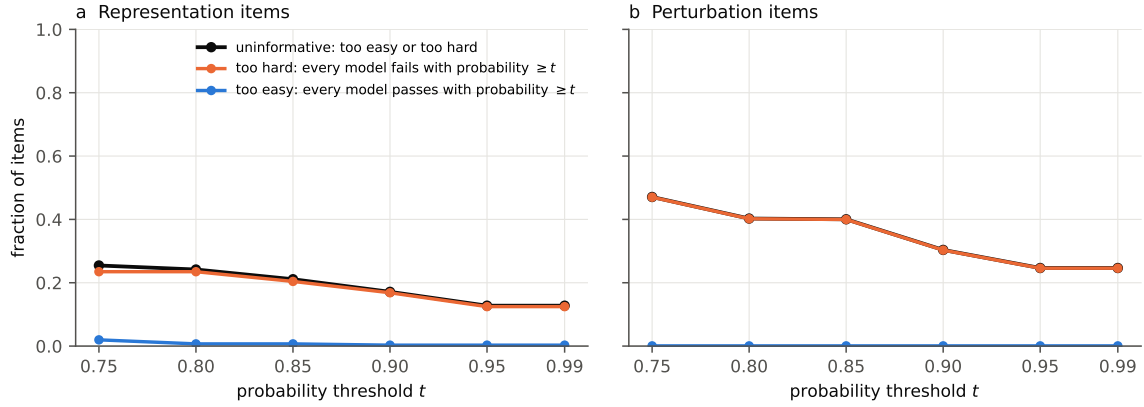


Figure 5 Fraction of items that are uninformative for the evaluated models, as a function of the probability threshold t . An item is too easy when every model other than the floors passes it with probability at least t , and too hard when every one fails it with probability at least t ; items passed by none or by all models count in the respective class. (a) Part A items. (b) Part B items.

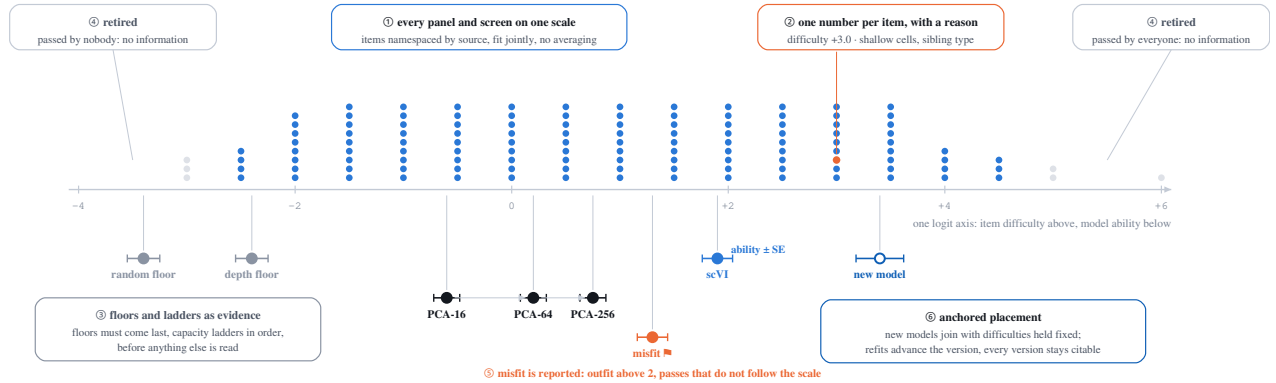


Figure 6 What the scale holds. Items from every panel and screen sit above one axis by difficulty, and models sit below it by ability, each with an error bar. Floors must come out last and capacity ladders in order before anything else is read. Items passed by nobody or by everybody carry no information and are retired. A model whose passes do not follow the order of the scale is flagged as misfit. A new model is anchored with difficulties held fixed, and every version of the scale is kept.

combination split and, for Parse, as a floor in the unseen-perturbation split.

6 Keeping the Benchmark Alive

A benchmark built this way does not have to be thrown away when it gets stale. It is maintained in a loop with four steps. Figure 6 shows what the scale holds and where each step acts.

Anchor. A new model is placed on the current scale with item difficulties held fixed, as in Section 4.3. Its ability, standard error, pass rate and outfit go on the board, tagged with the scale version it was placed on. Published numbers do not move.

Recalibrate. When enough new models have answered, all abilities and difficulties are refitted together and the version number advances. Every version is kept, so any result can be cited with the version it was measured on. The simple pool is refitted every time, so the bottom of the scale is checked at every version: the floors must stay last and the ladders in order. If a floor climbs, labels are leaking into the test.

Retire. Items that every model passes, or that every model fails with high probability, carry no information about the current pool. They leave the active bank but stay in the archive, where they can return when a model arrives that handles them differently. [Section 5.5](#) measures how many items are already in this state and how many would join them for a model one unit stronger than the best current one.

Acquire. New panels and screens are added where the scale is thin. For Part A that means items above the current top of the ladder that are not label noise, for example annotations of cell state rather than cell type. For Part B it means screens with chemical descriptors, so that predicting a new drug becomes a learnable task, and a mapping from cytokines to genes, so that new cytokines can be described at all.

Two rules support the loop. Every model keeps the same name across every run and version, so that its history can be followed. And every pretrained model declares, for each data source, whether that source was in its pretraining data, in any fine-tuning data, or neither. Most entries for current foundation models will not be clean, because the public atlases they were trained on overlap with the atlases we draw from. That is expected. The register is published next to the ability and is what makes the ability interpretable. The current scales, every item’s difficulty, every model’s per-item record and the register are published on the benchmark’s website, which is regenerated from the run outputs after every refit.

7 Limits

The scale rests on one assumption: that one number per model and one number per item are enough to describe who passes what. The outfit statistic measures where this fails, and the foundation models show that it fails more for them than for the simple pool. A richer model with more than one ability per model is a natural extension. The pass rules are choices. For Part A the frozen classifier is one simple linear read-out; a representation that is excellent for a different read-out may look ordinary here. For Part B the mean shift reference is deliberately strict, and a model that reproduces the shared part of a response well but nothing beyond it will fail most items. The scBaseCount panels carry label noise that no model can resolve, which we report as the fraction passed by nobody rather than hide. And the two parts of the benchmark are fitted separately, so an ability in one says nothing about the other.

8 Conclusion

We have described a benchmark for virtual cell models in which every test case has a measured difficulty and every model a measured ability, both on one scale with error bars. The bank holds about 110 million cells across fourteen annotation panels and fifteen perturbation screens. The scale is calibrated with a pool of simple methods that cannot have memorized the test data, and its floors and ladders come out in order. Seven pretrained foundation models have been placed on it without moving it; on the curated panels they fall inside the range of the simple pool, above it on some panels and below on others. Because the benchmark records which items each model passes, it can say which items have stopped being useful, how fast new ones must be written, and what makes an item hard. Those records are what let the benchmark change with the field instead of being replaced by it.

References

- Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingxin Dong, Chiara Ricci-Tam, Kelly Carpenter, Shiye Li, Hani Goodarzi, Dave P. Burke, Patrick D. Hsu, and Silvana Konermann. 2025. Predicting cellular responses to perturbation across diverse contexts with State. *bioRxiv*. Preprint, doi:10.1101/2025.06.26.661135.
- Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. 2025. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*.
- [authors to fill]. 2025. Systema: assessing systematic variation in perturbation-response prediction. *Nature Biotechnology*.
- Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. 2024. scgpt: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21:1470–1480.

- C. Domínguez Conde, C. Xu, L. B. Jarvis, et al. 2022. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376:eabl5197.
- Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, et al. 2021. Integrated analysis of multimodal single-cell data. *Cell*, 184:3573–3587.
- Jérémie Kalfon, Jules Samaran, Gabriel Peyré, and Laura Cantini. 2025. scPRINT: pre-training on 50 million cells allows robust gene network predictions. *Nature Communications*, 16:3607.
- Monika Litviňuková, Carlos Talavera-López, Henrike Maatz, et al. 2020. Cells of the adult human heart. *Nature*, 588:466–472.
- Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. 2018. Deep generative modeling for single-cell transcriptomics. *Nature Methods*.
- Parse Biosciences. 2026. 10 million PBMC cytokine atlas: 12 donors, 90 cytokines. Dataset, CC BY-NC 4.0. [fill URL].
- James D. Pearce, Sara E. Simon, Jonathan S. Courtney, Yun Veronica Kim, Hannah E. Lee, Ramon Chavez, Heidi ElAbd, Patrick R. Walker, Sanghyun Lim, et al. 2025. A cross-species generative cell atlas across 1.5 billion years of evolution: the TranscriptFormer single-cell model. *bioRxiv*. Preprint, doi:10.1101/2025.04.25.650731.
- Stefan Peidli et al. 2024. scperturb: harmonized single-cell perturbation data. *Nature Methods*.
- David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*.
- Yanay Rosen, Yusuf Roohani, Ayush Agarwal, Leon Samotoran, Tabula Sapiens Consortium, Stephen R. Quake, and Jure Leskovec. 2023. Universal cell embeddings: a foundation model for cell biology. *bioRxiv*. Preprint, doi:10.1101/2023.11.28.568918.
- Lisa Sikkema, Ciro Ramírez-Suástegui, Daniel C. Strobl, et al. 2023. An integrated cell atlas of the lung in health and disease. *Nature Medicine*, 29:1563–1577.
- The Tabula Sapiens Consortium. 2026. Tabula sapiens 2.0: a comprehensive transcriptomic atlas of human cell types. *Cell*. BioRxiv 2024.
- Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Mantineo, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and Patrick T. Ellinor. 2023. Transfer learning enables predictions in network biology. *Nature*, 618:616–624.
- Hongzhi Wen, Wenzhuo Tang, Xinnan Dai, Jiayuan Ding, Wei Jin, Yuying Xie, and Jiliang Tang. 2024. CellPLM: pre-training of cell language model beyond single cells. In *International Conference on Learning Representations*.
- Yan Wu et al. 2025. Perturbbench: benchmarking machine learning models for cellular perturbation analysis. In *NeurIPS Datasets and Benchmarks*.
- Nicholas D. Youngblut, Christopher Carpenter, Jaanak Prashar, et al. 2025. scbasecount: an AI agent-curated, uniformly processed, and continually expanding single cell data repository. *bioRxiv*.
- Jesse Zhang et al. 2025. Tahoe-100m: a giga-scale single-cell perturbation atlas for context-dependent gene function and cellular modeling. *bioRxiv*.

Appendix

A Ability tables

Table 5 *Part A models on the pooled cell-level scale of all fourteen panels, 424,478 items: ability and plain accuracy, which is the fraction of test cells labelled correctly.*

Model	Ability	Accuracy
Nyström-256D (kernel width = 0.5)	0.72	0.618
scVI-30D (tissue-corrected)	0.72	0.618
scVI-30D	0.71	0.617
PCA-128D	0.69	0.615
PCA-256D	0.67	0.612
Nyström-64D (kernel width = 0.5)	0.67	0.612
ICA-64D	0.67	0.612
PCA-64D	0.66	0.612
Nyström-256D (kernel width = 2)	0.62	0.607
PCA-32D	0.53	0.597
Nyström-64D (kernel width = 2)	0.42	0.585
NMF-64D	0.41	0.584
scVI-10D (tissue-corrected)	0.37	0.579
PCA-16D	0.11	0.550
ICA-16D	0.11	0.550
Random projection-128D (seed 0)	-0.07	0.529
Random projection-128D (seed 1)	-0.09	0.527
GMM-100 memberships	-0.18	0.516
NMF-16D	-0.37	0.492
PCA-8D	-0.67	0.453
GMM-30 memberships	-0.68	0.453
Random projection-32D (seed 0)	-1.29	0.372
Random projection-32D (seed 1)	-1.31	0.369
PCA-4D	-1.50	0.345
GMM-10 memberships	-1.88	0.295
Random projection-8D (seed 0)	-2.70	0.196
Random projection-8D (seed 1)	-2.70	0.196
Depth only (2D)	-3.14	0.151
Random-32D (seed 1)	-3.83	0.095
Random-32D (seed 0)	-3.83	0.094

B Implementation details

C The Tabula Sapiens panel by tissue

Table 6 *Part A models on the Tabula Sapiens v2 panel alone, 20,026 items, the panel on which the pool was first calibrated.*

Model	Ability	Accuracy
PCA-128D	0.98	0.654
PCA-256D	0.94	0.650
Nyström-256D (kernel width = 0.5)	0.85	0.640
ICA-64D	0.81	0.636
PCA-64D	0.81	0.635
Nyström-256D (kernel width = 2)	0.80	0.634
Nyström-64D (kernel width = 0.5)	0.76	0.630
scVI-30D	0.74	0.627
PCA-32D	0.55	0.606
NMF-64D	0.49	0.599
scVI-30D (tissue-corrected)	0.47	0.596
Random projection-128D (seed 0)	0.35	0.582
Random projection-128D (seed 1)	0.34	0.581
Nyström-64D (kernel width = 2)	0.28	0.574
scVI-10D (tissue-corrected)	0.10	0.552
ICA-16D	-0.11	0.526
PCA-16D	-0.11	0.526
GMM-100 memberships	-0.57	0.464
NMF-16D	-0.60	0.461
Random projection-32D (seed 1)	-0.75	0.440
Random projection-32D (seed 0)	-0.85	0.426
GMM-30 memberships	-1.08	0.394
PCA-8D	-1.34	0.359
PCA-4D	-2.14	0.251
GMM-10 memberships	-2.33	0.228
Random projection-8D (seed 0)	-2.47	0.211
Random projection-8D (seed 1)	-2.57	0.199
Depth only (2D)	-2.97	0.156
Random-32D (seed 1)	-4.14	0.066
Random-32D (seed 0)	-4.14	0.066

Table 7 Part B models on the pooled scale of all fifteen screens, 88,164 conditions. Low-rank completion ran only on multi-context screens and, in the unseen-perturbation setting, can only fall back to the context mean for a perturbation it has never seen.

Model	Ability
KNN (K=20)	-1.10
KNN (K=5)	-1.19
PCA regression-200D	-1.34
Gaussian process (kernel width = 0.5)	-1.38
Gaussian process (kernel width = 2)	-1.42
PCA regression-100D	-1.52
KNN (K=1)	-1.54
Ridge (d=50, $\lambda=100$)	-1.58
PCA regression-50D	-1.58
KNN (K=20), 30% train	-1.62
Ridge (d=50, $\lambda=100$), 30% train	-1.63
Ridge (d=50, $\lambda=1$)	-1.65
PCA regression-20D	-1.66
Ridge (d=10, $\lambda=100$)	-1.67
Ridge (d=10, $\lambda=1$)	-1.71
PCA regression-10D	-1.71
PCA regression-2D	-1.72
PCA regression-5D	-1.73
Ridge (d=50, $\lambda=100$), 10% train	-1.76
KNN (K=20), 10% train	-2.00
Random shift (seed 0)	-2.93
Random shift (seed 1)	-2.93
Low-rank completion, rank 10	-2.97
Low-rank completion, rank 5	-2.97
Low-rank completion, rank 2	-2.97
No change	-3.88
Mean shift + on-target change	-5.27

Table 8 Plain accuracy of the pretrained foundation models and the best simple models on the curated Part A panels: the fraction of test cells whose type the frozen classifier names correctly. Columns as in Table 4.

Model	Hao PBMC	Parse PBMC	Heart	HLCA d.	HLCA s.	Immune d.	Immune t.	Tabula
TranscriptFormer (sapiens)	0.885	–	0.622	0.809	0.763	0.808	0.800	0.6
Geneformer V2-316M	0.862	0.887	0.616	0.821	0.759	0.828	0.817	0.6
scGPT (whole-human)	0.815	0.856	0.622	0.820	0.809	0.814	0.795	0.6
STATE SE-600M	0.827	0.853	0.528	0.774	0.797	0.774	0.762	0.6
UCE (4 layer)	0.806	0.856	0.600	0.802	0.776	0.802	0.781	0.6
CellPLM 85M	0.682	0.820	0.534	0.739	0.761	0.700	0.704	0.6
scPRINT large v1	0.667	0.773	0.464	0.695	0.676	0.633	0.614	0.5
scVI-30D	0.884	0.888	0.601	0.811	0.834	0.763	0.771	0.6
Nystrom-256D (kernel width = 0.5)	0.855	0.892	0.611	0.814	0.812	0.782	0.773	0.6
PCA-128D	0.848	0.888	0.603	0.820	0.806	0.781	0.765	0.6
Random-32D (seed 0)	0.058	0.058	0.028	0.042	0.051	0.061	0.048	0.0

Table 9 Part A panels in full. Held out is the unit whose members form the test items and N held out counts them: donors, public samples, studies or tissues. Contexts are the tissues or studies represented among the items; strata are context, cell type and donor combinations.

Panel	Held out	N held out	Contexts	Cell types	Train cells	Items	Strata
Tabula Sapiens v2	donors	2	22	102	146,800	20,026	433
Hao 2021 PBMC	donors	3	3	24	42,441	7,999	171
Adult heart	donors	5	4	59	81,001	14,396	315
HLCA core	donors	11	11	52	134,615	11,389	247
HLCA core	studies	3	3	51	85,476	32,221	724
Cross-tissue immune atlas	donors	6	16	41	66,684	20,548	474
Cross-tissue immune atlas	tissues	5	5	38	59,565	23,038	521
Parse 10M PBMC	donors	5	2	18	45,948	16,811	176
scBaseCount, lung	samples	70	1	25	145,543	64,635	883
scBaseCount, skin	samples	35	1	15	68,112	28,812	346
scBaseCount, bone marrow	samples	62	1	20	104,777	41,262	513
scBaseCount, liver	samples	29	1	19	51,432	17,559	227
scBaseCount, blood	samples	80	1	14	129,019	46,625	578
scBaseCount, 12 tissues	tissues	5	5	22	180,127	79,157	979
Total				299	1,341,540	424,478	6,587

Table 10 Tabula Sapiens v2 panel by tissue. Donors and cell types count the whole panel; training cells are the classifier’s training set; items are the scored test cells and strata their tissue, cell type and donor combinations. Tissues are ordered by item count. Because the split is by donor, six tissues whose donors all fall on the training side contribute training cells but no items, and Testis, with a single donor on the test side, contributes items but no training cells.

Tissue	Donors	Cell types	Training cells	Items	Strata
Small Intestine	5	31	5,684	1,800	40
Spleen	6	21	10,604	1,326	28
Fat	6	20	5,897	1,234	26
Stomach	3	18	1,252	1,232	26
Large Intestine	5	25	4,785	1,228	26
Tongue	4	20	3,040	1,226	26
Lung	4	28	10,472	1,220	25
Thymus	5	25	9,098	1,219	29
Bladder	6	21	8,776	1,140	25
Lymph Node	6	19	9,119	1,039	22
Heart	5	15	2,618	1,026	22
Muscle	5	22	5,394	999	21
Bone Marrow	6	20	7,737	916	21
Vasculature	4	20	5,223	844	18
Salivary Gland	4	23	6,501	679	15
Trachea	5	20	4,086	621	15
Prostate	3	12	2,771	555	12
Blood	9	18	11,510	504	11
Mammary	2	12	1,837	498	10
Ear	2	9	298	300	6
Ovary	3	9	2,268	270	6
Testis	1	8	0	150	3
Eye	4	28	7,045	0	0
Kidney	1	10	1,371	0	0
Liver	4	20	7,673	0	0
Pancreas	2	14	2,594	0	0
Skin	4	14	4,954	0	0
Uterus	2	17	4,193	0	0
Total	23	159	146,800	20,026	433