Toward a Framework for Evaluating the Clinical Relevance of Cancer Cell Lines at Single-Cell Resolution

Toward a Framework for Evaluating the Clinical Relevance of Cancer Cell Lines at Single-Cell Resolution

Jiatong Li

A Thesis Submitted to the Faculty of

The Harvard Medical School

in Partial Fulfillment of the Requirements

for the Degree of Master of Medical Sciences in Biomedical Informatics

Harvard University

Boston, Massachusetts.

May, 2026

Thesis Advisor: Dr. Srivatsan Raghavan, Dr. Peter Winter Jiatong Li

Toward a Framework for Evaluating the Clinical Relevance of
Cancer Cell Lines at Single-Cell Resolution

Abstract

Human cancer cell lines (CCLs) play a critical role in propagating our understanding of cancer biology and enabling preclinical testing of novel anti-cancer compounds. Previous studies have shown how myriad factors (e.g., media formulation, genetic and transcriptomic evolution) can compromise how faithfully CCLs represent *in vivo* phenotypes. Prior computational strategies have compared CCLs and TCGA bulk RNA-seq expression profiles, but the relationship between CCLs and patient tumors at single-cell resolution remains poorly characterized. To this end, we established a robust benchmarking framework that compares the effectiveness of diverse embedding approaches at relating CCLs to patient single-cell RNA-seq (scRNA-seq) profiles. To facilitate robust benchmarking, we designed a suite of control experiments, standardized preprocessing pipelines, and adapted metrics to evaluate mapping performance for each benchmarked method. We benchmarked seven methods: principal component analysis (PCA), canonical correlation analysis (CCA), Gene program, Celligner, Symphony, expiMap, and Geneformer. We observed that most methods excelled at tasks in simpler experiments, but showed reduced accuracy for certain lineages when mapping CCLs to their corresponding lineage in patient samples. We identified CCA as a promising candidate for integrating large-scale cross-system mapping of CCLs and tumor datasets. In sum, we have taken initial steps toward a framework to compare preclinical cancer models and patient tumor samples at single-cell resolution. Ultimately, with further development, this framework will guide the selection of preclinical models for specific clinical contexts.

Table of Contents

List of Figures and Tables

List of Supplementary Figures

List of Supplementary Tables

Acknowledgements

The following people have kindly provided advice, feedback, and support for me and my project to date: Yuzhou Evelyn Tong, Xiang Zhang, Sebastiano Cultrera di Montesano, John Jacob, Sharon Bader, Jacob Smigiel, Andrew Navia, Akshaya Thoutam, Ava Amini, Lorin Crawford, Peter Winter, Srivatsan Raghavan.

I would also like to thank my parents for supporting me during my time here, and friends who were kind enough to listen to me explaining my research and offered support or advice.

Special thanks to Vester Cambridge for providing a stable source of coffee and food.

Chapter 1: Introduction

Cancer Cell Lines (CCL) are immortalized patient tumor cells used as *in vitro* models for studying primary tumors and testing anti-cancer therapeutics. To date, large-scale efforts to catalog and characterize CCLs like the Cancer Cell Line Encyclopedia (CCLE) and Genomics of Drug Sensitivity in Cancer (GDSC) enable preclinical model comparison[1,2]. Indeed, several factors compromise their fidelity to patient tumors, limiting the translatability of findings from these models. First, patient cancer cells are surrounded by complex tumor microenvironments (TME) of non-malignant cells and growth factors. The crosstalk between malignant cells and their environment was shown to enhance tumor progression and reduce therapeutic efficacy[3]. However, culture media often lack the metabolites and non-malignant cells that would otherwise shape dynamic cancer cell activity *in vivo*. Second, patient tumors often show a high degree of intratumor heterogeneity in genetic, transcriptomic, and other phenotypic features[4,5]. In contrast, the lack of TME and strong selective pressure under *in vitro* conditions are generally thought to limit phenotypic heterogeneity in CCLs compared to their tumor counterparts[6,7]. Finally, direct comparison between CCLs and patient tumors have revealed discrepancies in both genetic and phenotypic attributes. Previous studies identified a suite of cellular pathways selectively up- and down-regulated in CCLs across lineages, and to a lesser extent, somatic mutation discrepancies between cancer samples and CCLs[8,9]. As CCLs empower early preclinical drug-screening efforts and primary validation platforms for AI-powered drug discovery, it is critical that we understand if and how well CCLs represent their corresponding patient tumor samples[10].

Several works have attempted to relate CCLs to patient tumors. Najgebauer *et al.* leveraged genomics features and cancer functional events as a basis for comparison, enabling the selection of *in vitro* models that recapitulate recurrent genomic subtypes[11,12]. However, genomic similarity does not necessarily capture phenotypic heterogeneity across tumors. In contrast, Warren *et al.* introduced Celligner, which aligns cell lines and tumors using transcriptional profiles. Their approach applied contrastive PCA to remove non-malignant signals from bulk tumor RNA profiles, followed by mutual nearest neighbor (MNN) to mitigate batch effects[13]. However, transcriptional programs central to cancer treatment response and progression, like epithelial-mesenchymal transition (EMT), may be restricted to subsets of tumor cells and therefore obscured in bulk RNA profiles[14].

With the advent and widespread adoption of single-cell RNA sequencing (scRNA-seq), a large repository of cancer single-cell RNA seq data have revealed common patterns of intratumor heterogeneity programs[5]. Notably, Kinker *et al.* profiled a broad panel of CCLE cell lines at single-cell resolution[6], while Tyler *et al.* created the Curated Cancer Cell Atlas (3CA), aggregating over 100 datasets and 2800 patient tumor samples[15]. Despite these advances, no existing study has systematically investigated pan-cancer alignment of cell lines to patient tumors at single-cell resolution.

To address this gap, we turn to the class of single-cell integration methods, many of which hold potential for relating CCLs to patient tumor phenotypes, which we hereafter refer to as *cell line mapping*. These methods can be broadly characterized by whether embedding weights are computed *a priori* or *de novo*. Principal component analysis (PCA) is a commonly used dimensionality reduction method that computes embedding transformation at runtime with no batch correction[16]. Similar to PCA, canonical correlation analysis (CCA) and Symphony also

create *de novo* embedding weights, but with analytical modules that aim to mitigate technical variation between datasets[17,18]. An interesting approach in the *de novo* class is expiMap, which uses an interpretable conditional variational autoencoder (cVAE) to learn a shared latent representation of single-cell data at runtime while conditioning on study-specific covariates to account for batch effects[19].

An alternative approach in weights creation is using precomputed weights. These weights can be the result of large-scale pre-training procedures in single-cell foundation models (scFM), capable of integrating single-cell transcriptomic profiles in shared latent spaces that enable downstream tasks like cell type prediction or perturbation simulation[20]. Geneformer adopts a pre-training procedure using a transformer-based architecture, where genes are ranked and tokenized, enabling the model to implicitly learn gene regulatory network structure[21]. Fine-tuned variants specialized for cancer biology further adapt these representations, making Geneformer a promising approach for creating meaningful mapping CCLs to patient tumors. Another way to precompute embedding weights is by using gene expression programs. In general, gene programs are functional modules composed of dozens of genes identified via matrix decomposition methods like PCA or non-negative matrix factorization (NMF)[22]. Using scoring methods like AddModuleScore or AUCell, one is able to quantitatively evaluate the transcriptomic activity of gene programs in cells[16,23,24]. Thus, a gene program embedding can be manually constructed with a set of meaningful gene programs for cell line mapping. Notably, neither Geneformer nor gene programs conduct active batch correction, even though batch integration is commonly an evaluation objective of foundation models like Geneformer.

Finally, Celligner bypassed batch effects using MNN, effectively aligning bulk expression profiles of cell lines and patient tumors[13]. A similar approach could be adapted in

single-cell level alignment, where the Celligner approach creates a *de novo* embedding after correcting for mutual nearest neighbors. Though both CCA and Celligner do not explicitly consider batch variables at a per-cell basis like Symphony or expiMap, they are dataset-aware and can potentially produce embeddings that are free of technical variations.

Overall, these scRNA-seq mapping methods can all be characterized by an embedding approach. By building a framework for shared embedding, we establish a comparable benchmarking system for any potential methods or approach that is capable of cell line mapping. Although alternative approaches that don't involve embedding spaces exist (such as the comparison of DE genes between samples and cell lines), they usually involve significant user supervision and discretion to focus on specific pathways. In contrast, integrating CCL and patient tumor profiles with scRNA-seq integration methods enable a more holistic, data-driven assessment of cellular states by considering many features simultaneously, rather than focusing on a limited set of predefined attributes.

Although these embedding methods all hold potential for relating CCL and patient tumors at single-cell level, no existing study has robustly compared their performance on this objective, and such an objective can be challenging. Unlike normal cells with clear biological functions, malignant cells do not have concrete labels that describe their biological state. Though recurrent states that underlie intratumor heterogeneity can be defined across patient tumors, such effort remains incomplete and their definitions are often not as clear as that of normal-functioning cells[5]. This complicates both the task and design of metrics for evaluating the efficacy of cell line mapping in a single-cell integration framework. More specifically, while a T-cell from one patient is expected to resemble a T-cell from another, malignant cells from different patients can exhibit substantial phenotypic variation[25]. As such, separation between

clusters of malignant cells could be biologically meaningful and not due to technical variation, making it difficult to distinguish meaningful biological divergence from mapping error. Moreover, cell culture models can shift their phenotypes due to culture conditions[26]. In fact, Warren et al. observed that some cell lines from CCLE did not map to patient samples of the same lineage and instead formed clusters that displayed an undifferentiated state[13]. It is unclear to what extent the same effect will be observed in single-cell mapping, where such separation could be exacerbated by technical variation and the high-sparsity nature of scRNA-seq data. Consequently, such an effort at the single-cell level demands for well-controlled evaluation schemes and standardized metrics for malignant cell mapping. This highlights the need for a systematic single-cell benchmarking framework that can shed light into biologically meaningful separation, while accounting for technical artifacts.

To address this, we take the first steps toward establishing a single-cell framework for relating cancer cell lines with patient tumor transcriptomic profiles. Specifically, we design and validate controlled evaluation settings where ground truth relationships are known, and introduce quantitative metrics to assess mapping performance under these conditions. We will present the results in three aims. In Aim 1, we will discuss the general considerations in establishing the framework used in this study, including how to evaluate neighborhoods in the single-cell embedding space and the adaptation of appropriate metrics. In Aim 2, we will design and present results for control experiments. In Aim 3, we will present results for a large-scale cell line mapping experiment.

Chapter 2: Framework Considerations, Dataset Preprocessing, and Implementation of Benchmarked Methods (Aim 1)

Framework Overview

We developed a standardized framework to facilitate consistent evaluation across experiments and methods. Each experiment involved two sets of scRNA-seq datasets: reference and query. We filtered each dataset through a fixed quality control pipeline (see *Datasets Curation*), but the derivation of query and reference varied between experiments. In each experiment, we further reduced the dimensionality of the query-reference pair using either shared genes or shared highly variable genes, depending on the specific embedding method used (see *Unified Feature Selection*). Finally, each embedding method applies a series of transformations to the expression matrices of both query and reference, generating a shared embedding space that integrates the query dataset with the reference dataset (see *Implementation of Benchmarked Methods).* This action of co-embedding a query with the reference will be referred to as "mapping query to reference" in this report. We then computed the pairwise Euclidean distances between each query cell and each reference cell, and obtained a ranked list of nearest reference neighbors for each query cell, where a lower rank is closer in proximity. To obtain a cell line–level ranking of reference neighbors, we devised a procedure that we termed "min-rank pooling". Briefly, we first computed nearest neighbors for each individual cell within a given cell line. For each reference cell, we then recorded its closest observed position across all cells from that cell line. These ranks were used to generate a single list of reference cells ordered from closest to farthest relative to the cell line. Ties were resolved randomly. The ranked list of reference cells will be generally referred to as the "neighborhood" in this report. This procedure resembles a modified, biased version of single linkage-based hierarchical clustering where instead of looking

at the smallest distance globally, we only consider the special subset of distances that connect query and reference cells. We will discuss alternative approaches in the *Discussion* section[27].

To ensure consistent language usage, the general phrase "CCL X mapped to Y" means that when the CCL X was embedded into a reference using some embedding method, reference cells with attribute Y were ranked among the highest collective neighbors of CCL X, or synonymously, the neighborhood of CCL X is enriched with reference cells with attribute Y. We will quantify this notion with metrics discussed in the *Metrics* subsection below.

Datasets Curation

In this section, we will discuss the preprocessing steps for the datasets used in the experiments in aim 2 and 3. For each dataset, we applied the same quality control criteria. Specifically, we filtered for genes that were expressed in at least 50 cells, and filtered for cells that had at least 500 genes expressed, with at least 1000 total read counts per cell, and at most 10 percent mitochondrial gene expression.

For CCL scRNA-seq datasets, we used two datasets. First, we used an unpublished dataset which we will refer to as Tong2023. This scRNA-seq library was sequenced using Seq-Well[28]. It contains a set of 9 CCLs in the pancreatic lineage: ASPC1, BXPC3, CFPAC1, KP4, PANC1, PANC0403, PATU8988S, PATU8988T, and PK1. After QC, we kept all 2232 cells and 11900 genes (Supplementary Table 1). Next, we downloaded the counts and metadata from Kinker *et al*., which contained scRNA-seq profiling data from a subset of CCLE cell lines[1,6]. We will refer to this dataset as Kinker2020. After QC, we kept 48552 of the original 53513 cells and 19793 of the original 30314 genes (Supplementary Table 1).

For patient tumor scRNA-seq datasets, we first downloaded all available datasets on 3CA (accessed January 2026)[15]. Next, we selected for all samples sequenced using 10x Genomics technology, and then manually filtered out studies that had ambiguous cell type definition or lacked cell type labels. We identified 10 lineages that were shared between Kinker2020 and the 3CA atlas. To account for minor naming discrepancies, we manually reannotated "central_nervous_system" lineage in Kinker2020 as "brain", "large_intestine" as "colorectal", and combined "liver" and "biliary_tract" in Kinker2020 as "liver_biliary". For each 3CA study in these 10 lineages, we applied the same quality control filter, and recorded the before and after cell and gene count in Supplementary Table 2. Finally, since each 3CA study contains existing cell type annotations, we filtered for only malignant cells to enable direct comparison with cell lines. To facilitate faster runtime while preserving key biological signals, we also performed down-sampling for the 3CA dataset. Specifically, for each patient sample with more than 200 malignant cells sequenced, we randomly chose 200 cells; for those with less than 200 malignant cells, we kept samples that had at least 50 cells. In the end, we concatenated all patient samples from 3CA into one dataset, which contained 101599 cells and shared 6223 genes. For reference, in Supplementary Table 1 we also included the cell and gene count for the resulting 3CA dataset if quality control wasn't applied. We will refer to this filtered, down-sampled and concatenated dataset as "3CA" in future chapters.

Unified Feature Selection

One major axis that separates the benchmarked methods is whether embedding weights are computed on-the-spot (*de novo* methods) or pre-computed (*a priori* methods). Geneformer and gene programs both use pre-computed weights, while other methods like PCA and expiMap

compute the weights at time of inference. We reasoned that the quality of the embedding spaces built by *de novo* methods critically depend on the accessible features of the pair of input datasets, and we should therefore ensure that all *de novo* methods have access to the same highly variable gene space. On the other hand, limiting the features accessible to *a priori* methods might artificially handicap its performance, defeating the purpose of using precomputed weights. Thus, we devised a shared feature selection pipeline that precedes embedding generation. Doing so fixes a comparable initial data condition across all benchmarked methods, ensuring a fair and unbiased comparison. Specifically, for a priori methods, we simply kept the subset of the genes shared between query and reference datasets (Supplementary Table 3). For de novo methods, we keep their shared top 2000 highly variable genes using a custom procedure. Given a pair of query and reference datasets, we first ranked each gene according to their normalized variance within the dataset, then took the top 2000 genes with the highest average rank across the two datasets (Supplementary Figure 1). To compute the normalized variance, we set `flavor="seurat_v3"` in *Scanpy*'s `highly_variable_genes()` function, which operates on raw counts. We chose this approach over intersection or union of HVG across the two datasets because it enables us to control the exact number of shared HVGs, and it allows us to preserve genes that are uniquely variable in one dataset but not the other, while controlling for their overall variability across the two datasets.

Implementation of Benchmarked Methods

PCA

PCA is a common dimensionality reduction technique that distills biological signals and enables downstream analyses like clustering or trajectory analysis[16]. PCA integrates query and

reference by projecting the cells onto axes that maximizes explained variance found in the shared HVG space of query and reference. We first applied log-normalization to both the query and reference datasets using natural log base and with target sum set to 10,000. We then concatenated these two datasets and scaled the normalized counts with zero center set to true and max value set to 10. These steps were performed using the *Scanpy* package. Similar preprocessing steps in the following methods used the same arguments unless otherwise specified. We then applied PCA to the scaled normalized counts of the merged dataset on the shared highly variable genes and used the top 50 principal components as the embedding.

Symphony

Symphony is a parametric single-cell reference mapping method that maps query datasets using soft reference cluster assignment[18]. Whether it can successfully embed single cell CCL samples into a tumor reference atlas remains unknown. We first log-normalized both query and reference datasets. We then scaled the normalized counts of the reference dataset, and took the top 50 principal components with zero center set to false. We then projected the query dataset into the reference-derived PCA space using Symphony's `map_embedding()` function. Symphony assigns a soft probability that each query cell belongs to an automatically inferred reference cluster and applies a cluster-specific correction weighted by the cell's membership in that cluster. We then concatenated the PCA embeddings of the reference cells and the Symphony-assigned PCA embeddings of the query cells as our embedding matrix. In our control experiments in aim 2, "SymphonyR" will be used to indicate this implementation. In aim 3, however, due to technical difficulties we weren't able to include Symphony with its full functionality. Thus, the "Symphony" method in figures in aim 3 indicates a functionally simplified version which simply

transforms the query cells with the PC loadings learned from the reference cells, projecting the query into the reference PC space.

Celligner

This framework developed by Warren *et al*. was used to assess the alignment of bulk-level CCL expression profiles to patient tumor expression profiles. It is unknown whether the Celligner framework can map CCL samples into a patient tumor reference at single-cell resolution[13]. The original framework included cPCA (contrastive PCA) as the first step to account for immune cell contamination of TCGA expression profiles[29]. We have effectively performed this step by filtering for malignant cells from the reference dataset, and therefore continue to the second step, which corrects for CCL-patient separation using mutual nearest neighbor (MNN)[30]. Briefly, MNN identifies pairs of query and reference cells that are mutually listed as each other's top-k nearest neighbors.

In Celligner, the authors used differing radius *k* between the two datasets. They argued that a larger *k* is used for the much larger tumor samples. They set *k=50* for around 10000 tumor samples, and *k=5* for around 1000 cell lines. In this work, we initially used a python implementation of MNN from *mnnpy*. To remain faithful to Warren *et al*.'s implementation, however, we added a second implementation of Celligner that leveraged the authors' modified implementation of the `fastMNN()` function from the *scran* package in R. We will refer to the Python implementation as "CellignerPY" and the R implementation as "CellignerR". More importantly, in CellignerPY, we fixed the *k* for both datasets to be 20. For CellignerR, we set *k* to be 0.5% of the dataset sample size, emulating Warren *et al.*'s specification for k. We also log-normalized both datasets before inputting them into MNN functions. In either case, the function

returns corrected log-normalized expression values of both query and reference, which we output as the embedding matrix.

Gene programs

This method repurposes gene program scores directly as a low-dimensional embedding space. There are two main ways of evaluating gene program activity for single-cell transcriptomes: AddModuleScore and AUCell. AddModuleScore measures the relative expression of the user-specified gene set against randomly selected background genes across the dataset using binning, thereby measuring the upregulation or downregulation of the program in a cell relative to other cells[23]. We reasoned that this approach might result in fluctuating scores depending on the samples included at the time of analysis. Therefore, for this project, we implemented AUCell instead. For each cell, AUCell first ranks the genes by their expression value, then constructs a recovery curve indicating the number of genes recovered from the input gene program. The resulting score quantifies the relative expression magnitude of the gene program of interest compared to the rest of the genes[24]. This approach provides stable scoring regardless of background dataset, and is also consistent between raw and log-normalized counts.

The Molecular Signatures Database (MSigDB) is a resource cataloging annotated genes sets used for gene set enrichment analysis and for scoring[31,32]. From there, we curated three collections of gene programs. The first set, which we will refer to as "Geneprogram", contains gene sets from Human Protein Atlas (HPA), MSigDB Hallmark gene sets, MSigDB-C4-3CA collection, and MSigDB-C8 collection (cell type signature gene sets)[5,31,33]. The second set, which we will refer to as "Geneprogram_c43ca", only includes the MSigDB-C4-3CA collection, and

the third set, which we will refer to as "Geneprogram_c8", only includes the MSigDB-C8 collection.

In our implementation, we first log-normalized the query and reference datasets, and then we concatenated the two datasets and calculated gene program scores for each cell over the gene programs in each curation. We skipped any programs if less than 50% of the genes were absent in the input datasets. Finally, we extracted the scores for the programs that passed the threshold as the embedding for each cell.

expiMap

This method performs reference mapping by examining the query's activity in both known and *de novo* gene programs learned from the reference dataset during training[19]. We first kept the gene programs among the Reactome gene sets provided by Lotfollahi *et al*. containing at least 12 genes considered as highly variable in query and reference. Next, we initialized an expiMap model with ALPHA=0.7, set the experiment-dependent batch variable as condition_key, and trained the model on reference raw expression counts. We then trained the model on query raw counts. Finally, we extracted the cVAE embedding using the `get_latent()` function. We used all hyperparameters as provided in the tutorial except for the above-mentioned changes for ALPHA and "condition_key". The seed used for the training reference model was set to 15.

Geneformer

This method performs reference mapping by projecting query and reference into an embedding space using weights learned through self-supervised learning. Importantly, we

decided to use a published variant of Geneformer generated from continual learning with cancer cell transcriptomes, which increases the chances that the weights reflect cancer-specific gene expression and regulatory patterns[21,34]. Briefly, we merged the query and reference datasets and tokenized them using the provided tokenization function from the *geneformer* package. We then extracted the second to last layer of embedding using weights saved in the "Geneformer-V2-104M-CLcancer" model card from Huggingface.

CCA

Canonical correlation analysis (CCA) was developed to extrapolate highly correlating components between two datasets that share experimental units[35]. Zhang *et al*. adapted CCA for the mapping of bulk and single-cell RNA sequencing experiments[17]. Here, we further adapt their approach to establish co-varying components that reflect shared biology between CCLs and patient tumors. Initially, we attempted to replicate their approach by deriving the shared components on the 2000 HVG shared between query and reference (we used identity mapping experiment for testing this implementation). However, all covariates derived using this approach had near 1 correlation. This could be related to CCA overfitting in a low-observation, high-dimensional configuration. Crucially, in Zhang *et al*.'s adaptation, samples were treated as features and genes as observations. They didn't encounter overfitting because they reported much higher shared HVG than samples. In our case, however, we limited HVG to 2000, and have much higher sample (cell) numbers.

To circumvent this problem, we first log-normalized, scaled, centered, and took the top 50 principal components for query and reference separately. Next, we performed CCA on the gene loading matrix for PCA, calculating 20 linear projections of query and reference PC space

such that the highly variable genes are maximally correlated. We then applied this transformation, learned separately for query and reference, to the respective PC coordinates. As a result, the 20-dimensional space captures shared variation found in both datasets. Since the CCA approach generates the least number of dimensions, we also investigated a variant of this approach by taking 40 covariates instead of 20 to test for whether the low-dimensionality conferred an unfair advantage.

To summarize the implementations, we have created Figure 1 to indicate the major parameter and steps for each method group.

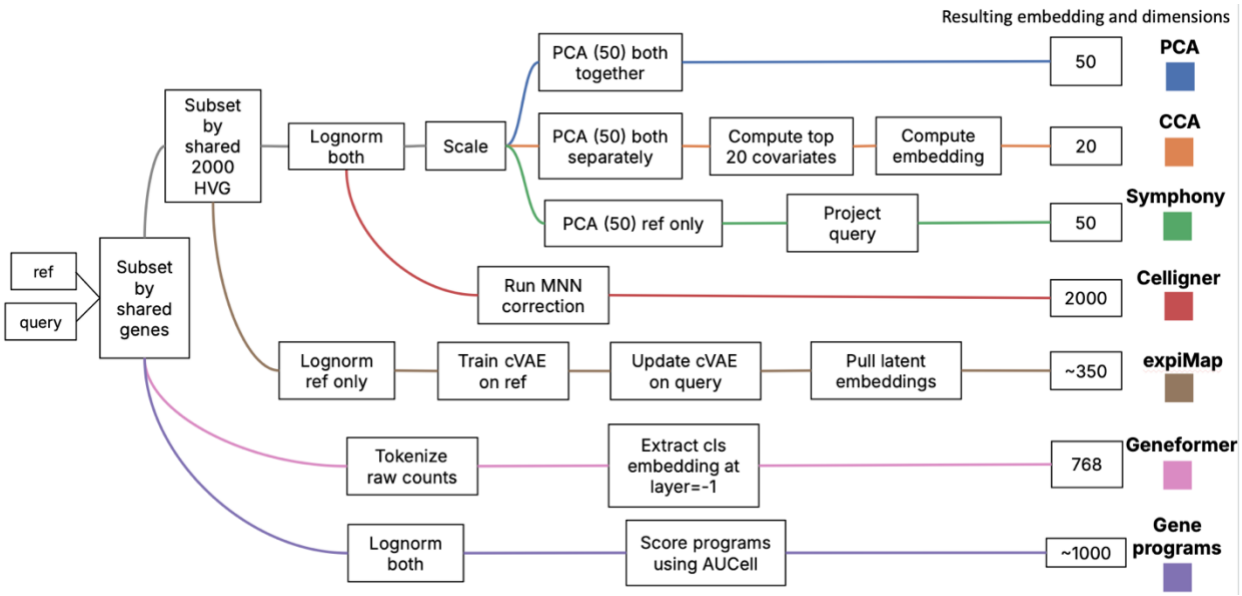


Figure 1. Flowchart overview of implemented methods.

Metrics

VCA

We next sought to define metrics that would allow us to assess mapping performance. In our experiments, we focused on quantifying two aspects of mapping: batch effect removal and neighborhood coherence. To quantify batch effect removal, we implemented variance component

analysis (VCA), which estimates the contribution of biological and batch variables to total embedding variance by modeling them as random effects using a linear mixed effect model in

Equation 1

$$Y = X\beta + Zu + e, \quad u \sim N(0, \tau^2)$$

the form of Equation 1[36]. Here, Y denotes the coordinate assigned to a cell in an embedding dimension, X the observed values for each of the β fixed effects parameters, Z the design matrix of our variables of interest, u the vector of unknown random effect variables parameterized by its variance contribution τ^2, and e the identically and independently distributed Gaussian random error. For each experiment, we have one batch variable and one biological variable. A method successfully removes batch effect from its embedding space if the variance contribution from a batch variable diminishes or reduces, compared to PCA as a baseline. After mapping query cell lines to the patient reference data, we arrive at a cell-by-embedding dimension matrix. For each embedding dimension, we first calculated its variance, and then fitted the two variables using the `lmer()` function from the R *lme4* package[37,38]. Finally, to obtain each variable's total contribution to variance, we calculated its weighted average contribution across dimensions, where weights are calculated as a fraction of each dimension over total variance in the embedding space.

AUROC

To quantify the coherence of cell line neighborhoods, we focused on quantifying whether expected reference cells are more proximal to query than irrelevant cells. Area under the ROC (AUROC) curve summarizes the relationship between positive rate (TPR) and false positive rate (FPR) across all possible decision thresholds and is commonly used to quantify classification

performance. In our context, each mapped query produces a list of reference neighbors ranked by proximity to our query. We defined positives as whether a target cell is labeled with the expected value, and all remaining reference cells as negatives. As we traversed the ranked neighbors list from most to least proximal, we computed the TPR and FPR at each threshold, yielding a ROC curve for each mapped query. The resulting AUROC thus quantifies how closely the relevant reference cells are placed in the query neighborhood in comparison to irrelevant reference cells, independent of query neighborhood size.

Accuracy@*k*

In Celligner, Warren *et al*. evaluated their mapping using label transfer. Briefly, for each cell line, they consider the lineage label of the 25 nearest patient tumor neighbors, and designated the majority label as the "predicted lineage" of the cell line. They then compared the predicted lineage to the annotated lineage of the cell line in CCLE to decide whether a cell line is clustered with their respective disease type. At single cell resolution, the same number of neighbors usually sample much less of biological variation than in bulk. Thus, for analyses where *k* is needed to define a neighborhood, we decided to empirically investigate the effect of neighborhood radius on *k*-dependent metrics: Accuracy@*k* and entropy (see *Shannon Entropy*). We used *k=25* as a baseline, *k=1000* as an upper bound by the smallest lineage class size in the reference, and *k=100* as a middle ground. In each case, identical to Warren *et al*., we obtained the labels of *k* nearest neighbors and evaluated the majority label for correct lineage prediction of a cell line.

Shannon Entropy

Equation 2

Maximum Shannon Entropy for K classes: In a uniform distribution over K classes, entropy is maximized because each class $i \in K$ is equally likely, i.e., $p_i = \frac{1}{K}$. Its Shannon entropy can be derived as follows:

$$H = -\sum_{i=1}^{K} \frac{1}{K} \log \frac{1}{K} \tag{1}$$

$$= -K \cdot \frac{1}{K} \log \frac{1}{K} \tag{2}$$

$$= -\log \frac{1}{K} \tag{3}$$

$$= \log K \tag{4}$$

Furthermore, we reasoned that the diversity of a neighborhood could be of interest when investigating cell line phenotype and mapping quality. To quantify the label diversity, we adapted and calculated scaled Shannon entropy for each neighborhood. Briefly, for each label list, we calculated the frequency of each label as the empirical probability distribution, and calculated Shannon entropy on the derived distribution. Next, we divided the entropy over the natural log of the number of all possible labels, which corresponds to a uniform distribution (Equation 2). This way, we scaled the metric to between 0 and 1 by accounting for the highest possible entropy any list can obtain. We used entropy implementations from the *SciPy* package, which uses natural log by default.

We developed the above metrics to enhance our capability to probe into the complex neighborhoods constructed in cell line-patient mapping experiment. For the control experiments, we will use a subset of these metrics where appropriate.

Chapter 3: Control Benchmarking Experiments Using CCLs (Aim 2)

Control Experiment Setup

In this section, we present control experiments using CCLs only: identity mapping and batch correction. In identity mapping, we divided Tong2023 into query and reference datasets. Specifically, for each of the 10 technical replicates of the experiment, we constructed the query dataset by selecting half of the cells from each CCL at random, and collected the remaining cells as the reference dataset. For this experiment, we expected the query cells to map to reference cells from the same cell line. Though there is no batch variable, in figures we will refer to the splitting of query and reference as "mapping".

For the batch correction experiment, we use the entirety of Tong2023 as the query, and the pancreatic subset of Kinker2020 (Kinker2020Pancreas) as reference. Three cell lines - ASPC1, PANC1, and PATU8988S - were shared between Tong2023 and Kinker2020Pancreas. Tong2023 was sequenced using Seq-Well, while Kinker2020Pancreas used 10X Genomics as their sequencing platform, and thus the batch variable in this experiment is "platform". Assuming minimal genetic drift, contamination, and other factors that would otherwise add biological difference between the two batches for the shared cell lines, we expect that a method capable of mitigating for batch effects would map ASPC1 from Tong2023 to ASPC1 from Kinker2020Pancreas, and so on.

To generate UMAPs of each experiment, we first calculated the top 15 neighbors of the cells in their respective embedding space, and ran UMAP with minimum distance set to 0.3, both from the Scanpy package. We will use VCA to evaluate batch correction and AUROC to evaluate neighborhood coherence in the control experiments.

Control Experiment Results

To test whether these methods could successfully map cell lines to cell lines without batch effect, we conducted mapping experiment using pancreatic cells from Tong2023 as the query and randomly assigned equal size query and reference batches. After obtaining the embedding for each experiment replicate, we calculated AUROC by testing for the proximity of the query cell line label to the reference. The AUROC values were averaged across the technical replicates. We then calculated 95% confidence intervals across the 9 query cell lines, and

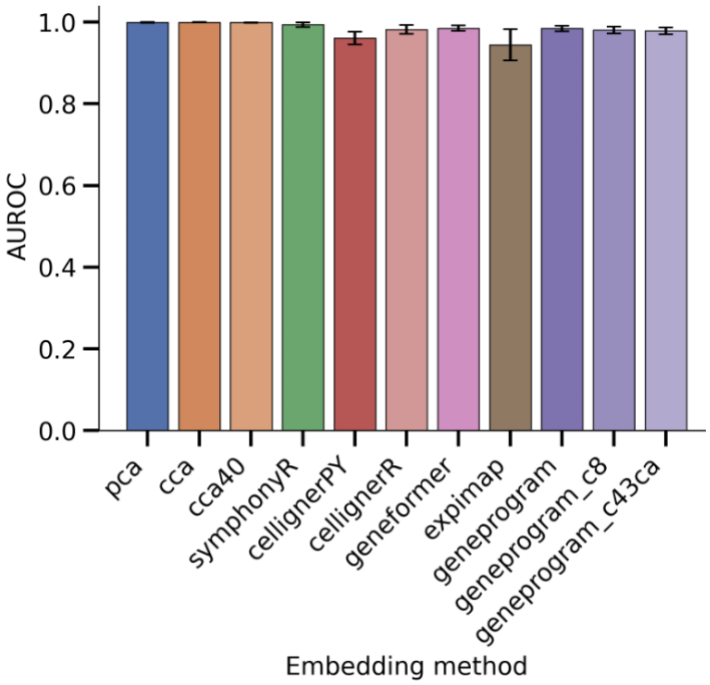


Figure 2. Mean AUROC in identity mapping experiment.
Error bars indicate 95% confidence interval (n=9); stars indicate significantly different mean compared to every other method at the least stringent alpha. P-value derived with Wilcoxon signed-rank test and adjusted using Benjamini-Hochberg correction.

repeated this procedure for each method evaluated. In the identity mapping experiment, all methods had an AUROC score of close to 1 (Figure 2). This was expected, as identity mapping was the simplest experiment. To further visualize the embedding space, we created UMAPs of one of the replicates (Supplementary Figure 2). In addition to SymphonyR and expiMap, we also

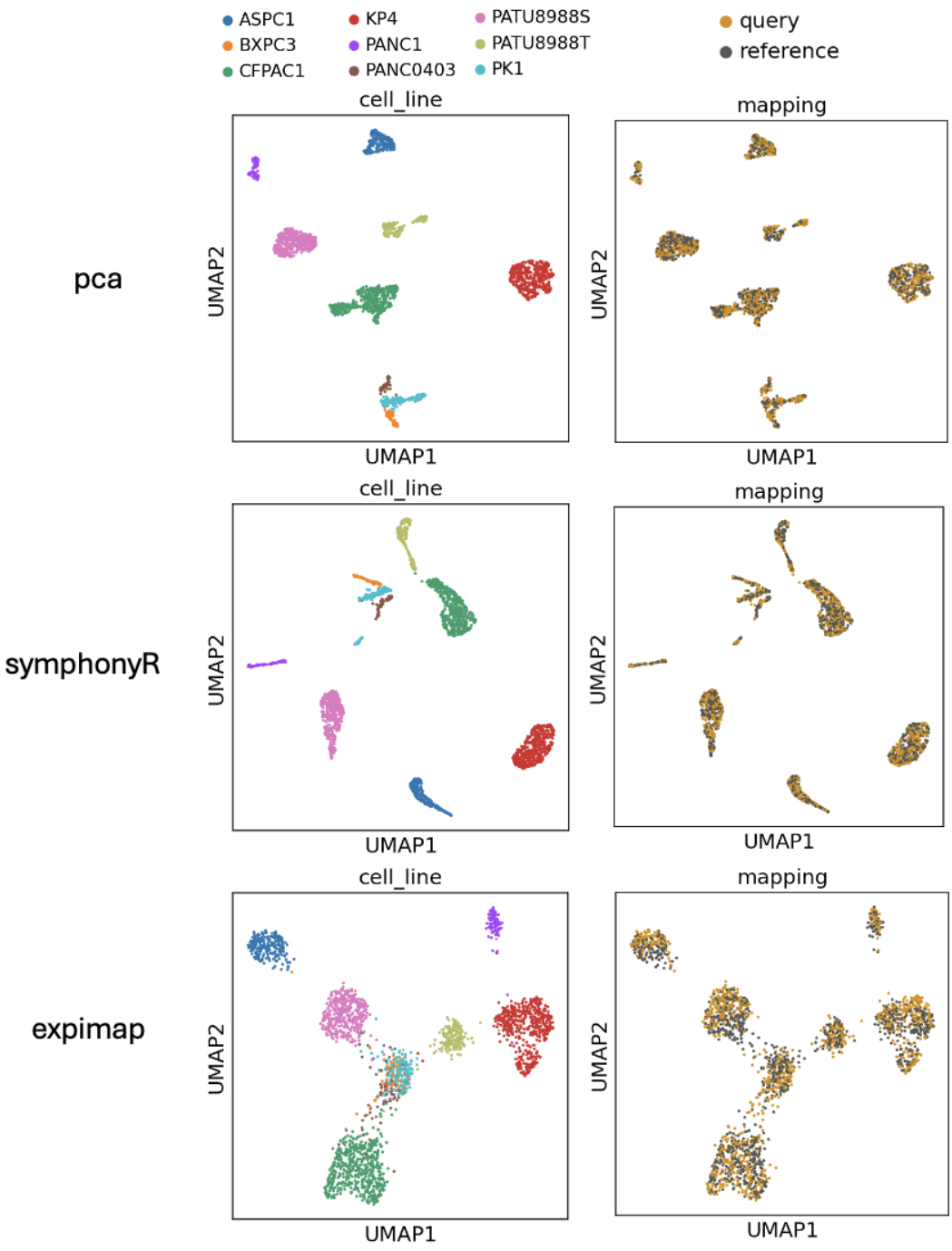


Figure 3. UMAPs of select evaluated embedding methods in identity mapping experiment. UMAPs of PCA, SymphonyR, and expiMap embedding spaces. Left column is colored by cell line in Tong2023, and right column row is colored by query/reference designation

showed PCA as a baseline for reference (Figure 3). In PCA, query cells overlapped with reference cells from the same cell line, and each cell line was separate from each other. The same pattern is observed in SymphonyR and expiMap, although some of the cell line clusters were

much more scattered in expiMap. Two possibilities could explain this observation. ExpiMap works by learning an embedding space from the reference and then modifying the space using the newly projected query. First, expiMap was able to separate the cell lines more cleanly in reference, but the space is distorted due to overly sensitive training on query. This can be tested by extracting the UMAP of the reference space prior to query training and visually comparing the two UMAP spaces. Second, expiMap struggles to cleanly separate the cell lines even in reference-only embedding space. This might be due to suboptimal hyperparameter choices such as ALPHA, which regularizes the number of gene programs that are activated in the embedding space.

We also performed VCA for this experiment as a negative control test. Since both batches were arbitrarily created from the same batch, we expected minimal to no batch variable contribution to the overall embedding space. Indeed, variance contribution from batch remained at or very closely to 0 for all embedding methods (Supplementary Figure 3).

Next, to test whether these methods could successfully map cell lines to cell lines *with* batch effect, we also conducted control mapping experiment using pancreatic cells from Tong2023 as the query and Kinker2020 as reference. In VCA, batch variable contributed 0.22 to PCA embeddings variations (Figure 4). Comparing to this baseline, CCA, CCA40, SymphonyR,

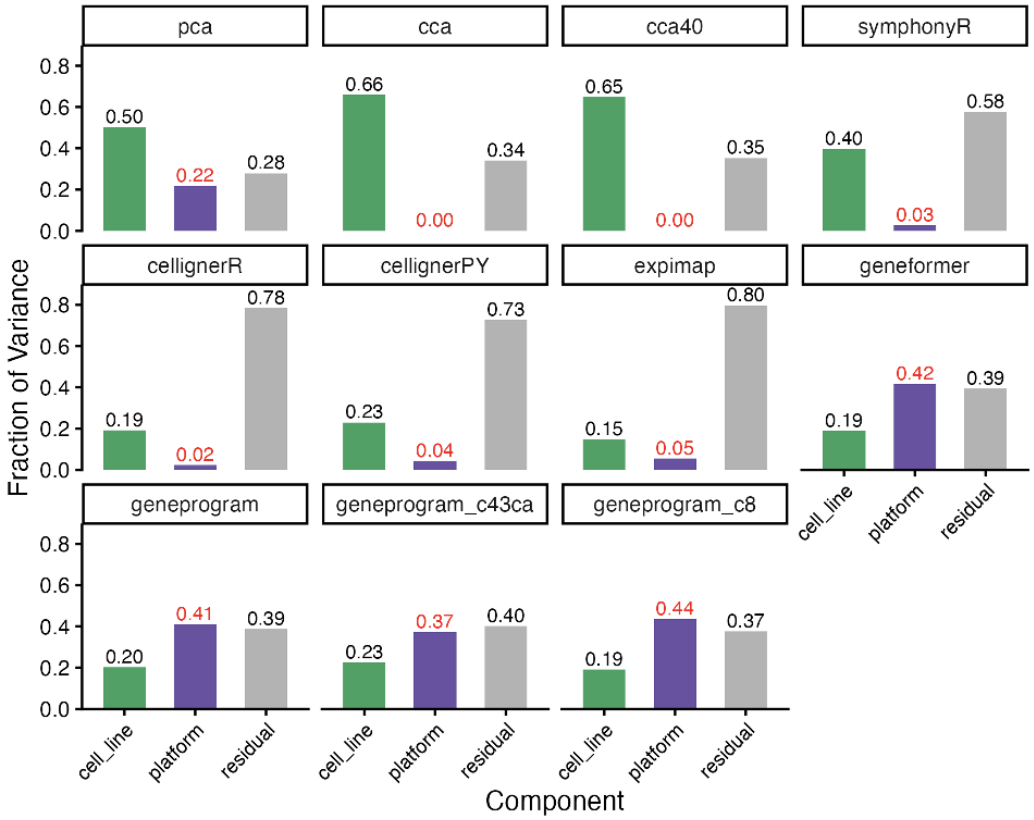


Figure 4. VCA analysis bar plots in batch correction experiment.
Proportion of variation explained by cell line, platform, or residual in the evaluated embeddings. Batch variable contribution is highlighted in red.

CellignerR, CellignerPY and expiMap saw a significantly reduced or diminished variance contributed by batch variable, while the rest of the methods had similar level if not more contribution from batch. Among these methods, only CCA, SymphonyR, Celligner methods, and expiMap included mechanisms that actively applied corrections to their query cells. Thus, it's not surprising that in their resulting embedding dimensions, the role of batch variable in determining variation was diminished. On the other hand, Geneformer and gene programs both used embeddings weights obtained *a priori*. Such weights, even if derived using training data

that contained multiple batches, are not guaranteed to correct for batch effects observed here. Moreover, depending on the program used, it's possible that certain dimensions correlated with expression patterns unique to technical batches, which could exacerbate batch effect, and potentially explaining our observation in Figure 4.

We then turned to investigate the AUROC performance of each embedding method. Since we only had 3 cell lines that overlapped between reference and query datasets, the 95% confidence interval bar was calculated using a sample size of 3 (Figure 5). To provide a more precise measure, we included the AUROC value for each cell line across methods in

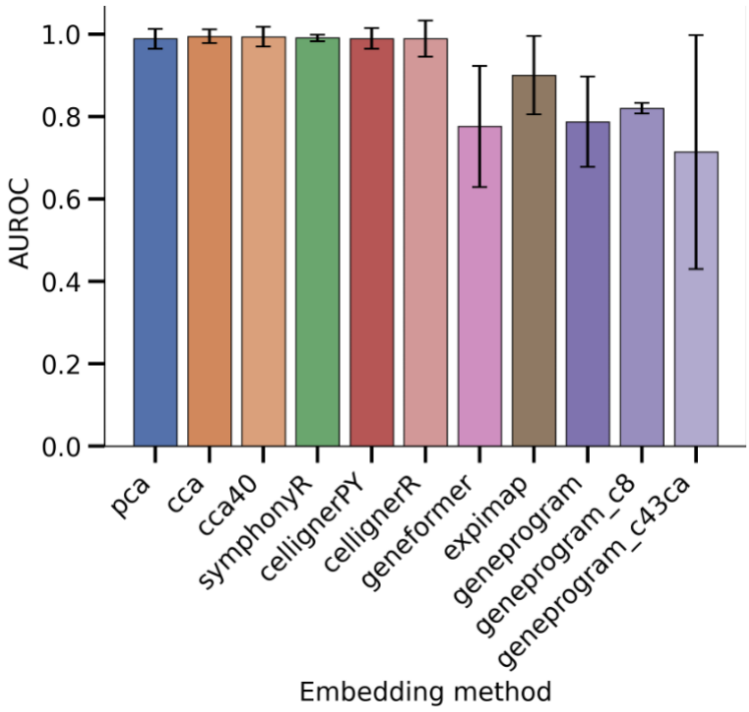


Figure 5. Mean AUROC in batch correction experiment.
Error bars indicate 95% confidence interval (n=3); stars indicate significantly different mean compared to every other method at the least stringent alpha. P-value derived with Wilcoxon signed-rank test and adjusted using Benjamini-Hochberg correction.

Supplementary Table 4. In evaluation, PCA, CCA, CCA40, SymphonyR, and Celligner approaches all had exceptionally high performances, while Geneformer, expiMap and gene programs all had degraded performance that varied across the 3 cell lines (Figure 5). For the *a*

priori methods, combining observation in AUROC with VCA suggests that these methods constructed a cell line neighborhood that was driven by batch similarities instead of the biological variation that defines cell lines (Supplementary Figure 4). This might mean that these methods are not suitable for correcting batch effects in aligning pancreatic cell lines. For expiMap, UMAP reveals that it again might suffered from a fuzzy separation between different cell line entities, contributing to a slightly degraded performance. Interestingly, though PCA didn't actively correct for batch effects, its AUROC performance was relatively high. This might be due to a strong underlying variation contribution from cell line variable, which is evident from the VCA results (Figure 4). In deriving AUROC, we excluded neighbors from the query batch. We also performed this evaluation including the same-batch neighbors (Supplementary Figure 5). As expected, Geneformer and Gene programs performance decreased even further, confirming that cell lines were grouping by batch in those spaces. Surprisingly, PCA's performance remained high in the alternative AUROC evaluation. This suggests that the principal components derived from this pair of datasets were not heavily driven by batch effect (Figure 5).

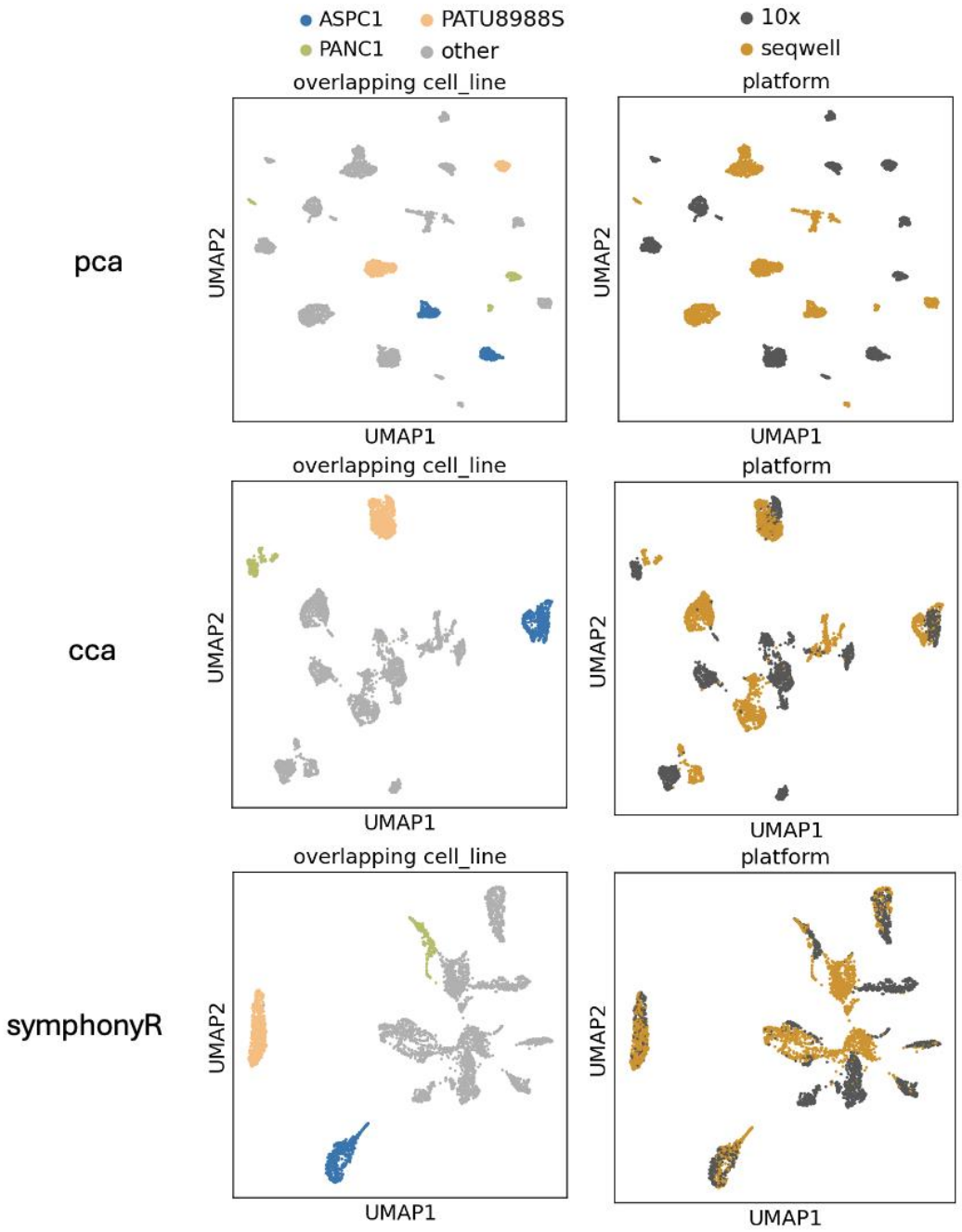


Figure 6. UMAPs of select evaluated embedding methods in identity mapping experiment. UMAPs of PCA, CCA, and SymphonyR embedding spaces. Left column is colored by shared cell line across Kinker2020 and Tong2023, and right column row is colored by sequencing platform.

We also examined the UMAP of PCA as a baseline, and CCA and SymphonyR (Figure 6, Supplementary Figure 5). Both CCA and Symphony embeddings overlapped the shared cell lines across query and reference, although PCA showed some degree of separation between the batches within the shared cell lines.

Based on the control experiments, we began to identify groupings of methods based on their shared performance patterns. First, CCA, Symphony, and Celligner methods consistently scored the highest in AUROC and fully corrected for batch effect where appropriate. CCA and Celligner both co-embed datasets by identifying shared structures between query and reference. In CCA methods, it does so through correlation analysis, while in Celligner, it corrects the embedding of query cells by applying a correction vector that decreases the distance between shared mutual nearest neighbors across the two datasets. Symphony does so through identifying the most appropriate reference cluster that each query cell should map to. However, in cases where query contains cell types not found in reference, this could result in forced, faulty mappings. Next, Geneformer and gene program methods all had low performances that averaged in the 0.6-0.8 range, and their neighborhood construction were biased by batch. Both of these approaches use pre-calculated embedding weights, suggesting that the weights were not effective at mitigating the batch effect observed in the current experiment. ExpiMap was able to correct for most of the batch effects present, but did not score as high as the other methods. Its autoencoder approach for embedding learning might have introduced noise, resulting in a lower resolution for biological variation, which was especially evident through its low cell line variance contribution in batch experiment. During the project, we encountered faulty performance and implementation errors with the python implementation of Symphony. Though we were able to switch to the R implementation for the control experiments (SymphonyR), we were unable to fully finish its results for the aim 3 experiments. We will continue to use Symphony to refer to an altered implementation without soft clustering correction in aim 3. Notably, from our experiment thus far, achieving a high AUROC, indicative of a coherent neighborhood, does not necessarily require a method to fully correct for batch effects. Though

this might be true for the simpler control experiments, this might not hold true for the much more complex CCL-tumor mapping experiment.

Chapter 4: Mapping CCLs to Patient Tumor Samples (Aim 3)

Additional Data Processing

We next examined whether each embedding method could map a query CCL to patient tumor samples of the same lineage. Here, our reference is the 3CA dataset, and the query is Kinker2020, both after the extensive preprocessing described in *Dataset Curation*. To further ensure the quality of the comparison, we checked the cell count distribution across all selected cell lines in Kinker2020 and found outlier cell lines with exceptionally high (>500) or low cell counts (<100, Supplementary Figure 6a,b). By our neighborhood derivation, overly large or small cell count could influence the quality of the neighborhood at the cell line level. As a filter, we set the lower bound to 100 arbitrarily to filter out cell lines with small populations, and upper bound to 580, which was two standard deviations above the average number of cells in each cell line. Doing so further removed 28 cell lines of the original 134 cell lines from the 10 shared lineages from initial preprocessing as described in *Dataset Curation* (Supplementary Figure 6c). To check for lineage class imbalance, we plotted the cell count distribution by lineage for each dataset. Breast lineage was the largest class in the reference, and lung lineage was the largest class in the query (Supplementary Table 5, Supplementary Figure 6d). We measured imbalance by the ratio of tumor to CCL cells in each lineage. We identified colorectal, breast, and prostate as the top 3 most imbalanced lineages, and adjusted the ratio by randomly taking half of the samples originally included in these lineages. The resulting lineage-wise ratio is included in Supplementary Table 5.

CCL-Patient Mapping Results

Here, we evaluated the embedding spaces created by our benchmarked methods in mapping cell lines to their respective lineages in the manually curated patient tumor atlas. For evaluation, we used lineage as the biological variable, and source—whether a cell is taken from cell line or from tumor—as the batch variable. We leveraged AUROC, label transfer accuracy, and entropy to evaluate biological mapping, and VCA to evaluate batch effect correction. Overall, the variance contribution from the batch variable followed a similar pattern observed in the batch correction control experiment, with CCA and Celligner methods strongly removing batch contribution (Figure 7). On average, CCA methods scored the highest in AUROC

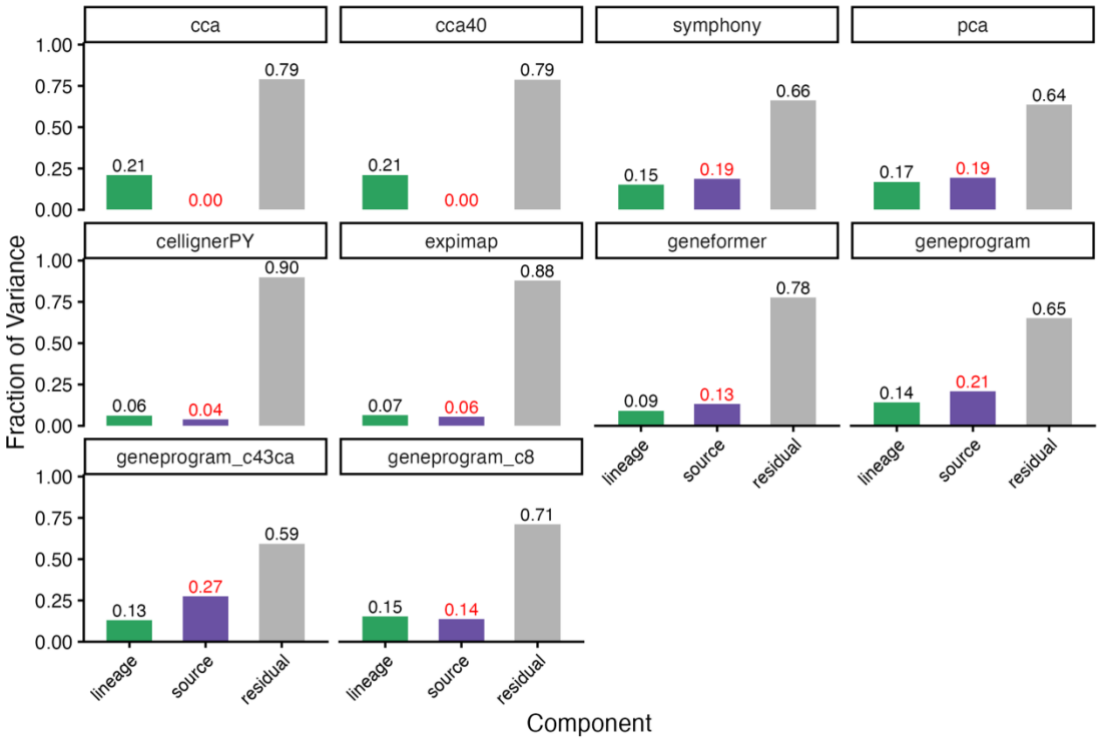


Figure 7: VCA analysis bar plots in CCL-tumor mapping experiment.
Proportion of variation explained by lineage, source, or residual in the evaluated embeddings. Batch variable contribution is highlighted in red.

evaluation at 0.67, followed by PCA and Symphony at 0.59 (Figure 8). This means that across all cell lines evaluated, CCA created an embedding space that most consistently placed the cell line into a patient tumor cell neighborhood with the corresponding lineage label. Comparing the

embedding space UMAPs of CCA to PCA as a control, in agreement with VCA results, CCA

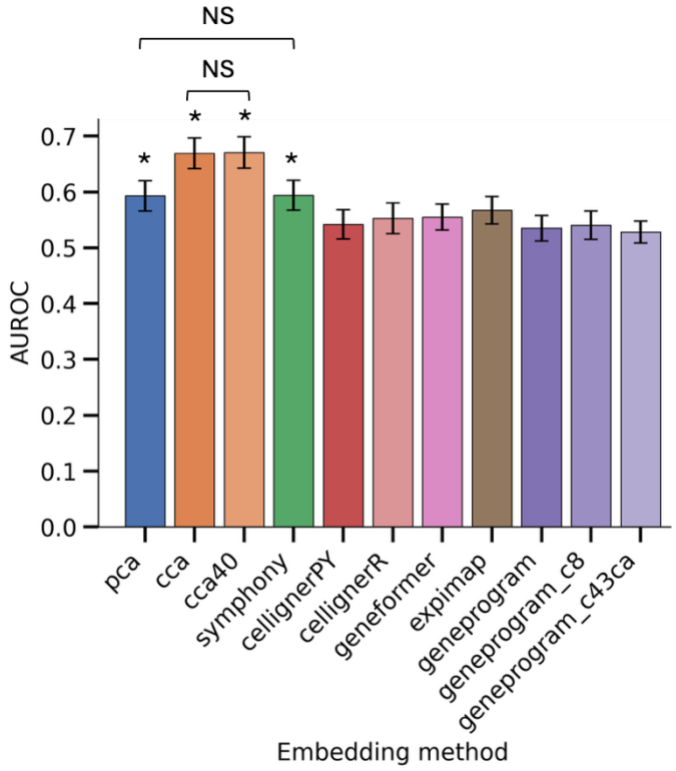


Figure 8. Mean AUROC in CCL-tumor mapping experiment.
Error bars indicate 95% confidence interval (n=106); stars indicate significantly different mean compared to every other method at the least stringent alpha (except for the indicated comparison). P-value derived with two-sided paired T-test and adjusted using Benjamini-Hochberg correction.

showed higher intermixing between the *in vivo* and *in vitro* groups and showed signs of co-clustering of breast and skin lineages (Supplementary Figure 7). This is in contrast with methods like PCA, Gene programs, and Geneformer, where the two batch sources remained separated. Interestingly, although PVCA results suggested that CCA, Celligner, and expiMap all effectively removed batch effects, we did not observe the same degree of batch mixing in UMAP as in the batch correction experiment. In this experiment, the residual variance also became much larger than that in control experiments. This hints at a more complex variance structure that might require more variables to fully analyze. One possibility is that although these methods are corrected for the artificially labeled “source” effect, the main driver of batch separation could be

a variable not fully accounted for, such as the study associated with each dataset, or varying procedure in experimental techniques. To investigate whether this was the case, we ran another VCA that accounted for study and sample parameters (Supplementary Figure 8). As expected, study variables contributed more variance when compared to lineage or batch variables in all embedding spaces. For most methods, the residual variance reduces to below 0.5 when study and sample were accounted for, but in expiMap and CellignerPY (we couldn't finish running that for CellignerR due to compute constraints, but we expect similar result as CellignerPY), residual contribution remained high. This indicates that other unidentified parameters dominate the variance structure in these spaces. However, to further dissect its composition, we will need to identify other variables and perform in-depth, exact analysis of variance, which exceeds the estimation offered by VCA.

To further dissect AUROC score performance of each method, we asked whether there are trends across cell lines. We compared the AUROC score of each cell line individually across methods and used hierarchical clustering to identify cell lines that held similar patterns (Figure

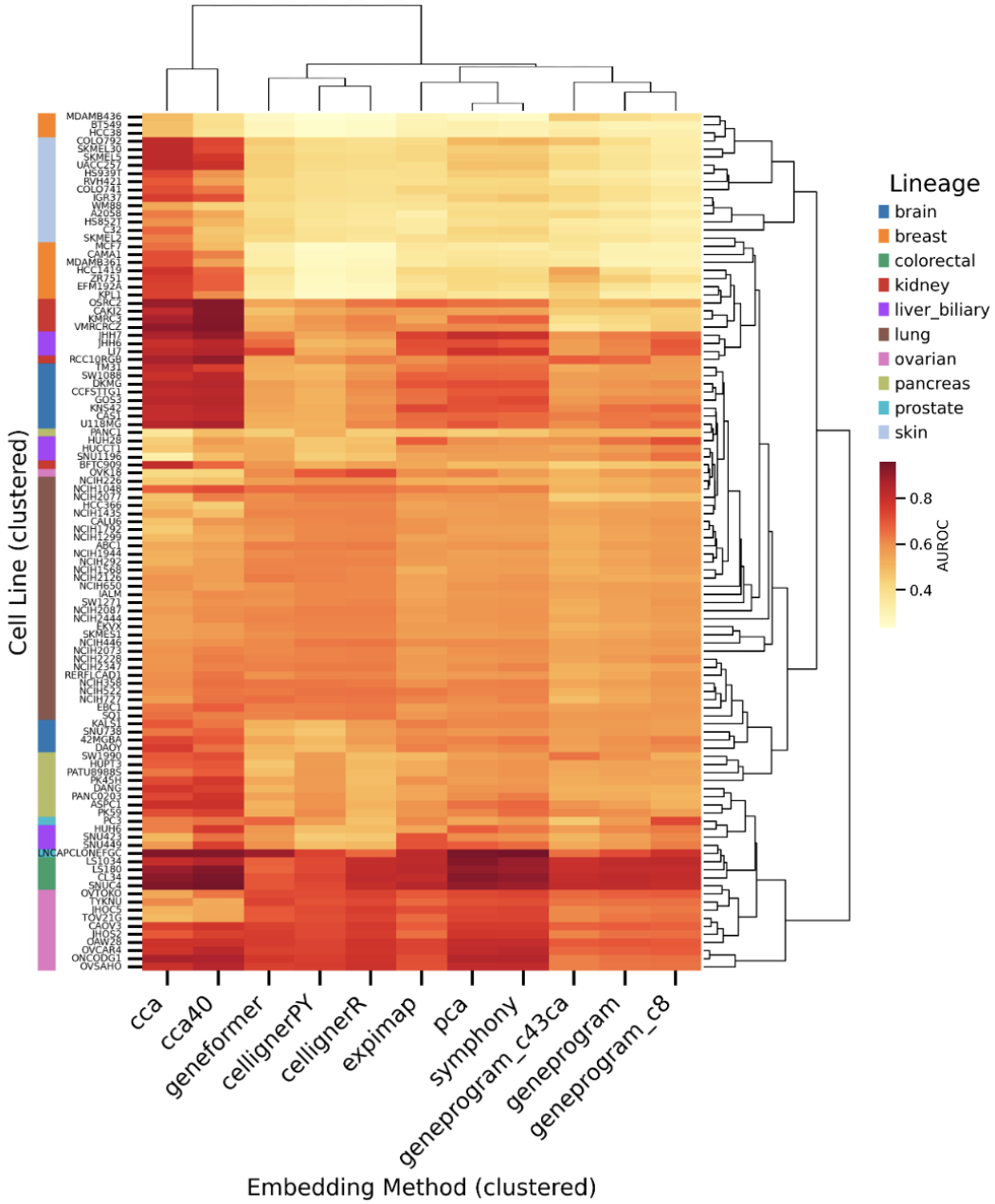


Figure 9. Heatmap of AUROC in CCL-tumor mapping experiment.
Rows and columns ordered by hierarchical clustering. Color strip on the left indicates cell line lineage.

9). The cell lines were divided into a few groups. First, almost all breast and skin cell lines received high AUROC scores only in CCA embedding spaces, suggesting that CCA was

uniquely good at mapping breast and skin cell lines into a coherent neighborhood overall. Next, subset of cell lines from kidney, liver-biliary and brain clustered, showing elevated AUROC scores in CCA methods and moderately high scores in expiMap, PCA, and Symphony. All methods had overall low AUROC score for a collection of cell lines composed mostly of lung lineage, suggesting that it might have been particularly difficult to produce a coherent neighborhood for lung cell lines. Finally, A group of cell lines from ovary, prostate and colorectal collectively shared high AUROC performance in all methods, suggesting that all methods mapped these cell lines coherently. Compared to other methods, CCA methods generate a unique AUROC scoring pattern across cell lines, suggesting that its approach be unique in overall neighborhood construction. Taken together, CCA had the highest overall AUROC score and lowest variance contribution from the batch variable, corresponding to the observed batch mixing in UMAP. This suggests that CCA methods create a batch-corrected embedding space that is coherent when evaluated using AUROC on lineage labels.

Although AUROC quantifies the lineage coherence of cell line neighborhoods, it does so across the entire embedding space. In comparison, the label transfer task focuses only on each cell line's *k*-nearest neighbors. Because of this, we reasoned that these two metrics might produce seemingly contradicting results depending on the specific cell line neighborhood. We

summarized this in Table 1, and demonstrated two cases where these two metrics might disagree.

	High AUROC	Low AUROC
Correct label prediction	Cell line is placed near center of its corresponding tumor lineage	Cell line is placed near *some*, but distant from *most* tumor cells of the same lineage
Incorrect label prediction	Cell line is placed near *most* tumor cells of the same lineage, but the immediate neighbors are incoherent.	Cell line is placed distant from its corresponding tumor lineage

Table 1. Possible cases in discrepancy between AUROC and label transfer prediction.

We selected the breast cancer line MCF7 in Geneformer (low AUROC among correctly predicted lines) and the brain cancer line DKMG from CCA (high AUROC among incorrectly predicted lines). We compared the distribution of true and predicted lineage labels across both the full embedding and in the local neighborhood. Despite CCA concentrating most of DKMG's true lineage within the top 10,000 of ~120,000 neighbors, consistent with its high AUROC, the nearest 100 neighbors used for label transfer were nevertheless enriched for breast rather than brain lineage, leading to misclassification (Figure 10a,b). Conversely, although Geneformer placed most of MCF7's true lineage label to more distant neighborhoods, yielding a low AUROC, its nearest 100 neighbors still contained sufficient breast cancer cells to produce a

correct prediction (Figure 10c,d; the blue line is hidden since true lineage is the same as predicted lineage).

This comparison highlights shortcomings in both metrics. First, AUROC can be more lenient than label transfer accuracy under scenarios like Figure 10a, where the embedding method "cheated" by mapping a cell line to tumor cells from a different lineage while still producing an overall coherent neighborhood list. Under this scenario, the class size of each lineage could also pose a layer of bias in calculation. Although TPR and FPR are normalized by class size, larger classes may still exhibit greater label incoherency under random assignment, as

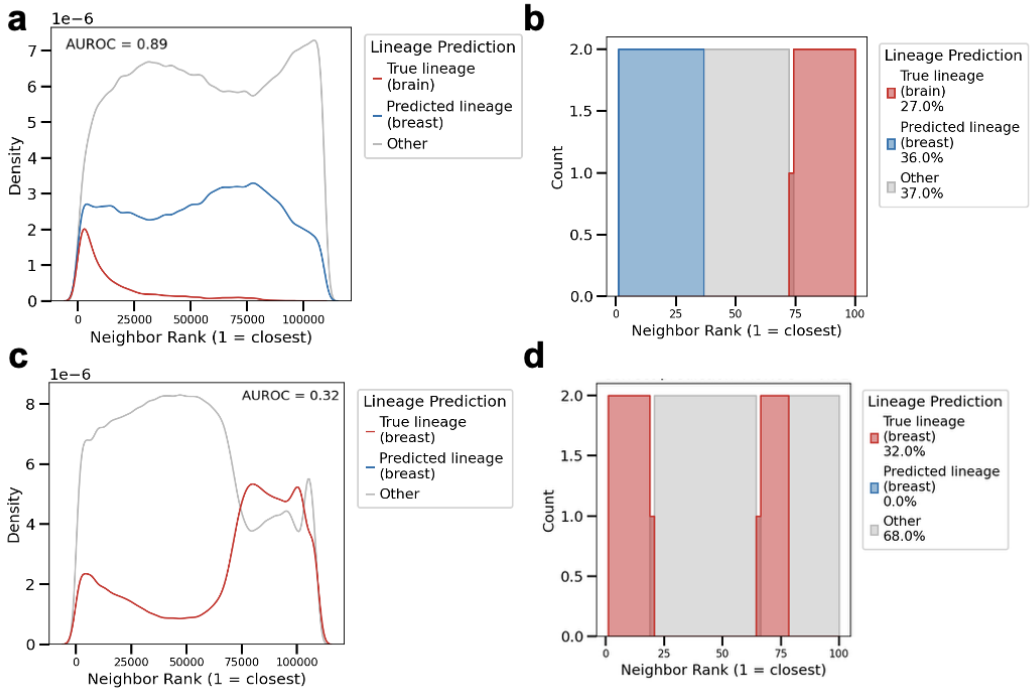


Figure 10. Select edge cases for disagreement between AUROC and label transfer prediction.
a–b. Neighborhood label distributions for DKMG in CCA space, shown as **b.** rank-ordered density and **c.** histogram of lineage labels among the top 100 neighbors.
c–d. Same as **a–b**., but for MCF7 in Geneformer space.

they contain more opportunities for misclassification. On the other hand, in a scenario shown in Figure 10c, neither AUROC nor label transfer accuracy fully reflects the "bimodal" nature of the neighborhood. Furthermore, label transfer accuracy can vary with the choice of k, as different neighborhood sizes could yield different predictions. Warren *et al.*, adopted *k=25* as their

evaluation threshold. However, we reasoned that this choice may not be appropriate in the single-cell setting, where each neighbor represents an individual cell rather than a bulk sample. Although a more systematic approach for choosing *k* would be preferable, here, we compared the overall prediction accuracy across methods at *k=25*, *k=100*, and *k=1000*. We found that most methods reached highest prediction accuracy *at k=100*, and thus decided to use *k=100* as a threshold for all *k*-nearest neighbor analyses hereafter (Supplementary Figure 9).

Overall, CCA40 achieved the highest label transfer accuracy performance at 59.43%, followed by PCA, CCA and Symphony at 54.72%, 54.72%, and 50.94% respectively (Figure 11). The rest of the methods all had accuracy at or below 41%. Almost all methods achieved

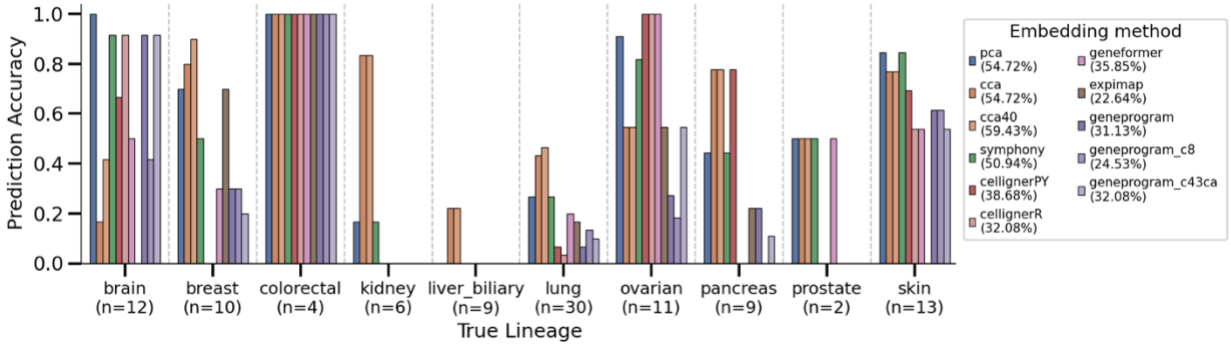


Figure 11. Lineage label transfer accuracy and confusion matrices of selected methods.
Lineage-wise accuracy reflects the proportion of cell lines with predicted lineage matching annotated lineage (n=106).

perfect prediction for colorectal lineage, in agreement with its high AUROC in Figure 9. Many methods had a difficult time predicting labels for kidney, liver-biliary, and lung cell lines except for CCA methods. This also agrees with the AUROC heatmap, which indicated that for a few cell lines in these two lineages, CCA methods had an edge compared to other methods. For the other lineages, each method had varying degrees of performance with no clear trend judging by only correctness. This shows that all methods, regardless of successful batch effect removal, were able to map colorectal cell lines to colorectal tumors, while most methods could not map kidney, lung, liver-biliary cell lines by lineage. This might mean that all embedding spaces

learned features that were crucial to mapping colorectal lines, while mapping features in kidney, lung, and liver-biliary cancers were more difficult to learn. Combining with the observation that CCA most successfully minimized batch variable's contribution to variance, it's possible that correctly mapping most lineages, especially kidney, lung, and liver-biliary, required optimal batch effect correction.

We next investigated whether there were trends in misclassification modes. For each cell line, we compiled its predicted lineage in each embedding space and visualized these in a heatmap (Figure 12). CCA and CCA40 (and CellignerPY to some extent) tended to misclassify a

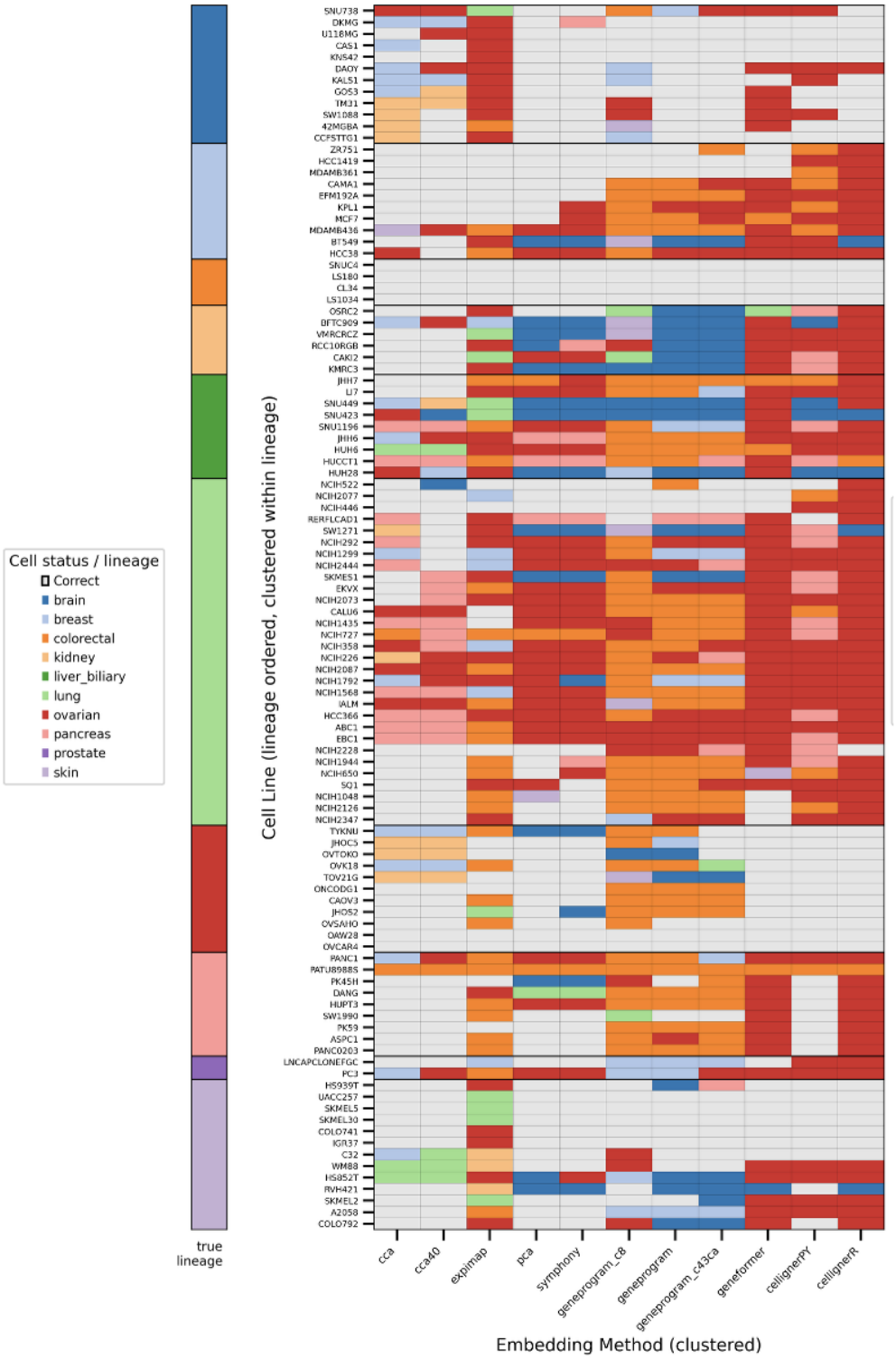


Figure 12. Heatmap of misclassifications in CCL-tumor mapping experiment.
Columns are ordered by hierarchical clustering. Rows are first ordered by lineage, then hierarchical clustering. Greyed-out blocks indicate the given method correctly assigned its lineage based on its nearest 100 neighbors, otherwise the misclassified lineage color is labeled according to legend.

similar set of cell lines to pancreas, while PCA, Symphony, CellignerR and Geneformer tended

to collectively misclassify a large set of lung and a small set of mixed cell lines as ovarian and brain, respectively. Finally, the Gene program methods (and to some degree, expiMap) had a trend of misclassifying them as colorectal lineage. All three gene programs methods might have included programs that were biased in cell line and colorectal lineages, causing them to cluster closely together. For the other methods, ovarian was the majority mode of failure. This means that they mapped the majority of cell lines into ovarian tumor neighborhoods, which might indicate that they all failed to mitigate for an expression pattern that's uniquely found in ovarian tumors that dominated the embedding dimensions construction.

To get a better understanding of why embedding methods have these misclassification patterns and biases, we first characterized the neighborhood using entropy. We found that for most methods, neighborhoods of misclassified cell lines had higher entropy than those correctly predicted (Figure 13). Celligner methods and Geneprogram_c8 did not have significantly

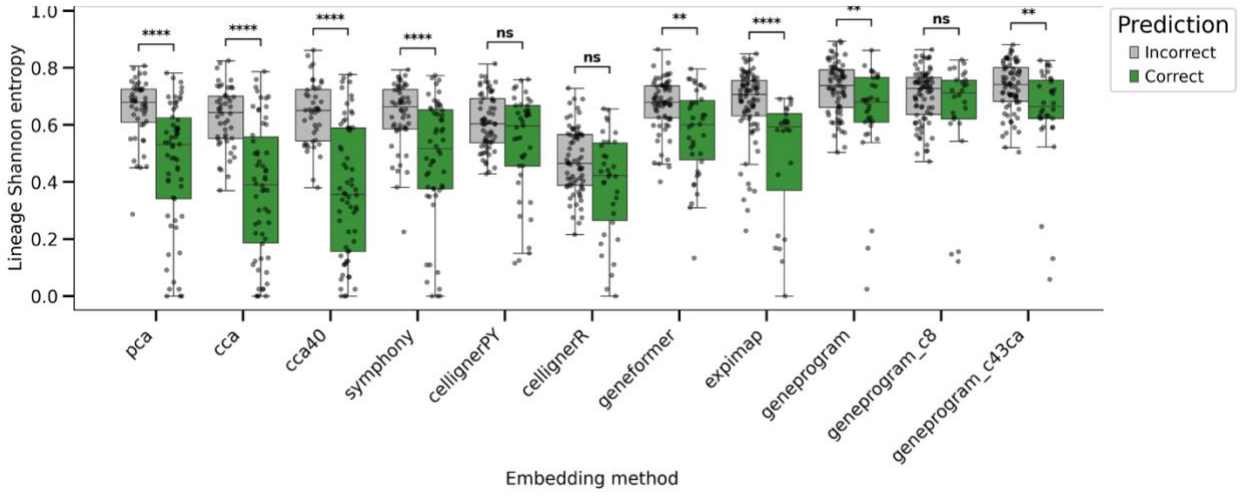


Figure 13. Boxplots for comparison of neighborhood entropy between correctly and incorrectly predicted cell lines.
Each dot is a cell line; stars indicate significantly different mean. P-value derived with Wilcoxon rank-sum test and adjusted using Benjamini-Hochberg correction.

different prediction groups. This indicates that when the embedding methods correctly classified a cell line's lineage, the neighborhood was more homogenous, enriching for the correct label,

and when the cell line was misclassified, the cell line likely mapped to patients from multiple lineages. This suggests that entropy could serve as a confidence measure for some of these methods, but would require further calibration for each method.

Entropy relates misclassification with diversity of neighborhood at a high level across all cell lines, but we wanted to understand if it's the same lineages that drove misclassification across all or a subset of cell lines. Further, for the methods with high batch variable contributions, it should be generally true that cell lines are poorly separated from each other, and therefore a large portion of cell lines should have mapped to a similar spot in the reference dataset. If this was true, then not only would the same lineages drive misclassification, but we should also observe the same patients driving misclassification. To validate this hypothesis, for each embedding method, we collected all tumor cells that were included in any cell line's neighborhood used for label transfer evaluation, and labeled each appearance of the patient as either "coherent", "driver", or "passenger". A patient's appearance is coherent if that patient appeared in the neighborhood of a cell line in the same lineage as the patient. However, if a patient appeared in the neighborhood of a cell line that has a differing lineage as the patient, then that appearance contributed to either misclassification or entropy. If that neighborhood ended up misclassifying the cell line's lineage as the patient in question, then that appearance "drove" this misclassification, otherwise, this appearance was a "passenger". We then count the number of each category of appearance for each patient across all cell line neighborhoods for each method. If a method mapped cell lines to just a handful of patients, we should see just a few patients with

high counts of driver appearance. To illustrate, we will compare this neighborhood analysis between CCA and CellignerR (Figure 14,15).

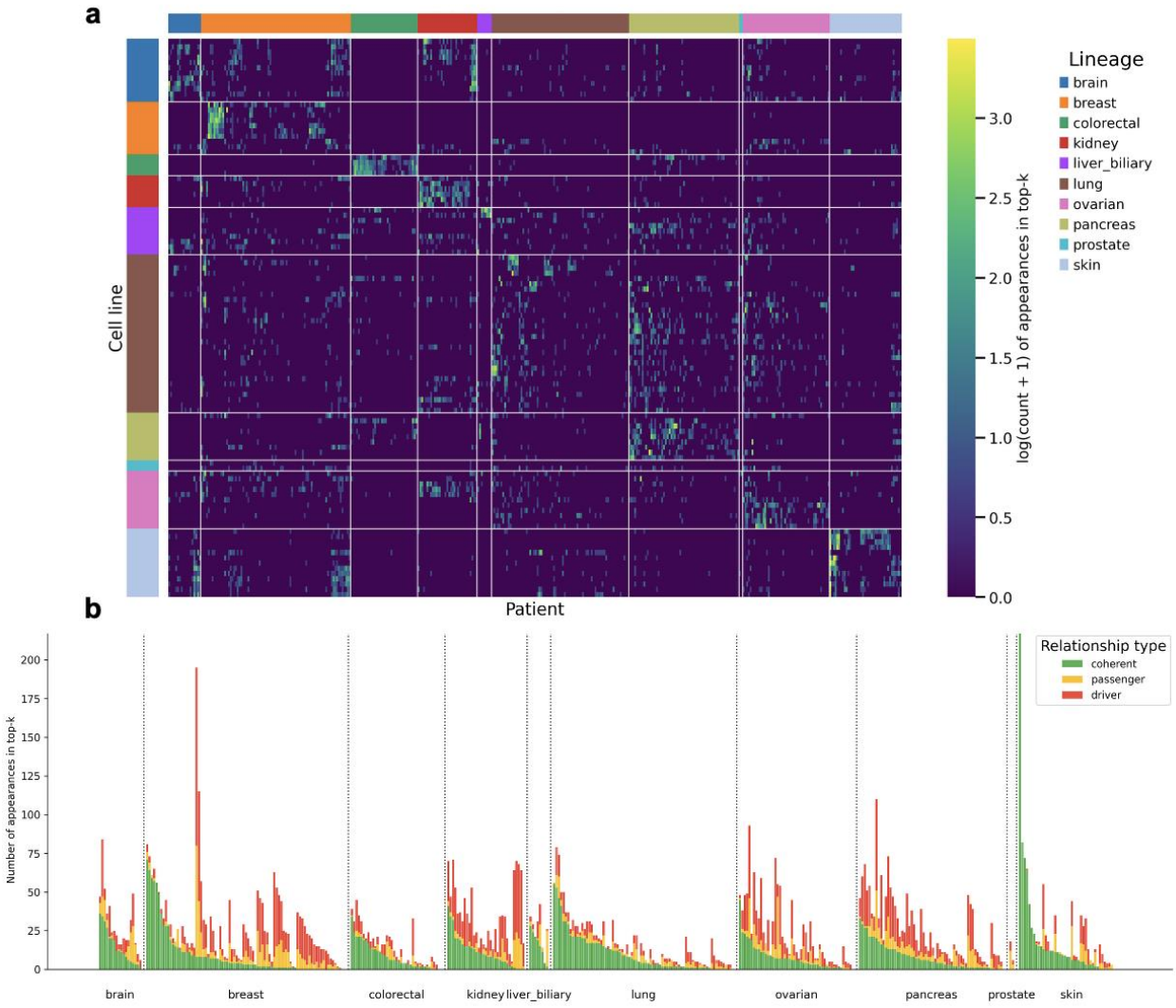


Figure 14. Neighborhood relationship analysis for CCA.
a. Heatmap showing the log appearance of each patient in each cell line's 100 nearest neighbors. Patients with 0 appearance in any cell line neighborhood are not shown.
b. Bar plot showing the number of each type of appearance by patient across all cell line neighborhoods. Patients with 0 appearance are kept blank.

Across all lineages, almost every patient had some of their cells included in the neighborhood of cell lines mapped by CCA (Figure 14). Each patient had varying ratios of relationship types. For example, a few patients in the skin lineage were repeatedly mapped to by skin lines, resulting in a high peak composed of coherent types. In agreement with the relatively

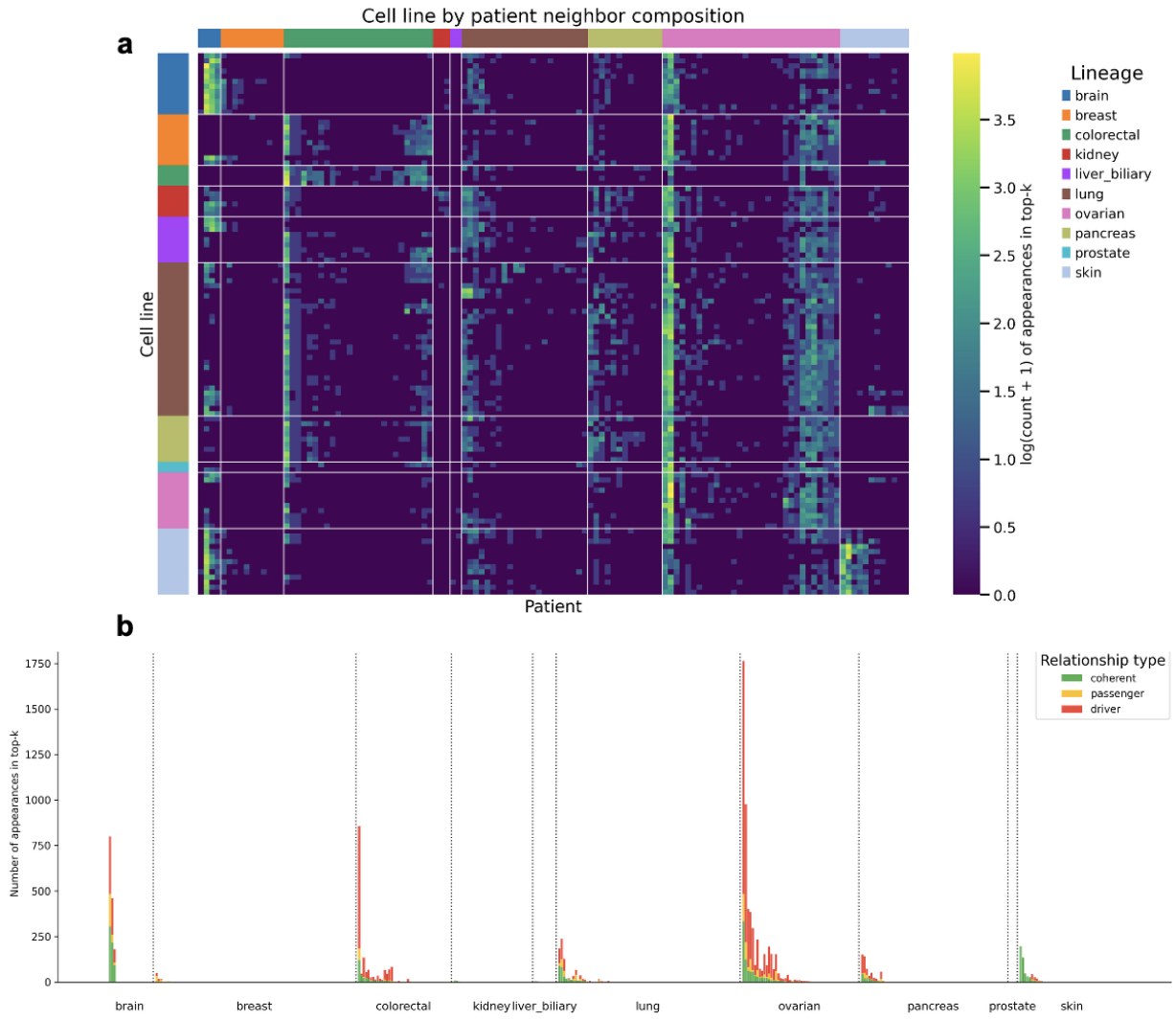


Figure 15. Neighborhood relationship analysis for CellignerR.
a. Heatmap showing the log appearance of each patient in each cell line's 100 nearest neighbors. Patients with 0 appearance in any cell line neighborhood are not shown.
b. Bar plot showing the number of each type of appearance by patient across all cell line neighborhoods. Patients with 0 appearance are kept blank.

high accuracy value reported earlier, in the heatmap, cell lines mapped to mostly patient's tumors that were from the same lineage, indicated by the high log-count appearance values in diagonal blocks. In comparison, in CellignerR, just a few ovarian patient tumors accounted for nearly one-third of all patient neighbors used for label transfer evaluation (Figure 15). Across all lineages, very few patients appeared in cell line neighborhoods (empty spaces in a lineage segment indicate patients with no counts). This is confirmed by the heatmap, showing that every cell line

mapped to just a few patients, and all neighborhoods mostly contained the same handful of ovarian patients, indicated by a column of high log-count appearance in the ovarian patient region.

These observations confirm that CellignerR mapped all cell lines to just a few patients. We checked whether this was true for the other methods that had a bias for ovarian misclassification, and found that PCA and Symphony didn't map to as many patients as CCA did, but sampled a wider array of patients than CellignerR did. Interestingly, though gene programs tended to misclassify cell lines as colorectal lineage, unlike CellignerR that mapped cell lines to mostly just a few patients of the ovarian lineage, gene programs mapped cell lines more evenly across the colorectal patients.

Taken together, across AUROC, label transfer accuracy, and VCA, CCA consistently ranked as the top performer. Among Celligner methods, PCA, Symphony and Geneformer, these methods had a tendency to map cell lines to a subset of ovarian patients, while gene programs mapped to mostly colorectal patients, which drove their degraded performance in the evaluation tasks. Though this explains partially the reason behind their performance, further analysis is required to fully unpack their respective failure modes, and means for improvements.

Chapter 5: Discussion

Here, we present initial efforts toward establishing a robust benchmarking framework for evaluating the fidelity of *ex vivo* cancer models at single-cell resolution. As a part of this framework, we standardized data pre-processing and neighborhood derivation across all methods, and introduced AUROC as a metric to quantify how well embeddings preserved expected mapping relationships. To further characterize large-scale cell line–patient mappings, we incorporated complementary metrics including label transfer accuracy and Shannon entropy to evaluate neighborhood composition across multiple scales and VCA to assess batch correction performance.

Considerations on Framework Design

Compared to the other implemented methods, expiMap, Geneformer, were applied in an out-of-the-box fashion. However, additional preprocessing or hyperparameter optimization could likely further improve their performance. For example, expiMap was implemented using Reactome gene sets as suggested in authors' tutorial, but incorporating cancer-specific pathways such as MSigDB Hallmark gene sets may allow the model to better capture biologically relevant variation, potentially improving its performance in aligning cell lines with patient tumors. With Geneformer, it's possible that extracting other embedding layers might offer features that capture more fine-grained relationships, and therefore improve its performance. Though we couldn't implement Symphony with soft-clustering correction for CCL-patient mapping, we aim to include its full functionality in future work, and expect it to be able to perform lineage mapping In the framework for Symphony, users could first perform harmony on the reference dataset to remove batch effects within reference before projecting query. In our case, adding this extra step

might significantly improve its batch correction capability and accurate neighborhood construction.

As an initial attempt to aggregate a large set of scRNA-seq tumor data, we opted to first down sampling and then manually reducing patient samples in over-represented lineages. However, this process might have inadvertently removed cell populations with unique variations that would otherwise serve significant roles in embedding construction for the de novo methods. Though ideally a comparison without needing to reduce dataset size would be ideal, one potential unbiased approach would be to first identify clusters with unique expression patterns in patient cell populations and then select a subset of cells from each binned cell population. Since patient tumor data serve as the reference for embedding construction, future work should focus on ensuring the optimal composition and diversity of populations included in the reference.

In analyzing the more complex CCL-tumor mapping experiment, we identified that "study" contributed a significant amount of variation in each embedding space. In our experiments, all cell lines were sourced from Kinker2020. However, comparing cell lines and tumors where both modalities contain multiple batches might require a more rigorous batch effect removal framework to fully account for the contribution of both batch and study variables.

In this work, we chose 100 as the label transfer evaluation radius through a manual comparison of method performance across all lineages at three threshold values of 25, 100 and 1000. We identified 100 as our testing radius both through empirical observation but also because it provides a reasonable middle ground between noise and signal. At a small *k* like 25, the neighborhood might not include enough biological variation at the single cell level. With increasing neighborhood radius, the average entropy across cell line neighborhoods steadily increases, which inherently includes more noise and will eventually become influenced by class

imbalance at large radius like 1000. Though each embedding approach might reach optimal accuracy at different neighborhood radius, the cross-method comparison result should remain valid at a moderate *k* value like 100, since the most effective method will place the most relevant cells in the nearest neighbors and should therefore be unaffected by radius size.

Here, we decided to evaluate mappings using cell line–level neighborhoods derived from nearest neighbor relationships. Neighborhood construction in this framework is based on a modified single-linkage approach, where proximity between cell line and patient cells is iteratively aggregated. This approach implicitly assumes that the cells within a given cell line occupy a relatively homogeneous transcriptional state. When this assumption is violated, neighborhood assignments may become unstable due to random tie-breaking. One potential alternative is to define neighborhoods based on distances to a cell line centroid, which may provide more stable estimates. Because neighborhood derivation is central to embedding evaluation, future work should focus on developing diagnostics to assess and validate assumptions of cell state homogeneity.

Though characterizing neighborhoods at a cell-line level enabled interpretable metrics such as entropy and AUROC, alternative strategies like clustering could also aid our interpretation. In Warren *et al*., clustering was done at varying resolutions to evaluate alignment of cancer subtypes. In doing so, they were able to identify stand-alone cell line clusters that didn't map to any tumor samples, serving as a negative control measure. Moreover, adapting clustering for evaluation provides an unbiased way to integrate batch removal and biological coherence evaluation. Under the current framework, both label transfer and AUROC were calculated considering each cell line's nearest *reference* neighbor, when cells the rest of the query could rank before or within reference neighbors, effectively discarding the relative

position of same-batch neighbors. We incorporated this in the batch correction experiment by assessing alternative AUROC calculation, but ultimately a more systematic and established metric would be to cluster and analyze the lineages or cell lines found within each cluster.

Considerations on Evaluated Methods

Our benchmarked methods are divided into two main categories: *de novo* methods and *a priori* methods. We constructed shared feature selection pipelines that standardizes the comparison across these categories. In all experiments except for identity mapping, the *a priori methods*, composed of Gene programs and Geneformer, consistently ranked in the lower performance group. They did not meaningfully correct for existing batch effects, and produced neighborhoods that did not align with our expectation. Both of these methods mapped most cell lines into patient neighborhoods that were incoherent with the existing annotated lineage, resulting in misclassification mode in ovarian for Geneformer, and colorectal in Gene programs. These results might be the result of improper batch correction and an ineffective embedding space for representing variation necessary for CCL-patient mapping. This suggests that pre-training and fine tuning used by Geneformer might not yield an embedding space that's inducible to more complex tasks like cell line lineage alignment; on the other hand, the learned gene programs used in this work might not capture phenotypic programs shared across *in vitro* and *in vivo* expression profiles.

In aim 3, within the *de novo* methods, PCA and Symphony don't apply active batch correction, while CCA, Celligner and expiMap actively remove batch signals. Our VCA results suggest that CCA applies the strongest batch correction, followed by Celligner, then expiMap, while PCA and Symphony retains much of the batch effect present. Among these, expiMap often performs poorly in neighborhood coherence as a result of poor resolution of cell line or lineage

labels. Though Symphony and PCA didn't actively correct for batch effect, they had fewer misclassification errors compared to Celligner. Since Celligner applies correction based on neighbor calculation, it's possible that our dataset contained CCL-like expression profiles in the ovarian lineage, and Celligner's correction exacerbated these outlier pairs. Though further tests are needed to validate this hypothesis, if true, this means that the Celligner approach is particularly sensitive to its initial HVG space. One solution might be to perform MNN after correcting for such anomaly in neighbor pairs, however such correction would need to be analyzed uniquely for each CCL-tumor dataset pair, and might not be feasible for increasingly complex mapping scenarios with larger datasets. Though the current work wasn't able to evaluate the efficacy of lineage mapping using the full Symphony implementation, given its robust batch effect removal we would expect it to be able to map cell lines to their corresponding lineages. Future work should focus on testing this hypothesis, and probe its failure mode in a similar fashion as performed in aim 3 if unexpected behavior is observed.

Among all methods tested, CCA stood out as the consistent top-performer. Unlike other methods, CCA was the only approach that identifies and applies correction separately for both datasets, and fundamentally guarantees that the resulting subspace is composed of shared variation. Other methods attempt to remove portions of batch effect through local corrections in Celligner or SymphonyALT, or non-linear objective optimization in expiMap. In lineage mapping, Warren et al observed that cell lines co-clustered with tumors from the same lineage, though brain lineage had poor agreement. Of all methods tested, CCA most accurately recapitulated this observation, despite some difficulties aligning lung cell lines. In addition, CCA also mapped more uniformly to most patients, whereas other methods tended to narrowly map cell lines to just a few patients. Though further characterization is needed to fully validate the

matched cell lines in CCA embedding space, utilizing the metrics we have developed in this framework, our results suggest that CCA most effectively eliminated batch effects, and holds strong potential as a basis for a method capable of aligning CCL with tumor samples at a single-cell level.

One major limitation of this work lies in the incomplete construction of evaluation label for CCL-patient mapping. Under both AUROC and label transfer tasks, metrics are derived with the assumption that each cell line should map to its corresponding lineage. However, Raghavan et al demonstrated that within the same lineage, cell lines with specific states might only map to a subset of patients[26], and Warren et al showed that certain cell lines with EMT state might not map to tumors to begin with[13]. Therefore, future efforts should consider using transcriptomic state as an alternative evaluation label. Since states are often shared across patients[5], this new evaluation lens could shed light on the mode of misclassifications observed across methods in aim 3. Accordingly, future experiments could focus on single-lineage mapping, where multiple cell lines of the same lineage are mapped to patient tumors of that lineage. This could enable a lineage-specific dissection of in vivo states as represented by existing cell lines.

Given that CellignerR successfully integrated cell lines and tumors at the bulk level, the observation that it suffered from biased mapping was surprising. We thought there were a few possible explanations. First, Celligner identifies mutual neighbors between the two datasets, then applies a correcting vector that pulls the query closer to its mutual nearest neighbor in the reference cohort. Our observations suggest that after correction, Celligner strongly biases mapping to ovarian tumors. Since the neighbors were calculated in HVG space, the mutual nearest neighbors calculated in HVG space may have already been strongly biased towards select ovarian tumor samples. If so, the MNN procedure may have further reinforced those biases,

pulling all cell lines towards specific samples. Notably, CellignerR differs from CellignerPY by mainly the number of reference neighbor considered when constructing mutual neighbor pairs. Given that CellignerR showed increased misclassification bias towards ovarian compared to CellignerPY, it's likely that the additional mutual nearest neighbor from CellignerR's *k*-neighbor radius connected cell lines to ovarian samples, therefore pulling them closer in correction (Figure 12). Therefore, if we included an HVG baseline, we might also expect ovarian-biased misclassification, but at a lesser degree. In contrast, gene programs might have simply contained a combination of features that represented a limited, biased aspect of the features in HVG space, distorting the relationship between cell lines and tumors, such that many non-colorectal cell lines were mapped to colorectal tumors. To test this, we could evaluate the distances at each dimension, and identify gene programs that contributed the most to each CCL-tumor relationship. Next, we evaluated Celligner on drastically different datasets compared to the original work. Here, we included much less variation on the cell line side. While the cell lines were a subset of CCLE lines evaluated in Warren *et al*, we curated a completely new tumor atlas, whereas Warren *et al*. mapped to TCGA. Finally, single-cell level comparisons enable the dissection of heterogeneous tumor cell populations. This added level of variation likely led to a different set of HVG that were used for MNN correction in our experiment compared to Warren *et al*.

Here, we are unable to fully benchmark our observation with Warren *et al.* mainly due to different dataset compositions. Previous studies showed that CCA enables the alignment of bulk and single-cell data[17]. As a next step, we could consider Kinker2020 as a "bridge" that connects the much larger scale CCLE bulk sequencing data with single cell level 3CA, leveraging the shared cell lines between CCLE and Kinker2020 as an anchor. This approach would allow us to

make use of the abundant bulk RNA profiles as an additional tool for batch correction, and expand on the cell line lineages available for use. In addition, future experiments could also benefit from fewer lineages to reduce the amount of potential noise present in the embedding space.

Chapter 6: Conclusion

In summary, we present a systematic framework for evaluating the fidelity of cancer cell lines to patient tumors at single-cell resolution. By benchmarking multiple embedding approaches and introducing complementary evaluation metrics, we show that mapping performance is not only influenced by technical aspects such as batch effect removal, but also determined by effective variance representation learned in embedding space construction.

These findings hold important implications for both experimental design and computational modeling. Selection of cancer cell lines is often guided by genomic similarity or lineage label, yet our results suggest that these criteria alone might be insufficient to ensure transcriptional fidelity. Moreover, many machine learning models rely on experimental data sourced from a small number of widely adapted cancer models for training and evaluation. However, doing so implicitly assumes that these models are faithful representations of patient tumor biology. Our results challenge this assumption, suggesting that discrepancies in cell states might limit translatability of findings.

Ultimately, this framework serves as a foundation for the principled selection and evaluation of preclinical cancer models. Our current work suggests that lineage alone is insufficient to fully capture the relationship between CCLs and patient tumors, highlighting the need to incorporate cell state–specific information into mapping strategies. As a next step, this framework can be extended to evaluate the concordance of cell states and to integrate bulk expression–based mappings as orthogonal validation. Incorporating both lineage and cell state information, while ensuring the alignment of shared variation across sequencing technology and biological systems, could enable the identification and development of models that better capture clinical phenotypes.

References

1. Ghandi, M. *et al.* Next-generation characterization of the Cancer Cell Line Encyclopedia. *Nature* **569**, 503–508 (2019).
2. Yang, W. *et al.* Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research* **41**, D955–D961 (2012).
3. Anderson, N. M. & Simon, M. C. The tumor microenvironment. *Current Biology* **30**, R921–R925 (2020).
4. Marusyk, A., Almendro, V. & Polyak, K. Intra-tumour heterogeneity: a looking glass for cancer? *Nat Rev Cancer* **12**, 323–334 (2012).
5. Gavish, A. *et al.* Hallmarks of transcriptional intratumour heterogeneity across a thousand tumours. *Nature* **618**, 598–606 (2023).
6. Kinker, G. S. *et al.* Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity. *Nat Genet* **52**, 1208–1218 (2020).
7. Trastulla, L., Noorbakhsh, J., Vazquez, F., McFarland, J. & Iorio, F. Computational estimation of quality and clinical relevance of cancer cell lines. *Molecular Systems Biology* **18**, e11017 (2022).
8. Ertel, A., Verghese, A., Byers, S. W., Ochs, M. & Tozeren, A. Pathway-specific differences between tumor cell lines and normal and tumor tissue cells. *Mol Cancer* **5**, 55 (2006).
9. Jiang, G. *et al.* Comprehensive comparison of molecular portraits between cell lines and tumors in breast cancer. *BMC Genomics* **17**, 525 (2016).

10. Niu, N. & Wang, L. *In Vitro* Human Cell Line Models to Predict Clinical Response to Anticancer Drugs. *Pharmacogenomics* **16**, 273–285 (2015).
11. Iorio, F. *et al.* A Landscape of Pharmacogenomic Interactions in Cancer. *Cell* **166**, 740–754 (2016).
12. Najgebauer, H. *et al.* CELLector: Genomics-Guided Selection of Cancer In Vitro Models. *Cell Systems* **10**, 424-432.e6 (2020).
13. Warren, A. *et al.* Global computational alignment of tumor and cell line transcriptional profiles. *Nat Commun* **12**, 22 (2021).
14. Pastushenko, I. & Blanpain, C. EMT Transition States during Tumor Progression and Metastasis. *Trends in Cell Biology* **29**, 212–226 (2019).
15. Tyler, M. *et al.* The Curated Cancer Cell Atlas provides a comprehensive characterization of tumors at single-cell resolution. *Nat Cancer* **6**, 1088–1101 (2025).
16. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* **36**, 411–420 (2018).
17. Zhang, F. *et al.* Defining inflammatory cell states in rheumatoid arthritis joint synovial tissues by integrating single-cell transcriptomics and mass cytometry. *Nat Immunol* **20**, 928–942 (2019).
18. Kang, J. B. *et al.* Efficient and precise single-cell reference atlas mapping with Symphony. *Nat Commun* **12**, 5890 (2021).
19. Lotfollahi, M. *et al.* Biologically informed deep learning to query gene programs in single-cell atlases. *Nat Cell Biol* **25**, 337–350 (2023).

20. Baek, S., Song, K. & Lee, I. Single-cell foundation models: bringing artificial intelligence into cell biology. *Exp Mol Med* **57**, 2169–2181 (2025).

21. Theodoris, C. V. *et al.* Transfer learning enables predictions in network biology. *Nature* **618**, 616–624 (2023).

22. Kotliar, D. *et al.* Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. *eLife* **8**, e43803 (2019).

23. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).

24. Aibar, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nat Methods* **14**, 1083–1086 (2017).

25. Hanahan, D. & Weinberg, R. A. Hallmarks of Cancer: The Next Generation. *Cell* **144**, 646–674 (2011).

26. Raghavan, S. *et al.* Microenvironment drives cell state, plasticity, and drug response in pancreatic cancer. *Cell* **184**, 6119-6137.e26 (2021).

27. Sibson, R. SLINK: An optimally efficient algorithm for the single-link cluster method, *The Computer Journal* **16**, 30–34 (1973).

28. Gierahn, T. M. *et al.* Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nat Methods* **14**, 395–398 (2017).

29. Abid, A., Zhang, M. J., Bagaria, V. K. & Zou, J. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nat Commun* **9**, 2134 (2018).

30. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* **36**, 421–427 (2018).

31. Liberzon, A. *et al.* Molecular signatures database (MSigDB) 3.0. *Bioinformatics* **27**, 1739–1740 (2011).

32. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.* **102**, 15545–15550 (2005).

33. Regev, A. *et al.* The Human Cell Atlas. *eLife* **6**, e27041 (2017).

34. Chen, H. *et al.* Quantized multi-task learning for context-specific representations of gene network dynamics. Preprint at https://doi.org/10.1101/2024.08.16.608180 (2024).

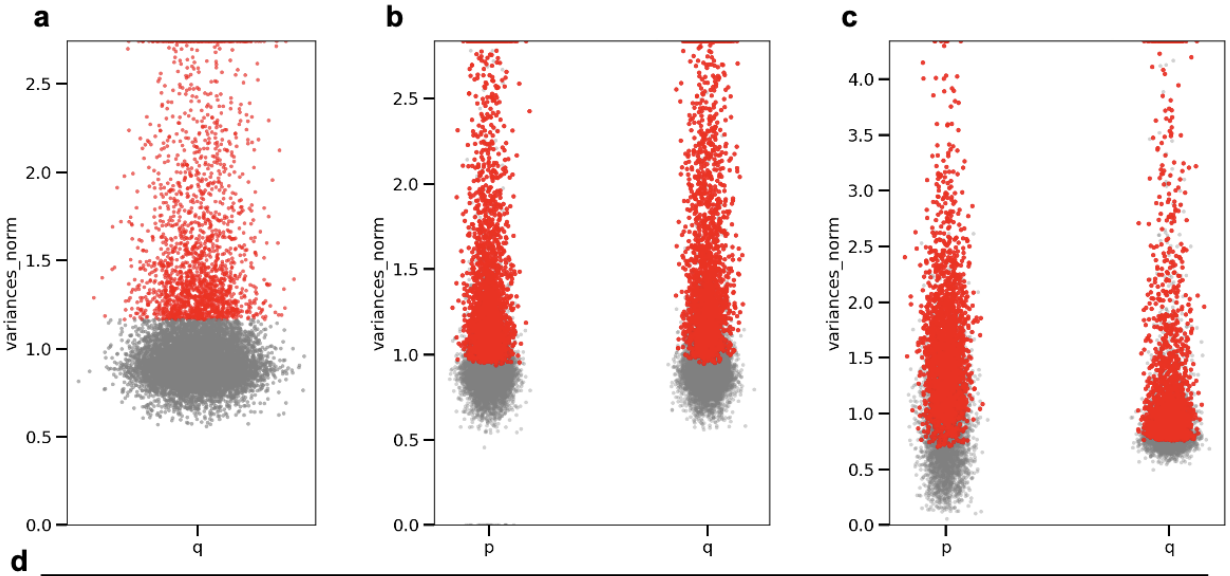
35. Gonzalez, I., Déjean, S., Martin, P. & Baccini, A. **CCA** : An *R* Package to Extend Canonical Correlation Analysis. *J. Stat. Soft.* **23**, (2008).

36. Bushel, R. Principal Variance Component Analysis. *National Institute of Environmental Health Sciences* (2025).

37. Millard, N. *et al.* Maximizing statistical power to detect clinically associated cell states with scPOST. Preprint at https://doi.org/10.1101/2020.11.23.390682 (2020).

38. Kothari, S. *et al.* Removing Batch Effects from Histopathological Images for Enhanced Cancer Diagnosis. *IEEE J. Biomed. Health Inform.* **18**, 765–772 (2014).

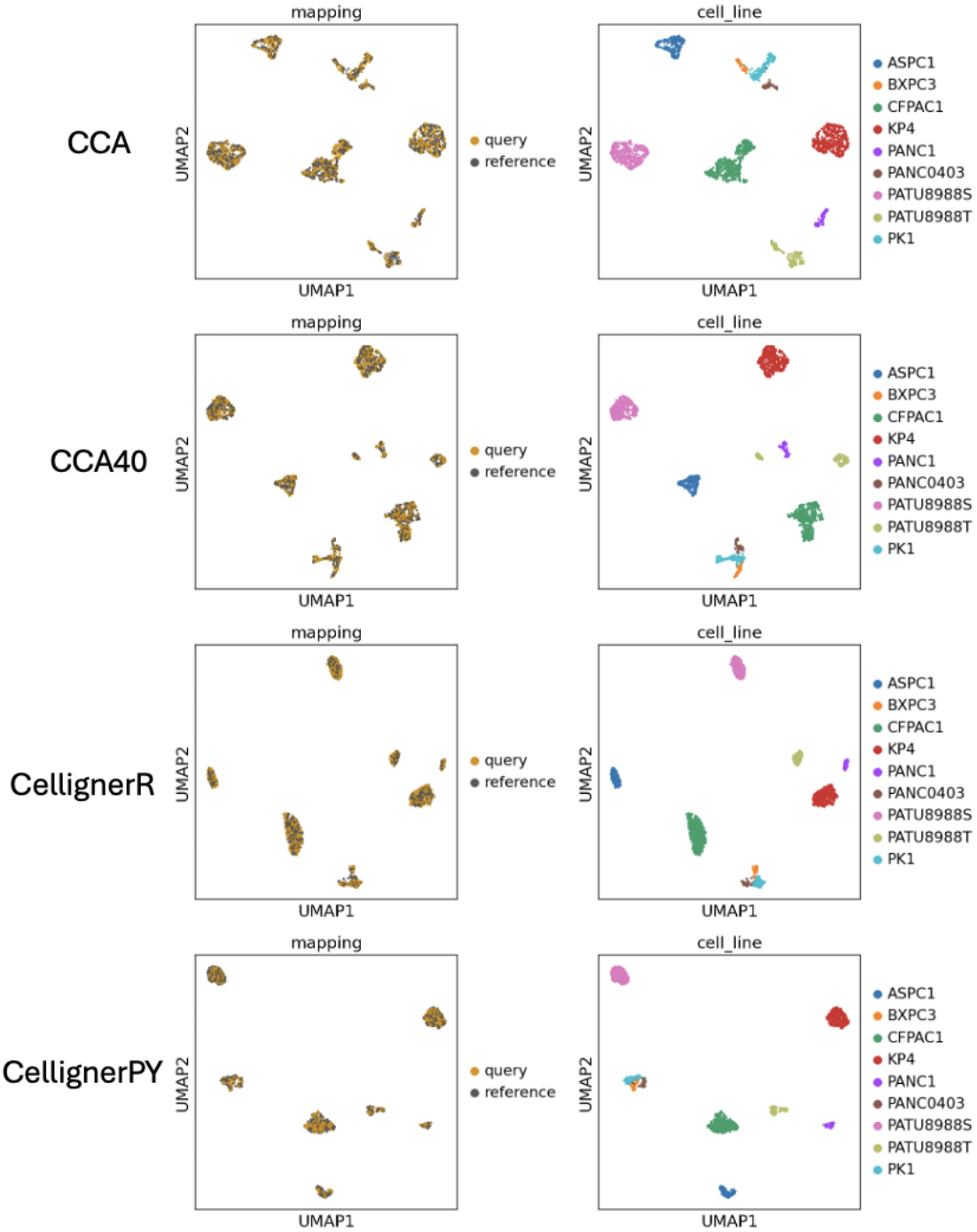
Supplementary Figures



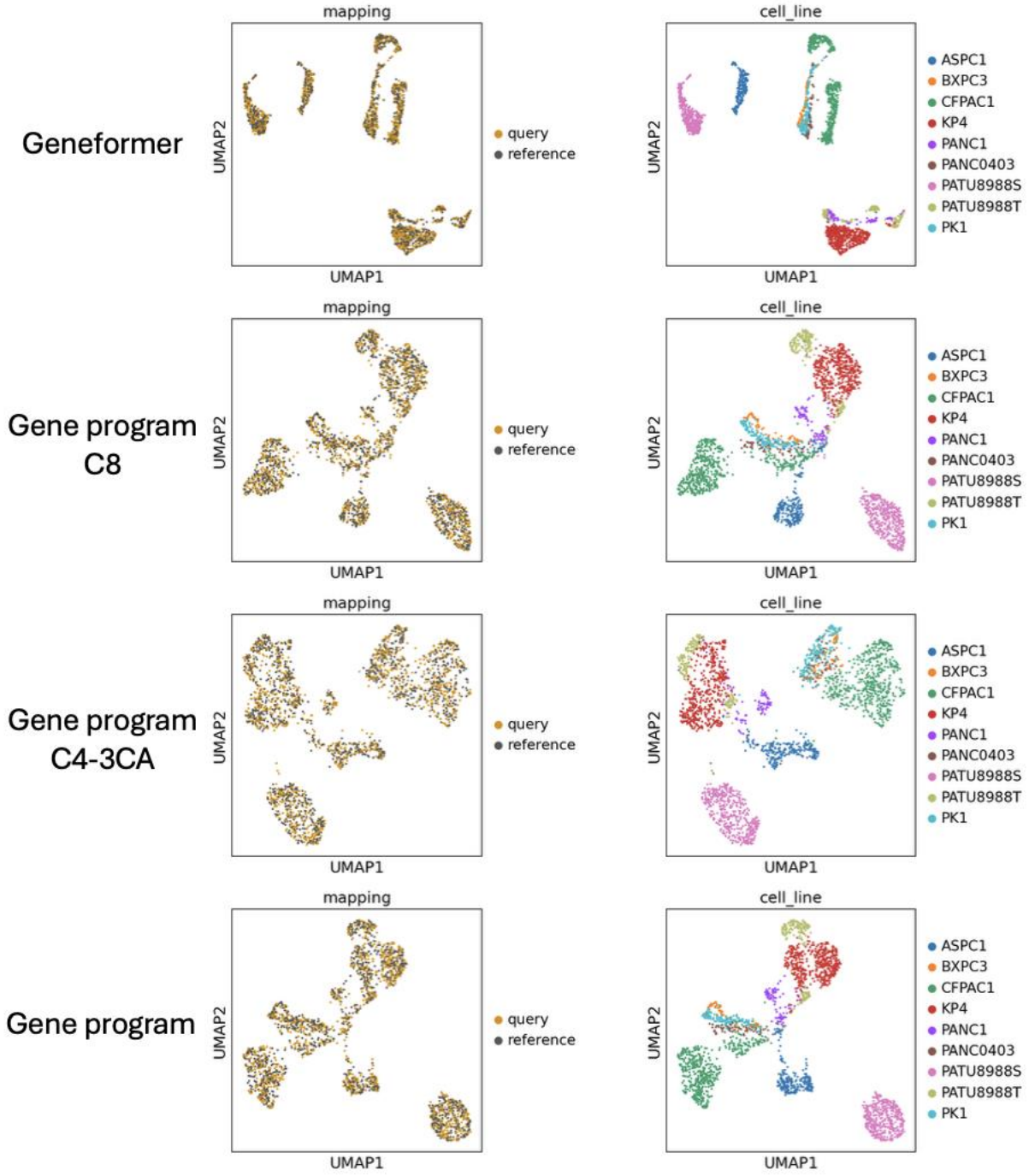
d

	Reference dataset	Query dataset	Shared HVG (used by *de novo*)	Shared genes (used by *a priori*)
Identity mapping	Tong2023	Tong2023	2000	11900
Batch correction	Kinker2020_pancreas	Tong2023	2000	11123
CCL-patient	3CA	Kinker2020	2000	6213

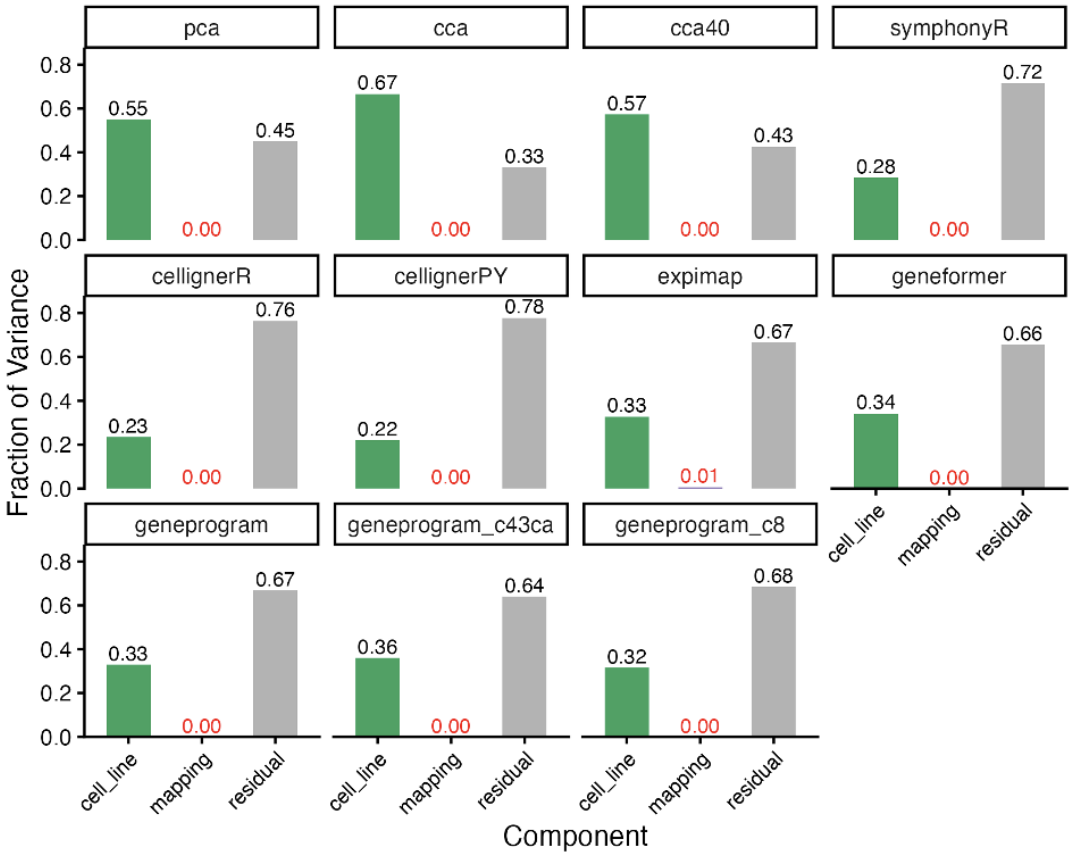
Supplementary Figure 1. Feature selection overview for each experiment.
a-c. Normalized variances of each gene in Tong2023 (**a**), batch correction experiment (**b**), and ccl-patient mapping experiment (**c**). p=reference; q=query.
d. Summary of feature selection dimensionality.



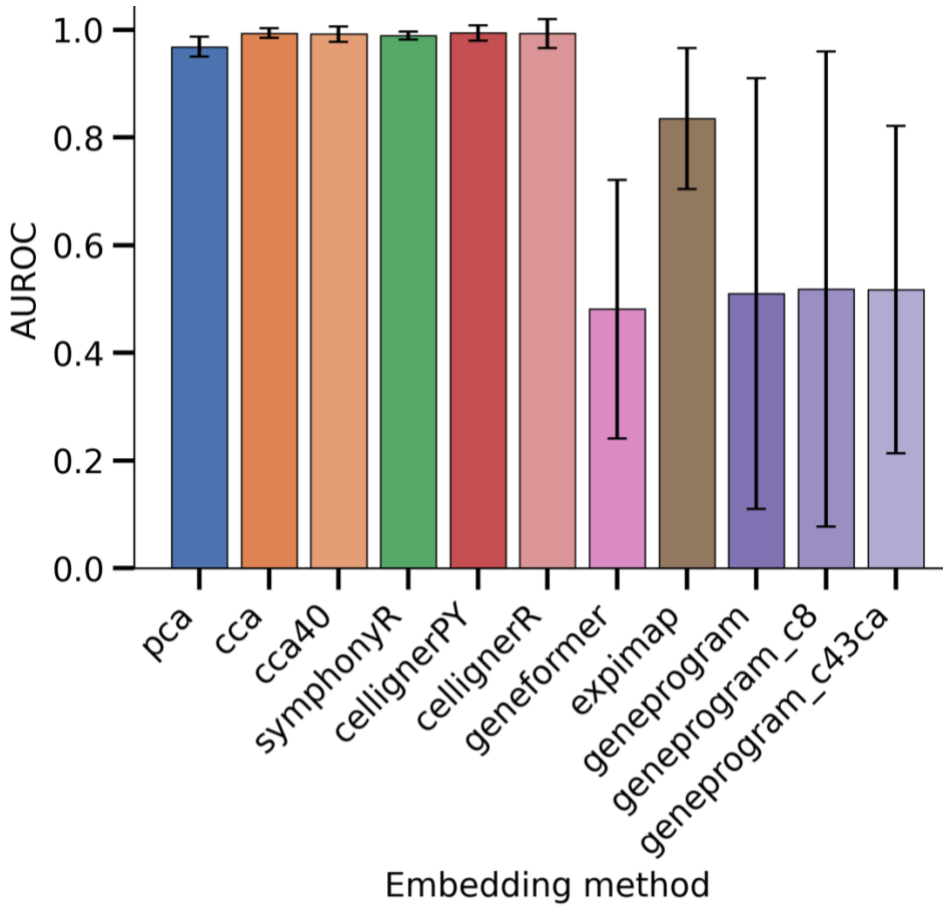
Supplementary Figure 2. UMAPs of evaluated embedding methods in identity mapping experiment. Left column is colored by query/reference designation, and right column is colored by cell lines in Tong2023.



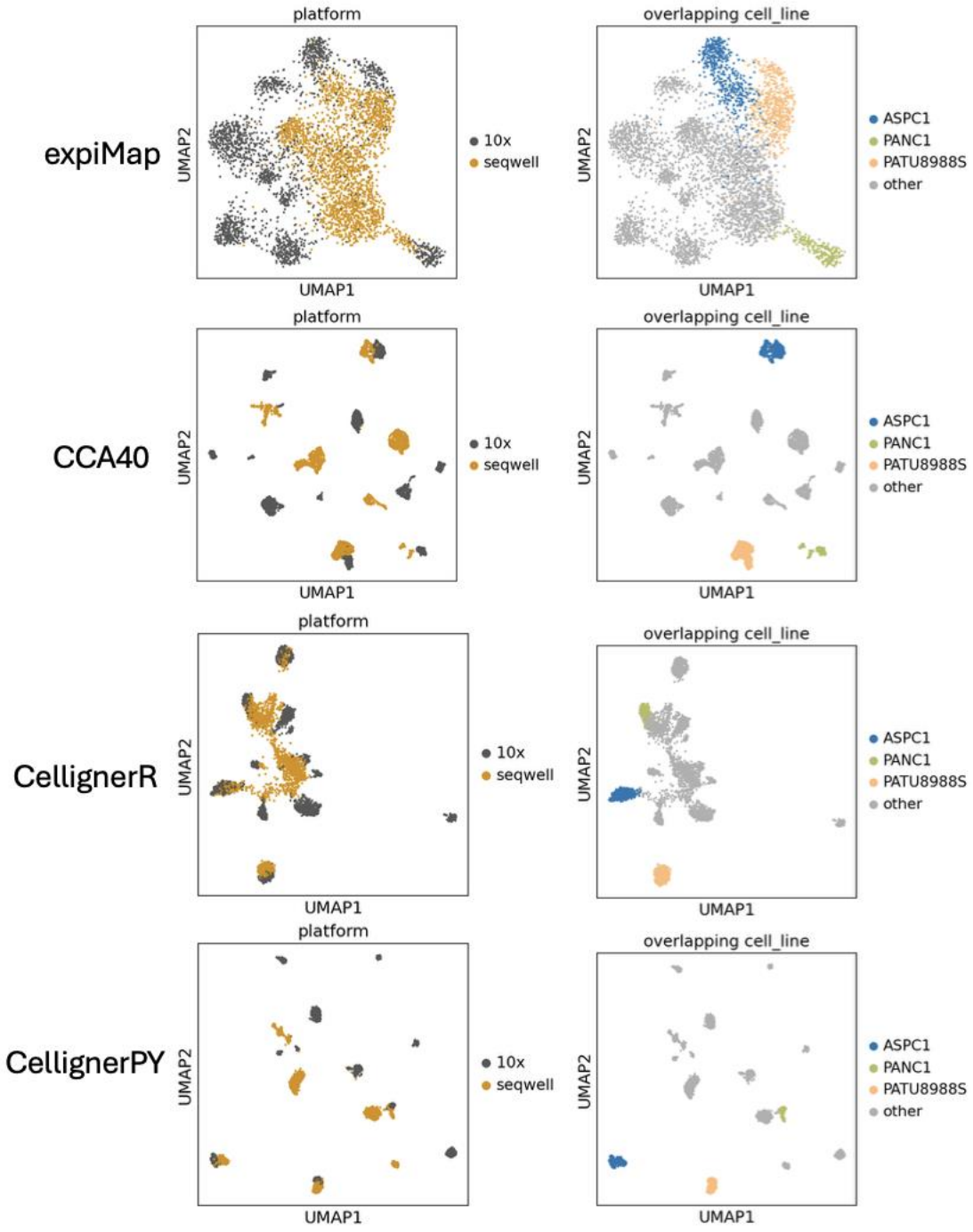
Supplementary Figure 2. UMAPs of evaluated embedding methods in identity mapping experiment (cont.)
Left column is colored by query/reference designation, and right column is colored by cell lines in Tong2023.



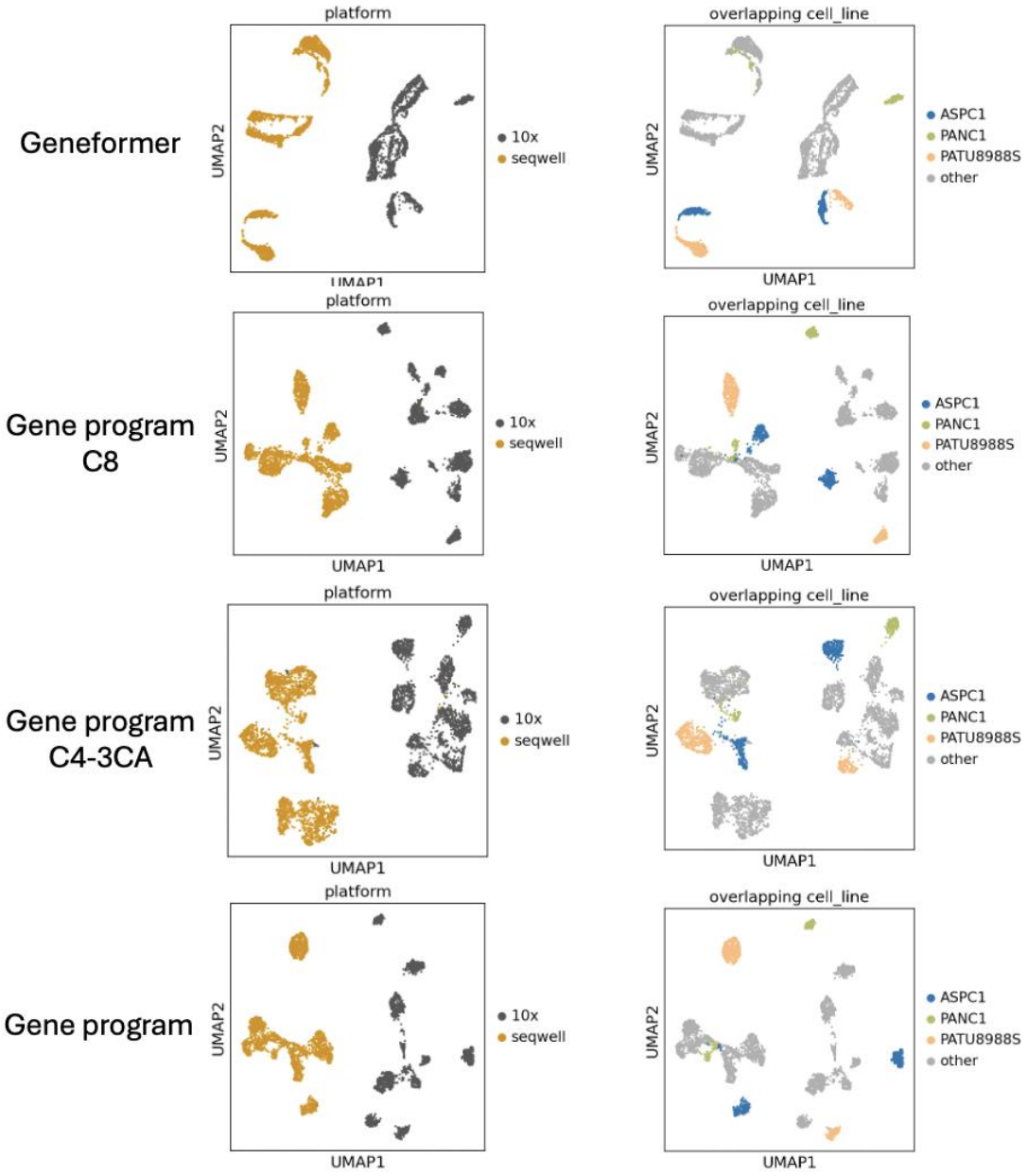
Supplementary Figure 3. VCA analysis bar plots in identity mapping experiment.
Proportion of variation explained by cell line, mapping, or residual in the evaluated embeddings. Batch variable contribution is highlighted in red.



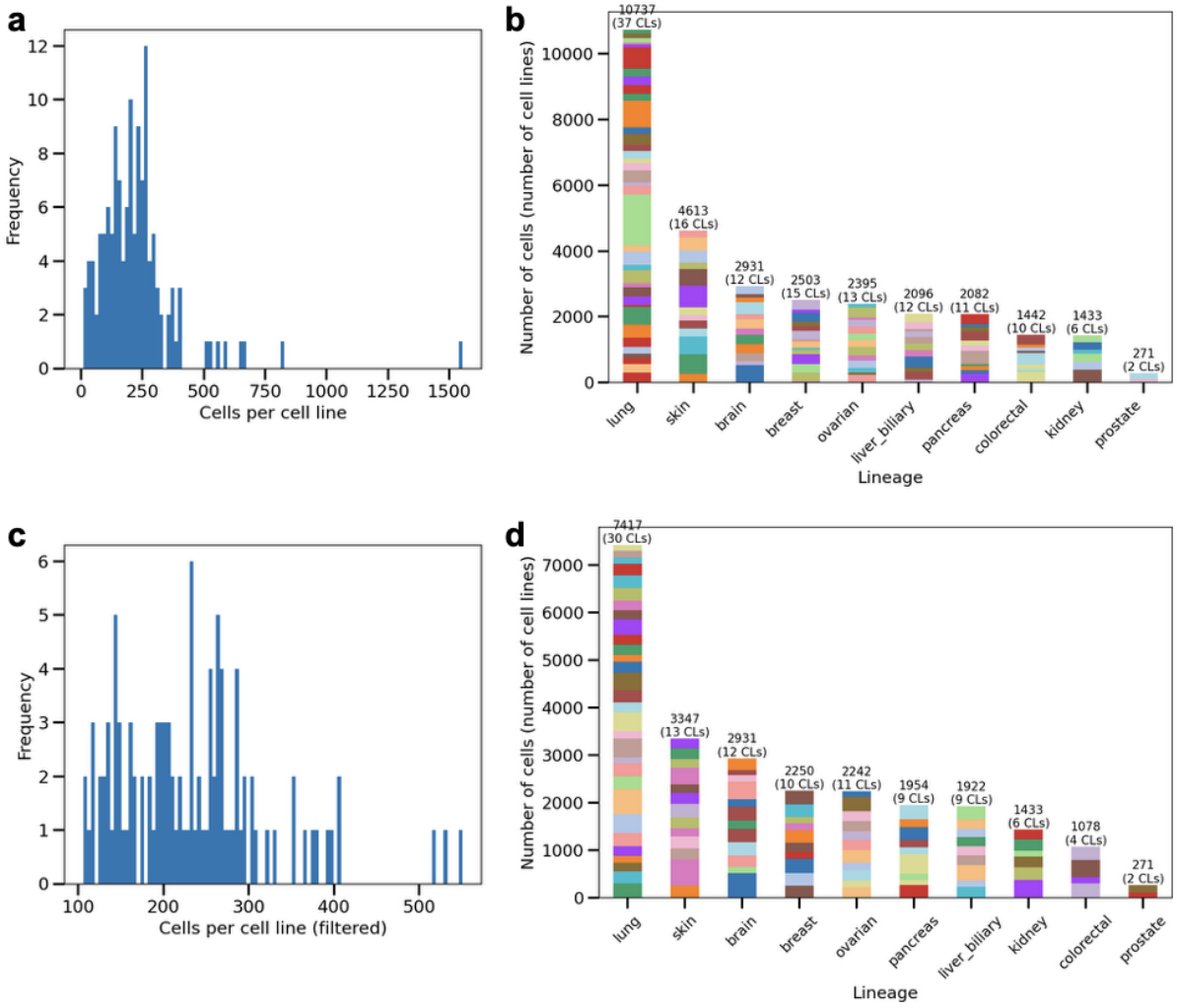
Supplementary Figure 4. Mean AUROC for alternate evaluation in batch correction experiment.
Error bars indicate 95% confidence interval (n=3); stars indicate significantly different mean compared to every other method at the least stringent alpha. P-value derived with Wilcoxon signed-rank test and adjusted using Benjamini-Hochberg correction.



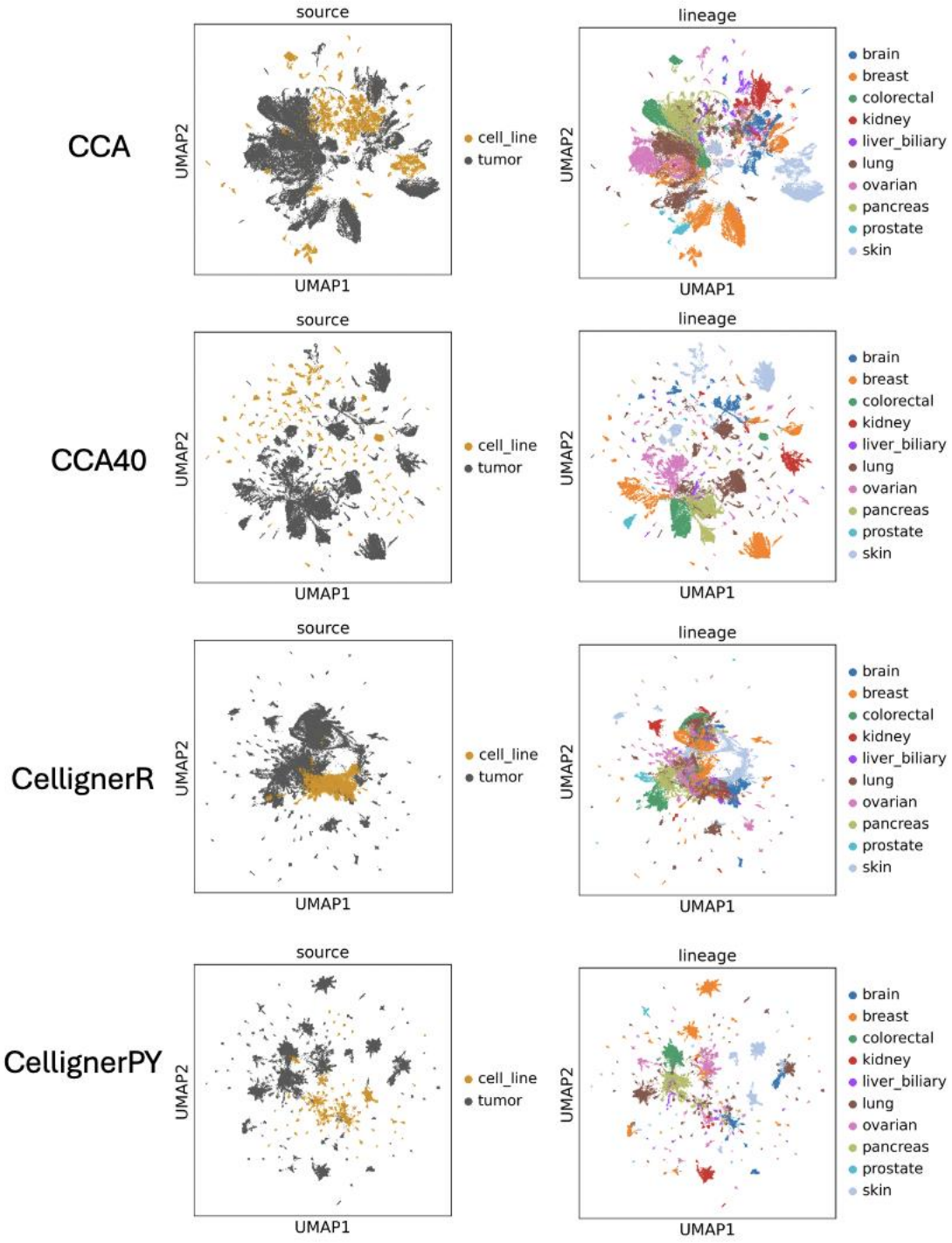
Supplementary Figure 5. UMAPs of evaluated embedding methods in batch correction experiment.
Left column is colored by sequencing platform, and right column is colored by shared cell lines across Tong2023 and Kinker2020.



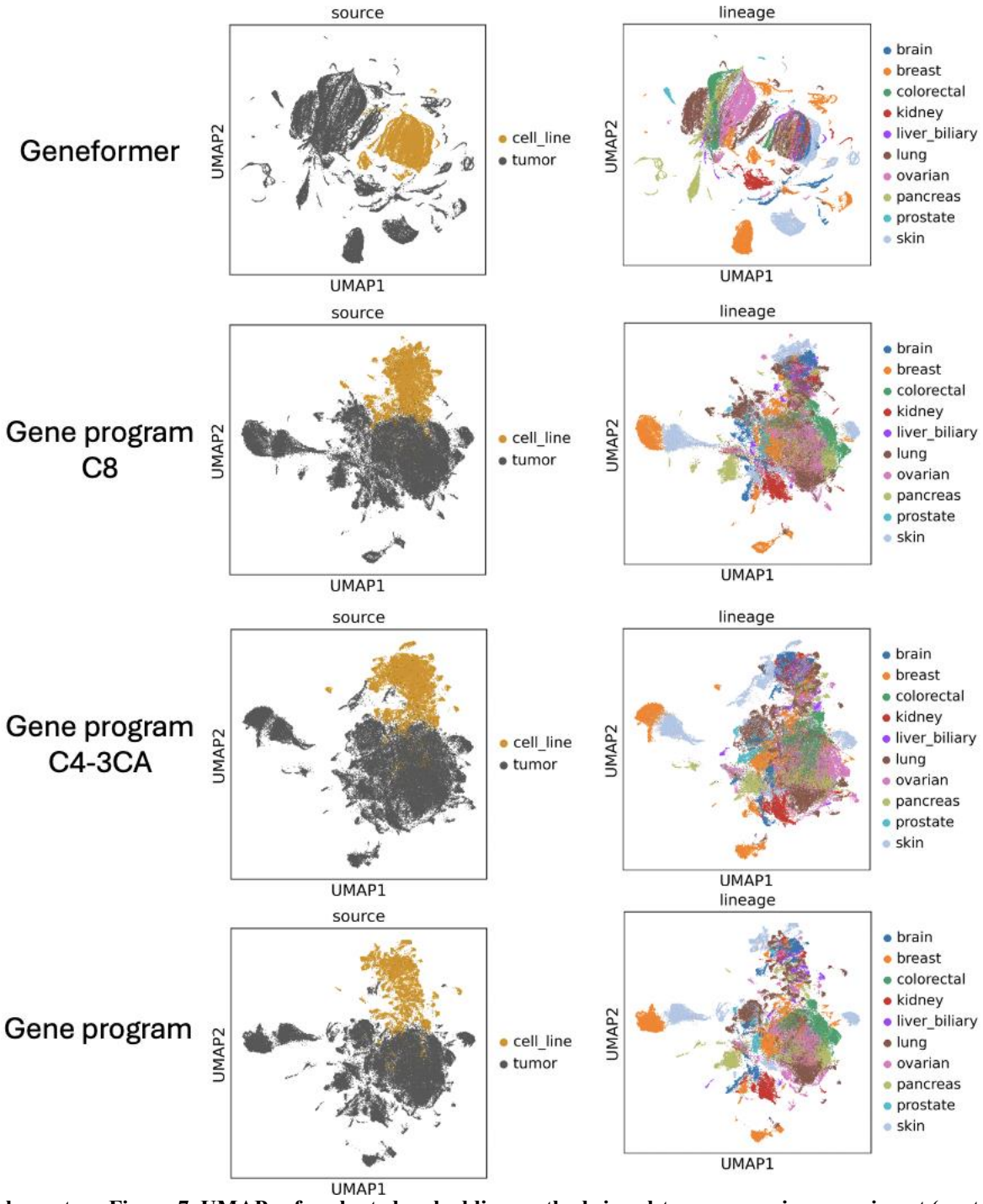
Supplementary Figure 5. UMAPs of evaluated embedding methods in batch correction experiment (cont.) Left column is colored by sequencing platform, and right column is colored by shared cell lines across Tong2023 and Kinker2020.



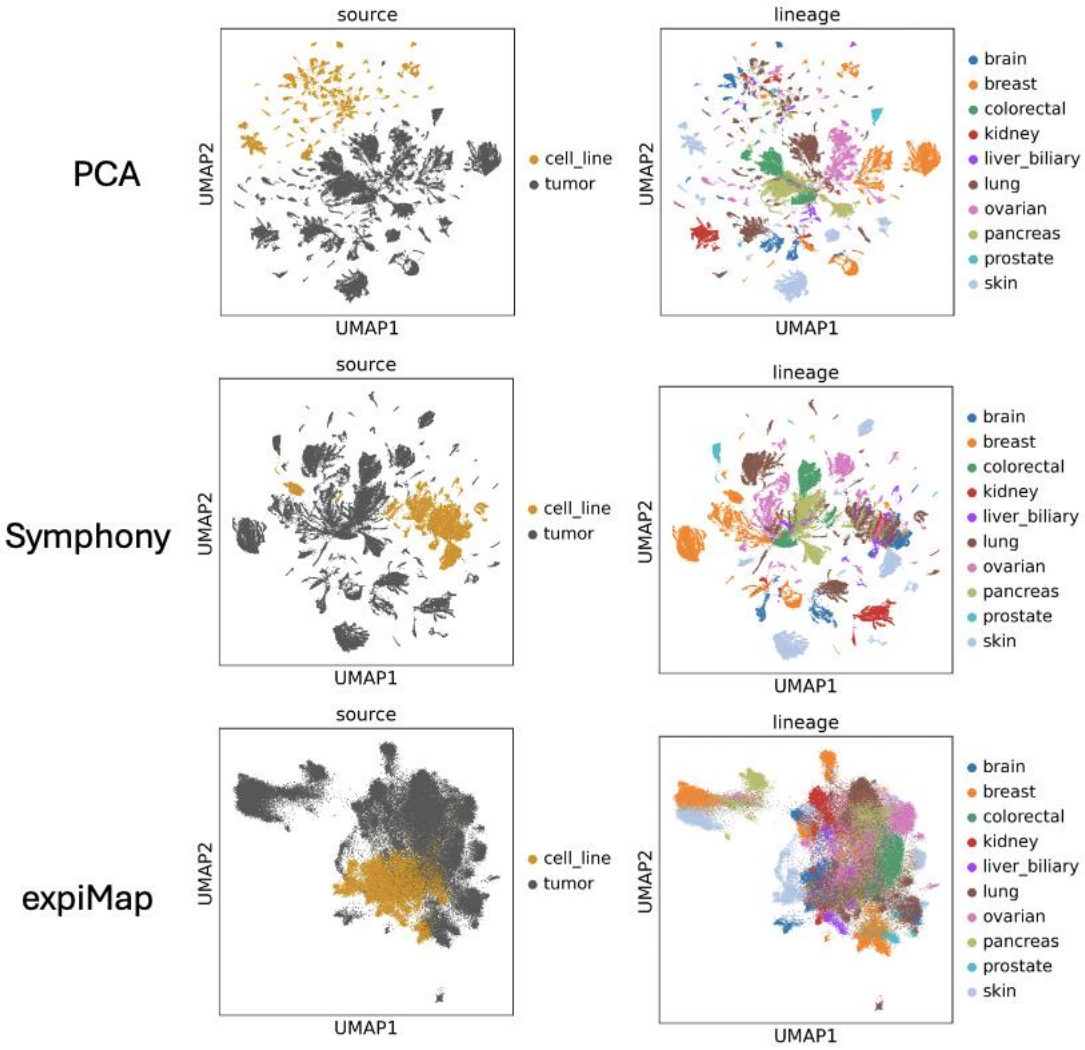
Supplementary Figure 6. Cell count analysis for trimming Kinker2020.
a,c. Frequency of cell counts in each cell line before (a) and after (c) cell count threshold.
b,d. stacked bar plot of cell counts in each lineage before (a) and after (c) cell count threshold.



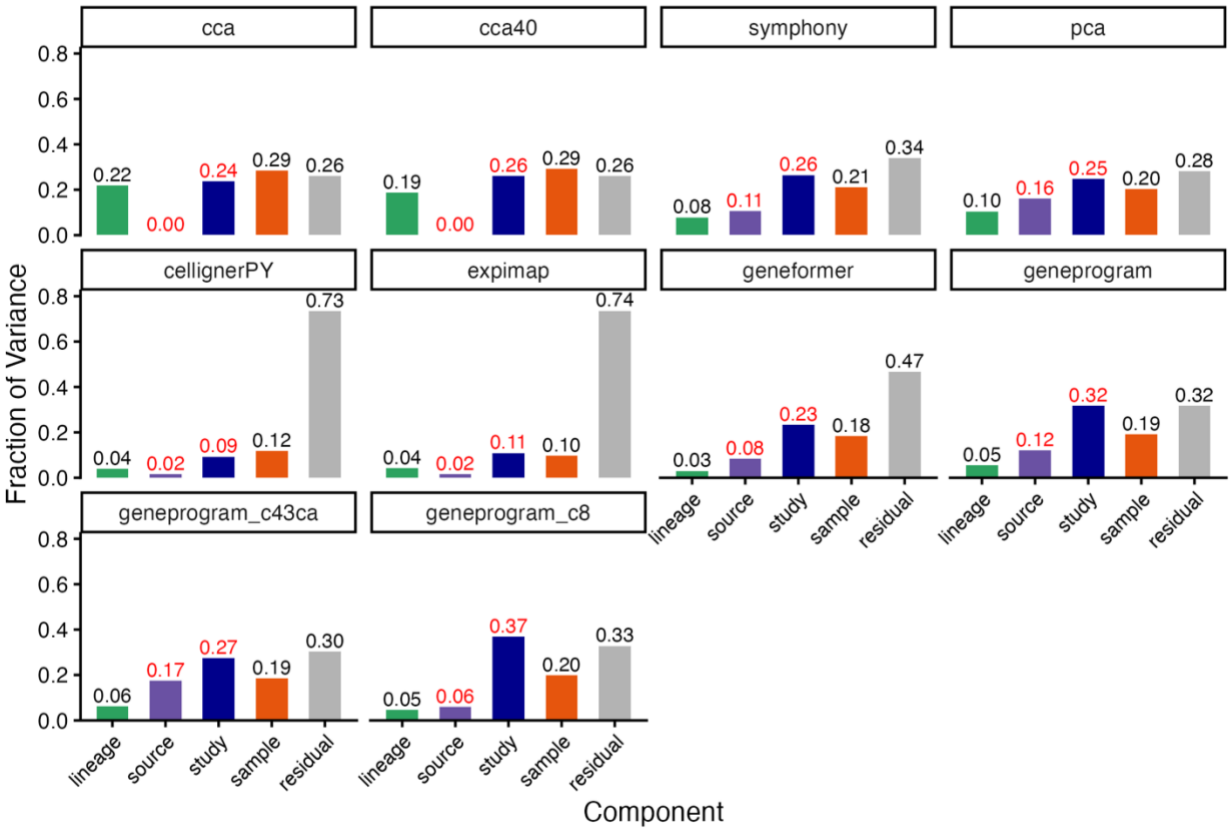
Supplementary Figure 7. UMAPs of evaluated embedding methods in ccl-tumor mapping experiment.
Left column is colored by biological source (from tumor or from cell lines), and right column is colored by lineage.



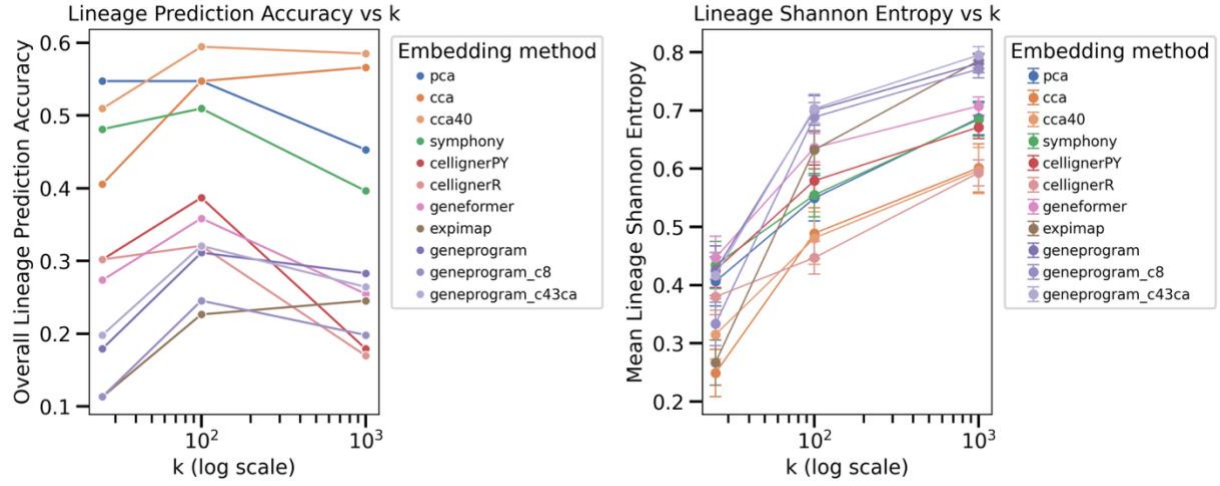
Supplementary Figure 7. UMAPs of evaluated embedding methods in ccl-tumor mapping experiment (cont.) Left column is colored by biological source (from tumor or from cell lines), and right column is colored by lineage.



Supplementary Figure 7. UMAPs of evaluated embedding methods in ccl-tumor mapping experiment (cont.) Left column is colored by biological source (from tumor or from cell lines), and right column is colored by lineage.



Supplementary Figure 8. VCA analysis bar plots in CCL-tumor mapping experiment using all variables. Proportion of variation explained by lineage, source, study, sample, or residual in the evaluated embeddings. Batch variable contributions are highlighted in red.



Supplementary Figure 9. Relationship between accuracy and entropy and neighborhood radius *k*.
Left: Accuracy across all mapped cell lines in CCL-tumor mapping experiment at k=25,100,1000, for each embedding method.
Right: Same as left but for Shannon entropy.

Supplementary Tables

Dataset	Data shape (before QC)	Data shape (after QC)
Tong2023	2232 cells x 30667 genes	2232 cells x 11900 genes
Kinker2020	53513 cells x 30314 genes	48552 cells x 19793 genes
3CA	109319 cells x 8070 genes	101599 cells x 6223 genes

Supplementary Table 1. Dataset curation preprocessing summary.

Dataset	Lineage	Data shape (before preprocessing)	Data shape (after preprocessing)
Pelka2021G4	Colorectal	111212 cells x 43113 genes	24027 cells x 23022 genes
Pelka2021G1	Colorectal	63744 cells x 43113 genes	30203 cells x 20077 genes
Lee2020	Colorectal	21657 cells x 22276 genes	21657 cells x 15684 genes
Pelka2021G2	Colorectal	117573 cells x 43113 genes	60818 cells x 20985 genes
Pelka2021G3	Colorectal	77586 cells x 43113 genes	36430 cells x 20237 genes
Neftel2019	Brain	16201 cells x 30401 genes	13137 cells x 16316 genes
Choudhury2022	Brain	58843 cells x 33538 genes	25147 cells x 20368 genes
Xing2021	Lung	118293 cells x 17541 genes	98978 cells x 17541 genes
Kim2020	Lung	32493 cells x 20793 genes	32493 cells x 16388 genes
Bischoff2021	Lung	120961 cells x 33514 genes	69233 cells x 22143 genes
Chan2021	Lung	86662 cells x 26036 genes	65592 cells x 20929 genes
Laughney2020	Lung	40505 cells x 19222 genes	38273 cells x 16835 genes
Qian2020	Lung	27262 cells x 22276 genes	27262 cells x 16166 genes
Song2019	Lung	8772 cells x 10147 genes	7861 cells x 8019 genes
Olbrecht2021	Ovarian	15528 cells x 33694 genes	14054 cells x 18591 genes
Regner2021	Ovarian	65144 cells x 24516 genes	47734 cells x 18907 genes
Geistlinger2020	Ovarian	41031 cells x 15328 genes	39517 cells x 15307 genes
Zhang2022	Ovarian	51786 cells x 32847 genes	50167 cells x 22733 genes
Qian2020	Ovarian	16951 cells x 22276 genes	16951 cells x 16066 genes
Nath2021	Ovarian	41581 cells x 21994 genes	14501 cells x 16810 genes

Pal2021G2	Breast	44225 cells x 33538 genes	28631 cells x 16965 genes
Wu2021	Breast	100064 cells x 29733 genes	76193 cells x 21293 genes
Pal2021G5	Breast	68782 cells x 33538 genes	42215 cells x 19656 genes
Pal2021G4	Breast	65129 cells x 33538 genes	27829 cells x 19158 genes
Griffiths2021	Breast	110568 cells x 21275 genes	109233 cells x 18537 genes
Pal2021G1	Breast	62887 cells x 33538 genes	48174 cells x 17814 genes
QIan2020	Breast	16537 cells x 22276 genes	16537 cells x 16021 genes
Gao2021	Breast	4143 cells x 33694 genes	3571 cells x 16014 genes
Bassez2021	Breast	226635 cells x 22567 genes	144053 cells x 21052 genes
Pal2021G3	Breast	64201 cells x 33538 genes	18153 cells x 18491 genes
Ma2019	Liver-biliary	3018 cells x 17917 genes	3018 cells x 13075 genes
Yost2019BCC	Skin	53030 cells x 23309 genes	52884 cells x 17733 genes
Biermann2022	Skin	136973 cells x 35652 genes	136973 cells x 29602 genes
Paulson2020	Skin	31376 cells x 16373 genes	17758 cells x 14594 genes
Ji2020	Skin	48164 cells x 32738 genes	48081 cells x 18587 genes
Dong2020	Prostate	21292 cells x 15709 genes	19819 cells x 15095 genes
Chen2021	Prostate	36424 cells x 25044 genes	32712 cells x 18201 genes
Krishna2021	Kidney	167283 cells x 15588 genes	148209 cells x 15581 genes
Obradovic2021	Kidney	19781 cells x 15003 genes	19781 cells x 15003 genes
Zhang2021	Kidney	29474 cells x 33694 genes	21542 cells x 18919 genes
Young2018	Kidney	125139 cells x 33694 genes	40409 cells x 20620 genes
Moncada2020	Pancreas	3659 cells x 19738 genes	3621 cells x 11403 genes
Lin2020	Pancreas	14926 cells x 22217 genes	14915 cells x 17019 genes
Raghavan2021	Pancreas	24412 cells x 16814 genes	13043 cells x 16570 genes
Peng2019	Pancreas	27953 cells x 18302 genes	27953 cells x 16191 genes
Steele2020	Pancreas	48570 cells x 32738 genes	31420 cells x 20510 genes

Supplementary Table 2. 3CA dataset overview.

Experiment	Reference dataset (# genes)	Query dataset (# of genes)	Shared HVG (used by *de novo*)	Shared genes (used by *a priori*)
Identity mapping	Tong2023 (11900)	Tong2023 (11900)	2000	11900
Batch correction	Kinker2020_pancreas (19793)	Tong2023 (11900)	2000	11123
CCL-patient	3CA (6223)	Kinker2020 (19793)	2000	6213

Supplementary Table 3. Feature selection results for each experiment.

embedding	cell_line	AUROC
pca	ASPC1	0.98051111
pca	PANC1	0.99932852
pca	PATU8988S	0.98579866
cca	ASPC1	0.99803948
cca	PANC1	0.99897696
cca	PATU8988S	0.98709571
cca40	ASPC1	0.99882244
cca40	PANC1	0.99972227
cca40	PATU8988S	0.98225163
symphonyR	ASPC1	0.98932306
symphonyR	PANC1	0.98819807
symphonyR	PATU8988S	0.99418976
cellignerPY	ASPC1	0.99744551
cellignerPY	PANC1	0.99230431
cellignerPY	PATU8988S	0.97797998
cellignerR	ASPC1	0.99738113
cellignerR	PANC1	0.99983828
cellignerR	PATU8988S	0.96807676
geneformer	ASPC1	0.80549942
geneformer	PANC1	0.70763416
geneformer	PATU8988S	0.8140359
expimap	ASPC1	0.8676317
expimap	PANC1	0.94227707
expimap	PATU8988S	0.88965837
geneprogram	ASPC1	0.76560576
geneprogram	PANC1	0.75805164
geneprogram	PATU8988S	0.83787575

geneprogram_c8	ASPC1	0.81884093
geneprogram_c8	PANC1	0.81518044
geneprogram_c8	PATU8988S	0.82522293
geneprogram_c43ca	ASPC1	0.7076292
geneprogram_c43ca	PANC1	0.60303398
geneprogram_c43ca	PATU8988S	0.83103979

Supplementary Table 4. AUROC scores of batch correction experiment by cell line.

lineage	n_3ca	n_3ca_fixed	n_kinker	ratio_before	ratio_after
brain	3354	3354	2931	1.14431934	1.14431934
breast	30241	15445	2503	12.0819017	6.17059529
colorectal	13237	6838	1442	9.17961165	4.74202497
kidney	4791	4791	1433	3.34333566	3.34333566
liver_biliary	1169	1169	2096	0.55772901	0.55772901
lung	14572	14572	10737	1.35717612	1.35717612
ovarian	11295	11295	2395	4.71607516	4.71607516
pancreas	10662	10662	2082	5.12103746	5.12103746
prostate	3117	1517	271	11.501845	5.59778598
skin	9161	9161	4613	1.98590939	1.98590939

Supplementary Table 5. Class imbalance correction summary.