CellO: Comprehensive and hierarchical cell type classification of human cells with the Cell Ontology

Matthew N. Bernstein[1], Zhongjie Ma[2], Michael Gleicher[2,4], and Colin N. Dewey[2,3,4,5]

1. Morgridge Institute for Research, Madison, WI, 53715, USA
2. Department of Computer Sciences, University of Wisconsin - Madison, Madison, WI, 53706, USA
3. Department of Biostatistics and Medical Informatics, University of Wisconsin - Madison, Madison, WI, 53792, USA
4. Senior author
5. Lead contact

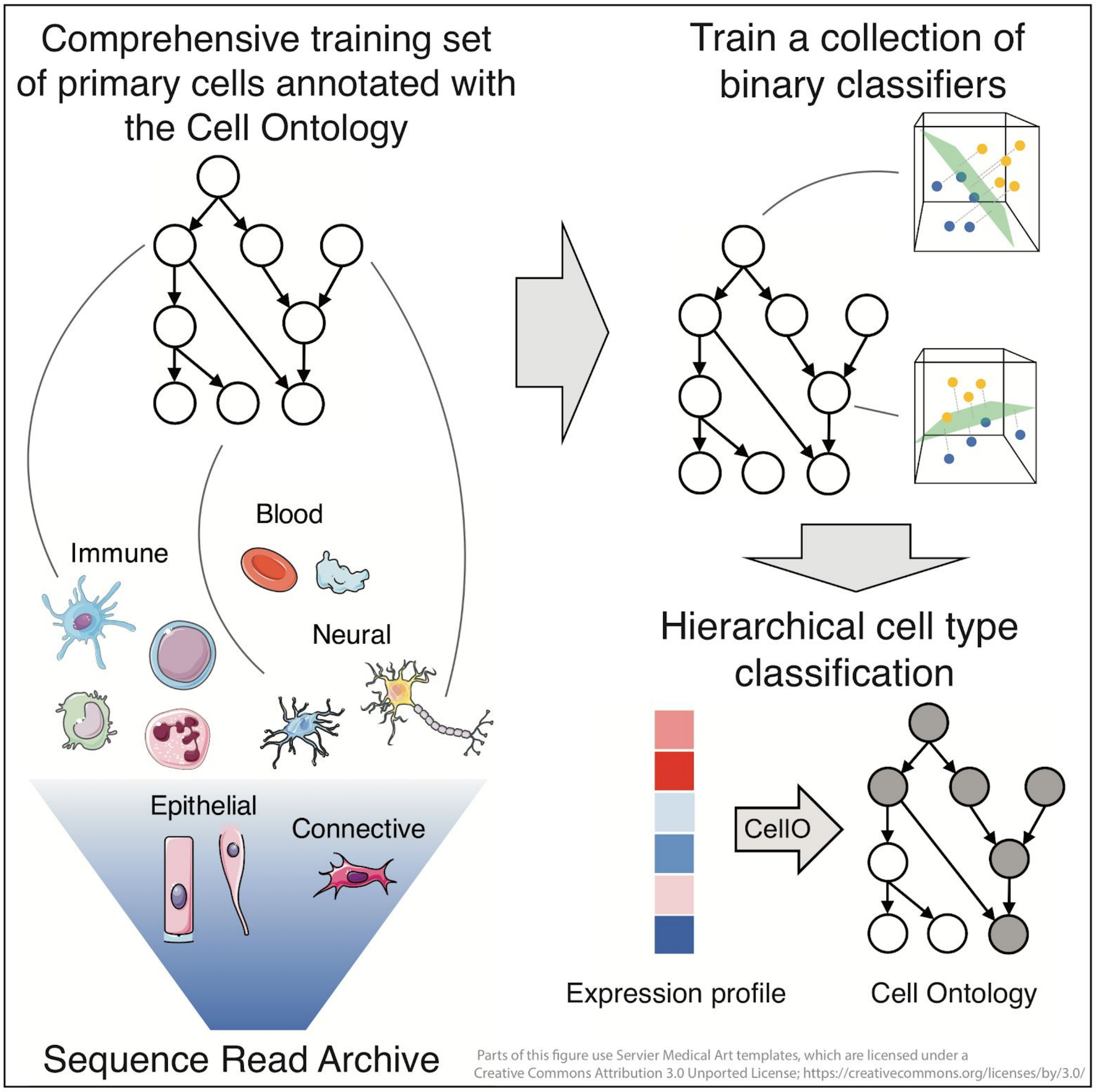
Correspondence: colin.dewey@wisc.edu

Keywords: cell type; ontology; classification; single cell RNA-Seq

Highlight

- We present CellO, a tool for hierarchically classifying cell type from single-cell RNA-seq data against the graph-structured Cell Ontology
- CellO is pre-trained on a comprehensive dataset comprising nearly all bulk RNA-seq primary cell samples in the Sequence Read Archive
- CellO achieves superior or comparable performance with existing methods while featuring a more comprehensive pre-packaged training set
- CellO is built with easily interpretable models which we expose through a novel web application, the CellO Viewer, for exploring cell type-specific signatures across the Cell Ontology

Graphical Abstract


Comprehensive training set of primary cells annotated with the Cell Ontology
Train a collection of binary classifiers
Immune
Blood
Neural
Epithelial
Connective
Sequence Read Archive
Hierarchical cell type classification
CellO
Expression profile
Cell Ontology
Parts of this figure use Servier Medical Art templates, which are licensed under a Creative Commons Attribution 3.0 Unported License; https://creativecommons.org/licenses/by/3.0/

Summary

Cell type annotation is a fundamental task in the analysis of single-cell RNA-sequencing data. In this work, we present CellO, a machine learning-based tool for annotating human RNA-seq data with the Cell Ontology. CellO enables accurate and standardized cell type classification of cell clusters by considering the rich hierarchical structure of known cell types, a source of prior knowledge that is not utilized by any existing methods. Furthermore, CellO comes pre-trained on a novel, comprehensive dataset of human, healthy, untreated primary samples in the Sequence Read Archive, which to the best of our knowledge, is the most diverse curated collection of primary cell data to date. CellO's comprehensive training set enables it to run out-of-the-box on diverse cell types and achieves competitive or even superior performance when compared to existing state-of-the-art methods. Lastly, CellO's linear models are easily interpreted, thereby enabling exploration of cell type-specific expression signatures across the ontology. To this end, we also present the CellO Viewer: a web application for exploring CellO's models across the ontology.

Introduction

Cell type annotation is a fundamental task in the analysis of single-cell RNA-sequencing (scRNA-seq) data. Recently, a number of computational tools have been developed for automating the cell type annotation task. Nonetheless, many of these tools suffer from certain disadvantages that inhibit their use. First, many existing methods require the user to provide either a set of marker genes associated with each cell type (A. W. Zhang et al. 2019; Pliner, Shendure, and Trapnell 2019) or a suitable training dataset with cells already annotated with cell type labels (Ma and Pellegrini 2020; Alquicira-Hernandez et al. 2019; Tan and Cahan 2019). Marker gene-based approaches are challenged by the fact that there is not a canonical set of marker genes for most cell types (X. Zhang et al. 2019). Furthermore, finding an appropriate and labelled training set that contains all of the cell types in the target dataset can be challenging, especially considering that existing approaches are sensitive to the chosen training set (Abdelaal et al. 2019).

Second, many existing methods use flat-classification. Flat classification suffers from the possibility that predictions are logically inconsistent with the hierarchy of cell types. Specifically, for a given query, a flat classifier may output a probability for a cell type that is larger than the classifier's output for its parent cell type in the hierarchy (Obozinski et al. 2008). Such incoherent outputs reduce the scientific usefulness of the classifier. We assert that framing the cell type classification task as that of hierarchical classification against the Cell Ontology (Bard, Rhee, and Ashburner 2005) poses a number of advantages over flat-classification. The Cell Ontology provides a comprehensive hierarchy of animal cell types encoded as a directed acyclic graph (DAG). This DAG provides a rich source of prior knowledge to the cell type classification task that remains un-utilized in flat classification. In addition, if the algorithm is uncertain about which specific cell type the cell may be, the use of a hierarchy allows the algorithm to place a cell internally within the graph rather than at a leaf node. Thus, for cells whose specific cell types are absent from the training set, a classifier that uses a hierarchy is capable of providing more informative output than simply claiming that the cell is “uncertain” as is implemented by some flat classifiers such as ACTINN (Ma and Pellegrini 2020).

Finally, those methods that do perform hierarchical classification do not make use of the rich hierarchical relationships between known cell types encoded by the Cell Ontology. For example, CHETAH (de Kanter et al. 2019) classifies cells against a hierarchy; however, CHETAH infers this hierarchy from the data rather than utilizing the existing hierarchy encoded by the Cell Ontology. Garnett (Pliner, Shendure, and Trapnell 2019) utilizes a hierarchy of cell types; however, these hierarchies must be pre-specified by the user. Furthermore, Garnett requires that each cell within the hierarchy be associated with a set of marker genes. To the best of our knowledge, the only method that utilizes the graph-structure of an ontology for the task at hand is URSA (Lee et al. 2013), which classifies gene expression profiles against the BRENDA Tissue Ontology (Gremse et al. 2011).

In this work, we present Cell Ontology-based Classification (CellO) a tool for annotating cells against the graph-structured Cell Ontology (**Figure 1A**). CellO is a discriminative, supervised machine learning approach for classifying clusters of cells in scRNA-seq data. CellO comes pre-trained on a novel comprehensive dataset comprising nearly all human primary samples in the Sequence Read Archive (Leinonen et al. 2011) and therefore arrives ready-to-run on diverse scRNA-seq datasets. CellO offers a complementary approach to marker gene-based methods for scenarios in which the test set contains cell types with poorly characterized marker genes.

Lastly, CellO makes extensive use of linear models, which are particularly amenable to interpretation. To enable their interpretation, we present a web-based tool, the CellO Viewer, for exploring the cell type expression signals uncovered by the model (https://uwgraphics.github.io/CellOViewer/). We benchmarked CellO on a collection of diverse single-cell datasets and found CellO capable of accurately annotating datasets that existing state-of-the-art and ready-to-run (i.e. come pre-trained) annotation methods were unable to accurately annotate, thus highlighting CellO's ability to annotate diverse datasets out-of-the-box. Through its use of the Cell Ontology, its comprehensive training set, and the interpretability of its models, CellO addresses the aforementioned limitations of existing tools, thus providing a practical tool for scRNA-seq cell type annotation. (**Figure 1B**).

Results

A novel curated RNA-seq dataset of human primary cells

In order to capture robust cell type signals, we sought a dataset of bulk RNA-seq samples comprising only healthy primary cells that originate from cells that have been isolated based on phenotypic characteristics downstream of gene expression itself (such as cell surface proteins). We thus avoid the circularity in using ground truth cell type labels determined by gene expression (via the expression of cell type-specific marker genes) as are often provided in scRNA-seq datasets. We did not wish to include cells that underwent multiple passages, were diseased, or underwent other treatments, such as in vitro differentiation, because these conditions alter gene expression. We therefore curated a novel dataset from the SRA consisting of healthy, untreated, primary cells. To do so efficiently, we leveraged the annotations provided by the MetaSRA project (Bernstein, Doan, and Dewey 2017) which includes sample-specific information including cell type, disease-state, and sample type. We then manually curated the samples selected via the MetaSRA by both annotating technical variables and refining cell type annotations (Transparent Methods).

This curation effort resulted in a dataset comprising 4,293 bulk RNA-seq samples from 264 studies. These samples were labeled with 310 cell type terms, of which 113 were the most-specific cell types in our dataset (i.e., no sample in our data was labelled with a descendent cell type term). These cell types were diverse, spanning multiple stages of development and differentiation (**Figure 1C**). We uniformly quantified and normalized (via log transcripts per million) gene expression from the raw RNA-seq data for these samples (**Figure**

2A, Transparent Methods). To the best of our knowledge, this dataset is the largest and most diverse set of bulk RNA-seq samples derived from only primary cells. Prior to this work, the most comprehensive bulk primary cell transcriptomic dataset was compiled by (Aran, Hu, and Butte 2017), which contains data for 64 cell types from 6 studies. Whereas our dataset consists of only RNA-seq data, this prior dataset included samples assayed with several other technologies, such as microarrays. Another comprehensive set of primary cell expression data was collected by Mabbott et al. (2013), which contains primary cell data from 745 samples from 105 studies; however, these data are exclusively from microarrays.

Novel applications of hierarchical classification methods

We frame the cell type classification task as hierarchical classification against the Cell Ontology's DAG. The hierarchical classification task is inherently a multi-label classification task where each input sample (i.e. cell) is mapped to a *set* of output labels (i.e. cell types). Hierarchical classification extends multi-label classification by further requiring that the output labels are *consistent* with the labels' DAG. That is, for each label in a given output set of labels, the label's parent labels are also in the output set (**Figure 1A**). Moreover, when training a hierarchical classifier, samples that are annotated with specific labels (*i.e.*, terms that are lower in the DAG) can be aggregated in order to train the classifier to recognize more general, ancestral labels in the DAG.

We implemented two strategies for performing hierarchical classification against the Cell Ontology's DAG that both come packaged with CellO. First, we implemented cascaded logistic regression (CLR; Obozinski et al., 2008), which entails classifying a sample in a top-down fashion from the root of the ontology downward via a collection of binary classifiers. Specifically, each binary classifier is associated with a cell type and is trained to classify a sample conditioned on the sample belonging to all of the cell type's parents in the ontology.

Next, we implemented a collection of one-vs.-rest binary classifiers for each cell type in the DAG. We will refer to this as the ``independent classifiers'' approach. This approach suffers from the possibility that the classifiers' outputs will be inconsistent with the hierarchical structure of the ontology. An inconsistency occurs when the output probability for a given cell type exceeds that of one of its parent cell types in the ontology (**Figure 2B**). We tested the use of independent logistic regression classifiers and found inconsistencies to be a frequent source of errors. Specifically, we performed leave-study-out cross-validation on the full set of bulk

RNA-seq data and examined the consistency of all edges that were adjacent to at least one cell type whose classifier produced a non-negligible probability (> 0.01) of the sample originating from that cell type. Of these edges, 12.1% were inconsistent (**Figure S1**). Nearly all samples (>99%) contained at least one inconsistent edge and 34% of samples contained at least one severely inconsistent edge in which the child classifier's probability exceeded the parent classifier's probability by at least 0.25. We will use the term "correction" to refer to the task of reconciling the outputs of independent classifiers with a hierarchy (**Figure 2B**).

To date, the one correction method that has been applied to the task at hand is Bayesian network correction (BNC), which is implemented in the URSA tool (Lee et al., 2013). Therefore, as a baseline, we implemented a BNC algorithm following the description in Lee *et al.* (2013) (Transparent Methods). We also tested two correction methods that have yet to be applied to the cell type classification task: isotonic regression correction (IR; Obozinski et al. 2008) and a heuristic procedure called the True Path Rule (TPR; Notaro et al. 2017). IR uses a projection-based approach for correction that entails finding a set of consistent output cell type probabilities that minimize the sum of squared differences to the raw, and possibly inconsistent, classifier output probabilities. In contrast, TPR uses a heuristic procedure that involves a bottom-up pass through the ontology such that the output of children classifiers are averaged with the output of the parent classifier to allow information flow across the ontology graph.

To test these correction methods, we first partitioned the bulk RNA-seq dataset into a pre-training and validation set (**Figure 2A**; Transparent Methods). Using this validation set, we performed a grid search to find the optimal parameters for training each binary logistic regression classifier, and given the optimal set of parameters, compared how well the aforementioned correction methods either enhanced or degraded accuracy over the samples in the validation set. Overall, we find that IR and TPR output probabilities similar to those output by the independent classifiers in regards to both average precision scores across the cell types in the validation set (**Figure 3A**) and precision-recall curves when considering each sample-cell type pair as an independent prediction (**Figure 3B**). This indicates that IR and TPR do not degrade performance in comparison to independent classifiers. In contrast, we found that the BNC approach significantly degraded performance (**Figure 3A, B**). We note that these results are in line with work by Obozinski *et al* (2008), which demonstrates that IR outperforms BNC on the hierarchical protein function prediction task. Although both IR and TPR yielded similar results, we use IR as our correction method of choice due to its simplicity.

We also used this partition of the bulk RNA-seq samples to tune the parameters of the CLR algorithm (Transparent Methods). We found that after tuning, both IR and CLR achieved similar median F1-scores and median average-precisions across cell types on the validation set (**Figure S13**) and therefore both are included in the CellO software and evaluated throughout the remainder of this study

Comparison to existing methods

We trained both CLR and IR on the full set of bulk RNA-seq samples in order to test their performance on single-cell data (**Figure 2A**). We note that training a single-cell classifier with bulk RNA-seq data may lead to models being poorly calibrated to the sparse single-cell expression profiles. To address this challenge, we first cluster single-cell data using the Leiden community-detection algorithm (Traag, Waltman, and van Eck 2019) using the default resolution parameter of 1.0, as implemented in the Scanpy Python package (Wolf, Angerer, and Theis 2018), and then compute each cluster's mean expression profile. The mean expression profiles are less sparse than those of the individual cells and thus better resemble the bulk RNA-seq data on which the algorithms were trained. CellO first classifies each cluster based on its mean expression profile and then assigns each cell to its cluster's assigned cell types.

We compiled a dataset consisting of 7,366 healthy primary cells originating from non-droplet-based RNA-seq assays, such as SMART-Seq2 (Picelli et al. 2013) and MARS-seq (Jaitin et al. 2014), from the SRA in a manner similar to that used for compiling the bulk RNA-seq training data. This dataset originated from 14 studies and were labeled with 125 cell type terms, of which 32 were most specific to the data. Of these cells, 4,936 were of cell types that were included in the bulk RNA-seq training set. This subset of cells originate from 12 studies and were labeled with 71 cell type terms of which 16 were most specific to the dataset. We note that for many of the cells used in this analysis, the ground-truth cell types provided by the authors of the data were determined via *in silico* and/or manual approaches (e.g. via heuristic marker-gene based approaches) and thus, this analysis can be understood as an analysis of the *consistency* between the cell types as annotated by the authors and those annotated by the automated methods explored in this work.

We use the subset of 4,936 cells to evaluate IR, CLR, as well as a baseline one-nearest-neighbor (1NN) algorithm that simply returns the cell type labels of the most similar sample in the training set to the query expression profile using Pearson correlation as the similarity metric. We evaluated two aspects of these algorithms' classifications. First, we compute the average-precision (a measure of area under the precision-recall curve) on each cell type's output probabilities. Second, for each cell, we evaluate a set of binary yes-no decisions for each cell type that result from thresholding the raw output probabilities and enforcing each cell to be annotated with only one most-specific cell type (**Figure 2B;** Transparent Methods). We evaluate these binary decisions using precision, recall, and F1-score (harmonic mean of precision and recall). We modified the evaluation metrics to take into account samples that were labelled with a general cell type, but not a specific cell type (e.g. T cell versus CD8+ T cell; Transparent Methods).

We found that IR and CLR out-performed 1NN according to F1-score, precision, and average-precision (**Figure 4A, B, C**). Specifically, IR, CLR, and 1NN produced median F1-scores of 0.81, 0.85, and 0.63 respectively across all cell types. The cell types on which classification performance was poor were generally more-specific and clustered within the hierarchy (**Figure 4D**). We note that CellO's average precision scores across cell types tend to be higher than its F1-scores (**Figure 4D**). This discrepancy indicates that CellO is doing well at discriminating among these cell types; however, the decision thresholds used by CellO to output hard-classifications for some cell types may be non-optimal. We hypothesize that this is due to some classifiers being poorly calibrated, especially for cell types lower in the ontology. This may due to there being fewer *studies* in the training set generating these cell types and thus, for a given cell type, CellO's binary classifiers may be more prone to fitting the batch effects present in these fewer studies (**Figure S16**).

We also compared CellO to two existing methods, scMatch (Hou, Denisenko, and Forrest 2019) and SingleR (Aran et al. 2019). scMatch and SingleR are most comparable to CellO because they come packaged with comprehensive reference datasets of human cells. Like CellO, these methods are designed to run out-of-the-box on diverse single-cell datasets. scMatch comes packaged with a reference dataset comprising data from the FANTOM5 project (Lizio et al. 2017). SingleR comes packaged with two comprehensive human reference datasets: a dataset comprising data from the Blueprint (Fernández et al. 2016) and ENCODE (Sloan et al. 2016) projects, and a reference set from the Human Primary Cell Atlas (Mabbott et al. 2013). We also built a reference set for SingleR from CellO's training set in order to isolate methodological differences between SingleR and CellO. Unfortunately, scMatch does not support the creation of a custom reference set, and therefore we were unable to perform this experiment with scMatch. To enable a comparison between scMatch, SingleR, and CellO, we project the outputs of scMatch and SingleR onto the Cell Ontology in order to evaluate scMatch and SingleR within the hierarchical classification framework. Specifically, for a given cell annotated by one of these methods with some cell type C, we also annotate the cell with all ancestors of C according to the Cell Ontology.

First, we note that CellO's training set included over 50% more of the cell types in this test set compared to those of scMatch and SingleR. In fact, most cell types in this test set, such as pancreatic islet cell types, are absent from scMatch and SingleR's reference sets, thus indicating that a user would be required to supply their own reference set for annotating these cell types (**Figure 5A**). We thus evaluated each method on only cell types that exist in each respective method's prepackaged reference set and found that CellO outperformed existing approaches (**Figure 5B**). SingleR performed poorly with CellO's training set, which may be due to the high number of samples in CellO's training set, the high number genes (i.e. including non-coding genes), and the fact that CellO contains bulk RNA-seq samples at a higher level in the ontology (**Figure S4A**). We note that in this analysis, for a given method, we may remove from the analysis a specific cell type that is absent from a given method's training set (e.g. stomach epithelial cell), but we would keep an ancestral cell type term (e.g. epithelial cell) if the method's training set contains a sample labelled with this ancestral term (e.g. a sample labeled as intestinal epithelial cell).

Next, we evaluated CellO on FAC-sorted peripheral blood mononuclear cells from Zheng et al. (2017) that were sequenced with Chromium 10x. We selected this dataset because it is one of the few droplet-based datasets for which the cell type labels are determined phenotypically (via sorting) rather than computationally (via expression analysis). To reflect the size of typical single cell datasets, we subsampled 2,000 cells from each of the ten sorted cell types and aggregated these cells together creating a dataset consisting of 20,000 cells. We first compared IR and CLR to the 1NN baseline and again found that IR and CLR outperformed 1NN with respect to median F1-score, although the difference between IR/CLR and 1NN was smaller than in the comparison than on the aforementioned non-droplet-based scRNA-seq dataset (**Figure 6D**). Specifically, IR, CLR, and 1NN produced median F1-scores of 0.97, 0.96, and 0.95 respectively across all cell types.

Next, we compared CellO to scMatch and SingleR on this PBMC dataset. In this analysis, we also ran SingleR using an immune-specific reference set of purified immune cells from Monaco et al. (2019). The cell types in this dataset are better represented in scMatch and SingleR's respective reference sets, and thus, a comparison between CellO and these methods on this data better isolates performance differences between these methods. Like scMatch and SingleR, CellO struggled to accurately classify the T cell subtypes (**Figure 6A-C, S2C, S3, S4B**). Among the methods compared here, SingleR with the Monaco et al. reference set most accurately classified the T cell subtypes (**Figure S3**), though we note that this reference set is specialized for immune cell types whereas the other reference sets, including CellO's, are more broad. We also note that CellO produced high average precision scores on most cell types including many of these T cell subtypes. Again, this indicates that CellO's classifiers have learned to discriminate between these cell types; however, the threshold for calling these cell types may be non-optimal.

Inspection of performance on challenging and diseased samples

We examined CellO's classifications on three challenging datasets. Two of these datasets comprised subsets of the 7,366 cells from non-droplet-based assays and are challenging because they contained cells for which their combination of cell types was absent from CellO's training set. First, we examined CellO's accuracy on 1,978 healthy pancreatic islet cells from Segerstolpe et al. (2016), which includes cell types that are absent from the bulk RNA-seq training set, specifically, ductal cells, acinar cells, epsilon cells, and delta cells. We found that CellO was able to correctly annotate the acinar cells as glandular epithelial cells, which is an ancestral cell type to acinar cells in the Cell Ontology (**Figure 7A, S2A**). This highlights the advantage of classifying against a hierarchy in that it enables CellO to annotate cells with a term higher in the ontology DAG when it is unsure about a cell's more-specific cell type. We also note that a number of cells were uncharacterized in the original study due to not meeting a stringent quality control filter. CellO annotated many of these cells as pancreatic A cells (a.k.a. pancreatic alpha cells), which is plausible owing to both their close position to annotated A cells according to UMAP, which is known to preserve some level of global structure in high dimensional data (Becht et al. 2018), as well as the fact that A cells were found to be the most abundant endocrine cell type in Segerstolpe et al. (2016) of those that passed their stringent quality control filtering.

We also further examined CellO's classification on 1,977 fetal neural cells from La Manno et al. (2016). Although the bulk RNA-seq training data contains samples of both embryonic cells and cells of various neural cell types, it does not contain any sample labeled as *both* neural cell *and* embryonic cell. Despite this discrepancy, CellO was able to annotate these cells with reasonable cell type labels (**Figure 7B, S2B**). We note that the microglial cells were annotated as phagocytes, which are an ancestral term to microglial cells in the Cell Ontology. Similarly, CellO annotated the glial cells as neuron associated cells. These examples again highlight CellO's ability to annotate cells with a term higher in the ontology DAG when it is unsure about a cell's more-specific cell type.

Finally, we examined CellO's classification on eight lung adenocarcinoma tumor samples from Laughney et al. (2020) that were sequenced with 10x. This dataset provides a good, yet challenging test set for CellO due to a) the heterogeneity of constituent cell types, b) the fact that it contains cell types absent from CellO's training set, and c) these cells originate from diseased tissue thereby providing insight into how CellO will perform on non-healthy samples. We compared CellO's output to the annotations provided by the authors, which were the result of a custom, *in silico* cell type annotation pipeline.

Overall, we found a high correspondence between the labels provided by CellO and the authors' annotation pipeline (**Figure 8, S5-S12**). When the two methods differed, we attempted to determine which method is likely to be correct using manual inspection of known marker genes and in many of these discrepancies, CellO produced the correct cell type labels. For example, in Tumor LX675 and LX682, CellO correctly annotates putative endothelial cells (**Figure 8, S5**), based on expression of PECAM1 (**Figure 8C**) and CD34 (data not shown), whereas the authors of these data labelled these as epithelial cells. In tumor LX675, CellO labelled the myeloid dendritic cell population as CD1c+ myeloid dendritic cells, a more specific cell type than that provided by the authors (**Figure 8**). In tumor LX682, CellO correctly annotates the putative fibroblasts as evidenced by their expression of FAP (Puré and Blomberg 2018) (**Figure S5**) and S100A4 (Strutz et al. 1995; data not shown). In tumor LX679, CellO appears to correctly annotate the plasmacytoid dendritic cell population as evidenced by the expression of IL3RA (**Figure S6**), a known marker for this cell type (Collin, McGovern, and Haniffa 2013). We also note that for cells whose likely true cell types are absent from CellO's training set, CellO was able to label these cells using a correct, but more general cell type. For example, in tumor LX675, mast cells were absent from CellO's training set, yet CellO accurately labelled these cells as hematopoietic cells (**Figure 8**). Similarly, when CellO was unsure of the labels for cells of cell types for which training data was sparse, such as for plasma cells and pericytes, CellO labelled these cells as more general cell type terms lymphocyte of B lineage and connective tissue cell, respectively (**Figure 8**). Lastly, we note some of the most clear-cut cases in which the authors' pipeline produced correct labels, but CellO erred. For example, in tumor LX679, CellO labelled many of the epithelial cell types as prostate epithelial cells (**Figure S6**), which is clearly incorrect given that these are lung tumor samples. Other errors produced by CellO are due to rarer cell types within the sample clustering together with a more common, similar cell type. For example, in tumor LX682, the myeloid dendritic cells are labelled by CellO as macrophages due to these cells clustering together with the macrophage population (**Figure S5**).

Evaluation of robustness to clustering

In order to test CellO's robustness to cluster granularity when clustering single-cell data, we tested CellO on the Zheng et al. PBMC dataset and Laughney et al. lung cancer dataset using differing values for Leiden's resolution parameter. On the Zheng et al. dataset, we tested CellO with five values for the resolution parameter and found both CLR and IR to perform similarly across these settings (**Figure S15**). In contrast to the Zheng et al. dataset, the lack of robust ground-truth labels for the Laughney et al. dataset made it difficult to perform a similar assessment. Instead, on this dataset, we ran CellO using the default resolution of 1.0 as well as a higher resolution of 8.0 and then manually inspected the differences in the outputs produced by these two settings (**Figure S5-S12**). Overall, we found that the higher resolution led CellO to classify clusters with more granularity, but at the cost of some errors. For example, on tumor LX679 and LX682, CellO was able to correctly classify the putative myeloid dendritic cells at the resolution of 8.0 whereas at the default resolution of 1.0, these cells were labelled incorrectly as alveolar macrophages (**Figure S5, S6**). This is likely due to the fact that the dendritic cells were subsumed into the macrophage cluster at the lower resolution. However, in other instances, the higher resolution led to errors. These errors are likely due to the fact that at a more granular clustering resolution, each cluster's mean expression value remained sparse due to the averaging of fewer cells. For example, in tumor LX679, cells correctly labelled as respiratory epithelial cells at a resolution of 1.0 were incorrectly labelled as prostate epithelial cells at a resolution of 8.0 (**Figure S6**).

User-friendly software

We provide a Python package for running CellO, using either IR or CLR, on a user-provided gene expression matrix (https://github.com/deweylab/CellO). CellO reduces the burden of reformatting and preprocessing an input expression matrix by accepting a variety of input file formats, including comma or tab-separated text files and HDF5, and by accepting expression data in a variety of units including counts or transcripts per million (TPM). To address the scenario in which the input dataset's genes do not match those expected by the pre-trained classifiers, we provide functionality for a user to re-train the models on the bulk RNA-seq training set with a custom gene set. On a personal laptop, training a new classifier took 31 minutes to train IR and 11 minutes to train CLR on the full set of 58,243 GENCODE genes. Training time is reduced when trained on a smaller set of genes (e.g. only protein-coding genes). We also note that the time required for CellO to perform classification is low because of the fact that it uses pre-trained logistic regression classifiers operating on cluster-averaged expression profiles. On the Zheng et al. (2020) dataset, CellO took six minutes to run on a personal laptop (including time for clustering) whereas scMatch required six hours and nine minutes (run with five cores), and SingleR required between nine and 22 minutes depending on the reference set used. Finally, we note that relative performance of IR and CLR varies across cell types. To guide a user on their selection of either IR or CLR, we provide the average precision values achieved by both methods on each cell type in the bulk RNA-seq validation set (**Table S1**). We also provide average precision values and F1-scores achieved by both methods on each cell type on the test set of 4,936 non-droplet-based single-cells whose cell types were present in the bulk RNA-seq training set (**Table S2**).

Lastly, we note that due to CellO's comprehensive training set that comprises cell types from many organs and tissues, some of CellO's errors are due to CellO annotating cells using a cell type that is unique to a tissue type that differs from the known tissue type of the target sample. For example, in some lung cancer tumors from Laughny et al., endothelial cells were classified as "endothelial cell of umbilical vein" (**Figure S5, S6, S9**), which is clearly incorrect given that these samples were taken from the lung. This error is likely due to the fact that the endothelial cells in CellO's training set largely originate from umbilical cord samples. Because these errors can be easily caught by the user, the CellO package enables a user to fix such errors by enabling the user to supply a blacklist of tissue types that do not pertain to the target sample. CellO then uses edges between the Cell Ontology and the Uberon ontology (which encodes anatomical entities; Mungall et al. 2012) to filter out cell types from CellO's output that are uniquely located in the blacklisted tissue types. For example, by blacklisting the Uberon term "umbilical vein", CellO correctly classifies the endothelial cells from Laughney et al.

Interpretability of models

CellO makes extensive use of linear models, which are particularly amenable to interpretation especially when the coefficients are sparse (Gleicher 2013). Although CellO's models are not regularized to be sparse (as in Gleicher 2013), we sparsify them by selecting the top-ten genes per cell type according to the magnitude of the coefficients associated with each gene within each cell type's one-vs.-rest binary classification model, which is used for CellO's IR classifier. To enable their interpretation, we present a web-based tool, the CellO Viewer, for exploring these discriminative genes uncovered by the models (https://uwgraphics.github.io/CellOViewer/). The tool supports two modes of operation: a *cell-centric* mode (**Figure 9A**) and a *gene-centric* mode (**Figure 9B**). In the cell-centric mode the user can select cell types via a graphical display of the Cell Ontology in order to view and compare the most important genes for distinguishing those cell types. In the gene-centric view, the user can select genes and explore which cell types these genes are most important for distinguishing from the remaining cell types. The CellO Viewer uses an interactive display of the Cell Ontology's graph to enable the user to navigate between cell types across the ontology.

We found that across diverse cell types, many known cell type-specific marker genes were recovered by the CellO models and are presented by the CellO Viewer. For example, CD3D, CD3E, and CD3G, which are canonical markers for T cells, were all present within the top ten genes ranked according to the magnitude of their coefficients within the logistic binary regression model used for distinguishing T cells from all other cell types. Similarly, CD4 and CD8 were present in the top genes for the CD4+ T cell and CD8+ T cell models, respectively (**Figure 89**). In a more complex example, the genes GCG, LOXL4, DPP4, GC, and FAP, known markers for pancreatic alpha cells, and INS, IAPP, and ADCYAP1, known markers for pancreatic beta cells Segerstolpe et al. (2016), all appear within the top ten genes for their respective cell types.

Interestingly, certain genes appear in the top ten coefficients for broad cell types, but not more specific cell types, indicating that CellO is able to find signals specific to broad cell type categories. For example, DDX4 appeared in the top ten genes for distinguishing germ line cells, but did not appear within the top ten genes for any of the more specific germ cell subtypes. DDX4 is known to be expressed in germ cells across both sexes (Hickford et al. 2011). Similarly, the gene NRG1 appeared in the top ten genes for distinguishing precursor cells, but did not appear within the top ten genes for any of the more specific cell types that are descendents of precursor cells within the ontology. NRG1 is known to play a role in the development of a number of organ systems (Lemmens, Doggen, and De Keulenaer 2007; Mei and Xiong 2008).

Discussion

In this work, we explore the application of hierarchical classification algorithms to cell type classification with the Cell Ontology using a novel, well-curated set of human primary cell RNA-seq samples. This dataset may prove useful for future investigations of cell type expression patterns or for use in cell type deconvolution methods (Aran et al. 2019; Newman et al. 2015). We demonstrate that the trained classifiers perform well across cell types in diverse single-cell datasets and outperformed existing cell type annotation methods when trained on their comprehensive reference sets. We packaged these classifiers into an easy-to-run Python package called CellO.

In our exploration of methods for correcting the independent one-vs.-rest classifiers, we found that discriminative methods outperformed the generative BNC approach implemented in URSA (Lee et al. 2013). We hypothesize that BNC suffers in comparison due to two causes. First, BNC's probabilistic model makes strong assumptions regarding the generative process of classifier scores and true cell type assignments. Second, BNC requires estimating the conditional probability distribution of each classifier's output scores (i.e. distance from the decision boundary) conditioned on the true cell type labels, which may be difficult to estimate accurately given the limited quantity of training data available for each cell type.

By using linear models, CellO's trained parameters are easily interpreted as cell type specific signatures across the ontology. However, we note that since certain cell types undergo similar sorting and preparation procedures (e.g., fluorescence activated cell sorting), it remains unclear to what extent these procedures affect gene expression and thus confound cell type. We sought to mitigate this effect by using data from a diversity of studies. We also note that the CLR algorithm may help to further mitigate this effect, since the binary classifiers trained in this framework for each cell type condition on the sample belonging to the parent cell types. Thus, for a given cell type, if samples of its parent cell types were prepared through similar procedures, the learned model parameters for that cell type will better capture biological cell type signatures.

There are a number of avenues that require further investigation. First, future work will entail curating comprehensive training sets from the SRA for other species such as mouse. This work will rely partly on future inclusion of standardized mouse metadata in the MetaSRA.

Second, CellO is a cluster-based annotation method and thus, its accuracy relies, in part, on the robustness of the clustering algorithm. If the clustering is too coarse, rare cell types may be missed. If clustering is too fine, the algorithm may not be combining enough data to accurately annotate each cluster. Determining the optimal clustering in scRNA-seq data is a challenging, open problem that will require further investigation (Kiselev, Andrews, and Hemberg 2019). Nonetheless, we demonstrated that CellO accurately classified a number of diverse datasets using Lieden's default parameter in the Scanpy package.

Third, we note that calibrating discriminative models trained on bulk RNA-seq data and applying them to single-cell data is challenging. In this work, we developed techniques for closing the gap between the performance of CellO when evaluated with average precision versus when evaluated with F1-score. The very high average-precision scores across many cell types indicates CellO is learning an accurate representation of these cell types, and that with better calibration, CellO's accuracy when making binary yes-no decisions for each cell type could be improved. Future work will investigate alternative approaches to calibrating CellO's models in order to improve CellO's binary cell type decisions.

Fourth, the Cell Ontology encodes anatomical and functional relationships between cell types; however, there exist a number of other relationships between cell types that could be utilized to improve accuracy. Such examples include lineage-based relationships (i.e. one cell type derives from another cell type) or evolutionary relationships between extant cell types and ancient cell types (Arendt et al. 2016). For example, the evolutionary relationships between cell types may be utilized to address inconsistencies in the independent classifiers approach that arise when certain cell types share a parent cell type via the currently encoded "is a" relationship, but are purported to have divergent evolutionary origins.

Fifth and finally, we expect the performance of hierarchical classifiers to improve as both more data is collected and as the Cell Ontology is expanded. Most importantly, we expect the calibration of the classifiers to improve as more training data becomes available for each cell type. More training data will be collected both as data is continually added to the SRA and as improvements are made to the SRA's metadata thereby allowing retrieval of previously undiscovered primary cell samples.

Acknowledgements

M.N. Bernstein thanks Gary H. Bernstein, John Steill, Ron Stewart, and Christina Kendziorski for helpful conversations.

Funding

This project has been made possible in part by grant U54 AI117924 from the National Institutes of Health and grant 2018-182626 from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation. M.N. Bernstein acknowledges support of the Computation and Informatics in Biology and Medicine Training Program funded by NLM grant: NLM 5T15LM007359. M.N. Bernstein also acknowledges support from the Morgridge Institute for Research. M.G. acknowledges support from the National Science Foundation award 1841349.

Author Contributions

M.N.B. led the conceptual development of the ideas present in this article and implemented both the CellO software and experiments. C.N.D. supervised the conceptual development and implementation of the software and experiments. M.N.B. and C.N.D. wrote the manuscript. Z.M. implemented the CellO Viewer. M.G. supervised the development of the CellO Viewer. M.N.B., Z.M., M.G., and C.N.D. contributed to the conceptual design of the CellO Viewer.

Figure Legends

Figure 1. **Overview of CellO.** (**A**) A schematic overview of CellO's hierarchical classification approach. CellO performs hierarchical classification with the Cell Ontology. Given a gene expression profile, CellO annotates the cell with a set of cell types (grey nodes) that are consistent with the hierarchical structure of the Cell Ontology. (**B**) We compare CellO to eight recent cell type annotation methods regarding the criteria we surmise are desirable in a cell type classification approach: whether the method (1) arrives pre-trained and can run out-of-the-box, (2) incorporates a hierarchy of cell types, (3) specifically uses the Cell Ontology as its hierarchy, (4) requires cell type-specific marker genes, (5) uses a model that can be interrogated to better understand how it arrived at its decision, and (6) whether the method operates on clusters or single-cells. We compare CellO to scMatch (Hou, Denisenko, and Forrest 2019), SingleR (Aran et al. 2019), scCatch (Shao et al. 2020), CHETAH (de Kanter et al. 2019), Garnett (Pliner, Shendure, and Trapnell 2019), CellAssign (A. W. Zhang et al. 2019), ACTINN (Ma and Pellegrini 2020), scPred (Alquicira-Hernandez et al. 2019), CaSTLe (Lieberman, Rokach, and Shay 2018), and SingleCellNet (Tan and Cahan 2019). CellO meets more desirable criteria than existing methods. (**C**) Euler diagrams of the cell types within the bulk RNA-seq expression profiles used to train CellO. This training set comprises most primary cell bulk RNA-seq samples within the SRA and consists of diverse cell types spanning various tissues, developmental stages, and stages of differentiation. These diagrams were created with nVenn (Pérez-Silva, Araujo-Voces, and Quesada 2018).

Figure 2. **Overview of analyses and CellO's algorithm.** (**A**) A schematic illustration of the datasets and analyses performed in this study. Initial candidate bulk RNA-seq samples were selected from the SRA via the MetaSRA, filtered for errors, and quantified, which resulted in a comprehensive bulk RNA-seq training set consisting of healthy human primary cells. This

training set was split into a pre-training and validation set for tuning the parameters of the binary classifiers as well as for evaluating the graph-correction methods (Transparent Methods). The full bulk RNA-seq dataset was then used to train the final models that were then evaluated on three sets of scRNA-seq data. The first set consisted of an aggregation of diverse non-droplet-based datasets from the SRA. The second dataset consisted of FAC-sorted PBMCs from Zheng et al. (2017). The third set consisted of primary lung tumor cells from Laughney et al. (2020). (**B**) A schematic illustration of CellO's classification procedure. First, for a given sample, the raw classifier probabilities are corrected with the cell ontology using IR (if CLR is used, this step is not necessary). We illustrate one edge of the graph whose incident nodes have probabilities that are logically inconsistent with the hierarchy, and thus require correction, because the child node has a higher probability than the parent. Once corrected, cell types whose raw probabilities meet their respective decision-threshold are selected. Among these, the most-specific cell types (i.e. lowest in the ontology) are examined and the cell type with the highest output probability is selected. CellO outputs this final selected cell type along with all ancestor terms.

Figure 3. **Reconciling the outputs of independent classifiers with a hierarchy.** (**A**) Average precision scores across all cell types for the independent classifiers (Ind.), as well as for IR, TPR, and BNC on the validation set. (**B**) Each paired sample and cell type prediction was considered independently. The set of all such predictions was ordered according to their prediction probability and the corresponding precision-recall curve was constructed for the independent classifiers, IR, TPR, and BNC.

Figure 4. **Results on non-droplet-based single-cell data.** CellO's performance on the 4,936 non-droplet-based cells considering only cells whose cell types are present in the bulk RNA-seq training set. We compare the distributions of (**A**) F1-score (**B**) precision, and (**C**) average-precision across all such cell types. (**D**) The subgraph spanning the non-droplet-based cells where each cell type is colored according to CellO's (IR) F1-score (top) as well as by average-precision (bottom).

Figure 5. **Comparison of CellO to existing approaches on non-droplet-based single-cell data.** Evaluating CellO, SingleR, and scMatch on the non-droplet-based cells. (**A**) The fraction of cell types in the single-cell test dataset that are also present in each method's training set. IR and CLR are not shown separately because they share the same training set. We evaluate SingleR's built in reference sets from the Human Primary Cell Atlas (HPCA) and BluePrint+ENCODE (BE). (**B**) The distribution of both F1-scores (left) and precisions (right) for only those cell types that are in each method's training set. We compare CellO to scMatch, SingleR with the Human Primary Cell Atlas (HPCA), and SingleR with the Blueprint+ENCODE reference (BE). Note, each distribution evaluates different sets of cell types depending on the particular subset of cell types present in each method's training set.

Figure 6. **Results on 10x PBMC data.** (**A**) The subgraph of the Cell Ontology spanning the 10x PBMC dataset from Zheng et al. (2017). Each cell type is colored according to CellO's (IR) F1-score as well as (**B**) average-precision. (**C**) UMAP plots of the single-cell dataset where cells are colored by their true cell type (top) as well as the most-specific predicted cell type (i.e. lowest in the ontology) as output by CellO (bottom). (**D**) Boxplots displaying the distribution of F1-scores across all cell types for IR, CLR, 1NN, scMatch, SingleR with the Human Primary Cell Atlas (HPCA), SingleR with the Blueprint+ENCODE reference (BE), and SingleR with the

Monaco et al. reference (M).

Figure 7. **Examination of CellO's performance on difficult datasets.** (**A**) UMAP plots of all healthy cells in Segerstolpe et al. (2016) including cells for which their specific cell types are not present in Cello's bulk RNA-seq training set. Cells are colored according to their true cell type (left) and (IR) predicted cell type (right). Highlighted are CellO's predictions made on pancreatic acinar cells (top-right ovals) as well as a subset of uncharacterized pancreatic epithelial cells predicted as A cells (center ovals). (**B**) UMAP plots of human, embryonic neural cells from La Manno et al. (2016). Cells are colored according to their true cell type (left) and predicted cell type (right). Highlighted are CellO's predictions made on both the microglial and glial cells and note that CellO annotates these cells using terms that are higher in the ontology's graph than their true terms.

Figure 8. **Examination of CellO on diseased cells.** (**A**) UMAP plots of lung adenocarcinoma tumor LX675 from Laughney et al. (2020) colored by CellO's output using IR and a Leiden resolution parameter of 1.0 (left) and the original cell type labels provided by the authors. We highlight four subpopulations comprising putative CD1C+ myeloid dendritic cells (top left), endothelial cells (top right), plasma cells (bottom left), and mast cells (bottom right). (**B**) The legend for coloring cells in panel (A) and (B). (**C**) For UMAP plots of cells colored by their expression, in units log(TPM+1), of CD1C, a marker for CDC1+ myeloid dendritic cells, PECAM1, a marker for endothelial cells, SDC1, a marker for plasma cells, and KIT, a marker for mast cells.

Figure 9. **The CellO Viewer.** Screenshots of the CellO Viewer web application for enabling the exploration of cell type-specific expression signatures across the Cell Ontology. (**A**) Comparing the top-ten genes between CD4+ T cells and CD8+ T cells (red nodes in the Graph View) ranked by the magnitude of their coefficients in their corresponding models. Genes that are shared between the two lists are highlighted with the same color. The CellO Viewer displays genes whose expressions are both positively correlated (green) and negatively correlated (red) with the selected cell types. (**B**) A screenshot of the gene-centric mode of the CellO Viewer with GFAP, an astrocyte marker, selected. For a given selected gene, the CellO Viewer will display the cell types within the DAG (top) and in list-form (bottom) for which the selected gene appears within the top-ten genes ranked by each model's coefficients.

Transparent Methods

Methods Details

Data curation

To create the training set of primary cell bulk RNA-seq samples, we first selected all samples labelled as a “primary cell” sample by the MetaSRA (v1.4). Thus, we followed the conservative definition for a primary cell sample by Bernstein et al. (2017), which requires that a sample has not undergone passaging beyond the first culture. We followed this selection with a manual curation of each sample’s technical variables by consulting sources of metadata that are not captured by the MetaSRA annotation process, such as fields in Gene Expression Omnibus (Clough and Barrett 2016) records and each study's publication. In total, we annotated 27,097 samples (available to download at http://deweylab.biostat.wisc.edu/cell_type_classification/). We then removed all samples that were either incorrectly labelled as primary cell samples or had been experimentally treated. When found, we also corrected errors in the MetaSRA-provided Cell Ontology labels by both adding additional cell types that were missed by the MetaSRA as well as removing incorrect cell type labels. This curation effort resulted in the final set of 4,293 samples.

Data processing

We quantified the gene expression of all RNA-seq samples from the SRA (both bulk and non-droplet-based single-cell samples) with kallisto (v0.43.1) (Bray et al. 2016) using human genome release GRCh38 with GENCODE annotation version 27. We chose kallisto for gene expression quantification in order to prioritize processing speed on this large dataset, figuring that any small loss in accuracy (at the gene level) relative to a less approximate, but slower approach would not be significant for the cell type classification task. This produced estimated counts for 200,401 isoform-level genomic features. We summed the transcripts per million (TPM) values by gene to produce TPM’s for 58,243 gene-level features. The curated metadata and associated quantified samples are available to download at http://deweylab.biostat.wisc.edu/cell_type_classification.

Notation

In the following descriptions of the methods used in this work, we let $\boldsymbol{x} \in \mathbb{R}^G$ denote a gene expression profile, where G is the number of considered genes and x is measured in units of log(TPM+1) where TPM are transcripts per million. We let n denote the number of samples, m denote the number of considered cell types, $y_{i,j} \in \{0,1\}$ denote the cell type assignment for cell type $j \in [m]$and sample $i \in [n]$.

Binary classification with logistic regression

We use L2-penalized logistic regression for each binary classifier as implemented by scikit-learn (Pedregosa et al. 2011) using the LIBLINEAR solver (Fan et al. 2008). To speed up the training of each binary classifier, we preprocessed the bulk RNA-seq training data using principal components analysis (PCA) as implemented in scikit-learn. Specifically for each cell type j, each classifier is trained by minimizing the following loss-function:

$$\min_{\boldsymbol{\beta}_j, c_j} \frac{1}{2}||\boldsymbol{\beta_j}||_2^2 + C\sum_{i=1}^{n} w_{i,j}\left[-y_{i,j}\log(h_{\boldsymbol{\beta}_j,c_j}(\mathbf{x}_i)) - (1-y_{i,j})\log(1-h_{\boldsymbol{\beta}_j,c_j}(\mathbf{x}_i))\right]$$

where

$$h_{\boldsymbol{\beta}_j,c_j}(\mathbf{x}_i) := \left[1+\exp((\mathbf{U}\mathbf{x}_i)^T\boldsymbol{\beta}_j + c_j)\right]^{-1}$$

is the logistic function, $\boldsymbol{\beta}_j \in \mathbb{R}^k$ are the model-coefficients for cell type j, c_j is the intercept for cell type j, C controls the strength of the regularization, n is the number of training samples, $w_{i,j}$ is a per-sample weight for handling class-imbalance in cell type j's model, and $\mathbf{U} \in \mathbb{R}^{k \times m}$ is the PCA loadings matrix.

We also note that most training sets are highly unbalanced. We found that the models were better calibrated when the loss-function weighted each training sample such that the positive and negative samples contribute equally to the loss function (via the $w_{i,j}$ weights above) as evidenced by the improved F1-scores when using a threshold of 0.5 for making discrete yes-no cell type decisions (**Figure S13C**). We implemented this class-balancing by setting the *class_weight* parameter to 'balanced' in scikit-learn's *LogisticRegression* class constructor.

Isotonic regression correction

We train a binary classifier for each cell type $j \in [m]$ to model $p(y_j \mid \boldsymbol{x})$ using logistic regression and a one-versus-rest training strategy. As proposed by Obozinski *et al.* (2008), these probabilities are then reconciled with the ontology graph using isotonic regression. Specifically, we output the set of probabilities:

$$p_1, \ldots, p_m := \arg\ \min_{p'_1, \ldots, p'_m} \sum_{i=1}^{m} (p'_j - \hat{p}_j)^2$$

subject to

$$\forall j \in [m], \forall k \in \mathrm{Par}(j),\ p_j \leq p_k \text{ and } \forall j \in [m],\ 0 \leq p_j \leq 1$$

where $\forall j \in [m], \hat{p}_j := p(y_j = 1 \mid \boldsymbol{x})$ as output by each classifier and $\mathrm{Par}(j)$ is the set of parent cell types for cell type j. CellO uses the quadprog Python package (https://pypi.org/project/quadprog/) for solving this quadratic optimization problem.

Bayesian network correction

A binary classifier is trained for each cell type and a one-versus-rest training strategy. The classifier outputs are then reconciled with the ontology graph using a Bayesian network as proposed by Lee et al. (2013). The true assignments for each cell type within a given sample, denoted $y_1, \ldots, y_m$, are modelled as latent random variables, and the classifier outputs, denoted $f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$ (signed distances to each decision boundary), are modelled as observed random variables in a Bayesian network. The final output probability for cell type j is then the marginal probability

$$p(y_j = 1 \mid f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$$

More specifically, for a given cell type j, we model the conditional distribution of the classifier's output $f_j(\mathbf{x})$ (distance to the decision boundary) conditional on its true label y_j as a discrete random variable constructed as follows. We partition the training data for cell type j into two folds ensuring that no study is split between folds while attempting to keep the sizes of the two folds as similar as possible. We then train on one fold and compute the classifier scores from the second fold (for each of the two folds).

Using all of these scores, we then compute a histogram where the bin sizes are determined using a second 2-fold cross-validation scheme. Specifically, we then test a number of bin sizes by first estimating a histogram density function using data in one fold and then computing the likelihood of the data in the second fold (performing this procedure for both folds). The histogram density function is given by

$$f(x) := \frac{1}{nh}\text{Count(x)}$$

where n is the total number of data points, h is the width of each bin, and $Count(x)$ is the number of data points sharing the same bin as x. We choose a bin size that maximizes the mean of the two data log likelihoods computed on each fold.

As described by Lee et al. (2013), the true cell type assignments $p(y_1, ..., y_m)$ factor according to the ontology graph:

$$p(y_1, \dots, y_m) := \prod_{j=1}^{m} p(y_j \mid \text{Children}_j)$$

where

$$\text{Children}_j := \{y_k \mid k \;\text{ is a child of }\; j\}$$

are the assignments to the children cell types of cell type j in the ontology. These conditional distributions $p(y_j|\text{Children}_j)$ enforce consistency with the ontology by defining

$$p(y_j = 1 \mid \text{Children}_j) := 1.0$$

if $1 \in \text{Children}_j$. Otherwise,

$$p(y_j = 1 \mid \text{Children}_j) := \text{prior}_j$$

The prior_j values are computed from counts in the training data. Specifically, the prior for each leaf-node cell type c is simply the fraction of samples in the data set labelled as c. For each internal node c, the prior is computed as the fraction of all samples labelled as c, but not labelled as any child of c. A pseudocount of one was used in the calculation for all priors. Due to the size of the ontology, we perform approximate inference using Gibbs sampling rather than exact inference using the Laurintzen algorithm as was performed by Lee et al.

To test the impact of the graph-structured prior on this algorithm, we also tested a naive Bayes variant in which each cell type is predicted independently (i.e., without the graph-structured prior). We found a significant improvement in performance for the graph-structured BNC algorithm over the naive Bayes algorithm indicating that the graph structured prior is an important component of this algorithm (**Figure S14**).

True Path Rule

We train a binary classifier for each cell type $j \in [m]$ to model $p(y_j \mid \boldsymbol{x})$ using logistic regression and a one-versus-rest training strategy. As proposed by Notaro et al. (2017), this method involves two passes across the ontology: on a bottom-up pass, each cell type's output probability is averaged with the outputs of all child cell types classifiers for which the classifier makes a positive prediction according to a predefined threshold. More specifically, each cell type j's output probability is set to

$$\bar{p}_j := \frac{1}{|C_j| + 1}\left(\hat{p}_j + \sum_{k \in C_j} \bar{p}_k\right)$$

where $\hat{p}_j := p(y_j = 1 \mid \boldsymbol{x})$ according to the classifier and

$$C_j := \{k \in \text{Children}(j) \ : \ \hat{p}_j > t\}$$

is the set of children of cell type j for which the classifier output a positive prediction according to a predefined threshold t. We used a threshold of $t = 0.5$. This bottom-up pass allows for sharing of information across the classifiers. In the top-down pass of the ontology, the output probabilities are set to ensure consistency with the ontology. This procedure works as follows: each node in the ontology is visited according to the topologically sorted order of nodes and for a given visited cell type j, its final probability is set to

$$p_j := \begin{cases} \bar{p}_j & \text{if } j \text{ is a root node} \\ \min_{k \in \text{Par}(j)} p_k & \text{if } \min_{k \in \text{Par}(j)} p_k < \bar{p}_j \\ \bar{p}_j & \text{otherwise} \end{cases}$$

where $Par(j)$ are the parent nodes of node j in the DAG.

Cascaded logistic regression

Classification is made in a top-down fashion starting from the root of the ontology downward as proposed by Obozinski et al. (2008). This is accomplished by training a logistic regression, binary classifier (although any binary classifier that outputs a prediction probability can be used) for each cell type $j \in [m]$ to model the distribution

$$q_j := p(y_j = 1 \mid \pi_j = 1, \boldsymbol{x})$$

where $\pi_j \in \{0, 1\}$ indicates whether the sample belongs to all of the parents of j in the ontology. In order to model these distributions, each cell type's negative training examples consist of those samples that are labeled with all parent cell types, but not the target cell type. Given these learned distributions, the probability that x originates from cell type j is computed via

$$p(y_j = 1 \mid \boldsymbol{x}) = q_j \prod_{k \in A_j} q_k$$

where A_j denotes the ancestors of cell type j in the ontology's DAG.

One-nearest neighbors

Given a query gene expression profile $\boldsymbol{x}$, we return all cell type labels belonging to the training set expression profile

$$\arg\min_{\boldsymbol{x}' \in \boldsymbol{X}} 1 - \mathrm{Corr}(\boldsymbol{x}, \boldsymbol{x}')$$

where $\mathrm{Corr}(\boldsymbol{x}, \boldsymbol{x}')$ is the Pearson correlation of the expression values in $\boldsymbol{x}$ and $\boldsymbol{x}'$ as implemented in Python's SciPy package (Virtanen et al. 2020).

Quantification and statistical analysis

Partitioning bulk RNA-seq data into training and test sets

In order to find the optimal parameters and configurations for CellO, we partitioned the bulk RNA-seq training dataset into a pre-training set and validation set (**Figure 2A**). When creating this partition, we sought to satisfy a number of criteria that would enable unbiased estimation of performance across cell types. First, we required that no study be split between the pre-training and validation sets in order to ensure that a model is never tested on data from a study on which it was trained. This mitigates the possibility that the algorithm will provide an overly optimistic estimate of the generalization error when run on the validation set. Second, we sought an approximately 80/20 split of the data between the pre-training and validation sets. Third and finally, we sought for all cell types to be represented in both the pre-training and validation sets. We framed this partitioning task as an optimization problem where our four criteria were encoded in an objective function. Minimizing this objective function entails creating a partition that most closely meets the aforementioned four criteria. To optimize the objective function, we performed a simple hill-climbing procedure where we moved a study's data from the validation set to the pre-training set if such a move resulted in a partition that improved the objective function.

Parameter tuning

To choose the optimal number of principal components, we evaluated various numbers of principal components by training the collection of one-vs.-rest classifiers on the bulk RNA-seq pre-training set and evaluating on the bulk RNA-seq validation set. We found that using 3,000 principal components performed equally well to using the raw gene expression values as features. Further, we found performance degraded as the number of principal components decreased (**Figure S13A**). We then performed a parameter sweep of penalty-weight parameters (i.e. regularization-strength parameters) for IR and CLR and chose the values that maximized the median average precisions for each method (**Figure S13D**).

Model calibration

We found that for many cell types, there were relatively few studies in the training set for which samples of these cell types originated and thus, we found that the models tended to overfit to the training set leading to poorly calibrated models. This was evidenced by the fact that the average-precision scores were high across most cell types; however, the F1-scores were much lower when using a default threshold of 0.5 for making binary cell type decisions (**Figure S13D**). The high average precision scores indicate that the positive and negative examples of each cell type are well separated in the ranking of samples when ranked according to the classifier output probability for the given cell type. However, the poor F1-scores indicate that the default threshold of 0.5 is non-optimal for separation.

To address this issue, we used a data-driven approach to find empirical thresholds according to a leave-study-out cross-validation experiment. Specifically, we performed leave-study-out cross-validation on the pre-training set of bulk RNA-seq samples and used these results to empirically choose a threshold for each cell type such that if a given threshold less than 0.5 led to a higher F1-score than the default threshold of 0.5, we select this empirical threshold. When applying these data-driven thresholds to the validation set, we observed a significant increase in the F1-scores across cell types under the independent one-vs.-rest classifiers approach (**Figure S13B**). Thus, when training the final CellO models (both CLR and IR), we performed this same cross-validation procedure on the entire bulk RNA-seq training set to select data-driven thresholds.

We note that even with these selected thresholds, CellO often output more than one specific cell type for a given sample (e.g. both natural killer cell and T cell), which may confuse a user of the tool. This phenomenon is a consequence of both mis-calibrated models (despite the data-driven thresholding procedure) and because CellO performs multi-label hierarchical classification. To address this issue, whenever CellO outputs more than one specific cell type, we select only the cell type with highest output probability along with all ancestor cell types. All F1-scores reported in the main text followed from this correction procedure and thus, are an apt measure of the practical utility of CellO on real data (**Figure 2B**).

Model interpretation

We note that although the logistic regression model's coefficients $\boldsymbol{\beta}_j \in \mathbb{R}^k$ (where j denotes the index for a given cell type) weight the k principal components rather than genes, each gene's contribution to the model's decision can be recovered by

$$\mathbf{w}_j := \mathbf{U}\boldsymbol{\beta}_j$$

where $\mathbf{U} \in \mathbb{R}^{m \times k}$ is the PCA loading matrix and $\mathbf{w}_{j,g}$ describes the contribution of gene g to the model's decision. The full set of these vectors over all cell types can be explored within the CellO Viewer web application.

Evaluation metrics

For a given input dataset, CellO generates two sets of outputs: binary yes-no decisions for each cell type assignment as well as probability scores that quantify how likely each cell should be assigned to any given cell type. Both of these outputs can be represented as matrices. More specifically, given an input dataset, $\mathbf{X} \in \mathbb{R}^{n \times G}$ where n is the number of cells and G is the number of genes, the binary yes-no decisions can be represented as a matrix $B \in \{0,1\}^{n \times m}$ where m is the number of cell types and $B_{i,j} = 1$ if the classifier predicts cell i to be of cell type j . The probability scores can also be represented as a matrix $S \in [0,1]^{n \times m}$ where $S_{i,j}$ denotes the classifier's confidence that cell type i should be labelled as cell type j. Finally, the true cell type assignments can also be represented as a matrix $T \in \{0,1\}^{n \times m}$ where $T_{i,j} = 1$ if cell i is truly of cell type j.

We first define metrics for comparing B to T. For cell type j, we define the number of true positives (TP), false positives (FP), and false negatives (FN) as:

$$TP := |\{i \mid T_{i,j} = 1 \wedge B_{i,j} = 1\}|$$
$$FP := |\{i \mid T_{i,j} = 0 \wedge B_{i,j} = 1\}|$$
$$FN := |\{i \mid T_{t,j} = 1 \wedge B_{i,j} = 0\}|$$

We then evaluate the classifier's performance on cell type j using precision, recall, and F1-score, which are defined as

$$\text{precision} := \frac{TP}{TP + FP}$$
$$\text{recall} := \frac{TP}{TP + FN}$$
$$\text{F1-score} := \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

We note that for a given cell, the ground-truth assignments for cell types that are more-specific than the cell's most-specific ground-truth cell type are ambiguous since these samples could, in theory, truly be of a more specific cell type than they are labelled with (e.g., a cell labelled as a T cell could be a CD8+ T cell even if it isn't annotated as such). Thus, when computing the aforementioned metrics for a given cell type within these single cell datasets, we exclude those cells that are labelled most-specifically as an ancestor of the cell type. For example, for metrics calculated for CD8+ T cells, we would exclude from the calculations those cells that are most-specifically labelled as T cells.

Resource Availability

Lead Contact

Further information and requests for resources should be directed to and will be fulfilled by the Lead Contact, Colin Dewey (colin.dewey@wisc.edu).

Data and Code Availability

A Python package for running CellO can be found at https://github.com/deweylab/CellO. The data used in this work can be found at http://deweylab.biostat.wisc.edu/cell_type_classification/. The CellO Viewer can be accessed at https://uwgraphics.github.io/CellOViewer/. The code implementing the CellO Viewer can be found at https://github.com/uwgraphics/CellOViewer. All code for performing the experiments in this work can be found at https://github.com/deweylab/cell-type-classification-paper.

Materials Availability Statement Examples
This study did not generate new unique reagents.

References

Abdelaal, Tamim, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel J. T. Reinders, and Ahmed Mahfouz. 2019. “A Comparison of Automatic Cell Identification Methods for Single-Cell RNA Sequencing Data.” *Genome Biology* 20 (1): 194.

Alquicira-Hernandez, Jose, Anuja Sathe, Hanlee P. Ji, Quan Nguyen, and Joseph E. Powell. 2019. “scPred: Accurate Supervised Method for Cell-Type Classification from Single-Cell RNA-Seq Data.” *Genome Biology* 20 (1): 264.

Aran, Dvir, Zicheng Hu, and Atul J. Butte. 2017. “xCell: Digitally Portraying the Tissue Cellular Heterogeneity Landscape.” *Genome Biology* 18 (1): 220.

Aran, Dvir, Agnieszka P. Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, et al. 2019. “Reference-Based Analysis of Lung Single-Cell Sequencing Reveals a Transitional Profibrotic Macrophage.” *Nature Immunology* 20 (2): 163–72.

Arendt, Detlev, Jacob M. Musser, Clare V. H. Baker, Aviv Bergman, Connie Cepko, Douglas H. Erwin, Mihaela Pavlicev, et al. 2016. “The Origin and Evolution of Cell Types.” *Nature Reviews. Genetics* 17 (12): 744–57.

Bard, Jonathan, Seung Y. Rhee, and Michael Ashburner. 2005. “An Ontology for Cell Types.” *Genome Biol*. https://doi.org/10.1186/gb-2005-6-2-r21.

Becht, Etienne, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. 2018. “Dimensionality Reduction for Visualizing Single-Cell Data Using UMAP.” *Nature Biotechnology*, December. https://doi.org/10.1038/nbt.4314.

Bernstein, Matthew N., Anhai Doan, and Colin N. Dewey. 2017. “MetaSRA: Normalized Human Sample-Specific Metadata for the Sequence Read Archive.” *Bioinformatics* 33 (18): 2914–23.

Bray, Nicolas L., Harold Pimentel, Páll Melsted, and Lior Pachter. 2016. “Near-Optimal Probabilistic RNA-Seq Quantification.” *Nature Biotechnology* 34 (5): 525–27.

Clough, Emily, and Tanya Barrett. 2016. “The Gene Expression Omnibus Database.” *Methods in Molecular Biology* 1418: 93–110.

Collin, Matthew, Naomi McGovern, and Muzlifah Haniffa. 2013. “Human Dendritic Cell Subsets.” *Immunology* 140 (1): 22–30.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. “LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research.” *Journal of Machine Learning Research: JMLR* 9: 1871–74.

Fernández, José María, Victor de la Torre, David Richardson, Romina Royo, Montserrat Puiggròs, Valentí Moncunill, Stamatina Fragkogianni, et al. 2016. “The BLUEPRINT Data Analysis Portal.” *Cell Systems* 3 (5): 491–95.e5.

Gleicher, Michael. 2013. “Explainers: Expert Explorations with Crafted Projections.” *IEEE Transactions on Visualization and Computer Graphics* 19 (12): 2042–51.

Gremse, Marion, Antje Chang, Ida Schomburg, Andreas Grote, Maurice Scheer, Christian Ebeling, and Dietmar Schomburg. 2011. “The BRENDA Tissue Ontology (BTO): The First All-Integrating Ontology of All Organisms for Enzyme Sources.” *Nucleic Acids Research* 39 (Database issue): D507–13.

Hickford, Danielle E., Stephen Frankenberg, Andrew J. Pask, Geoff Shaw, and Marilyn B.

Renfree. 2011. “DDX4 (VASA) Is Conserved in Germ Cell Development in Marsupials and Monotremes.” *Biology of Reproduction* 85 (4): 733–43.

Hou, Rui, Elena Denisenko, and Alistair R. R. Forrest. 2019. “scMatch: A Single-Cell Gene Expression Profile Annotation Tool Using Reference Datasets.” *Bioinformatics* 35 (22): 4688–95.

Jaitin, Diego Adhemar, Ephraim Kenigsberg, Hadas Keren-Shaul, Naama Elefant, Franziska Paul, Irina Zaretsky, Alexander Mildner, et al. 2014. “Massively Parallel Single-Cell RNA-Seq for Marker-Free Decomposition of Tissues into Cell Types.” *Science* 343 (6172): 776–79.

Kanter, Jurrian K. de, Philip Lijnzaad, Tito Candelli, Thanasis Margaritis, and Frank C. P. Holstege. 2019. “CHETAH: A Selective, Hierarchical Cell Type Identification Method for Single-Cell RNA Sequencing.” *Nucleic Acids Research* 47 (16): e95.

Kiselev, Vladimir Yu, Tallulah S. Andrews, and Martin Hemberg. 2019. “Challenges in Unsupervised Clustering of Single-Cell RNA-Seq Data.” *Nature Reviews. Genetics* 20 (5): 273–82.

La Manno, Gioele, Daniel Gyllborg, Simone Codeluppi, Kaneyasu Nishimura, Carmen Salto, Amit Zeisel, Lars E. Borm, et al. 2016. “Molecular Diversity of Midbrain Development in Mouse, Human, and Stem Cells.” *Cell* 167 (2): 566–80.e19.

Laughney, Ashley M., Jing Hu, Nathaniel R. Campbell, Samuel F. Bakhoum, Manu Setty, Vincent-Philippe Lavallée, Yubin Xie, et al. 2020. “Regenerative Lineages and Immune-Mediated Pruning in Lung Cancer Metastasis.” *Nature Medicine* 26 (2): 259–69.

Lee, Young-Suk, Arjun Krishnan, Qian Zhu, and Olga G. Troyanskaya. 2013. “Ontology-Aware Classification of Tissue and Cell-Type Signals in Gene Expression Profiles across Platforms and Technologies.” *Bioinformatics* 29 (23): 3036–44.

Leinonen, Rasko, Hideaki Sugawara, Martin Shumway, and International Nucleotide Sequence Database Collaboration. 2011. “The Sequence Read Archive.” *Nucleic Acids Research* 39 (Database issue): D19–21.

Lemmens, Katrien, Kris Doggen, and Gilles W. De Keulenaer. 2007. “Role of Neuregulin-1/ErbB Signaling in Cardiovascular Physiology and Disease: Implications for Therapy of Heart Failure.” *Circulation* 116 (8): 954–60.

Lieberman, Yuval, Lior Rokach, and Tal Shay. 2018. “CaSTLe - Classification of Single Cells by Transfer Learning: Harnessing the Power of Publicly Available Single Cell RNA Sequencing Experiments to Annotate New Experiments.” *PloS One* 13 (10): e0205499.

Lizio, Marina, Jayson Harshbarger, Imad Abugessaisa, Shuei Noguchi, Atsushi Kondo, Jessica Severin, Chris Mungall, et al. 2017. “Update of the FANTOM Web Resource: High Resolution Transcriptome of Diverse Cell Types in Mammals.” *Nucleic Acids Research* 45 (D1): D737–43.

Mabbott, Neil A., J. Kenneth Baillie, Helen Brown, Tom C. Freeman, and David A. Hume. 2013. “An Expression Atlas of Human Primary Cells: Inference of Gene Function from Coexpression Networks.” *BMC Genomics* 14 (September): 632.

Ma, Feiyang, and Matteo Pellegrini. 2020. “ACTINN: Automated Identification of Cell Types in Single Cell RNA Sequencing.” *Bioinformatics* 36 (2): 533–38.

Mei, Lin, and Wen-Cheng Xiong. 2008. “Neuregulin 1 in Neural Development, Synaptic Plasticity and Schizophrenia.” *Nature Reviews. Neuroscience* 9 (6): 437–52.

Monaco, Gianni, Bernett Lee, Weili Xu, Seri Mustafah, You Yi Hwang, Christophe Carré, Nicolas Burdin, et al. 2019. “RNA-Seq Signatures Normalized by mRNA Abundance Allow Absolute Deconvolution of Human Immune Cell Types.” *Cell Reports* 26 (6): 1627–40.e7.

Mungall, Christopher J., Carlo Torniai, Georgios V. Gkoutos, Suzanna E. Lewis, and Melissa A. Haendel. 2012. “Uberon, an Integrative Multi-Species Anatomy Ontology.” *Genome Biology*

13 (1): R5.

Newman, Aaron M., Chih Long Liu, Michael R. Green, Andrew J. Gentles, Weiguo Feng, Yue Xu, Chuong D. Hoang, Maximilian Diehn, and Ash A. Alizadeh. 2015. “Robust Enumeration of Cell Subsets from Tissue Expression Profiles.” *Nature Methods* 12 (5): 453–57.

Notaro, Marco, Max Schubach, Peter N. Robinson, and Giorgio Valentini. 2017. “Prediction of Human Phenotype Ontology Terms by Means of Hierarchical Ensemble Methods.” *BMC Bioinformatics* 18 (1): 449.

Obozinski, Guillaume, Gert Lanckriet, Charles Grant, Michael I. Jordan, and William Stafford Noble. 2008. “Consistent Probabilistic Outputs for Protein Function Prediction.” *Genome Biology* 9 Suppl 1 (June): S6.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research: JMLR* 12: 2825–30.

Pérez-Silva, José G., Miguel Araujo-Voces, and Víctor Quesada. 2018. “nVenn: Generalized, Quasi-Proportional Venn and Euler Diagrams.” *Bioinformatics* 34 (13): 2322–24.

Picelli, Simone, Åsa K. Björklund, Omid R. Faridani, Sven Sagasser, Gösta Winberg, and Rickard Sandberg. 2013. “Smart-seq2 for Sensitive Full-Length Transcriptome Profiling in Single Cells.” *Nature Methods* 10 (11): 1096–98.

Pliner, Hannah A., Jay Shendure, and Cole Trapnell. 2019. “Supervised Classification Enables Rapid Annotation of Cell Atlases.” *Nature Methods* 16 (10): 983–86.

Puré, Ellen, and Rachel Blomberg. 2018. “Pro-Tumorigenic Roles of Fibroblast Activation Protein in Cancer: Back to the Basics.” *Oncogene* 37 (32): 4343–57.

Segerstolpe, Åsa, Athanasia Palasantza, Pernilla Eliasson, Eva-Marie Andersson, Anne-Christine Andréasson, Xiaoyan Sun, Simone Picelli, et al. 2016. “Single-Cell Transcriptome Profiling of Human Pancreatic Islets in Health and Type 2 Diabetes.” *Cell Metabolism* 24 (4): 593–607.

Shao, Xin, Jie Liao, Xiaoyan Lu, Rui Xue, Ni Ai, and Xiaohui Fan. 2020. “scCATCH: Automatic Annotation on Cell Types of Clusters from Single-Cell RNA Sequencing Data.” *iScience* 23 (3): 100882.

Sloan, Cricket A., Esther T. Chan, Jean M. Davidson, Venkat S. Malladi, J. Seth Strattan, Benjamin C. Hitz, Idan Gabdank, et al. 2016. “ENCODE Data at the ENCODE Portal.” *Nucleic Acids Research* 44 (D1): D726–32.

Strutz, F., H. Okada, C. W. Lo, T. Danoff, R. L. Carone, J. E. Tomaszewski, and E. G. Neilson. 1995. “Identification and Characterization of a Fibroblast Marker: FSP1.” *The Journal of Cell Biology* 130 (2): 393–405.

Tan, Yuqi, and Patrick Cahan. 2019. “SingleCellNet: A Computational Tool to Classify Single Cell RNA-Seq Data Across Platforms and Across Species.” *Cell Systems* 9 (2): 207–13.e2.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. “From Louvain to Leiden: Guaranteeing Well-Connected Communities.” *Scientific Reports* 9 (1): 5233.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” *Nature Methods* 17 (3): 261–72.

Wolf, F. Alexander, Philipp Angerer, and Fabian J. Theis. 2018. “SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis.” *Genome Biology* 19 (1): 15.

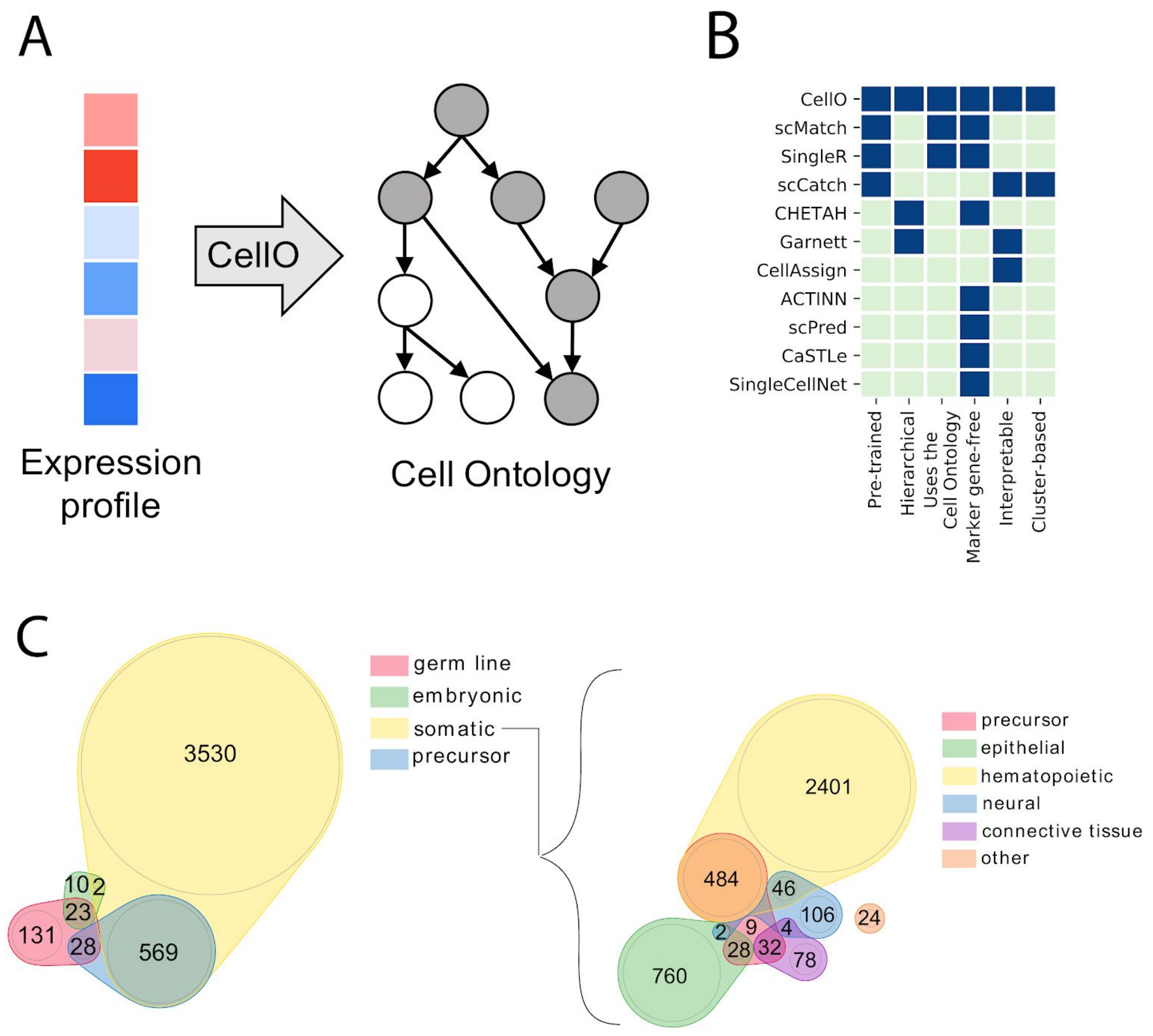
Zhang, Allen W., Ciara O’Flanagan, Elizabeth A. Chavez, Jamie L. P. Lim, Nicholas Ceglia, Andrew McPherson, Matt Wiens, et al. 2019. “Probabilistic Cell-Type Assignment of Single-Cell RNA-Seq for Tumor Microenvironment Profiling.” *Nature Methods* 16 (10): 1007–15.

Zhang, Xinxin, Yujia Lan, Jinyuan Xu, Fei Quan, Erjie Zhao, Chunyu Deng, Tao Luo, et al. 2019.

“CellMarker: A Manually Curated Resource of Cell Markers in Human and Mouse.” *Nucleic Acids Research* 47 (D1): D721–28.

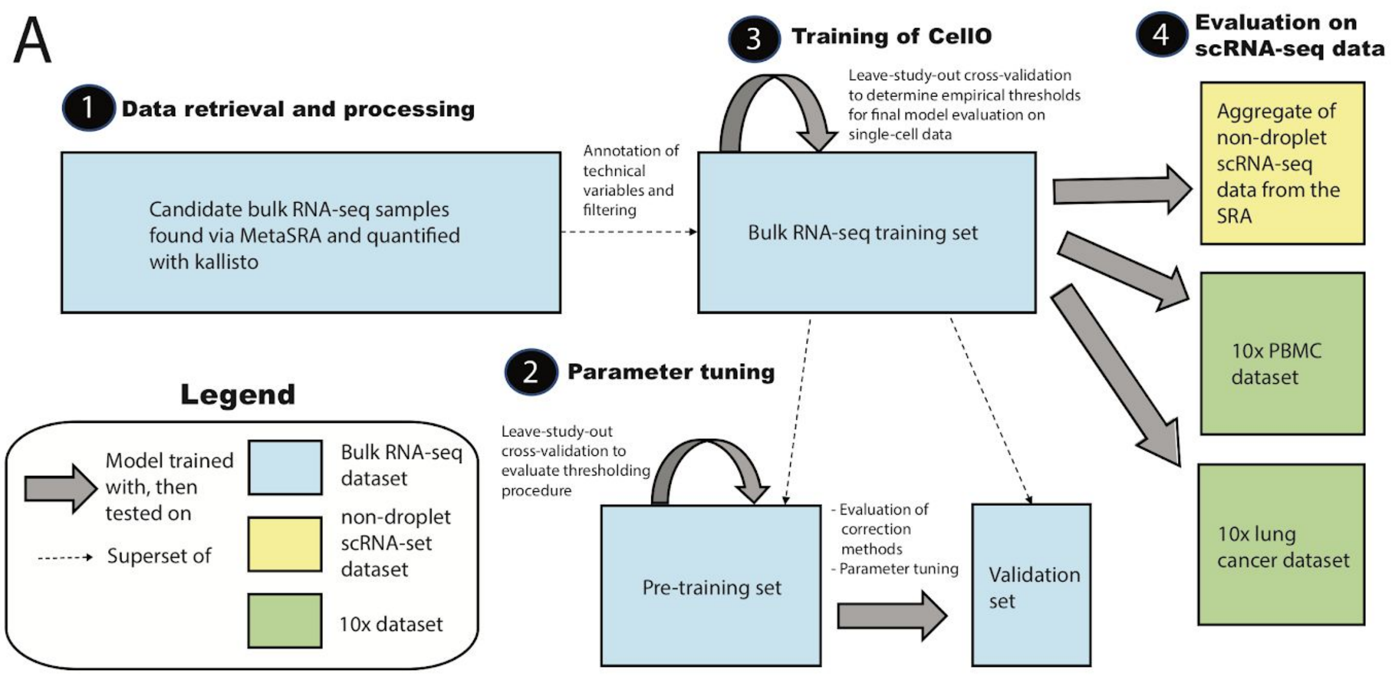
Zheng, Grace X. Y., Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, et al. 2017. “Massively Parallel Digital Transcriptional Profiling of Single Cells.” *Nature Communications* 8 (January): 14049.

Figure 1

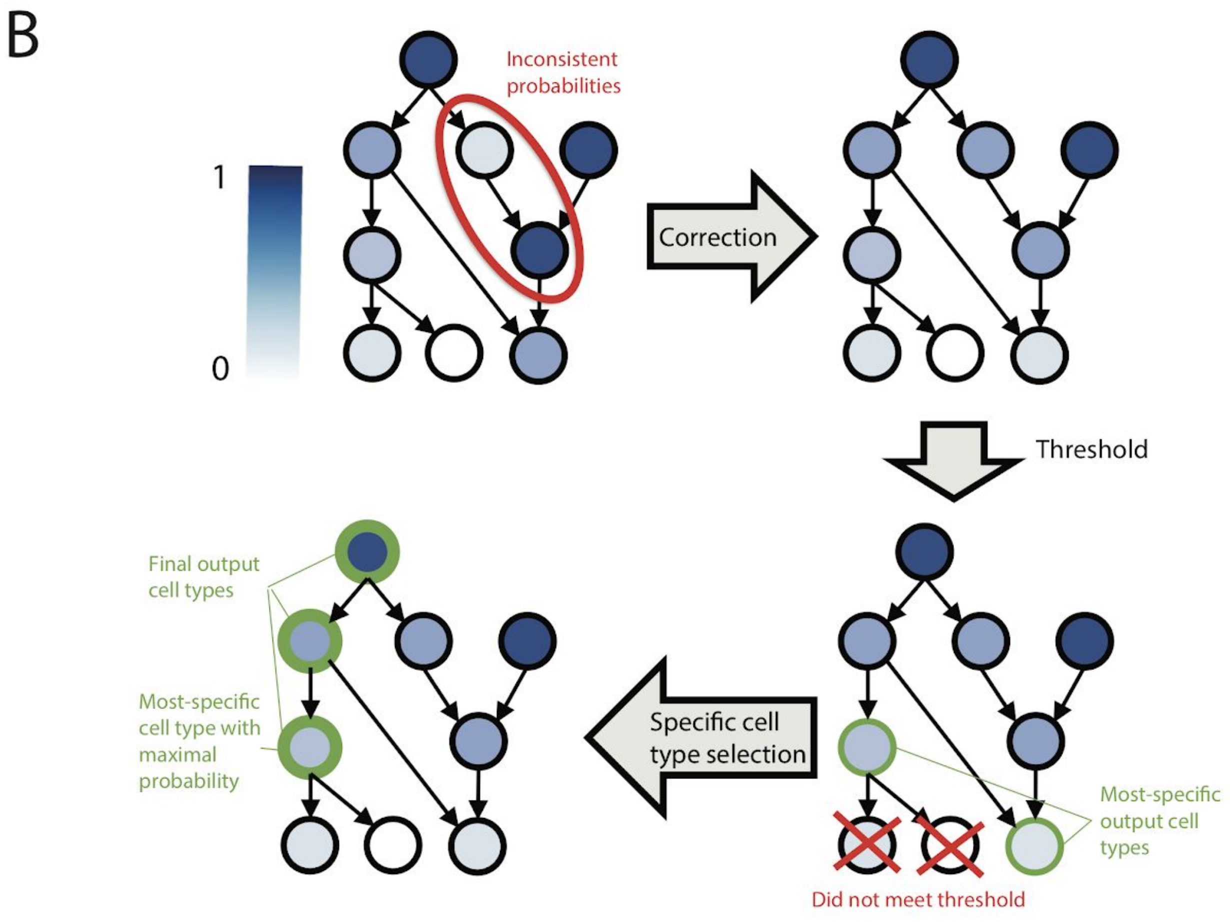


	Pre-trained	Hierarchical	Uses the Cell Ontology	Marker gene-free	Interpretable	Cluster-based
CellO	■	■	■	■	■	■
scMatch	■	□	■	■	□	□
SingleR	■	□	■	■	□	□
scCatch	■	□	□	□	■	■
CHETAH	□	■	□	■	□	□
Garnett	□	■	□	□	■	□
CellAssign	□	□	□	□	■	□
ACTINN	□	□	□	■	□	□
scPred	□	□	□	■	□	□
CaSTLe	□	□	□	■	□	□
SingleCellNet	□	□	□	■	□	□

Figure 2

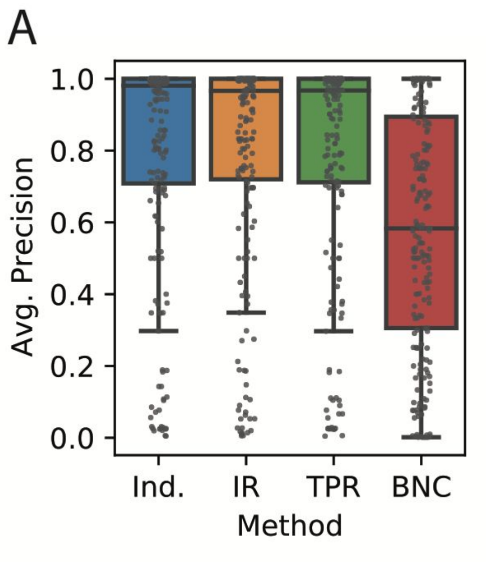


A
1 Data retrieval and processing
Candidate bulk RNA-seq samples found via MetaSRA and quantified with kallisto
Annotation of technical variables and filtering
3 Training of CellO
Leave-study-out cross-validation to determine empirical thresholds for final model evaluation on single-cell data
Bulk RNA-seq training set
4 Evaluation on scRNA-seq data
Aggregate of non-droplet scRNA-seq data from the SRA
10x PBMC dataset
10x lung cancer dataset
2 Parameter tuning
Leave-study-out cross-validation to evaluate thresholding procedure
Pre-training set
- Evaluation of correction methods
- Parameter tuning
Validation set
Legend
Model trained with, then tested on
Superset of
Bulk RNA-seq dataset
non-droplet scRNA-set dataset
10x dataset



B
Inconsistent probabilities
1
0
Correction
Threshold
Did not meet threshold
Most-specific output cell types
Specific cell type selection
Final output cell types
Most-specific cell type with maximal probability

Figure 3


A
1.0
0.8
0.6
0.4
0.2
0.0
Avg. Precision
Ind.
IR
TPR
BNC
Method

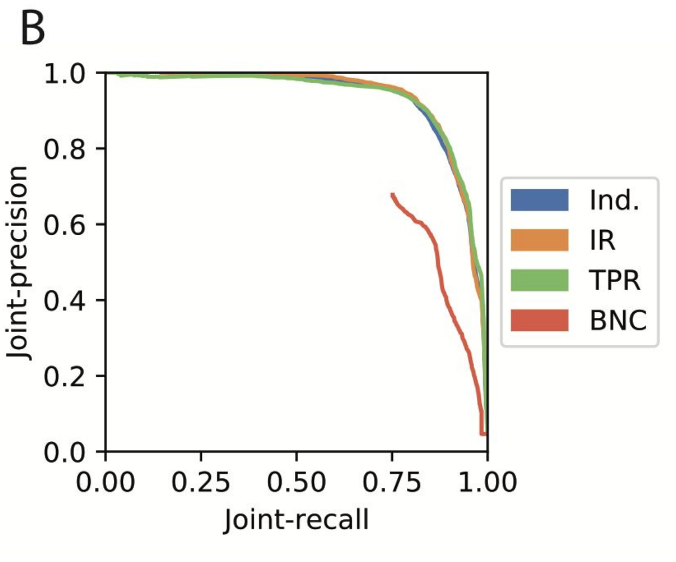
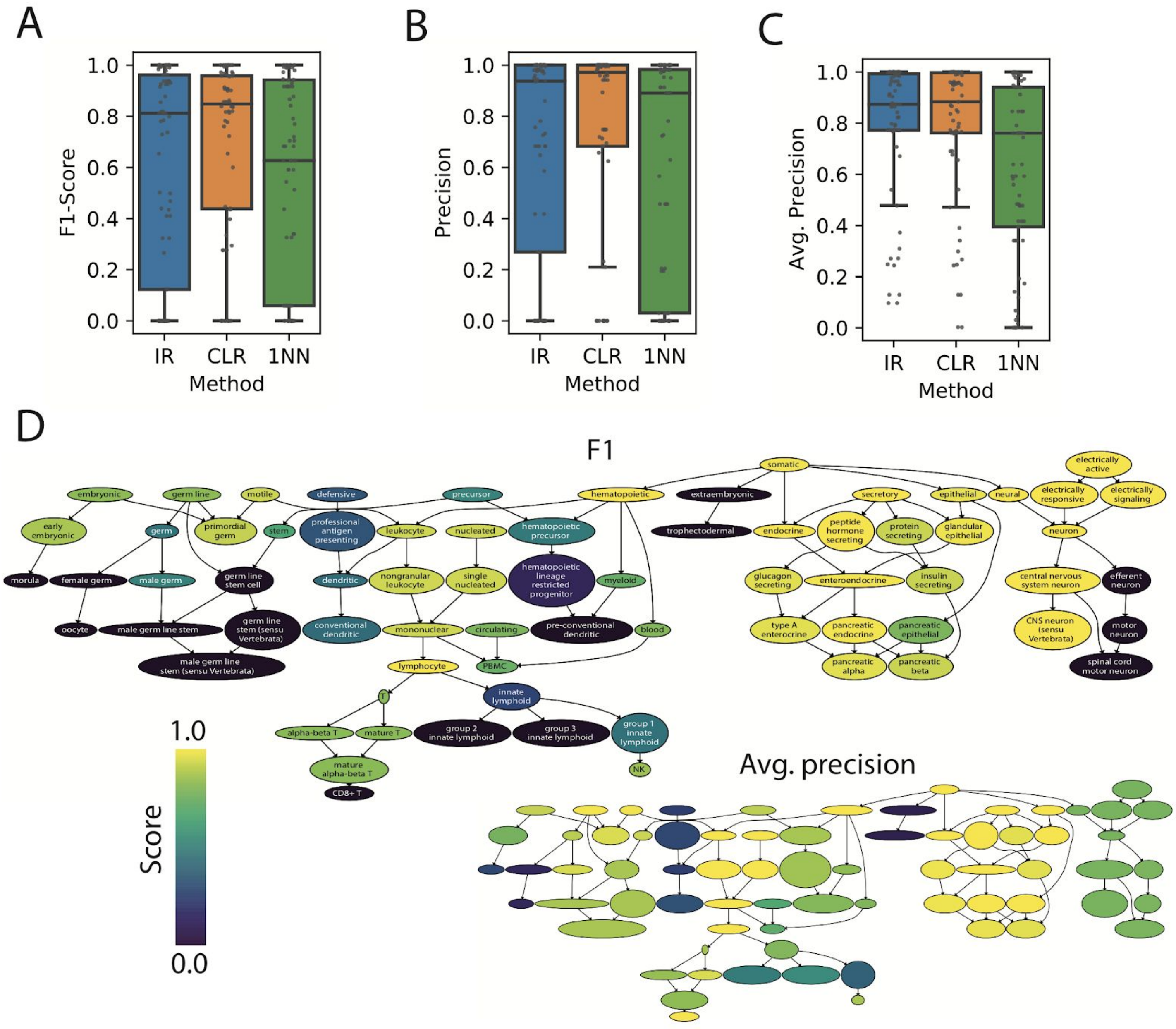
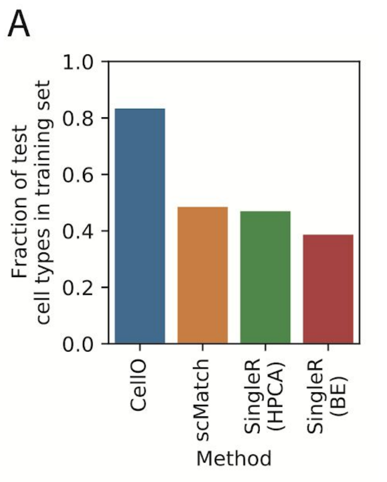

B
1.0
0.8
0.6
0.4
0.2
0.0
Joint-precision
0.00
0.25
0.50
0.75
1.00
Joint-recall
Ind.
IR
TPR
BNC

Figure 4



A
B
C
F1-Score
Precision
Avg. Precision
0.0
0.2
0.4
0.6
0.8
1.0
IR
CLR
1NN
Method
D
F1
Avg. precision
Score
1.0
0.0
somatic
electrically active
embryonic
germ line
motile
defensive
precursor
hematopoietic
extraembryonic
secretory
epithelial
neural
electrically responsive
electrically signaling
early embryonic
germ
primordial germ
stem
professional antigen presenting
leukocyte
nucleated
hematopoietic precursor
trophectodermal
endocrine
peptide hormone secreting
protein secreting
glandular epithelial
neuron
morula
female germ
male germ
germ line stem cell
dendritic
nongranular leukocyte
single nucleated
hematopoietic lineage restricted progenitor
myeloid
glucagon secreting
enteroendocrine
insulin secreting
central nervous system neuron
efferent neuron
oocyte
male germ line stem
germ line stem (sensu Vertebrata)
conventional dendritic
mononuclear
circulating
pre-conventional dendritic
blood
type A enterocrine
pancreatic endocrine
pancreatic epithelial
CNS neuron (sensu Vertebrata)
motor neuron
male germ line stem (sensu Vertebrata)
lymphocyte
PBMC
pancreatic alpha
pancreatic beta
spinal cord motor neuron
T
innate lymphoid
alpha-beta T
mature T
group 2 innate lymphoid
group 3 innate lymphoid
group 1 innate lymphoid
mature alpha-beta T
NK
CD8+ T

Figure 5


A
Fraction of test cell types in training set
1.0
0.8
0.6
0.4
0.2
0.0
CellO
scMatch
SingleR (HPCA)
SingleR (BE)
Method

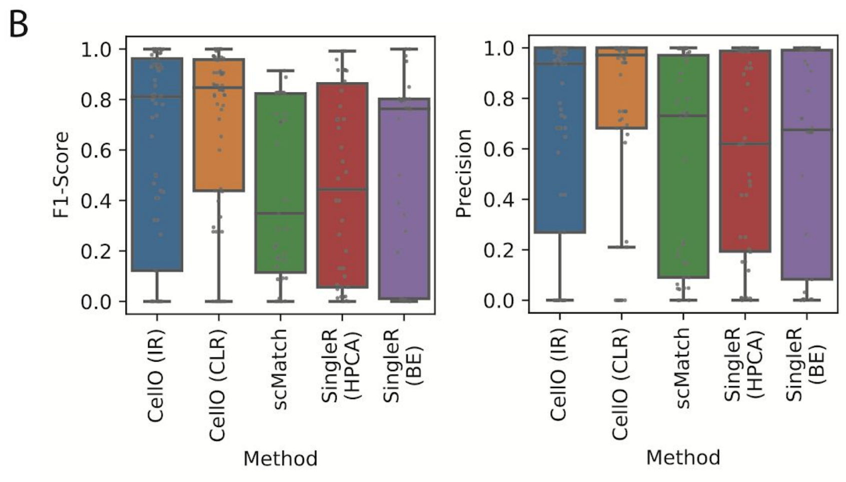

B
F1-Score
Precision
1.0
0.8
0.6
0.4
0.2
0.0
CellO (IR)
CellO (CLR)
scMatch
SingleR (HPCA)
SingleR (BE)
Method

Figure 6

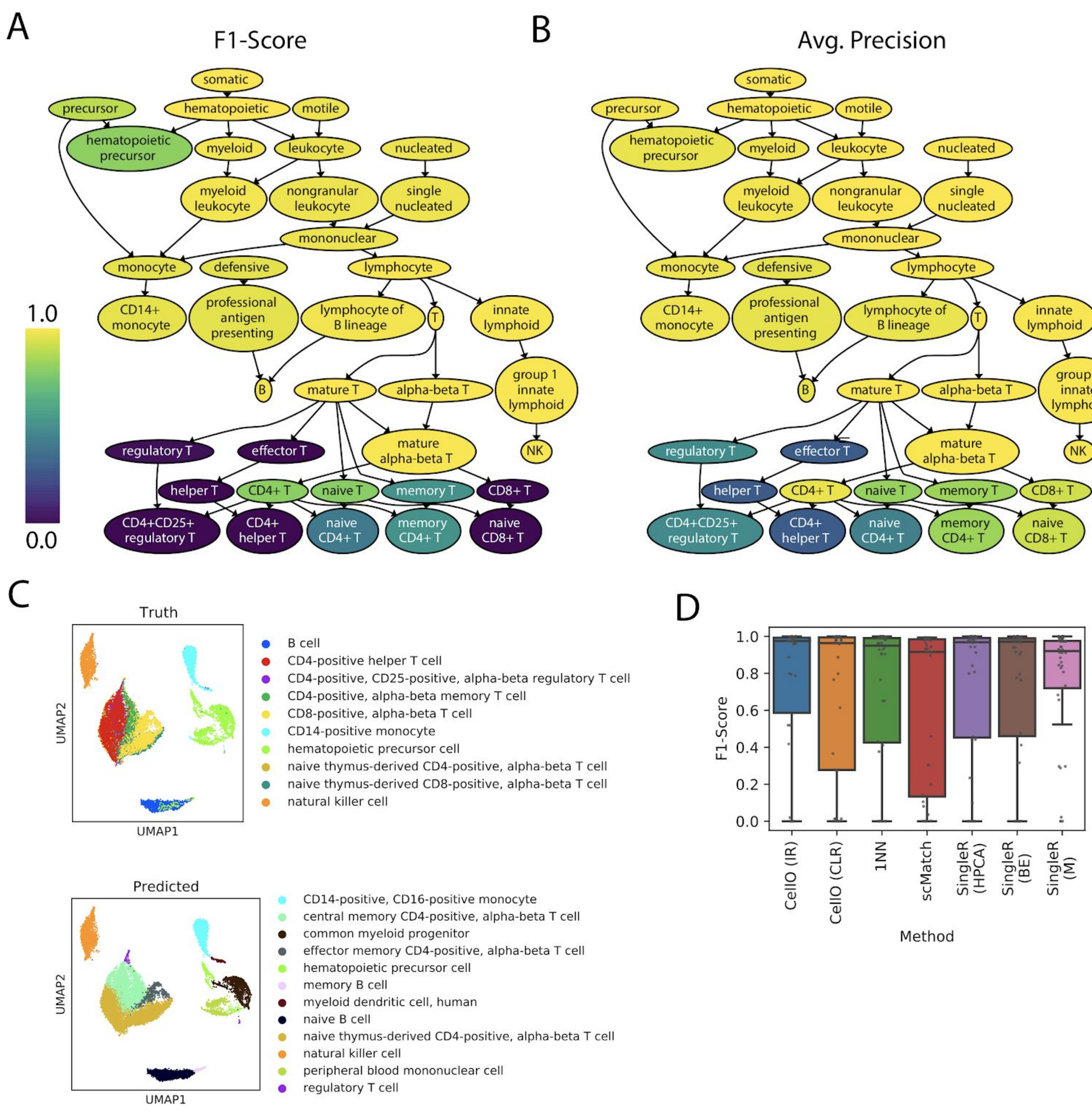
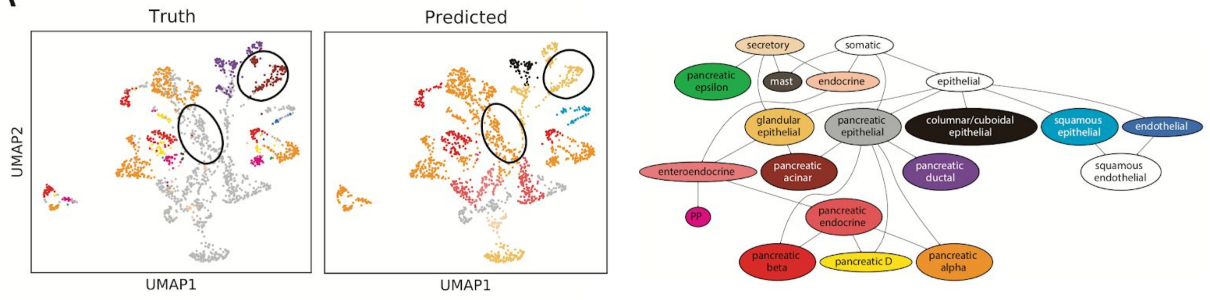

A
F1-Score
B
Avg. Precision
somatic
precursor
hematopoietic
motile
hematopoietic precursor
myeloid
leukocyte
nucleated
myeloid leukocyte
nongranular leukocyte
single nucleated
mononuclear
monocyte
defensive
lymphocyte
CD14+ monocyte
professional antigen presenting
lymphocyte of B lineage
T
innate lymphoid
B
mature T
alpha-beta T
group 1 innate lymphoid
regulatory T
effector T
mature alpha-beta T
NK
helper T
CD4+ T
naive T
memory T
CD8+ T
CD4+CD25+ regulatory T
CD4+ helper T
naive CD4+ T
memory CD4+ T
naive CD8+ T
1.0
0.0
C
Truth
UMAP2
UMAP1
B cell
CD4-positive helper T cell
CD4-positive, CD25-positive, alpha-beta regulatory T cell
CD4-positive, alpha-beta memory T cell
CD8-positive, alpha-beta T cell
CD14-positive monocyte
hematopoietic precursor cell
naive thymus-derived CD4-positive, alpha-beta T cell
naive thymus-derived CD8-positive, alpha-beta T cell
natural killer cell
Predicted
CD14-positive, CD16-positive monocyte
central memory CD4-positive, alpha-beta T cell
common myeloid progenitor
effector memory CD4-positive, alpha-beta T cell
hematopoietic precursor cell
memory B cell
myeloid dendritic cell, human
naive B cell
naive thymus-derived CD4-positive, alpha-beta T cell
natural killer cell
peripheral blood mononuclear cell
regulatory T cell
D
F1-Score
1.0
0.8
0.6
0.4
0.2
0.0
CellO (IR)
CellO (CLR)
1NN
scMatch
SingleR (HPCA)
SingleR (BE)
SingleR (M)
Method

Figure 7

A


Truth
Predicted
UMAP2
UMAP1
UMAP1
secretory
somatic
pancreatic epsilon
mast
endocrine
epithelial
glandular epithelial
pancreatic epithelial
columnar/cuboidal epithelial
squamous epithelial
endothelial
enteroendocrine
pancreatic acinar
pancreatic ductal
squamous endothelial
PP
pancreatic endocrine
pancreatic beta
pancreatic D
pancreatic alpha

B

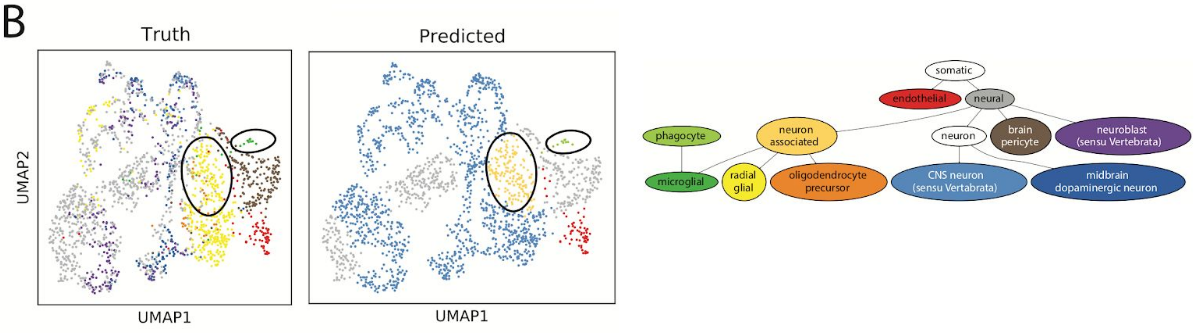

Truth
Predicted
UMAP2
UMAP1
UMAP1
somatic
endothelial
neural
phagocyte
neuron associated
neuron
brain pericyte
neuroblast (sensu Vertebrata)
microglial
radial glial
oligodendrocyte precursor
CNS neuron (sensu Vertebrata)
midbrain dopaminergic neuron

Figure 8

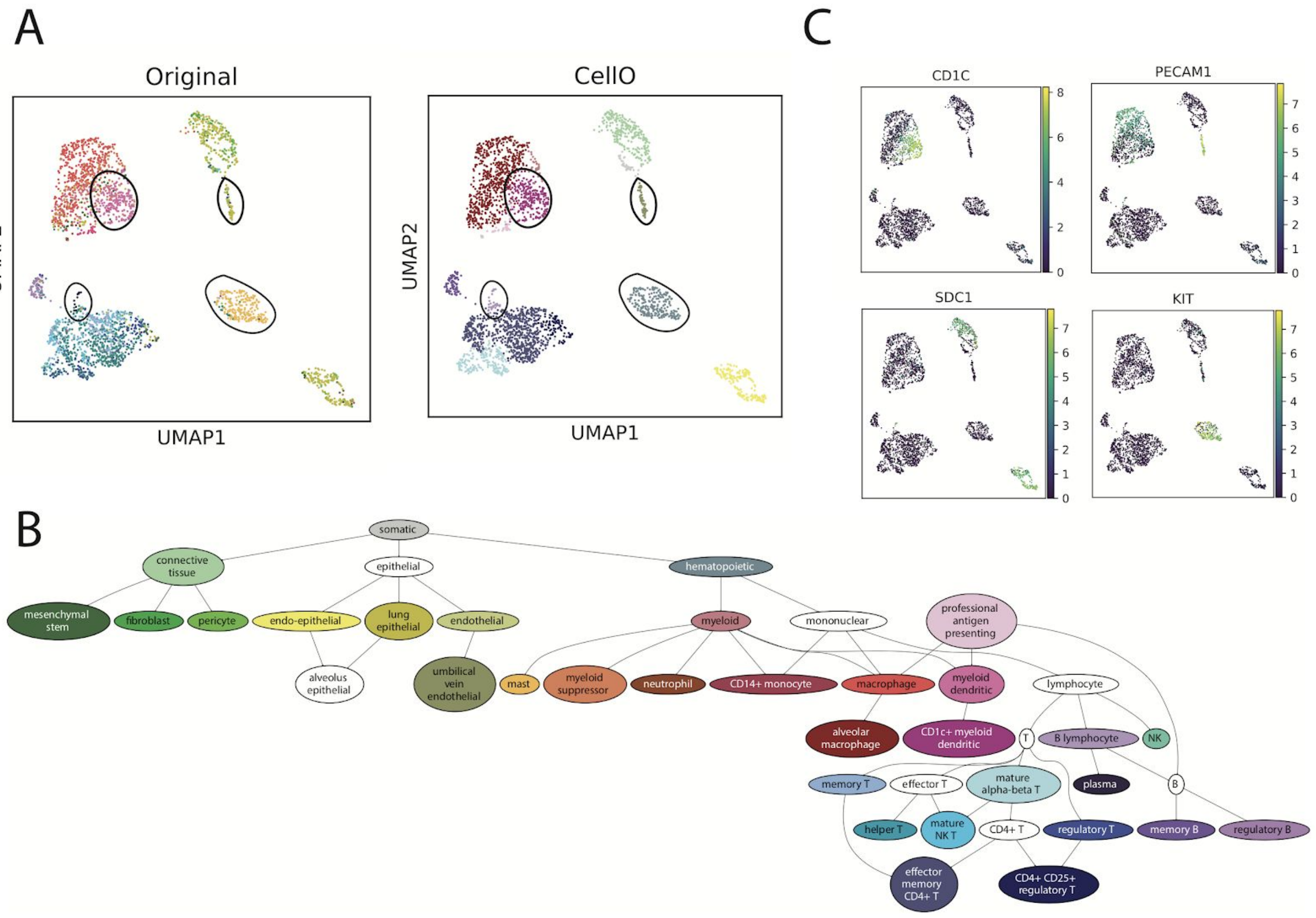
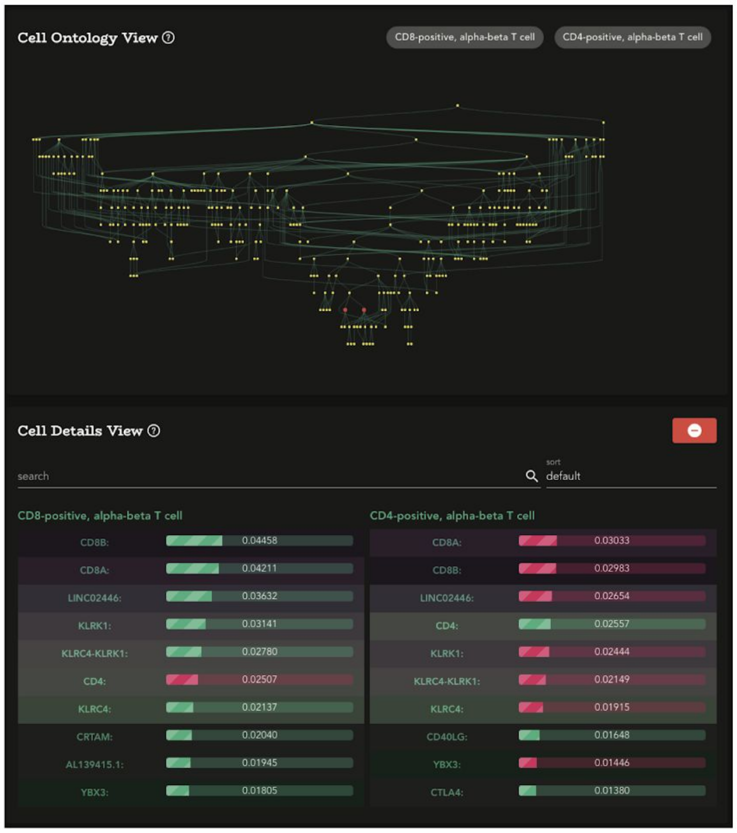

A
Original
CellO
UMAP1
UMAP2
C
CD1C
PECAM1
SDC1
KIT
B
somatic
connective tissue
epithelial
hematopoietic
mesenchymal stem
fibroblast
pericyte
endo-epithelial
lung epithelial
endothelial
myeloid
mononuclear
professional antigen presenting
alveolus epithelial
umbilical vein endothelial
mast
myeloid suppressor
neutrophil
CD14+ monocyte
macrophage
myeloid dendritic
lymphocyte
alveolar macrophage
CD1c+ myeloid dendritic
T
B lymphocyte
NK
memory T
effector T
mature alpha-beta T
plasma
B
helper T
mature NK T
CD4+ T
regulatory T
memory B
regulatory B
effector memory CD4+ T
CD4+ CD25+ regulatory T

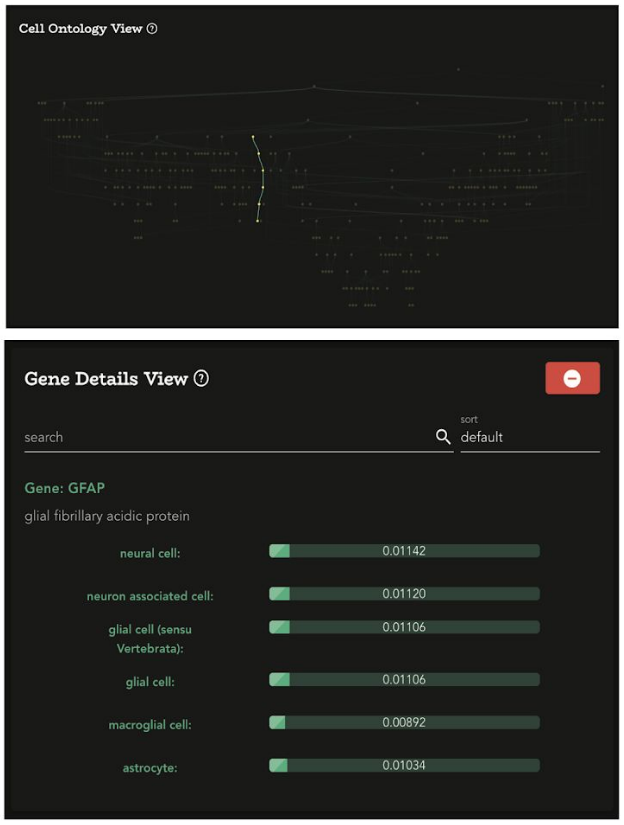
Figure 9

A


Cell Ontology View
CD8-positive, alpha-beta T cell
CD4-positive, alpha-beta T cell
Cell Details View
search
sort
default

CD8-positive, alpha-beta T cell		CD4-positive, alpha-beta T cell	
CD8B:	0.04458	CD8A:	0.03033
CD8A:	0.04211	CD8B:	0.02983
LINC02446:	0.03632	LINC02446:	0.02654
KLRK1:	0.03141	CD4:	0.02557
KLRC4-KLRK1:	0.02780	KLRK1:	0.02444
CD4:	0.02507	KLRC4-KLRK1:	0.02149
KLRC4:	0.02137	KLRC4:	0.01915
CRTAM:	0.02040	CD40LG:	0.01648
AL139415.1:	0.01945	YBX3:	0.01446
YBX3:	0.01805	CTLA4:	0.01380

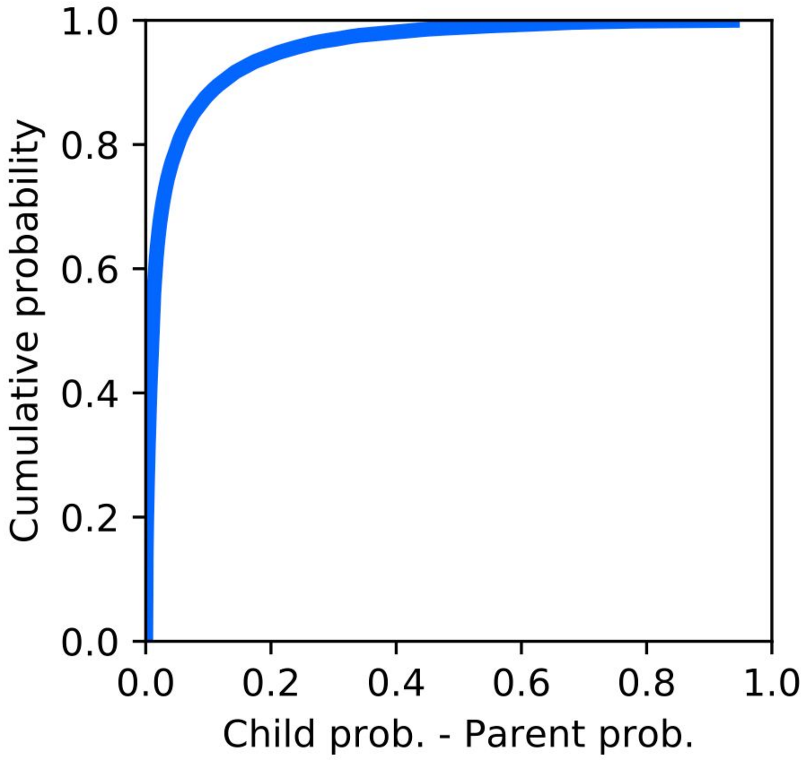
B



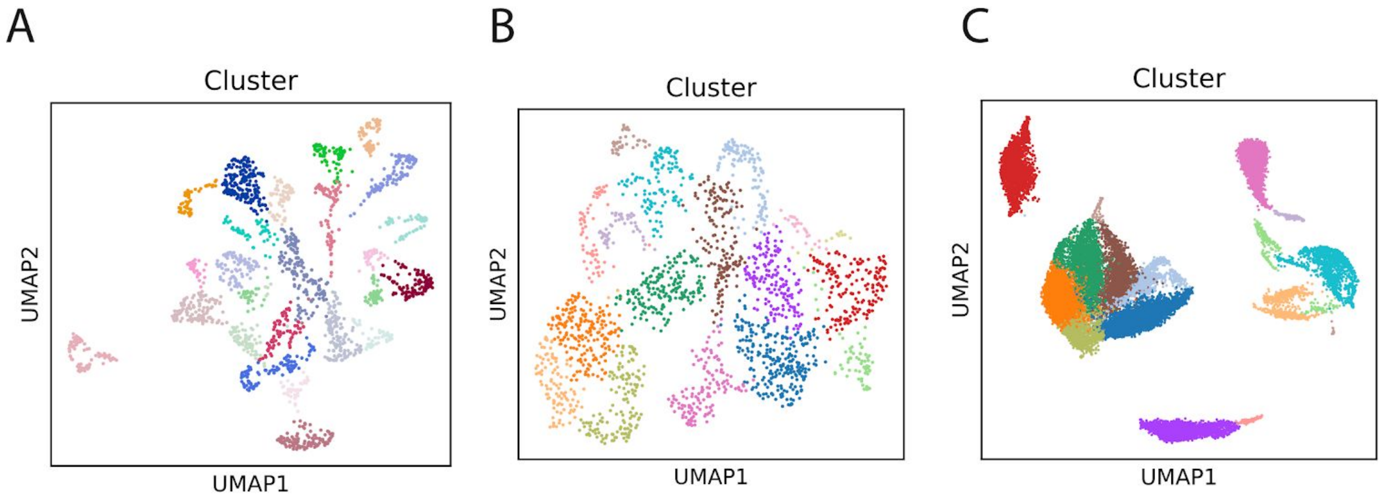
Gene: GFAP

glial fibrillary acidic protein

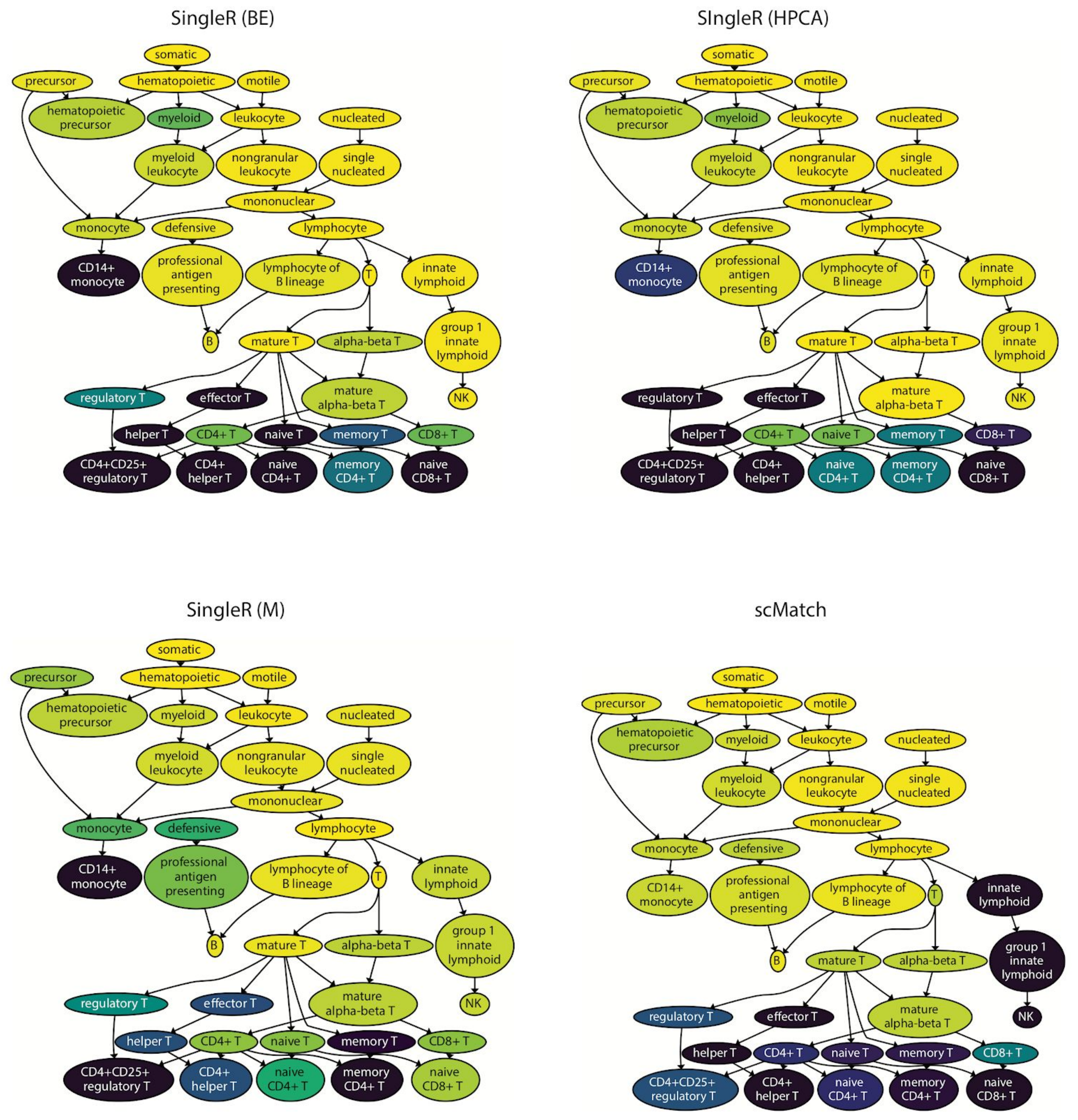
neural cell:	0.01142
neuron associated cell:	0.01120
glial cell (sensu Vertebrata):	0.01106
glial cell:	0.01106
macroglial cell:	0.00892
astrocyte:	0.01034



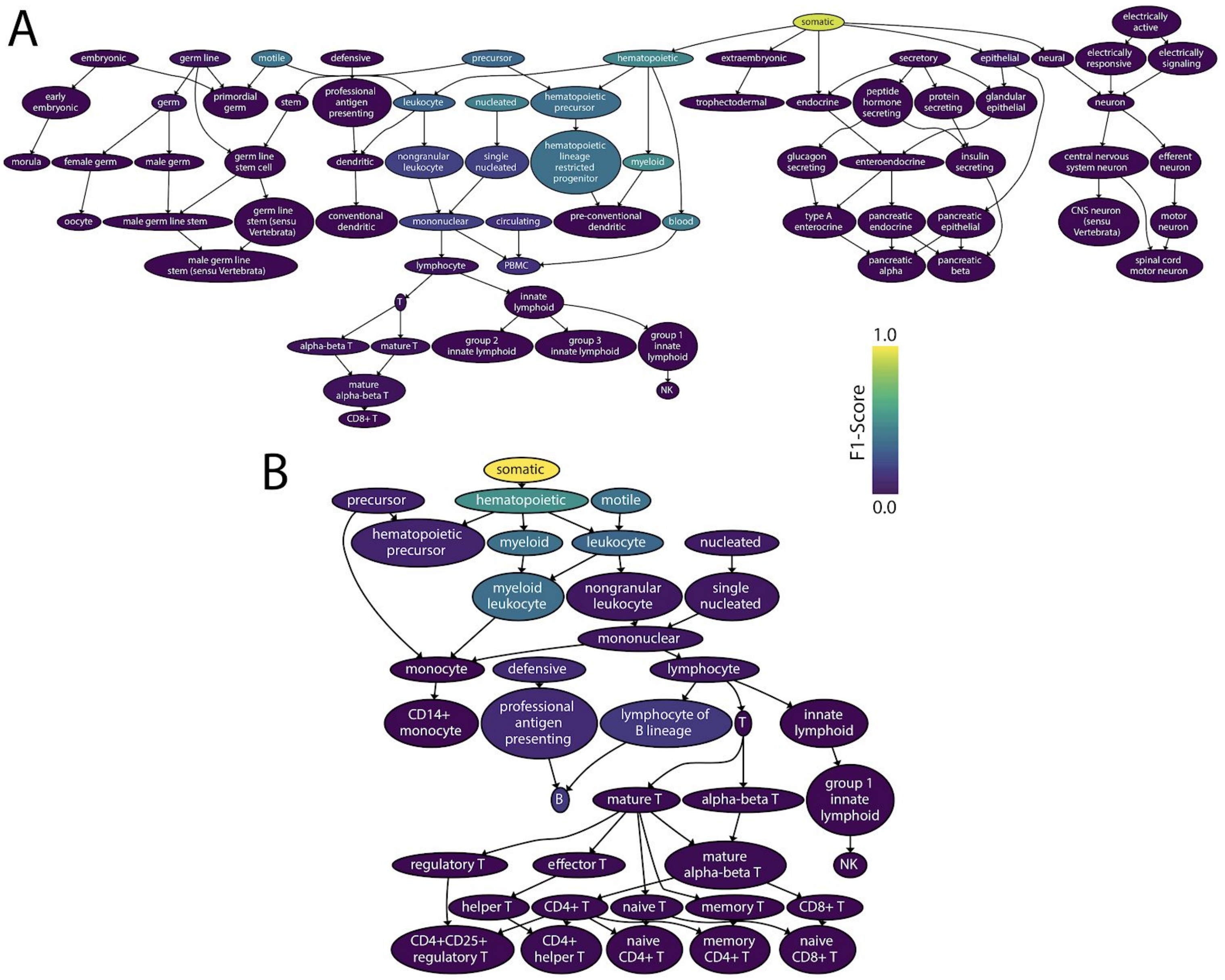
Supplemental Figure 1. **Distribution of edge inconsistencies.** The cumulative distribution function over the difference in probability between the parent and child classifiers for all edges for which either the parent or child classifier output a probability greater than 0.01.



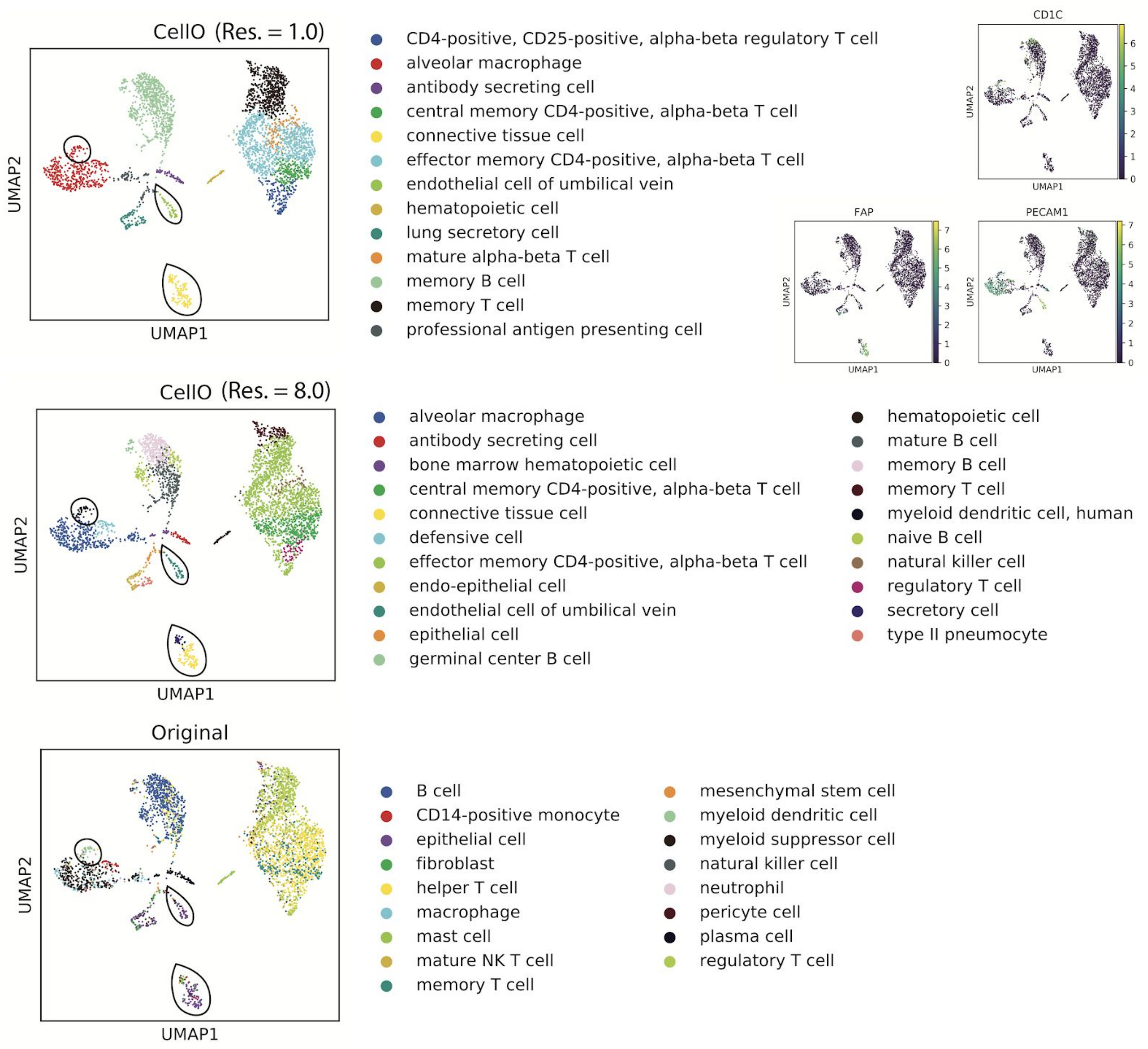
Supplemental Figure 2. **Cluster assignments for single-cell datasets evaluated in this work.** UMAP plots of the single-cell datasets examined this work where each cell is colored according to its CellO cluster assignment. These clusters were aggregated to compute a mean expression profile and on which CellO was run. Plots are displayed for (**A**) the Segerstolpe et al. (2016) dataset of healthy pancreatic cells, (**B**) the La Manno et al. (2016) dataset of fetal neural cells, and (**C**) the Zheng et al. (2017) dataset of PBMCs.



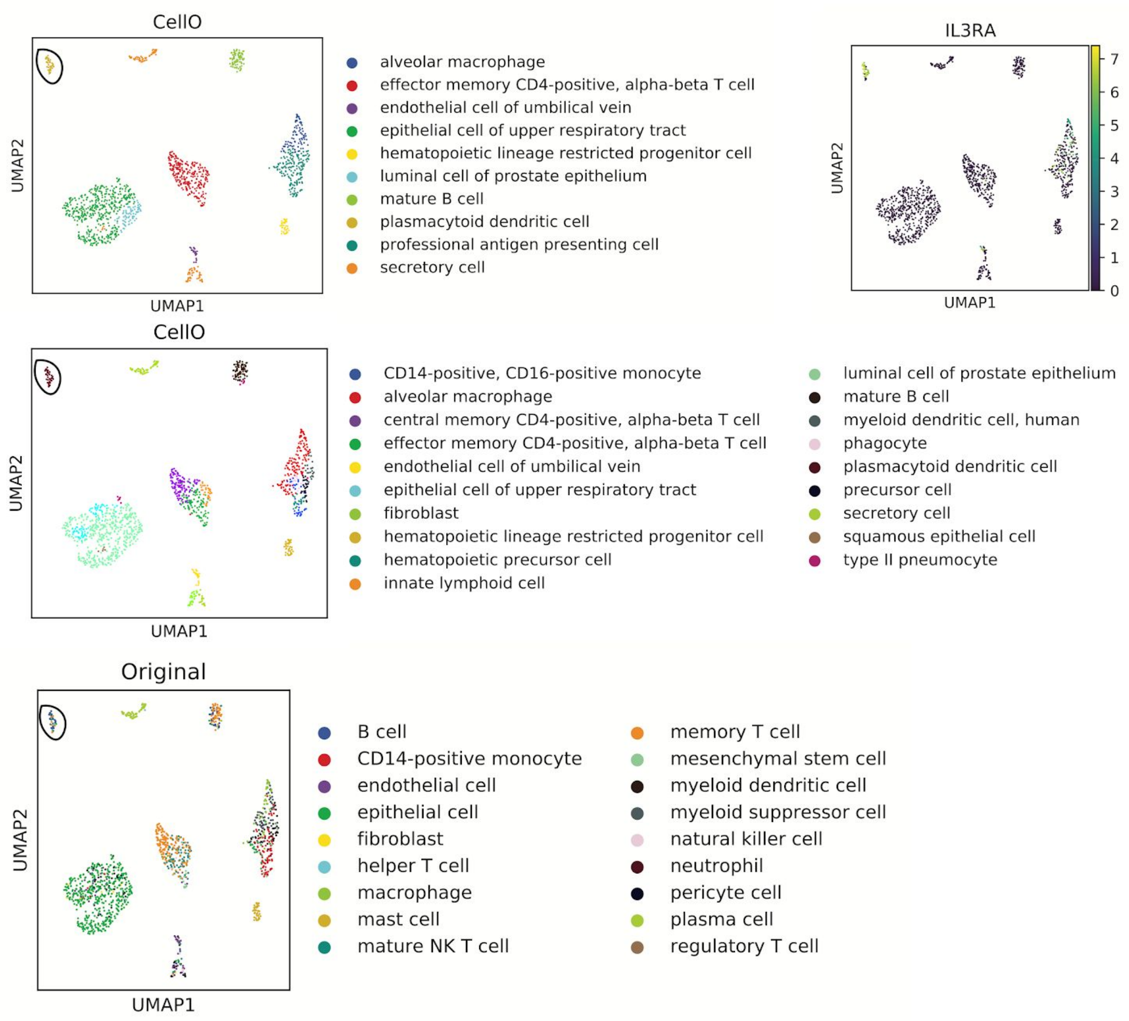
Supplemental Figure 3. **PBMC F1-scores produced by existing methods.** The subgraph spanning the PBMC's from Zheng et al. (2017) where each node is colored according to the F1-scores produced by SingleR with the Blueprint+Encode reference (BE), SingleR with the Human Primary Cell Atlas reference (HPCA), SingleR with the Monaco et al. (2019) reference (M), and scMatch.



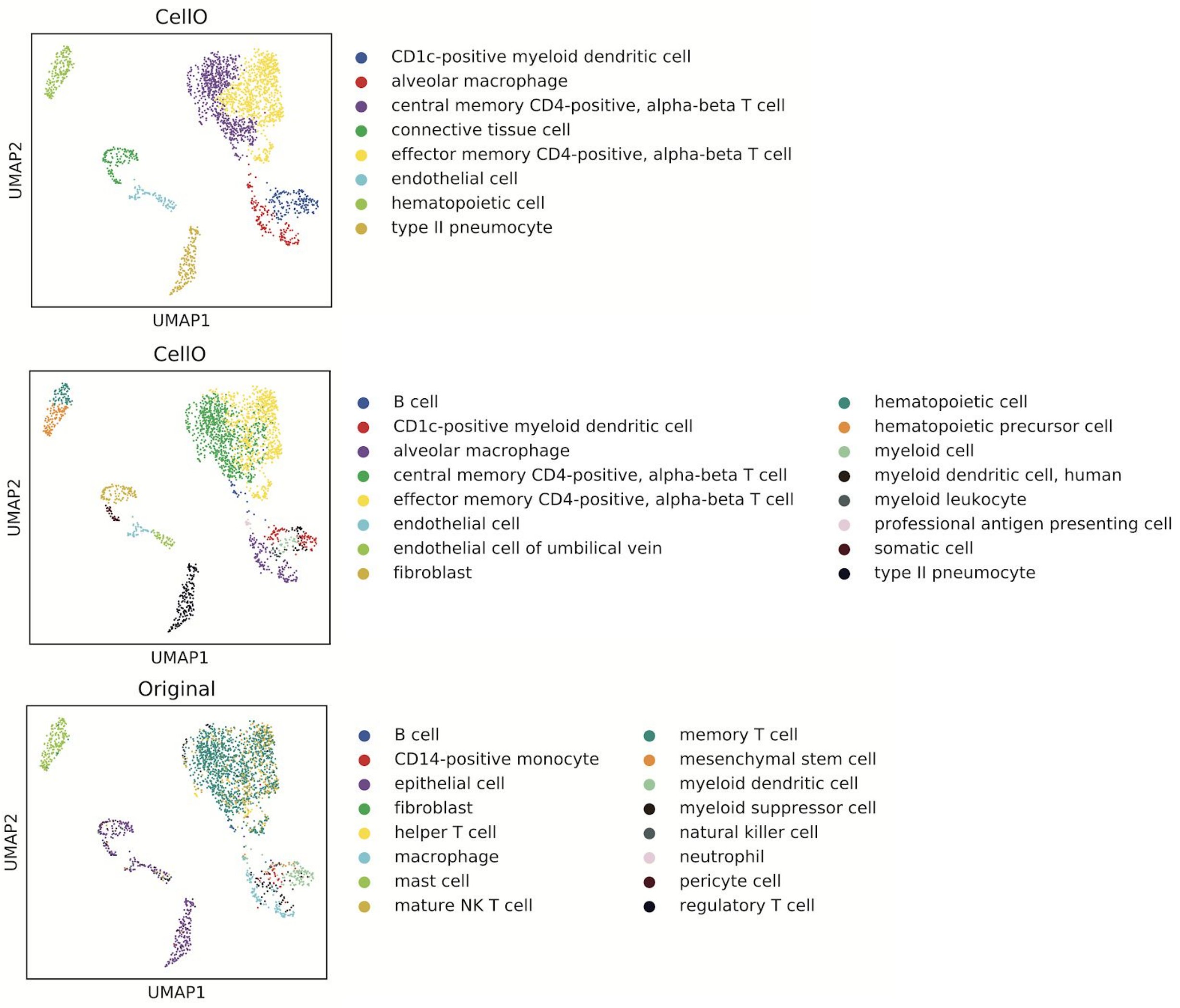
Supplemental Figure 4. **PBMC F1-scores produced by SingleR using CellO's training set.** (**A**) The subgraph spanning the non-droplet-based cells and (**B**) the subgraph spanning the PBMC's from Zheng et al. (2017) where each node is colored according to the F1-scores produced by SingleR with CellO's reference.



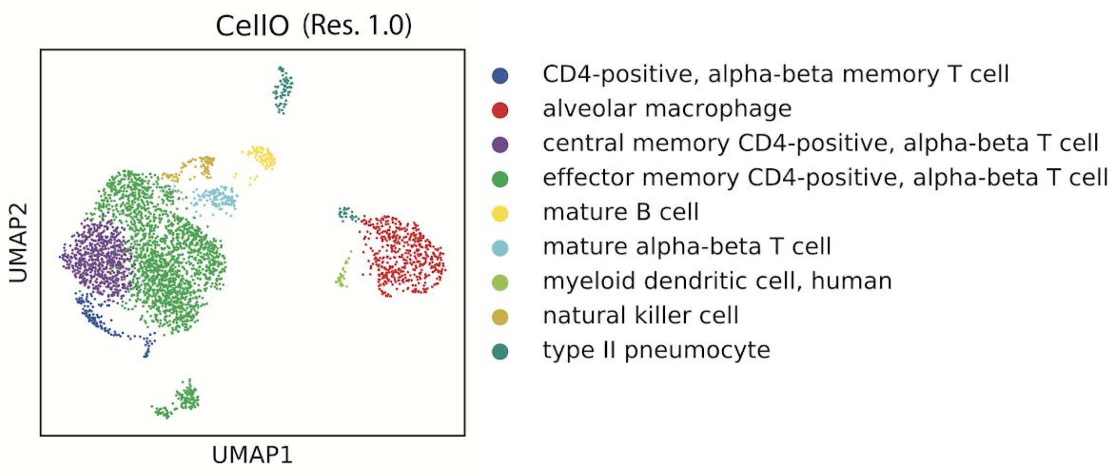
Supplemental Figure 5. **Examination of CellO's Output for Tumor LX682.** UMAP plots of lung adenocarcinoma tumor LX682 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors. We highlight three subpopulations of putative dendritic cells (top subpopulation), endothelial cells (middle subpopulation) and fibroblasts (bottom subpopulation). We verified these subpopulations using known marker genes for these cell types (top right).

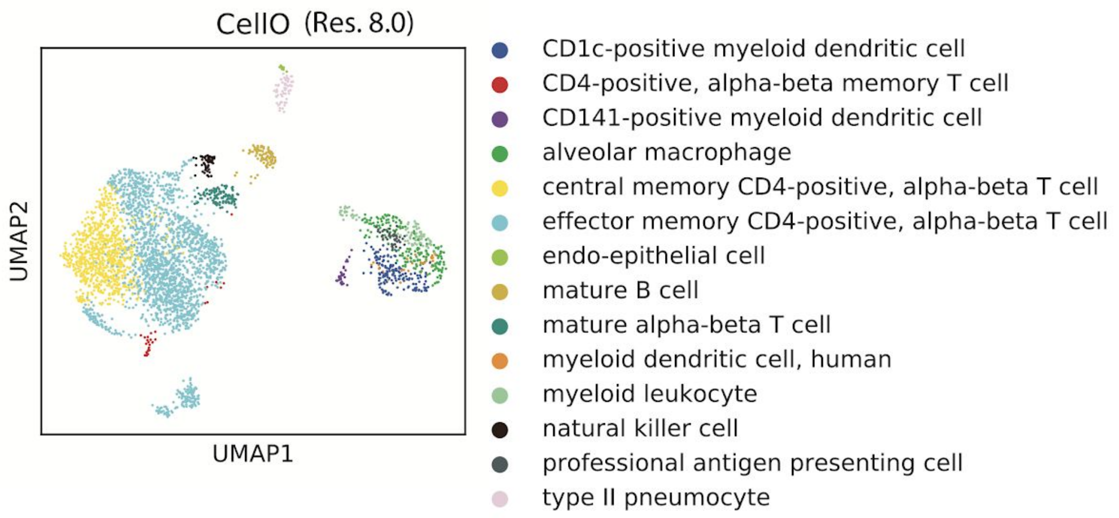


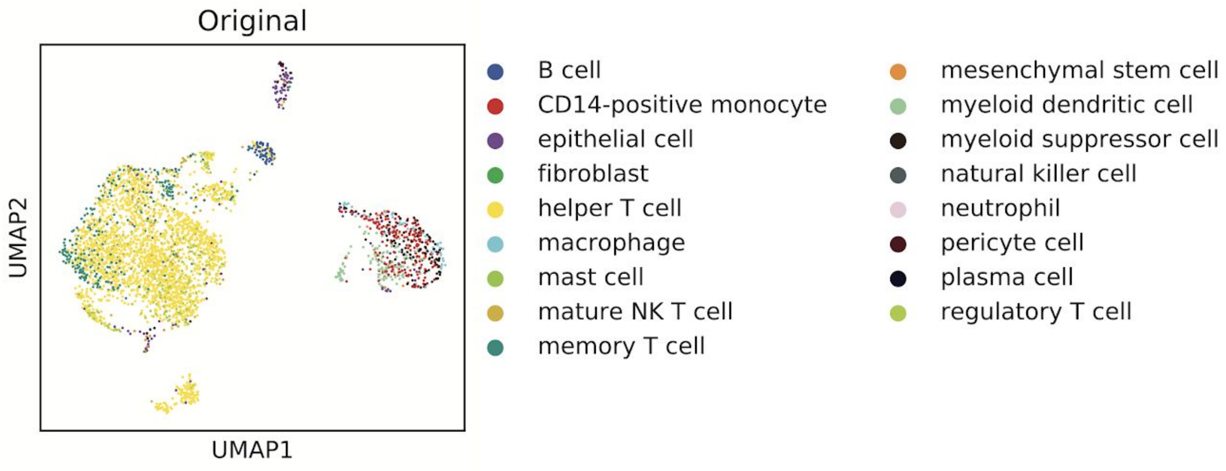
Supplemental Figure 6. **Examination of CellO’s Output for Tumor LX679.** UMAP plots of lung adenocarcinoma tumor LX679 from Laughney et al. colored by (top) CellO’s output using IR and a Leiden resolution parameter of 1.0 (middle) CellO’s output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors. We highlight the plasmacytoid dendritic cell population and verify its identity using known marker genes for these cell types (top right).



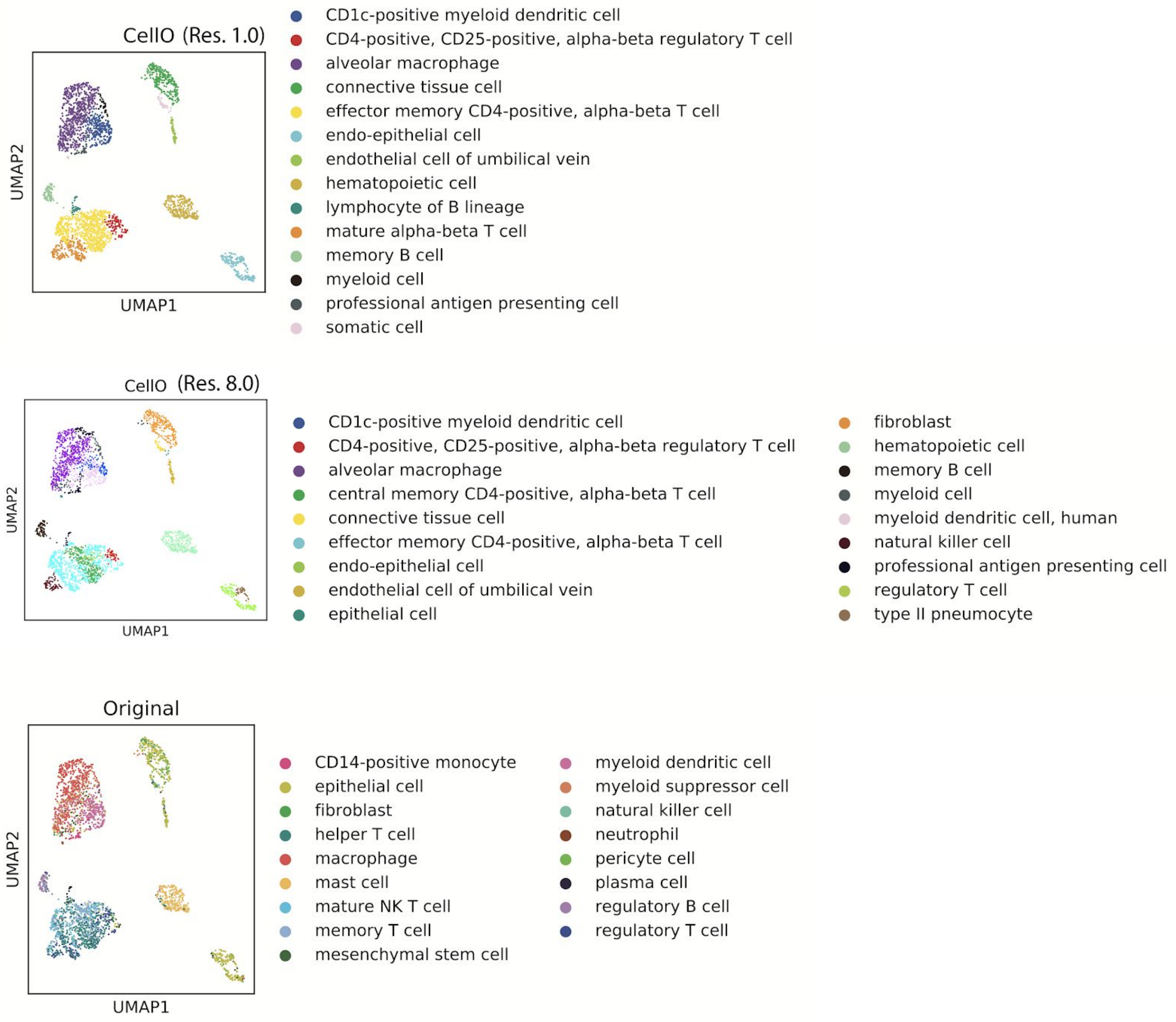
Supplemental Figure 7. **Examination of CellO's Output for Tumor LX653.** UMAP plots of lung adenocarcinoma tumor LX653 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors.



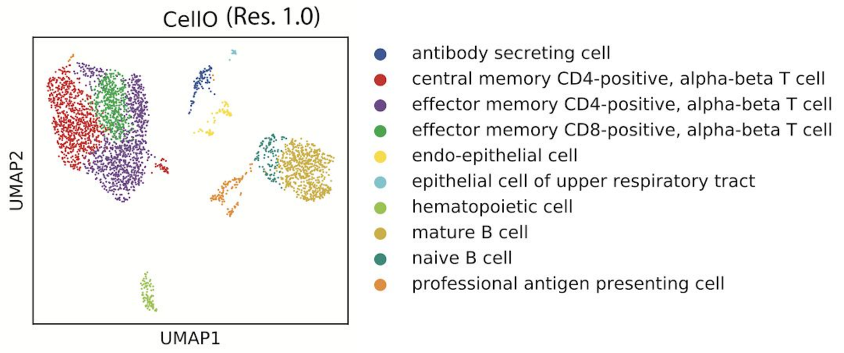


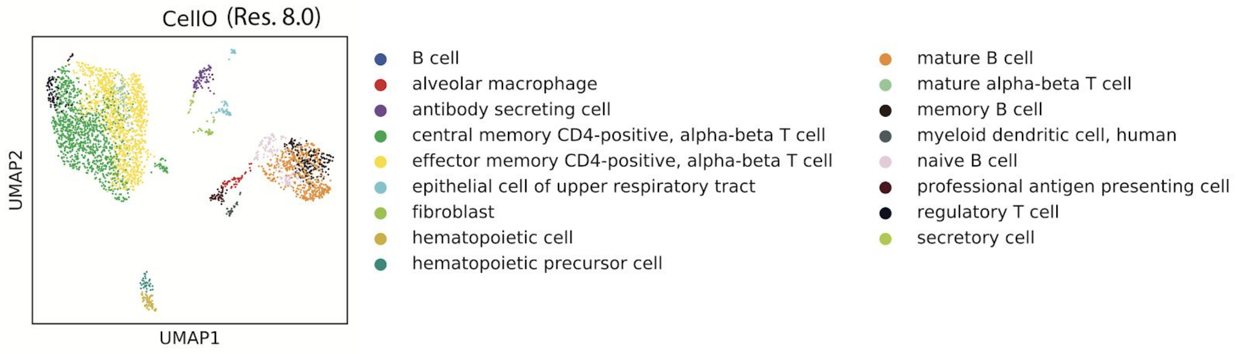


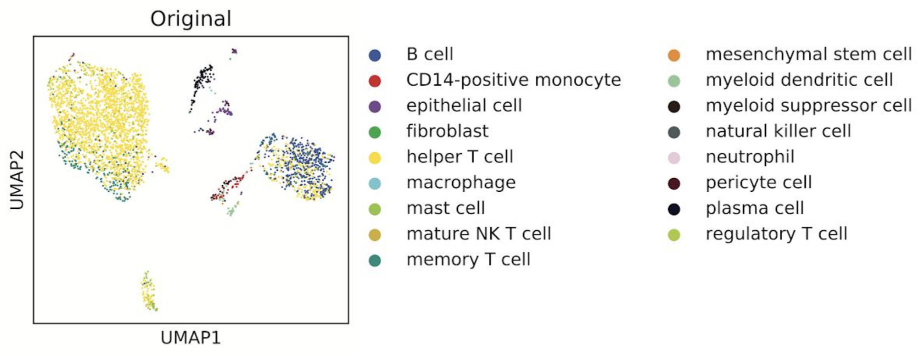
Supplemental Figure 8. **Examination of CellO's Output for Tumor LX661.** UMAP plots of lung adenocarcinoma tumor LX661 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors.



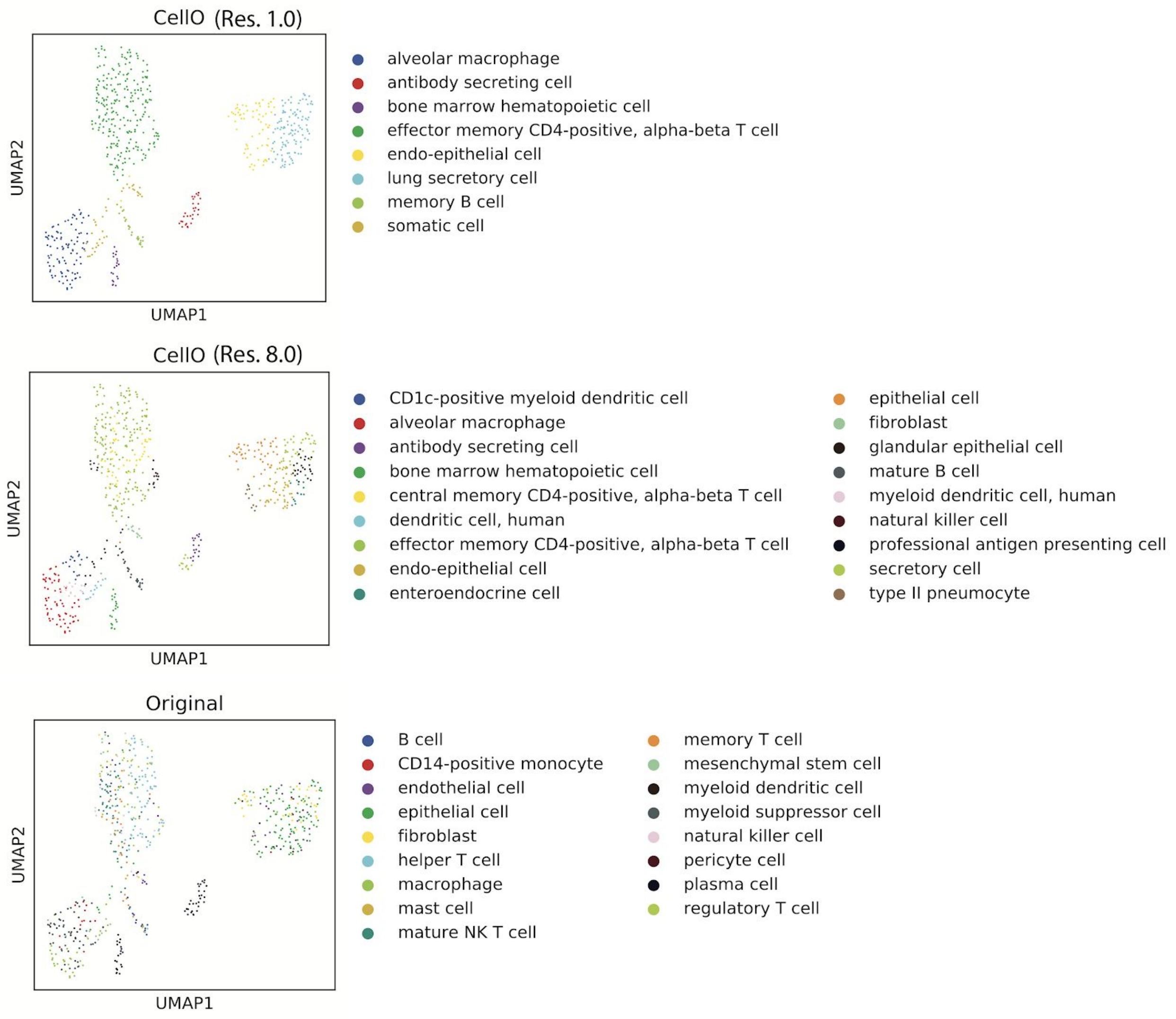
Supplemental Figure 9. **Examination of CellO's Output for Tumor LX675.** UMAP plots of lung adenocarcinoma tumor LX675 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors.



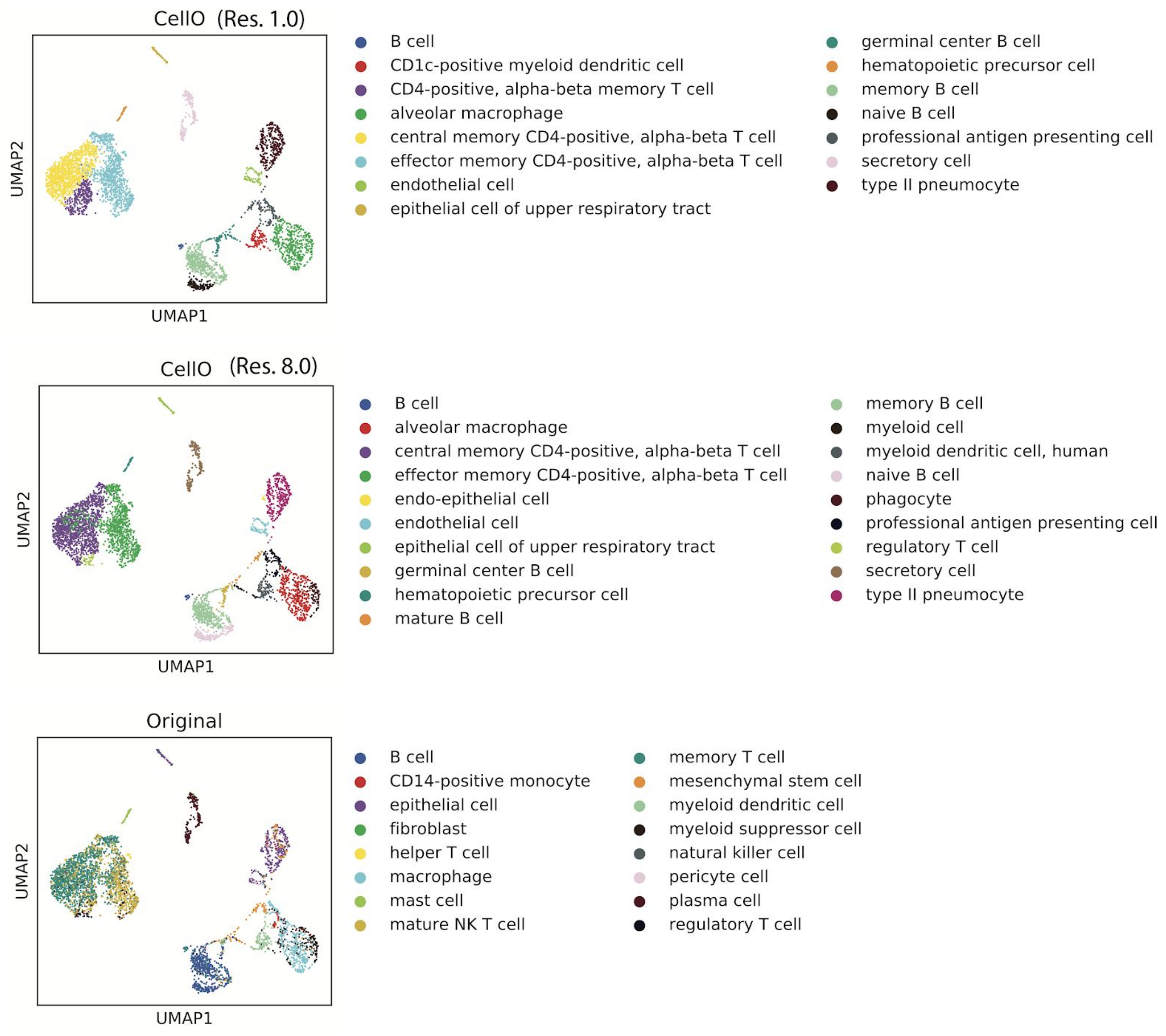




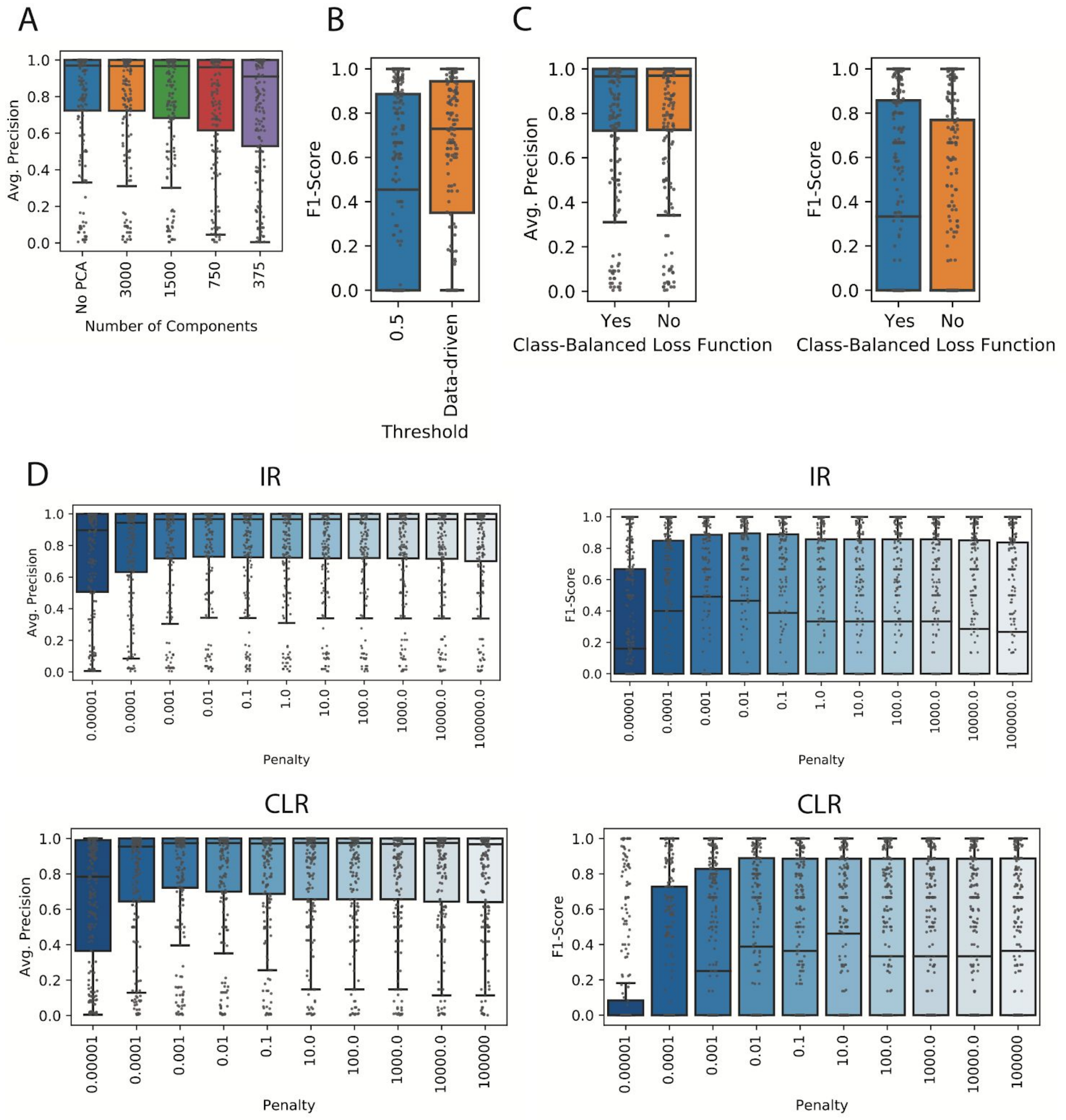
Supplemental Figure 10. **Examination of CellO's Output for Tumor LX676.** UMAP plots of lung adenocarcinoma tumor LX676 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors.



Supplemental Figure 11. **Examination of CellO's Output for Tumor LX680.** UMAP plots of lung adenocarcinoma tumor LX680 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors.

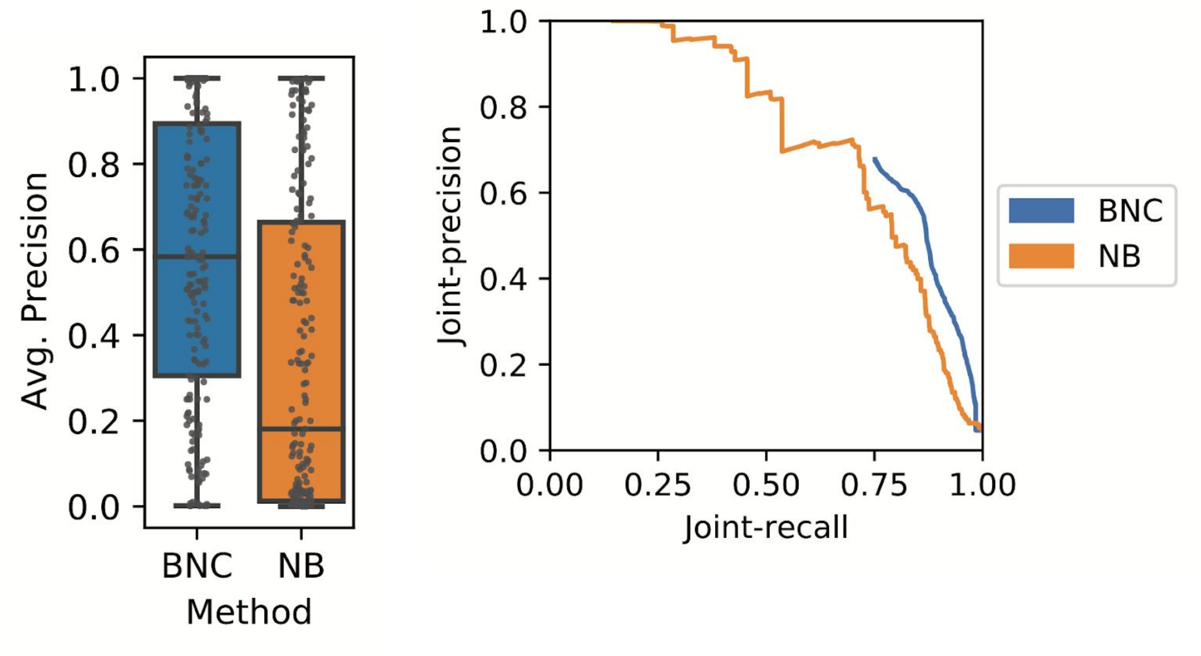


Supplemental Figure 12. **Examination of CellO's Output for Tumor LX684.** UMAP plots of lung adenocarcinoma tumor LX684 from Laughney et al. colored by (top) CellO's output using IR and a Leiden resolution parameter of 1.0 (middle) CellO's output using IR and a Leiden resolution parameter of 8.0, and (bottom) the original cell type labels provided by the authors.

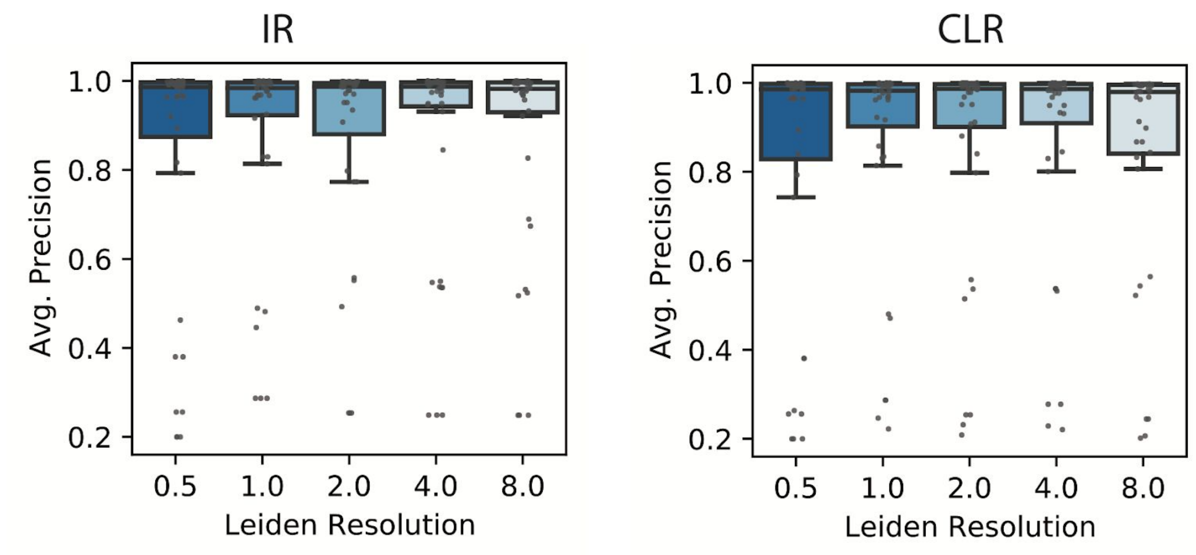


Supplemental Figure 13. **Parameter tuning.** (**A**) Average precision scores across cell types in the bulk RNA-seq validation set produced by the independent one-vs.-rest classifiers on data preprocessed with various numbers of principal components. (**B**) F1-scores produced by the independent classifiers when using a threshold of 0.5 for making a binary yes-no decision for all cell types versus a custom threshold for each cell type as empirically determined via a leave-study-out cross-validation experiment on the pre-training set of bulk RNA-seq samples.

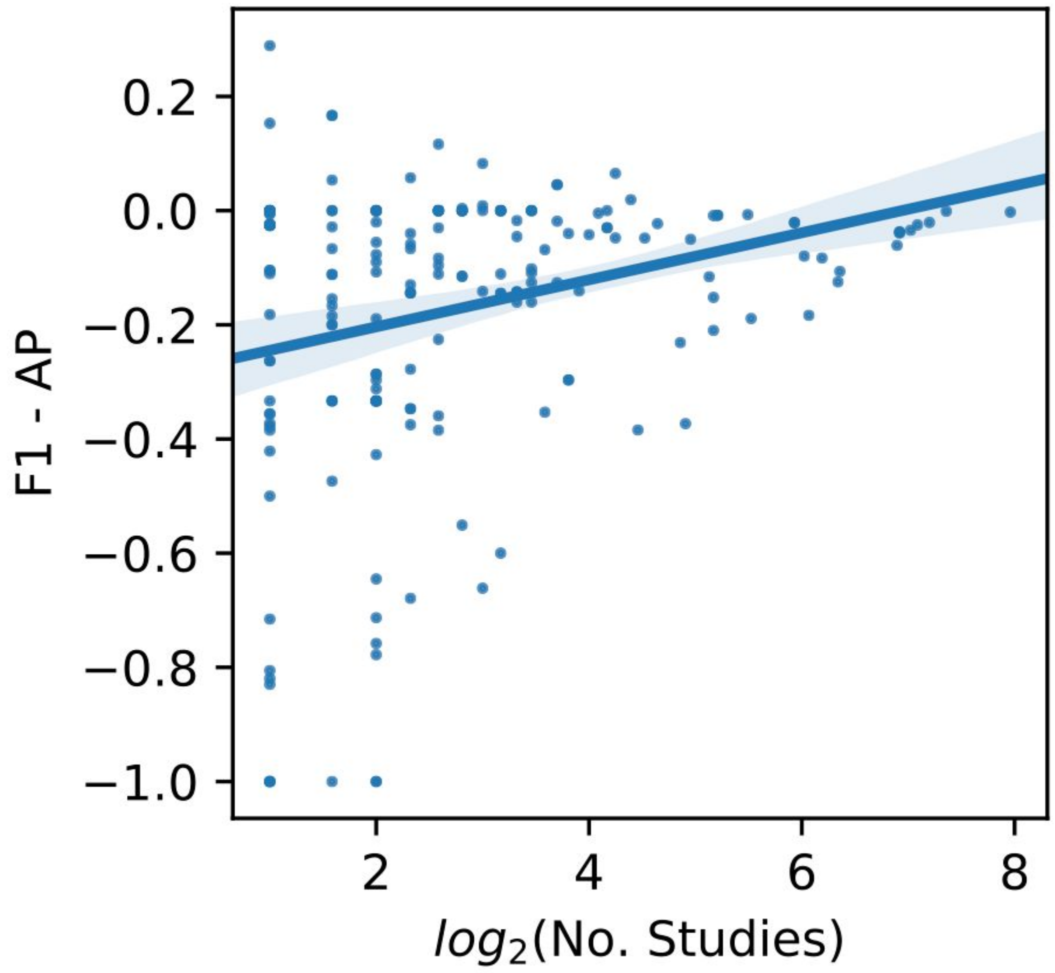
(**C**) Average precision (left) and F1-scores (right) across cell types in the bulk RNA-seq validation set produced by the independent one-vs.-rest classifiers trained either with or without the class-balanced loss-function. (**D**) Average precision scores across cell types in the bulk RNA-seq validation set produced by the one-vs.-rest and CLR conditional classifiers when trained with various regularization strengths (i.e. penalties). These penalty values correspond to the inverse of the regularization strength and thus, smaller numbers indicate stronger regularization.



Supplemental Figure 14. **Comparing Bayesian network correction with naive Bayes.** A comparison between BNC and a naive Bayes variant of BNC that considers each cell type independent of the graph-dependencies between cell types. We compared these approaches in terms of both the distribution of average precisions across cell types (left) and joint precision-recall curves constructed by considering each cell-cell type pair as an independent prediction.



Supplemental Figure 15. **Effect of clustering parameter on performance.** We clustered the Zheng et al. (2017) PBMC dataset using various resolution parameters for the Leiden community-detection algorithm and then computed the average-precision across all cell types produced by IR (left) and CLR (right) on the mean expression profiles for the clusters generated under each parameter.



Supplemental Figure 16. **Effect of number of training studies on calibration quality.** We evaluated IR on the bulk RNA-seq validation set and for each cell type, we computed the F1-score (F1) using a decision-threshold of 0.5 and average precision (AP). We then plot the difference between F1 and AP against the logarithm of the number of studies in the bulk RNA-seq pre-training set that sequenced each cell type. A larger gap between F1 and AP (i.e. negative numbers with large magnitude) indicates poorer calibration. Error bands around the ordinary least-squares regression line indicate 0.95 confidence intervals estimated via bootstrapping.