RESEARCH

Hierarchical cell type classification using mass, heterogeneous RNA-seq data from human primary cells

Matthew N Bernstein[1] and Colin N Dewey[1,2*]

*Correspondence: colin.dewey@wisc.edu
[1]Department of Computer Sciences, University of Wisconsin - Madison, 1210 W Dayton St, WI 53706 Madison, USA
Full list of author information is available at the end of the article

Abstract

Gene expression-based classification of a biological sample's cell type is an important step in many transcriptomic analyses, including that of annotating cell types in single-cell RNA-seq datasets. In this work, we explore the novel application of hierarchical classification algorithms that take into account the graph structure of the Cell Ontology to this task. We train these algorithms on a novel curated dataset comprising nearly all human public, primary bulk samples in the NCBI's Sequence Read Archive. These algorithms improve on state-of-the-art methods and produce accurate cell type predictions on both bulk and single-cell data across diverse and fine-grained cell types.

Keywords: Machine learning; Cell type; RNA-seq; Hierarchical classification; Ontology; Sequence Read Archive; Gene expression

1 Background

Gene expression-based computational classification of a biological sample's constituent cell type is an important task in many gene expression analysis tasks including that of annotating cell types in single-cell RNA-seq datasets [1, 2], improving the metadata in public genomic databases [3, 4], and verifying outcomes of experiments that entail inducing cellular differentiation [5, 6]. Furthermore, interpretable cell type classifiers may enable greater understanding of cell-type-specific expression patterns and may prove useful towards efforts, such as the Human Cell Atlas [7], that seek to define and catalog all cell types in the human body. The NCBI's Sequence Read Archive (SRA) [8] promises to be a valuable resource for training machine learning algorithms for this task due to the high number and large variety of cell type samples it contains. However, it has remained underutilized due to both the poor structure of the metadata [9, 10] and the difficulty in obtaining uniformly processed expression data. These challenges have recently been addressed through new metadata normalization efforts [11], efficient RNA-seq quantification algorithms [12, 13], and mass data processing efforts [14, 15, 16] thus paving the way towards the utilization of the SRA for training cell type classifiers.

In this work, we address three goals pertinent to this task:

1. To capture robust cell type signals by training on, and evaluating with, only healthy, primary, purified human samples.
2. To take advantage of the hierarchical nature of cell type definitions by exploring novel applications of hierarchical machine learning classification methods.

3 To build interpretable models that can be used to gain deeper understanding into the expression patterns that distinguish cell types.

Existing approaches for cell type prediction address some of these goals, but there has yet to be an investigation that addresses them all simultaneously.

First, to the best of our knowledge, none of the existing machine learning-based cell type classification approaches that train on public expression data distinguish between treated versus untreated cells [1, 5, 3]. By training on non-primary cells or treated cells, a classifier becomes more susceptible to batch effects when treatment or disease confounds cell type. This also leads to difficulty in model interpretation as it is unclear whether the derived signal is indicative of cell type or of a confounding variable such as treatment or disease. In this work, we compiled a set of training data from the SRA comprising only healthy, primary cells.

We also assert that framing the cell type classification task as that of hierarchical classification against the Cell Ontology [17] 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. Flat classification suffers from the possibility that predictions are logically inconsistent with the hierarchy of cell types in that the classifier for some cell type may, for a given query, output a probability that is larger than the classifier's output for its parent cell type in the hierarchy [18]. Such outputs reduce the interpretability, and therefore scientific usefulness, of the model. In addition, the use of hierarchical classification approaches allows for the placement of a bulk RNA-seq sample at a level of the hierarchy appropriate to its heterogeneity. For example, a population of cells enriched for T cells may be heterogeneous in the sub-types of T cells (e.g., CD4+ T cells and CD8+ T cells). Finally, by utilizing the hierarchy during training rather than using flat classification, more accurate classifiers can be learned [19].

To the best of our knowledge, only work by Lee *et al.* (2013) frames the cell type prediction task as a hierarchical classification problem. To this end, they developed an algorithm called URSA, which uses a framework called Bayesian Network Correction (BNC) [19]. There also exist discriminative methods for hierarchical classification that have yet to be applied to the cell type prediction task. Thus, we applied a number of such approaches including cascaded logistic regression (CLR), isotonic regression correction (IR) [18], and a heuristic procedure called the True Path Rule (TPR) [20]. We compared these discriminative methods to BNC and in our hands found them to outperform the BNC approach.

Furthermore, we sought for our methods to be interpretable in order for our trained classifiers to be of use not only in classification, but also for investigating cell type-specific expression patterns. To this end, this work makes extensive use of linear models, which are particularly amenable to interpretation. We tested the interpretability of the aforementioned frameworks and found that CLR is particularly interpretable as it is able to delineate functional differences between similar cell types.

We tested these algorithms on single-cell RNA-seq (scRNA-seq) data, resulting in promising performance. We propose that hierarchical, machine learning-based

classification of single-cell expression data will help overcome a number of challenges in cell type labeling of single-cell datasets. Currently, labeling cell types in scRNA-seq data is an ad hoc process that involves clustering the cells and then searching for differential expression of certain cell-type-specific marker genes across these clusters. This process is challenged both by the fact that there is not a canonical set of marker genes for most cell types [21] and that this process is affected by the clustering algorithm [22]. Approaches are beginning to emerge, such as work by Alavi *et al.* (2018), that rely on training cell type classifiers on single-cell data.

We note that the process of training a cell type classifier on single-cell data is somewhat circular in that the ground truth cell type labels are most commonly based upon gene expression (via the expression of cell type-specific marker genes), which is then also used for constructing the machine learning features. In this work, we train our algorithms on only bulk RNA-seq data, that originate from cells that have been isolated based on phenotypic characteristics downstream of gene expression itself (such as cell surface proteins). Thus, we suggest that bulk RNA-seq data in the SRA cannot only be utilized, but also may be preferred, for the training of cell type classifiers applied towards scRNA-seq datasets.

Finally, we created a Python package, CellO (*Cell O*ntology-based classification) that allows users to run pre-trained classifiers on their own RNA-seq data. CellO is available at `https://github.com/deweylab/CellO`.

2 Results and discussion

A novel curated RNA-seq dataset of human primary cells

In order to capture robust cell type signals, we sought a dataset of RNA-seq samples comprising only healthy primary cells. 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. We leveraged the annotations provided by the MetaSRA project [11], which includes sample-specific information including disease-state, treatment, and sample type (i.e., their status as primary cells). Consequently, 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 used the MetaSRA to capture an initial candidate set of primary samples and then within this set, manually annotated these samples for technical variables (such as bulk vs. single-cell status) by consulting sources of metadata that are not captured by the MetaSRA annotation process such as fields in Gene Expression Omnibus [23] records and each study's publication. When found, we corrected errors in the MetaSRA-provided Cell Ontology labels.

This process resulted in a dataset comprising 11,569 total samples. We uniformly quantified and normalized (via log counts per million) gene expression from the raw RNA-seq data for these samples. Of these samples, 4,167 were bulk RNA-seq samples from 263 studies and labeled with 294 cell type terms in the Cell Ontology. Of these cell types, 105 cell types 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

(Fig. 1). 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 *et al.* (2017), which contained 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 a microarrays. In addition to bulk samples, our dataset also includes 7,402 single-cell samples from 15 studies and labeled with 118 cell types.

To evaluate the implemented machine learning methods on their ability to classify bulk RNA-seq data, we split the bulk RNA-seq data set into a training and a test set, ensuring both that samples from the same study were never split across the training and test sets and that the training and test partitions had a high overlap of cell types. This partition yielded a training set with 3,480 samples across 206 studies and a test set with 687 samples across 57 studies. The training set and test set shared 197 cell types.

We separately evaluated these methods on their ability to classify scRNA-seq data. To this end, we created a second test set of all single-cell samples whose cell types appeared in the bulk RNA-seq data. This resulted in a test set of 4,961 samples across 13 studies from 66 cell types. As detailed below, we separately examined how the classifiers handled the remaining single-cell samples whose cell types do not appear in the bulk RNA-seq training data.

Novel applications of hierarchical classification methods

One straightforward approach to performing cell type prediction against the Cell Ontology entails training an independent binary classifier for each cell type in the ontology. We will refer to this as the "independent classifiers" approach. Such an 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. We tested the use of independent classifiers and found inconsistencies to be an important source of errors (Fig. S3). 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 (> 0.1) of the sample originating from that cell type. Of these edges, 6.7% were inconsistent (Supporting Methods).

Hierarchical classification algorithms ensure that the output probabilities are consistent with the ontology. We tested three ensemble-based hierarchical classification algorithms that have yet to be applied to the gene expression-based cell type prediction task: cascaded logistic regression (CLR), isotonic regression correction (IR) [18], and a heuristic procedure called the True Path Rule (TPR) [20]. Cascaded logistic regression entails classifying a sample in a top-down fashion from the root of the ontology downward via an ensemble 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. In contrast, IR and TPR train independent, unconditional, one-versus-rest binary classifiers for each cell type and then, for a given query sample, reconcile the output of these independent classifiers to be consistent with the ontology. IR uses a

projection-based approach for reconciliation, 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 date, the one hierarchical classification method that has been applied to the task at hand is BNC [19], and therefore, as a baseline, we implemented a BNC algorithm following the description in Lee *et al.* (2013). We tested a number of variants of this algorithm and report here the best-performing variant (Fig. S4). Lastly, as a naïve baseline, we implemented a one-nearest-neighbor algorithm that simply returns the cell type labels of the most similar sample in the training set to the query sample using Pearson correlation as the similarity metric.

We performed three modes of evaluation: per-cell-type, per-sample, and joint [18]. In the per cell type mode of evaluation we evaluate the performance of each method on each cell type independently. The results of this mode of evaluation are provided for users who are interested in examining each method's performance on specific cell types. Specifically, for each cell type, we compute both the average precision (a measure of the area under the precision-recall curve) as well as the maximum achievable recall at 0.9 precision. This latter metric is provided for users who cannot tolerate low precision.

In the per-sample mode of evaluation, we examine the average performance of the classifiers on a per-sample basis. To this end, we used two variants of precision and recall that are sample-centric. Given a sample, the first variants are the standard precision and recall over the sample's true cell types and predicted cell types. The second variants, which we call *specific-precision* and *specific-recall*, take into account only the sample's most-specific true cell types and predicted cell types according to the ontology (i.e., the deepest terms in the ontology – see Methods). Then, for a given prediction threshold, we compute the mean precision and recall (as well as mean specific-precision and mean specific-recall) across all samples. By varying our prediction threshold, we compute a *mean precision-recall curve*, where an operating point on this curve describes an achievable mean precision and mean recall across all samples.

A disadvantage to these mean precision and mean recall metrics is that they can be dominated by large studies due to samples from the same study sharing batch effects and similar cell type labels. To counteract this, we also compute curves in which in our calculation of the mean precision and mean recall at a given threshold down-weights samples according to the number of samples in its study in order to ensure that each study contributes equally to the mean precision-recall curve. We refer to these curves as *study-weighted, precision-recall curves*. An operating point on such a curve describes an expected precision and recall that is achievable given that a study is first sampled uniformly from all available studies, and then an RNA-seq sample is sampled uniformly from that study.

Lastly, for each method, we performed a joint evaluation that entailed treating each paired sample and cell type prediction independently. The set of all such predictions was ordered according to prediction probability and the corresponding precision-recall curve was constructed.

Advantages of training on data from heterogeneous sources

We hypothesized that by leveraging data from multiple studies, we could mitigate the models fitting a single study's batch effects, and would therefore learn more robust signals for each cell type. We tested this hypothesis using a flat classification experimental setup in which the hierarchy of cell types was first ignored. Specifically, for a variety of cell types, we compared the performance between logistic regression binary classifiers trained on homogeneous data (data originating from a single study) versus those trained on heterogeneous data (data originating from different studies).

The experiment proceeded as follows: we first queried the bulk RNA-seq data for all cell types that included at least three studies with over 10 experiments for that cell type. For each of these cell types, c, we partitioned the data labelled with c according to their study of origin, and then iteratively held out each study-partition as a test set. From the remaining held-in partitions, we constructed two sets of training sets. The first set of training sets included positive examples (i.e. data labeled with c) from only one study, which we call *homogeneous training sets*. The second set of training sets included positive examples from all held-in study-partitions, which we call the *heterogeneous training sets*. For all training sets, we use a consistent set of negative examples randomly chosen from the samples that are not labelled as c. Furthermore, when constructing each training set, we ensured each had an equal number of positive examples. We then trained a binary classifier on each training set and evaluated them on the held-out study-partition. Figure 2a provides a schematic of the experiment (See Supplementary Materials for full details). We computed the mean average-precision for the homogeneously trained and heterogeneously trained classifiers across each held out study-partition and cell type pair and found that heterogeneously trained classifiers tended to have a higher mean average precision (Fig. 2b). These results support the hypothesis that better generalization can be achieved by training on data from multiple studies.

Given these results, we hypothesized that the hierarchical classification algorithms would be less likely to fit the study-specific batch effects of the larger studies if we increased the contribution of the small studies to each logistic regression, binary classifier's loss function. To this end, we tested variants of the aforementioned hierarchical classification algorithms for which the loss function of each binary classifier down-weights each sample according to the number of samples in its study so that each study contributes equally (Methods). After training on the bulk RNA-seq training set and testing on the bulk RNA-seq test set, we found that the version of CLR, IR, and TPR that used a sample-weighted loss function in each of their binary classifiers outperformed the unweighted version with respect to mean average-precision across the cell types and mean achievable recall at 0.9 precision (Fig. 2). Due to the fact that we use logistic regression for each of IR, TPR, and CLR's binary classifiers, using larger weights for samples from small studies is equivalent to oversampling training data from these studies. In effect, this increases the diversity of studies that contribute to each learned model. Thus, this result provides further evidence that it is advantageous to use training data from a diversity of studies. For all further analysis in this paper, we use the variant of IR, TPR, and CLR that utilize the weighted loss function.

Evaluation on bulk RNA-seq data

We evaluated the aforementioned hierarchical classification algorithms using the per-cell type (Fig. 3a-b, Fig. S5, Fig. S6), per-sample (Fig. 3c), and joint (Fig. 3d) modes of evaluation. Overall, we find that IR, TPR, CLR, and independent classifiers performed similarly and better than the baseline BNC and nearest-neighbor algorithms. The similar performance of IR, TPR, and CLR to the independent classifiers demonstrates that reconciling the outputs of the independent predictions with the ontology structure does not degrade performance. We note that these results are in line with work by Obozinski *et al.* (2008), which demonstrates that IR and CLR outperform BNC on the hierarchical protein function prediction task.

Regarding the per-sample mode evaluation, we note that mean performance on a sample's most-specific cell types was below that of the mean performance when considering all of the sample's cell types (Fig. 3c). We posit three reasons for this: first, it is likely easier for the classifiers to distinguish broad categories of cell types than it is to distinguish fine-grained cell types for which cell-type-specific expression signatures may be more subtle. Second, the amount of training data supporting each cell type strictly decreases down the ontology. Third, we note that a subset of the errors are due to the classifiers providing significant probability to more specific cell types than the most-specific true cell types for a given sample (e.g., a T cell sample predicted to be a CD4+ T cell sample). This may be due to the prevalence of an unlabeled, more-specific cell type in some heterogeneous bulk RNA-seq samples.

Evaluation on single-cell RNA-seq data

We trained the IR, TPR, and CLR algorithms on the entire set of bulk RNA-seq data and evaluated them on the test set consisting of 4,961 single-cell RNA-seq samples whose cell types appear in the bulk RNA-seq training data. We note that many cells were labeled as a broad cell type rather than a specific cell type. For example, in study ERP017126, many cells are described by the data as general pancreatic cells. These samples are likely missing cell type labels because they should, in theory, be labeled with a specific cell type (i.e., lower in the ontology) due to the facts that each sample originates from a single cell and that there are known subtypes of pancreatic cells. We therefore modified our evaluation metrics to take into account these ambiguous, generally-labeled single-cell samples so as not to penalize the algorithms for predicting cell types more specific than their given labels (Supporting Methods). We found that these algorithms perform well in all modes of evaluation using these modified metrics (Fig. 4, Fig. S7).

Next, we examined the predictions performed on cells that represented challenging cases for the classifiers. Specifically, we identified two categories of challenging samples: samples that were only labeled with a broad cell type and samples labeled with a combination of cell types that do not appear in the training data. We examined two studies, SRP067844 and ERP017126, that contained samples that were representative of these challenges and examined their predictions in depth.

When given samples labeled as a general cell type, but not a more specific cell type, the algorithm often predicted a more specific cell type than the labeled cell types. Study SRP067844 included a set of samples labelled only as embryonic, neural cells, but not as a more specific cell type. In such instances, the algorithm often

labeled them as the more specific label, "neuron", which may be accurate given that this study sought to sequence cells from the developing nervous system (Fig. 5a) [24]. Study ERP017126 contained a set of pancreatic cells that were unlabeled for a specific pancreatic cell type. Many of these cells were predicted as a specific endocrine cell type such as pancreatic alpha cells (Fig. 5b).

When the methods were provided a query that should be assigned with a combination of labels that it had not seen before during training, its outputs were reasonable. Study SRP067844 consisted of embryonic neural cells. Although the 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. For these samples, we found that the algorithm was often able to label these samples as "neural cell", but often failed to label them as "embryonic cell". Furthermore, the algorithm had difficulty labeling these samples with their specific neural cell types such as "radial glial cell" (Fig. 5c). Similarly, study ERP017126 contained various pancreatic cell types that did not exist in the training data such as pancreatic delta cells and pancreatic ductal cells. We found that the delta cells were often predicted correctly as enteroendocrine cells (Fig. 5d) and were not confused with similar pancreatic endocrine cell types such as alpha cells or beta cells. Similarly, pancreatic ductal cells were often predicted as secretory cells (Fig. 5e). Although the term "secratory cell" is not an ancestral term of "pancreatic ductal cell" in the Cell Ontology, these predictions may nonetheless be considered correct predictions given that pancreatic ductal cells are known to secrete biocarbonate [25].

Comparison of interpretability between frameworks

With the exception of the one-nearest neighbor classifier, the methods that we have explored can be subdivided into two categories: those that train a set of one-versus-rest binary classifiers (BNC, IR, TPR) and the CLR framework, which trains a set of "local" binary classifiers that classify a sample as a given cell type conditioned on the sample belonging to its parent cell types. We explored the question of whether one framework provides an advantage in model interpretability. To address this question, we analyzed the gene coefficients in each binary classifier's linear model for enrichment of genes involved in known biological processes. We use the number of enriched biological processes as a quantitative measure of model interpretability. Specifically, for each learned binary classifier, we rank the genes by their corresponding coefficients in the linear model. We then performed a gene set enrichment analysis with GSEA [26] on these ranked genes using all "biological process" gene sets from the Gene Ontology (GO) [27] that were associated with at least 5 genes. This analysis targeted enrichment at both the top and bottom of the ranked list of genes, which identified biological processes that were either relatively upregulated or downregulated in a given cell type. We then use a false discovery rate q-value cutoff of 0.05 for proclaiming enrichment.

We found that the models learned in the CLR framework tended to be enriched for more GO terms than the one-versus-rest frameworks (Fig. 6). We posit that this phenomenon is due to the fact that since the CLR framework involves the training of binary classifiers that seek to distinguish only between a small set of

similar cell types, the CLR's classifiers are "more focused" than the one-versus-rest classifiers, which seek to distinguish each cell type from *all* other cell types. Thus, the CLR framework may prove more useful for exploring cell type-specific expression patterns and for finding expression patterns that distinguish similar cell types. The trained model coefficients can be downloaded for further analysis from http://deweylab.biostat.wisc.edu/cell_type_classification.

3 Conclusions

In this work, we explore the application of hierarchical classification algorithms towards cell type prediction 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 [28, 29]. We demonstrate that the trained classifiers perform well across cell types on bulk RNA-seq data and offer a promising approach to cell type annotation in single cell datasets.

We also found that classification performance is not only dependent on the number of training samples, but also on the diversity of those samples. Specifically, we found that the classifier benefits from training on data from multiple studies. Thus, we argue that the heterogeneity present in the public expression data presents an opportunity to learn robust models. This observation may extend beyond cell type prediction to other phenotype prediction tasks such as expression-based disease prediction.

Furthermore, by using linear models, the 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 with 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 the 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, a number of newly developed single-cell sequencing protocols were absent from our data. Whereas our data included single-cell samples from protocols such as MARS-seq [30] and SMART-Seq2 [31], it did not include data from droplet-based protocols such as Chromium 10x. Future work will require evaluating the performance of these algorithms on such protocols. We expect that the methods described in this work will require modifications for these protocols to account for the extremely low read depths per cell common to these protocols.

Finally, we expect the performance of hierarchical classifiers to improve as both more data is collected and as the Cell Ontology is expanded. More 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.

Methods

A schematic diagram of the experiments in this study is given in Figure S1.

Data processing

We quantified the gene expression of all samples with kallisto (v0.43.1) [13] against the 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 method would not be significant for the cell type classification task. This produced estimated counts for 200,401 isoform-level genomic features. We summed these counts by gene to produce counts 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.

Partitioning bulk RNA-seq data into training and test sets

When creating a training and test partition of the bulk RNA-seq data, 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 training and test 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 test set. Second, we sought an approximately 80/20 split of the data between the training and test sets. Third, we sought for all cell types to be represented in both the training and test sets. Fourth, and finally, for all cell types represented by three or more studies, we sought for at least two of those studies to be assigned to the training set to enable an estimate of performance on that cell type when performing leave-study-out-cross validation on the training set during development. We framed this partitioning task as an optimization problem where our four criteria were encoded in an objective function (Supporting Methods). Minimizing this objective function entails creating a partition that most closely meets the aforementioned four criteria.

Description of algorithms

In the following descriptions of the algorithms used in this work, we let $\boldsymbol{x} \in \mathbb{R}^G$ denote a gene expression profile, in units of log-counts per million (log-CPM), where G is the number of considered genes. Specifically, for gene i in $\boldsymbol{x}$, log-CPM is defined as

$$x_i := \log\left(\left[\frac{c_i}{\sum_{j=1}^{G} c_j} \times 10^6\right] + 1\right)$$

where c_i is the expected number of reads mapped to the ith gene. We let n denote the number of samples, m denote the number of considered cell types, $y_i \in \{0, 1\}$ denote the cell type assignment for cell type $i \in [m]$, and $\boldsymbol{X}$ denote the training set.

Independent binary classifiers
We used logistic regression with L2-regularization, using scikit-learn (v.0.20.2), for all independent binary classifiers trained in the CLR, IR, and TPR frameworks as well as in the independent classifier baseline method. Our choice of L2 penalty over L1 penalty was motivated by our goal of training interpretable models. Specifically, because the L1 penalty induces sparsity, we were concerned that it would lead to zeroing-out the coefficients of important cell-type-specific genes when such genes correlated highly with other predictive genes. That is, we sought for our models to weight *all* genes according to their predictive ability.

One-nearest neighbor
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 $\boldsymbol{x}'$.

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 for each cell type $i \in [m]$ to model the distribution

$$q_i := p(y_i = 1 \mid \pi_i = 1, \boldsymbol{x})$$

where $\pi_i \in \{0, 1\}$ indicates whether the sample belongs to all of the parents of i 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 $\boldsymbol{x}$ originates from cell type i is computed via

$$p(y_i = 1 \mid \boldsymbol{x}) = q_i \prod_{j \in A_i} q_j$$

where A_i denotes the ancestors of cell type i in the ontology's DAG.

Bayesian Network Correction
A support vector machine (SVM) binary classifier is trained for each cell type using a linear kernel 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, 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 i is then the marginal probability

$$p(y_i = 1 \mid f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$$

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.*.

Isotonic regression correction

We train a binary classifier for each cell type $i \in [m]$ to model $p(y_i \mid \boldsymbol{x})$ 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

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

subject to

$$\forall i \in [m], \forall j \in \mathrm{Par}(i),\ p_i < p_j$$

where $\forall i \in [m], \hat{p}_i := p(y_i = 1 \mid \boldsymbol{x})$ as output by each classifier and $\mathrm{Par}(i)$ is the set of parent cell types for cell type i.

True Path Rule

We train a binary classifier for each cell type $i \in [m]$ to model $p(y_i \mid \boldsymbol{x})$ 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 i's output probability is set to

$$p_i := \frac{1}{|C_i| + 1} \left(\hat{p}_i + \sum_{j \in C_i} \hat{p}_j \right)$$

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

$$C_i := \{j \in \mathrm{Children}(i)\ :\ \hat{p}_i > t\}$$

is the set of children of cell type i 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 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.

Sample-weighted loss function

In logistic regression, the loss on a given sample is given by

$$\ell(y, \boldsymbol{x}) := -y \log(\hat{p}(\boldsymbol{x})) + (1 - y) log(1 - \hat{p}(\boldsymbol{x}))$$

where $\boldsymbol{x}$ is the feature vector, $y \in \{0,1\}$ is the true label and $\hat{p}(\boldsymbol{x}) \in [0,1]$ is the classifier's estimate of $p(y = 1 \mid \boldsymbol{x})$. The full loss over the data set $D := \{(y_1, \boldsymbol{x}_1), \ldots, (y_n, \boldsymbol{x}_n)\}$ is then

$$\mathcal{L}(D) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \boldsymbol{x}_i)$$

We tested a variant of logistic regression in which each sample's term in the loss function is weighted according to the number of samples in its study so that each study contributed equally to the loss function. For a given sample i, let S_i be the set of samples in the study for which i belongs. The sample-weighted loss function is then

$$\mathcal{L}_w(D) := \frac{1}{s} \sum_{i=1}^{n} \frac{1}{|S_i|} \ell(y_i, \boldsymbol{x}_i)$$

where s is the number of studies. This loss function is equivalent to the loss function that would obtained by oversampling samples from each study in proportion to the number of samples that study.

Per-sample evaluation metrics

In the per-sample mode of evaluation, we analyze the average performance over each sample. For a given sample, let T be the full set of a true cell type labels and P be the set of predicted labels. The per-sample precision and recall are then defined as

$$\text{Precision} := \begin{cases} \frac{|T \cap P|}{|P|} & : |P| > 0 \\ 1 & : |P| = 0 \end{cases}$$

$$\text{Recall} := \begin{cases} \frac{|T \cap P|}{|T|} & : |T| > 0 \\ 1 & : |T| = 0 \end{cases}$$

respectively. We further define a version of precision and recall, termed *specific-precision* and *specific-recall*, that seek to summarize how well the classifier is retrieving the most granular cell types that describe the sample. Given a cell type label c from the ontology, let $Ch(c)$ be the children of c in the ontology DAG. We then define the most-specific set of true labels and most-specific set of predicted labels as

$$T' := \{c \in T : |Ch(c) \cap T| = 0\}$$

$$P' := \{c \in P : |Ch(c) \cap P| = 0\}$$

respectively. Specific precision and recall are then defined as

$$\text{Specific-Precision} := \begin{cases} \frac{|T \cap P'|}{|P'|} & : |P'| > 0 \\ 1 & : |P'| = 0 \end{cases}$$

$$\text{Specific-Recall} := \begin{cases} \frac{|T' \cap P|}{|T'|} & : |T'| > 0 \\ 1 & : |T'| = 0 \end{cases}$$

respectively. Given these per-sample measures of precision and recall, we then compute mean precision (MP), mean recall (MR), mean specific-precision (MSP), and mean specific recall (MSR) across all samples.

Finally, as was noted previously, since samples from the same study perform similarly, these metrics will be most effected by these large studies. To counteract this effect we also define a set of average metrics that use a weighted mean so that each study contributes equally. These metrics, which we call weighted-mean precision (WMP), weighted-mean recall (WMR), weighted-mean specific-precision (WMSP), and weighted-mean specific-recall (WMSR) are defined as

$$WMP := \frac{1}{s}\sum_{i=1}^{n}\frac{1}{|S_i|}\text{Precision}_i$$

$$WMR := \frac{1}{s}\sum_{i=1}^{n}\frac{1}{|S_i|}\text{Recall}_i$$

$$WMSP := \frac{1}{s}\sum_{i=1}^{n}\frac{1}{|S_i|}\text{Specific-Precision}_i$$

$$WMSR := \frac{1}{s}\sum_{i=1}^{n}\frac{1}{|S_i|}\text{Specific-Recall}_i$$

where n is the total number of samples, s is the total number of studies, and S_i is the set of samples in the study that includes sample i. Finally, by varying the prediction threshold, we can compute curves for all of these metrics. Specifically, we compute mean PR-curves, mean specific-PR-curves, weighted-mean PR-curves, and weighted-mean specific-PR-curves.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
M.N. Bernstein thanks G. H. Bernstein 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.

Availability of data and materials
A Python package for running the cell type classifiers discussed in this work can be found at https://github.com/deweylab/CellO. The dataset used in this work can be found at http://deweylab.biostat.wisc.edu/cell_type_classification. All code for performing the experiments in this work can be found at https://github.com/deweylab/cell-type-classification-paper.

Author details
[1]Department of Computer Sciences, University of Wisconsin - Madison, 1210 W Dayton St, WI 53706 Madison, USA. [2]Department of Biostatistics and Medical Informatics, University of Wisconsin - Madison, 600 Highland Ave, WI 53792 Madison, USA.

References
1. Alavi, A., et al.: A web server for comparative analysis of single-cell RNA-seq data. Nature Communications **9**(1) (2018)
2. Lin, C., *et al.*: Using neural networks for reducing the dimensions of single-cell RNA-seq data. Nucleic Acids Research **45**(17), 156 (2017)

3. Lee, Y., *et al.*: Ontology-aware classification of tissue and cell-type signals in gene expression profiles across platforms and technologies. Bioinformatics **29**(23), 3036–3044 (2013)
4. Ellis, S.E., *et al.*: Improving the value of public RNA-seq expression data by phenotype prediction. Nucleic Acids Research **46**(9), 54 (2018)
5. Cahan, P., *et al.*: CellNet: Network biology applied to stem cell engineering. Cell **158**(4), 903–915 (2014)
6. Radley, A.H., *et al.*: Assessment of engineered cells using CellNet and RNA-seq. Nature Protocols **12**(5), 1089–1102 (2017)
7. Regev, A., *et al.*: The Human Cell Atlas. eLife **6**, 27041 (2017)
8. Leinonen, R., Sugawara, H., Shumway, M.: The Sequence Read Archive. Nucleic Acids Research **39**(Suppl. 1), 19–21 (2011)
9. Wang, Z., Lachmann, A., Ma'ayan, A.: Mining data and metadata from the gene expression omnibus. Biophysical Reviews, 1–8 (2018)
10. Gonçalves, R.S., *et al.*: Metadata in the BioSample online repository are impaired by numerous anomalies. In: Proceedings of the First Workshop on Enabling Open Semantic Science (SemSci), pp. 39–46 (2017)
11. Bernstein, M.N., Doan, A., Dewey, C.N.: MetaSRA: normalized human sample specific metadata for the Sequence Read Archive. Bioinformatics **33**(18), 2914–2923 (2017)
12. Patro, R., *et al.*: Salmon provides fast and bias-aware quantification of transcript expression. Nature Methods **14**, 417–419 (2017)
13. Bray, N.L., *et al.*: Near-optimal probabilistic RNA-seq quantification. Nature Biotechnology **34**(5), 525–527 (2016)
14. Lachmann, A., et al.: Massive mining of publicly available RNA-seq data from human and mouse. Nature Communications **9**(1366) (2018)
15. Collado-Torres, L., *et al.*: Reproducible RNA-seq analysis using recount2. Nature Biotechnology **35**(4), 319–321 (2017)
16. McCall, M.N., *et al.*: The Gene Expression Barcode 3.0: improved data processing and mining tools. Nucleic Acids Research **42**(D1), 938–943 (2014)
17. Bard, J., Rhee, S.Y., Ashburner, M.: An ontology for cell types. Genome Biology **6**(2), 21 (2005)
18. Obozinski, G., et al.: Consistent probabilistic outputs for protein function prediction. Genome Biology **9**(S6) (2008)
19. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. Bioinformatics **22**(7), 830–836 (2006)
20. Notaro, M., *et al.*: Prediction of Human Phenotype Ontology terms by means of hierarchical ensemble methods. BMC Bioinformatics **18**(1), 1–18 (2017)
21. Zhang, X., *et al.*: CellMarker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Research **Database Issue**, 1–8 (2018)
22. Kiselev, V.Y., Andrews, T.S., Hemberg, M.: Challenges in unsupervised clustering of single-cell RNA-seq data. Nature Reviews Genetics (2019)
23. Barrett, T., *et al.*: NCBI GEO: archive for functional genomics data sets—update. Nucleic Acids Research **41**(D1), 991–995 (2013)
24. Manno, G.L., *et al.*: Molecular diversity of midbrain development in mouse, human, and stem cells. Cell **167**(2), 566–580 (2016)
25. Grapin-Botton, A.: Ductal cells of the pancreas. The International Journal of Biochemistry Cell Biology **37**(3), 504–510 (2005)
26. Subramanian, A., *et al.*: Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences **102**(43), 15545–15550 (2005)
27. Ashburner, M., *et al.*: Gene Ontology: tool for the unification of biology. Nature Genetics **25**, 25–29 (2000)
28. Aran, D., Hu, Z., Butte, A.J.: xCell: digitally portraying the tissue cellular heterogeneity landscape. Genome Biology **18**(220), 1–14 (2017)
29. Newman, A.M., *et al.*: Robust enumeration of cell subsets from tissue expression profiles. Nature Methods **12**(5), 453–457 (2015)
30. Jaitin, D.A., *et al.*: Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. Science **343**(6172), 776–779 (2014)
31. Picelli, S., *et al.*: Smart-seq2 for sensitive full-length transcriptome profiling in single cells. Nature Methods **10**(11), 1096–1098 (2013)
32. Perez-Silva, J.G., Araujo-Voces, M., Quesada, V.: nVenn: generalized, quasi-proportional Venn and Euler diagrams. Bioinformatics **34**(14), 2322–2324 (2018)

Figures

Additional Files

Additional file 1 — Supplement

This file includes Supporting Methods, Supporting Figures, and Supporting Tables.

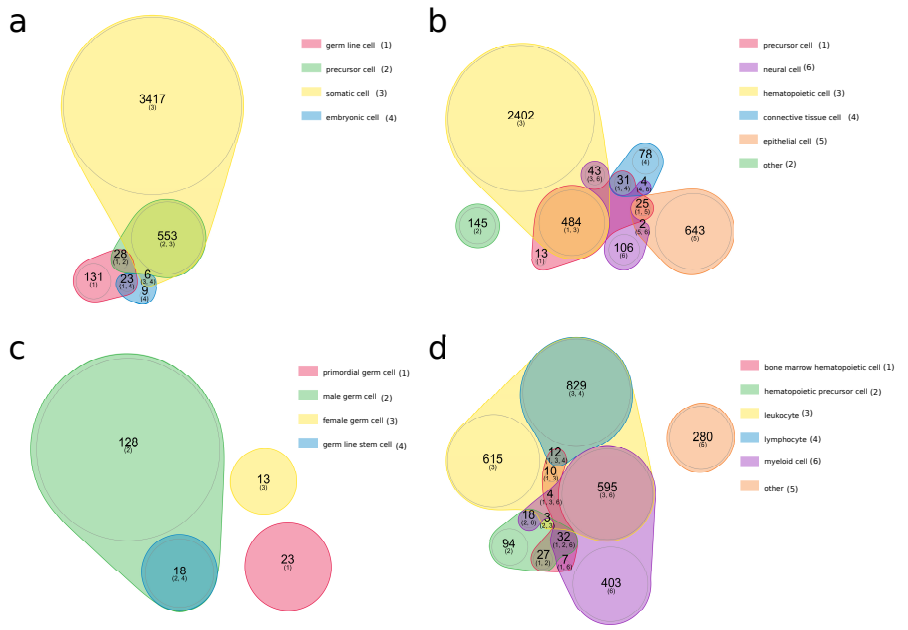

Figure 1 Data set summary. Euler diagrams of the cell types in the bulk RNA-seq data set divided by (**a**) broad cell types, (**b**) somatic cell types, (**c**) germ line cell types, and (**d**) hematopoietic cell types. Diagrams were created with nVenn [32].

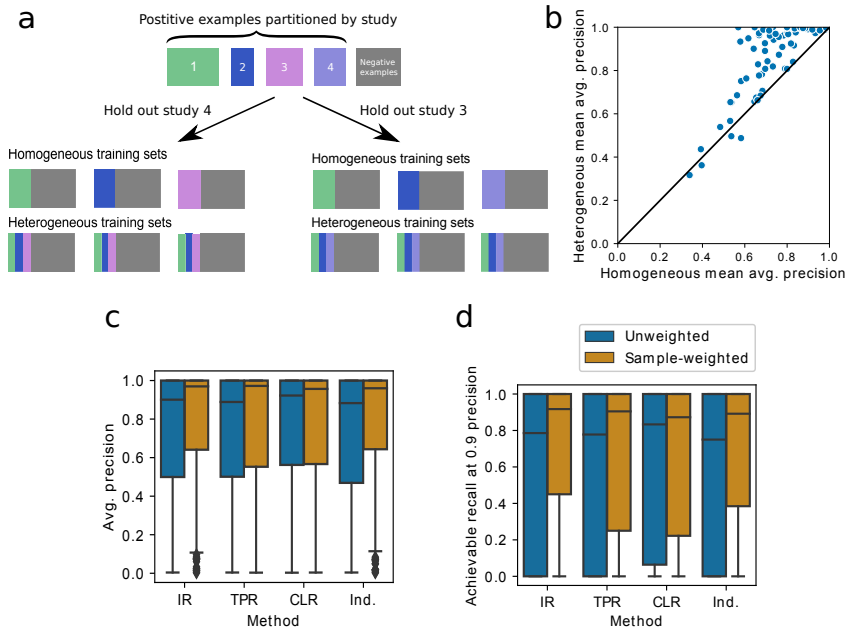

Figure 2 Effects of training on heterogeneous datasets. (a) A schematic illustrating the experimental setup for our investigation into the effects of training on data from multiple studies versus training on data from a single study. For a given cell type, we find all studies that contained at least ten samples for that cell type. The numbered colored rectangles illustrate such samples partitioned by their study. We then hold out each study and construct two sets of training sets – one set of homogeneous training sets and another of heterogeneous training sets. A classifier is trained on each training set and evaluated on data in the held out study. Above, we illustrate the training sets constructed when holding out studies 3 and 4. Each training set uses an identical set of negative examples and identical sized sets of positive examples (the minimum number of samples in a given study partition). (b) Comparing the mean-average precision across cell types between the homogeneously trained classifiers and the heterogeneously trained classifiers on each held out study. (c) Comparing two variants of IR, TPR, CLR, and independent one-vs.-rest classifiers (Ind.). In the first variant (Unweighted), we use the standard log-loss function for each logistic regression classifier. In the second variant (Sample-weighted), we modify the loss function to down-weight each sample by the number of samples in its study. We compare the distributions of average-precision across all cell types. (d) Comparing the distributions over the highest achievable recalls when precision is fixed at 0.9 across all cell types.

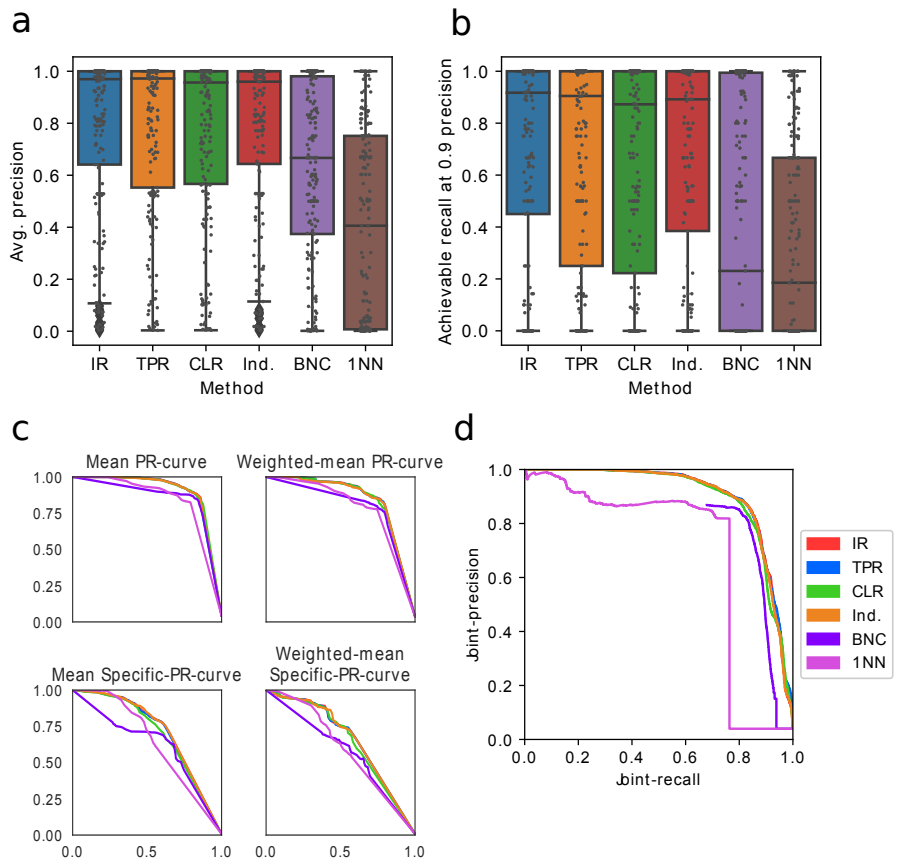

Figure 3 Evaluation on bulk RNA-seq data. (a) Comparison between the distributions of average-precision generated by each method across all cell types. (b) Comparison of the distributions over the highest achievable recalls when precision is fixed at 0.9 across all cell types. (c) Variants of the mean precision-recall curves for comparing the average performance of each method across all samples. (d) The joint-precision recall curves for all methods generated by ranking all sample-cell type output probabilities jointly.

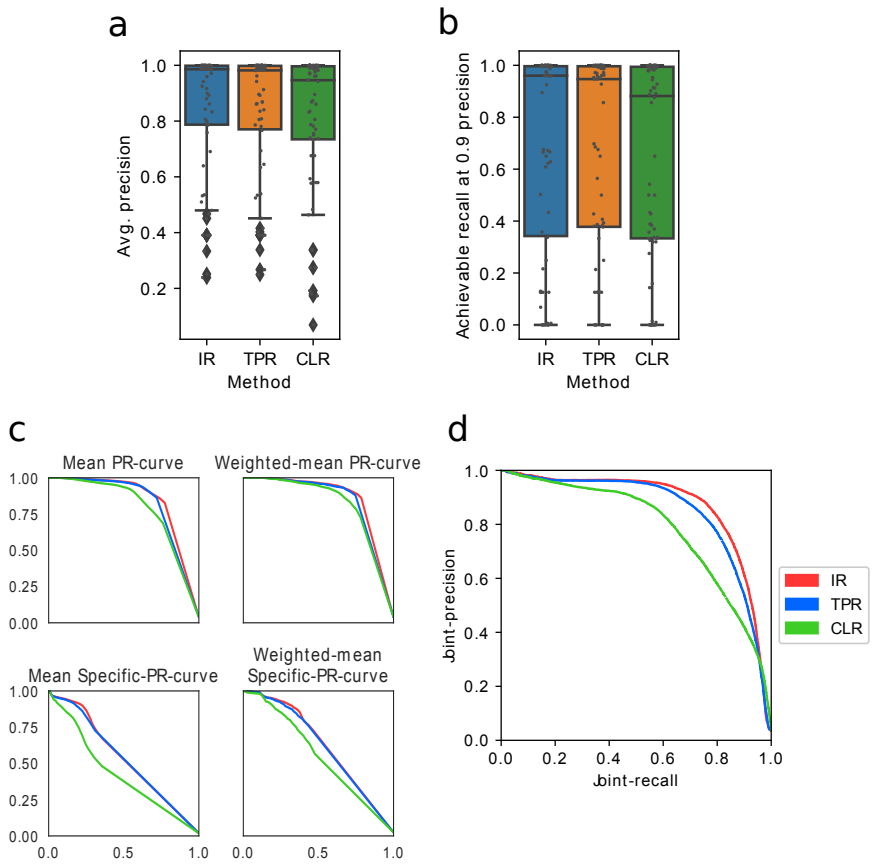

Figure 4 Evaluation on single-cell RNA-seq data. (a) Comparison between the distributions of average-precision generated by each method across all cell types. (b) Comparison of the distributions over the highest achievable recalls when precision is fixed at 0.9 across all cell types. (c) Variants of the mean precision-recall curves for comparing the average performance of each method across all samples. (d) The joint-precision recall curves for all methods generating by ranking all sample-cell type output probabilities jointly.

Figure 5 Example predictions on challenging single-cell samples. Randomly sampled output from the IR classifier on difficult-to-classify single-cell samples. Columns correspond to cells and rows correspond to cell types that appeared in the training data. The intensity of each element is proportional to the output probability for the corresponding sample and cell type. Each element is colored according to the relationship between the sample and the cell type. Green denotes a prediction of a most-specific true cell type (annotated with an 'X') for the sample. Blue denotes a prediction of a less-specific, but true cell type. Purple denotes ambiguous predictions that cannot be verified as correct or incorrect (descendents of the sample's true cell types as well as ancestors of those descendents). Red denotes a likely error (a cell type that is neither a true cell type, descendant of a true cell type, nor ancestor of a descendent of a true cell type). We investigated the predictions of samples that are labeled as general cell types, but not more specific cell types from studies ERP017126 (a) and SRP067844 (b). We also investigated predictions on cell types that did not appear in the training data including embryonic radial glial cells (c), delta cells (d), and ductal cells (e).

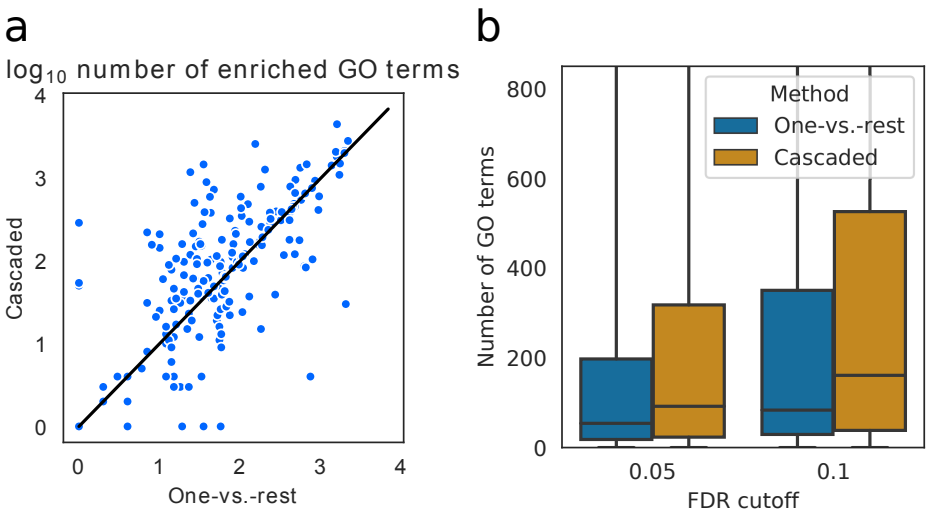

Figure 6 GO enrichment comparison. (a) Comparing the number of GO terms enriched in the ranked list of each binary classifier's coefficients between the cascaded logistic regression and one-vs.-rest frameworks. (b) The distribution of the number of enriched GO terms between these two frameworks using two FDR thresholds for enrichment.