Figures

* Final:
  + <https://drive.google.com/drive/u/0/folders/1Mq0absqX2ayjWbhu8SDevgmq_tbrNiph>
* Individual raw panels:
  + <https://drive.google.com/drive/u/0/folders/1QWV_-dnNl01WktmI127qHJ8Vn2_QJ6K_>

Main points in outline form

* Current annotation/classification methods tend to ignore ontological hierarchy
* Probability propagation strategy makes the annotation ontology aware
* The Hop scoring technique is more suitable to benchmarking ontologically aware annotation

Draft begins on following page

# A supervised ontology-aware cell annotation method for single-cell transcriptomic data

Nimish, Ebtisam, Fedor, Steve McCarroll, Stephen, Mehrtash, Yerdos (???)

## Abstract

Current cell type annotation methods often overlook ontological hierarchies, leading to inconsistencies in outputs and benchmarking. We present a probability propagation strategy that enforces ontological consistency and we demonstrate its effectiveness using a lightweight logistic regression model trained on 42 million annotated cells from the CellxGene data repository. Additionally, we propose a hop-based F1 scoring metric that better reflects ontology-aware annotation performance. We show that the probability propagation strategy can be applied to existing pre-trained models at inference time, improving their performance, and we release our state-of-the-art logistic regression model open source.

## Main

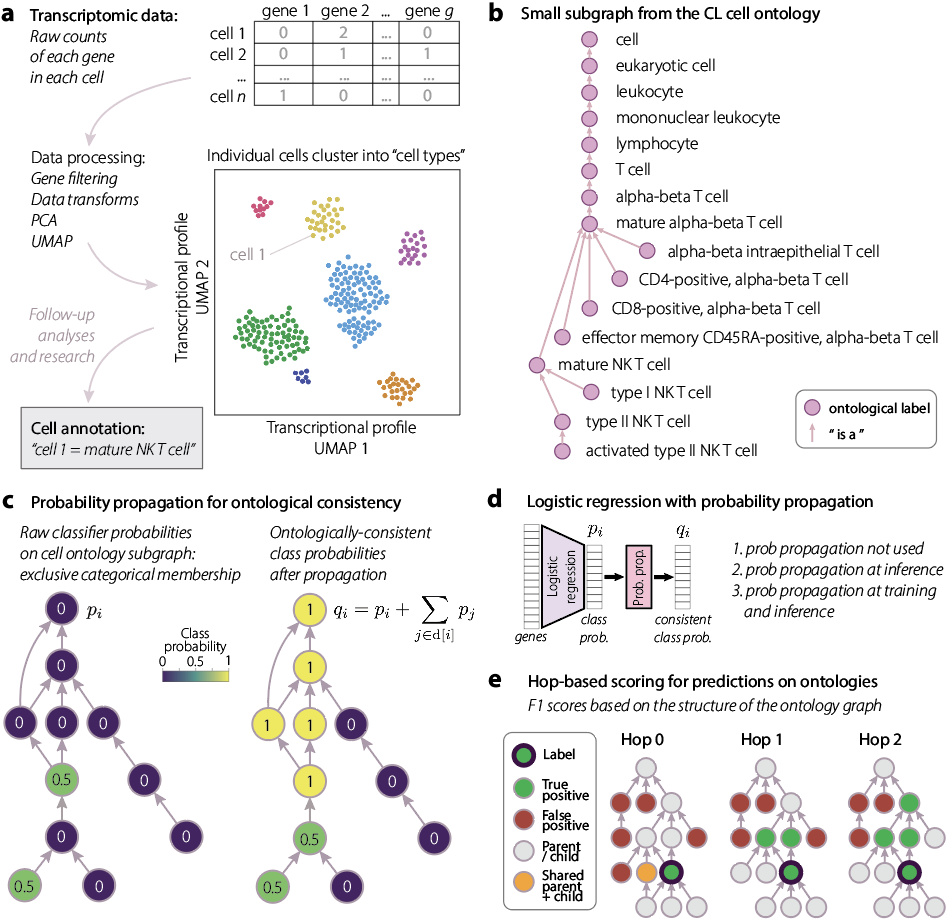


Figure 1. (Legend to go here)

### 

### Intro segment (each heading below becomes a paragraph)

**General background**

* **Introduce the scRNA-seq experiment and data type (Fig. 1a):**

Single-cell RNA sequencing (scRNA-seq) is a powerful experimental technique that enables the profiling of gene expression at the resolution of individual cells, offering insights into cellular heterogeneity that bulk RNA-seq cannot provide. The process typically involves isolating single cells (e.g., using droplet-based methods such as 10x Genomics Chromium), capturing mRNA molecules, reverse transcribing them into cDNA, amplifying the cDNA, and sequencing it to generate a digital expression matrix of genes by cells \cite{zheng2017, macosko2015}. This approach has been instrumental in identifying new cell types, revealing dynamic cell states during development or disease, and constructing cellular atlases of organs and organisms \cite{regev2017, cao2020}. The resulting data provides high-dimensional, sparse, and noisy gene expression profiles, requiring advanced computational methods for downstream analysis \cite{luecken2019}.

* **Introduce the cell type annotation task in scRNA-seq (Fig. 1a):**

Cell type annotation using scRNA-seq data refers to the process of assigning biological identities to cells based on their gene expression profiles. Several supervised and semi-supervised tools have been developed, including SingleR \cite{aran2019singler}, which compares each cell to reference datasets; scPred \cite{alquicira2020scpred}, which uses support vector machines; and scClassify \cite{lin2020scclassify}, which employs ensemble learning. While effective, these methods often assume discrete and non-overlapping cell identities and may not leverage the hierarchical structure of cell types encoded in ontologies like the Cell Ontology \cite{diehl2016cell}. Cell types in biological systems rarely exist as mutually exclusive, clearly separable categories. Instead, they often share overlapping transcriptional programs, especially within lineages or functionally similar groups \cite{haghverdi2018batch, rothe2021computational}. Studies of hematopoiesis, for example, have revealed that progenitor and mature immune cells span a transcriptional continuum that defies hard classification boundaries \cite{paul2015transcriptional}. Assigning a single label per cell can lead to a loss of biologically meaningful relationships between cells.

**Supervised classification is a difficult problem when categories are not mutually exclusive, as in an ontology**

* **Introduce the cell type ontology (Fig. 1b):**

The Cell Ontology (CL) is a structured, expert-curated ontology that defines standardized cell type terms and their biological relationships across species \cite{bard2005ontology, diehl2016cell}. It encodes hierarchical knowledge through directed relationships such as "is\_a" (e.g., a *memory B cell* is\_a *B cell*), enabling inference of "ancestor\_of" and "descendent\_of" links between cell types. For example, since a *plasma cell* is\_a *B cell*, and a *B cell* is\_a *lymphocyte*, the ontology captures that *plasma cell* is a descendent\_of *lymphocyte*.

Despite this hierarchical structure, most current scRNA-seq annotation methods—whether unsupervised (e.g., clustering + marker genes) or supervised (e.g., SingleR, scPred, scClassify)—treat cell types as flat, mutually exclusive labels and ignore the ontology's hierarchy. This can lead to biologically inconsistent annotations, such as misclassifying a transitional *naive B cell* as unrelated to *memory B cells*, even though they are ontologically close. Ignoring these relationships reduces annotation robustness and interpretability, particularly in dynamic or ambiguous cell states \cite{pasquini2021automated, diehl2016cell}.

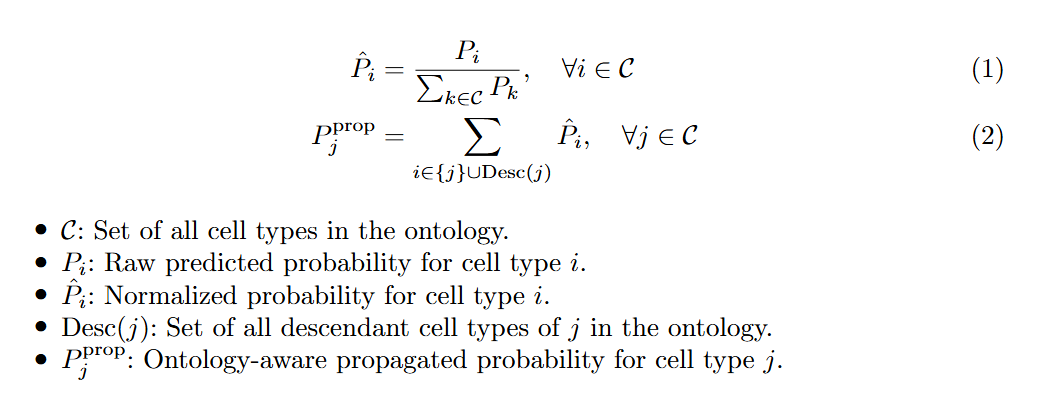
* **Explain the idea of a supervised classifier (logistic regression) and its drawbacks when it comes to categories that come from an ontology (Fig. 1b)**

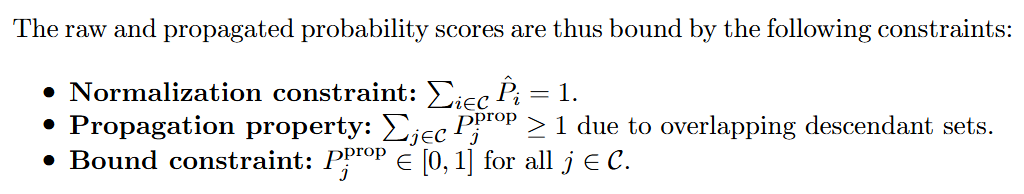
**Probability propagation solves the ontological classification problem by enforcing ontological consistency in output probabilities**

* **Introduce probability propagation (Fig. 1c)**
  + **Show node colors proportional to probabilities before and after propagation:**

Here we introduce an easily adaptable probability score propagation strategy that provides ontologically relevant cell type annotations with labels consistent with the CL labeling convention\footnotemark{\ref{note1}}.

To enforce ontological consistency in predicted cell type probabilities, we introduce the following post-processing probability propagation technique:





Starting from the raw probability scores output by any classifier, we normalize and distribute each cell type’s score upward to all its ancestors in the cell ontology via the "is\_a" hierarchy, ensuring that the overall probability mass remains unchanged. This approach guarantees that each ancestor cell type receives a probability score greater than or equal to that of its descendants, preserving the ontological hierarchy while ensuring all probabilities remain bounded below 1.

### SOCAM model

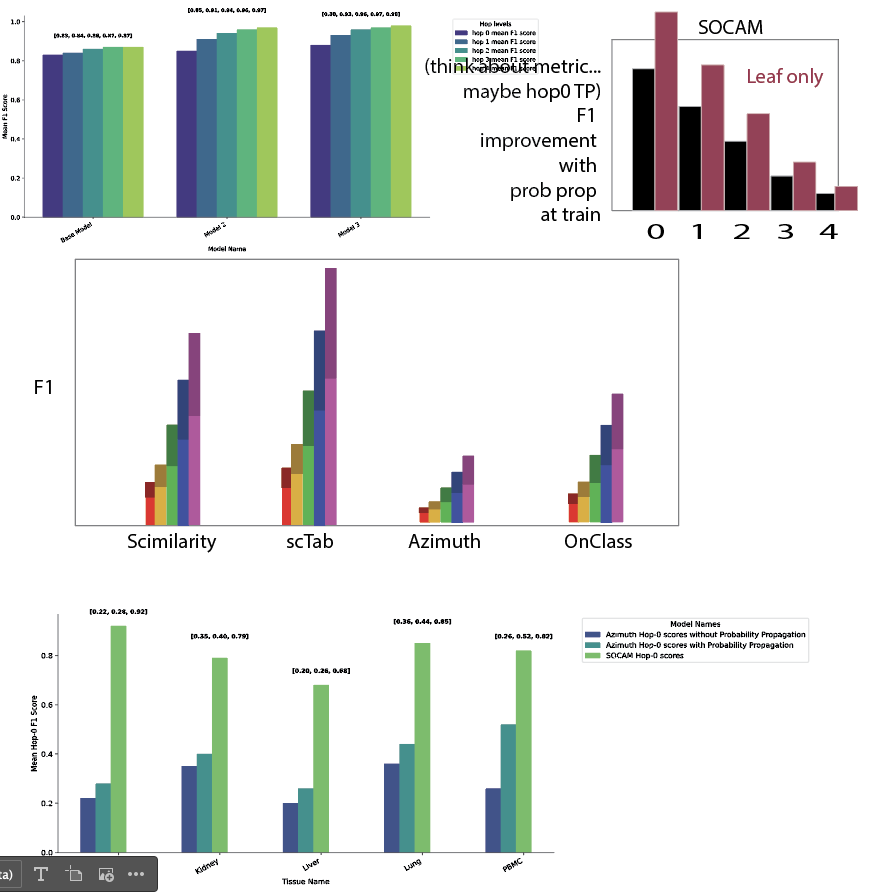


Figure 2. Sketchy sketch of figure 2

**Describe the SOCAM model**

* **Simple, lightweight model: logistic regression:**

To present the effectiveness of the probability score propagation transform, we choose the standard logistic regression model for multi-class classification as the baseline and add the probability score propagation strategy as a final layer after the softmax non-linearity layer to boost ancestor cell type annotation scores. Specifically, we trained and evaluated three variations of the simple logistic regression model:  
a) baseline model: simple logistic regression with no probability score propagation transform,

b) model variation 1: probability score propagation transform added during the inference stage only,

c) model variation 2 (The SOCAM model): probability score propagation transform used as a final layer during training and at inference time.

* + **How many genes used as input, how many params in model:**

The training data we use consists of 20,867 unique genes with the specific gene ensembl ids provided here: <https://github.com/cellarium-ai/cellarium-ml/blob/SOCAM/cellarium/ml/metadata_files/fiitered_gene_ensembl_ids.csv>

Total trainable parameters for the SOCAM model (model variation 3): 13,981,560

* **How was it trained**
  + **Dataset, number of cells, tissues, refer to methods for more:**

For reference scRNA data, CZ CELLxGENE \cite{cziscienceChanZuckerberg} Discover provides access to over 436 diverse single-cell and spatial transcriptomic datasets, encompassing more than 33 million cells and 2,700+ cell types across human and mouse samples. This extensive dataset supports targeted analyses across various diseases, training sets, and gene expression thresholds, accessible through both web tools and APIs. To train, validate and benchmark all model variations, we make use of the single cell data from human samples representing 670 unique cell types coming from 263 unique tissues, subtissues and 19 assays, ensuring a broad representation of cellular diversity for model training and evaluation.

* + **How long did it take (just mention the regression training here) on what hardware**

Following are the hardware and model details used to train these regression models with the 42 million samples in the training set:

* Machine Type: n1-highmem-16
* Accelerator: Nvidia-Tesla-T4 GPU
* Accelerator Count: 2
* Precision: 32-true
* Epochs: 6
* Optimizer: Adam
* Batch Size: 2048
* Number of Workers: 4
* Total training time: 2 days + 14 hrs (62 hrs)
* **What output do you get exactly?**
  + **Probabilities over all ontological categories, where the probabilities are ontologically consistent (Fig. 1c):**

While the baseline logistic regression model provides probability scores over the 670 cell types in the training set (where probabilities add up to 1), the SOCAM model (model variation 3 with probability score propagation transform applied during training) provides hierarchically consistent probability scores for the 670 cell types based on the cell type ontology.

**Describe the benchmarking process: classification in an ontology necessitates nuanced computation of true and false positives**

* **Introduce the idea of hop-based determination of true and false positives based on the ontology (Fig. 1d)**

To evaluate model performance while incorporating the hierarchical structure of the Cell Ontology, we propose a novel hop-based F1 scoring metric. This approach acknowledges that fine-grained cell type annotations may be uncertain or noisy and instead allows for the evaluation of predictions at progressively coarser levels of the ontology. Unlike the standard F1 score, which treats only the exact target label as a true positive, the hop-based metric expands the set of acceptable predictions to include ontologically related ancestors. A hop level of 0 corresponds to the target cell type itself, while a hop level of 1 includes its immediate parent(s) in the ontology. By evaluating predictions across increasing hop levels, this metric enables a more nuanced assessment of classification performance, particularly in cases where hierarchical proximity may be more biologically informative than exact matches.

* **Clearly spell out the rules for TP and FP determination (Fig. 1d)**

Rules for calculating the evaluation metric: (with formulas in overleaf)

The *TP* at hop k is defined as: (formula)

That is, the highest probability score among all cell types in the True Positive set at hop level k. (formula)

Similarly, the *FP* value at hop k is defined as:

That is, the highest probability score among all cell types in the False Positive set. This set includes cell types that are not ancestors, descendants, or siblings of the target cell type.

Precision at hop k is defined as: (formula)

Recall at hop k is given by: (formula)

Finally, the F1-score at hop k is calculated as: (formula)

* Give formula for F1 score based on TP and FP
* What metric was computed (hop-based F1 scores)
* Which other methods were included in benchmarking
  + Mention what each of those models is, very briefly

### Benchmarking results segment (each heading below becomes a paragraph)

**Probability propagation during training is best, but it is also beneficial when applied only at inference**

* Table with SOCAM and other models using probability propagation (Table 1) - https://docs.google.com/document/d/1-JAgayIJJKzth40d0odsXRt1JUepxs0FPoJUOx1Kssk/edit?usp=sharing
  + SOCAM is the top performer
* SOCAM model results (Fig. 2a) are best when probability propagation is applied during training as well
* Azimuth improves when probability propagation is applied at inference (Fig. 2b)
* OnClass improves when probability propagation is applied at inference (Fig. 2c)
* ScTab improves when probability propagation is applied at inference (Fig. 2d)

**Brief discussion about how to find and use SOCAM and limitations**

* Brief recap:
  + Probability propagation improves existing trained models
  + Simple linear regression model trained with probability propagation beats benchmarks
  + The output is a nice consistent probability for each label in the ontology
    - Far easier to interpret than some existing simple classifiers
* Limitations
  + Human data only
  + Generalization across tissue, assay, and donor
    - But what do we mean here exactly??
  + Mention that the benchmarks were not necessarily trained on the same data SOCAM was
    - (But benchmarking against their published model seems like a sensible thing to do… it’s just that maybe their models could be improved by training on this data… we’re just not doing that here)
* Available on github
  + Trained model where? Huggingface?

## Methods (each bullet is a paragraph or more, but keep everything as brief as possible)

* The CL ontology
* Data
  + (Where it comes from, how many cells, what was the date it was released, how many tissues, how many cell types)
* Cell type filtering
  + (What filters were applied to limit the input data and why)
* Splitting data into train and test
* Feature selection and data preprocessing
  + Describe the onepass model and its training
* Any further details on probability propagation?
  + Could potentially include the explicit table of calculation for several example nodes, if we think it’s necessary
* Any further details on the hop-based F1 score?
  + Explicit algorithm for determining TP and FP nodes
* Benchmarking
  + (Details of running the other methods benchmarked… what was done if outputs were not using the CL ontology exactly, any parameter choices, where trained models were obtained, etc.)
  + (How exactly was probability propagation applied at inference time to benchmarking models?)
* SOCAM code
  + (Was written using the Cellarium ML library, explain what that is.)
* Training details for SOCAM
  + (Hyperparamter choices, hardware, training time)
  + Can describe cellarium-ml
  + Can describe the vertex-ai setup for training runs