Supervised Cell type classification

January 7, 2025

Contents

1 Abstract

The advent of single-cell technologies that produce increasingly complex cellular level datasets has significantly increased the need for accurate and robust cell type classification techniques to maximize the relevance of these datasets. Apart from the reliance on complex preprocessing steps and lack of scalability, the current classification techniques tend to overlook the cell type dependency through their hierarchical relationships. To tackle these specific shortcomings, we introduce Onto-Classifier, a lightweight logistic regression based cell-type classifier that accurately predicts cell type probabilities based on scRNA data while enforcing the hierarchical nature of cell types represented through their ontology.

2 Introduction

Single-cell technologies produce extensive data for understanding complex biological systems at a cellular level. Along with providing insights into disease mechanisms, tissue heterogeneity and cellular functions, accurate classification of cell types facilitates the discovery of biomarkers and novel cell types[1]. These discoveries play a crucial role in assessing the cellular compositions of tumors [2], immune responses and help identify differential pathways within oncology [3]. As single-cell technologies continue to produce increasingly complex and large datasets, robust and scalable classification techniques would become essential to maximize data relevance.

Recently developed cell type classification techniques through single-cell RNA sequencing data often require sophisticated pre-processing methods to mitigate the high dimensional and sparse nature of the data. Popular unsupervised clustering techniques such as Suerat's graph-based clustering [4] or Scanpy's Louvain algorithm [5] rely heavily on computationally intensive techniques such as PCA or t-SNE for clustering in low-dimensional spaces. While advanced algorithms using consensus clustering and multiple distance metrics such as SC3 [6] significantly enhance robustness but due to their demand for large computational resource, they suffer tremendously when scaling for large datasets.

Therefore with the unavailability of robust and scalable cell type classification techniques, this project aims at introducing some lightweight and relatively robust versions of the previously develop classification techniques using variants of the supervised logistic regression model with modifications to accommodate the unique characteristics of available scRNA data. Specifically, the following structural constraints and unique characteristics of available scRNA data are taken into consideration:

1. Cell Type Hierarchy: $P_{AncestorCellType} \geq P_{DescendantCellType}$ If a query cell type has a certain probability of being categorized as a particular cell type ($P_{DescendantCellType}$), then the probability ($P_{AncestorCellType}$), of the query cell type being categorized as an ancestor of the $DescendantCellType$, is at least as much as this probability ($P_{DescendantCellType}$).
2. Unique Cell Type Count: $|\text{Unique\_Cell\_Types}_{\text{Ontology}}| > |\text{Unique\_Cell\_Types}_{\text{Reference Data}}|$
3. Uncertainty in granularity of ground-truth labels: In the reference data, if a cell type target is labeled to be from $\{AncestorCellType\}$, then it is possible that the actual target could belong to $\{descendantCellType\}$.

3 Related Work

3.1 Traditional methods

3.2 Automatic cell-type identification methods

Prior research work on automatic cell-type identification methods are categorized into three categories: lazy learning, eager learning and marker learning based on the usage of gene-cell or cell-gene expression matrix and canonical cell markers [7].

3.2.1 Lazy Learning methods

Current lazy learning methods such as scmap-cell, CellAtlasSearch, CellFishing.jl, and CELLBLAST work by comparing cell types against the presented cells in the dataset to find the closest match of the cell type by calculating its nearest neighbors [7].

3.2.2 Eager learning methods

Existing eager learning methods such as scPred, SingleCellNet, SingleR, Superscan, Garnett, MarkerCount, MARS, scCapsNet, scPretrain, scVI, Seurat, ACTINN, CaSTLe, CHETAH, clustify, scClassifR, SciBet, scID, scLearn, scmap-cluster, scMatch and scHPL, in contrast to lazy learning methods, work by organizing the data into distinct groups, and then the classification of a new cell type is determined by assigning it to the closest group [7].

3.2.3 Marker learning methods

Recent marker learning methods such as MarkerCount, scCATCH, CellAssign, DigitalCellSorter, SCINA, SCSA and scTyper work by classifying cell types based on specific genetic markers (distinctive genes) [7].

3.3 Unsupervised methods

Various unsupervised scRNA-seq clustering methods exist, like graph-based, hierarchical, and partition clustering. Users often manually set parameters, and are required to preprocess data significantly to be used with established methods and this impacting results. Clustering approaches and parameter choices can influence downstream interpretations [7].

3.3.1 Cell annotation service (CAS)

CAS is a machine learning based search engine system that specializes in vector database sorting and searching for efficient and fast cell similarity search and identification. This system was trained on 50 million cells from CZ CELLXGENE, and it reduced the cell annotation process by hours. It utilizes a large cell vector embedding model and embeds user data into a unified latent space. The method performs fast approximate nearest neighbor searches to identify similar cells from a vast pool. However, this method limits users' query quota for their cell data, and it performs PCA, which may eliminate vital biological features.

3.4 Supervised Classification Methods

3.4.1 Data-centric cell ontology methods

Inconsistent terminology of cells across different scientists and practitioners remains an issue. There are methods to tackle such challenges using a standardized cell-type hierarchical structure of a family tree terminology based on the relationships between cell types. A study by [7] proposed the OnClass method, the first algorithm to identify marker genes for every cell type in the Cell Ontology, as well as to analyze the data based on the cell ontology structure. OnClass uses a bilinear neural network to classify the cell based on the cell ontology term for unseen cell types.

4 Data Description

Source of Ontology and scRNA Reference Data

Ontology

In defining the ground-truth labels as well as model predictions from the sc-RNA data to classify the cell types, the systematically defined, controlled and structured Cell Onotlogy (CL) vocabulary is used. This provides well organized defnitions for cell types based on their characteristics, lineage and functional roles [8]. The "CL" nomenclature uniquely identifies each cell type and represents the hierarchical relationship between the cell types (parent-descendant).

Specifically, the Cell Ontology developed under strict quality, consistency and semantic interoperability by the Open Biological and Biomedical Ontology (OBO) Foundry is used as the standard in model trained as part of this project. This effort is a community-driven and collaborative operation and so the Cell Ontology structure goes through regular updates. Therefore, to remain consistent with ontology used across all model variations, we make use of the ***2024-01-04*** [1] release throughout the project.

scRNA Reference Data

CZ CELLxGENE[9] Discover offers 436+ diverse single-cell and spatial transcriptomic datasets (33M+ cells, 2700+ cell types; human & mouse) enabling targeted analyses across diseases, training sets, and gene expression thresholds, accessible via web tools and APIs.

The training dataset consists of a total of 670 unique cell types.

Gene Count and Cell Type Filtering

Gene Filtering

All gene counts/genes in the training extract are filtered using the following steps:

[1] `https://github.com/obophenotype/cell-ontology/releases/download/v2024-01-04/cl.owl`

1. Genes are initially subset to the 10x GRCh38 reference genes set [2].

2. The remaining genes are then filtered such that the mRNA count per million (often called "TPM") is less than 1. Transcripts Per Million is used to normalize reads such that the sum of all reads is exactly 1 million [10]. Cell type classification relies significantly on connecting marker genes with specific expression patterns to certain cell types. Attributes of TPM such as length-normalized measure of expressions and scaling based on total number of sequence reads (library size) ensure accurate comparison of gene expressions levels across cell types and make the gene expressions comparable across datasets[11, 12, 13, 14].

Cell Type Filtering - a total of 2604 target cell types

The following steps were taken to filter cell types:

1. When creating the extract, any cell types labeled "unknown" are removed as they do not provide useful information for supervised learning.

2. Also, any samples labeled with "Cell" as cell type are removed as "Cell" is the root node (making it an ancestor to all cell types). Therefore, every cell type can be classified to be of "Cell" type.

3. total unique cell types in ontology = 2914, Total unique cell types without "Cell" label: 2914 - 1 = 2913

4. total unique cell types in extract = 670

5. Remove cell types that do not have any ancestor or descendant (187 such cell types): Total cell type count = 2913 - 187 = 2726

6. Remove any cell types that do not have at least 1 descendant or 1 ancestor represented in the extract: Total cell type count = 2726 - 113 = 2613

5 Proposed Model Variations

Given that the goal of the project is to work with a function mapping gene expression data to cell types such that it is relatively robust and easily scalable (lightweight), the standard linear logistic regression model for multi-class classification is chosen as the base. Building on this base model, further additions are made to the consecutive variations of the base model to accommodate for the unique data characteristics.

5.1 Baseline Model: Logistic Regression Model for Multi-Class Classification

Treating this task as a multi-class classification problem, the standard logistic regression (linear model) was trained as the baseline. The softmax activation function applied over the model logits provided the mapping from gene expression data to probability outputs of the query cell data being categorized as one of the 2613 reference cell types.

Softmax(logits) = (P1,P2,P3,....,Pc), where

$$\sum_{i=1}^{C} P_i = 1$$

, 'C' = Total number of reference cell types, $0 \leq P_i \leq 1$

The standard Categorical Cross Entropy Loss over the specified target cell type was used to train this model:

$$Loss = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{C} y_{i,k} log(\pi_{i,k})$$

, where $y_{i,k}$ is a binary indicator (1 if sample i belongs to class k, 0 otherwise); and $\pi_{i,k}$ is the probability output for sample i belonging to class k.

This model did not involve specific application of constraints to enforce the hierarchical characteristic between the descendant and ancestor cell types and was therefore not expected to align its probability outputs well with the descendant/ancestors relationship between cell types.

[2] https://github.com/cellarium-ai/cellarium-cas/tree/main/cellarium/cas/assets/cellarium_cas_tx_pca_002_grch38_2020_a.json

5.2 Model Variation 2: Baseline Model + Probability Propagation during inference

To adapt to the hierarchy constraint: The softmax function based probability outputs from the baseline model were updated to calculate new probabilities (π) as follows:

$$\pi_j = \frac{P_j + \sum_{i \in descendants(j)} P_i}{\sum P_j}$$

This method propagated the individual probabilities of all descendant nodes (Cell types) to their ancestors and then normalized the new probabilities.

The probability predictions from the trained Baseline Model 5.1 were propagated according to the equation above during inference to meet the hierarchy constraint of the given data.

5.3 Model Variation 3: Update on model 2 to perform probability propagation during training

Progressing one step further from the previous model variations, this model variation adapts the probability propagation step during the training phase as well. Adding the probability propagation step while training enforces a soft constraint for the model to adapt to the hierarchical characteristic of the cell types.

The model is expected to provide comparatively higher probability predictions for target cell types (ground-truth cell types) that are not leaf nodes and could possibly learn to assign most weight to these leaf nodes as it would expect the probabilities to be propagated.

Due to this probability propagation step, this model variation would be expected to perform at least as well as the baseline model in terms of accuracy of predictions but due to the same probability propagation measure, the model would also propagate probabilities for cell types that do not belong to the same subtree as the target cell. For this reason, the model may not perform as well in terms of F1-scores in comparison to the previous model variations.

5.4 Model Variation 4: Update on model 3 to perform probability propagation during training, using binary cross-entropy loss and performing multi-label multi-class classification where all ancestors of the target cell type are also made to be targets

As a further update on model variation 3, and to better accommodate the hierarchical constraint of the data while keeping the probability outputs for cell types that don't belong to the {target cell type ∪ ancestors of target cell type} minimal, the target cell type and its ancestors are labeled to be positive targets while all other cell types are labeled to be negative targets, essentially treating this as a multi-label multi-class classification problem with Binary Cross Entropy Loss.

$$BinaryCrossEntropyLoss = -\frac{-1}{N}\sum_{i=1}^{N} y_i log(P_i) + 1 - y_i log(1 - P_i)$$

5.5 Model Variation 5: Update on model 4 to perform probability propagation during training, using binary cross-entropy loss, performing multi-label multi class classification with all ascendants of the target cell type also made to be targets, α parameter to control how much probability outputs for descendants are penalized

This model variation is specifically designed to accommodate the last data characteristic wherein there is uncertainty in the granularity of the target cell type.

Weighing the loss term for the descendant cells helps control the probability assigned to these descendant cell types while still maintaining the hierarchy characteristic of $P_{AncestorCellType} \geq P_{descendantCellType}$
The loss term here is broken into three sub-terms where $Loss_A$ specifies the loss over the target and its ancestor cell types, $Loss_B$ specifies the loss over the descendants to the target cell type and $Loss_C$ measures the loss over cell types that do not belong to the sub-tree containing the target cell type.

$$Loss = \frac{Loss_A}{|A|} + \alpha\frac{Loss_B}{|B|} + \frac{Loss_C}{|C|}$$

Each loss term here is normalized by the count of cell types in each of the groups, ensuring equal contribution by each group to the overall loss.

Specifically, we train and test this model variation with α values of 0, 0.5 and 1 to understand the trend in performance for differently punishing the probability outputs for group B. Also, each loss term individually makes use of the binary cross entropy loss function.

$$BinaryCross - EntropyLoss_{A,B,C} = -\frac{-1}{N}\sum_{i=1}^{N} y_i log(P_i) + 1 - y_i log(1 - P_i)$$

6 Experimental Setup

6.1 Data Preprocessing:

- Step 1 (Total Count Normalization): The total count across each cell types total expressions is rescaled to add up to 10,000 to address variability in sequencing depth. A small epsilon value of 1e-6 is also chosen to avoid any divide by zero errors during the scaling process for this sparse data.
- Step 2 (Log 1p normalization): Due to a high variability in gene expression and over representation of lowly expressed genes, sc-RNA data often exhibits a heavy-tailed distribution [15] which is mitigated though this log(1+x) transformation.
- Step 3 (Divide By Scale): To eliminate any bias that arises from gene-specific variability, all gene expression values are centered around a common scale using the division by median scaling.

6.2 Experiment Details

Each of the model variations is trained to completion with the following setup:

- Total Epochs: 6
- Batch Size: 2048
- No. of Samples: (10000*2987) + 5442
- Maximum Learning Rate: 5e-3
- Initial Learning Rate: 0.0001
- Learning Rate Scheduler: Linear Warmup to maximum LR during 20% initial steps and then Cosine Annealing to 0 for the remaining 80% of steps.
- W_prior_scale: 1e-2.
- Optimizer: Adam

6.3 Evaluation Metrics

To evaluate the model performance, the F1-scores are calculated at various hop-levels for each query cell type. A *hop-level* of 0 is defined to be the exact target cell type itself, a *hop-level* of 1 is then defined to be the set of cell types that are 1 hop above the target cell (i.e: all cell types that form the immediate parent of the target cell type) and so on as shown in the figure below:

Here the *True Positive* value at hop 'k' is calculated to be:

$$TP_k = max\{score(node)|nodeinhop_k truepositiveset\}$$

Similary, the *False Positive* value at hop 'k' is calculated to be:

$$FP_k = max\{score(node)|nodeinhop_k falsepositiveset\}$$

Precision at hop 'k' is then calculated to be:

$$Precision_k = TP_k/TP_k + FP_k$$

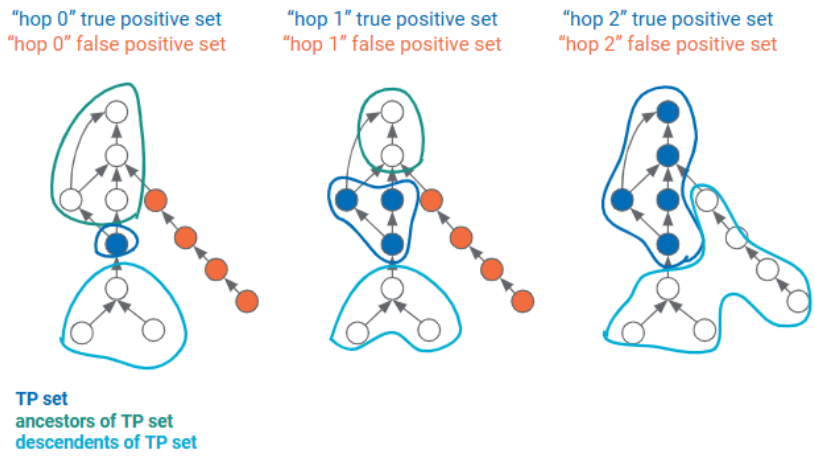


Figure 1: hop-level based F1 score calculation

, Similarly, *Recall* at hop 'k' is calculated to be:

$$Recall_k = TP_k$$

, And *F1-score* at hop 'k' is calculated to be:

$$F1_k = 2 * Precision_k * Recall_k / Precision_k + Recall_k$$

Based on these equations, the following set of metrics have been used to evaluate the performances of each of the model variations:

- Hop-based mean F1 Scores
- Hop-based F1 score box plots
- Hop-based mean F1 scores by assay in appendix
- Hop-based mean F1 scores by tissue in appendix
- Mean Confidence Score (mean recall at hop-0)
- Mean Min/Max descendant cell type score

7 Results and Discussion

7.1 Comparison across model variations

Mean Confidence Score (mean recall at hop-0)

From the results expressed in the table above, it is evident that model variation 3 5.3 wherein the multi-class classification model model is trained with single targets and with a soft-constraint to promote cell type hierarchy performs best in predicting individual target cell types with high confidence. Also important to note here is that model variation 5[5.5] with $\alpha = 0$ (i.e no loss term for the descendant cell types group - group B), performs better than both the baseline model with and without propagation during inference, suggesting the importance of applying the soft probability propagation constraint during training as well as the importance of minimally punishing probability outputs for descendant cell types. Model variation 4[5.4] on the other hand, tends to rely on the number of positive and negative samples during the training phase and may have suffered in its predictions due to data imbalance during training.

Hop-based Mean max FP Scores

This table provides an insight into the maximum confidence with which each model variation predicts a cell type that does not belong to the sub-tree of the target cell type for different hop-levels. An important factor in the base model performing best in this case is the probability propagation step applied during training and inference for the other variations. While these

models	Mean Confidence Score (mean recall at hop-0)	
LR_Model_1	0.803344	
LR_Model_2	0.839566	
LR_Model_3	0.879354	
LR_Model_4	0.730891	
LR_Model_5	0.849697	
LR_Model_6	0.756674	
LR_Model_7	0.735711	

Table 1: Mean Confidence Score (mean recall at hop-0)

models	hop_0_mean_fp	hop_1_mean_fp	hop_2_mean_fp	hop_3_mean_fp	hop_4_mean_fp
LR_Model_1	0.088203	0.049393	0.030604	0.015111	0.007896
LR_Model_2	0.112003	0.078503	0.051584	0.026832	0.013149
LR_Model_3	0.442756	0.155348	0.061989	0.027778	0.013824
LR_Model_4	0.261027	0.141276	0.064976	0.036005	0.016631
LR_Model_5	0.41086	0.148979	0.073095	0.03314	0.017893
LR_Model_6	0.34693	0.1482	0.087829	0.049692	0.025853
LR_Model_7	0.32317	0.151268	0.088785	0.05254	0.031944

Table 2: Hop-based Mean max FP Scores

propagations help in improving the prediction probabilities for target cell types and in enforcing the hierarchy constraint, they also contribute to propagations for group C, (i.e the cell types outside the sub-tree to which the target cell type belongs in the ontology).
Another interesting observation is also that of model variation 4 [5.4] and the different variations of model variation 5 [5.5]. The loss function in model variation 4 is influenced by the number of positive and negative samples, unlike variations of model 5 where the loss terms are normalized to have equal contribution from positive and negative targets. Possibly due to the presence of more negative targets in the training set, model 4 seems to have outperformed model variations 3 and 5, coming extremely close to the performance of the model variation 2 [5.2] as the hops increase.

Mean Hop-k F1 Scores

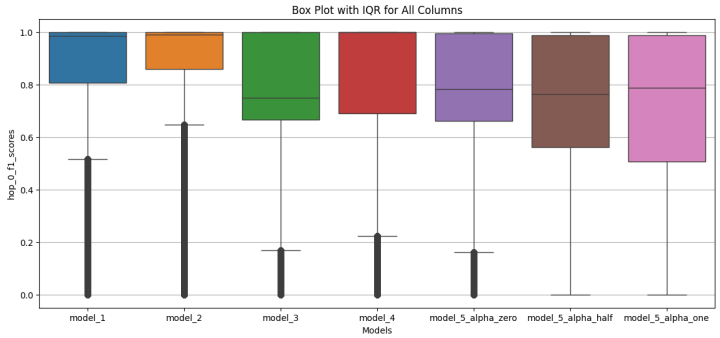
Model Variation	hop_0_mean_f1_score	hop_1_mean_f1_score	hop_2_mean_f1_score	hop_3_mean_f1_score	hop_4_mean_f1_score
Base Model	0.826389	0.84306	0.854438	0.865738	0.868199
Model 2 (Base Model + PP during inference)	0.848696	0.902162	0.935279	0.958369	0.970527
Model Variation 3 (PP during training)	0.769061	0.897515	0.945401	0.964588	0.974719
Model Variation 4 (multiple targets, PP during training and BCE Loss)	0.788385	0.867767	0.912641	0.946162	0.959932
Model Variation 5 (variation 4 + alpha=0 for descendant cells)	0.759236	0.892897	0.938031	0.960561	0.971638
Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)	0.71233	0.881797	0.92662	0.951998	0.967503
Model Variation 5 (variation 4 + alpha=1 for descendant cells)	0.702362	0.873516	0.92371	0.949644	0.963991

Table 3: Mean Hop-k F1 Scores

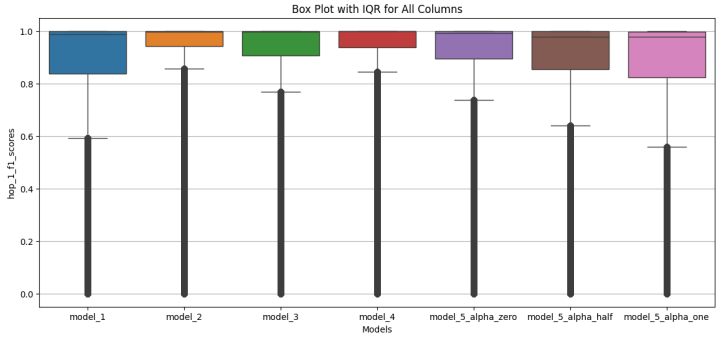
As in evident from the table above, model variation 2 [5.2] outperforms the other model variations at initial hop levels of 0 and 1 and as discussed in the prior results, an important contributor to this result is the soft-constraint of probability propagation applied during the training phase for model variations 3 and higher leading to a relatively large max FP score. However, for hop levels 2 and above, model variation 3 [5.3] outperforms all other model and a very important contributor to this result would be that as hop levels increase, max TP outputs near 1 due to propagation which leads to the difference in TP scores significantly outweighing the difference in FP scores between model variation 3 and 2.

Another trend observed throughout the results for different α values in model variation 5 [5.5] is also that an α value of 0 where there is no punishment on the probability outputs of descendant cell types, tends to outperform the other variations for the given F1 score metrics on this specific validation dataset.

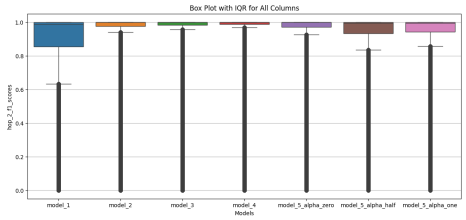
Hop-based F1 Score Box Plots



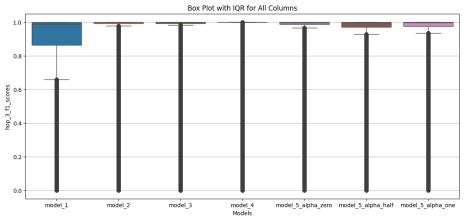
(a) Hop-0



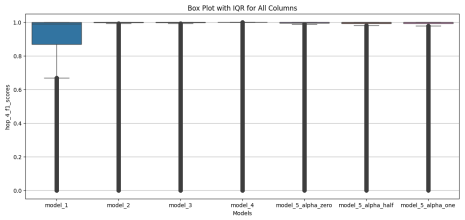
(b) Hop-1



(c) Hop-2



(d) Hop-3



(e) Hop-4

From the box plots above, especially for the first 2 hop levels, the superior performance of model variation 2 [5.2] across all classes is evident through the larger median and small box size (IQR). Having the median line closer to the upper end of the box plot also supports the better performance conclusion for model variations 1 [5.1] and 2 [5.2]. However, almost all models seem to have longer whisker lengths for the lower 25% in the box plots at hop 0 indicating greater variability in performance compared to the plots at higher hop levels. This difference for model variations 2 and higher could be attributed to the probability propagation step significantly improving model performances at higher hop levels.

Hop-based Mean F1 Score comparison with CAS

models	hop_0_f1_scores	hop_1_f1_scores	hop_2_f1_scores	hop_3_f1_scores	hop_4_f1_scores
CAS	0.347735	0.678512	0.880135	0.954549	0.972794
LR_Model_1	0.850112	0.86959	0.877648	0.883641	0.88755
LR_Model_2	0.859346	0.917405	0.944617	0.962194	0.97171
LR_Model_3	0.771108	0.916294	0.954775	0.97022	0.976354
LR_Model_4	0.81145	0.762727	0.8394	0.928168	0.938944
LR_Model_5	0.757178	0.904742	0.948133	0.966965	0.973183
LR_Model_6	0.747154	0.892797	0.935998	0.96071	0.968444
LR_Model_7	0.741475	0.87714	0.928774	0.957523	0.965644

References

[1] V. Svensson, K. N. Natarajan, L. H. Ly, R. J. Miragaia, C. Labalette, I. C. Macaulay, and S. A. Teichmann, "Power analysis of single-cell rna-sequencing experiments," *Nature Methods*, vol. 15, no. 5, pp. 381–387, 2018.

[2] N. Aizarani, A. Saviano, R. Sagar, L. Mailly, S. Durand, J.-S. Herman, and J. Zucman-Rossi, "A human liver cell atlas reveals heterogeneity and epithelial progenitors," *Nature*, vol. 572, no. 7768, pp. 199–204, 2019.

[3] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, and J. L. Rinn, "Pseudotime analysis of single-cell rna-seq reveals cellular trajectories," *Nature Biotechnology*, vol. 32, no. 4, pp. 381–386, 2014.

[4] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija, "Integrating single-cell transcriptomic data across different conditions, technologies, and species," *Nature Biotechnology*, vol. 36, no. 5, pp. 411–420, 2018.

[5] F. A. Wolf, P. Angerer, and F. J. Theis, "Scanpy: large-scale single-cell gene expression data analysis," *Genome Biology*, vol. 19, no. 1, p. 15, 2018.

[6] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. S. Andrews, A. Yiu, T. Chandra, and M. Hemberg, "Sc3: consensus clustering of single-cell rna-seq data," *Nature Methods*, vol. 14, no. 5, pp. 483–486, 2017.

[7] B. Xie, Q. Jiang, A. Mora, and X. Li, "Automatic cell type identification methods for single-cell rna sequencing," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 5874–5887, 2021.

[8] A. D. Diehl, T. F. Meehan, Y. M. Bradford, M. H. Brush, W. M. Dahdul, D. S. Dougall, Y. He, D. Osumi-Sutherland, A. Ruttenberg, S. Sarntivijai, C. E. Van Slyke, N. A. Vasilevsky, M. A. Haendel, J. A. Blake, and C. J. Mungall, "The cell ontology 2016: enhanced content, modularization, and ontology interoperability," *J. Biomed. Semantics*, vol. 7, p. 44, July 2016.

[9] "Chan Zuckerberg CELLxGENE Discover — cellxgene.cziscience.com." `https://cellxgene.cziscience.com/`. [Accessed 19-12-2024].

[10] R. P. Hilbert Lam Yuen In, "Transcripts per million ratio: applying distribution-aware normalisation over the popular tpm method," *arXiv*, 2022.

[11] B. Li and C. N. Dewey, "Rna-seq gene expression estimation with read mapping uncertainty," *Bioinformatics*, vol. 27, no. 8, pp. 1067–1075, 2010.

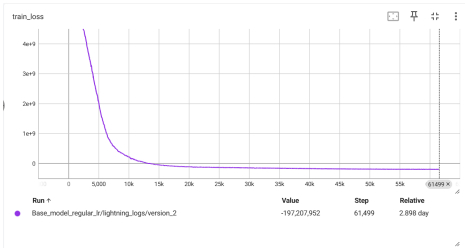
[12] G. P. Wagner, K. Kin, and V. J. Lynch, "Measurement of mrna abundance using rna-seq data: Rpkm measure is inconsistent among samples," *Theory in Biosciences*, vol. 131, no. 4, pp. 281–285, 2012.

[13] L. Zappia, B. Phipson, and A. Oshlack, "Exploring the single-cell rna-seq analysis landscape with the scrna-tools database," *PLoS Computational Biology*, vol. 14, no. 6, p. e1006245, 2018.

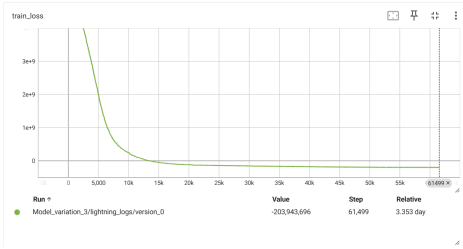
[14] A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, R. Sánchez, A. Del Carrater, and A. Mortazavi, "A survey of best practices for rna-seq data analysis," *Genome Biology*, vol. 17, no. 1, p. 13, 2016.

[15] A. T. Lun, D. J. McCarthy, and J. C. Marioni, "A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor," *F1000Research*, vol. 5, p. 2122, 2016.

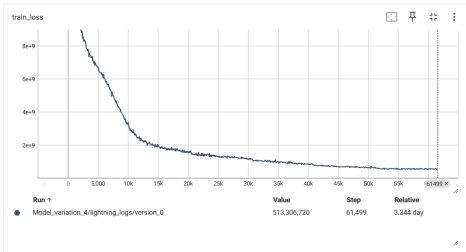
A Training loss curve outputs for each model variation



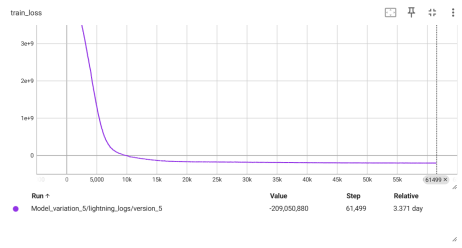
(a) Base Model



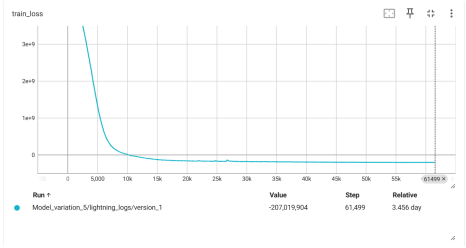
(b) Model Variation 3 (PP during training)



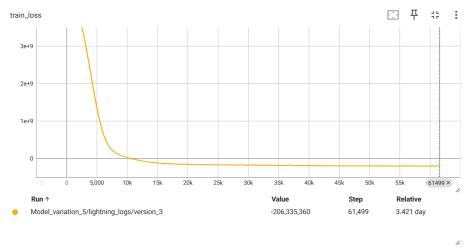
(c) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(d) Model Variation 5 (variation 4 + alpha=0 for descendant cells)



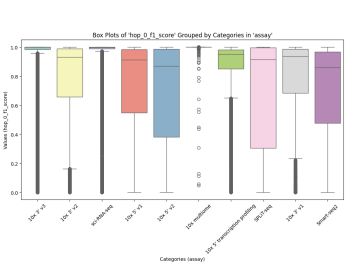
(e) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)



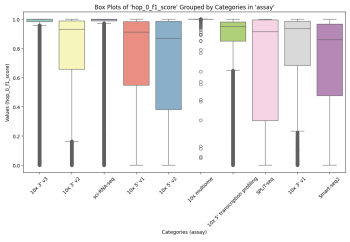
(f) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

B Hop-based F1 Score Box Plots by Assay for each model

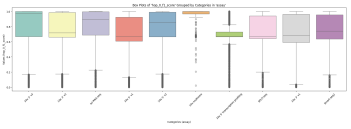
Hop 0 F1-Score Box Plots



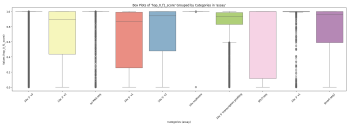
(a) Base Model



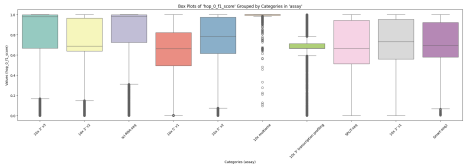
(b) Model 2 (Base Model + PP during inference)



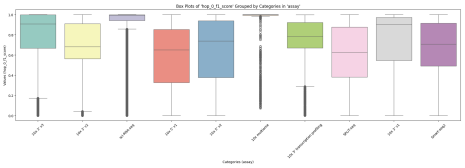
(c) Model Variation 3 (PP during training)



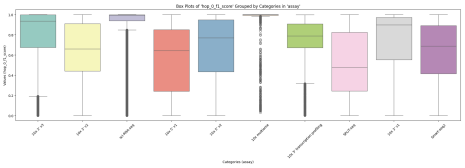
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

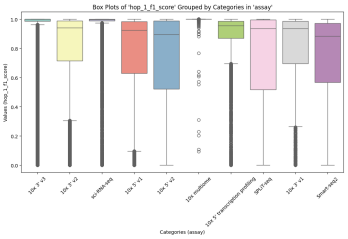


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

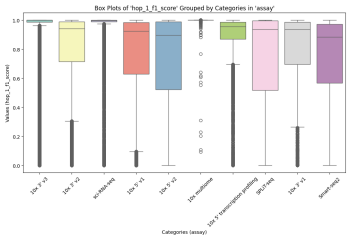


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

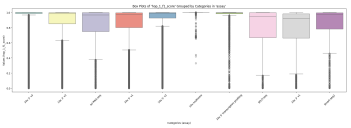
Hop 1 F1-Score Box Plots



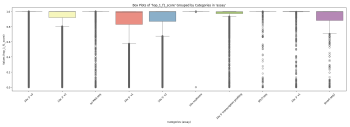
(a) Base Model



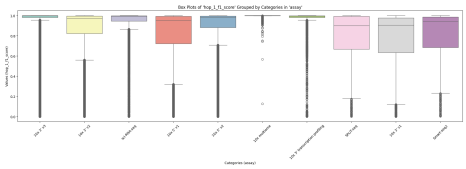
(b) Model 2 (Base Model + PP during inference)



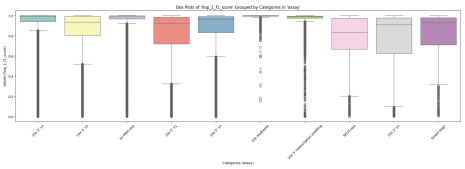
(c) Model Variation 3 (PP during training)



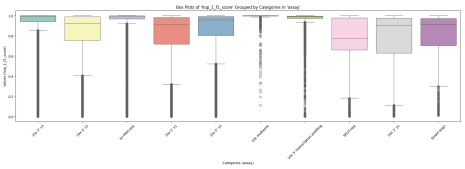
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

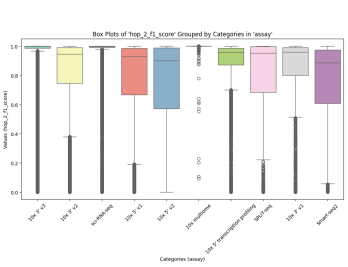


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

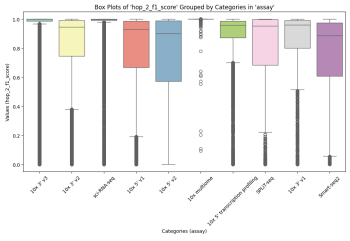


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

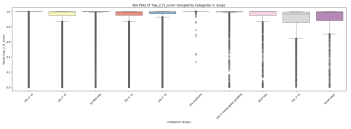
Hop 2 F1-Score Box Plots



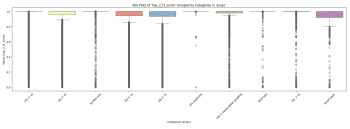
(a) Base Model



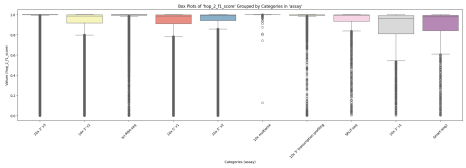
(b) Model 2 (Base Model + PP during inference)



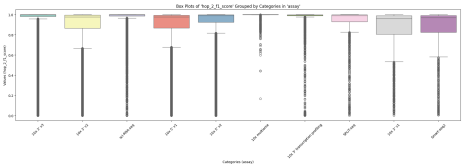
(c) Model Variation 3 (PP during training)



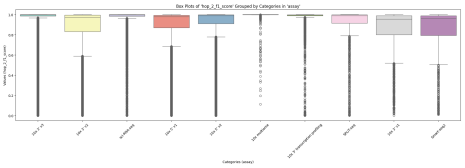
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

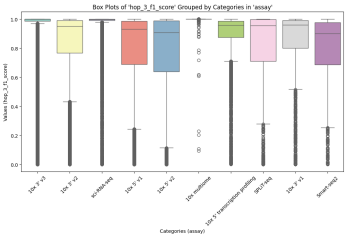


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

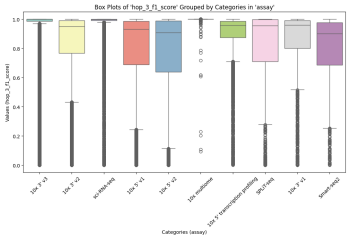


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

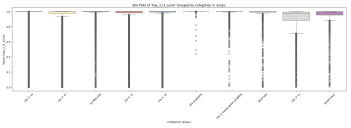
Hop 3 F1-Score Box Plots



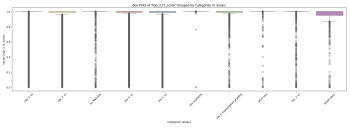
(a) Base Model



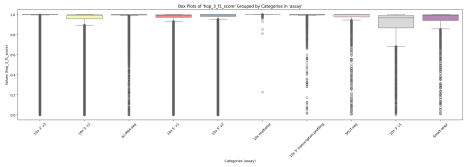
(b) Model 2 (Base Model + PP during inference)



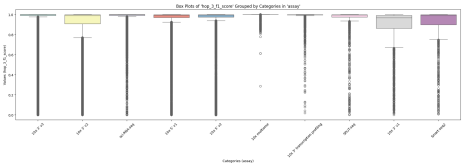
(c) Model Variation 3 (PP during training)



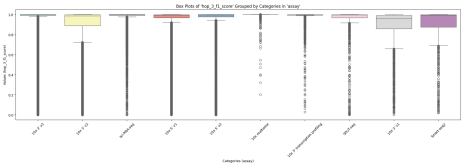
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

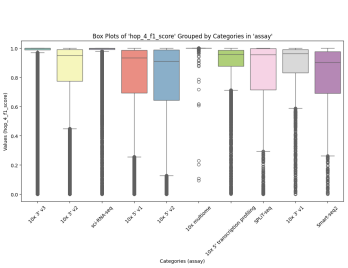


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

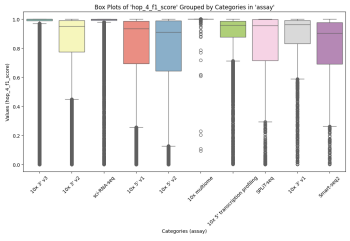


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

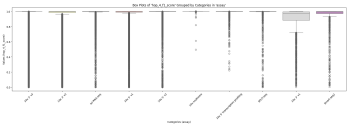
Hop 4 F1-Score Box Plots



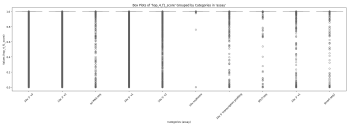
(a) Base Model



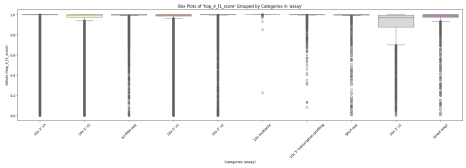
(b) Model 2 (Base Model + PP during inference)



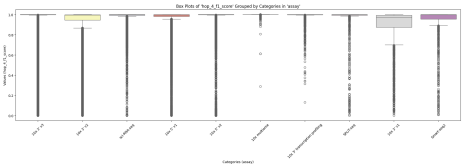
(c) Model Variation 3 (PP during training)



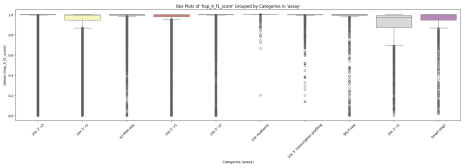
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)



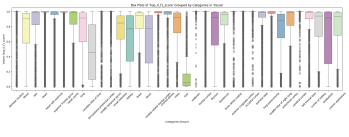
(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)



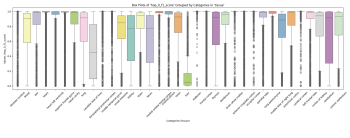
(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

C Hop-based F1 Score Box Plots by Tissue for each model

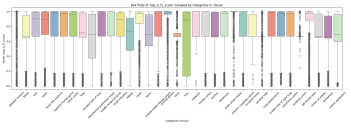
Hop 0 F1-Score Box Plots



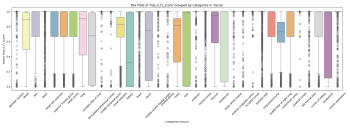
(a) Base Model



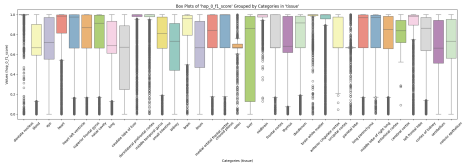
(b) Model 2 (Base Model + PP during inference)



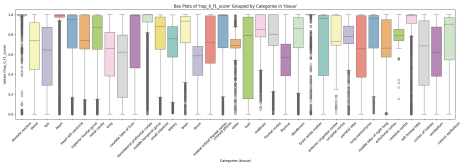
(c) Model Variation 3 (PP during training)



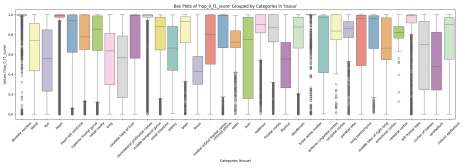
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

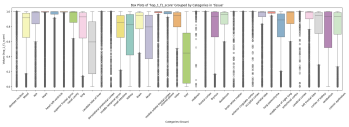


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

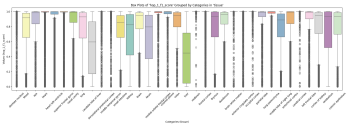


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

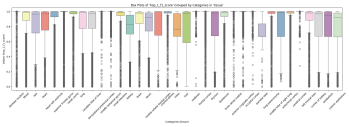
Hop 1 F1-Score Box Plots



(a) Base Model

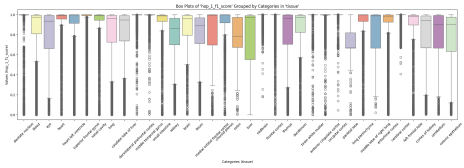


(b) Model 2 (Base Model + PP during inference)

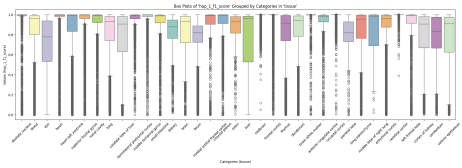


(c) Model Variation 3 (PP during training)

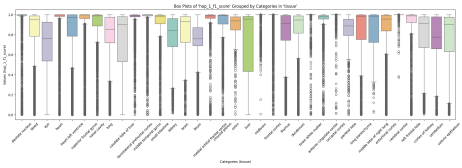
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

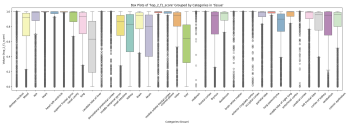


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

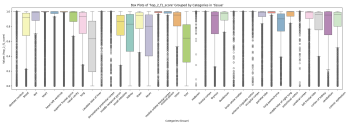


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

Hop 2 F1-Score Box Plots



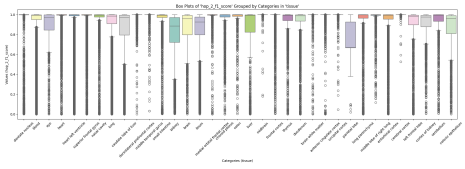
(a) Base Model



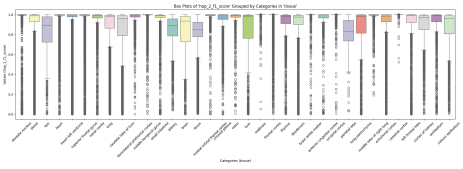
(b) Model 2 (Base Model + PP during inference)

(c) Model Variation 3 (PP during training)

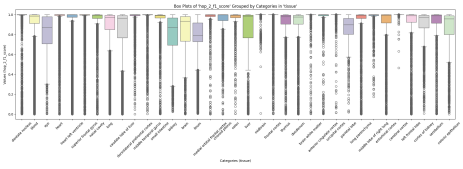
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

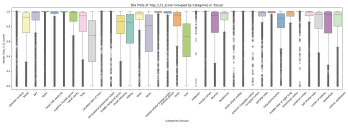


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

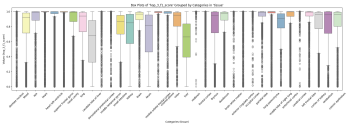


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

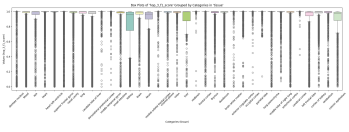
Hop 3 F1-Score Box Plots



(a) Base Model

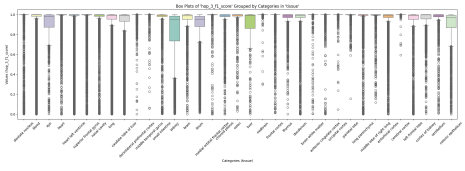


(b) Model 2 (Base Model + PP during inference)

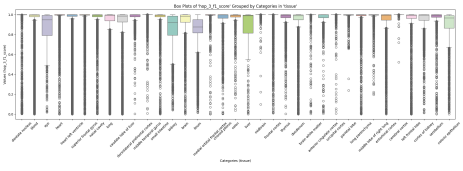


(c) Model Variation 3 (PP during training)

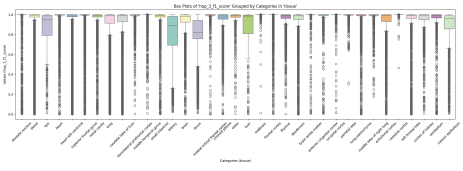
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)

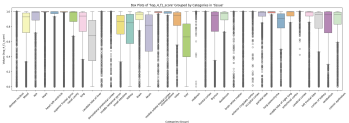


(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)

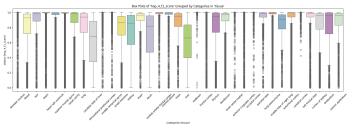


(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)

Hop 4 F1-Score Box Plots



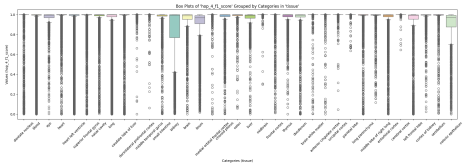
(a) Base Model



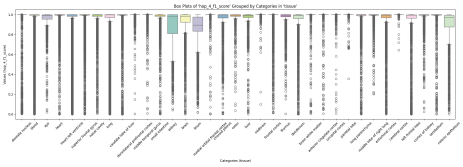
(b) Model 2 (Base Model + PP during inference)

(c) Model Variation 3 (PP during training)

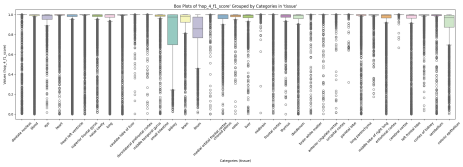
(d) Model Variation 4 (multiple targets, PP during training and BCE Loss)



(e) Model Variation 5 (variation 4 + alpha=0 for descendant cells)



(f) Model Variation 5 (variation 4 + alpha=0.5 for descendant cells)



(g) Model Variation 5 (variation 4 + alpha=1 for descendant cells)