NEU — Cellarium Challenge

Nimish Magre

June 16, 2024

Abstract

This document presents a novel methodology for aggregating and ranking nearest neighbors in single-cell RNA sequencing (scRNA-seq) data by integrating gene expression similarity measures with cell type hierarchical information. The approach leverages the distances provided by a nearest neighbor search engine, adjusted by a hierarchical component to reflect the biological relevance of cell type relationships, aiming to enhance the interpretability and biological significance of the results.

1 Problem Description (My Understanding)

The raw count matrix in single-cell RNA sequencing (scRNA-seq) data is a sparse, high-dimensional matrix where each column represents a gene and each row represents a cell type from a sample tissue. Each entry in the matrix denotes the count of RNA molecules captured for a specific gene in a particular cell type during the sequencing process. Typically, these matrices can contain 30,000 to 40,000 columns and vary significantly in row count, potentially reaching into the millions. The majority of these entries are zeros, with only a small fraction containing nonzero counts, indicative of gene expression.

To manage this high dimensionality, Principal Component Analysis (PCA) is initially applied to reduce the dimensions of this raw data to 512-dimensional vectors representing each cell type. These reduced representations are then processed through a Nearest Neighbor Search engine to identify cell types that are closely related to a given query cell type in terms of gene expression, providing potentially meaningful biological insights.

Additionally, a cell type hierarchy, structured by a specific ontology, is available. A key complexity is that a particular cell type could have multiple higher level parent cell types where the parent cell types may not have direct lineage to one another, while simultaneously, the cell type in focus could individually have multiple child nodes (cell types).

The main challenge then, is to devise an algorithm that effectively combines the proximity measures from the nearest neighbor search engine with hierarchical cell type information to provide meaningful insights into cell type similarities such that the cell type clusters ranked closest to the query cell type do not represent extremely granular or generalized information.

2 Key Assumptions

- **Data Preprocessing for PCA:** All necessary preprocessing steps required for effective PCA application on the raw scRNA-seq data were adhered to. This includes normalization of the data to account for differences in sequencing depth and gene expression variability, log transformation to stabilize variance, and the filtering of genes that are not expressed in a significant number of cell types. These steps ensure that the PCA results in meaningful dimensionality reduction and captures the major axes of variation in the data.

- **Data Preparation for Nearest Neighbor Search:** Prior to input into the nearest neighbor search engine, the data underwent appropriate preprocessing steps similar to those for PCA. This includes normalization and potential feature selection to enhance the meaningfulness of the proximity measures derived from the search engine. Additionally, any noise reduction techniques, such as batch effect correction, were applied to ensure that the distances calculated reflect true biological similarities rather than technical artifacts.

- **Reliability of Nearest Neighbor Search Engine:** The search engine used for identifying nearest neighbors is assumed to be robust, capable of handling high-dimensional data efficiently, and producing reliable, reproducible results that are sensitive to subtle variations in the data reflective of genuine biological differences.

- **Normalized distance values from Nearest Neighbor Search Engine output:** It is also assumed that the distance values that are output from the nearest neighbor search engine are normalized within the [0,1] range.

- **Cell Type representation in cell type hierarchy:** Each cell type in the reference dataset is assumed to have a place in the cell type hierarchy structure.

- **Unavailability of Ground-truth dataset:** There is no ground-truth dataset that provides the best cluster of cell types to describe a query cell type from the reference dataset. For ex: the antibody-secreting cell, B cell, and lymphocyte of B lineage are described to be the closest cluster of cell types to the CD86-positive plasmablast cell, but a dataset with similar information is not available for other cell types. The only instruction is to not consider cell types that are too granular or generalized.

- **Information on unkown query cell:** For an unknown/unlabeled query cell type, the only available information is the output from the nearest neighbor search engine. No information about its position in the cell-type hierarchy is known.

- **Consideration of cell types only with direct vertical linkage:** Based on the CD86-positive plasmablast example provided in the problem description, it is assumed that cell types that have direct vertical lineage should only be considered to describe the query cell type.

  For ex: Consider the following example cell type hierarchy structure:

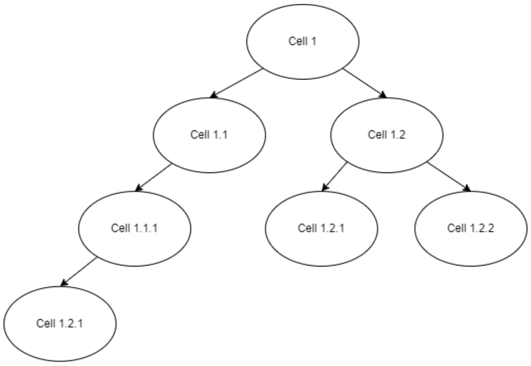
  

  Figure 1: cell-type hierarchy example

  If *Cell1.2.2* is my query cell type, then I must only consider *Cell1.2* and *Cell1* when clustering and ranking the cell types to describe my query cell type (*Cell1.2.2*) since only those two cell types have direct vertical lineage to my query cell type. *Cell1.1* does not have direct vertical lineage to the query cell type *Cell1.2.2* and therefore will automatically become part of the last ranked cluster.

3 Background Research

Due to unfamiliarity with the provided data, and the process of annotating scRNA seq data, the process of background research was performed with the following steps:

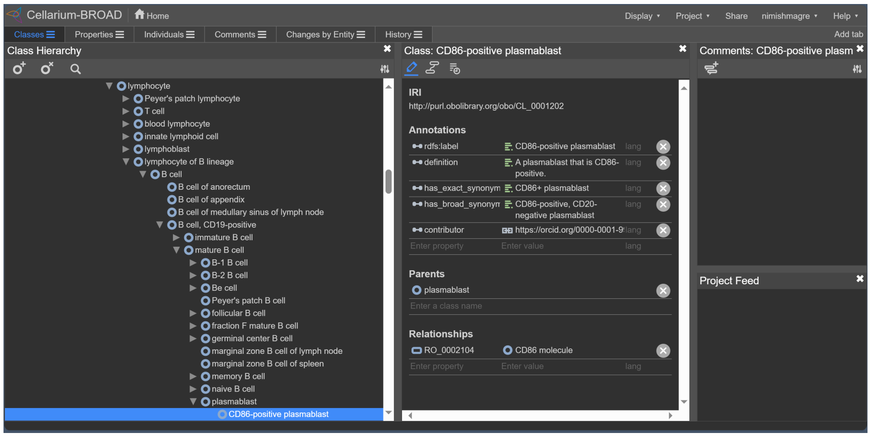
3.1 A Detailed Survey of Ontology-Based Single-Cell RNA Annotation Methods

This survey explored various research papers ([1], [11], [3], [8], [9]) that utilize hierarchical ontology data for annotating single-cell RNA-seq (scRNA-seq) data. The focus was on methods that applied Principal Component Analysis (PCA) for dimensionality reduction, followed by nearest neighbor search to identify cells similar to a query cell, and finally leveraged ontology data to contextualize the identified cell types within a hierarchical framework. Each research paper proposed a unique approach to incorporating cell type hierarchies through ontology data, enhancing the accuracy and biological relevance of the annotations. The surveyed methods were reviewed based on their algorithms, tests conducted, and their respective advantages and disadvantages, highlighting the contributions and computational complexities of each approach.

The detailed survey paper is available in my *Cellarium-BROAD* GitHub repository[1].

3.2 Data Exploration

To gain a comprehensive understanding of the characteristics for the provided data, the given Ipython notebook was further expanded upon to include a Data Exploration Section and is available on my *Cellarium-BROAD* GitHub repository[2]. Different data visualization techniques along with parameter exploration code were used to gain an understanding of the provided h5ad file using the anndata library, the provided pickle file using the pickle library as well as the provided ontology data using the owl file. Also, the WebProtege[3] platform was used to visualize the cell type hierarchy provided through ontology as shown in the image below:



- lymphocyte
  - Peyer's patch lymphocyte
  - T cell
  - blood lymphocyte
  - innate lymphoid cell
  - lymphoblast
  - lymphocyte of B lineage
    - B cell
      - B cell of anorectum
      - B cell of appendix
      - B cell of medullary sinus of lymph node
      - B cell, CD19-positive
        - immature B cell
        - mature B cell
          - B-1 B cell
          - B-2 B cell
          - Be cell
          - Peyer's patch B cell
          - follicular B cell
          - fraction F mature B cell
          - germinal center B cell
          - marginal zone B cell of lymph node
          - marginal zone B cell of spleen
          - memory B cell
          - naive B cell
          - plasmablast
            - CD86-positive plasmablast

IRI
http://purl.obolibrary.org/obo/CL_0001202

Annotations

rdfs:label	CD86-positive plasmablast	lang
definition	A plasmablast that is CD86-positive.	lang
has_exact_synonym	CD86+ plasmablast	lang
has_broad_synonym	CD86-positive, CD20-negative plasmablast	lang
contributor	https://orcid.org/0000-0001-9	lang
Enter property	Enter value	lang

Parents

plasmablast
Enter a class name

Relationships

RO_0002104	CD86 molecule	
Enter property	Enter value	lang

Figure 2: cell-type hierarchy visualization

3.3 Algorithm Reviews based on key solution conditions

The following key conditions were identified based on the problem description, example of the *CD86-positive plasmablast* query cell and the sample data

[1] Cellarium_Ontology_Research_Review.pdf: https://github.com/magrenimish/Cellarium-BROAD/blob/672809f96948e084894d3a4f74259dd8dec9d573/Cellarium_Ontology_Research_Review.pdf

[2] Cellarium_BROAD_Challenge.ipynb: https://github.com/magrenimish/Cellarium-BROAD/blob/cbd375c6072f0eeb848bede2fc4bf862023d3a4c/Cellarium_Broad_Challenge.ipynb

[3] Webprotege Platform for oncology visualization: https://webprotege.stanford.edu/

provided in the iPython notebook[4] so that online resources could be explored to develop a solution algorithm:

- The reference dataset is large and the clustering algorithm must contain data preprocessing steps as well as clustering steps that are capable of handling large scRNA seq reference data with computational efficiency.
- A query cell type from the reference data may have a variable number of neighbor cell types that best describe it and therefore the clustering algorithm must be able to clusters with different sizes for different query cells.
- There is minimal difference in nearest neighbor distances, often at the second or third decimal place.
- The provided ontology displays complex cell type hierarchy with multiple parent and child relationships. This hierarchy, along with the nearest neighbor distance, must be considered when aggregating and ranking potential neighbor cells.
- There is need for a ranking system that avoids over-prioritizing granularity or generality between hierarchical proximity at least when considering the reference data.

Clustering is an essential step in single-cell RNA sequencing (scRNA-seq) data analysis, aimed at grouping cells with similar gene expression profiles into distinct clusters. This process helps in identifying cell types and understanding cellular heterogeneity within a tissue or organism. Based on the conditions above, the following algorithms and associated research papers were explored:

- **k-means Clustering[5]**: k-means is a popular partitioning method that divides the data into k predefined clusters. It works by minimizing the variance within each cluster. Each cell is assigned to the cluster with the nearest mean, which serves as a prototype of the cluster. While k-means is computationally efficient and easy to implement, it requires the number of clusters (k) to be specified in advance, which can be a limitation. Additionally, it may not perform well on data with complex structures or non-globular clusters, which are common in scRNA-seq datasets.
- **Hierarchical Clustering[6]**: Hierarchical clustering builds a hierarchy of clusters that can be represented as a tree or dendrogram. This method does not require the number of clusters to be specified beforehand. It can be agglomerative (bottom-up approach, starting with individual cells and merging them into clusters) or divisive (top-down approach, starting with all cells and recursively splitting them). Hierarchical clustering is particularly useful for visualizing the nested structure of data and understanding

[4] Neu-Broad-Challenge.ipynb: `https://github.com/magrenimish/Cellarium-BROAD/blob/cbd375c6072f0eeb848bede2fc4bf862023d3a4c/Cellarium_Broad_Challenge.ipynb`

the relationships between clusters. However, it can be computationally intensive and less scalable for large datasets.

- **Graph-based Clustering[2], [4], [10]**: Graph-based methods, such as the Louvain algorithm, construct a graph where nodes represent cells and edges represent similarities between cells. Community detection algorithms are then applied to identify clusters within the graph. These methods are highly effective for capturing complex and irregular structures in scRNA-seq data. Graph-based clustering can handle large datasets efficiently and is less sensitive to noise and outliers. The Louvain algorithm, in particular, is widely used due to its ability to optimize modularity and identify communities within the graph structure.

A further detailed survey paper discussing each of these techniques individually is available at [7]. Each of these methodologies and papers was selected for review due to their direct applicability to handling the complexities presented by scRNA-seq data analysis, specifically in the contexts of reducing dimensionality while improving clustering accuracy.

4 Solution Strategy (Methodology)

This section delineates the methodological framework designed to efficiently aggregate and rank nearest neighbors in scRNA-seq data by synergistically considering both the high-dimensional proximity (gene expression similarity or nearest neighbor distance) and the hierarchical cell type relationships.

The proposed solution is divided into two primary stages, each addressing different aspects of the cell type annotation process. Please not that the python implementation of the examples provided to explain each stage is provided with detailed comments in the IPython notebook stored on the submission GitHub repository[4]. Below is a detailed description of each stage:

4.1 Stage 1: Working with Reference Dataset with Known Cell Type Hierarchy

This stage focuses on leveraging a reference dataset where the position of each cell type within the cell type hierarchy is known. It involves three critical steps:

1. **Data Preprocessing:** In this first step, we preprocess the reference data based on the key assumption about direct vertical lineage with the query cell type specified in the *Key Assumptions*[2] section. The cell types that do not have direct vertical lineage to the query cell type will not be considered in the clustering and ranking process. This helps also in reducing the computation cost of clustering and ranking since a lower number of

[4] Neu-Broad-Challenge.ipynb: `https://github.com/magrenimish/Cellarium-BROAD/blob/cbd375c6072f0eeb848bede2fc4bf862023d3a4c/Cellarium_Broad_Challenge.ipynb`

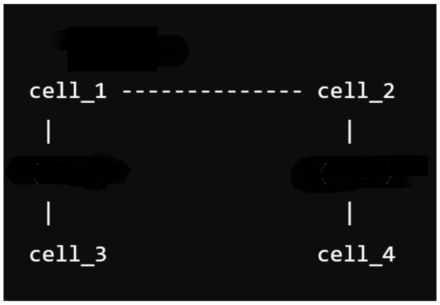
cell types (only the ones with direct vertical lineage to the query cell type) will be considered for further processing.

2. **Clustering Reference Cell Types for Each Query Cell Type:**

   - This step involves clustering cell types from the reference dataset in relation to a query cell type from the same reference data after preprocessing it. Based on the previous research *Algorithm Reviews based on key solution conditions[3.3*] subsection for clustering scRNA data, we will make use of a modified version of the graph based Louvain algorithm because of its ability to handle large amounts of data, ability to form dynamic clusters (clusters with different sizes), previous success in omprehensive scRNA-seq data analysis such as esolving fine-grained cellular heterogeneity [7], and finally, the ability to incorporate nearest neighbor distances as weights in the graph edges.

     In this modified version, the edges in the graphs will have weights that reflect the nearest neighbor distance such that a higher weight value is considered better (representing a neighbor that is closer in nearest neighbor distance to the query cell).

     Please consider the example in the figure below for better explanation:



```
  [redacted]

cell_1 -------------- cell_2
 |                      |
[redacted]          [redacted]
 |                      |
cell_3                cell_4
```

Figure 3: Example graph with nearest neighbor distance values

     This graph represents cell type hierarchy structure with the following parent-child relations:

     (*cell_1,cell_2*), (*cell_1,cell_3*), (*cell_2,cell_4*)

     If we consider *cell_4* to be the query cell type, then from the data preprocessing step, we eliminate *cell_3* from consideration since *cell_3* does not have direct vertical lineage to *cell_4*.

     The filtered lineage nodes include:

     – *cell_1*
     – *cell_2*
     – *cell_4*

Following are the assumed nearest neighbor distances between the query cell type *cell_4* and the filtered neighbors:

$$\begin{aligned}\text{distance}(\textit{cell\_4}, \textit{cell\_1}) &\approx 0.5\\ \text{distance}(\textit{cell\_4}, \textit{cell\_2}) &\approx 0.4\end{aligned}$$

Using these distances, we can set the edge weights between the query cell type (*cell_4*) and its neighbors as follows:

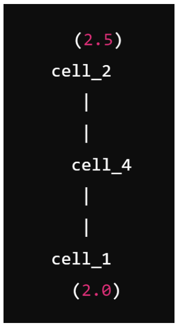
- **Edge (*cell_4*, *cell_1*):**

$$\begin{aligned}\text{Distance:}&\quad 0.5\\ \text{Weight:}&\quad \frac{1}{0.5} = 2.0\end{aligned}$$

- **Edge (*cell_4*, *cell_2*):**

$$\begin{aligned}\text{Distance:}&\quad 0.4\\ \text{Weight:}&\quad \frac{1}{0.4} = 2.5\end{aligned}$$

Considering the reciprocal of the nearest neighbor distance value helps calculate the weights such that neighbor cell types closer in terms of nearest neighbor distance are considered to be better. Therefore the final graph output to pass to the Louvain algorithm can be visualized as follows:



```
  (2.5)
cell_2
   |
   |
  cell_4
   |
   |
cell_1
  (2.0)
```

Figure 4: updated cell-type hierarchy graph with nearest neighbor distance as weights

In this graph, the values in red represent the weights between the query cell type and the filtered neighbor nodes.

- The clustering process ensures that cell types with similar transcriptomic profiles are grouped together, facilitating more accurate downstream analyses.

3. **Ranking Clusters by Hierarchical Distance:**

   - Once the clusters are formed, the next step is to rank these clusters based on their average hierarchical distance to the query cell type. This distance is calculated as the average hierarchical position of all cell types within a cluster relative to the query cell type using the given cell type hierarchy structure (ontology).

     Assume you have a query cell type Q and several other cell types with their hierarchical distances to Q. The Louvain algorithm has divided these cell types into clusters in the previous clustering step. For simplicity, let's say we have three clusters:

     – **Cluster 1**: {A, B, C}
     – **Cluster 2**: {D, E}
     – **Cluster 3**: {F, G, H}

     Assume the hierarchical distances from the query cell type Q to each of these cell types are as follows:

$$\begin{aligned}
\text{distance}(Q, A) &= 2 \\
\text{distance}(Q, B) &= 3 \\
\text{distance}(Q, C) &= 1 \\
\text{distance}(Q, D) &= 5 \\
\text{distance}(Q, E) &= 4 \\
\text{distance}(Q, F) &= 7 \\
\text{distance}(Q, G) &= 8 \\
\text{distance}(Q, H) &= 6
\end{aligned}$$

Calculating the Average Hierarchical Distance for Each Cluster

Cluster 1:

$$\text{Average distance} = \frac{\text{distance}(Q, A) + \text{distance}(Q, B) + \text{distance}(Q, C)}{3} = \frac{2 + 3 + 1}{3} = 2$$

Cluster 2:

$$\text{Average distance} = \frac{\text{distance}(Q, D) + \text{distance}(Q, E)}{2} = \frac{5 + 4}{2} = 4.5$$

Cluster 3:

$$\text{Average distance} = \frac{\text{distance}(Q, F) + \text{distance}(Q, G) + \text{distance}(Q, H)}{3} = \frac{7 + 8 + 6}{3} = 7$$

Calculating the Median of the Average Hierarchical Distances

The average distances are 2, 4.5, and 7. The median of these values is 4.5.

Ranking the Clusters Based on Proximity to the Median

Calculate the absolute difference from the median for each cluster's average distance:

$$\begin{aligned}\text{Cluster 1} &: |2-4.5| = 2.5\\ \text{Cluster 2} &: |4.5-4.5| = 0\\ \text{Cluster 3} &: |7-4.5| = 2.5\end{aligned}$$

Rank the clusters based on these differences (the smaller the difference, the higher the rank):

- **Cluster 2**: Difference 0 (Rank 1)
- **Cluster 1**: Difference 2.5 (Rank 2)
- **Cluster 3**: Difference 2.5 (Rank 3)

Summary of the Ranking Process:

(a) **Cluster 2** (Rank 1) - Average hierarchical distance to Q is closest to the median (4.5).

(b) **Cluster 1** (Rank 2) - Average hierarchical distance to Q is tied for second closest to the median.

(c) **Cluster 3** (Rank 3) - Average hierarchical distance to Q is tied for second closest to the median but less preferable due to larger absolute values.

- This ranking method helps consider the condition of best ranked cluster cell types not being too granular or too generalized in comparison to the position of the query cell type in the cell type hierarchy structure.

4.2 Stage 2: Working with Unseen Data with Unknown Cell Type Hierarchy

In this stage, the goal is to annotate cell types for an unseen dataset, where the hierarchical position of each cell type is not known. This stage comprises of three steps:

1. **Identifying Nearest Neighbor Cell Types:**

   - For each query cell in the unseen dataset, the top 10 nearest neighbor cell types are identified from the reference dataset. This is done by selecting the 10 cell types with the lowest nearest neighbor distance values to the query cell.

2. **Selecting Nearest Clusters:**

   - For each of the 10 nearest neighbor cell types, the top 3 ranked clusters are selected. These ranked clusters were obtained during the last stage. This results in a total of 30 clusters being considered for each query cell type.

     We also add the nearest neighbor cell type to each of these 2 clusters.

3. **Ranking Clusters by Average Nearest Neighbor Distance:**

   - The 30 selected clusters are then ranked in ascending order based on the average nearest neighbor distance for all cells within each cluster. Clusters with the smallest average distance are ranked highest, as they are considered to be more similar to the query cell.

   Note: The exact values of considering the top 10 nearest neighbors and the 3 top clusters for each number have been selected with the thought that 30 clusters are enough to annotate a single cell type .These values could be changed after further discussions with experts in the field and depending on the computational requirements of the project

   Please consider the following detailed example to explain each step in this stage:

   Assume we have a query cell Q and a reference dataset with cell types $A, B, C, D, E, F, G, H, I, J, K, L, M$. The nearest neighbor distances to the query cell Q are as follows:

$$\begin{aligned}
\text{distance}(Q, A) &= 0.2 \\
\text{distance}(Q, B) &= 0.3 \\
\text{distance}(Q, C) &= 0.4 \\
\text{distance}(Q, D) &= 0.5 \\
\text{distance}(Q, E) &= 0.6 \\
\text{distance}(Q, F) &= 0.7 \\
\text{distance}(Q, G) &= 0.8 \\
\text{distance}(Q, H) &= 0.9 \\
\text{distance}(Q, I) &= 1.0 \\
\text{distance}(Q, J) &= 1.1 \\
\text{distance}(Q, K) &= 1.2 \\
\text{distance}(Q, L) &= 1.3 \\
\text{distance}(Q, M) &= 1.4
\end{aligned}$$

The 10 nearest neighbors to Q are $A, B, C, D, E, F, G, H, I, J$.

Assume each nearest neighbor cell type has already been clustered using the Louvain algorithm, and we have their top three ranked clusters based on the previous stage as shown below.

Table 1: Top 3 ranked clusters for each of the 10 nearest neighbor cell types

Nearest Neighbor	**Top 3 Clusters**
A	{A1, A2, A3}
B	{B1, B2, B3}
C	{C1, C2, C3}
D	{D1, D2, D3}
E	{E1, E2, E3}
F	{F1, F2, F3}
G	{G1, G2, G3}
H	{H1, H2, H3}
I	{I1, I2, I3}
J	{J1, J2, J3}

Add each nearest neighbor cell type to its respective top three clusters (For ex: A is added to *A1,A2,A3*). Therefore, we have a total of $10 \times 3 = 30$ clusters to consider.

We calculate the average nearest neighbor distance of all cell types within each cluster with the query cell Q and rank these clusters based on the least to maximum average nearest neighbor distance.

For simplicity, let's consider a subset of clusters:

Table 2: Subset of the 30 clusters for average nearest neighbor distance calculation

Cluster	**Cells in Cluster (with Nearest Neighbor Added)**
A1	{A, A1_1, A1_2, A1_3}
A2	{A, A2_1, A2_2, A2_3}
A3	{A, A3_1, A3_2, A3_3}
B1	{B, B1_1, B1_2, B1_3}
B2	{B, B2_1, B2_2, B2_3}
B3	{B, B3_1, B3_2, B3_3}

The sum of nearest neighbor distance of cell types in each cluster without the nearest neighbor is assumed to be:

$$\text{Sum of distance in A1 (without A)} = 0.5$$
$$\text{Sum of distance in A2 (without A)} = 0.6$$
$$\text{Sum of distance in A3 (without A)} = 0.7$$
$$\text{Sum of distance in B1 (without B)} = 0.7$$
$$\text{Sum of distance in B2 (without B)} = 0.8$$
$$\text{Sum of distance in B3 (without B)} = 0.11$$

Adding the nearest neighbor:

$$\begin{aligned}
\text{Average distance in A1 (with A)} &= \frac{\text{distance}(Q, A) + \text{sum of distances in A1 (without A)}}{\text{number of cell types in A1 (with A)}} \\
&= \frac{0.2 + 0.5}{4} = \frac{0.7}{4} = 0.175 \\
\text{Average distance in A2 (with A)} &= \frac{0.2 + 0.6}{4} = \frac{0.8}{4} = 0.2 \\
\text{Average distance in A3 (with A)} &= \frac{0.2 + 0.7}{4} = \frac{0.9}{4} = 0.225 \\
\text{Average distance in B1 (with B)} &= \frac{0.3 + 0.7}{4} = \frac{1.0}{4} = 0.25 \\
\text{Average distance in B2 (with B)} &= \frac{0.3 + 0.8}{4} = \frac{1.1}{4} = 0.275 \\
\text{Average distance in B3 (with B)} &= \frac{0.3 + 0.11}{4} = \frac{0.41}{4} = 0.1025
\end{aligned}$$

Table 3: Average Nearest Neighbor Distance for each Cluster

Cluster	**Average Distance**
A1	0.175
A2	0.2
A3	0.225
B1	0.25
B2	0.275
B3	0.1025

Rank the clusters from least to maximum average nearest neighbor distance:

(a) **B3**: 0.1025 (Rank 1)

(b) **A1**: 0.175 (Rank 2)

(c) **A2**: 0.2 (Rank 3)

(d) **A3**: 0.225 (Rank 4)

(e) **B1**: 0.25 (Rank 5)

(f) **B2**: 0.275 (Rank 6)

This process continues for all 30 clusters, and the final ranked list will determine the best cluster for annotating the query cell Q.

By following these stages, the algorithm effectively integrates hierarchical information from the reference dataset and applies it to annotate cell types in an unseen dataset, ensuring accurate and reliable annotations.

5 Possible Evaluation Techniques Without Ground Truth Data

Since we do not have well labelled ground-truth data that tells us the preferred cluster of neighbor cell types that would best describe a query cell type, here are some strategies that can be used to evaluate the outputs of the algorithm:

5.1 Biological Plausibility and Consistency

- **Expert Feedback:** Use biological knowledge and feedback from the experts to evaluate whether the output ranked clusters make sense.
- **Consistency:** Run the algorithm multiple times with different subsets of the data to see if the results are consistent. Consistent results across different runs suggest that the algorithm is capturing robust patterns.

5.2 Internal Validation Metrics

- **Silhouette Score:** Measures how similar a cell type is to its own cluster compared to other clusters. A higher silhouette score indicates better-defined clusters.

  $$\text{Silhouette Score} = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

  where $a(i)$ is the average distance from the i-th point to the other points in the same cluster, and $b(i)$ is the average distance from the i-th point to points in the nearest cluster.
- **Calinski-Harabasz Index:** Measures the ratio of the sum of between-cluster dispersion to within-cluster dispersion. Higher values indicate better clustering.

$$\text{CH Index} = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{n-k}{k-1}$$

where B_k is the between-cluster dispersion matrix, W_k is the within-cluster dispersion matrix, n is the number of samples, and k is the number of clusters.

- **Davies-Bouldin Index:** Measures the average similarity ratio of each cluster with the cluster that is most similar to it. Lower values indicate better clustering.

$$\text{DB Index} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{s_i + s_j}{d_{ij}}$$

where s_i is the average distance of all points in the i-th cluster to its centroid, and d_{ij} is the distance between the centroids of the i-th and j-th clusters.

5.3 External Validation via Related Datasets

- **Cross-Dataset Validation:** Apply the clustering algorithm to related datasets that may have partial annotations or are from similar biological conditions. Compare the results to see if similar clusters are identified.

5.4 Cluster Quality Metrics

- **Within-Cluster Sum of Squares (WCSS):** Measures the compactness of the clusters. Lower values indicate tighter clusters.

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

where K is the number of clusters, C_k is the set of points in cluster k, x_i is a data point, and μ_k is the centroid of cluster k.

By using a combination of these strategies, it is possible to gain confidence in the validity of the clustering results even in the absence of ground truth data. The goal is to ensure that the clusters are biologically meaningful, consistent, and robust across different analyses and datasets.

6 Possible Future Work

This section outlines several promising directions for further development and enhancement of the current methodology. These advancements aim to refine the dimensionality reduction processes, optimize nearest neighbor search engines, improve preprocessing steps, and better integrate advanced machine learning techniques.

6.1 Advanced Dimensionality Reduction Techniques

While PCA is a robust method for reducing dimensionality, it primarily captures linear relationships. Future work could explore more sophisticated techniques that capture non-linear relationships in the data:

- **T-distributed Stochastic Neighbor Embedding (t-SNE)**: This technique is particularly well-suited for the visualization of high-dimensional datasets and could provide more intuitive clustering by preserving local structures.
- **Uniform Manifold Approximation and Projection (UMAP)**: UMAP could be employed to better preserve both local and global data structures, potentially offering more meaningful biological insights.

6.2 Enhancements to Nearest Neighbor Search

Current nearest neighbor searches could be enhanced by integrating more efficient algorithms like Kmeans++:

- **Kmeans++ for Initialization**: Using Kmeans++ for selecting initial centroids before running more complex clustering algorithms could improve the efficiency and accuracy of these searches, reducing the likelihood of falling into local minima.
- **Approximate Nearest Neighbor (ANN) Algorithms**: Techniques such as Locality-Sensitive Hashing or tree-based approaches like Annoy (Approximate Nearest Neighbors Oh Yeah) could be explored to speed up searches without significant loss of accuracy.

6.3 Improvements in Preprocessing Steps

Optimizing the preprocessing of scRNA-seq data is crucial for enhancing the overall analysis:

- **Advanced Filtering Techniques**: Implementing more robust noise reduction and outlier detection methods to ensure that the data fed into the analysis pipelines is of the highest quality.
- **Batch Effect Correction**: Further refining methods for correcting batch effects, which is crucial for ensuring that biological conclusions are not confounded by technical variability.

6.4 Adaptation of Machine Learning Techniques

The integration of advanced machine learning techniques could be further explored to enhance the clustering, aggregation, and ranking processes:

- **Supervised Learning Approaches**: Leveraging labeled datasets to train models that can predict cell types or states more accurately.
- **Deep Learning Models**: Utilizing neural networks, especially autoencoders, for unsupervised feature extraction and dimensionality reduction, which could be particularly effective in capturing complex, non-linear patterns in scRNA-seq data.
- **Reinforcement Learning**: Developing models that iteratively improve their predictions of cell type hierarchies based on feedback loops, dynamically adjusting their parameters to optimize clustering and ranking outcomes.

These future avenues of research promise not only to refine the technical aspects of the analysis but also to enhance the biological relevancy and accuracy of the results obtained from scRNA-seq data.

7 Conclusion

The proposed solution offers a robust framework for enhancing the analysis of scRNA-seq data by thoughtfully integrating cell type hierarchy into the ranking and aggregation of nearest neighbor searches. This methodology not only improves the biological relevance of the analysis results but also provides flexibility to accommodate diverse analytical needs and data complexities.

References

[1] D. Aran, A. P. Looney, L. Liu, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature Immunology*, 20(2):163–172, 2019.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[3] B. B. Lake, S. Codeluppi, Y. C. Yung, D. Gao, J. Chun, P. V. Kharchenko, S. Linnarsson, and K. Zhang. Neuronal subtypes and diversity revealed by single-nucleus rna sequencing of the human brain. *Nature*, 531(7599):173–178, 2016.

[4] Yun Lin, Ling Pan, Weixiong Zhang, and Yuying Xiang. Quantifying the clusterness and trajectoriness of single-cell rna-seq data. *PLOS Computational Biology*, 16(8):e1007965, 2020.

[5] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

[6] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, 2014.

[7] Megha Patel, Nimish Magre, Himanshi Motwani, and Nik Bear Brown. Advances in machine learning, statistical methods, and ai for single-cell rna annotation using raw count matrices in scrna-seq data. *arXiv preprint arXiv:2406.05258*, June 07 2024.

[8] K. Shekhar, P. Brodin, M. M. Davis, S. Chakravarthy, C. Sun, R. Chen, K. A. Christie, B. W. Chow, B. S. Clark, K. Deisseroth, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell Systems*, 1(1):73–84, 2014.

[9] B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience*, 19(2):335–346, 2016.

[10] Xinyu Wang, Zehong Yu, Yang Li, Lei Liu, and Chengwen Liu. Attention-based deep clustering method for scrna-seq cell type identification. *PLOS Computational Biology*, 18(4):e1011641, 2022.

[11] A. W. Zhang, C. O'Flanagan, E. A. Chavez, et al. Automated single-cell rna-seq analysis and interpretation. *Nature Methods*, 16(4):311–314, 2019.