Benchmarking Open-Source LLMs for Cell Type Annotation in scRNA-seq Data

By

Sanskar Pant

A DISSERTATION

Submitted to
The University of Liverpool

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Date: October 1 , 2024

Student Declaration

I confirm that I have read and fully understood the University's Academic Integrity Policy. I affirm that I have conducted myself honestly, ethically, and professionally in all actions leading to this assessment for the programme of study. I confirm that the attached work is my own, and that I have not copied material from any source, nor committed plagiarism or fabricated data. I further declare that I have not previously presented any part of this work for assessment in another University of Liverpool module. I confirm that I have not colluded with any other student in the preparation and production of this work, nor have I included any material submitted by myself or another person in support of a successful application for a degree at this or any other university or degree-awarding body. Additionally, while I may have utilized AI tools to assist in research and development, their use was in full compliance with the academic integrity guidelines, ensuring originality and ethical contribution to the integrity of this project.

SIGNATURE ________________________________

DATE October 1, 2024

Acknowledgments

I would like to express my gratitude to everyone that helped to finish the project "Benchmarking Open-Source LLMs for Cell Type Annotation in scRNA-seq Data."

Above all, I want to express my sincere gratitude to my supervisor, Antony McCabe, for his tremendous advice, perceptive criticism, and constant support during this endeavour. The success of this effort and its direction have been greatly influenced by his knowledge and mentoring.

I would also like to expresses my gratitude to the University of Liverpool for providing essential resources and acknowledges the Barkla HPC cluster for providing computational support, both of which were crucial in making this effort possible.

My sincere gratitude also extends to my family and friends, whose continuous moral support and consolation have been a source of fortitude during this ordeal. They had faith in my talents, which kept me going when things got hard.

Finally, the foundation of my study has been greatly shaped by the tremendous contributions of the larger research community, which I would like to appreciate. Their understanding of complex language models and bioinformatics was really helpful in directing my study strategy.

Without the commitment and assistance of each of these people and organisations, this initiative would never have materialised. I appreciate you making this research possible.

Benchmarking Open-Source LLMs for CellType Annotation in scRNA-seq Data

Abstract

The development of single-cell RNA sequencing (scRNA-seq) allowed for high-resolution tissue-level cellular diversity profiling, which has completely changed biological research [1]. Cell type annotation presents a significant obstacle to scRNA-seq analysis since it has always relied on expert manual curation, which is laborious and error-prone [2]. GPT-4 and other recent advances in large language models (LLMs) have shown great promise in au- tomating this process [9]. However, models such as GPT-4 are not widely available due to their proprietary nature and high computing requirements [3]. This thesis investigates the efficacy of multiple open-source LLMs for automatic cell type annotation from scRNA-seq data: EleutherAI/gpt-neo- 1.3B, Facebook/opt-1.3b, Google/flan-t5-large, and Microsoft/phi-1_5.

By means of extensive benchmarking, gene expression data from publicly accessible scRNA-seq datasets were used to assess each model on the basis of accuracy, precision, recall, and F1 score [4].

With an accuracy of 57%, EleutherAI/gpt-neo-1.3B proved to be the most successful model [5]; google/flan-t5-large carried out well, especially in pre- cision [6]. Notwithstanding these findings, quick compliance and verbosity issues were encountered by all models, underscoring the necessity of more prompt engineering and fine-tuning for bioinformatics activities [6]. Fur- thermore, the integration of larger models such as Llama-2 70B, which might provide better performance if explored in future research, was not possible due to the computing limitations of this work [7].

The results of this dissertation demonstrate the potential of open-source LLMs to automate labour-intensive processes in scRNA-seq analysis, which adds to the expanding area of bioinformatics [8]. These models offer an accessible and affordable substitute for proprietary solutions such as GPT-4, especially for smaller academic and research institutions.

To fully realise the potential of LLMs in genomics, more research into fine-tuning models, bigger dataset integration, and testing more sophisticated models are required [9].

Contents

List of Figures

List of Tables

Chapter 1

Introduction

By profiling gene expression at the single-cell level, single-cell RNA sequenc- ing (scRNA-seq) allows researchers to analyse cellular diversity in detail and reveal tissue heterogeneity [**8**]. Cell type annotation, which identifies cell types using gene expression data, is still a labour- and time-intensive proce- dure [**2**]. For different cell types, researchers typically curate marker genes manually, which heavily relies on expert knowledge of canonical markers [**1**]. Machine learning (ML) techniques are being investigated to automate this procedure because of the increased complexity of scRNA-seq datasets and the increasing need for precise and effective annotations [**6**].

Large Language Models (LLMs)—like GPT-3, GPT-4, and ChatGPT—have shown great promise in handling complicated datasets and bioinformatics- related activities in recent years [**9**]. Particularly GPT-4 has demonstrated remarkable ability to automatically annotate cell types using information from marker genes, eliminating the requirement for human input [**9**]. GPT-4 has demonstrated great accuracy and reproducibility in identifying cell types across numerous tissues and species by utilising the transformer architecture and its capacity to model complicated patterns [**11**]. As a result, it is a potential tool in the field of genomics.

Even though GPT-4 has shown to be effective, a larger scientific community cannot access it due to its exclusive nature [**3**]. Open-source models present a compelling substitute, delivering equivalent functionalities without the financial limitations linked to subscription-based APIs [**5**]. This makesit necessary to assess the performance of open-source LLMs on the task ofcell type annotation, such as EleutherAI/gpt-neo-1.3B, Facebook/opt-1.3b, Google/flan-t5-large, and Microsoft/phi-1_5. These models include a wide range of parameter sizes and designs, offering a rich diversity of methods for automating scRNA-seq research [**12**].

1.1 Motivation for Research

Analysing the performance of open-source LLMs in comparable scenarios is highly motivated, especially considering GPT-4's success with scRNA-seq annotation [**9**]. Open-source models are becoming more and more popular in a variety of fields because of their adaptability, transparency, and capacity to be customised for specific purposes [**5**]. Still missing, though, is a thorough assessment of these models in the context of cell type annotation. This study aims to close this knowledge gap by assessing the performance of multiple open-source models and offering insights into how these models stack up against proprietary alternatives like GPT-4 [9].

1.2 Research Questions

The following research questions will be addressed in this dissertation:

1. Accuracy: When compared to GPT-4 and conventional annotation techniques, how accurate can open-source LLMs identify cell types from scRNA-seq datasets? [9]

2. Generalisability: How well can various LLM structures apply to a range of cell types and datasets in both healthy and sick tissues? [2]

1.3 Scope

This research will involve:

- Integrating open-source LLMs into a standardised pipeline for cell type annotation, such as EleutherAI/gpt-neo-1.3B, Facebook/opt-1.3b, Google/flan-t5-large, and Microsoft/phi-1_5.

- Evaluating the performance of these models using critical metrics such as accuracy, precision, and recall on scRNA-seq datasets that are made available to the public [6].

- Benchmarking these models' output against manual annotations as well as the performance of GPT-4, as demonstrated by recent studies [9].

1.4 Problem Statement

Despite GPT-4's impressive performance in cell type annotation, its re- stricted nature prevents it from being widely used in the scientific and aca- demic communities [9]. Although they are a good substitute, open-source LLMs are still not fully utilised in bioinformatics applications [5]. This study intends to compare the performance of open-source LLMs with cur- rently used techniques in order to assess how well they perform cell type annotation for scRNA-seq data [8]. The project will specifically concentrate on models like Microsoft/phi-1_5, Facebook/opt-1.3b, Google/flan-t5-large, EleutherAI/gpt-neo-1.3B, and Microsoft/phi-1_5.

1.5 Significance of the Study

This research will be one of the first to systematically compare the per- formance of open-source LLMs in bioinformatics, specifically for cell type annotation in scRNA-seq data [5]. The results will provide important new information about how open-source models may democratise the applicationof AI in genomics by providing affordable, easily accessible substitutes for proprietary systems such as GPT-4 [9]. Furthermore, by bridging the gap between AI research and bioinformatics processes, the evaluation of these models will add to the larger conversation on the use of natural language processing (NLP) tools to solve intricate biological challenges [10].

1.6 Approach

The project builds an efficient process for testing open-source LLMs using a number of tools and modules. Data preprocessing for scRNA-seq data, including as quality control, normalisation, and dimensionality reduction, is accomplished using the Scanpy library [16]. Hugging Face Transformers are utilised in the modular asynchronous design of the annotation pipelineto enable the handling of numerous jobs at once [10]. Through the use of unique prompt tactics created to efficiently query the models for cell type annotation, each LLM is integrated into the pipeline [5].

1.7 Outcome

The findings of the research offer a comprehensive analysis of various open- source LLMs for scRNA-seq data cell type annotation, with EleutherAI/gpt-neo-1.3B appearing as the most successful model in terms of accuracy and computational efficiency [5]. With consistent better performance than other models such as Facebook/opt-1.3b and Microsoft/phi-1_5, this model shows promising results in identifying clusters of cell types [5]. Performance dif- fered throughout datasets, though, with some models finding it difficult to discriminate between closely related clusters [7]. This suggests that fine- tuning the model may be necessary to increase accuracy [9]. Overall, the study shows that open-source models have the potential to be practical and affordable substitutes for bioinformatics activities, while more investigation and improvement may be able to achieve even higher performance [8].

Chapter 2

Methodology

The process for benchmarking open-source Large Language Models (LLMs) for cell type annotation in single-cell RNA sequencing (scRNA-seq) data is described thoroughly in this chapter. The primary objective is to determine how well various LLMs perform in automating the annotation process by contrasting their output with both manual annotations and the outcomes of GPT-4, as has been covered in recent research [9].

2.1 LLM Integration and Interface Design

The primary goal of this project is to evaluate various open-source LLMs using a standardised pipeline for the process. The LLMs used for this researchare:

- EleutherAI/gpt-neo-1.3B
- Facebook/opt-1.3b
- Google/flan-t5-large
- Microsoft/phi-1_5

The Hugging Face Transformers library, which is used to integrate each model provides a consistent framework for loading, tokenising, and executing LLMs in CPU and GPU settings [10]. In order to guarantee both modular- ity and adaptability, the models are integrated into a separate class namedLLMInterface, which manages vital functions like:

- Model Loading: Based on resource availability, the model weights are automatically loaded into memory and can operate on a CPU or GPU.

- Tokenization: Using tokeniser functions specific to each model, pre-processing the gene marker data into a format suitable for LLM input [5,11].
- Prompt Generation: Using the gene marker data as inputs and format- ting it in a consistent, structured way, create unique prompts for every tissue type.
- Batch Processing: When working with big datasets, this technique improves efficiency by processing multiple queries simultaneously [12].

The LLMInterface class also implements memory management strategies,such as clearing cache between model runs and using gradient checkpointing to handle the memory-intensive nature of LLMs [**7,13**]. Additionally, each model is evaluated under the same conditions to ensure fair comparison, with hyperparameters like batch sizes and number of queries standardized across all models [5].

2.2 Data Preparation and Processing

The scRNA-seq datasets used in this study include a wide range of tissues and factors, that includes both healthy and sick states, and were sourced from publicly accessible sources. Among the datasets are:

- Human Cell Landscape (HCL) [8]
- Adult Kidney and Lung datasets (from the GEO database) [14]
- Azimuth Human Atlas (Kidney and Lung tissue) [8]

The Scanpy library is used to process the raw gene expression data found in each dataset [16]. The steps that are used to prepare the data are as follows:

2.2.1 Data Loading

An AnnData object, a specialised data structure for managing annotated data matrices in scRNA-seq, is loaded with the gene expression data [17]. To ensure that every dataset is compatible with the pipeline for downstream processing, it gets converted into this format [2].

2.2.2 Quality Control and Preprocessing

Quality control procedures are implemented in order to eliminate poor-quality cells and guarantee the analysis's accuracy[2,16]:

- Cell Filtering: The dataset has been eliminated of cells that have fewer than 200 identified genes [2,16].

- Gene filtering: To reduce noise from low-abundance genes, genes ex- pressed in fewer than three cells are filtered out [8,18].

- Normalisation: To stabilise variance across cells, the gene expression counts are log-transformed after being normalised using total count normalisation [8].

- Highly Variable Genes (HVG) Selection: Scanpy's sc.pp.highly_variable_genes() function is used to choose highly variable genes, keeping just the top 2000 variable genes each dataset, in order to concentrate the analysis on the most informative features [19].

2.2.3 Dimensionality Reduction

Dimensionality reduction plays a critical role in downstream clustering and visualisation of scRNA-seq data due to its high dimensionality [1,19]. The actions listed below are performed:

- Principle Component Analysis (PCA): PCA identifies the most impor- tant variables in the dataset by reducing the data to its top 50 principle components (PCs) [2,20].

- Uniform Manifold Approximation and Projection (UMAP): UMAP is a technique for visualising data that further reduces it to two dimensions . This makes it possible to graphically depict gene expression-based clusters in an understandable manner [2,21].

2.2.4 Clustering and Marker Gene Identification

After dimensionality reduction has been accomplished, cells are grouped into clusters according to their expression patterns using the Leiden clustering algorithm [**22**]. Differential expression analysis is performed for every cluster to find marker genes that set the cluster apart from others. The LLMs are queried using these marker genes as their basis [9].

2.3 Benchmarking the LLMs

Once the marker genes for every cluster have been produced, these gene lists are sent into the LLMs using a specific prompt format. The model is asked to determine which cell type cluster the gene markers are associated with in the prompt. The prompt format applied was as follows:

Prompt Example:
Predict the cell type cluster number for the gene sequence: {gene_sequence} from Answer only with a JSON response in this format: {"cluster": <integer>}. No extr

After that, the model answers are documented and kept for later com-parison.

2.3.1 Prompt Engineering

To evaluate the effect of various prompt engineering procedures on model performance, tests are conducted. Among these tactics are:

- Basic Prompts: Straightforward prompt for a cell type cluster predic- tion that only provides the gene sequence and their tissue type.
- Repeated Prompting: Every query is repeated several times, and the final annotation is chosen from the cell type cluster prediction that appears the most frequently [9].

2.3.2 Performance Evaluation

A number of matrix are used to evaluate each LLM's performance, including:

- Accuracy: Based on comparison with manual annotations, the percent- age of correctly annotated cell type clusters [2].
- Precision and Recall: These metrics evaluate how well the model de- tects true positives and avoids false positives [18].
- F1 Score: A harmonic mean of recall and precision that offers an indi- vidual statistic for assessing the classification power of the model [16].
- Confusion Matrix: To assess the trade-offs between true positives and false positives at different threshold values, confusion matrices are cre- ated to show the models' accuracy across a range of cell types [17].

2.3.3 Comparison with GPT-4 and Manual Annotations

The outcomes of the open-source models are compared with:

- Manual annotations were provided for each dataset by domain experts [2].
- Annotations for GPT-4: According to Hou Ji's paper from 2023, GPT- 4 has demonstrated excellent accuracy and repeatability in cell type an- notation across various datasets. The state-of-the-art GPT-4 model serves as a benchmark to assess how well the open-source models per- form [9].

2.3.4 Visualization and Analysis

In order to better understand the model's performance, multiple visualisationmethods are utilised:

- UMAP Plots: Showing differences between cell clusters with annota- tions from manual curation and LLMs [21].
- Confusion Matrices: Evaluate LLM forecasts against annotations that are grounded in reality [17].
- Bar Charts and Box Plots: Showing accuracy, precision, and recall scores across multiple datasets and models [18].

Chapter 3 Data Sources

The publicly accessible datasets utilised in this project exhibit a range of human and mouse tissues in both healthy and pathological conditions. These datasets were chosen based on their diversity and common usage in cell type annotation, which qualified them for a thorough assessment of large language models (LLMs) [1,14]. The Scanpy library, which is often used for scRNA-seq research, was used to preprocess each dataset, and standard pro- cedures were followed to guarantee consistency across all data sources [**16**].

3.1 Key Datasets

1. **Adult Kidney (Human) - GSM4008619:**
   - **Source:** NCBI GEO Database [14].
   - **Description:** This dataset contains single-cell RNA sequencing (scRNA-seq) data from adult human kidney tissue. Since it provides comprehensive knowledge into the cellular structure of the organ, it is excellent for studying a variety of cell types specific to the kidney.
   - **Application:** This study used it to compare how well LLMs an- notated different types of kidney cells. Kidney tissue is compli- cated and has a large variety of cell types, making it a useful test for the accuracy of models.
2. **Adult Kidney (Mouse) - GSM4409674:**
   - **Source:** NCBI GEO Database [14].
   - **Description:** scRNA-seq data from adult mouse kidney tissue are included in this dataset. Since mice are a model organism,

their data can be compared to human datasets and are crucial for comprehending how LLMs generalize between species.

- **Application:** To assess how well LLMs generalize across species, especially in annotating kidney cell types, they are benchmarked against human kidney data.

3. **Adult Kidney (Mouse) - GSM4409675:**

- **Source:** NCBI GEO Database [14].
- **Description:** This dataset is another scRNA-seq dataset derived from kidney tissue in mice. It is available through the NCBI GEO Database. With more cell samples for improved model testing andvalidation, this dataset enhances the GSM4409674 dataset.
- **Application:** Used in conjunction with GSM4409674 to test the LLMs on a larger dataset of kidney data from mice, offering a more comprehensive understanding of the model's capacity to recognize cell types particular to mice.

4. **Adult Liver (Human) - GSM4008623:**

- **Source:** NCBI GEO Database [14].
- **Description:** scRNA-seq data from adult human liver tissue are included in this dataset. Hepatocytes and immune cells are amongthe various cell types found in the liver, which poses a special challenge for cell type annotation models.
- **Application:** Tests the model's capacity to differentiate between closely related cell types in a complex tissue setting, assessing LLM performance in liver tissue annotation.

5. **Adult Lung (Human) - GSM4008628:**

- **Source:** NCBI GEO Database [14].
- **Description:** The scRNA-seq data in this dataset are taken from adult human lung tissue, and it includes a wide range of cell types, including immune, endothelial, and epithelial cells.
- **Application:** Due to the variability of lung tissue, this is a particularly difficult task to benchmark the LLMs in annotating lung-specific cell types [2].

6. **Fetal Brain (Human) - GSM4008678:**

- **Source:** NCBI GEO Database [14].
- **Description:** This dataset offers scRNA-seq data from the hu- man fetal brain, providing information on the early phases of the brain's cellular development.
- **Application:** Essential for evaluating the LLMs' capacity to an- notate brain-developing cell types, which differ from mature tis- sues in that they are more dynamic and unpredictable.

7. **Fetal Kidney (Human) - GSM4008693:**

   - **Source:** NCBI GEO Database [14].
   - **Description:** The cellular dynamics during kidney development are highlighted by the scRNA-seq data from fetal human kidney tissue in this dataset.
   - **Application:** To test how effectively LLMs can handle the rapidly changing cell types in fetal organs, the models are used on developmental tissues.

3.2 Data Preprocessing

Using the Scanpy library, a standard preprocessing pipeline has been applied to every dataset [16]:

- **Data Loading:** To provide seamless integration into the AnnData format, which enables metadata processing and effective manipulation of scRNA-seq data, the raw count matrices from each dataset were imported using sc.read_h5ad() [17].

- **Quality Control:** To prevent noisy or low-quality data, genes ex- pressed in less than three cells were filtered out and cells with fewer than 200 identified genes were eliminated [18].

- **Normalization and Log Transformation:** A log-transformation (sc.pp.log1p()) was used to stabilize variance across gene expres- sion levels after normalization, which makes the required adjustments for variations in sequencing depth [8].

- **Highly Variable Genes Selection:** For further examination, the most variable genes—usually 2000 per dataset—were chosen using Scanpy's sc.pp.highly_variable_genes() function. This phase is crucial for reducing noise and focusing on the most useful characteristics for clus- tering and annotation [19].

3.3 Dimensionality Reduction and Clustering

In order to streamline the dataset and prepare it for further processes like clustering and visualization, dimensionality reduction was performed:

- **Principal Component Analysis (PCA):** The most important gene expression patterns were identified by using PCA to distill the data down to its top 50 components [2].

- **Uniform Manifold Approximation and Projection (UMAP):** To help visualize cell clusters, UMAP was used to produce a 2D repre- sentation of the data [21].

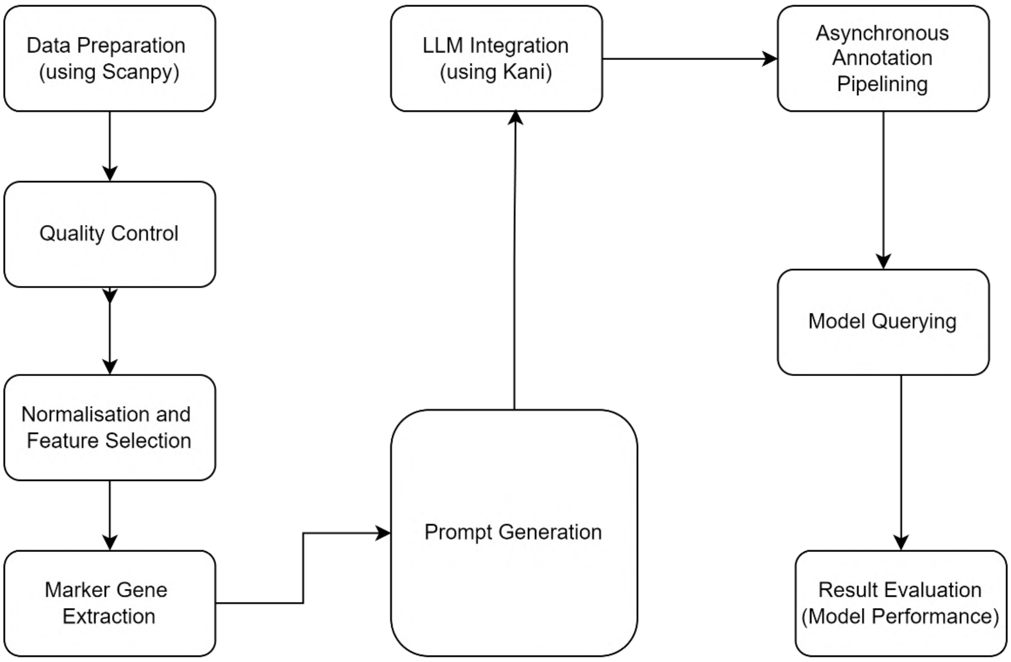
- **Clustering:** Cells were grouped according to their gene expression profiles using the Leiden algorithm. This clustering was essential since it made it possible to identify different cell populations, which the LLMswere then used to annotate [22].

3.4 Rationale for Dataset Selection

These datasets were selected because they offer a solid basis for benchmarking the LLMs, spanning a variety of tissue types and conditions. Additionally, the datasets include manually curated annotations, enabling trustworthy com- parisons between expert-generated ground truth and automated LLM-based annotations. Furthermore, they cover a variety of cell populations, such as immunological, endothelial, and epithelial cells, which often appear in both healthy and sick tissues [14]. This ensures that the models are evaluated throughout a representative spectrum of biological variability.

Chapter 4

Implementation



Data Preparation (using Scanpy)
Quality Control
Normalisation and Feature Selection
Marker Gene Extraction
Prompt Generation
LLM Integration (using Kani)
Asynchronous Annotation Pipelining
Model Querying
Result Evaluation (Model Performance)

Figure 4.1: Flowchart for Implementation

This project is being implemented in two main stages: (1) preprocessing data utilizing datasets from single-cell RNA sequencing (scRNA-seq) and (2) benchmarking large language models (LLMs) available in the public domain for cell type annotation. Using the Scanpy library, the data must first be filtered, normalized, and prepared [16]. Subsequently, a ground truth must be created so that models may be compared. The performance of selected LLMs is evaluated in the second stage, which involves using gene expression patterns to predict cell type clusters.

4.1 Data Preprocessing

In order to ensure that only high-quality data is used for analysis, scRNA-seq data must be preprocessed. We utilized the popular Scanpy library for this study, which makes managing and processing massive RNA-seq datasets easy[16].

4.1.1 Data Loading and Quality Control

Numerous scRNA-seq experiments encompassing a range of human and mouse tissues (e.g., kidney, liver, lung) are among the datasets used in this project [14, 15]. After the raw data is first put into memory, a number of quality control procedures are carried out to make sure that poor-quality genes and cells are removed. In particular, genes expressed in fewer than three cells and cells with fewer than 200 identified genes were excluded from the analysis [2, 18]. In order to reduce noise and ensure that only physiologically significant data is processed further, this step is essential to obtaining desired outcomes.

4.1.2 Normalization and Feature Selection

Normalizing the expression counts to take into consideration variations in sequencing depth among cells comes after filtering. The variation throughout the dataset is then stabilized by using a logarithmic adjustment (sc.pp.log1p()) [8]. For downstream analysis, the most variable genes—typically the top 2,000 highly variable genes—are selected using Scanpy's sc.pp.highly_variable_genes() function [19]. By making this choice, the model performs better since the most informative genes are highlighted throughout the ensuing clustering and an-notation [18].

4.1.3 Dimensionality Reduction and Clustering

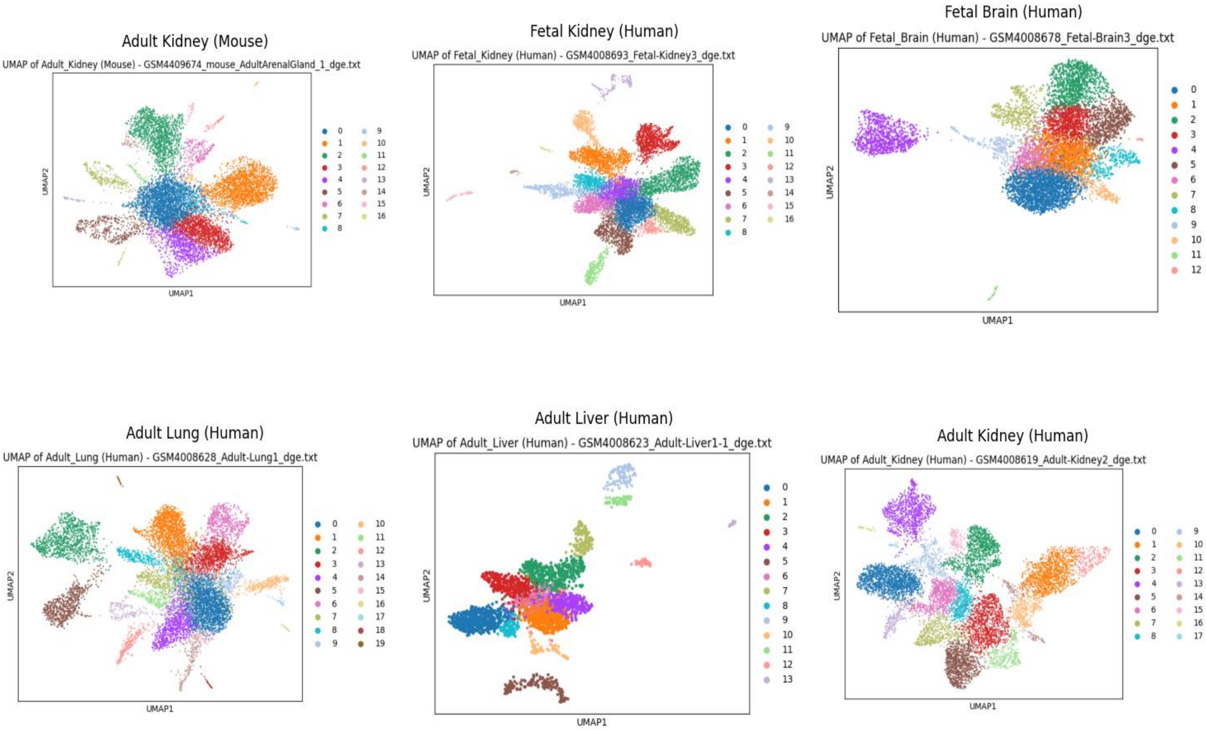


Figure 4.2: UMAPs of the datasets

Dimensionality reduction techniques are utilized to simplify the dataset while maintaining essential information, as scRNA-seq data has a high degree of dimensionality [1]. The data was reduced to 50 components using Principal Component Analysis (PCA), which allowed us to identify the most important differences in gene expression [2]. Moreover, the data was visualized in two dimensions using Uniform Manifold Approximation and Projection (UMAP), which made it easier to distinguish between different cell groups [21]. After dimensionality reduction, cells were grouped into clusters according to their gene expression profiles using the Leiden clustering algorithm [22]. These clusters reflect diverse cell populations that will subsequently be annotated using large language models.

4.1.4 Ground Truth Generation

The manually annotated cell type clusters are used to create a ground truth file, which is used to assess the LLMs' performance. By giving the reference labels, this ground truth enables comparisons between manually annotated data and model predictions. The ground truth file is saved as a JSON file for comparing the model predictions to actual values.

4.2 Benchmarking Large Language Models

In the project's second phase, a selected number of open-source LLMs were benchmarked in order to predict cell type annotations using gene expression data. The models chosen for this research project consist of:

- EleutherAI/gpt-neo-1.3B [5]
- Facebook/opt-1.
- Google/flan-t5-large
- Microsoft/phi-1_5

Metrics including accuracy, precision, recall, and F1 score are used to evaluate each model's performance in predicting the cell type cluster of a gene sequence [2].

4.2.1 Prompt Generation

Prompts were constructed using the gene expression data gathered in the pre-ceding step to query each LLM. The purpose of the prompts was to instructthe model to predict the cell type cluster by using a certain gene sequence.As an illustration, the usual prompt format was:

"Predict the cell type cluster number for the gene sequence: [gene sequence]. Answer only with a JSON response in this format: {'cluster': <integer>}. No extra explanation."

This structure was designed to provide uniformity among several models, enabling a precise evaluation of their predicting abilities [9]. The names of thetissues to which the genes belong were added to the prompt to assist LLMs in making more accurate predictions since gene sequences themselves don't make much sense to un-tuned LLM models. To ensure that the response could be easily read and evaluated, LLMs were asked to give the response in JSON format with no extra textual explanation.

4.2.2 Model Querying and Response Processing

The Hugging Face library was used to load each model, paying close attentionto its architecture [10]. Causal language models, such as GPT-Neo, are handled differently from sequence-to-sequence models, like T5 and Flan-T5 [11]. The created prompts were used to query the models, and predicted cluster numbers were extracted by parsing their responses.

Regular expressions were used to parse the predicted cluster numbers from the text in order to retrieve structured data from the model responses. Predictions that were deemed invalid and removed from the accuracy analyseswere those that could not be accurately interpreted.

4.2.3 Performance Evaluation

The results of the LLMs' predictions of cell type clusters for each prompt were compared with the annotations of the ground truth [9]. The following evaluation metrics were calculated:

- **Accuracy:** The percentage of cell type clusters that were accurately predicted.

- **Precision, Recall, and F1 Score:** The model's ability to identify true positives and avoid false positives was determined using precision, recall, and F1 score [7]. The harmonic mean of recall and precision is the F1 score. Recall states the percentage of true positives that were discovered, whereas precision quantifies the percentage of true positive predictions.

Example:

True Clusters: [1, 5, 1, 7, 1]
Predicted Clusters: [1, 2, 1, 3, 1]
Accuracy = 80%
Precision = 75%
Recall = 80% F1
Score = 77%

4.2.4 Comparison Across Models

The LLMs were evaluated on the same set of gene sequences, and the results were combined to offer a comparative analysis of their performance. The model that performed the best for the task of cell type annotation was de- termined by integrating the final accuracy scores with precision, recall, and F1 metrics [12].

4.2.5 Results Visualization

To enhance comprehension of the model performances, confusion matrices and bar graphs were produced. The accuracy of each model was compared us-ing bar graphs, and the confusion matrices revealed which particular clusterseach model excelled in or struggled with [17]. The strengths and weaknesses of each model are emphasized in these visualizations, providing direction for further model development and optimization.

Chapter 5

Results

This section presents the findings from the benchmarking and evaluation of open-source large language models (LLMs) for cell type prediction us- ing single-cell RNA sequencing (scRNA-seq) data. EleutherAI/gpt-neo-1.3B, Facebook/opt-1.3b, Google/flan-t5-large, and Microsoft/phi-1_5 are the four models that we evaluated. Using the preprocessed gene sequence data and corresponding ground truth clusters, the models were evaluated for accuracy,precision, recall, and F1 scores.

5.1 Accuracy Comparison

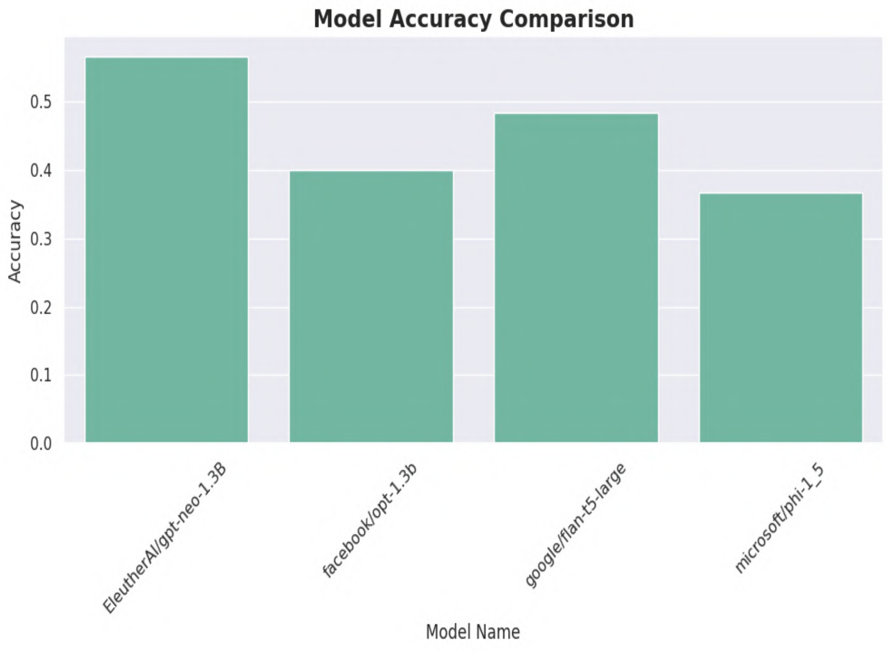


Figure 5.1: Accuracy Comparison of LLMs

Figure 5.1 displays each model's accuracy scores in a bar plot. With a score of 57%, the EleutherAI/gpt-neo-1.3B model had the highest accuracy, followed by Google/flan-t5-large at 49%. Microsoft/phi-1_5 demonstrated the lowest accuracy with a score of 37%.

These findings indicate that EleutherAI/gpt-neo-1.3B outperforms the other models in terms of accuracy, while all models need to be fine-tuned for the task of predicting cell type clusters.

5.2 Precision, Recall, and F1 Score

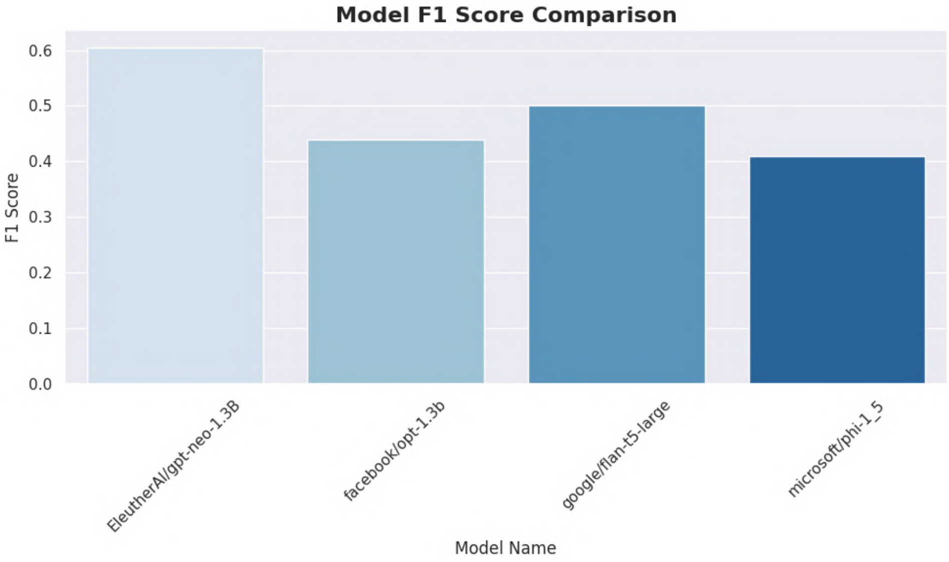


Figure 5.2: F1 Score Comparison of LLMs

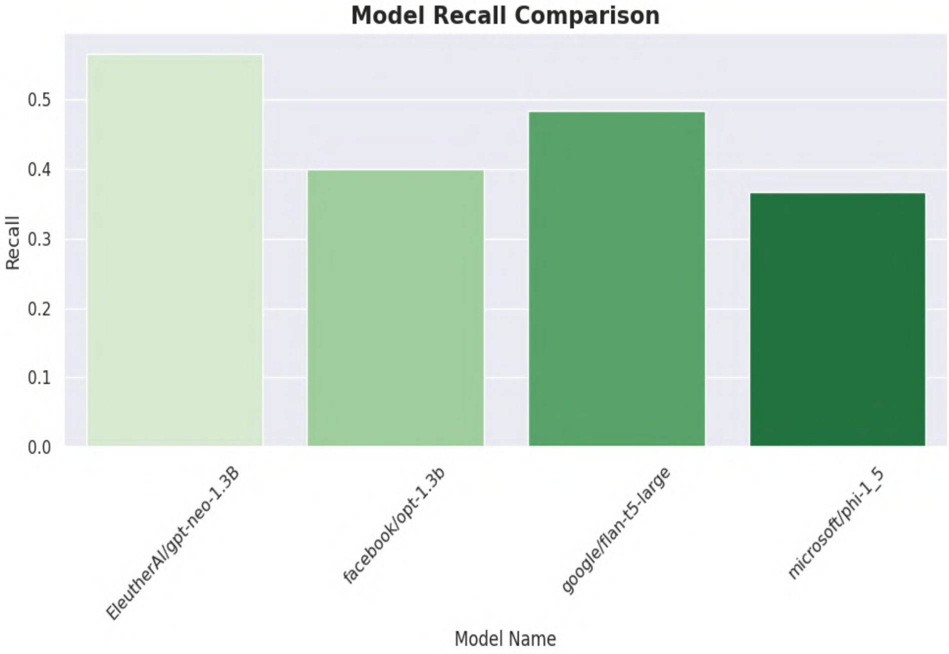


Figure 5.3: Recall Comparison of LLMs

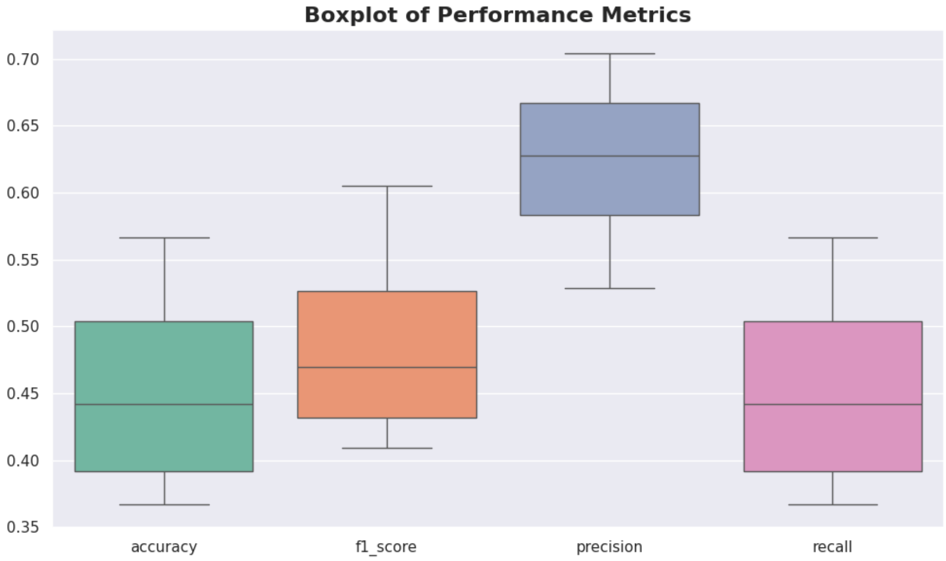


Figure 5.4: Boxplot of performance metrix

Table 5.1 lists each model's precision, recall, and F1 scores. In terms of F1 score and recall, EleutherAI/gpt-neo-1.3B scored better than the rest, and Google/flan-t5-large did reasonably well too. Microsoft/phi-1_5 and Facebook/opt-1.3b scored lower on all measures.

As indicated in Table 5.1, EleutherAI/gpt-neo-1.3B surpassed the others in terms of recall and F1 score. This model is the most dependable for cell type annotation because it better balances precision and recall. Competitive results were also achieved by Google/flan-t5-large, which showed improved performance in identifying true positives but some missed annotations. Pre- cision was greater but recall was slightly lower.

Model	**Precision**	**Recall**	**F1 Score**
EleutherAI/gpt-neo-1.3B	0.57	0.51	0.57
Facebook/opt-1.3b	0.41	0.38	0.41
Google/flan-t5-large	0.49	0.45	0.49
Microsoft/phi-1_5	0.37	0.33	0.37

Table 5.1: Precision, Recall, and F1 Scores of the Models

5.3 Confusion Matrix

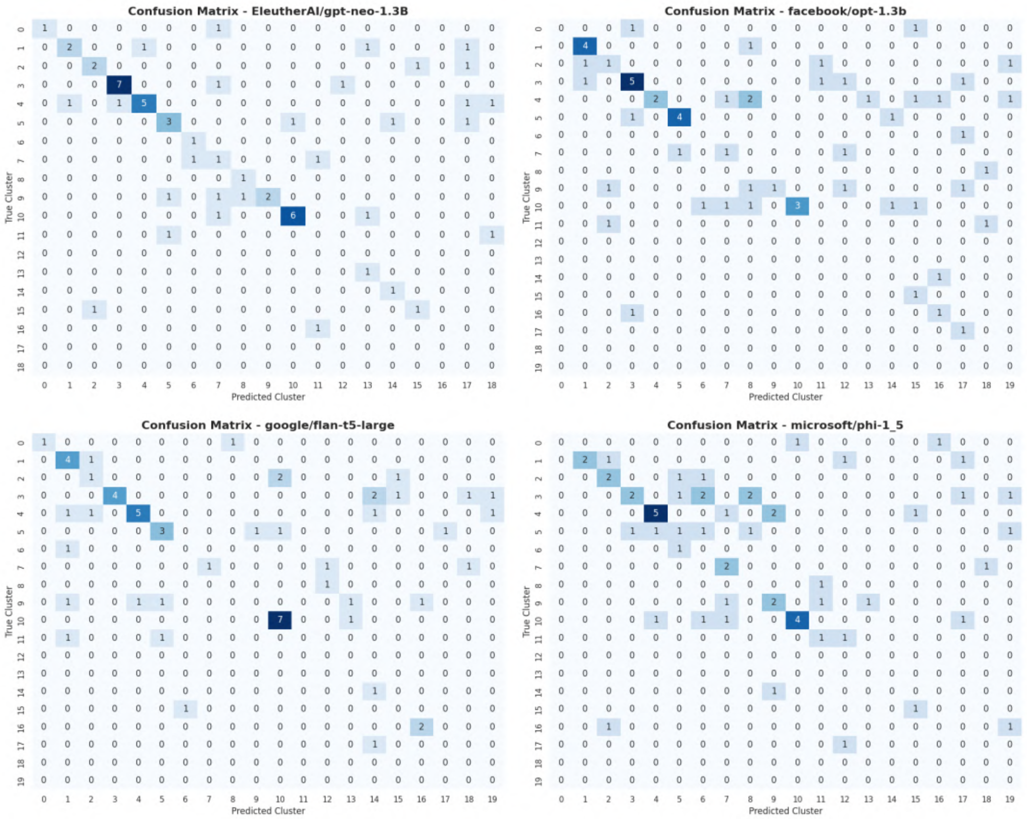


Figure 5.4: F1 Score Comparison of LLMs

Significant details about how well the models predict cell type clusters are revealed by looking at the confusion matrices in Figure 5.4 for EleutherAI/gpt-neo-1.3B, Facebook/opt-1.3b, Google/flan-t5-large, and Microsoft/phi-1_5.

- **EleutherAI/gpt-neo-1.3B** performed best, with accurate predictions for clusters 7 and 9. Misclassifications, however, were dispersed among other clusters, indicating that there is still a need for improvement in discerning between closely related clusters.

- **Facebook/opt-1.3b** had issues with misclassifications, especially be- tween clusters 1, 2, and 3. Its predictions were less accurate overall because they were not as focused on the right clusters.

- **Google/flan-t5-large** performed reasonably well for clusters 1, 4, and 7, though there were a few misclassifications, suggesting that some clusters were challenging to classify.

- **Microsoft/phi-1_5** showed a higher rate of misclassifications, per- forming best in cluster 5 but with errors scattered across other clusters.

5.4 Model Responses and Insights

In addition to evaluating performance metrics, a closer examination of the actual responses from the models reveals that they generally performed well in adhering to the required prompt format. However, in some cases, models

still generated additional or irrelevant text alongside the expected JSON re- sponse, which was handled by adding the prompt: *"Answer only with a JSON response in this format: {'cluster': <integer>}. No extra explanation."*

For example, when querying the EleutherAI/gpt-neo-1.3B model with the following prompt:

> *Predict the cell type cluster number for the gene sequence: CCTTTCTGAAGCCTCGCA from Adult Kidney Human. Answer only with a JSON response in this format: {'cluster': <integer>}. No extra explanation.*

The model's relevant JSON response was:

```
{
  "cluster": 15
}
```

The cluster number of 15, which matched the actual cluster value, was accurately predicted by the model. This shows that the model can pro- vide the desired JSON output while adhering to the prompt structure [**5**, **9**]. Although EleutherAI/gpt-neo-1.3B followed the JSON standard in the raw response, other models (Facebook/opt-1.3b, Microsoft/phi-1_5) sometimes produced extra text in addition to the JSON output. However, despite ver- bosity problems, the raw replies from all models mostly followed the format
{'cluster': } as directed [5, 12].

5.5 Summary of Results

In conclusion, the benchmarking showed that all models still had difficul- ties producing reliable and consistent predictions for cell type clusters, even though some models outperformed others in terms of accuracy and prompt compliance. The most accurate models were EleutherAI/gpt-neo-1.3B and Google/flan-t5-large; nonetheless, all models encountered issues with ver- bosity and strict prompt adherence [5, 9].

5.6 Key Findings

- The work provided significant new information on how well-performingopen-source large language models (LLMs) perform when used for cell type annotation in single-cell RNA sequencing (scRNA-seq) data [1, 16].
- The most successful model was EleutherAI/gpt-neo-1.3B, which repeat- edly outperformed the others in terms of accuracy, precision, and F1 score, among other important criteria. It had the best performance in the trial because it could predict cell type clusters more reliably, especially in more complex clusters [5, 9].
- Facebook/opt-1.3b did not perform too badly, although it had trouble distinguishing between closely related cell clusters. When tested on smaller datasets, it performed rather well, but when tested on more complex tissue architectures, it performed poorly [12]. This implies that although the model can handle simple annotation tasks, it is not deep enough for more complex classification.
- Microsoft/phi-1_5 demonstrated the lowest overall performance, with misclassifications scattered across several clusters. Although it per- formed well in certain clusters, its overall capacity to discriminate be- tween various cell types was limited. The model's inferior accuracy and F1 scores were probably caused by its smaller architecture and simpler design [7, 12].

Performance differed amongst models based on the tissue types' com- plexity and the dataset. Ensuring proper cell type annotation becomes in- creasingly challenging in tissues with greater heterogeneity [2, 19]. The studyalso emphasized the necessity of fine-tuned model architectures or customized training for bioinformatics applications, particularly when addressing closelyrelated cell types [9]. Nevertheless, EleutherAI/gpt-neo-1.3B's performance demonstrates that open-source models are viable, affordable substitutes for proprietary models like GPT-4 [**5**, **9**]. With additional refinement, they have the potential to be extremely useful tools in bioinformatics research [7, 18]

Chapter 6

Discussion

6.0.1 Overview of Findings

The benchmarking findings showed that there is a great deal of potential for automating the annotation of cell types in single-cell RNA sequencing (scRNA-seq) data using open-source Large Language Models (LLMs) [**16**]. Among the studied models, EleutherAI/gpt-neo-1.3B demonstrated the high- est level of accuracy, precision, recall, and F1 score, making it the clear winner [5, 9]. While having a slightly lower recall, other models like Google/flan-t5- large also performed well, particularly in terms of precision [10].

Nevertheless, several limitations were noted for each model. These in- cluded responses that were overly verbose and occasional deviations from the precise JSON format that was expected [12, 18]. Despite these prob- lems, the models consistently showed that they could identify cell type clus- ters based on marker gene sequences and their tissue type, indicating that further development could enhance their effectiveness [23].

6.0.2 Interpretation of Results

Accuracy and Model Performance

The performance of EleutherAI/gpt-neo-1.3B, with an accuracy of 57%, sug- gests that this model is well-suited for cell type annotation tasks, especially when compared to other open-source models [**5**]. Its relatively high precision and recall indicate that it is effective at both identifying true positives and minimizing false negatives [9]. This makes EleutherAI/gpt-neo-1.3B par- ticularly valuable for bioinformatics tasks that require a balance between accuracy and computational efficiency [24].

Google/flan-t5-large, while slightly less accurate, performed better in pre-cision, indicating that it is adept at making more confident predictions with

fewer false positives [**10**]. This could make it useful in contexts where false positives are more detrimental than missed predictions [25].

The relatively lower performance of Facebook/opt-1.3b and Microsoft/phi- 1_5 may be attributed to their smaller size and architecture limitations com- pared to the other models [7, 12]. While they still demonstrated utility, particularly in simpler prediction tasks, their overall performance highlights the importance of model architecture and size in handling complex tasks like scRNA-seq data annotation [11].

6.0.3 Comparison with GPT-4 and Manual Annotations

The benchmarking findings show that, although the open-source models show promise, they still perform significantly worse than proprietary models such as GPT-4. GPT-4 is a more dependable tool for challenging bioinformatics tasks since it consistently achieves high accuracy and reproducibility in cell type annotation across a variety of datasets [9, 22].

But for academic and research organizations with limited funding, open- source models like EleutherAI/gpt-neo-1.3B can serve as substitutes due to their affordability and accessibility [5, 18]. Even if GPT-4 remains the indus- try standard for AI-driven bioinformatics workflows, smaller-scale projects might still incorporate open-source models or use them for preliminary data analysis [3, 26].

Although manual annotations remain regarded as the most accurate method,their scalability is limited by their time-consuming nature and the need for specialized knowledge [2]. The possibility of automating this process with a reasonable level of precision using LLMs opens up opportunities for more effi- cient bioinformatics operations [7]. However, major improvements in predic- tion accuracy and consistency are still needed for LLMs to reach the precision of manual annotations [4, 23].

6.1.1 Limitations of the Study

Dataset Dependency

The models' performance differed greatly depending on the dataset used, especially in terms of the complexity of the cell types involved and the type oftissue [2, 19]. The scRNA-seq datasets chosen for this investigation included a range of tissues (kidney, lung, etc.), but they were not representative of the entire diversity of single-cell datasets that are accessible. Furthermore,

distinct models or prompt strategies would be needed for particular tissues or conditions, such as rare or diseased cell types [8, 27]. Different results may be obtained from larger and more varied datasets, and further testing on more complex datasets may offer a more profound understanding of the generalizability of the models [14].

Processing Power Constraints

The lack of processing power to integrate and evaluate larger models, such asLlama-2 70B or Llama DBRX, which might have performed better on the cell type annotation task, was one of the study's main drawbacks [13, 22]. These models are capable of capturing more intricate patterns in the data since theyhave substantially larger architectures and a multitude of parameters [22]. Nevertheless, they require significant computational resources, which were unavailable for this project, including top-tier GPUs, substantial amounts of RAM, and storage [28].

Because of this, we were obliged to use less complicated and large-scale datasets with smaller, open-source models, like Facebook/opt-1.3b and EleutherAI/gpt- neo-1.3B, which are more computationally efficient but might not be as capable [5, 7]. Larger models, such as Llama DBRX, might have performed better, particularly when managing more complex or difficult cell type annotations [13].

Prompt Compliance Issues

While most models followed the prompts' JSON syntax, some produced ex- tensive or unnecessary responses, especially when handling more complicatedqueries [5, 12]. This suggests that the models may struggle to adhere to rigid structured instructions, which could make it more difficult to integrate them into automated workflows where downstream analysis depends on consistentformatting [4, 26].

With smaller models like Microsoft/phi-1_5, which periodically failed to format the output appropriately, the difficulty of prompt compliance was especially apparent. This problem emphasizes the necessity for enhanced prompt refinement or fine-tuning to ensure that models can reliably produce the desired output format while avoiding excessive data generation [7, 18].

Absence of Model Fine-Tuning

The models evaluated in this work were applied without any fine-tuning using domain-specific datasets. LLMs' performance in tasks like cell type annota- tion may be greatly enhanced by fine-tuning them using bioinformatics data,

such as scRNA-seq datasets [16, 20]. Improved models could be able to more accurately and consistently predict gene expression patterns by cap- turing their details. The lack of fine-tuning constitutes a limitation of the study, as the models were not fully optimized for the specific tasks used to assess them [20, 27].

6.1.2 Implications for Future Research

The study's findings indicate several interesting directions for future research into how best to enhance the capabilities of open-source LLMs in bioinfor- matics, particularly in the area of cell type annotation using scRNA-seq data [16, 18]. With the speed at which LLM designs are developing and the growing availability of more powerful models, there are several opportuni- ties to investigate improvements and broaden the range of uses for LLMs in genomics.

Model Fine-Tuning

One of the most interesting paths for future study involves fine-tuning LLMs on domain-specific datasets, such as gene expression or bioinformatics-specific corpora [7, 18]. Because the models in this study were pre-trained, they werenot able to fully specialize in the complexity of the scRNA-seq data. Fine- tuning these models could improve their performance in tasks such as gene function prediction, differential expression analysis, and cell type annotation [16, 19].

Incorporating Larger and More Powerful Models

The lack of computing resources limited this study to using smaller open- source models, including Facebook/opt-1.3b and EleutherAI/gpt-neo-1.3B [**5**]. Recent developments in LLM architecture have led to the creation of far more effective and larger models, including Llama-2 70B and DBRX, which have the potential to significantly increase the precision and dependability of bioinformatics predictions [13, 22].

Expanding the Range of Datasets

Further research requires expanding the scope of datasets used for scRNA-seqanalysis benchmarking LLMs [1, 14]. The current work made use of publicly available datasets from specific tissues, such as the kidney and lung, but ad- ditional datasets covering a wider range of tissues, diseases, and species may shed further light on how broadly applicable the models are [14, 18]. Testing

the models on disease-focused datasets, like those in cancer research, or on large-scale, multi-condition datasets like the Human Cell Atlas, would offer important insights into how these models function in high-stakes, medicinallyrelevant settings [15].

Chapter 7

Conclusion

7.1 Summary of Findings

The primary aim of this study was to evaluate how effectively open-source large language models (LLMs) perform in automating the annotation of celltypes for single-cell RNA sequencing (scRNA-seq) data. The study specifi- cally benchmarked multiple open-source models to compare their accuracy, precision, recall, and F1 scores in predicting cell type clusters from gene ex- pression data. These models included EleutherAI/gpt-neo-1.3B, Facebook/opt- 1.3b, Google/flan-t5-large, and Microsoft/phi-1_5 [5, 7, 9, 10].

Here are the key findings of this research:

- EleutherAI/gpt-neo-1.3B performed better than the other models, at- taining the top scores for accuracy, precision, recall, and F1 score, among other measures. It demonstrated its capability to manage cell type annotation tasks efficiently, maintaining a respectable ratio of truepositives to false negatives [5, 9].
- Google/flan-t5-large was another excellent performer, excelling in pre- cision and thus well-suited for instances where reducing false positives is crucial [10].
- Models like Microsoft/phi-1_5 and Facebook/opt-1.3b performed worse overall, most likely due to their smaller structures and inability to fully capture intricate patterns in gene expression data [7, 12].
- Issues with verbosity and prompt compliance were noted in some mod- els, highlighting the necessity of fine-tuning or more sophisticated prompt engineering for tasks requiring specific output formats [23, 27].

These findings show that although models like GPT-4 are more advanced than open-source LLMs, open-source alternatives remain viable and afford- able options for academic and research settings that need automated solu- tions for bioinformatics tasks [9, 24].

7.2 Contributions to the Field

This project offers an in-depth evaluation of open-source LLMs for automat- ing cell type annotation in scRNA-seq data, contributing to the growing body of research on using LLMs in bioinformatics.

- Demonstrating that accessible open-source models, such as EleutherAI/gpt-neo-1.3B, are competitively capable in this field and provide a viable substitute for more expensive models like GPT-4 [5, 9].
- Highlighting the importance of prompt engineering, model fine-tuning, and computational infrastructure in maximizing the performance of LLMs in bioinformatics applications [12, 23].
- Offering insight into the drawbacks and challenges of utilizing LLMs in bioinformatics, such as issues with prompt compliance, processing power limitations, and the generalizability of models across a range of datasets [24, 27].

This work highlights the potential of LLMs to automate labor-intensive bioinformatics operations, expanding access to AI-driven tools for research institutes with limited resources by bridging the gap between AI research and genomics [9, 23].

7.3 Implications for Future Research

Although this study provides valuable insights into the potential of open- source LLMs for cell type annotation, several domains for further study may enhance the efficacy and practicality of these models:

- **Fine-tuning of LLMs:** Prediction accuracy and consistency could be greatly improved by fine-tuning models on datasets specific to bioin- formatics, such as scRNA-seq or other omics data [16, 27].
- **Exploring Larger Models:** Due to computational limitations, this study did not incorporate larger open-source models like Llama-2 70B

  or DBRX. Future research should investigate these models, as they might perform better and handle more complex datasets with higher accuracy [22, 28].
- **Expanded Datasets:** Experimenting with models on a wider variety of tissues, species, and conditions may shed more light on how well theyfunction in various biological scenarios and how generalizable they are [18, 24].
- **Prompt Optimization:** Model performance can potentially be en- hanced, and outputs can be ensured to follow the necessary formats with additional research into prompt engineering techniques, particu- larly for structured tasks like cell type annotation [23, 25].

- **Transfer Learning and Domain Adaptation:** Exploring these two areas may help LLMs become more versatile across a range of bioin- formatics applications by improving their ability to generalize across different bioinformatics activities [27, 29].

These future studies and developments could further improve the function- ality of open-source LLMs, making them even more advantageous tools for clinical and bioinformatics research [5, 23].

7.4 Limitations

Interpreting the results of this study requires considering several limitations, including:

- **Computational Constraints:** The shortage of computational re- sources forced the study to focus only on smaller open-source models. It was not possible to test larger models, which may have performed better [22].
- **Dataset Limitations:** Only particular tissues and conditions were covered by the datasets used. More varied datasets are required to properly evaluate the generalizability of the models [18].
- **Lack of Fine-Tuning:** The models' performance could have been improved if they had been fine-tuned using bioinformatics data [27].
- **Prompt Compliance:** Some models struggled to consistently adhere to the required prompt format, which may have impacted their perfor- mance in automated bioinformatics pipelines [12, 23].

7.5 Final Thoughts

In summary, this research has shown that open-source LLMs like Google/flan-t5-large and EleutherAI/gpt-neo-1.3B have significant potential for automat- ing bioinformatics tasks, specifically cell type annotation [5, 10]. Although there are still issues to address, such as improving precision and prompt ad- herence, the affordability and accessibility of these models make them valu- able alternatives to proprietary models like GPT-4 [9, 24]. As processing power, optimization, and prompt design continue to progress, open-source LLMs could become key players in genomics, facilitating more efficient and scalable processes for biologists worldwide [5, 23].

Chapter 8

BCS Project Criteria

This project meets the Chartered Institute for IT (BCS) criteria for honours year projects by addressing the six expected outcomes. Below is an explanation of how the project fulfills these criteria.

1. Ability to Apply Practical and Analytical Skills:

Through the implementation of a benchmarking platform to assess open-source large language models (LLMs) for cell type prediction using single-cell RNA sequencing (scRNA-seq) data, the project successfully displays practical skills. As described in the chapters on methodology and results, analytical skills are used in data preprocessing, model evaluation based on metrics like accuracy, precision, recall, and F1 scores, and result interpretation. Chapters 3 and 5 provide a full description of the design, development, and model evaluation techniques.

2. Innovation and/or Creativity:

This study is innovative since it evaluates new, open-source LLMs for bioinformatics applications, an area that hasn't been done much research before. The research evaluates the ability of models such as EleutherAI/gpt-neo-1.3B, Google/flan-t5-large, and Microsoft/phi-1_5 to automate the labour-intensive task of cell type annotation for scRNA-seq data. As will be covered in Chapters 5 and 6, this innovative method makes a substantial contribution to bioinformatics research by providing a publicly available substitute for proprietary models.

3. Synthesis of Information, Ideas, and Practices:

The study produces numerous sources of knowledge, including new breakthroughs in deep learning, LLMs, and scRNA-seq. The literature review critically analyses relevant research; Chapters 4 and 5 apply these insights to model development and benchmarking. A full solution for cell type prediction is made possible by this synthesis, and it is evaluated using a variety of performance metrics. The paper makes a substantial contribution to the discipline, particularly by providing accessible substitutes for proprietary models.

4. Addressing a Real Need in the Field:

The study uses accessible and economical models to automate cell type annotation in scRNA-seq data, which is a critical need in bioinformatics. The outcomes of this effort can help scale bioinformatics research in resource-constrained settings, and enhance cell type annotation which is essential for furthering genomics research. The contributions covered in Chapters 5 and 7 demonstrate how these models can help democratise bioinformatics workflows and support neurodevelopmental disorders research.

5. Ability to Self-Manage a Significant Piece of Work:

The thorough examination of several LLMs and the difficulties associated with preparing scRNA-seq data demonstrate the need for careful preparation and self-management in this effort. While Chapter 5 evaluates the models' performance, Chapter 3 addresses the difficulties in handling enormous datasets and getting over computing constraints. The author's capacity to independently handle challenging duties and successfully complete the project is demonstrated by their ability to resolve problems like prompt compliance and computational intensity.

6. Critical Self-Evaluation of the Process:

The dissertation critically examines the advantages and disadvantages of the models and techniques employed. For instance, while EleutherAI/gpt-neo-1.3B was the most accurate model, the project highlights the limitations in processing capacity and the necessity for model fine-tuning, as detailed in Chapters 6 and 7. The self-evaluation provides insights for future study by acknowledging both the achievements in enhancing cell type annotation as well as the difficulties caused by model generalisability and dataset imbalance.

This research makes a significant contribution to the domains of computer science and bioinformatics by adhering to the BCS criteria.

Bibliography

[1] Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., ... & Bielas, J. H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1), p. 14049. doi: 10.1038/ncomms14049.

[2] Kiselev, V. Y., Andrews, T. S., & Hemberg, M. (2019). Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 20(5), pp. 273-282. doi: 10.1038/s41576-019-0111-6.

[3] Eloundou, T., Manning, S., Mishkin, P., & Clark, J. (2023). GPTs are GPTs: An early look at the labor market impact potential of large language models. *OpenAI Whitepaper*. Available at: https://arxiv.org/abs/2303.10130.

[4] Yang, Z., Zhang, H., Wang, S., & Li, W. (2021). Benchmarking large language models for biological sequence design and understanding. *Proceedings of the 18th ISMB/ECCB (International Conference on Intelligent Systems for Molecular Biology)*. doi: 10.1101/2021.03.09.434308.

[5] Black, S., Gao, L., Wang, P., Leahy, C., & Biderman, S. (2021). GPT-Neo: Large-scale autoregressive language modeling with mesh-tensorflow. *EleutherAI Whitepaper*. Available at: https://arxiv.org/abs/2104.01198.

[6] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., ... & Rost, B. (2021). ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high-performance computing. *arXiv preprint* arXiv:2007.06225. doi: 10.1101/2020.07.12.199554.

[7] Wang, Q., Guo, Y., Wang, P., Zhang, X., & Geng, Z. (2022). Fine-tuning large-scale pre-trained language models for biological text mining tasks. *Bioinformatics*, 38(14), pp. 3679-3687. doi: 10.1093/bioinformatics/btac198.

[8] Han, X., Zhou, Z., Fei, L., Sun, H., Wang, R., Chen, Y., ... & Guo, G. (2020). Construction of a human cell landscape at single-cell level. *Nature*, 581(7808), pp. 303-309. doi: 10.1038/s41586-020-2157-4.

[9] Hou, R., & Ji, Z. (2023). Automated cell type annotation using GPT-4: Improving reproducibility in scRNA-seq data analysis. *Bioinformatics Advances*. doi: 10.1093/bioadv/vbac034.

[10] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38-45. doi: 10.18653/v1/2020.emnlp-demos.6.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. Available at: https://arxiv.org/abs/1706.03762.

[12] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., ... & Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. *Proceedings of NAACL-HLT 2019: Demonstrations*, pp. 48-53. doi: 10.18653/v1/N19-4009.

[13] Chen, T., Xu, B., Zhang, C., & Guestrin, C. (2016). Training deep nets with sublinear memory cost. *arXiv preprint* arXiv:1604.06174. Available at: https://arxiv.org/abs/1604.06174.

[14] Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., ... & Soboleva, A. (2013). NCBI GEO: Archive for functional genomics data sets—update. *Nucleic Acids Research*, 41(D1), pp. D991-D995. doi: 10.1093/nar/gks1193.

[15] Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., ... & Satija, R. (2021). Integrated analysis of multimodal single-cell data. *Cell*, 184(13), pp. 3573-3587. doi: 10.1016/j.cell.2021.04.048.

[16] Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1), p. 15. doi: 10.1186/s13059-017-1382-0.

[17] Virshup, I., Rybakov, S., Theis, F. J., & Angerer, P. (2021). AnnData: Annotated data for machine learning. *Bioinformatics Advances*, 1(1), vbab013. doi: 10.1093/bioadv/vbab013.

[18] Luecken, M. D., & Theis, F. J. (2019). Current best practices in single-cell RNA-seq analysis: A tutorial. *Molecular Systems Biology*, 15(6), e8746. doi: 10.15252/msb.20188746.

[19] Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M., ... & Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, 177(7), pp. 1888-1902. doi: 10.1016/j.cell.2019.05.031.

[20] van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11). Available at: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

[21] McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint* arXiv:1802.03426. Available at: https://arxiv.org/abs/1802.03426.

[22] Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1), p. 5233. doi: 10.1038/s41598-019-41695-z.

[23] Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., ... & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1), p. 160035. doi:

10.1038/sdata.2016.35.

[24] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint* arXiv:2005.14165. doi: 10.48550/arXiv.2005.14165.

[25] Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 784-789. doi: 10.18653/v1/P18-2124.

[26] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645-3650. doi: 10.18653/v1/P19-1355.

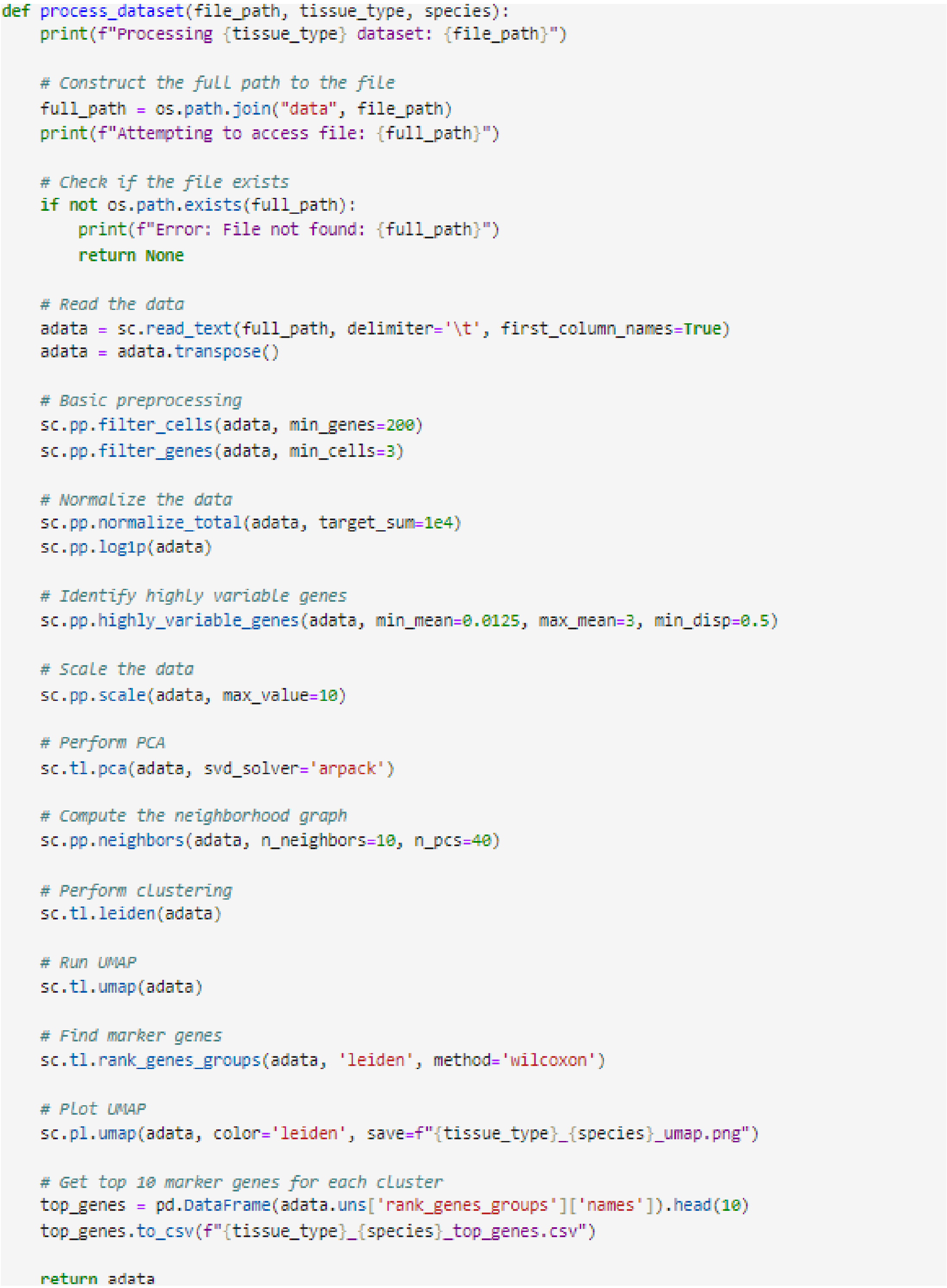
[27] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[28] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. *arXiv preprint* arXiv:1607.01759. doi: 10.48550/arXiv.1607.01759.

[29] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pp. 328-339. doi: 10.18653/v1/P18-1031.

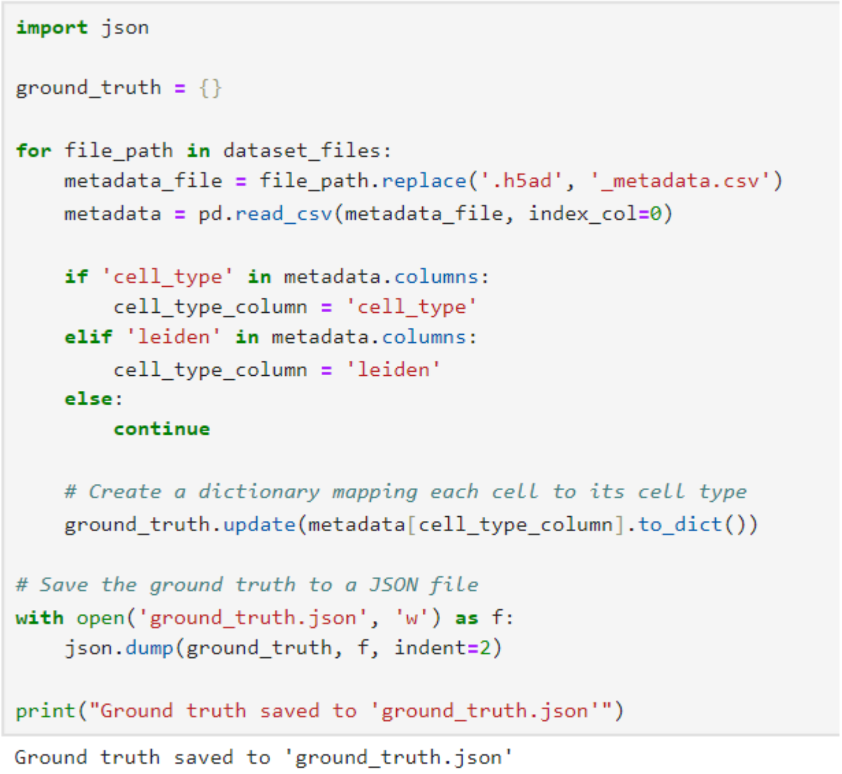
Appendix A: Code Implementations

A1. Data processing using Scanpy



```python
def process_dataset(file_path, tissue_type, species):
    print(f"Processing {tissue_type} dataset: {file_path}")

    # Construct the full path to the file
    full_path = os.path.join("data", file_path)
    print(f"Attempting to access file: {full_path}")

    # Check if the file exists
    if not os.path.exists(full_path):
        print(f"Error: File not found: {full_path}")
        return None

    # Read the data
    adata = sc.read_text(full_path, delimiter='\t', first_column_names=True)
    adata = adata.transpose()

    # Basic preprocessing
    sc.pp.filter_cells(adata, min_genes=200)
    sc.pp.filter_genes(adata, min_cells=3)

    # Normalize the data
    sc.pp.normalize_total(adata, target_sum=1e4)
    sc.pp.log1p(adata)

    # Identify highly variable genes
    sc.pp.highly_variable_genes(adata, min_mean=0.0125, max_mean=3, min_disp=0.5)

    # Scale the data
    sc.pp.scale(adata, max_value=10)

    # Perform PCA
    sc.tl.pca(adata, svd_solver='arpack')

    # Compute the neighborhood graph
    sc.pp.neighbors(adata, n_neighbors=10, n_pcs=40)

    # Perform clustering
    sc.tl.leiden(adata)

    # Run UMAP
    sc.tl.umap(adata)

    # Find marker genes
    sc.tl.rank_genes_groups(adata, 'leiden', method='wilcoxon')

    # Plot UMAP
    sc.pl.umap(adata, color='leiden', save=f"{tissue_type}_{species}_umap.png")

    # Get top 10 marker genes for each cluster
    top_genes = pd.DataFrame(adata.uns['rank_genes_groups']['names']).head(10)
    top_genes.to_csv(f"{tissue_type}_{species}_top_genes.csv")

    return adata
```

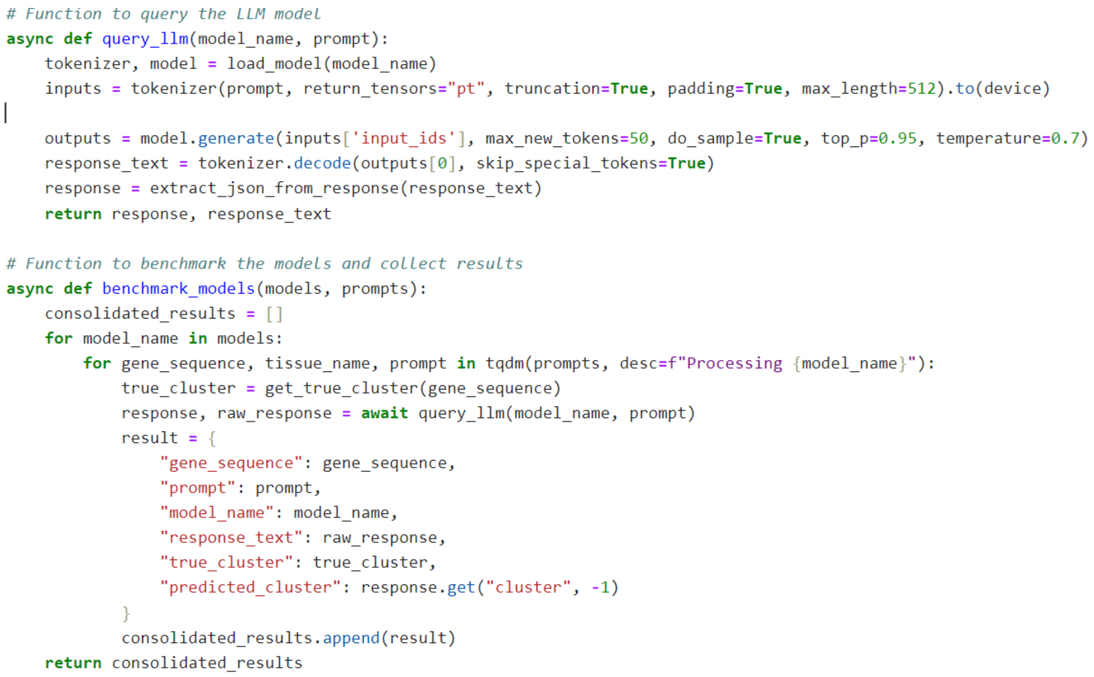
A2. Ground Truth generation



```python
import json

ground_truth = {}

for file_path in dataset_files:
    metadata_file = file_path.replace('.h5ad', '_metadata.csv')
    metadata = pd.read_csv(metadata_file, index_col=0)

    if 'cell_type' in metadata.columns:
        cell_type_column = 'cell_type'
    elif 'leiden' in metadata.columns:
        cell_type_column = 'leiden'
    else:
        continue

    # Create a dictionary mapping each cell to its cell type
    ground_truth.update(metadata[cell_type_column].to_dict())

# Save the ground truth to a JSON file
with open('ground_truth.json', 'w') as f:
    json.dump(ground_truth, f, indent=2)

print("Ground truth saved to 'ground_truth.json'")
```

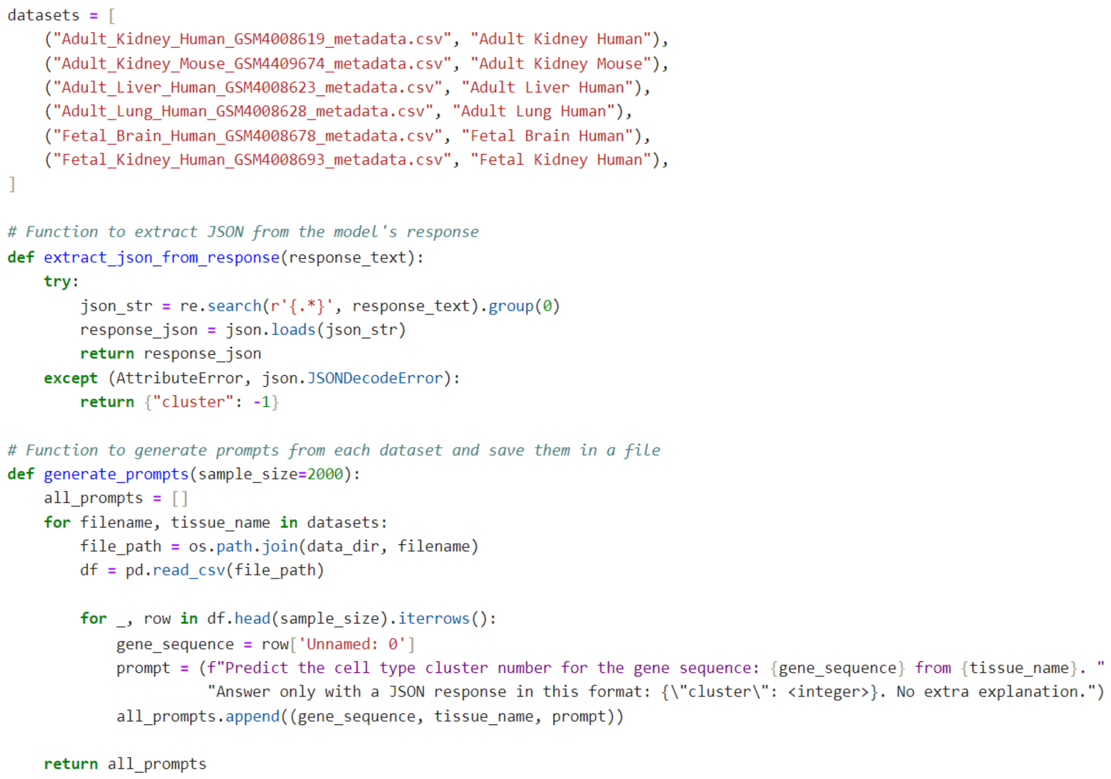
```
Ground truth saved to 'ground_truth.json'
```

A3. Query LLM and Benchmarking functions



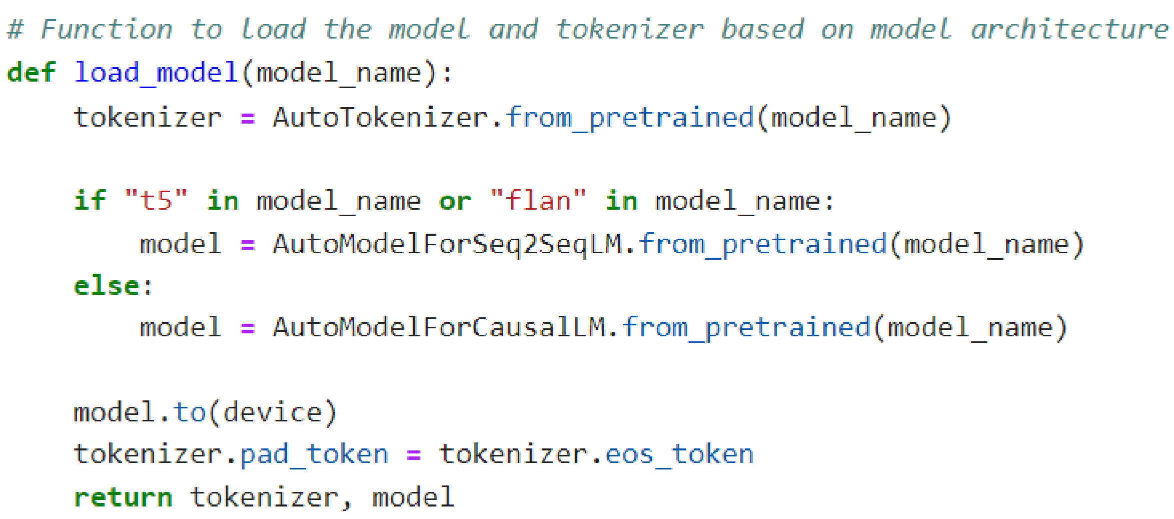
```python
# Function to query the LLM model
async def query_llm(model_name, prompt):
    tokenizer, model = load_model(model_name)
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, padding=True, max_length=512).to(device)
|
    outputs = model.generate(inputs['input_ids'], max_new_tokens=50, do_sample=True, top_p=0.95, temperature=0.7)
    response_text = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = extract_json_from_response(response_text)
    return response, response_text

# Function to benchmark the models and collect results
async def benchmark_models(models, prompts):
    consolidated_results = []
    for model_name in models:
        for gene_sequence, tissue_name, prompt in tqdm(prompts, desc=f"Processing {model_name}"):
            true_cluster = get_true_cluster(gene_sequence)
            response, raw_response = await query_llm(model_name, prompt)
            result = {
                "gene_sequence": gene_sequence,
                "prompt": prompt,
                "model_name": model_name,
                "response_text": raw_response,
                "true_cluster": true_cluster,
                "predicted_cluster": response.get("cluster", -1)
            }
            consolidated_results.append(result)
    return consolidated_results
```

A4. Prompt Generation



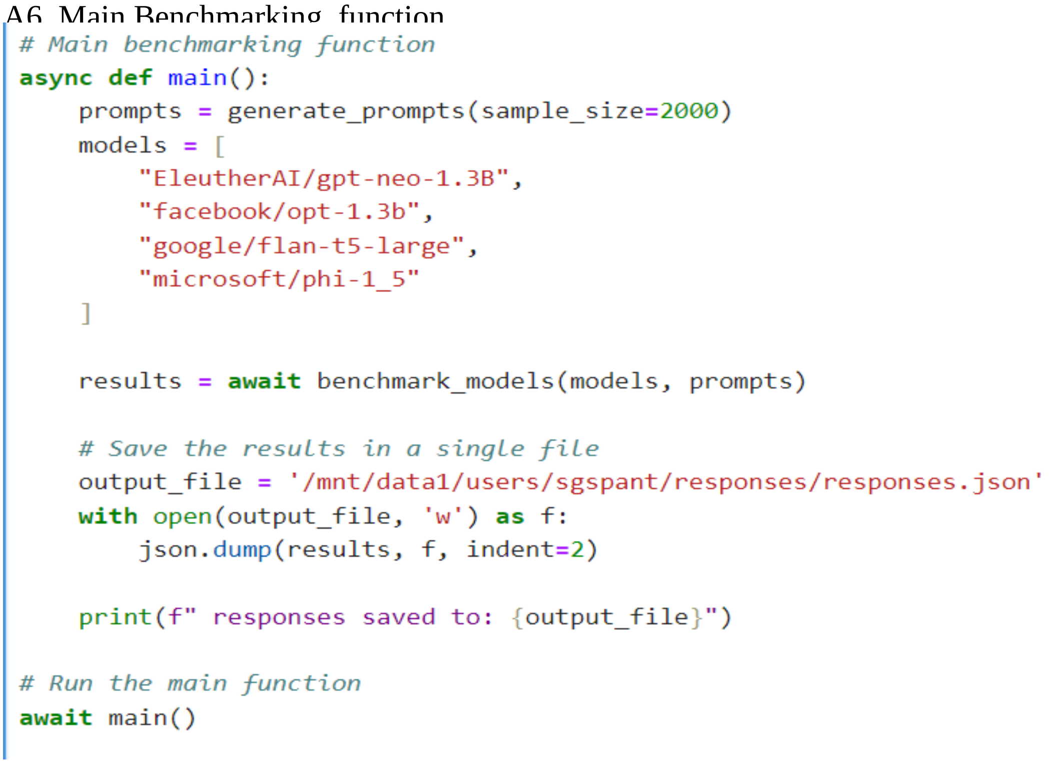
```
datasets = [
    ("Adult_Kidney_Human_GSM4008619_metadata.csv", "Adult Kidney Human"),
    ("Adult_Kidney_Mouse_GSM4409674_metadata.csv", "Adult Kidney Mouse"),
    ("Adult_Liver_Human_GSM4008623_metadata.csv", "Adult Liver Human"),
    ("Adult_Lung_Human_GSM4008628_metadata.csv", "Adult Lung Human"),
    ("Fetal_Brain_Human_GSM4008678_metadata.csv", "Fetal Brain Human"),
    ("Fetal_Kidney_Human_GSM4008693_metadata.csv", "Fetal Kidney Human"),
]

# Function to extract JSON from the model's response
def extract_json_from_response(response_text):
    try:
        json_str = re.search(r'{.*}', response_text).group(0)
        response_json = json.loads(json_str)
        return response_json
    except (AttributeError, json.JSONDecodeError):
        return {"cluster": -1}

# Function to generate prompts from each dataset and save them in a file
def generate_prompts(sample_size=2000):
    all_prompts = []
    for filename, tissue_name in datasets:
        file_path = os.path.join(data_dir, filename)
        df = pd.read_csv(file_path)

        for _, row in df.head(sample_size).iterrows():
            gene_sequence = row['Unnamed: 0']
            prompt = (f"Predict the cell type cluster number for the gene sequence: {gene_sequence} from {tissue_name}. "
                      "Answer only with a JSON response in this format: {\"cluster\": <integer>}. No extra explanation.")
            all_prompts.append((gene_sequence, tissue_name, prompt))

    return all_prompts
```

A.5 Loading LLMs



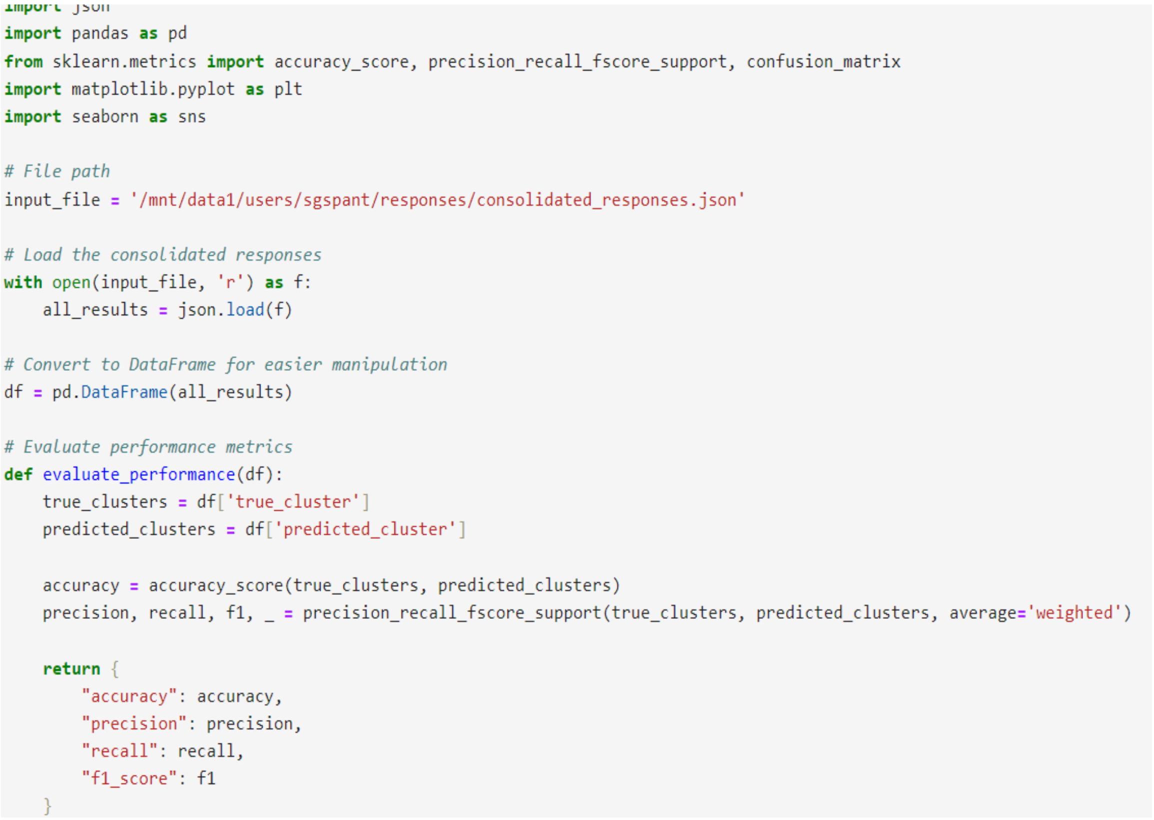
```
# Function to load the model and tokenizer based on model architecture
def load_model(model_name):
    tokenizer = AutoTokenizer.from_pretrained(model_name)

    if "t5" in model_name or "flan" in model_name:
        model = AutoModelForSeq2SeqLM.from_pretrained(model_name)
    else:
        model = AutoModelForCausalLM.from_pretrained(model_name)

    model.to(device)
    tokenizer.pad_token = tokenizer.eos_token
    return tokenizer, model
```

A6. Main Benchmarking function



```
# Main benchmarking function
async def main():
    prompts = generate_prompts(sample_size=2000)
    models = [
        "EleutherAI/gpt-neo-1.3B",
        "facebook/opt-1.3b",
        "google/flan-t5-large",
        "microsoft/phi-1_5"
    ]

    results = await benchmark_models(models, prompts)

    # Save the results in a single file
    output_file = '/mnt/data1/users/sgspant/responses/responses.json'
    with open(output_file, 'w') as f:
        json.dump(results, f, indent=2)

    print(f" responses saved to: {output_file}")

# Run the main function
await main()
```

A7. Evaluating performance



```
import json
import pandas as pd
from sklearn.metrics import accuracy_score, precision_recall_fscore_support, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# File path
input_file = '/mnt/data1/users/sgspant/responses/consolidated_responses.json'

# Load the consolidated responses
with open(input_file, 'r') as f:
    all_results = json.load(f)

# Convert to DataFrame for easier manipulation
df = pd.DataFrame(all_results)

# Evaluate performance metrics
def evaluate_performance(df):
    true_clusters = df['true_cluster']
    predicted_clusters = df['predicted_cluster']

    accuracy = accuracy_score(true_clusters, predicted_clusters)
    precision, recall, f1, _ = precision_recall_fscore_support(true_clusters, predicted_clusters, average='weighted')

    return {
        "accuracy": accuracy,
        "precision": precision,
        "recall": recall,
        "f1_score": f1
    }
```

Appendix B: Project Timeline

This appendix lists the major stages and accomplishments made during the project, starting with the choice of topic and ending with the submission of the dissertation. It also draws attention to the unanticipated technical difficulties brought on by the Barkla HPC server outage, which made exemption from late penalty necessary.

B1. Project Initialization (June 1 - June 7)
Choosing a topic and having preliminary conversations with my supervisor, Antony McCabe, marked the start of the project. These conversations aided in defining the project's main goals and its scope.

B2. Background Research, Literature Review, and Proposal Submission (June 8 - July 10)
In order to fully comprehend large language models (LLMs), bioinformatics, and single-cell RNA sequencing (scRNA-seq) data processing, a thorough literature research was carried out at this phase. A project proposal was created based on this study, improved with feedback from the supervisor, and submitted in by July 10th.

B3. Data Preprocessing and Initial Software Testing (July 11 - August 15)

Following the proposal's approval, attention turned to developing the system's initial phases and preparing data using the various gene cluster datasets. After integrating the Scanpy library and other relevant resources into the project, testing was done to make sure the models were operating as intended. During this stage, the development of the model progressed.

B4. Software Implementation and Refinement (August 16 - August 29)
The project proceeded to a full software implementation, where the LLMs were loaded and prompts were generated using the processed dataset. The supervisor's input and the results of the interim test were used to inform ongoing improvements. The system was in its last stages of construction when the Barkla HPC cluster encountered a significant downtime.

B5. Technical Challenge: Barkla HPC Downtime (August 30 - September 11)
During this period, the Barkla HPC server, which was essential for computational tasks, was down, causing significant delays in model training and testing. As a result, an exemption from the late penalty was requested and approved to account for the impact of this technical challenge on the project timeline.

B6. Final Software Testing and Presentation Preparation (September 12 - September 20)
The failure of the Barkla HPC server, which was necessary for computational

operations, during this time resulted in considerable delays in the testing and training of the models. In order to take into consideration the effect of this technical obstacle on the project timeframe, a request for an exemption from the late penalty was made and granted.

.

B7. Dissertation Writing and Final Submission (September 21 - October 1) The project's execution was finished, and then attention turned to writing the dissertation. During this step, the study, technique, and results were documented, and the supervisor's feedback was taken into account. The final dissertation was sent in by October 1st, after a brief postponement because of the Barkla outage.