**Open Problems - Multimodal Single-Cell Integration  
3rd place Model Summary**

**A. MODEL SUMMARY**

**A1. Background on you/your team**

Competition Name: Open Problems - Multimodal Single-Cell Integration

Team Name: Makotu (solo)

Private Leaderboard Score: 0.773518

Private Leaderboard Place: 3rd

Name: Makoto Hyodo

Location: Japan, Tokyo

Email: makotu1208@gmail.com

**A2. Background on you/your team**

* I have been working as a data scientist in Japan for about 7 years.

I usually analyze e-commerce sites and develop recommendation

related algorithms.

* I had no experience in the biotechnology field before this competition started.
* I decided to participate in the competition because I was genuinely interested in handling data in the bio domain as well.
* Roughly 150 hours were spent on this competition.

**A3. Summary**

For an explanation, check out the discussion on kaggle.

<https://www.kaggle.com/competitions/open-problems-multimodal/discussion/366428>

**A4. Features Selection / Engineering**

Multiome  
(For Multiome, preprocessing is important than feature engineering.)

The features were compressed into binary variables, which contributed somewhat. Features using word2vec technology were also created, which also contributed somewhat. This feature is a vector of the top 100 most highly expressed gene names. The method of creation is described in the notebook.

<https://www.kaggle.com/code/mhyodo/w2v-feature-sample>

Cite

For Cite, features involving clustering contributed to the accuracy.

The value obtained by averaging features for each leiden clustering contributed to the accuracy well. As with Multiome, features using the word2vec technique also contributed to accuracy to some extent.

**A5. Training Method(s)**

I used mlp and catboost as models. I created models with various combinations of features (different dimensions to compress with svd or lsi, several patterns of clusters, with and without word2vec features, etc.) and ensemble them. The ensemble is a simple weighted average based on the out-of-fold results.

As a result, the final result used the following number of models.

Multi: mlp 26 models catboost 3 models  
Cite: mlp 18 models catboost 2 models

There would also need to be a description of validation in the training methods.

As described in the Kaggle discussion, this is done by creating a model to classify data that is evaluated as PRIVATE and data that is not, and to identify data that is close to PRIVATE in the training data.  
To be precise, I selected the top 2000 cells from each donor's training data that are close to PRIVATE, and use them as the data for validation.  
For the final submission, I chose the submission with the highest CV above without considering LB, and it worked very well.

**A6. Interesting findings**

It was interesting to see how features related to clusters contributed to accuracy. I have never seen clusters contribute to accuracy in the table data I have come across, so I felt that they were useful for clusters, especially for "data where the graph structure is easy to assume," such as cells.  
  
It was also interesting that "ordinal" features such as word2vec contributed to the accuracy. To be honest, I did not have time to devise much this time, but I may be able to devise these order-related models further. If there are other cell-related data sets available, I felt that pre-training with them might contribute to further improvement of accuracy.

**A7. Simple Features and Methods**

I think that using only the important protein features as they are and the features related to clustering for cite will guarantee some accuracy.

On the other hand, I think that multi will be accurate enough if only the dimensionality reduction with okapi bm25 and lsi is used in the preprocessing.

If these are trained as a single model, I think it is simple, fast, and can guarantee a certain degree of accuracy.

**A8. Model Execution Time**

In the conditions using a GTX 1080 Ti GPU, the total learning time is about 12 hours and the prediction time is about 15 minutes.

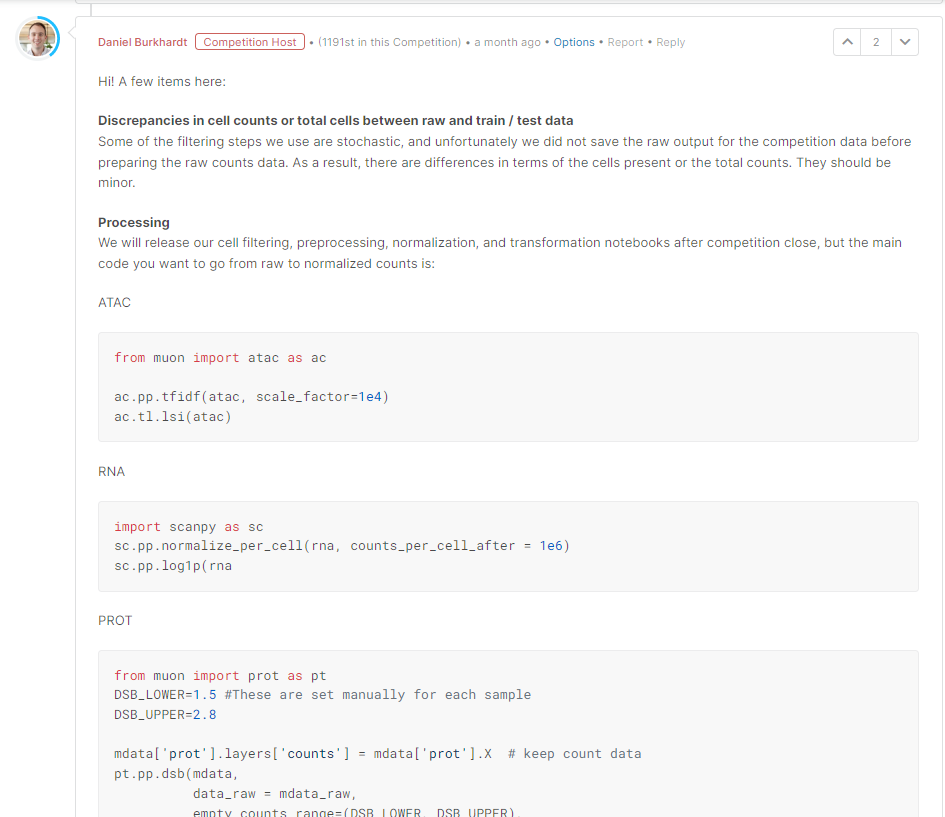
This training time is total of 50 models training.

If cite and multi are made into a single model of mlp each, the training would probably take about 15 minutes.

Prediction would take about 10 minutes instead of 15 minutes, since the shaping after the prediction is the main time consuming part.

**A9. References**

The pre-processing and libraries introduced by the host were very helpful to me without domain knowledge.



The libraries for bioinformatics you mentioned here, muon and scanpy, and their tutorials were also very helpful.

<https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html>

**B. SUBMISSION MODEL**

The code and data are stored in the "open-problems-multimodal-3rd-solution" folder in this zip. It is also available on github.

<https://github.com/makotu1208/open-problems-multimodal-3rd-solution>