710th Place Solution for the Open Problems – Single-Cell Perturbations

This is my first Kaggle competition. I want to thank the Kaggle team, competition organizers, and most importantly the Kaggle community for the opportunity for me to learn lots of fun ML.

Context

Business context:
https://www.kaggle.com/competitions/open-problems-single-cell-perturbations

Data context:
https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/data

"Judges" Prizes Scoring Rubrics

1. Integration of Biological Knowledge

Three biological knowledge concepts are used in our studies: blood cells differentiation, correlation between chemical structure and expression pattern, and model of gene expression.

1.1 Blood cells differentiation

Based on basic biology of blood cell differentiation, the progenitors of myeloid cells are separated from Lymphoid (B cells, T cells, NK cells) early on. Therefore, I expected that the gene expression and differential expression of Myeloid would have very distinct profiles from the rest of the cell type.

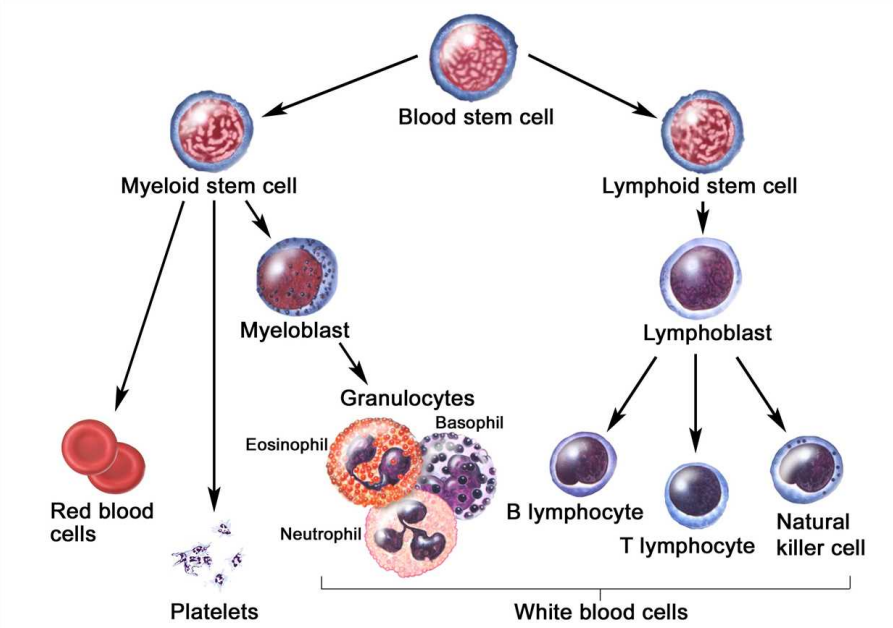

Figure 1: Blood cell differentiation. Source: https://www.creative-biolabs.com/stem-cell-therapy/static/img/Blood-Cells-Differentiation-from-iPSC-2.jpg

Based on this knowledge, I separate the model training for B cells and Myeloid cells. This led to improvement of the basic regression model from 0.754 to 0.743 in the public leaderboard. I also observe that the MSE of Myeloid in the training set is larger than those of B cells. For example, the basic elastic net has MSE 3.10 in B cells and 9.80 in Myeloid cells. Both of these evidence concur with blood cell differentiation that it will be harder to predict expression of Myeloid cells and the model for these two cell lines should be separated or factored in cell type.

1.2 Correlation between chemical structure and expression pattern

Predicting the effect of drugs based on chemical structure is challenging due to the extreme delicateness of how the molecule composition and structure affect the cell. Historically, two stereoisomers (type of isomer with same molecule composition and connectivity, but different in spatial arrangement) of Thalidomide have vastly different toxicity and lead to many birth defects due to side effects of one of the forms. The differential impact of the same drug on multiple cell types is also well known and also an inspiration for this competition.

Despite the complexity, I believe that many drug developments take inspiration from existing drugs, so I might be able to use small amounts of small molecules

for study just like this competition without a massive collection of drug response training sets.

To test this, I calculate the Tanimoto Similarity, a Jaccard-like measure of molecule similarity, for each pair of SMILES molecules (Figure). Most molecules have little similarity to others, but there are multiple pairs that have similarity above 0.2 and five of them are above 0.5. The two molecules of the pairs in top 5 highest Tanimoto Similarity pairs are all from the same group (drugs end with -tinib are tyrosine kinase inhibitor; Decitabine and Azacitidine are cytidine analog; both Navitoclax and ABT737 are anti-cancer drug)

```
[[0.7466666666666667, 'Masitinib', 'Imatinib'],
 [0.6590909090909091, 'Decitabine', 'Azacitidine'],
 [0.618421052631579, 'Cabozantinib', 'Foretinib'],
 [0.525, 'Navitoclax', 'ABT737'],
 [0.5227272727272727, 'Nilotinib', 'Imatinib']]
```

My hypothesis was that if the molecule similarity among pairs is highly correlated with either Euclidean or Pearson distance of two vectors of differential gene expression affected by those same pair of molecules, I could possibly predict the differential gene expression of unmeasured drug based on drop in replacement or linear model of differential gene expression of drugs that have high molecular similarity.

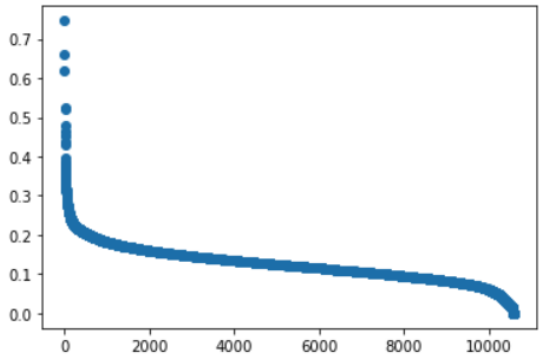

Figure 2: Tanimoto Similarity from SMILEs of all small molecules in this competition. X-axis is all possible pairs of SMILEs of small molecules sorted by value of Tanimoto Similarity in Y-axis.

In NK cells, the correlation of Tanimoto Similarity with both Euclidean or Pearson (Figure A, B) distance of two vectors of differential gene expression affected by those same pair of molecules is very low at -0.017 and 0.041 respectively).

A B

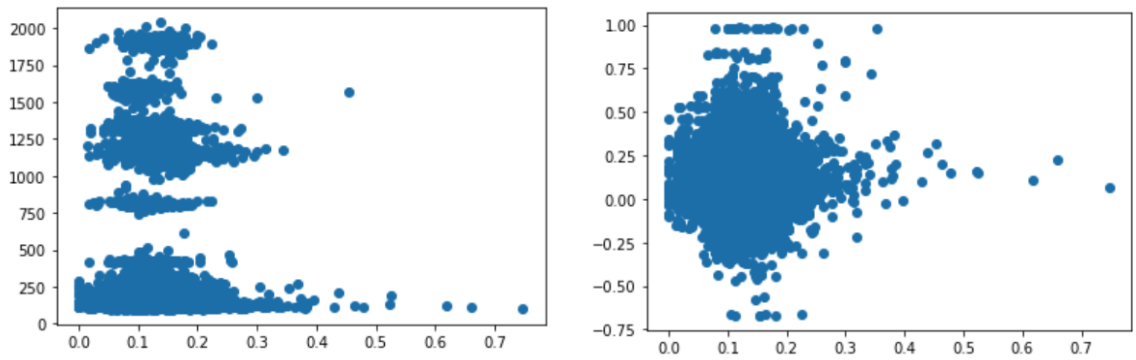

Figure 3: Scatter plot between Tanimoto Similarity (X-axis) and Euclidean (A) or Pearson (B) distance of two vectors of differential gene expression affected by those same pair of molecules

Although the overall trend is discouraging, the pairs with Tanimoto similarity above 0.5 also have low Euclidean distance. The molecule with similarity higher than certain level might be a good predictor to improve differential gene expression prediction.

Unfortunately, all the molecules (except Foretinib) that shared at least 0.5 Tanimoto similarity with other molecules in this competition are all in private LB test sets, so I could not observe if this information can be used to improve the estimation of differential expression on the public leaderboard. I decided to sacrifice one of my final submissions to test this hypothesis. I experimented by replacing the predicted value of Cabozantinib with the raw value of Foretinib in one of the two final submissions. It turns out that this idea leads to lower score in private leaderboard (0.817 vs 0.825)

1.3 Model of gene expression

1.3.1 Conflicting expression

There are several ways that gene expression could be altered by the small molecule. With minor exceptions, all the cell lines have the same genetic sequence, so their differential responses to small molecules are due to gene regulation. At the simplest regulatory form, the response of gene expression depends on whether chromatin is close or open. With that, the majority of cell lines should either have insignificant response to small molecules due to chromatin being close or have differential response all in the same way.

To explore this idea, I checked which small molecules that could have a cell significantly differential express but in opposite direction among different cell types in this study. I used the cut-off at log fold change value more than 10 in either positive or negative direction which is the round up number for multiple correction with number of genes, small molecule, and cell type. This cutoff would filter out most data and keep only 0.9527% of all expression values.

```
[37]: correction_factor=len(all_gene)*len(sm_name_label)*len(cell_type_label)
      correction_factor

[37]: 15952836

[38]: math.log10(0.05/correction_factor)

[38]: -8.503867896202365
```

Among small molecules in the training dataset, I identify 5 small molecules that have conflicting differential expression among the five cell types.

```
]: print(extreme_de_train.loc[extreme_de_train['sm_name']=='Belinostat'].cell_type)
   tmp=extreme_de_train.loc[extreme_de_train['sm_name']=='Belinostat']
   tmp.loc[:, (tmp == 1).any(axis=0)].loc[:, (tmp == -1).any(axis=0)]

275              B cells
276        Myeloid cells
277            NK cells
278          T cells CD4+
279          T cells CD8+
280    T regulatory cells
Name: cell_type, dtype: object
```

]:

	A2M	A2M-AS1	AATBC	ABCB4	ABCC3	AC004585.1	AC005300.1	AC005842.1	AC006077.2	AC006978.1	...	ZNF618	ZNF678	ZNF683	ZNF689	ZNF710	ZNF784	ZNF804A	ZNF
275	0	0	0	-1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
276	-1	0	-1	-1	-1	0	-1	0	-1	-1	...	-1	-1	0	0	-1	-1	-1	
277	0	-1	0	0	0	0	0	0	0	0	...	0	0	-1	0	0	0	1	
278	0	-1	0	0	0	-1	0	-1	0	0	...	0	-1	0	-1	0	0	0	
279	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
280	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	

6 rows × 1043 columns

[261]:
```
print(extreme_de_train.loc[extreme_de_train['sm_name']=='Dabrafenib'].cell_type)
tmp=extreme_de_train.loc[extreme_de_train['sm_name']=='Dabrafenib']
tmp.loc[:, (tmp == 1).any(axis=0)].loc[:, (tmp == -1).any(axis=0)]
```

```
104              B cells
105        Myeloid cells
106             NK cells
107          T cells CD4+
108          T cells CD8+
109    T regulatory cells
Name: cell_type, dtype: object
```

[261]:

	C1ORF162	PLP2
104	-1	0
105	1	1
106	0	0
107	0	-1
108	0	0
109	0	0

4]:
```
print(extreme_de_train.loc[extreme_de_train['sm_name']=='MLN 2238'].cell_type)
tmp=extreme_de_train.loc[extreme_de_train['sm_name']=='MLN 2238']
tmp.loc[:, (tmp == 1).any(axis=0)].loc[:, (tmp == -1).any(axis=0)]
```

```
371              B cells
372        Myeloid cells
373             NK cells
374          T cells CD4+
375          T cells CD8+
376    T regulatory cells
Name: cell_type, dtype: object
```

4]:

	ABCD2	ACTN1	CCR9	FCRL6	OSCAR	SATB1-AS1	SCML4	XCL2
371	0	1	1	1	1	0	0	1
372	1	0	1	1	-1	1	1	0
373	0	0	1	0	0	0	0	0
374	0	-1	0	0	0	-1	-1	0
375	-1	0	-1	-1	0	0	0	-1
376	0	0	0	0	0	0	0	0

262]:
```
print(extreme_de_train.loc[extreme_de_train['sm_name']=='Alvocidib'].cell_type)
tmp=extreme_de_train.loc[extreme_de_train['sm_name']=='Alvocidib']
tmp.loc[:, (tmp == 1).any(axis=0)].loc[:, (tmp == -1).any(axis=0)]
```

```
141              B cells
142        Myeloid cells
143             NK cells
144          T cells CD4+
145    T regulatory cells
Name: cell_type, dtype: object
```

262]:

	FOXP3	HHEX	LINC02397	MALAT1	SPIB	TCL1A	TLR10
141	1	-1	-1	0	-1	-1	-1
142	0	0	0	0	0	0	0
143	1	0	1	0	1	1	1
144	0	1	1	-1	1	0	1
145	-1	0	0	1	0	0	0

```
print(extreme_de_train.loc[extreme_de_train['sm_name']=='Oprozomib (ONX 0912)'].cell_type)
tmp=extreme_de_train.loc[extreme_de_train['sm_name']=='Oprozomib (ONX 0912)']
tmp.loc[:, (tmp == 1).any(axis=0)].loc[:, (tmp == -1).any(axis=0)]
```

```
535              B cells
536        Myeloid cells
537            NK cells
538         T cells CD4+
539    T regulatory cells
Name: cell_type, dtype: object
```

	AC005842.1	EIF3H	FOXP3	PCSK1N	RBM25	SAMD3	SOD1	TCEA3	ZAP70
535	1	-1	1	1	0	1	-1	1	1
536	0	0	0	0	-1	0	1	0	0
537	0	0	1	0	0	0	0	0	0
538	-1	1	0	-1	1	-1	1	-1	-1
539	0	0	-1	0	0	0	1	0	0

These conflicting differential expressions would be concerning for the model that predicts the differential expression of B cells or Myeloid cells solely based on the expression profile of other cells because the “sign” could be reversed and lead to high penalty in MSE score. In particular, most of these conflicting differentials involve B cells or Myeloid cells.

However, upon inspection, the Belinostat and Dabrafenib are positive controls in this study. As for the Alvocidib, MLN 2238, and Oprozomib (ONX 0912), our community member Antonina Dolgorukova has shared that those three small molecule has very low cell count (https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/discussion/445883) which raises the concern about the reliability of their log p-value.

Due to lacking of strong evidence for conflicting expression between B cells and Myeloid cells from the rest of the cell, I suspect that a large fraction of conflicting expression info from Alvocidib, MLN 2238, and Oprozomib (ONX 0912) are noise rather than real signal. I experiment by removing the three small molecules and also two positive controls. Both experiments lead to higher scores in the public leaderboard.

1.3.2 Gene by gene model

Another idea that I tried is to create one model per each gene with the idea that different genes might have different relative responses to other cell types. However, this idea led to a lower MSE, so I did not pursue in this direction.

2. Exploration of the problem

In the “1. Integration of Biological Knowledge”, I learn the following:

- B cell differential expression is easier to predict than that of Myeloid. (See “Blood cells differentiation”)
- At the high Tanimoto Similarity the differential expression, at least in NK cells, are similar (See “Correlation between chemical structure and expression pattern”)
- Except for positive control, the conflicting differential expression are found in samples with low cell count(See “Model of gene expression: Conflicting expression”)

3. Model design

In my opinion, this competition has a small dataset. I want to minimize overfit and make the model more generalizable to future cases.
The main philosophies of model design for this project are to
1) use hypothesis driven. Unless there is a good reason to add particular feature or subdivide the model, I would not include it.
2) use simple model with good explainability

In the final design, I create two models: one for B cell and one for Myeloid cell. Each of them uses the following features and models underneath the ensemble model.

3.1 Feature selection

I experiment with multiple features using additive selection strategy. The final model include:
Differential expression of NK cells,
Differential expression of CD4+ T cells,
Differential expression of CD8+ cells,
Differential expression of regulatory T cells,
Average expression of that gene in target cells.

Other features that were included, but do not improve the performance include variance of differential expression level and the differential expression of cell type that have highest correlation for that particular gene.

3.2 Model selection

I use simple voting to create the ensemble model and use additive model selection strategy. Only models with better performance in the validation dataset will be submitted to check the score. The final model contains elastic net regression and extra tree regression. The shallow neural network has almost no impact on the final model, but the large network increase the MSE in validation, so I keep the neural network shallow.

```
m1 = LinearRegression()
m1_1 = Lasso()
m1_2 = Ridge()
m1_3 = ElasticNet()
#m1_4 = KernelRidge()
m1_5 = HuberRegressor()
m1_6 = PassiveAggressiveRegressor()
m2_0 = DecisionTreeRegressor()
m2 = RandomForestRegressor(n_estimators=10, max_depth=10, random_state=1)
m2_3 = ExtraTreesRegressor(n_estimators=10, max_depth=10, random_state=1)
m2_4 = GradientBoosting('mse')
m3 = KNeighborsRegressor(n_neighbors=1)
#m4= SVR()
m5=BayesianRidge()
m6=MLPRegressor(hidden_layer_sizes=(8,2),random_state=1, max_iter=500)


model=VotingRegressor([('lr', m1_3), ('rfr', m2_3), ('knr=1', m3), ('nn', m6)])
```

Though this simple model would have high bias, it should have reasonable variance and give us good insight into which feature has a high impact on the model and how.

In retrospect, the equal contribution of each model and additive model selection are too simplistic. Some fine tune with different weights among them would most likely give a much better predictive power.

In addition, another part that I did not handle well is to handle the extreme value. Though in traditional data analysis, people would remove them. The scoring scheme of this competition put a lot more weight in extreme value. This decision actually makes sense from a biological perspective where the genes that are significantly differentially should warrant more attention than aggregation of a large number of genes that have insignificant expression change. I experiment on using models that consider “sign” and “magnitude” of signed log-value separately and also models that consider insignificant and significant differential

expressions. However, both cases lead to lower performance in the public leaderboard, so I did not further pursue in these directions.

4. Robustness

I did not perform a systematic evaluation of robustness. However, overall, the model is quite robust. This is most likely due to relatively simple model design (See 3. Model design) and decision on validation selection which I separate data into validation sets by dividing training data based on small molecules. I found that this strategy is robust and I got consistent relative performance of the models with the public leaderboard. See example:

```
#####  set 1 include all data; train 1 out of 3 fluctuate drug
set_training_sm=['Idelalisib', 'Crizotinib', 'Linagliptin', 'Palbociclib',
       'Alvocidib', 'LDN 193189', 'R428',
       'Penfluridol', 'Dactolisib', 'O-Demethylated Adapalene',
       'Oprozomib (ONX 0912)', 'CHIR-99021']
set_val_sm=[
       'Dabrafenib',
       'Porcn Inhibitor III', 'Belinostat', 'Foretinib', 'MLN 2238',]
```

```
#####  set 2 include all except Oprozombi and MLN; train 1 out of 3 fluctuate drug
set_training_sm=['Idelalisib', 'Crizotinib', 'Linagliptin', 'Palbociclib',
       'Alvocidib', 'LDN 193189', 'R428',
       'Penfluridol', 'Dactolisib', 'O-Demethylated Adapalene',
       'CHIR-99021']
set_val_sm=[
       'Dabrafenib',
       'Porcn Inhibitor III', 'Belinostat', 'Foretinib']
```

In addition, from the observation, the data from small molecules that are included in the training set has a strong impact on the final score. For example, removing data from small molecules with low cell count and positive control end up having higher scores in the public leaderboard. See “Model of gene expression: Conflicting expression”

5. Documentation & code style

5.1 Code

The Jupyter notebook distributed in this repo followed the PEP8 standard using the Black code style library <https://black.readthedocs.io/en/stable/the_black_code_style/current_style.html> version 23.11.0. The installation was done by `pip install "black[jupyter]" `.

The code sections are separated by Markdown subsection.

5.2 Requirement

Hardware requirement: The computing was done locally using Macbook pro 2.6 GHz 6-Core Intel Core i7.

Software requirement: The script for this Compeition is available at https://github.com/Arkarachai/Kaggle_OpenProblems_SingleCellPerturbations_YellowAvocado/blob/main/yellow_avocado.ipynb See section `6. Reproducibility` for list of dependency, public Docker, and Dockerfile.

6. Reproducibility

For reproducibility, the code for this competition is available in GitHub https://github.com/Arkarachai/Kaggle_OpenProblems_SingleCellPerturbations_YellowAvocado under MIT license.

The dependency is listed in https://github.com/Arkarachai/Kaggle_OpenProblems_SingleCellPerturbations_YellowAvocado/blob/main/requirements-dev.txt

The Dockerfile is available at https://github.com/Arkarachai/Kaggle_OpenProblems_SingleCellPerturbations_YellowAvocado/blob/main/Dockerfile

To build Docker images, first install Docker on a local computer. Then from path that contain our Dockerfile, run

```
docker build -t yellowavocado .
```

Follow by

```
docker run -p 8888:8888 -it --rm yellowavocado
```

After that you can put this in your primary browser and run notebook

```
http://localhost:8888/lab
```

The yellowavocado docker image is also available at https://quay.io/repository/chai/yellowavocado.