02715 Final report

Xinling Li, Zhenyu Yang

Part I: MAGIC retrieval followed by consensus clustering of single-cell RNA-seq data

Introduction

Single-cell RNA-seq provides sequence information in individual cell level and it provides crucial information for studying function of cells [1]. There are tag-based single-cell RNA-seq and full-length single-cell RNA-seq. In tag-based single-cell RNA-seq, unique molecular identifier(UMI) is used to remove amplification bias. UMI is random barcodes added to each molecule during reverse transcription and it's usually 4-10bp [2]. In single-cell RNA-seq data, dropout is usually very high. Sometimes, it can be as high as 90%. Some possible reasons are low sequence depth and failure of reverse transcription [3]. Therefore, imputation on the raw data is needed before downstream analysis. In this project, we analyze perturb-seq data using MAGIC for imputation and SC3 for clustering cells. Perturb-seq is a technique which combines single-cell RNA-seq and CRISPR. The technique can be used to address a variety of biological questions. For example, it can be used to study biological function of transcription factors in immune response, as shown in the paper written by *Dixit et al*. Our project consists of two parts. In the first part, the questions that we want to answer are the following. First of all, we want to investigate cells that have similar gene expression. Second of all, we want to investigate the significance of imputation by comparing the clustering result using data before and after imputation.

Methods

The detail of dataset that we use is shown below.

Cell type	sgRNA pool	Total cells	Total genes	Time points
Human K562	Cell cycle regulators (32 guides)	25,971	22,783	7 days

There are three input files. The first input file called genenames.txt contains gene_id and its corresponding name. The second input file called cellnames.txt contains cell_id and its corresponding name. The third input file called ccyclegenenames.txt contains gene_id, cell_id, and corresponding UMI counts.

Step 1: Filter cells

We first reduce the number of cells from 25,971 to 1,770 by filtering out the cells in which less than 5,000 genes have non-zero UMI counts. We tried different threshold and we find that 5,000 give reasonable amount of genes and the size of the filtered matrix is not too big for the downstream analysis. The distribution of gene expression across all the cells is shown in the figure below. The x-axis is the interval of number of genes expressed in the cell. The y-axis is the number of cells. Red line is the cutoff that we picked. The majority of cells have fewer than 5000 unique genes expressed.

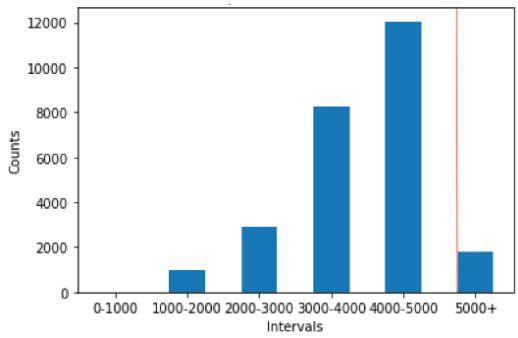


Figure 1. Distribution of total number of genes expressed in all the cells

Step 2: Normalization and square root transformation

Next, L1 normalization followed by square root transformation were performed over the whole dataset. The dropout rate before MAGIC is 0.765.

Step 3: Imputation

Then imputation was performed using MAGIC with default parameter (knn=10, decay=15, and t='auto').

Step 4: Filter genes

Then genes which have low variance of expression across cells are filtered, which was accomplished by two steps. First, all the genes are ranked by their variance. Next, the top 1,000 genes were chosen. 1,000 was selected as the threshold because it's neither too large for the downstream analysis nor too small for significant result. The distribution of variance of gene expression across cells are shown below. The x-axis is the cumulative number of genes and y-axis is the variance of gene expression across cells. Red line is the cutoff that we picked.

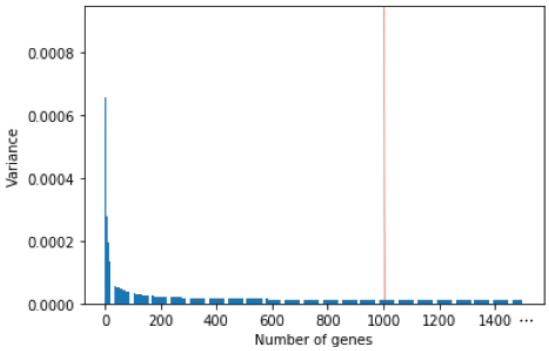


Figure 2. Distribution of variance of gene expression across cells

Step 5: Clustering cells

Consensus clustering of cells was performed using SC3 with ks=2 to 7 and k=3.

Results

To examine the significance of imputation, 3 genes were chosen randomly and their gene-gene relationship was shown in figure 3. These three genes are ENSG00000213934_HBG1, ENSG00000196565_HBG2, and ENSG00000173727_AP000769.1. As we can see from the figure on the left, there's no clear pattern among the three genes because of dropout. After magic, we can see clear gene-gene relationships. This match the biological progression we expect to see- both HBG1 and HBG2 are gamma globin genes and AP000769.1 is clone-based ensembl gene.

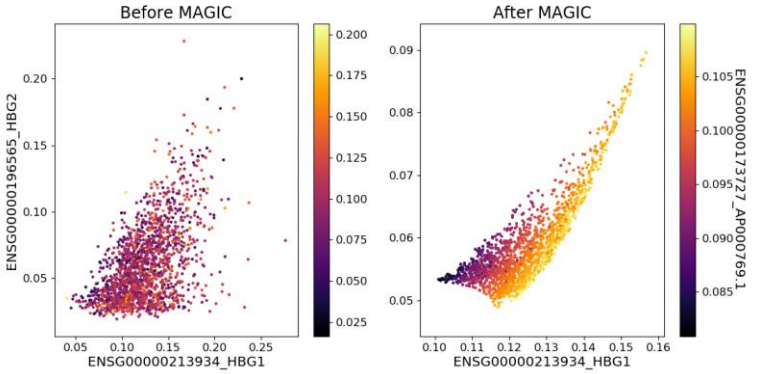


Figure 3. Gene-gene relationship before and after MAGIC

Also, we visualize the cell trajectories with PCA with and without MAGIC in 2D as shown in figure 4. Without MAGIC, there's no clear pattern in the first two component because of dropout. WIth MAGIC, the pattern is much more clear.

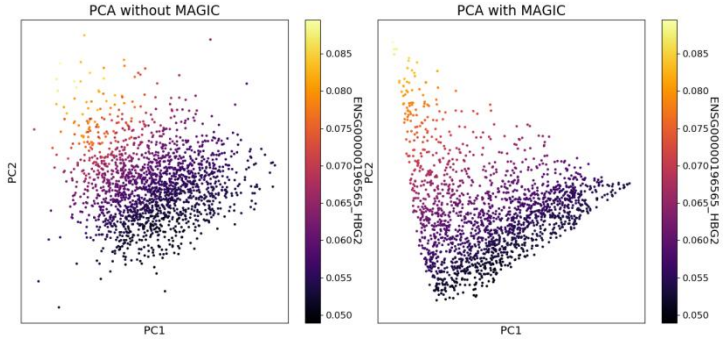


Figure 4. Cell trajectories with PCA on MAGIC in 2D

Moreover, we visualize the cell trajectories with PCA with and without MAGIC in 3D as shown in figure 5. Similar to figure 4, without MAGIC, there's no clear pattern in the first two component because of dropout. WIth MAGIC, the pattern is much more clear.

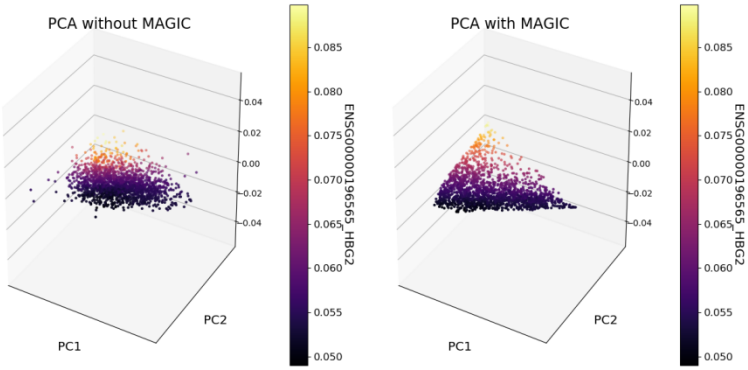


Figure 5. Cell trajectories with PCA on MAGIC in 3D

We then apply data before/after MAGIC to SC3 algorithm. In figure 6, we compared consensus matrix before and after MAGIC imputation. Consensus matrix means how often each pair of cells is located in the same cluster[4]. A node in the matrix has a value from 0.0(blue)-1.0(red), which stands for how often the pair of data points show up in the same cluster. We want the clustering to be more stable so that the more 'red' area in the matrix, the more accurate our imputation is. As you can see, the 'red' area after MAGIC is significant larger than that before MAGIC.

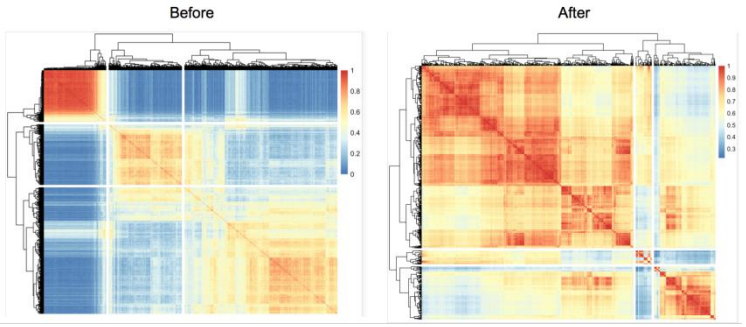


Figure 6: Consensus Matrix Before/After MAGIC imputation

We also measure the stability of the clustering. Stability stands for number of solution where most frequent solution was found by running for 100 times[4]. The physical meaning is generally the same as consensus matrix, but it can give quantitative measurement for individual clusters. After applying MAGIC, cluster 1 and 3 increased their stability for almost 80%.

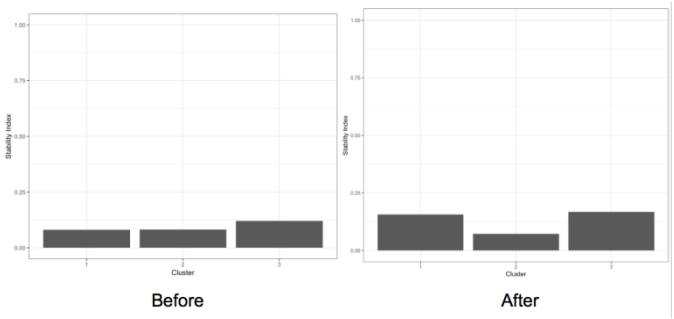


Figure 7: Stability comparison

Discussion

MAGIC restores gene-gene interaction and cell trajectories. Also, it's time efficient. It only takes several seconds to do the imputation and plot the result. It also improves cluster result and cluster stability.

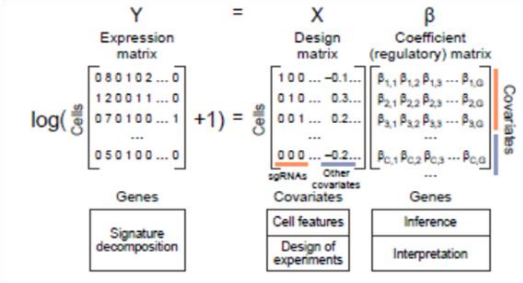
SC3 is easy to install and the manual is easy to understand. However, SC3 is very time consuming. In the paper, the authors claims that SC3 can run 2,000 cells in 20 mins. In our implementation, it takes about 4 hours to do the clustering in 1,770 cells. Also, it's not space efficient, our kernel died for several times.The time it takes to run SC3 on data after imputation is shorter than that on data before imputation.

Several suggestions for SC3 algorithm. When calculating the distance matrix, it is not necessary to use Spearman, Pearson, Euclidean distances. I would pick just Euclidean distance to save ⅔ running time. To select the best k, they have to run all the 'k's in a range which is very time consuming. I would suggest to use CH[5] index to find the suitable k which is time efficient.

Part II: Perturbation analysis

1. Introduction

Linear regression is a common way to find association between perturbation and gene expression. In perturb-seq paper, they used formula below [1]. Y is one of the inputs to the linear regression and it's the expression matrix where each row represents a cell and each column represents a gene. X is another input to the linear regression and each row represents a cell and each column represents a covariate (perturbation, cell cluster assignment, library size, etc). The matrix is created using one hot encoding. In other words, if the perturbation is performed on the cell, the index is 1. Otherwise, it's 0. β is coefficient matrix where each row represents a covariate and each column represents a gene.



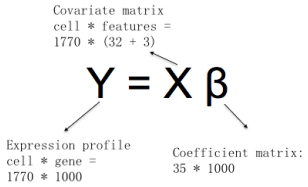
There are different types of linear regression called linear regression with no norm, lasso regression, ridge regression, and elastic-net regression. For lasso and ridge regression, different penalization is added, introducing bias into the estimation of β in order to reduce the variability of the estimate [2]. Elaticnet regression is a hybrid of lasso regression and ridge regression by taking into account both l1 and l2 normalization [3]. In part 2 of the project, our goal is to find out genes that have similar expression under different perturbations, perturbations that have similar effects on gene expression levels, and genes whose expression is affected by each perturbation using hierarchical clustering on coefficient matrix.

2. Methods:

Cell type	sgRNA pool	Total cells	Total genes	Time points
Human K562	Cell cycle regulators (32 perturbations)	1,770	1,000	7 days

2.1 Covariate matrix construction

Our first goal is to construct an input matrix X of cell * features as the covariate matrix. X consists of 32 perturbations and 3 cell cluster assignment as the SC3 algorithm suggested in part 1. So the dimension of the coefficient matrix should be 35 * 1000. We want to investigate whether it's necessary to include cell state in our covariate matrix.



2.2 Regularization selection

Our second goal is try to select a linear model from No regularization, L1, L2 regularization[4] to identify the genes that are affected by each perturbation. We will use mean square error, and Pearson's correlation to evaluate each type.

	No regularization	L1 regularization	L2 regularization
Equation	$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2$	$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\lvert\beta_j\rvert$	$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2$

2.3 Polynomial regression

After selecting the best normalization, we chose the one with the best performance and try polynomial regression with different degrees. The degree ***n*** of the polynomial regression is defined as:

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \beta_2 \boldsymbol{x^2} + \beta_3 \boldsymbol{x^3} + \cdots + \beta_n \boldsymbol{x^n} + \varepsilon$$

2.4 Significant, correlation test & clustering

After getting coefficient, we sort each column based on the absolute value to get the genes that are affected by each perturbation. We chose the highest absolute value in the rows of coefficient matrix ***β*** as the most perturbed gene. We want to find out which gene had been perturbed and how does perturb change the variance of the expression profile ***Y***.

Then we did clustering analysis on the genes and perturbations. Choosing the number of clusters is an art rather

than science. However, we found a very interest method CH index to determine the best choice of k[5]. The index measures how well your choice of k performs. It is calculated by the ratio of between cluster variance and within cluster variance. We think the best choice of k should be the one that has the highest ch index.

$$\mathrm{CH}(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}$$

Results:

The MSE/pearson correlation of with/without vs no norm/L1 norm/L2 norm is shown in the table below. Adding cell state as a feature of the input matrix X can improve the result significantly by reducing MSE error and increasing the Pearson's correlation. No norm performs slightly better than L1 norm or L2 norm. In further experiments, we will choose no norm and with cell state.

MSE	No norm(train/val)	L1 norm (train/val)	L2 norm (train/val)
Without cell state	0.0329426/0.034536	0.033211/0.034109	0.032944/0.034536
With cell state	**0.016231/0.017396**	0.033294/0.031201	0.016232/0.017406

Pearson Correlation	No norm(train/val)	L1 norm(train/val)	L2 norm(train/val)
Without cell state	0.998293/0.998122	0.998265/0.998169	0.99829/0.998122
With cell state	**0.999583/0.999635**	0.998254/0.998555	0.999583/0.999633

We want to know why 'no norm' performs the best among all, so we plot PC1 vs PC2 of the coefficient matrix $\boldsymbol{\beta}$. Figure 9 shows the cell PCA plot of each X type. The samples in the No norm with cell state are more'spread out' than the others. For L1 norm, all of the coefficient entries are 0. We decrease the value of penalty term to 1e-8 to make the entries to be non-zero. This is almost the same as L0 norm, so that we will not consider L1 norm. We think that it might be the L0 norm can recover more coefficient than the other methods.

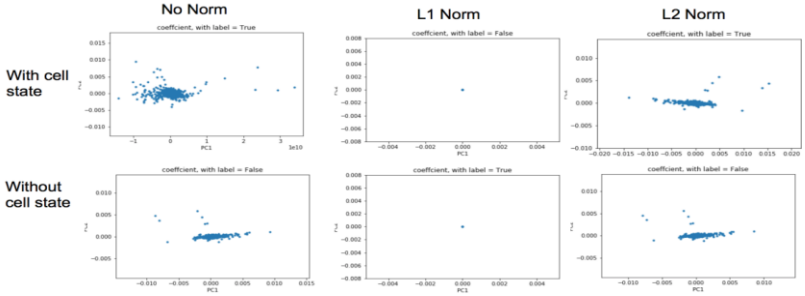


Figure 9. PCA plot of the coefficient matrix

Next, we cluster genes based on the coefficient matrix using hierarchical clustering with ward linkage. The number of clusters is determined using CH index. We found that the optimal number of cluster of genes is 2, as shown in figure 10 (left).

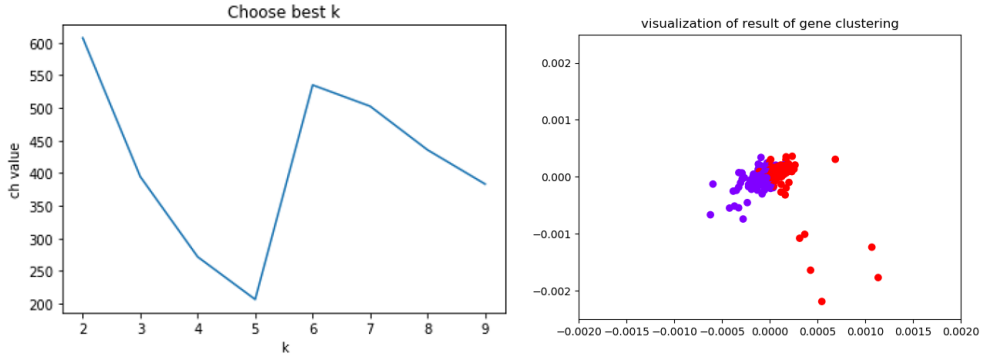


Fig. 10 Ch value for different choices of number of clusters of gene (left) and Visualization of result of gene clustering(right)

The visualization of the two clusters is shown in figure 10 (right). Different color represents different cluster. The two clusters are well separated.

Then we cluster perturbations based on the coefficient matrix using hierarchical clustering with ward linkage. The number of perturbations is determined using CH index. We found that the optimal number of cluster of perturbations is 4, as shown in figure 11 (left).

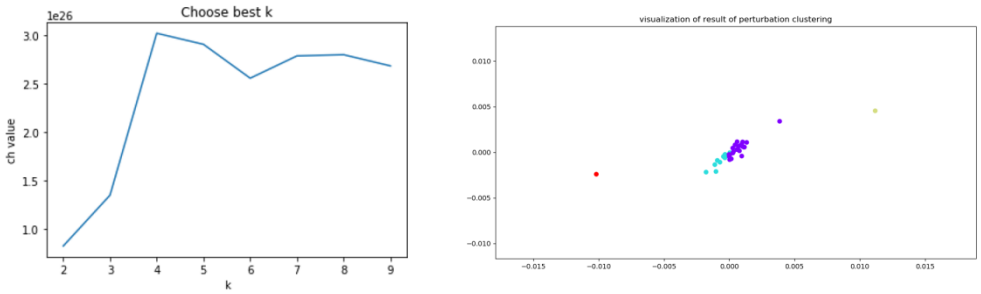


Fig 11. Ch value for different choices of number of clusters of perturbations and visualization of result of perturbation clustering

The visualization of the four clusters is shown in figure 11 (right). Different color represents different cluster. The four clusters are well separated. We also drew a dendrogram to illustrate the result of hierarchical clustering using ward linkage, as shown in figure 12. Different color represents different cluster of perturbations. We found that c_sgECT2_3, c_sgOGG1_2, c_sgPTGER2_4, c_INTERGENIC393453, c_sgAURKB_4, c_sgRACGAP1_9, c_sgCEP55_1, and c_sgARHGEF17_4 belong to one cluster. c_sgRACGAP1_3 belongs to the second cluster. c_sgCABP7_2, c_sgCENPE_4, c_sgPTGER2_3, c_sgAURKB_6, c_sgCENPE_1, and c_sgPTGER2_2 belong to the third cluster. c_sgAURKC_7, c_INTERGENIC345439, c_sgCABP7_1, c_sgOGG1_3, c_sgTOR1AIP1_1, c_sgCABP7_4, c_sgCIT_7, c_INTERGENIC216151, c_sgCIT_1, c_sgOGG1_4, c_sgARHGEF17_1, c_sgCEP55_4, c_sgCENPE_2, c_sgAURKA_3, c_sgECT2_2, and c_sgAURKC_1 belong to the fourth cluster.

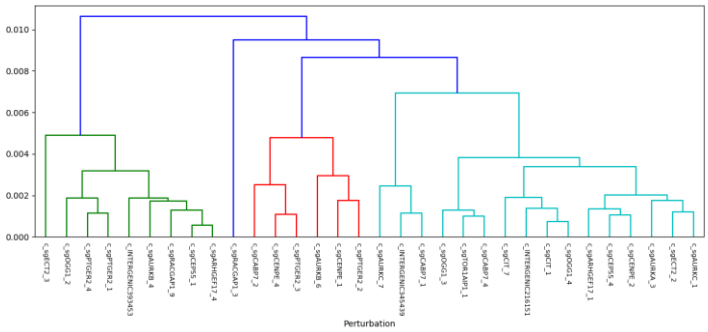


Figure 12. Dendrogram of result of perturbation clustering

We also plotted the distribution of coefficient of 1,000 genes for each perturbation. We found that there are 6 genes that significantly repressed by perturbation c_sgECT2_3 and 6 genes that are significantly activated by perturbation c_sgRACGAP1_3, as shown in the red box in figure 13. Also, the 6 genes that are significantly repressed by perturbation c_sgECT2_3 and activated by perturbation c_sgRACGAP1_3 are the same. They are ENSG00000213934_HBG1, ENSG00000196565_HBG2, ENSG00000198804_MT-CO1, ENSG00000173727_AP000769.1, ENSG00000198899_MT-ATP6, and ENSG00000228253_MT-ATP8. HBG1 and HBG2 genes are hemoglobin subunits that are normally expressed in the fatal liver, spleen, and bone marrow [6]. MT-CO1 gene encodes cytochrome c oxidase which is a component of the respiratory chain that catalyzes the reduction of oxygen to water [7]. AP000769.1 is Finkel-Biskis-Reilly murine sarcoma virus (FBR-MuSV) ubiquitously expressed (FAU) pseudogene [8]. MT-ATP6 encodes a protein that is important for normal mitochondrial function [9]. MT-ATP8 is a mitochondrial gene that encodes a subunit that belongs to transmembrane F-type ATP synthase [10].

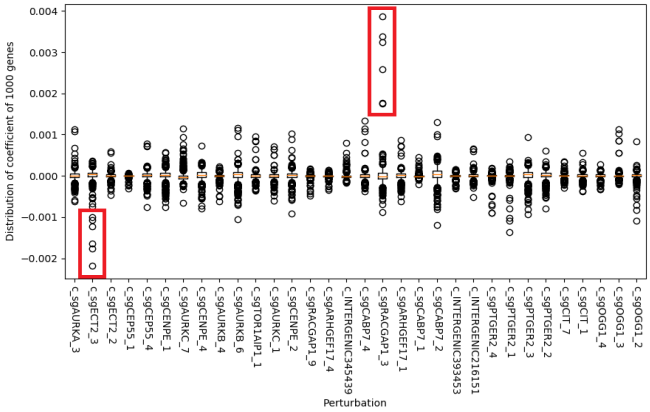


Figure 13. Distribution of coefficient of 1,000 genes for each perturbation

Genes corresponding to the highest absolute value of coefficient for each perturbation are shown in figure 14A.The ratio of each gene occurred as most perturbed gene is shown in figure 14B**.** Surprisingly, there are only 4 genes that had been significantly affected by each perturbation. They are ENSG00000196565_HBG2, ENSG00000213934_HBG1, ENSG00000173727_AP000769.1, and ENSG00000223609_HBD. We rank the variance of gene expression across cells in expression matrix and the top 5 genes are: ENSG00000213934_HBG1, ENSG00000223609_HBD, ENSG00000198804_MT-CO1, ENSG00000196565_HBG2, ENSG00000173727_AP000769.1. We found that the 4 most perturbed genes by the perturbations are the top 5 genes whose variance of expression across the cells are the highest ! This is very exciting result since we confirmed that genes targeted with perturbation will have very high variance and our model can successfully capture this feature.

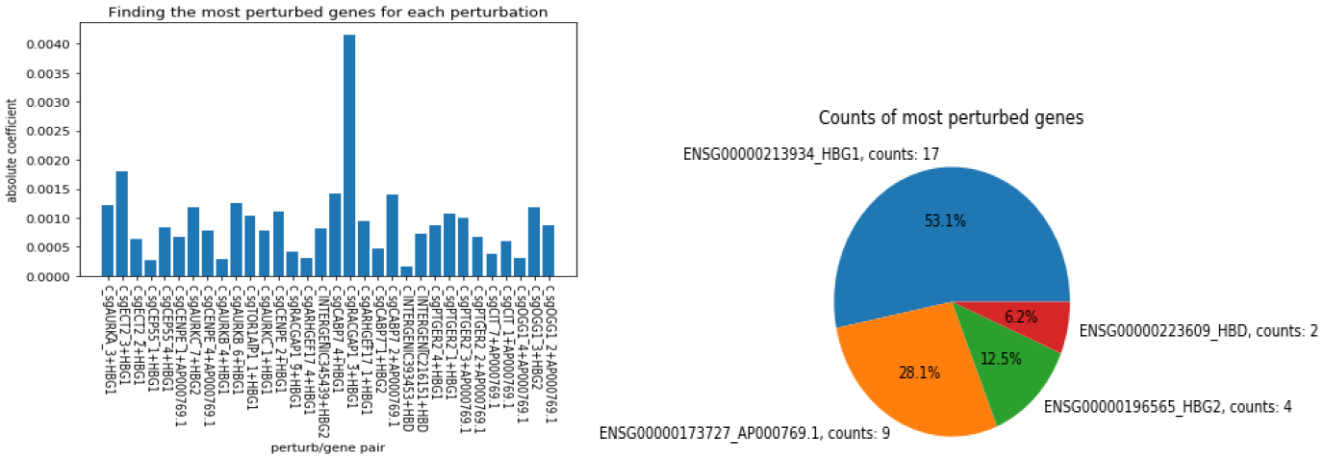


Figure 14 A & B. Genes most affected by each perturbation

Discussion & improvement:

No norm performs better than L1 norm and L2 norm with λ as1 for our dataset. Also, using cell cluster as additional features in design matrix improves the result of linear regression. In addition, we found 3 clusters of cells, 2 clusters of genes, and 4 clusters of perturbations in the original dataset after filtering and imputation. Moreover, the top 6 genes that significantly repressed by perturbation c_sgECT2_3 and 6 genes that are significantly activated by perturbation c_sgRACGAP1_3. We find there are only 4 genes that are mostly affected by 32 perturbations. The 4 genes have very high variance of expression across cells, suggesting that perturbation increases the variance in the expression profile and our model can successfully capture this. The most influenced gene will have the highest absolute value of coefficient. However, there are still problems remained to be solved. We cannot validate how many clusters in the genes or perturbations. We can only intuitively suggest that our CH index method works by PCA plot. Also, we only use mean squared error and Pearson's correlation to find the optimal model to choose the best linear regression model. However, it's possible that the best model doesn't have the lowest mean squared error and Pearson's correlation. Also, using MAGIC to do imputation may decrease the variation in gene expression and as a result influence the result of linear regression. In the future, we want to include library size of each cell to the design matrix and compare the result of including this feature and the one without the feature.

Reference for part 1:

[1] Single-cell RNA-seq from wikipedia
[2] UMI from lecture slide
[3] Dropout from lecture slide
[4] V.Yu Kiselev et al,SC3: consensus clustering of single-cell RNA-seq data. 2017, nature methods.
[5]http://ethen8181.github.io/machine-learning/clustering_old/clustering/clustering.html#content

Reference for part 2:

[1] Dixit et al, Pertub-seq: Dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetics screens. 2016, Cell.
[2] linear regression from wikipedia https://en.wikipedia.org/wiki/Linear_regression
[3] https://towardsdatascience.com/5-types-of-regression-and-their-properties-c5e1fa12d55e
[4] https://medium.com/datadriveninvestor/l1-l2-regularization-7f1b4fe948f2
[5] Description of CH index
http://ethen8181.github.io/machine-learning/clustering_old/clustering/clustering.html
[6] HBG1 and HBG2 gene description https://www.ncbi.nlm.nih.gov/gene/3048
[7] MT-CO1 gene description https://ghr.nlm.nih.gov/gene/MT-CO1
[8] AP000769.1 gene description
http://useast.ensembl.org/Homo_sapiens/Gene/Summary?g=ENSG00000173727;r=11:65455258-65466720
[9] MT-ATP6 gene definition on NIHhttps://ghr.nlm.nih.gov/gene/MT-ATP6
[10] MT-ATP8 gene definition from wikipedia https://en.wikipedia.org/wiki/MT-ATP8