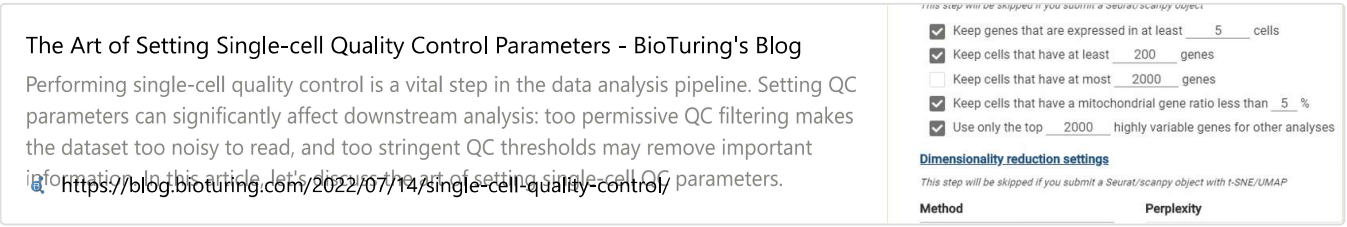
Note

▼ Single-cell Quality Controlling

resource:



Performing single-cell quality control is a vital step in the data analysis pipeline. Setting QC parameters can significantly affect downstream analysis: too permissive QC filtering makes the dataset too noisy to read, and too stringent QC thresholds may remove important information.

In this article, let's discuss the art of setting single-cell QC parameters. We'll explain some important QC parameters, the rationale behind some thresholds commonly used by the community, and pitfalls that you should watch out for when setting single-cell quality control parameters.

Understand Important Single-cell Quality Control parameters

The main goal of single-cell quality control is to filter out low-quality cells, such as dead cells, cells with technical issues during sequencing (e.g multiplet, broken cells, empty droplets, etc.), cells with too little information/value provided (e.g low reads, low number of genes expressed, etc.).

To judge whether a cell is of bad quality, quality control often looks at the three following parameters:

1. The number of counts per barcode, also known as the "count depth": this parameter corresponds to how many transcripts were sequenced in a cell. A low count depth may indicate poor sequencing or dead cells, while an abnormally high count may come from doublets. Doublets (or multiplets) arise in scRNA-seq data when two (or more) cells are mistakenly considered as a single cell.
2. The number of genes per barcode: similar to the count depth, this parameter also indicates whether a cell is poorly sequenced, dead, or doublets.
3. The number of mitochondrial genes per barcode: this parameter is often converted to the proportion of transcripts mapped to mitochondrial genes or 'mitochondrial fraction". A high mitochondrial fraction is an indicator of apoptotic cells or cells with broken membranes during sequencing (if cells are broken, cytoplasmic mRNAs get leaked out and only mitochondrial mRNAs are sequenced)

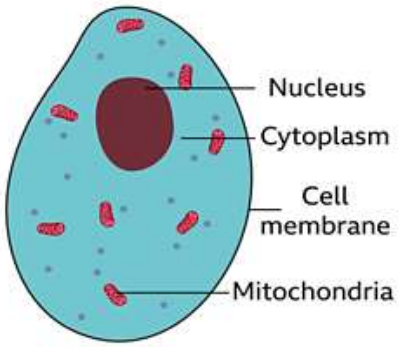
Despite no agreement on how to set single-cell quality control parameters, a rule of thumb exists:: **to exclude cells with less than 200 genes and more than 5% of mitochondria counts**. This set of thresholds comes from early studies, such as Ilicic et al, (2016), Lukassen et al., (2018) and was recommended by Seurat, thus making it widely popular in the community of single-cell RNA-seq researchers.

There are some additional parameters that help speed up downstream analysis and reduce noise, such as:

- Filter out genes that express in less than X number of cells: such genes are either not informative enough or a sign of dropouts. A common threshold for minimum cells per gene is 5. (One important characteristic of scRNA-seq data that feeds into all these challenges is a phenomenon called "dropout", where **a gene is observed at a low or moderate expression level in one cell but is not detected in another cell of the same cell type**)
- Filter out cells that express less than X number of genes: it's hypothesized that a cell needs at least 200 genes to function properly (Gil et al., 2004). Cells with fewer genes may suffer from damage or poor technical processing and thus provide less valuable information.

▼ Cell Biology/ Single-cell Preprocessing

Cells are the basic building blocks of all living things. The human body is composed of trillions of cells. They provide structure for the body, take in nutrients from food, convert those nutrients into energy, and carry out specialized functions. Cells also contain the body's hereditary material and can make copies of themselves.



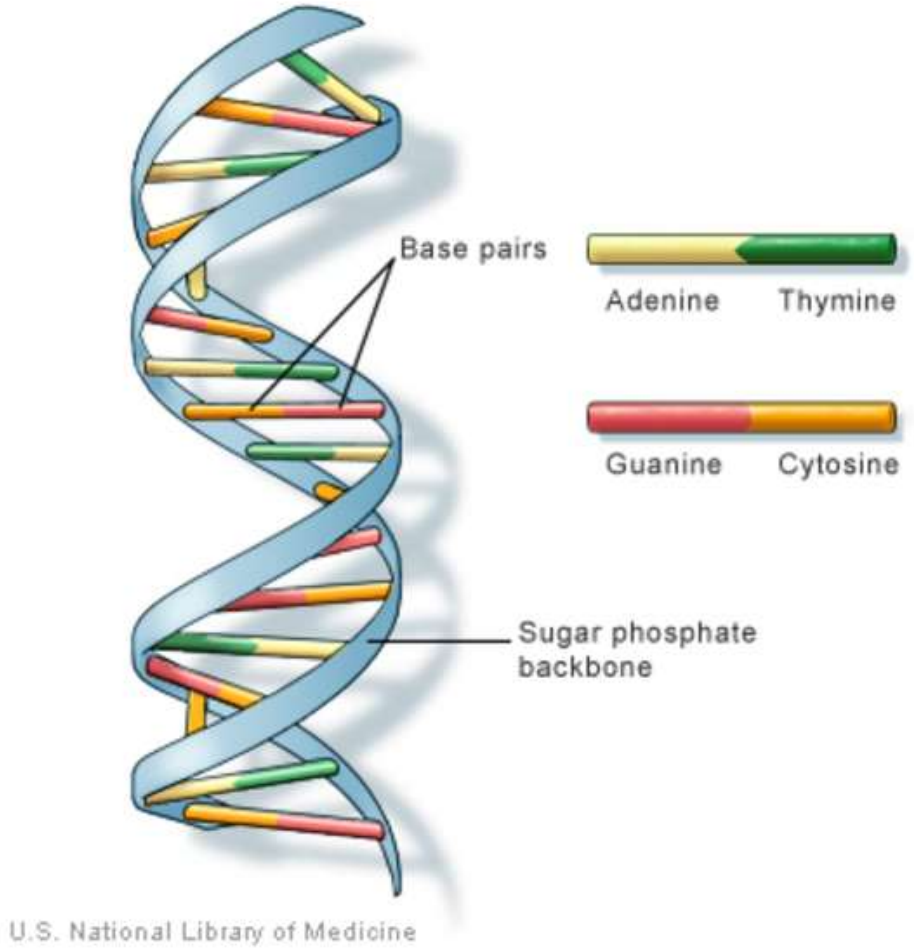
DNA

DNA, or deoxyribonucleic acid, is the hereditary material in humans and almost all other organisms. Nearly every cell in a person's body has the same DNA. Most DNA is located in the cell nucleus (where it is called nuclear DNA), but a small amount of DNA can also be found in the mitochondria (where it is called mitochondrial DNA or mtDNA). Mitochondria are structures within cells that convert the energy from food into a form that cells can use.

The information in DNA is stored as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people. The order, or sequence, of these bases determines the information available for building and maintaining an organism, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences.

DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a nucleotide. Nucleotides are arranged in two long strands that form a spiral called a double helix. The structure of the double helix is somewhat like a ladder, with the base pairs forming the ladder's rungs and the sugar and phosphate molecules forming the vertical sidepieces of the ladder.

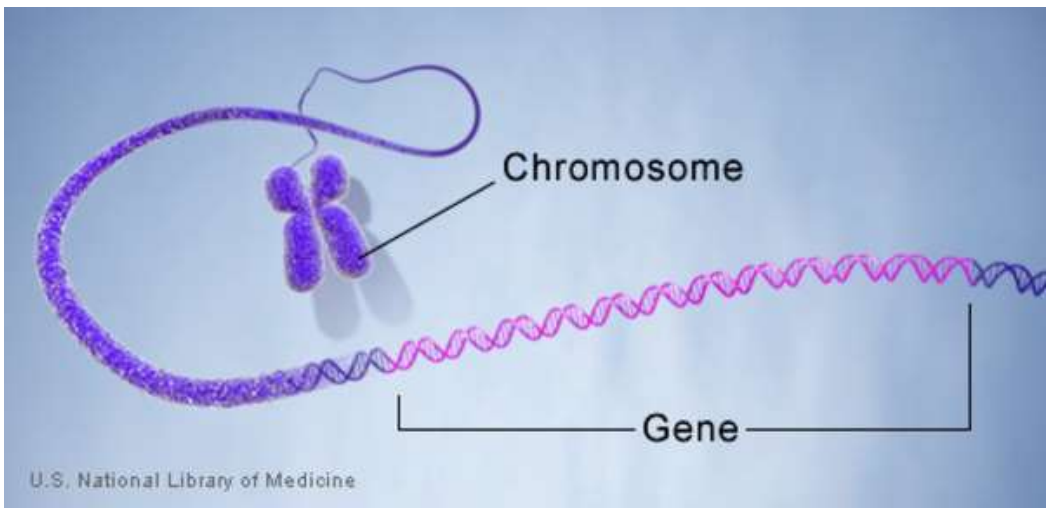
An important property of DNA is that it can replicate, or make copies of itself. Each strand of DNA in the double helix can serve as a pattern for duplicating the sequence of bases. This is critical when cells divide because each new cell needs to have an exact copy of the DNA present in the old cell.



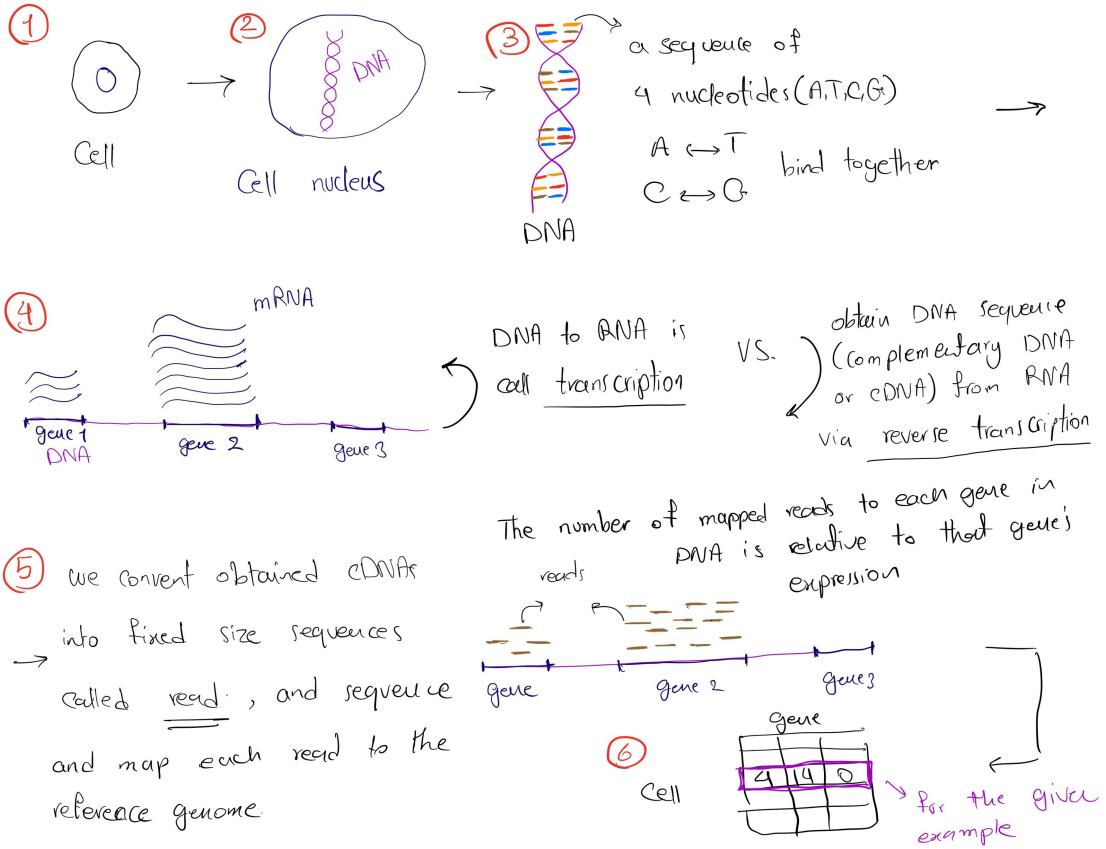
DNA is a double helix formed by base pairs attached to a sugar-phosphate backbone.

Gene

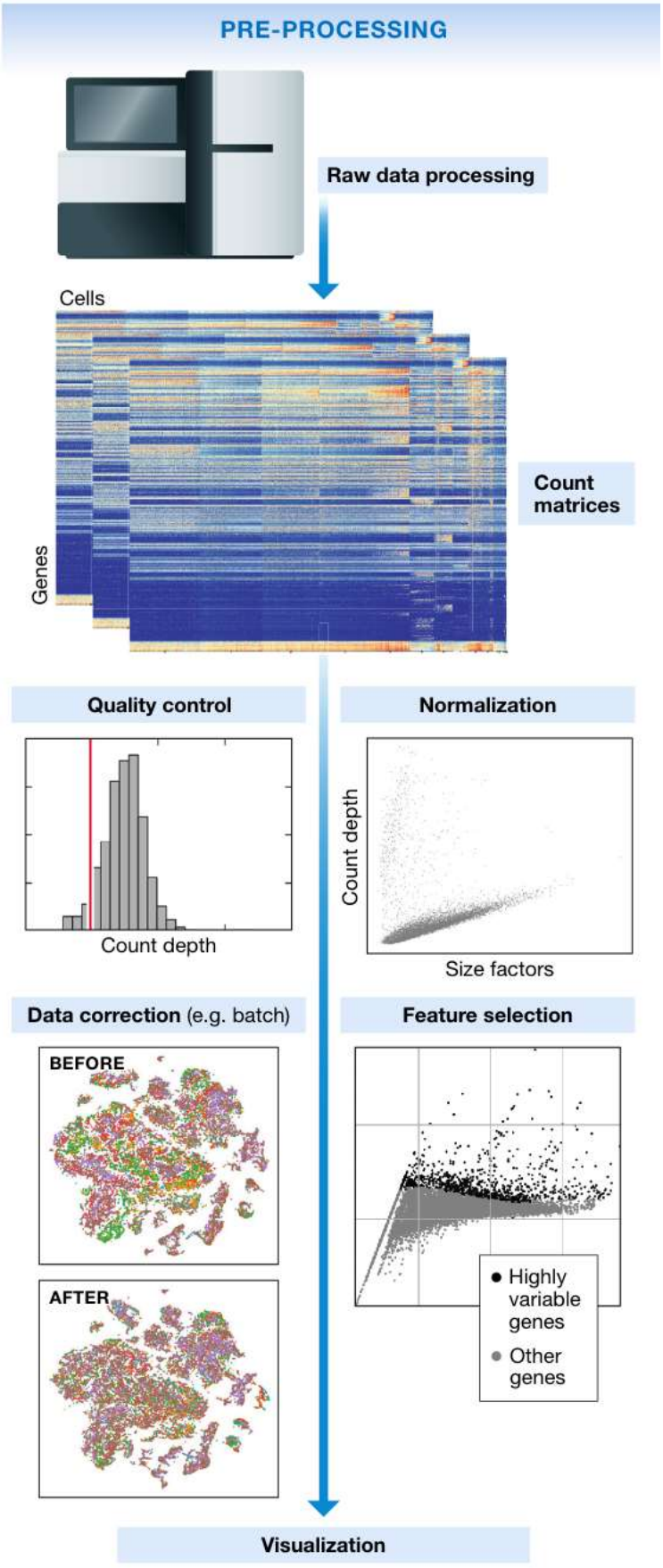
A gene is the basic physical and functional unit of heredity. Genes are made up of DNA. Some genes act as instructions to make molecules called proteins. However, many genes do not code for proteins. In humans, genes vary in size from a few hundred DNA bases to more than 2 million bases. An international research effort called the Human Genome Project, which worked to determine the sequence of the human genome and identify the genes that it contains, estimated that humans have between 20,000 and 25,000 genes.



DNA to Gene expression matrix



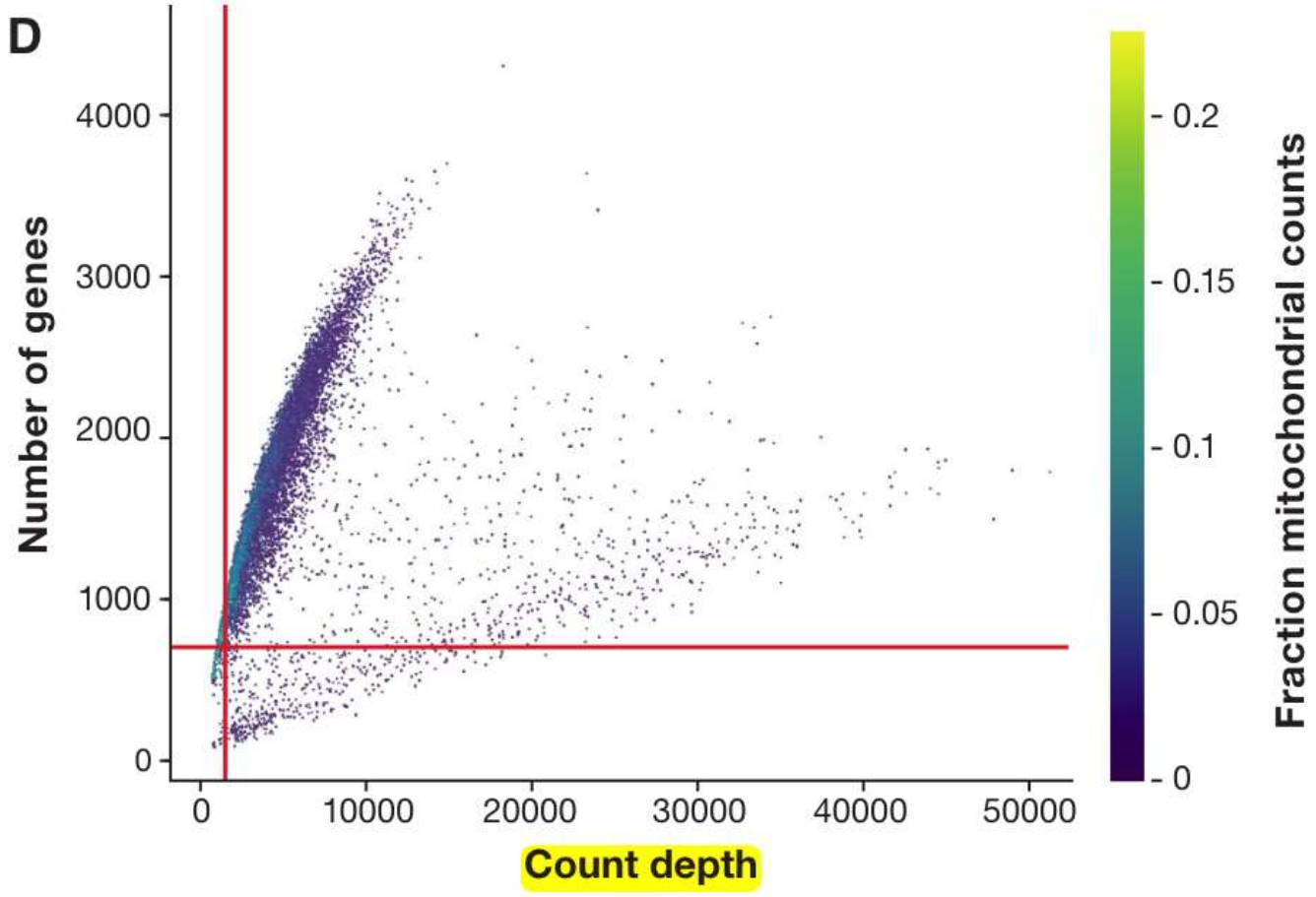
Single-cell RNA-seq Preprocessing



Quality control

Before analysing the single-cell gene expression data, we must ensure that all cellular barcode data correspond to viable cells. Cell QC is commonly performed based on three QC covariates: the number of counts per barcode (count depth), the number of genes per barcode, and the fraction of counts from mitochondrial genes per barcode (RNA can come from nucleus or mitochondria).

Considering any of these three QC covariates in isolation can lead to misinterpretation of cellular signals. Quality control is performed to ensure that the data quality is sufficient for downstream analysis. As "sufficient data quality" cannot be determined a priori, it is judged based on downstream analysis performance (e.g., cluster annotation).



Normalization

Each count in a count matrix represents the successful capture, reverse transcription and sequencing of a molecule of cellular mRNA (Box 1). Count depths for identical cells can differ due to the variability inherent in each of these steps. Thus, when gene expression is compared between cells based on count data, any difference may have arisen solely due to sampling effects. Normalization addresses this issue by e.g. scaling count data to obtain correct relative gene expression abundances between cells.

The most commonly used normalization protocol is count depth scaling, also referred to as "counts per million" or CPM normalization.

There are other approaches as well, like FPKM which stands for Fragments (or reads) per Kilobase of gene, Per Million mapped reads

After normalization, data matrices are typically log(x+1)-transformed. This transformation has three important effects. Firstly, distances between log-transformed expression values represent log fold changes, which are the canonical way to measure changes in expression. Secondly, log transformation mitigates (but does not remove) the mean–variance relationship in single-cell data (Brennecke et al, 2013). Finally, log transformation reduces the skewness of the data to approximate the assumption of many downstream analysis tools that the data are normally distributed.

Data correction and integration

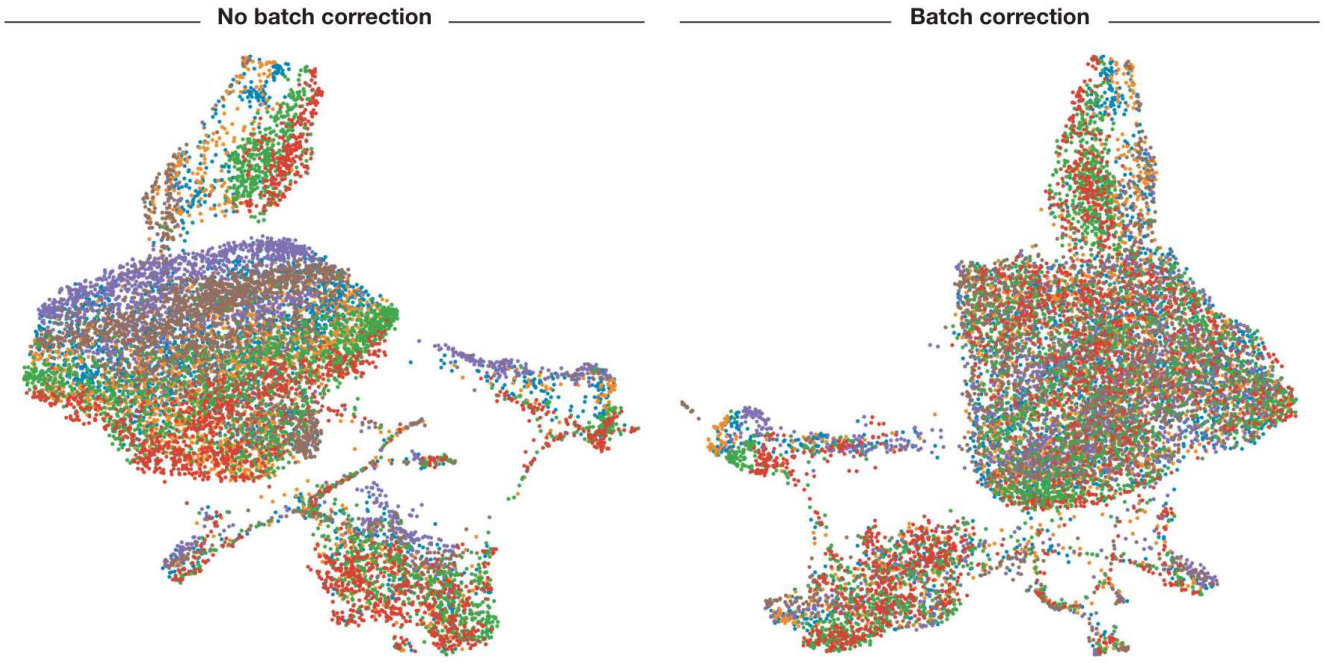


Figure 3. UMAP visualization before and after batch correction.
Cells are coloured by sample of origin. Separation of batches is clearly visible before batch correction and less visible afterwards. Batch correction was performed using ComBat on mouse intestinal epithelium data from Haber *et al* (2017).

Feature selection, dimensionality reduction and visualization

▼ Brain storming About Designing Data Imbalance Experiments

▼

Show the effect of a class imbalance

The simple approach consists of:

1. Start with a balanced dataset, train model and evaluate it with R2
2. Increase the imbalance rate and also compute score R2
3. Keep increasing the imbalance rate step-by-step and evaluating the score each time.

If the imbalance has a significant effect, we should see a decrease of R2 as we add more imbalance.

It is consistent with the idea from the The differential impacts of dataset imbalance in single-cell data integration article:

```
To address this gap, we developed the Iniquitate pipeline for assessing the stability of single-cell RNA sequencing (s
cRNA-seq) integration results after perturbing the degree of imbalance between datasets
```

Weaknesses:

1. Currently our custom VAE model have much worse score compared to scGen. We should improve it.

R2

2. As we add more instances to increase imbalance, we just give the model more data, which can imrove it's performance.
3. Perhaps, we should have selected only specific cell types, not all dataset.

@Pavlo's thoughts about the core of the problem of data imbalance

1. It assigns a false importance weights for the classes.

Imbalance is a problem, because model learns to give some classes more importance than to the others. But we need all cell types to be equally important. For example, if we have 95% of cells of type CD4T and only 10% of type B, we will learn the patterns of a former, but we will give less importance to the latter. This can be fixed by adding resampled data. But the core reason remains: how can we assign a weights of importance to our observarions? How can we evaluate, which cells are more important to take into account, and which are less importtant? One simple idea to start: leave-one-out technique. We hold out one cell type and train a model on others. After that we compare, how the score is changed. If score decreases significantly, this cell type is extremely important, otherwise - it is less important.

1. Also, it may be an issue, that for some classes we simply don't have enough data to catch the pattern of treatment effect.

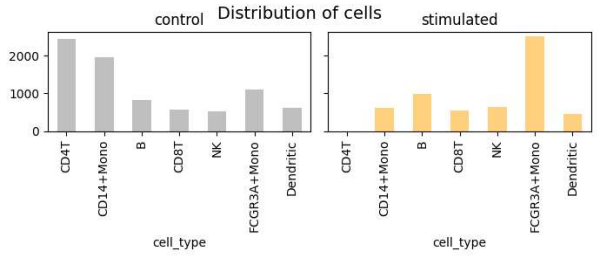
▼ [outdated] Effect of Upsampling on model training

Upsampling is one of the ways of balancing classes in dataset.

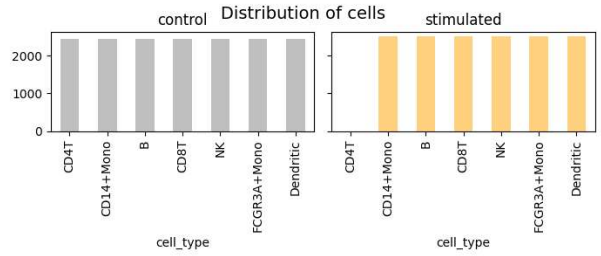
Let:

- *n_class* be a number of instances for a particular class
- *n_max* be the size of the largest class

Upsampling: for each class, get a sample of size n_max - n_class from class data and append this sample to the class data.



Original dataset



After upsampling

Experiment

I constructed four datasets:

1. *Usual dataset.* This is the original dataset, no changes made.
2. *Balance upsampling dataset.* This is a dataset after upsampling.
3. *Balance downsampling*. This is a dataset after downsampling.
4. *Balance to 800 instances dataset*. Classes, that have less than 800 instances, are upsampled to this number; classes, that have more - downsampled.

Experiment: For each dataset, I trained our custom VAE model for single-cell condition prediction and evaluated it with R^2 score.

Results:

	R^2 score
usual dataset	0.705139
balance upsampling	0.812074
balance downsampling	0.598309
balance to 800 instances	0.794079

	R^2 score
original dataset	0.734893
balance upsampling	0.817807
balance downsampling	0.761431
balance to 800 instances	0.720994
balance to 1200 instances	0.732880
balance to 1600 instances	0.760960

We see that:

- Upsampling **increases** predictive accuracy of the model,
- Downsampling does not show the decrease of accuracy. It but can compete with sampling to N instances.
- For downsampled datasets it's important to validate using Cross Validation, because scores are highly variable. The same is true for *balance to 800 instances*. But balance upsampling dataset seems to have stable score, probably because it achieved needed amount of data.

Conclusions:

- We can improve R^2 score by choosing the right balancing schema.
- Seems like at some point not only the proportions of classes are important, but the sample size as well.

Ideas:

- We can come up with a more sophisticated balance schema: to assign proportion to each class & assign number of instances for each class. We used the dataset *balance to 800 instances*. Basically, we assign each class an equal proportion and number of instances (800 in this case). But we can change the proportion and number of instances as we want. Maybe it will help achieve better scores?

Concerns:

- These conclusions need validation by scGen model. We should perform the same experiment for scGen using Vector cluster to speed up training.
- We have used CD4T stimulated cells for evaluation. Maybe we should try other celltypes?

▼ Effect of Diversity on model training

We will test the hypothesis that *diversity of data is important for model training*. A high diversity means that in our dataset we have enough instances of each class (cell type) and there are no large class disbalance.

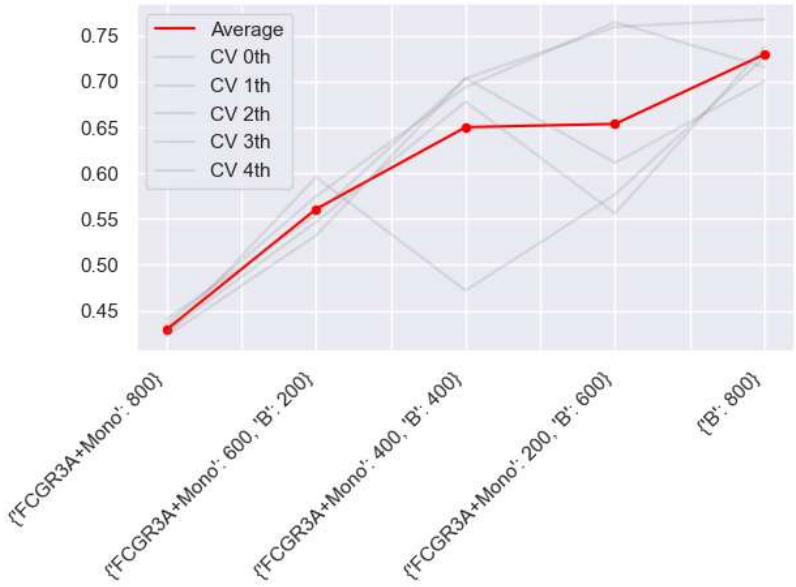
Experiment:

In order to validate this hypothesis we constructed datasets:

1. 800 cells of FCGR3A+Mono - lowest diversity.
2. 600 cells of FCGR3A+Mono & 200 cells of B - intermediate diversity.
3. 400 cells of FCGR3A+Mono & 400 cells of B - complete diversity.
4. 400 cells of FCGR3A+Mono & 400 cells of B - intermediate diversity.
5. B: 800 cells - lowest diversity.

Use our custom VAE model for training. Evaluate using R^2 score.

Results:



We see that:

- no evidence that diversity is important: we have the highest & the lowest scores for cases with only one class.

Concerns:

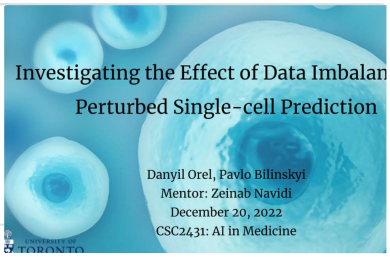
- We should validate on other cell types
- We should validate using scGen model

▼ Presentation

[CSC2431] Investigating the effect of data imbalance on perturbed single-cell prediction

Investigating the Effect of Data Imbalance on Perturbed Single-cell Prediction Danyil Orel, Pavlo Bilinskyi Mentor: Zeinab Navidi December 20, 2022 CSC2431: AI in Medicine Feel free to edit any of the slides (formatting, figure, etc) 1

https://docs.google.com/presentation/d/1OMIgQLfpRPDw8FqKjQ4yriGy5W7BtHyowb_X-Gd36hg/edit#slide=id.g1bc54d205cf_0_71



(slide 1, Cover)

Good morning everyone!

Today I`m going to present to you the results of our research. We have been investigating the Effect of Data Imbalance on Perturbed Single-cell Prediction. Together with Danyil Orel and under mentorship of Zeinab Navidi.

(slide 2)

Let me walk you through agenda.

First-things-first, I will briefly remind you about the topic of personalized treatment, which is a motivation of our research. Also, I will tell you about single-cell data and the methodology that we used. Then we will present the results of our work, the most valuable of our observations and conclusions.

(slide 3)

So, our research is motivated by a topic of personalized medicine.

If we will know, how a patient's cells react to a particular drug, we can recommend medicines, which meet the patient's need more precisely. It leads to a better quality of treatment.

There are a lot of studies now on this application. However, the laboratory experiments are really resourceful and costly, but luckily the single-cell data analysis can help. So let's talk about it.

(slide 4)

The data we are working with is obtained from living tissue samples using the single-cell sequencing.

(slide 5)

For each cell we have an expression of genes from it's DNA. So, the dataset is a count matrix, where unique instance is a cell, and gene expressions are it's features.

We have a data for control cells (cells in usual condition), and also we have *stimulated* cells, which were collected after patient took a drug. So, stimulated cells are in some way affected by drug. We utilize this data for learning how to predict the effect of a treatment on a cell.

(slide 6)

We worked with the dataset, that contains cells from human blood. There are 7 types of cells. And we have both control and stimulated by interferon cells.

(slide 7)

In particular, we were focused on the problem of imbalance in single-cell data. The dataset typically contains cells of different types. But sometimes some types we have much more data than for the other. This situation is called class imbalance, and it can be harmful for our predictive ability.

(slide 8)

Data imbalance is an issue in many studies and have been investigated in different areas. Here is a recent paper studying the impacts of dataset imbalance in single-cell integration. However, this factor has not been investigated in perturbation prediction area, and that is what we are interested in.

The importance of data imbalance problem in single-cell data was highlighted in this UoT paper.

(slide 9 Methodology)

Now, a few words about our approach and the tools we used?

First of all, we used scGen model for predicting perturbations in cells. Given the cells in usual condition (unperturbed) the model can predict, how the cell will look like after perturbation. It is one of the best models for this task.

(slide 10)

scGen is based on Variational Autoencoder with some modifications.

Actually, we implemented our custom VAE model on PyTorch, which replicates the scGen architecture. We use this model for the further analysis.

(slide 11 Modelling)

Basically, our research consist of a series of experiments. In every experiment we train our model on the dataset and evaluate the result.

We have found, that the best strategy of validation is:

- do K-fold cross-validation to get more reliable numbers
- Use R squared metric for evaluating the quality of prediction.
- Have a validation set, which is a mix of several types of cells
- Have a test set, which contains only cells of one type

Now, let's move on to the discussion of our experiments.

(slide 12 Experiment #1: Data imbalance extent effect)

The core idea of this experiment is to find out whether there is any improvement in score if we gradually reduce data imbalance of the training data by using downsampling threshold.

(slide 12 Experiment #2: Data imbalance method effect)

In the next experiment we tested a series of methods for making data balanced.

For example, let's fix a number of cells per class to one thousand. Then, for each class, if it has more cells, cut it to one thousand. If a class has less cells, you can upsample it to one thousand instances. As a result, we get a dataset, where all classes are balanced.

So, we can try different number of cells per class and select the winner.

It tuned out, that upsampling and downsampling is not always a good strategy. Downsampling cuts off the valuable data, that can be used for training. Upsampling actually may be good and may be not, depending on the cells, that we use for validation.

Actually, the best balanced dataset has one thousand cells.

We can see, that not the balance itself matters. The important is a relation between cells in validation set and cells in training set. Given a particular cell type, apparently some other cell types are important for perturbation predictions, and the other cell types are not so important.

(slide 17 Experiment #4: Diversity effect)

In this nex experiment we used dataset with only two types of cells. We tested different proportions of the classes. It turned out that diversity of cells doesn't matter, but one class is clearly much more useful for prediction than another.

(slide 18 More Interesting Questions)

Our experiment lead to these more intersting and specific questions:

1. Could we find a perfect combination of cell types that improves model's learning?
2. Would this perfect training set be balanced or imbalanced?
3. Can we estimate the importance of particular cell type using latent representations?
   It's obvious, that some

4. Do our results generalize to other datasets?

If we had more time for research, we would investigate these questions and use a few different datasets for visualization.

(slide 18 Learning objectives)

Let me summarized what we have learned from this project:

1. We got acquainted with single cell seuquencing data and key results in the field.
2. We learned how AI is used in the studies.
3. In particular, we learned about Variational Autoencoder and other types of generative models.
4. Also, we worked with sources and approaches to perturbation prediction from the studies.
5. Of course, we obtained the colaborative research experience.

Thanks for your attention!

▼ Report

Abstract: background, method, result, conclusion

Introduction: here should discuss background (single-cell analysis, perturbation prediction importance, data imbalance importance)

Method

Experiment/result

Discussion

Conclusion