



ABIN Project

Team Members-:

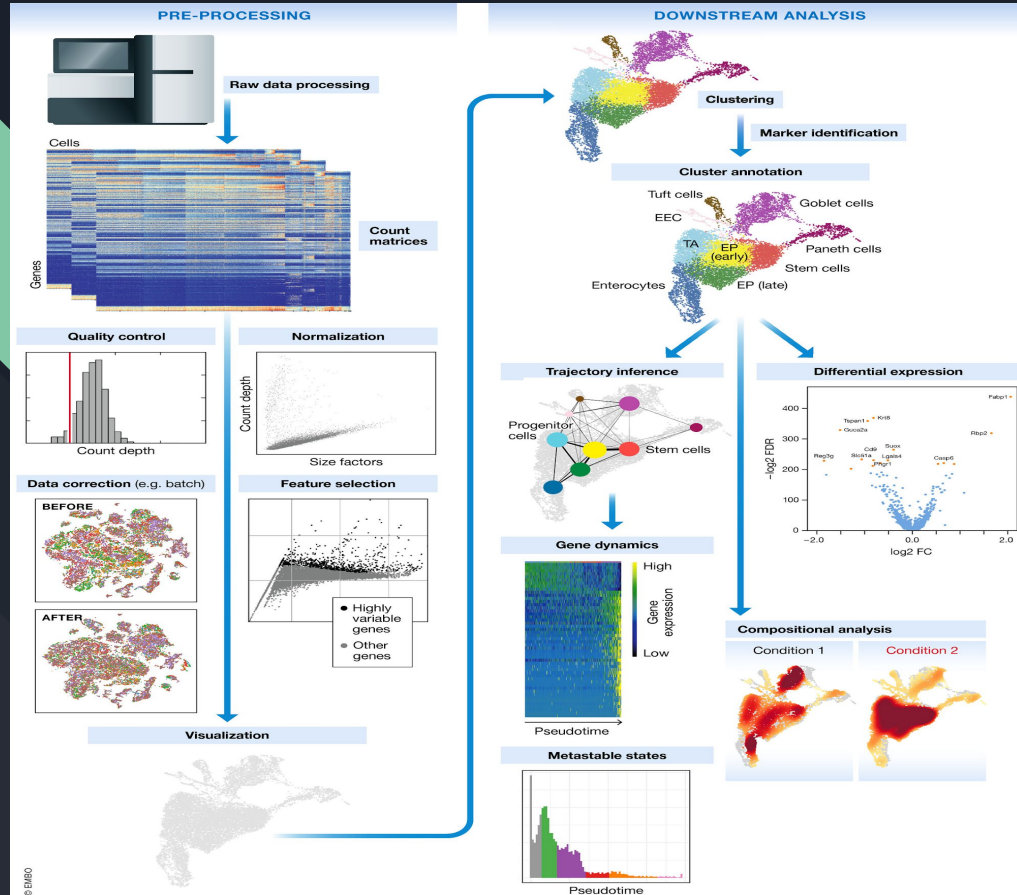
Shreeya Garg(2018415)

Sejal Singh(2018413)

Vrinda Singhal(2018421)

Chhavi Munjal(2018387)

ANALYSIS OF RNA SEQ DATA:





Motivation

Due to low RNA concentration from individual cells, the single-cell transcriptomic data usually consists of extremely high technical noise.


For accurate and precise identification of the differentially expressed genes and cell populations, we aim to reduce technical noise, filter out the lowly expressed genes and remove the genes which has low information .



Optimal Gene Filtering

Gene Filtering is kind of feature selection in unsupervised setting. We can perform gene filtering in three ways:-

- 1) Genes showing similar expression level throughout different cells can be removed as they do not provide much information.
- 2) We can also remove the cells which have most of the genes unexpressed.
- 3) We will filter out the genes which are not expressed or minimally expressed in most of the cells.



Research Paper : Optimal Gene Filtering for Single Cell

Single-cell transcriptomic data are accompanied by extremely high technical noise due to the low RNA concentrations. Identification expressed genes and cell populations is dependent on the reduction of technical noise. OGFSC performs optimized gene filtering for single-cell RNA-seq data

So for this paper they have proposed an algorithm, where they, **construct a thresholding curve based on gene expression levels and the corresponding variances**. Also they validated the method on multiple single-cell RNA-seq datasets. At the end noise were reliably discriminated in the simulated datasets and the results were sharply clustered

when they re-analyzed the dataset from an aging research they found a list of regulated genes which was different from that reported in the original study, because of using different filtering methods.

Reference: <https://pubmed.ncbi.nlm.nih.gov/30535000/>



Understanding R Code for OGFSC

Single cell RNA-seq data which consists of genes vs cells was used for the analysis. We used different techniques to filter out the primary uncorrelated components required for gene filtering. Then, we filtered out the genes using unsupervised and supervised learning algorithms. Further the methodology was tested using cross validation techniques. The number of folds in the cross validation was modified with the learning rate of the algorithm.

Then we further validated our methodology on multiple single-cell RNA-seq datasets. We took into account various simulated and published experimental datasets to conduct our experiment. The following observations were derived:-

- 1) The results show that the signal and noise are well discriminated in the datasets.
- 2) The results of seven experimental datasets clearly shows that the cells of the same types are more sharply clustered using our method.
- 3) Results show that this technique is indeed an alternative opportunity to probe into the true level of technical noise in single-cell RNA seq data.

Modules Needed (Python): OGFSC

1) **Scprep (Single Cell PREparation)**:-It makes it easier to use the Pandas / Scipy / Sklearn ecosystem for single cell RNA-seq analysis.

a) **scprep.plot.plot_library_size()**:It is a helper function for plotting library size from a gene expression matrix

b)**scprep.filter.filter_library_size()**:To filter the samples based on UMI(Unique Molecular Identifier)/cell value.It filters out the cells having too low or too high UMI/cell values.

c)**scprep.filter.remove_rare_genes()**:Remove lowly expressed genes

d)**scprep.filter.filter_values()**:This method takes data and an array values and removes all cells from data where values is above or below the set threshold.

e)**scprep.transform.log()**, **scprep.transform.sqrt()** or **scprep.transform.arcsinh()**:Used to transform the data.It makes sure that each gene or feature in our counts matrix is counted equally. It makes sure that genes that are more highly expressed will be considered more important and will have a larger impact on downstream analysis

f)**scprep.normalize.library_size_normalize(data)**:It is used to normalize the data.It is meant to align the scales of gene expression across cells that have different UMI/ cell values.

2) Numpy

WORKFLOW in Python

```
$ pip install --user scprep
```

```
[ ] data_removeLEG = scprep.filter.remove_rare_genes(data_filter,min_cells=5)
```

Installation / Data Collection

Using PyPI installed
scprep,Scipy,
Scikit-learn. Downloaded
ACTB
ENSG000000075624 from
ensemble

Filtering Cells

We Filter the gene cell
using library size.For
this we have to select
the cutoff threshold.
Which is usually the
number below or above
the mean or median
library size

Filtering lowly expressed genes

if a gene is detected in
fewer than 5 or 10 cells,
we removed those
genes and reduced the
dimension of matrix

Normalization

Through normalization
we divide the count of
each gene in each cell
by the using their **UMIs**
in that cell.

```
[ ] data_filter = scprep.filter.filter_library_size(data_f, percentile=50, keep_cells='below')
```

```
data_ln = scprep.normalize.library_size_normalize(data_removeLEG)
```

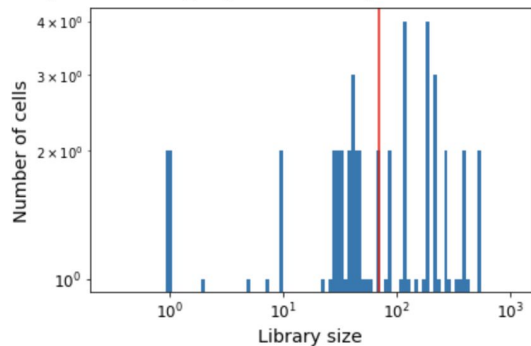

RESULT AND ANALYSIS IN PYTHON

- Filtering cells by library size: We select a cutoff and filter cells by library size

The red line displays the threshold

```
[ ] scprep.plot.plot_library_size(data_f, percentile=50)
```

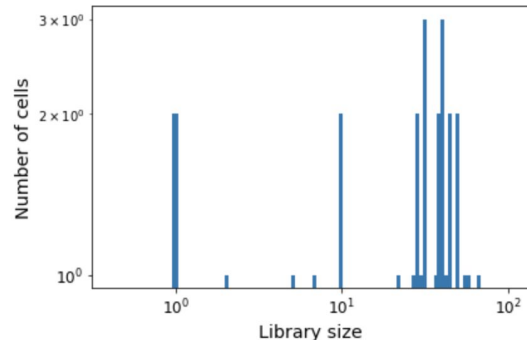
<matplotlib.axes._subplots.AxesSubplot at 0x7f57edb36a>



Filtered dataset Visualisation

```
[ ] scprep.plot.plot_library_size(data_filter)
```

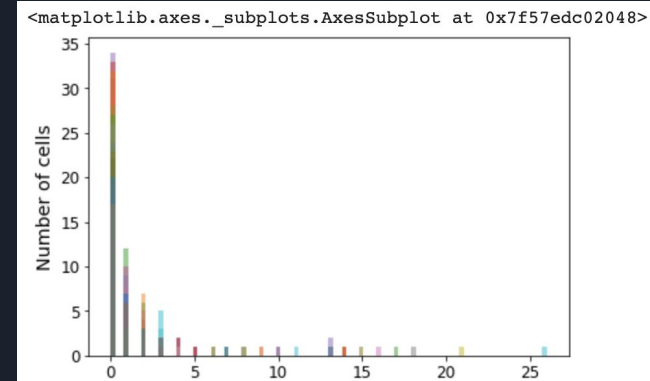
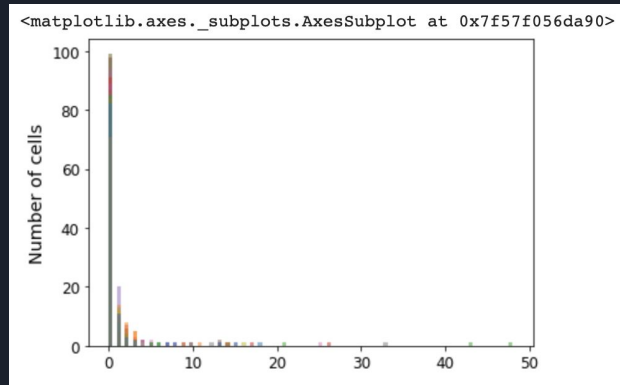
<matplotlib.axes._subplots.AxesSubplot at 0x7f57edf5d198>



Visualizing the library size distribution using scprep

In the graphs shown above, all cells with more than 25,000 UMI(Unique Molecular Identifier) / cell were removed.

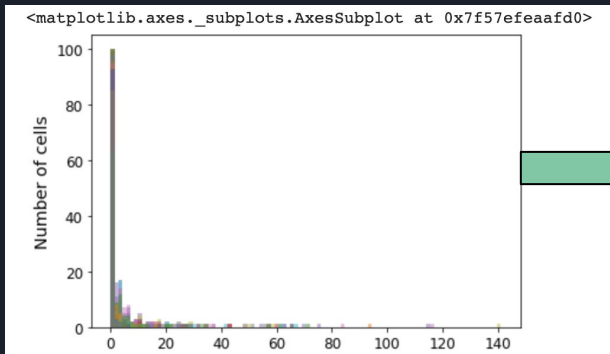
- **Filtering lowly expressed genes:** We removed lowly expressed genes from the gene expression matrix during preprocessing. If a gene is detected in fewer than 5 or 10 cells, it gets removed.



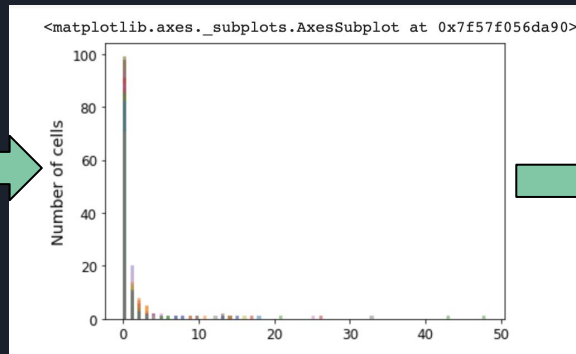
The figure shown above displays the number of cells in which a certain gene gets expressed. We analyzed that approximately 16,000 genes were lowly expressed. These are the genes that gets further deleted from the gene expression matrix

RESULT AND ANALYSIS (CONTD....)

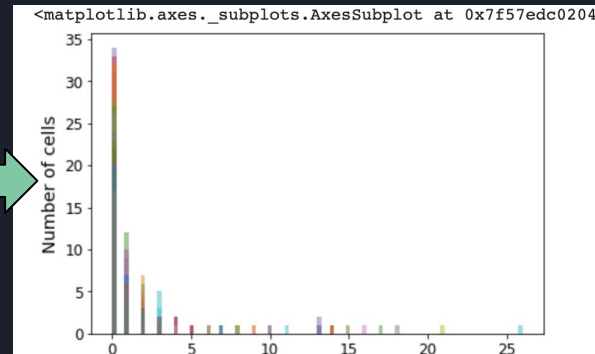
Original Data



Filtered
Library size



After removing lowly
expressed genes





Conclusion

As a result we obtained well filtered dataset devoid of lowly expressed genes. We have also filtered out the genes showing similar expression level throughout different cells as they do not provide much information. These gene filtering techniques help us to achieve a clean and pre-processed dataset which can be fed into any learning model.

Further, we have done Library size normalization to align the scales of gene expression across cells that have different values of UMIs per cell. The process of normalization mainly involves dividing the count of each gene in each cell by the number of UMIs in that cell.



Research Paper:

Voom- Variance modeling at the observation level

The voom method incorporates the mean-variance trend into a precision weight for each individual normalized observation. The normalized log-counts and associated precision weights can then be entered into any statistical pipeline for microarray data that is precision weight aware. Voom applies the mean-variance relationship at the level of individual observations. This method performs well in both the cases i.e. when the sequencing depths are the same for each RNA sample or when the sequencing depths are different. Voom give RNA-seq analysts immediate access to many techniques developed for microarrays that are not otherwise available for RNA-seq, including all the quality weighting, random effects and gene set testing techniques. It can handle heterogeneous data and complex experiments as well as facilitate pathway analysis and gene set testing.

Source : <https://genomebiology.biomedcentral.com/articles/10.1186/gb-2014-15-2-r29#Sec15>



Need for Voom

In RNA-seq applications, the count sizes may vary considerably from sample to sample for the same gene. Different samples may be sequenced to different depths, so different count sizes may be quite different even if the cpm values are the same. For this reason, modeling the mean-variance trend of the log-cpm values at the individual observation level, instead of applying a gene-level variability estimate to all observations from the same gene gives better results.



Algorithm for Voom:

STEP 1: Normalization:

- 1) Normalized expression of a gene should not correlate with the sequence depth of the cell
- 2) Variance of a normalized gene expression should reflect biological variation across cells
- 3) Size Factor method used: Global Scaling
 - a) Assumption: RNA levels do not vary much between cells
 - b) log-normalization

```
data_sq = scprep.transform.log(data_ln, base=2)
```

STEP 2: Visualising and analysing data after transformation

```
scprep.plot.plot_gene_variability(data_sq)
```



VOOM OUTPUT



INDIVIDUAL CONTRIBUTIONS-:

- **Shreeya Garg-:** Modules and softwares,Implemented the Algorithm in R and Python
- **Sejal Singh-:** Research paper,Implemented the Algorithm in R and Python
- **Chhavi Munjal-:** Studied the algorithms,Implemented the Algorithm in Python
- **Vrinda Singhal -:**Studied the algorithms,Implemented the Algorithm in Python



THANK YOU