**Tessa Therapeutics: Normalisation of Single-Cell RNA Sequencing in Immunotherapy**

Kasi Gopikanna Nadarajan

*Data Scientist Intern at Tessa*

**Introduction**

What is Cell Sequencing? Cell Sequencing is one of the most common techniques used in bioinformatics. It is the process of deciphering the nucleotide sequence of a strand of DNA or RNA. DNA helps to explain our genetic make-up while RNA is involved in many cellular processes, with the most significant being protein expression. It helps to dictate what cell would become and ready to do. The common types of RNAs are Ribosomal RNAs (rRNA), Transfer RNAs (tRNA), Messenger RNA (mRNA) with the latter being made into protein, allowing for further analysis. Studying mRNA helps us understand cell responses in response to Cell Stimulation (E.g. CD3), Cell Environment (T-Cells within Tumours), Viral Infections (EBV, HPV).

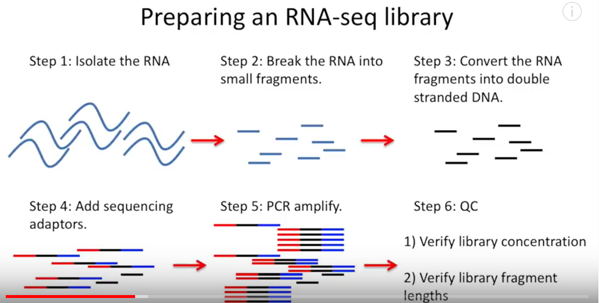
Hence, most RNA-Sequencing studies analyse mRNA. RNA-sequencing is the most commonly utilised, high-thoroughput sequencing method established over the last decade. It uses next-generation sequencing to reveal presence, quantity of RNA in biological sequence and analyses the continually changing cellular transcriptions. Furthermore, it provides ability to identify novel spliced variants, look at alternative gene spliced transcripts, post-transcriptional modifications, gene fusion, mutation, changes in gene expression over time, differences in gene expressions, differences in gene expression in difference groups or treatment (E.g Drug treatment vs Non-Treated). The opportunity to study entire transcriptomes in detail using RNA sequencing has led to many important discoveries in biomedical research in the past decade.

At Tessa, the research has shown that the body’s anti-viral immune response can be harnessed as a potent therapy for cancer treatment. Based on that, it develops EBVVST (Epstein-Bar Virus Virus-Specific T-Cells), a specific set of immune cells, which can be used in actual treatment once it clears all stages of clinical trials. The early results are promising. In particular, the treatment of solid tumours in clinical trial. Using RNA-Sequencing, we can compare the differences between the VST cells and control group of cells and understand the VSTs better. Currently, normal cancer treatment has 40% of success rate survival after two years, while at Tessa, it is 66% of success rate survival over the same duration after Stage II of trials. The aim is to understand the VSTs in terms of the biological composition and improve the accuracy, ultimately to convince insurers.

However, RNA-sequencing is usually performed in "bulk," and the data represents an average of gene expression patterns across a mixture of millions of cells. It is useful at getting general ideas of what is going on but this might not reveal biologically relevant differences between cells or display rare or highly diverse populations. Single-cell RNA-seq (scRNA-seq) is a feasible approach to overcome this problem. By isolating single cells from the mixtures, capturing their transcripts, and generating sequencing libraries in which transcripts are aligned to individual cells, evaluation of fundamental biological properties of cell populations and biological systems can be undertaken at unprecedented scale. Not to mention, the speed and costs of Single-Cell RNA sequencing and molecule input amounts have decreased over the years, making it a more attractive proposition for biotechnologists.

**Single-Cell RNA Sequencing**

Current scRNA-seq methods use 3rd generation sequencing, referring to long reads with no fragmentation and no need for amplification, unlike 2nd generation sequencing. The sequencing system generally used by biotechnologists is Illumina, though there are other competitors coming up. There are many repetitive regions and accuracy issues in the long reads so these need to be dealt with by padding with short reads or reference sequences. At Tessa, the 10X Genomics library preparatory kits are used to generate sequencing data in the raw format. They are outputted as FastQ files. The steps of preparing an RNA-seq library are as follows:



**Fig 1: RNA-Seq library**

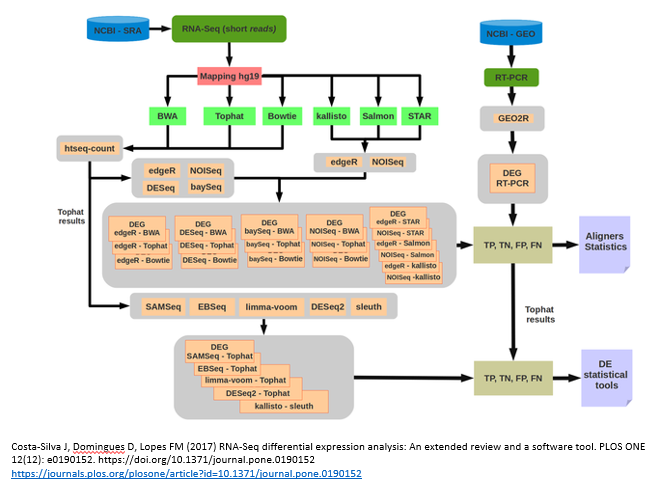
Hence the FastQ files are generated, we will check for the validity of the files before proceeding. The files should have the below format,



**Fig 2: Fastq File Format. Source: *Coursera, Experimental Methods in Systems Biology***

with each set of 4 lines, representing a base-pair sequence of paired-end read. There are separate files for forward and reverse paired-end reads so it will be important to verify this output, before feeding the two files for alignment. The information in the fastq file is not necessary to analyse at this stage. Some bioinformaticians analyse the Base-Pair quality values for Quality Control, but this is just the pre-processing data, so we will not do that. Instead, we will proceed to understand the key steps of the scRNA-seq pipeline and the final expected output, to get a better idea of the data needed to analyse for differential gene analysis.

The overview of the pipeline is as such:



**Fig 3: Overview of scRNA-seq pipeline**

The pipeline can be broken into 5 key steps:

1. Data Preparation using scRNA-seq toolkit
2. Alignment of Cells to Reference Genome
3. QC, Counting of reads by genes
4. Differential Expression Analysis
5. Visualisation and Pathway Analysis

**Alignment of Cells**

Data Preparation has been covered in the earlier section. Alignment of Cells is done to a specific location on chromosome of reference genome (Homo-Sapiens genome) for these paired-end reads, based on the reference genome annotation. The annotation will be in gene-transfer-format. The common human reference genome annotations used are grch38, hg19 and hg38 respectively. However, it is recommended to use the latter, hg38, as it is the more updated one. Alignment can be done using many tools. Star, Tophat are regarded to be the best tools in terms of alignments to well-referenced genomes like human or mouse, but STAR is the most readily used among biotechnologists.

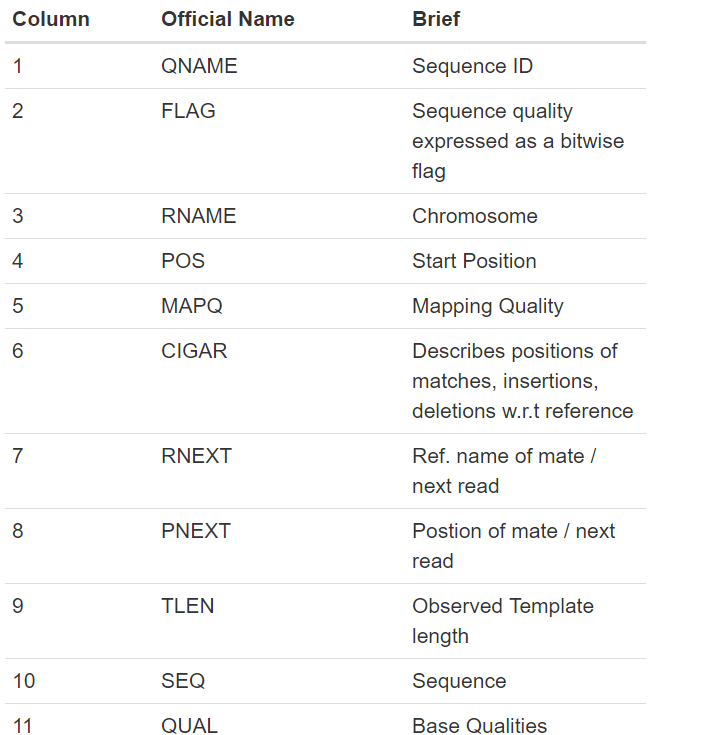
**QC, Counting of reads by genes**

Next, we will have to check the alignments. Ideally, the fastq file should have more than 50% of successful alignments, which can be verified in the BAM file output after alignment. The BAM (Binary Alignment Map) file is a binary compressed version of the SAM (Sequence Alignment Map) file, which is the output too sometimes. The SAM files can be as big as a few hundred gigabytes, which is why most preliminary analysis is done on BAM files. Unlike the fastq files, these files contain all reads, with reads being paired with each other appearing in consecutive lines. In SAM files, the Header lines start with an @character followed by tab-delimited lines. Header will give information about the alignment and reference sequences used. The first part of the header lists the names (SN) of the sequences(chromosomes) used in alignment, their length (LN) and a md5sum ‘fingerprint’ or hash check of the fastq file used for alignment



**Fig 4: Sample output of a SAM file run in Linux environment**

To do further QC, one can also check the various metrics in the bam file, such as CIGAR string, Flag and MAPQ. A description of the various metrics can be seen below.



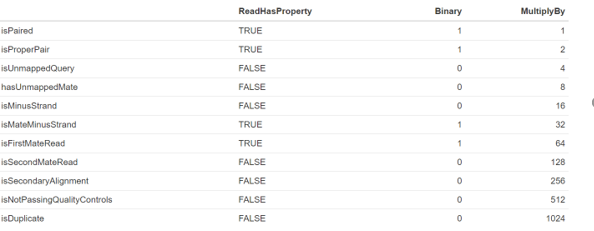
**Fig 5: Metrics in BAM file**

The CIGAR (Compact Idiosyncratic Gapped Alignment Report) string is a way of encoding the reference it has been assigned in the genome. It is comprised by a series of letters and numbers to indicate the mapping as indicated below:



**Fig 6: Code Description in Cigar Mapping**

For example, *68M* means 68 bases match the reference while 1S67M refers to 1 soft-clipped read followed by 67 matches. Mapping quality on the other hand usually has values of 0 or 255, with the latter referring to successful alignments usually. Flag on the other hand, is more intuitive, giving more information about the properties a read can possess. If a particular property is observed, a corresponding power of 2 is added multiplied by 1. The final value is derived by summing all the powers of 2. The maximum flag score possible is 1024 and this is probably a PCR-duplicated read. The list of properties can be seen below.



**Fig 7: Properties of Reads denoted by Flag Score**

So, for example, if a read is paired and is a proper pair, the flag score will be

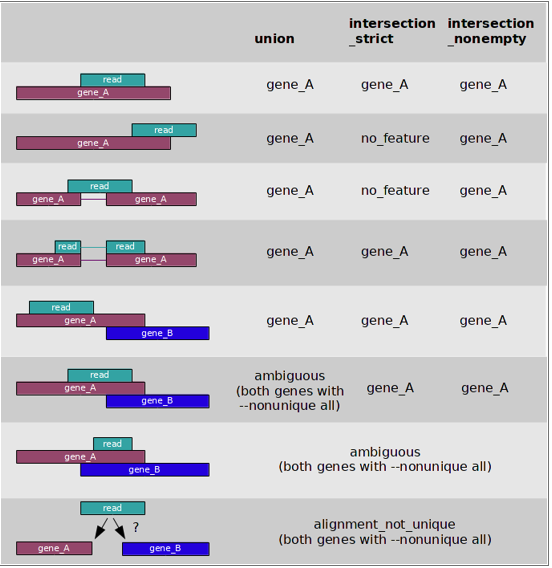
1\*1 + 1\*2 = 1 + 2 = 3

Ideally, flag scores should be low since there is a higher chance, they are the original reads mapped properly. Any flag score above 200, is worrying and worth looking at.

We will also want to know the types of alignments and improve the success of the alignments at the end of the day. There are 5 types of alignments:

1. Aligned Correctly - Gene is mapped to one location of reference genome
2. Alignment Not Unique – Gene is multiply mapped to two locations
3. Ambiguous - Read overlaps two genes and by default the alignment tool would map to the first gene
4. Not Aligned Correctly – Cannot find a match between the read and the gene though that gene is captured in the reference genome
5. No Feature - Cannot find that gene in the reference file

A more detailed breakdown can be seen below.

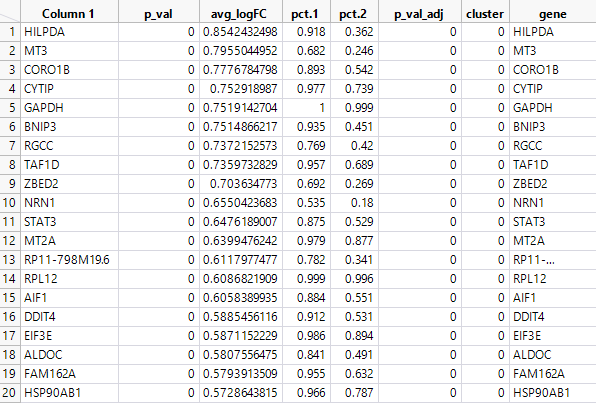


**Fig 8: Different Types of Alignments**

Once we check the alignments and QC of the BAM files, we can proceed to count the number of reads per gene. This is done by checking the RName (Chromosome) and Pos (Start Position) for each QName (Sequence ID) in the BAM file against the reference annotation file with genomic features. The common tool used by bioinformaticians for counting is HTSeq-Count, which will output a text or count CSV file. EdgeR in R can also be used but requires a lot of dependencies in R, so it is better to stick to HTSeq-Count. It is good then to normalise to RKPM(Reads Per Kilobase of transcript per Million mapped reads), to get a weighted value for comparison. Counting the number of reads per gene is especially useful for exploring the relationship between successfully mapped and unmapped reads with genes. Moving forward, it can help to improve the accuracy of the alignments, if we study the gene families of the unmapped reads. Similarly, it can be used for further differential expression analysis.

**Differential Expression Analysis**

Differential Expression Analysis using the count CSV file and the indicated experimental vs control groups to determine the significant Principal Components (PCs), reducing the linear dimensionality in the process. Then, it can be used to help discover the key Gene Markers: which genes are enriched or less expressed in the different groups. The final expected output is a list of genes that are differentially expressed, together with the associated probability of appearing in each sample and test group and the average log fold-change difference between sample and test group. This is also often represented as log 2- fold change. A sample output from running in R is as such:



**Fig 9: Output of Gene Marker Discovery across Clusters**

The common programs used by bioinformaticians for Differential Gene Analysis are DeSeq 2 and in recent times, Seurat. Both are packages in R, but Seurat is faster at processing large files and there is increasing online documentation and support.

**Visualization and Pathway Analysis**

Lastly, we can do Visual Analysis. A popular method is to do non-linear dimensional reduction (TSNE) for cluster visualisation using ggplot2 package in R or other software like Loupe Cell Browser, BioVinci. Also, we can show heatmaps, violin plots, stacked barcharts of top 10 Principal Components, for example, to extract further insights. However, in recent times, bioinformaticians are using Seurat package as it allows you to do the Differential Expression Analysis and Visualisation seamlessly, with the outputs after aligning the reads. Constant upgrades are being made to Seurat package to improve the analysis possible of scRNA-seq data.

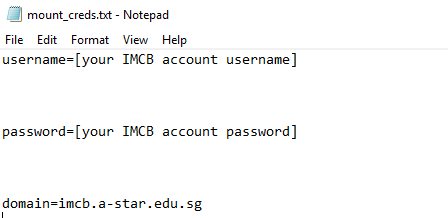
Pathway analysis is the final step where one takes his desired list of genes that he wants to explore and determines if these genes all fall into a specific pathway, curated by others. The Fischer’s exact test (Statistical classification based on contingency tables) is usually undertaken and based on the p-value calculated following the hypergeometric distribution, number of genes provided versus the expected number of genes in that pathway, one can determine enrichment of certain genetic pathways. Common computational tools used for this analysis are GSEA (Gene Set Enrichment Analysis) and Gene Ontology.

**Choice of scRNA-seq Pipeline**

There are many pipelines available for scRNA-seq. Chipster, 10X Genomics Cell Ranger are the commonly utilised ones. We evaluated these pipelines based on sample data processed from the 10X Genomics preparatory kit, using PCR (Polymerase Chain Reaction). However, based on the alignment of reads, in Chipster, 10X Genomics pipeline, the latter seemed to give many times better accuracy as compared to those analysed in Chipster. Not to mention, 10X Genomics, uses leading tools, namely STAR for its alignment and HTSeq-Count for its alignment of reads. So, we will suggest to use the Cell Ranger Pipeline, as it is integrated with other software well end-to-end, before using the popular Seurat package in R for further downstream, visualisation analysis.

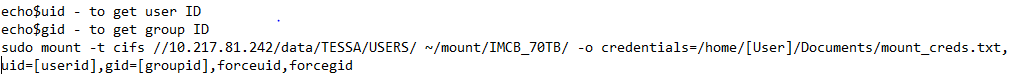
**Installation Steps**

First, one would need to have an IMCB account created, for mounting the IMCB 70TB directory to the mount folder where the files are stored. This is necessary to access large data files stored on the server. Once the account is created, create a mount\_creds.txt in the Linux machine you are using, containing the username, password of the IMCB account and domain. The domain by default is “imcb.a-star.edu.sg”. The layout should be as such:



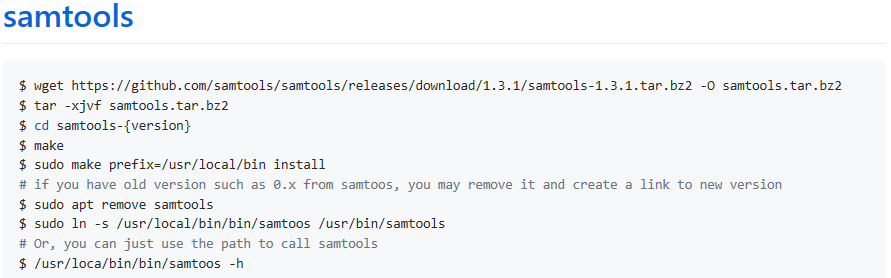
**Fig 10: Mount credentials document**

Next, use these commands to retrieve the user IDs, group IDs before mounting the IMCB 70TB directory to the respective mount folder



If this is done successfully, one should be able to access the IMCB directory from his own Linux Machine.

Next, we need to install samtools for the HTSeq-count option.



Once this is done, we can proceed to install the 10X Genomics Cell Ranger pipeline. The instructions for download and installation can be found at these links,

1)<https://support.10xgenomics.com/single-cell-gene-expression/software/downloads/latest>,

2) <https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/installation> .

Do take note to export the correct PATH on your Linux machine to prevent referencing issues. Also, ensure there is refdata-cellranger folder created, containing the reference genome

Next, we need to run the command to get the output after aligning the fastq files. Do ensure the fastq files have this format, **SampleName\_S3\_L001\_R1\_001.fastq.gz**

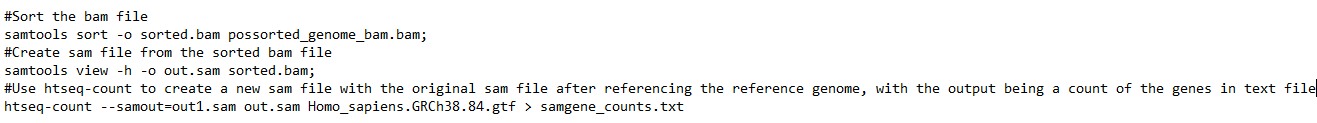
The command is as such:



This is expected to take about 4 h on average depending on the number of reads to align. One can analyse the BAM file output in R using RSamtools package, to do QC and see the alignments breakdown, using this script



After this is done, we will see an **outs** folder created within the directory we mentioned. We can proceed to use htseq-count to count the number of aligned reads per gene and do further analysis on those reads not aligned successfully



Do note, this is computationally heavy and can last for about 8 h on average for a 300GB SAM file. Also, if there is not enough computational power, one might not be able to extract information of the reads not aligned correctly, as I experienced in my project.

It is still possible to do downstream and visualisation analysis using Seurat package in R but Seurat will only take those reads aligned successfully to genes, for analysis. It is ideal to do the previous step successfully and analyse from the Count text file, to improve the accuracy of read alignments so that when the input is passed into Seurat, more genetic information can be extracted. The inputs passed to the Seurat are

1)barcodes.tsv

2)genes.tsv

3)matrix.mtx

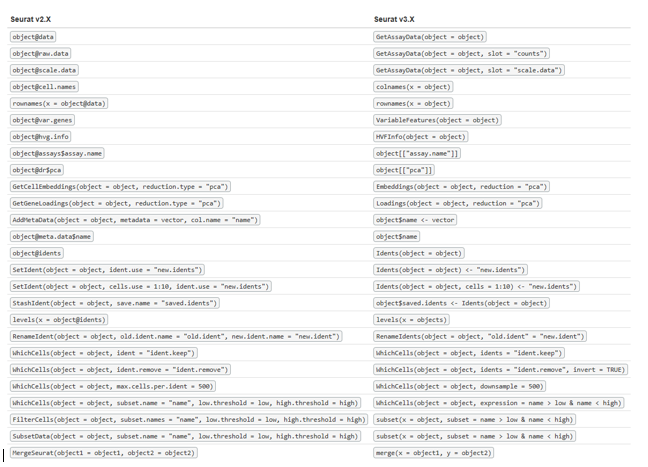
found within the filtered feature matrices sub-folder within the **outs** folder. We will need to get the inputs for the 3’ and 5’ scRNA-seq data as we will want to normalise them. 3’ and 5’ data refers to data prepared by two different preparatory methods to extract the scRNA-seq data. The 3’ data is generally preferred because it identifies genetic information meant for differential expression analysis better while 5’ data normally identifies the T-Cell receptors. A key part of my project is to normalise the 3’ and 5’ data set, so that much information is not lost and the shared overlap will not be counted twice. Reference the output to the outs folder so that Seurat can read the corresponding files.

**Differential Expression Genes Analysis**

Do refer to the Standard Seurat Workflow tutorial. For this, we will take the 3’ and 5’ data from the scRNA-Seq data processed. They can be found in the /home/kasi/20181008\_3GEX/

and /home/kasi/20181008\_5GEX/ folders on the 10.217.81.100 machines. There is 4 sets of donor data, namely that of DM, RH, ZHB and WYO09

Do take note that I mainly used Seurat package for my workflow, and it keeps being continually updated to accommodate more statistical functions and visual analysis. When I started, I was using Seurat 2.0 initially and currently it is Seurat 3.0 so do take note if you happen to come across some Seurat 2.0 scripts and they do not work.

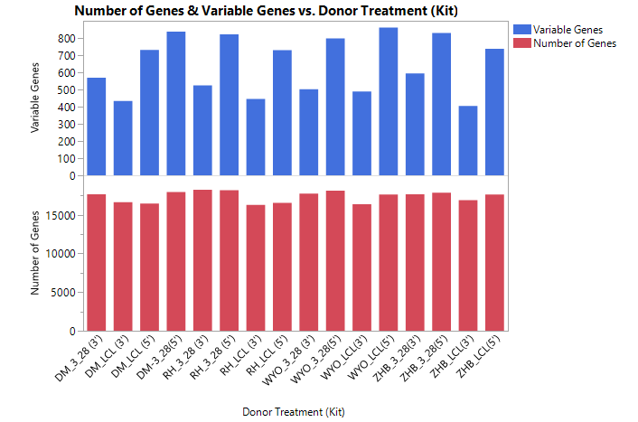


**Fig 11: Seurat Changes**

Do refer to the basic standard Seurat workflow tutorial. For illustration purposes, I will just take the 3’ and 5’ data for one sample donor, DM, for both ATC and VST conditions. Also, the plots derived are that of the 3’ data. Do take note 3\_28 and LCL could also be used interchangeably with ATC and VST respectively.



Once we read the files in R and load the necessary packages, we will filter by reading the meaningful data in with genes with minimum cell count of 3 and features of 200. Next, we will understand the dimensions of the data. In the human genome, there are approximately 20000 genes. Majority of the genes would be captured in our donor samples. A breakdown of the genes and highly variable genes is shown



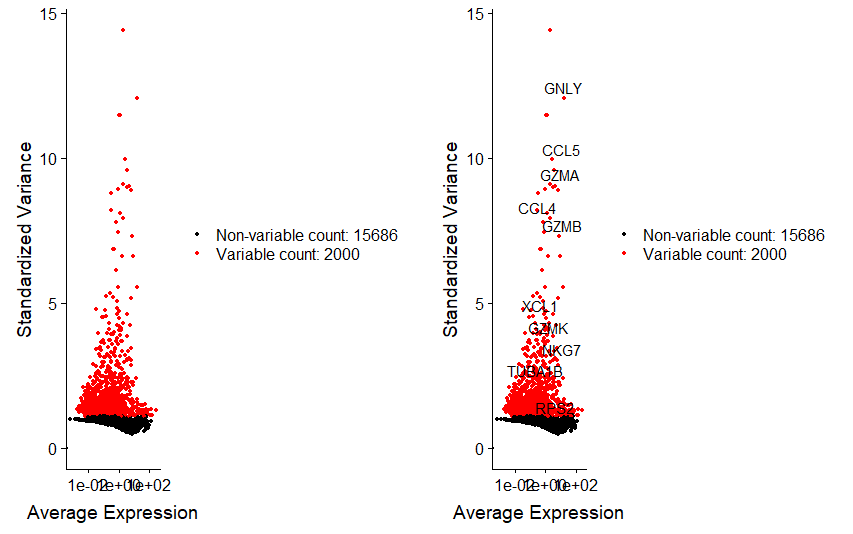
**Fig 12: Breakdown of Genes and Variable Genes by Donor Treatment**

One can see there is different number of genes across all donors. Also, the variable genes, the number of genes with significant dispersion to explain the variation in the data, is less than 5% of all the genes in each sample. This is what we will explore later in pathway analysis and use for PCA (Principle Component Analysis – Identifying key components to explain data while reducing multi-dimensionality).

Next, we will proceed with the pre-processing of data where we will identify the mitochondrial genes to regress out as they are not that significant before filtering out cells that have unique feature counts over 2,500 or less than 200

and cells that have more 5% mitochondrial counts. We will then do normalising of data using so that we can reduce the skewness of data.

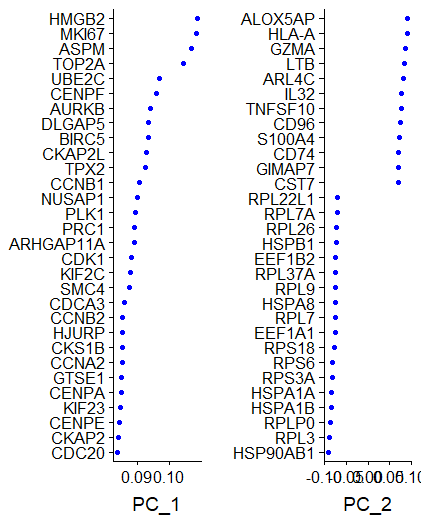
Following that, we will identify the variable genes and see some of the highly variable genes



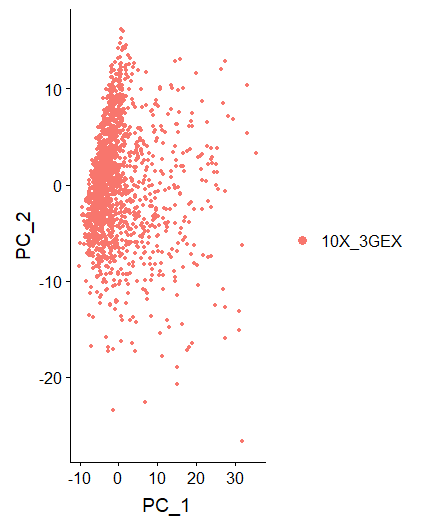
**Fig 13: Average gene expression and variance (dispersion) of Variable Genes**

Subsequently, we will do Scaling, a standard pre-processing step before PCA. Shifts the expression of each gene, so that the mean expression across cells and so that the variance across cells is 1. It is recommended to do the regressing out of mitochondrial genes and unique molecule identifiers(identifier to help reads be assigned to its original cell by effectively removing PCR bias and thus improving accuracy)

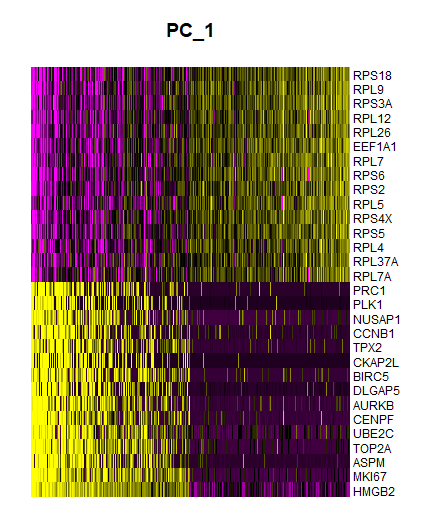
Now, we are ready to do PCA. We will run PCA on highly variable genes to improve performance of results. We can visualise the results of the PCA in a few ways based on your preference(Dimension Loadings, Dimension Plot, Heat Map)



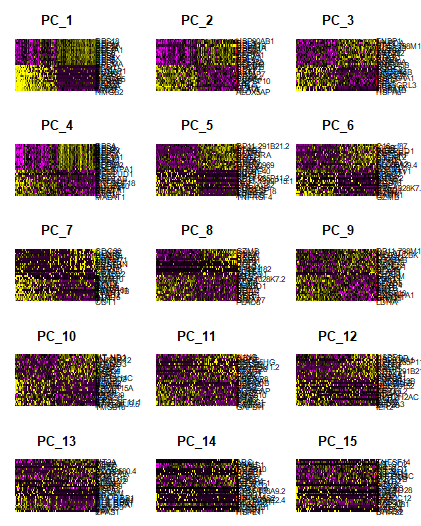
**Fig 14: Dimension Loadings of first two PCs**



**Fig 15: Dimension Loadings of first two PCs**

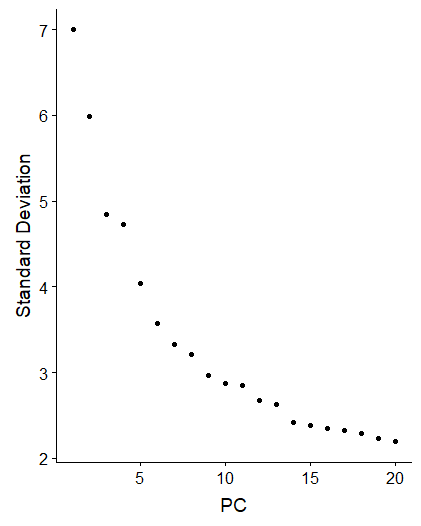


**Fig 15: PC Heatmap of first PC**



**Fig 16: PC Heatmap of first 15 PCs**

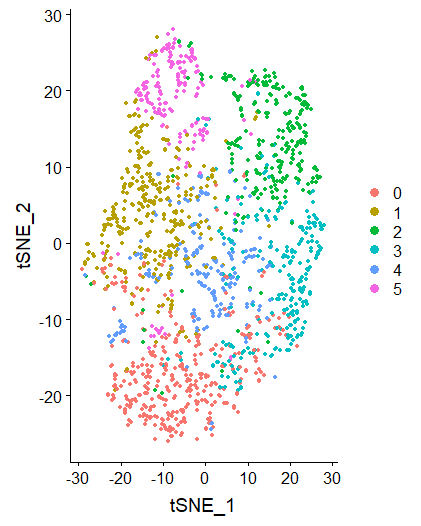
This will give you an idea of what is contained in the PCs and whether they contain your genes of interest, like CD4+ or CD8 genes, as what is of much interest in Tessa. But, a more important question is what is an ideal number of PCs to choose for further analysis? One can consider statistically popular method in many fields, the Elbow Plot, where you look at the kink, before adding any further PCs does not make much difference. Normally, a good range of PCs is 7 to 12. In the diagram below, you would find the kink is at about 10 PCs. So, we shall use that.



**Fig 17: Elbow Plot**

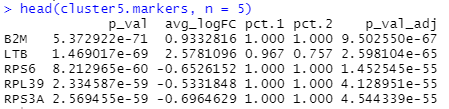
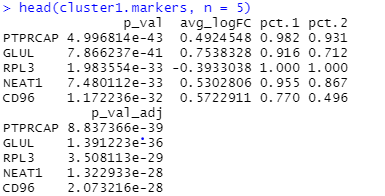
Based on that, we can proceed to visualise the clustering. tSNE plots are usually used in the field of bioinformatics, as unlike PCA plots which use linear dimension reduction technique, it is a non-linear dimensional reduction technique which helps to, high-dimensional data for visualization in a low-dimensional space of two or three dimensions better. Previously, PCA method was commonly used to calculate the clusters. Now, a recent method, CCA, is being used. Instead of the PCA method (do PCA and compute k-Nearest Neighbours (kNN) of all points which only identifies sources of variation even if only present in one individual experiment), CCA method considers neighbours of points which are shared between two different sets and maximises correlation between these sets.

In Seurat, using the ***FindNeighbours*** function, the CCA method uses gene features to maximise correlation and identify biological structure and batch/experiment effect and by default, considers the previously defined top 10 PCs derived from PCA. It constructs a K-nearest neighbours graph based on Euclidean distance based on shared overlap in their local neighbourhoods. We will use the CCA method and derive the tSNE plot as shown below.



**Fig 18: tSNE plot using CCA method**

Now, we have derived the tSNE plot, we can view some of the markers of clusters of interest to get a rough idea of the clustering. This can be done through the ***FindMarkers*** function. Also, we can find the markers present in one cluster, not present in other clusters. Similarly, we can derive the top two markers for each cluster for comparison purposes, using the ***FindAllMarkers function***.



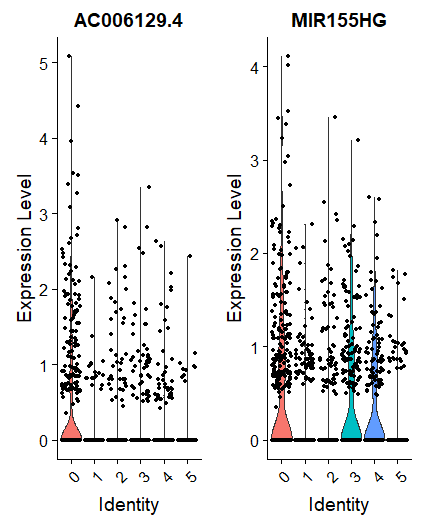
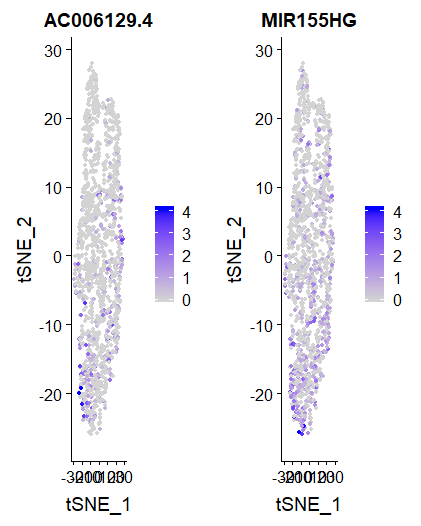
**Fig 19: Cluster Markers of Cluster 1 and Cluster 5 (Different from Cluster 0 and 3)**



**Fig 20: Top Two Markers of each Cluster**

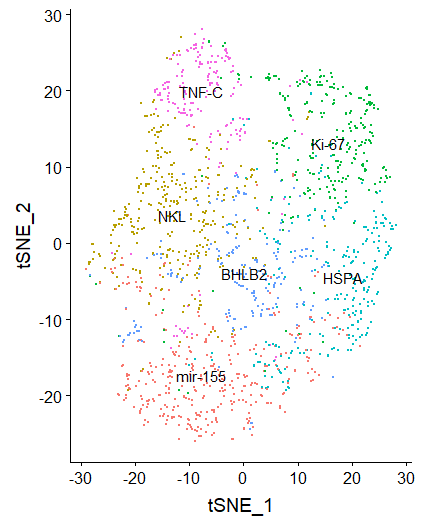
You might wonder how to interpret the data above. The avg\_logFC (also known as log2fold change) is the log-fold change of average expression of differentially expressed genes between 2 groups. Positive values indicate that gene is upregulated in first group. This is calculated using the Wilcox rank-sum test (test if equally probable that a randomly selected value from one group will be equal to a selected value with the same rank from a second group). The columns pct.1, pct.2 refer to the percentage of cells where genes is detected in 1st group and 2nd groups respectively.

Once this information is derived, one might want to see certain gene markers expression levels and tSNE dispersion values across clusters visually. The most commonly used are VInPlot and FeaturePlot as indicated below.

**Fig 21: VInplot, FeaturePlot of Markers**

Lastly, we can draw the tSNE plots again by assigning cell type identity or other features as displayed below.



**Fig 22: Redrawn tSNE plot**

There are other features that can be explored for differential expression gene analysis, but these are the basic ones. Do feel free to explore on your own.

For further clarifications on the standard Seurat workflow tutorial, can refer to <https://satijalab.org/seurat/v3.0/pbmc3k_tutorial.html>

**Pathway Analysis**

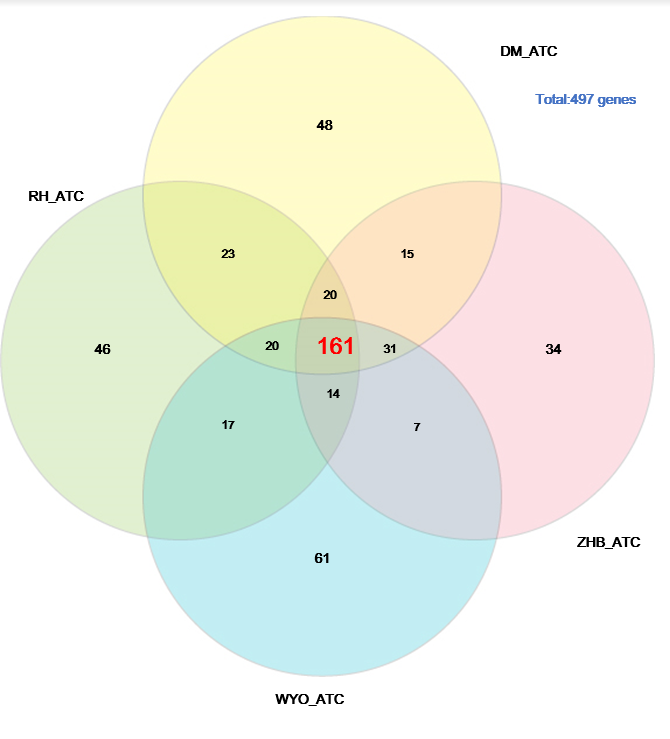
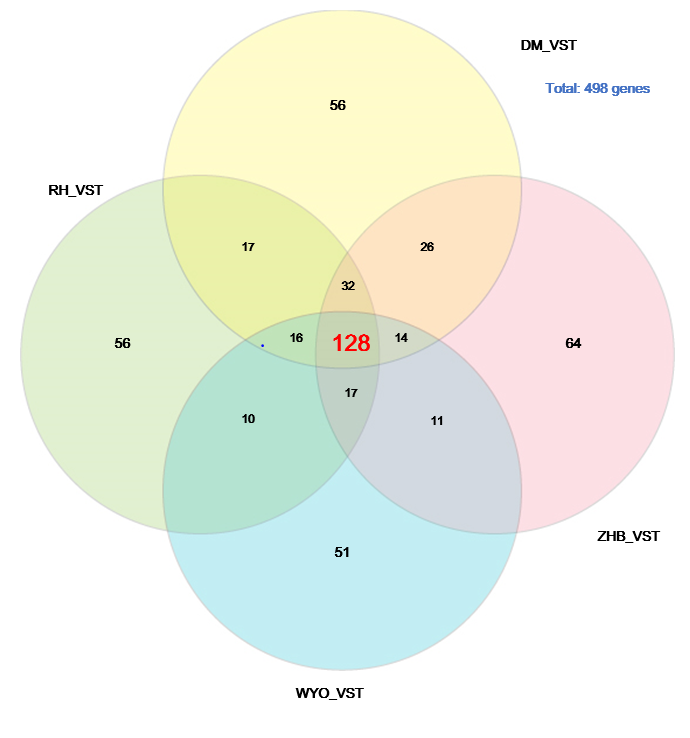
In biomedical research, pathway analysis is used to identify related proteins within a pathway and possibly help in identifying certain diseases that are a result of this pathway. We will do similarly for data of all the Donors. We will reuse some parts of the tutorial covered above. We will realise based on our Exploratory Data Analysis (EDA) of the Highly Variable genes, that there is a kit batch effect between 3’ and 5’ data for each donor(Refer to Fig. 12) and we will need to take care of it. The steps can be followed in the R script below.



Following that, we will have 8 sets of data by donor condition only and we will find the intersection of variable genes across all 8 datasets using Set Theory. This can be subsequently passed to the Gene Ontology Resource website, <http://geneontology.org/>, to find out the common genetic pathways. This is done by running the Fisher’s exact test and finding the upregulated pathways. Similarly, we can find the intersection of variable genes across VST data and ATC data, while capturing the variable genes that appear in at least 6 datasets. The latter would be useful for further analysis. Finding the different number of intersections across the datasets would have a lot of permutations and combinations. Hence, do refer to the steps in the R script below.

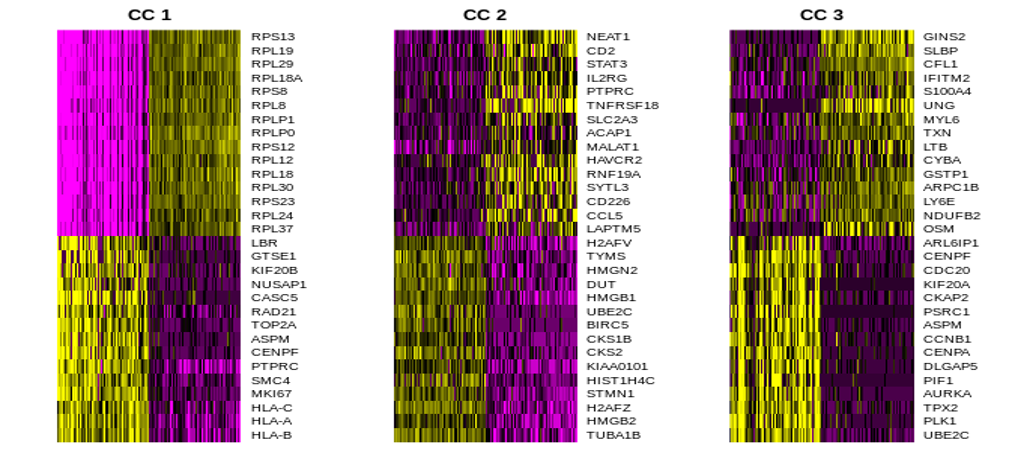


We will derive this output where only 61 variable genes intersect across both donor condition. Also, we can find only about 25% of all variable genes intersect across each donor condition, meaning there is donor-specific batch effect and this needs to be removed.

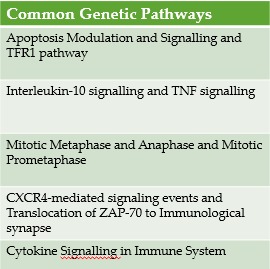
**Fig 24: Intersection of Variable Genes across various donors with same condition**

We can also plot a heatmap of the intersection of all variable genes, to investigate further.

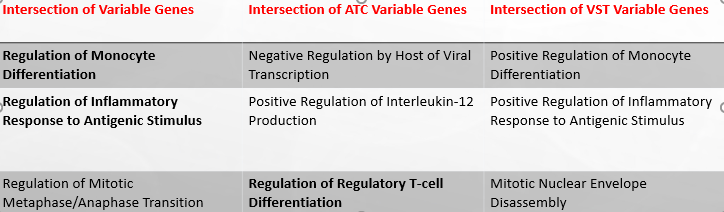


**Fig 25: Heatmap of the Intersection of Variable Genes across both conditions**

Next, we will pass the highly occurring genes, those that appear 6 times or more, and the intersection of variable gene genes across both conditions, to the Gene Ontology website. Similarly, we can derive the output for the intersection of variable genes across each condition. The results are as such:



**Fig 26: Common Genetic Pathways of variable genes that appear 6 times or more**



**Fig 27: Common Genetic Pathways of variable genes that appear across the different donor conditions**

We will see as expected, that the ATC control experiment has high CD4 regulatory, T-cell differential pathway causing viral transcription to be negatively regulated and other gene pathways to be suppressed. The pathways are a good gauge to see too if the data is reliable, since VST variable genes should not have Regulatory T-cell differentiation and if it does, you know something is amiss. Further analysis of genetic pathways can be done for the donors and conditions of interest and can be presented using other visual forms, like graphs.

**Visualisation Analysis**

# We now know that there are kit-specific and donor-specific batch effects and these need to be removed. We will need to do batch correction for the 3’ and 5’ data, when we combine all data. To account for the differences in composition between two replicates or two different experiments, must first match individual cells across batches to find the overlapping biological structure. A recent biological paper, *Batch effects in single-cell RNA sequencing data are corrected by matching mutual nearest neighbours* (can be accessed at

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6152897/>), shows the success of matching mutual nearest neighbours.

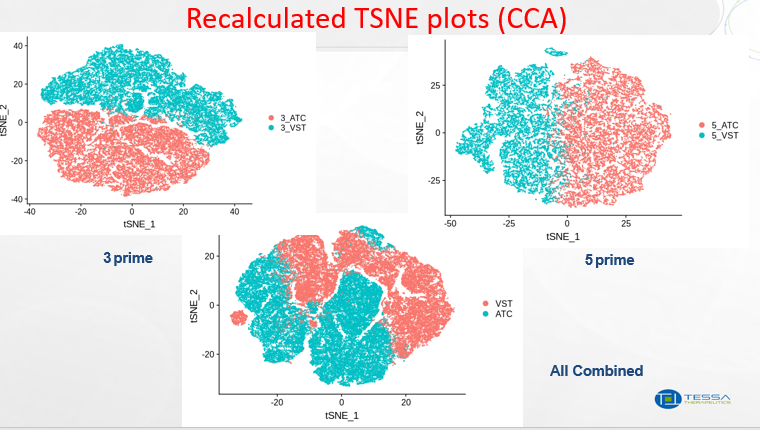
mNN helps to remove the batch/experiment effects from the entire expression matrix to return the corrected matrix and can be done in Seurat itself, using the scale-normalized log expression values already derived using initial normalization. After understanding it, I attempted to use mNN method for batch-correction and compare the results of batch correction, with CCA and PCA methods. CCA, can be seen as an intermediate step for mNN, as it learns the shared overlap in local neighbourhoods of data points and attributes to batch effect or biological state, before passing this vector to mNN. mNN assumes these dimensions are orthogonal to each other in high dimensional expression space before doing the correction. There are other batch correction methods like Combat but they are slower, outdated and would need to convert to log counts per million units based on read counts.

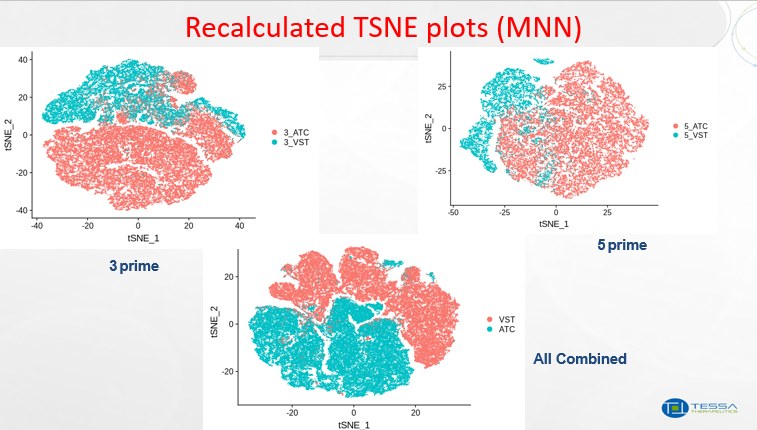
Based on results of PCA method, the batch correction was not effective as those with the same donor condition were largely segregated and verified by my supervisor. Unfortunately, I do not have the results to display that as I could not retrieve the data, from one of the GPUs that got burnt down and Seurat 3.0 does not allow that function anymore. Instead, Seurat 3.0 encourages you to use CCA method, at least.

We can derive the results of the batch-correction for both methods by redrawing and calculating the tSNE plots, based on cluster analysis of cell embeddings and analyzing which cluster corresponds to which donor condition. This can be done through the ***Embeddings*** function and you can follow the batch-correction workflow



The results of the clustering for both CCA methods and mNN methods are indicated below:





**Fig 28: Recalculated tSNE plot using mNN and CCA methods**

One can observe that the combined data of all donors by the various conditions and kits seems to be more clustered together in the mNN method as compared to CCA method, meaning that the batch effect is generally removed. However, it is interesting that the CCA method performs better for the kit-specific data. It is worth investigating why this occurs based on the genes that do not map from the kit-specific data to the combined data, as a starting point. Unfortunately, I have not managed to go past that step, but it is something worth exploring. Nevertheless, at the end of the day, mNN batch correction works well for the combined dataset and this can hopefully be applied to other scRNA-seq experiments. Similarly, we can apply the same concepts to bulkRNA-seq data and see if the batch correction is done successfully.

**Future Works**

Moving forward, it is important to first analyse the unmapped reads in Google Cloud or environments with great computational power and improve accuracy of read alignments hopefully. This will help to convince insurers, as more can be understood about why the EBVVST works better than normal cancer treatment at the genetic level. This can be then applied to improving the EBVVST and improving rates of survival beyond 2 years for Cancer treatment.

Next, can apply Genetic Pathway Analysis on the mNN normalised dataset using Gene Ontology and do further differential gene expression analysis. (E.g. Understanding of how the genes are mapped from one cluster to another and which gene markers are frequently mapped or analysis of markers of interest). Furthermore, can try to improve the batch correction by optimising the parameters or finding new methods. Hope this project findings found lays a small foundation for future work within Tessa Therapeutics.

**Acknowledgements**

Firstly, I would like to thank my main supervisor, Research Scientist, Yang Enjun for his continued guidance in explaining biological concepts, statistical methods I could explore, direction to head towards when I was stuck, sharing of expertise on RNA-Sequencing and Immunotherapy, being patient and understanding with me. Not to mention, he was accommodating of my school commitments.

Secondly, I would like to thank Chief Data Scientist, Sean Tan, for considering me and giving me the opportunity to work at Tessa, while being a good, supportive and approachable team leader.

Lastly, I would like to thank the rest of the Data Science Team (Xiang Wenwei, Rujun Guan, Julianne Weller, Aurore Gay, Damien Marlier, Richard) for their continued support, ideas and my ability to learn about different biological techniques, statistical methods from their different projects.

**Resources**



**Summary Slides of scRNA-seq Normalisation project**