Bioinformatics Final Project Report

Team 10 Members: *Elliot Liu, Christos Skoundridakis, Donovan Spall*

Link to Github: *https://github.com/Donovan55/BioinformaticsProject*

Table of Contents

1. Abstract 3

2. Introduction 3

2.1 Basic Intro 3

2.2 State Problem 3

2.3 Result 3

2.4 Context 3

2.5 Broader Impact 3

3. Methods 4

4. Results & Discussion 28

5. Conclusions 31

6. References 31

1. Abstract

Human gene expression can often change during the progression of a disease and after recovery, influencing the immune system's response and subsequent acquired immunity. These changes can be measured with mRNA sequencing, which can provide insight into the type and quantity of proteins that cells produce at different stages of a disease and after recovery [1]. Zika virus infection can be divided into three stages: early acute, late acute, and convalescent [1]. We wanted to identify a set of genes whose expression values would allow us to determine the stage of infection of a particular individual. Here we show that standard multidimensional analyses of gene expression and clustering only reveal two distinct gene expression groups instead of three, but we were able to achieve a somewhat high maximum predictive accuracy when separating early acute and late acute samples with a support vector machine. We found that consensus clustering strongly supported the existence of two clusters of gene expression instead of three. This suggests that there may be only two well-defined clusters of gene expression. Our results demonstrate that it may not be possible to distinguish between a late acute and convalescent subject infected with Zika using mRNA sequencing because there are no significant differences in gene expression between these two stages [1]. However, our results may be a starting point for further investigation into the support vector machine's performance, as it significantly outperformed our other models and could be refined to increase its accuracy. A possible confounding variable would be whether the subject was Zika-exposed and Zika-naïve, as our dataset contained both types of samples, but lacked this information. The difference in gene expression between Zika-exposed and Zika-naïve individuals in the same stage of infection may warrant further investigation.

2. Introduction

2.1 Basic Intro

The Zika virus is a mosquito-borne virus that has emerged as a significant global health concern due to its association with neurological complications, particularly in fetuses and newborns. Exploring the connection between Zika infection and gene expression holds the key to deciphering the molecular mechanisms underpinning its varying stages, offering crucial insights into its progression and potential therapeutic targets.

2.2 State Problem

Understanding the stage of Zika infection through gene expression could revolutionize diagnosis, treatment, and containment strategies, enabling targeted interventions and personalized medical approaches. Identifying distinct molecular signatures for each stage could offer invaluable insights into disease progression, aiding in early detection and potentially mitigating the severity of the infection. The purpose of our project was to answer the scientific question: can we determine the stage of Zika based on gene expression?

2.3 Result

Using a public dataset found on refine.bio and techniques through R programming, we worked to understand the connection between gene expression and the stages of Zika. The gene data involved 1,021 samples and were classified as either early acute, late acute, or convalescent. Here we show that the stage of Zika can in fact be determined, or accurately predicted, through the examination of gene expression.

2.4 Context

Our findings illuminate a promising link between gene expression patterns and the distinct stages of Zika infection, underscoring the potential for diagnostic and prognostic applications.

2.5 Broader Impact

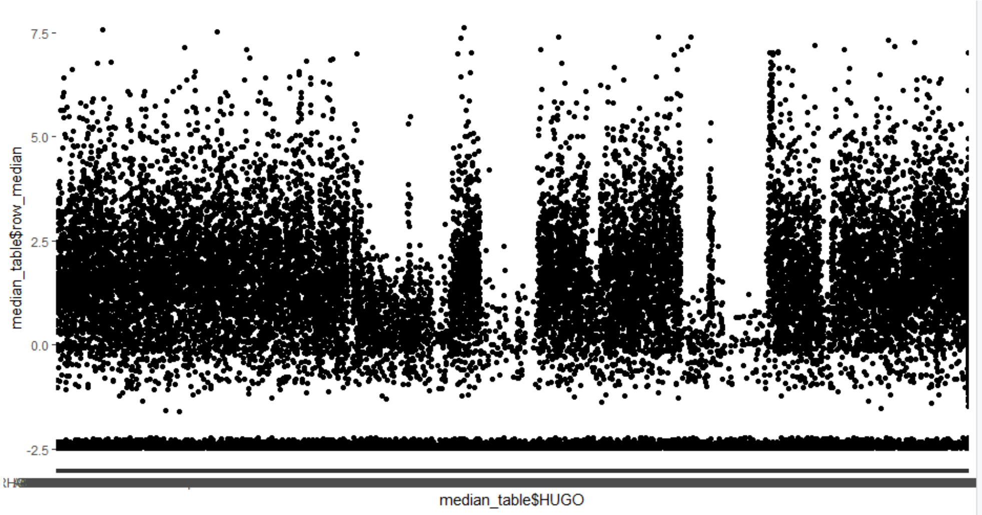
This understanding can help understand Zika, but also sets the precedent for further investigations into both the Zika virus as well as gene expression and its connection to other viruses, symptoms, and more. Our approach involved a variety of techniques including gene expression analyses, the production of plots and heatmaps, differential analysis, enrichment analysis, clustering algorithms, and predictive methods. Each of these methods allowed us to better understand the relationship between gene expression and the stages of Zika.

3. Methods

Below we will explore some of the different methods that were used to solve the scientific question.

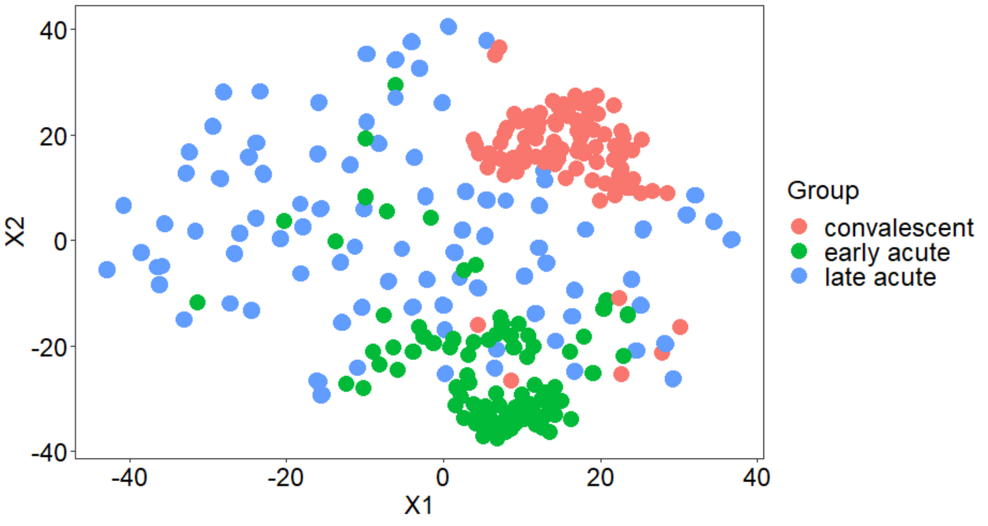
Expression Data Plots:

Density Plot:

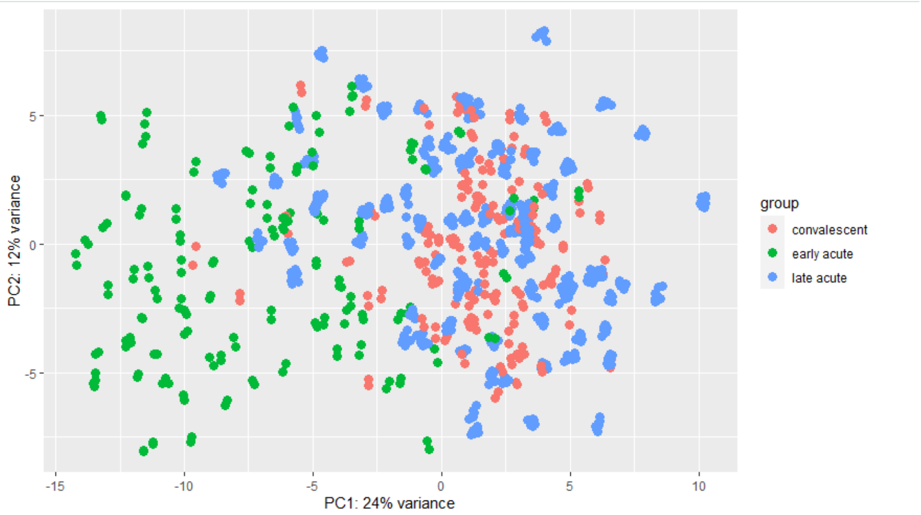


The expression matrix includes 43,363 genes (rows) and 1021 different samples/observations (columns). To create this graph, we used R to log scale every observation in the table and find the median expression of each gene. Then, we used ggplot to generate the plot. The distribution is generally wide and spread-out, which indicates high variation in gene expression. An interesting result is the gap in median gene expression between −2.5 and about −1. Considering how spread-out gene expression is in the other parts of the plot, this gap seems to be an exception.

PCA Plot:



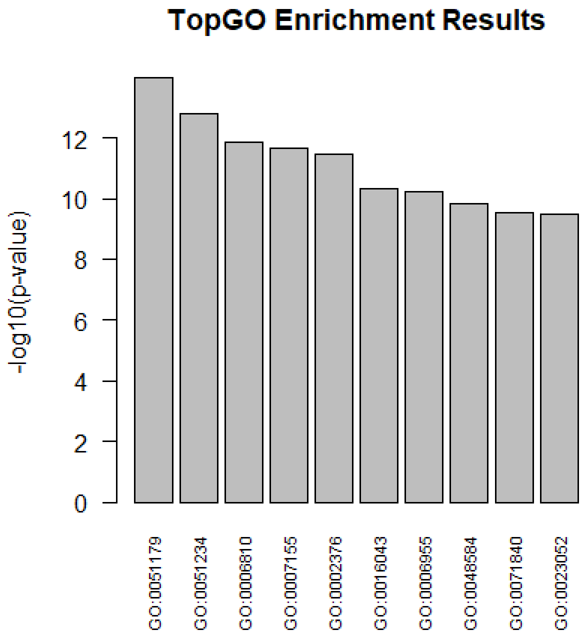
This PCA (principal component analysis) plot visualizes sample relationships based on the log-scaled data. To generate the plot, we used the raw expression matrix to create a dataset from the DESeqDataSetFromMatrix function, performed a variance stabilizing transformation on it with the vst function, and then used the plotPCA function with the disease stage as the grouping. An interesting result is that convalescent and early acute gene expression are more clustered, while late acute gene expression is much more spread out.



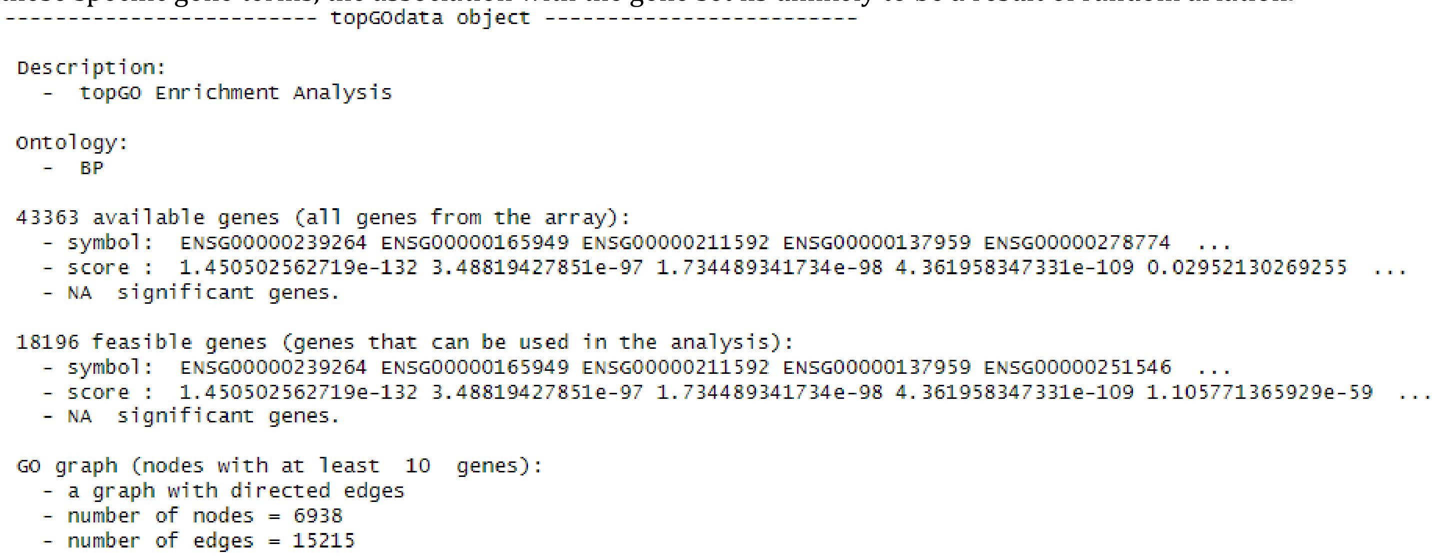
We generated it with the tsne function with default parameters and the metadata column with the disease stage to group observations. This t-sne plot does not reveal a strong global pattern in similarity between the different stages of Zika. Although we can see that small groups of individuals have similar gene expressions because they are clustered on the plot, there is a considerable amount of variation within the wider disease stage groups, which overlap significantly.

Enrichment Analysis (GSEA)

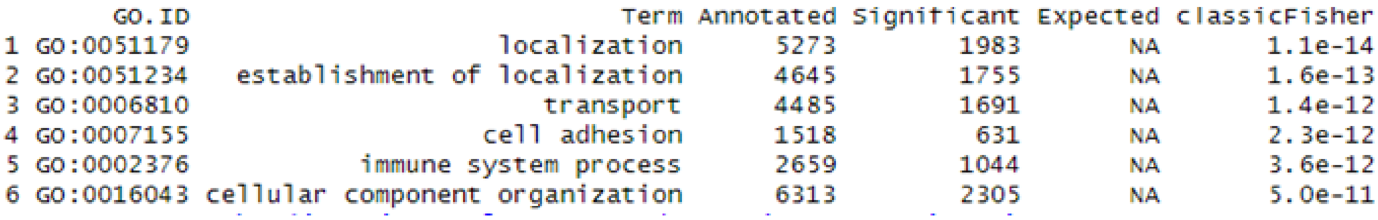
TopGO:



The topGO method was used to evaluate which gene ontology terms are associated with the set of differentially expressed genes. The topGO data object was configured with parameters, including pvalueCutoff=0.05 and qvalueCutoff=0.2, to set the significance thresholds for p-values and q-values respectively. Next the enrichment analysis was performed through GOresults <- runTest(…) using the classic algorithm and fisher’s exact test. Finally, before creating the bar chart, a table was created of the significant terms. In the chart and plot we can see the –log10(p-values) for the top 10 enriched GO terms based on statistical significance. As a result, we can tell that for these specific gene terms, the association with the gene set iis unlikely to be a result of random ariation.



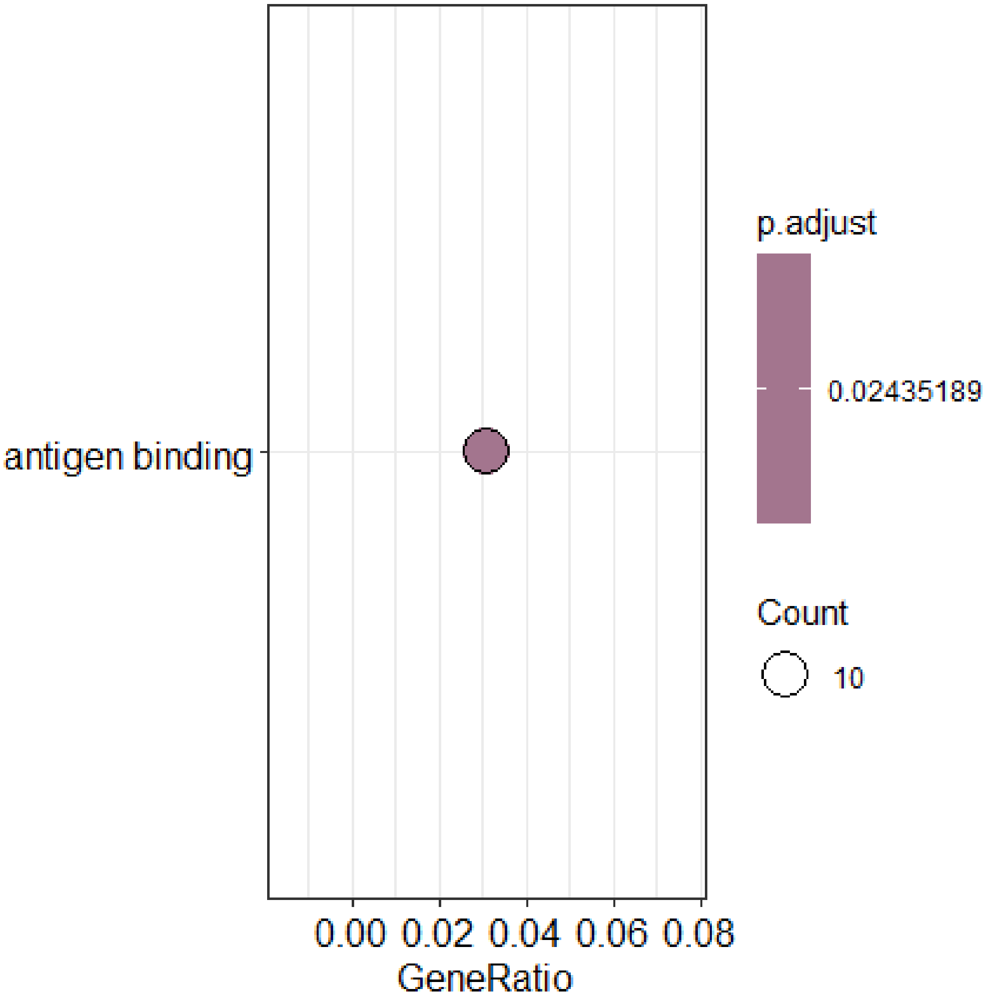
```
------------------------- topGOdata object -------------------------

 Description:
   -  topGO Enrichment Analysis

 Ontology:
   -  BP

 43363 available genes (all genes from the array):
   - symbol:  ENSG00000239264 ENSG00000165949 ENSG00000211592 ENSG00000137959 ENSG00000278774  ...
   - score :  1.450502562719e-132 3.48819427851e-97 1.734489341734e-98 4.361958347331e-109 0.02952130269255  ...
   - NA  significant genes.

 18196 feasible genes (genes that can be used in the analysis):
   - symbol:  ENSG00000239264 ENSG00000165949 ENSG00000211592 ENSG00000137959 ENSG00000251546  ...
   - score :  1.450502562719e-132 3.48819427851e-97 1.734489341734e-98 4.361958347331e-109 1.105771365929e-59  ...
   - NA  significant genes.

 GO graph (nodes with at least  10  genes):
   - a graph with directed edges
   - number of nodes = 6938
   - number of edges = 15215
```



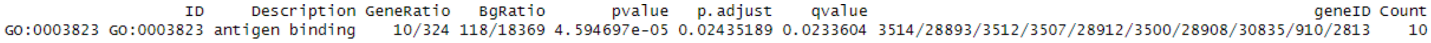
```
       GO.ID                           Term Annotated Significant Expected classicFisher
1 GO:0051179                   localization      5273        1983       NA       1.1e-14
2 GO:0051234   establishment of localization      4645        1755       NA       1.6e-13
3 GO:0006810                      transport      4485        1691       NA       1.4e-12
4 GO:0007155                  cell adhesion      1518         631       NA       2.3e-12
5 GO:0002376          immune system process      2659        1044       NA       3.6e-12
6 GO:0016043 cellular component organization      6313        2305       NA       5.0e-11
```

This table showcases results from a Gene Ontology enrichment analysis using the topGO method, applied to differentially expressed genes from our dataset. In the `topGO` package, the "classic" Fisher number is the statistical enrichment of GO among differentially expressed genes. It employs Fisher's exact test to calculate the significance of the association between specific GO terms and a given list of genes, helping to identify biological processes, cellular components, or molecular functions that are overrepresented in the dataset.

ClustProfiler:



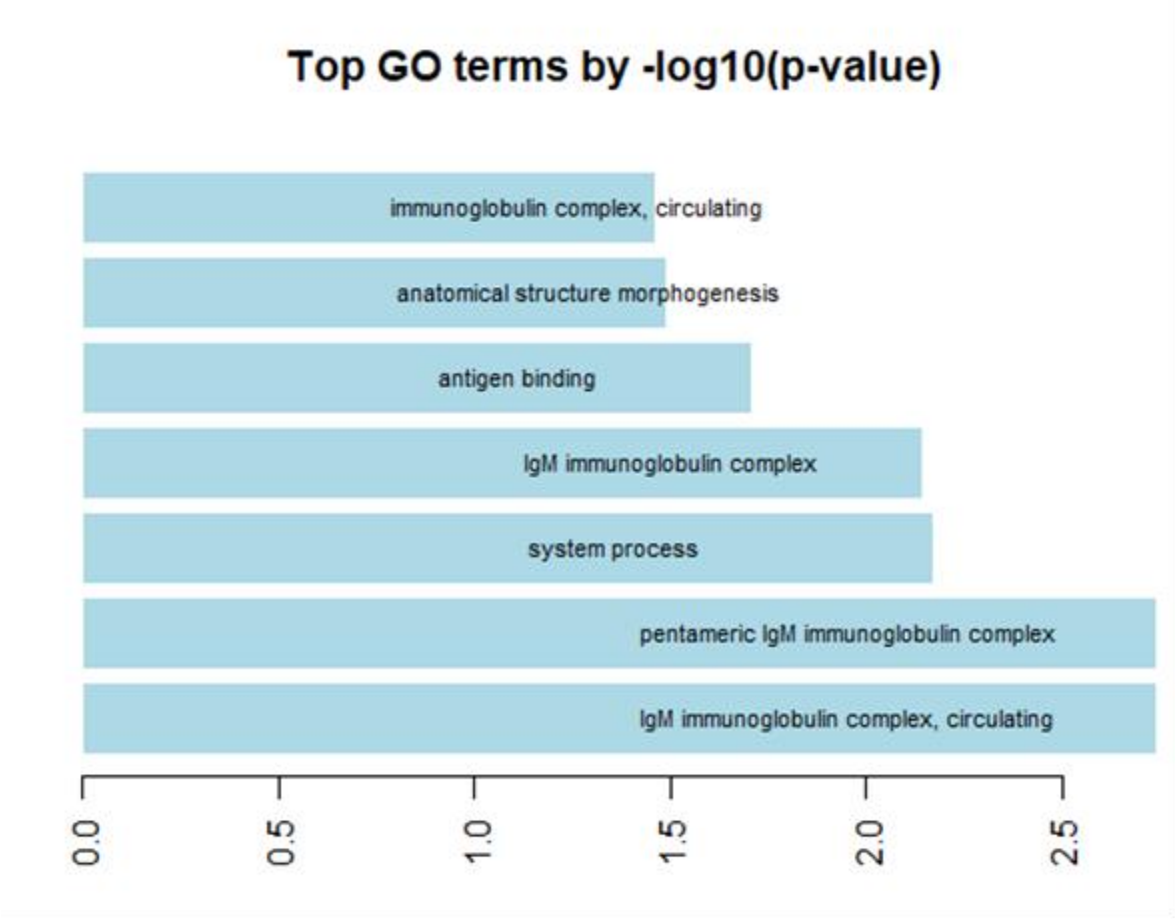
ClusterProfiler was another method we used to perform GO enrichment analysis. Parameters used include pvalueCutoff=0.05 and qvalueCutoff=0.2, to set the significance thresholds for p-values and q-values respectively. This allowed us to filter out genes with less statistical significance. Next, ENSEMBL gene ID were converted to ENTREZID to ensure compatibility with the subsequent analysis. Finally the analysis was performed on the Entrez IDs using the enrichGO function. The antigen binding term has a p.adjust of 0.02435189 and a gene ratio around 0.03, which indicates that this GO term is statistically significant. It is interesting in this case that only one term, antigen binding, was plotted on the chart.



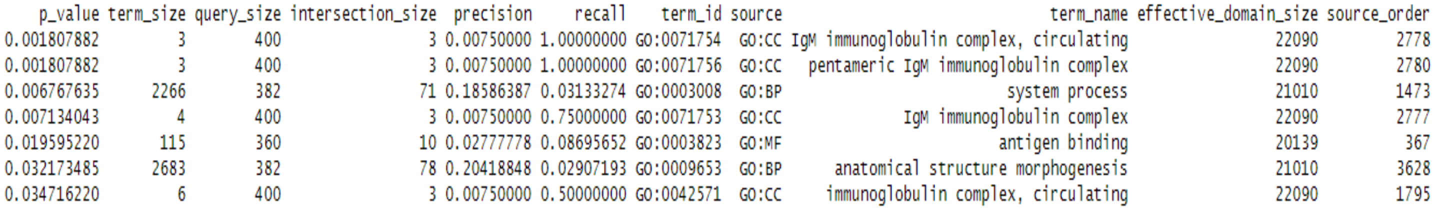
```
                  ID     Description GeneRatio   BgRatio       pvalue   p.adjust    qvalue                                                      geneID Count
GO:0003823 GO:0003823 antigen binding    10/324 118/18369 4.594697e-05 0.02435189 0.0233604 3514/28893/3512/3507/28912/3500/28908/30835/910/2813    10
```

This table showcases results from a Gene Ontology enrichment analysis using the clustProfiler2 method, applied to differentially expressed genes from our dataset. Each listed term is associated with a significant p-value, indicating an over-representation of these biological processes among the DEGs. The p-adjust value is also included which corrects for the issue of how the probability of observing at least one significant result by chance. An interesting result from this table is that only one term weas deemed significant and was included based on the measures that we used in our code (can be seen in the GitHub).

GProfiler2:



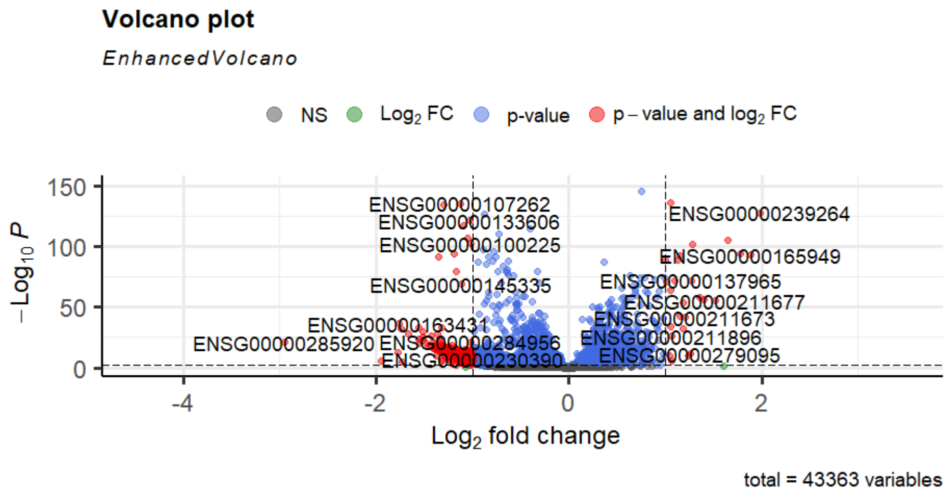
Gprofiler2 is another package/method we used to perform enrichment analysis. The 'gost' function was used to perform gene set enrichment analysis. The parameters of this included DEGs, the specific organism (in this case, 'hsapiens' or humans), and sources=c("GO:BP"), which specifies that we are performing enrichment analysis using gene sets related to the Biological Process category. The top results were selected based on p-values, stored, and then were used to create the horizontal bar chart. Regarding results, we can see that "IgM immunoglobulin complex, circulating" has the highest level of significance. Thus, the genes in our dataset are highly associated with this biological process.



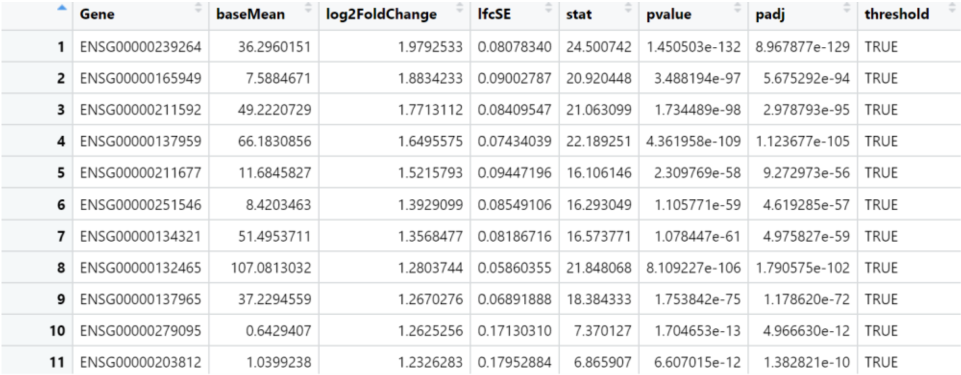
```
    p_value term_size query_size intersection_size  precision     recall    term_id source                              term_name effective_domain_size source_order
0.001807882         3        400                 3 0.00750000 1.00000000 GO:0071754  GO:CC IgM immunoglobulin complex, circulating                 22090         2778
0.001807882         3        400                 3 0.00750000 1.00000000 GO:0071756  GO:CC   pentameric IgM immunoglobulin complex                 22090         2780
0.006767635      2266        382                71 0.18586387 0.03133274 GO:0003008  GO:BP                          system process                 21010         1473
0.007134043         4        400                 3 0.00750000 0.75000000 GO:0071753  GO:CC              IgM immunoglobulin complex                 22090         2777
0.019595220       115        360                10 0.02777778 0.08695652 GO:0003823  GO:MF                          antigen binding                 20139          367
0.032173485      2683        382                78 0.20418848 0.02907193 GO:0009653  GO:BP      anatomical structure morphogenesis                 21010         3628
0.034716220         6        400                 3 0.00750000 0.50000000 GO:0042571  GO:CC     immunoglobulin complex, circulating                 22090         1795
```

This table showcases results from a Gene Ontology enrichment analysis using the gProfiler2 method, applied to differentially expressed genes (DEGs) from our dataset. Each listed GO term is associated with a significant p-value, indicating an over-representation of these biological processes among the DEGs. The term size and intersection size columns provide context on the prevalence of each term within the dataset and the reference genome, respectively.

Differential Analysis Volcano Plot:

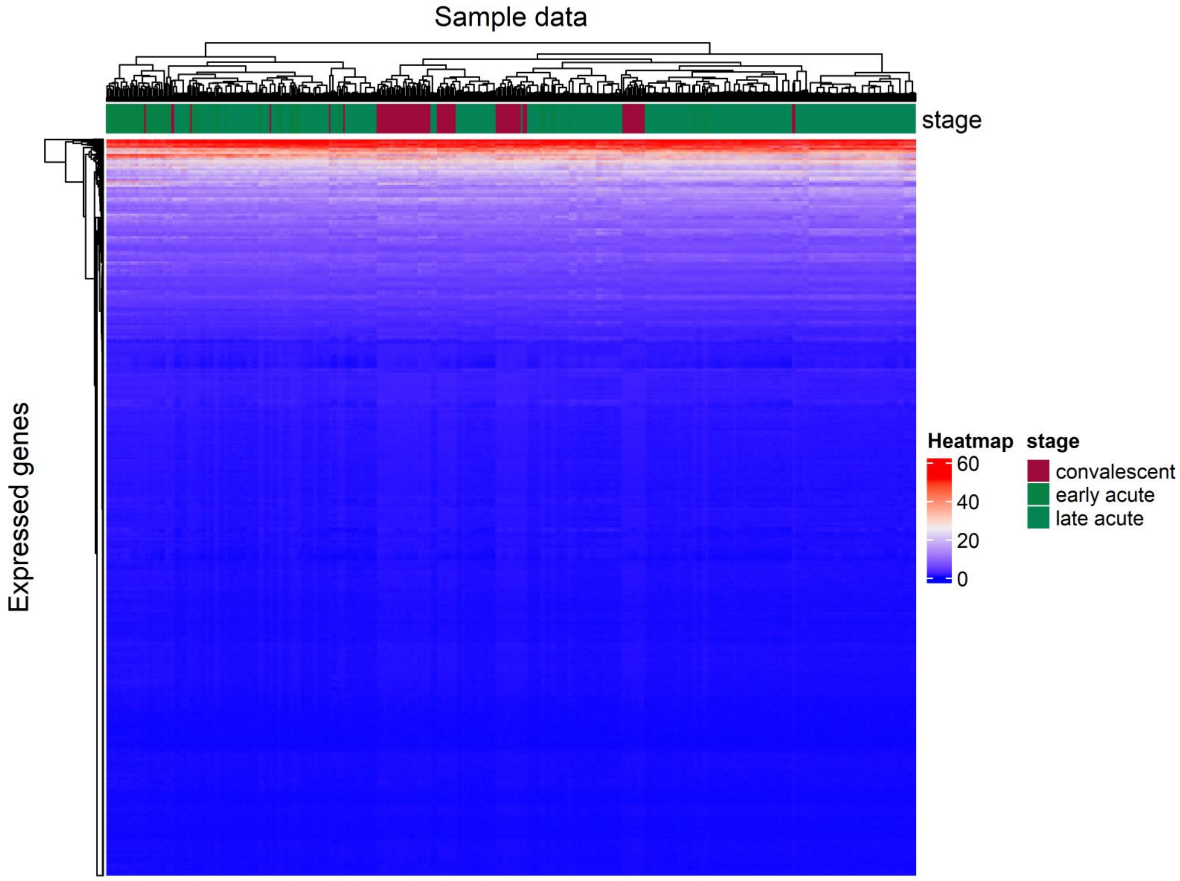


We generated this volcano plot by first removing the genes that had a total read count of less than 10, generating a dataset based on differential expression analysis much like in the last plot, and used the EnhancedVolcano function with an adjusted p-value cutoff of 0.05. Most of the genes in the dataset are not significant and most of the significant genes in the dataset do not meet the log 2-fold change threshold. The densest cluster of significant genes that meet the log2 fold change threshold have decreased expression. Despite this, we found a very large number of significant genes. Thus, this table will only include the most differentially expressed genes, while the full list can be found here.



	Gene	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	threshold
1	ENSG00000239264	36.2960151	1.9792533	0.08078340	24.500742	1.450503e-132	8.967877e-129	TRUE
2	ENSG00000165949	7.5884671	1.8834233	0.09002787	20.920448	3.488194e-97	5.675292e-94	TRUE
3	ENSG00000211592	49.2220729	1.7713112	0.08409547	21.063099	1.734489e-98	2.978793e-95	TRUE
4	ENSG00000137959	66.1830856	1.6495575	0.07434039	22.189251	4.361958e-109	1.123677e-105	TRUE
5	ENSG00000211677	11.6845827	1.5215793	0.09447196	16.106146	2.309769e-58	9.272973e-56	TRUE
6	ENSG00000251546	8.4203463	1.3929099	0.08549106	16.293049	1.105771e-59	4.619285e-57	TRUE
7	ENSG00000134321	51.4953711	1.3568477	0.08186716	16.573771	1.078447e-61	4.975827e-59	TRUE
8	ENSG00000132465	107.0813032	1.2803744	0.05860355	21.848068	8.109227e-106	1.790575e-102	TRUE
9	ENSG00000137965	37.2294559	1.2670276	0.06891888	18.384333	1.753842e-75	1.178620e-72	TRUE
10	ENSG00000279095	0.6429407	1.2625256	0.17130310	7.370127	1.704653e-13	4.966630e-12	TRUE
11	ENSG00000203812	1.0399238	1.2326283	0.17952884	6.865907	6.607015e-12	1.382821e-10	TRUE

Differential Analysis Heatmap:

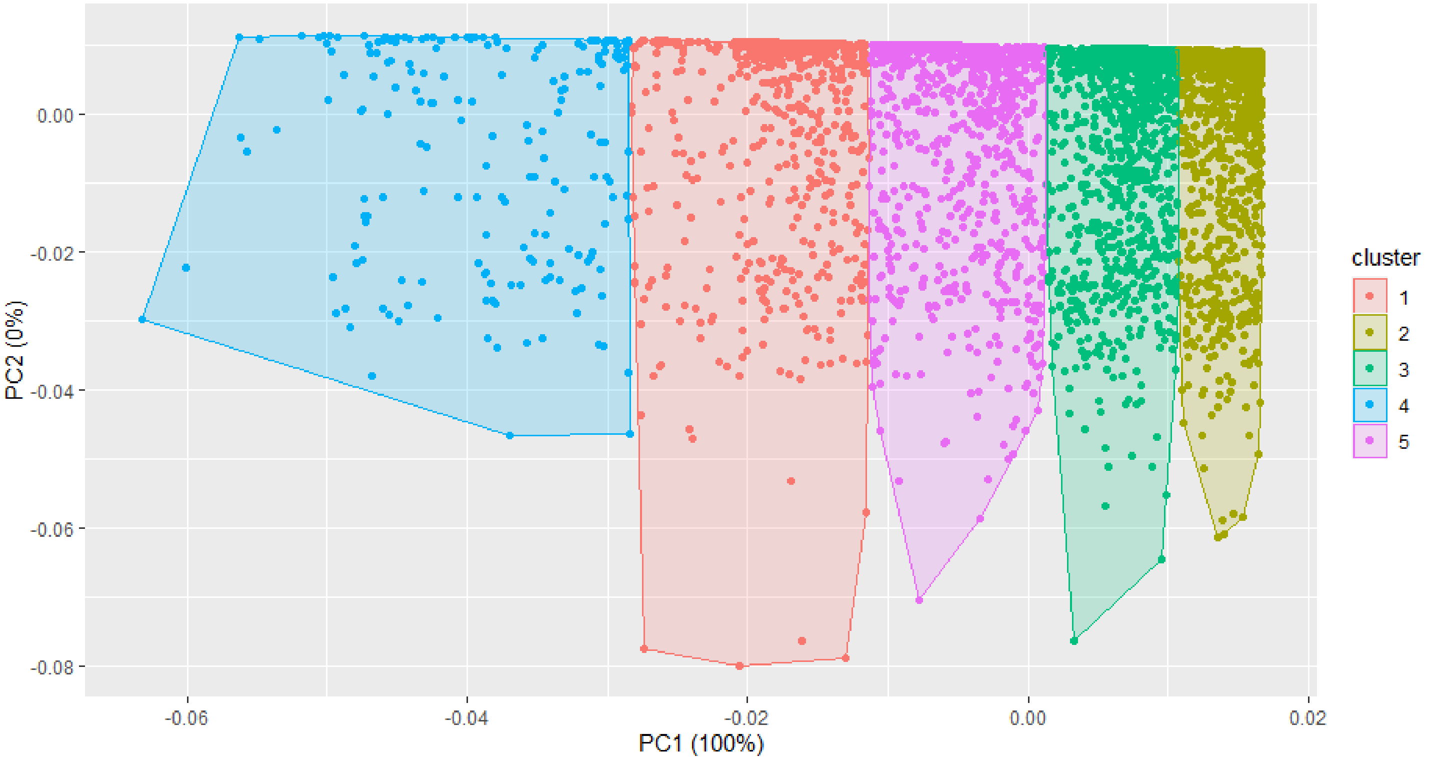


This heatmap uses the ComplexHeatmap package to display expression profiles of certain genes across different samples. we generated it by extracting the list of significant genes used in the volcano plot, joining it with the raw expression matrix, removing superfluous columns, and using the ComplexHeatmap function. One interesting result is that although there are many significant genes, a very small number of those genes are expressed much more than the other significant genes across the early acute and late acute stages of the disease. For these genes, it's difficult to observe differences based on disease stage. We also see slightly different patterns in the heatmap for convalescent patients and patients who are in the other stages of the disease, supporting our hypothesis that Zika infection influences gene expression.

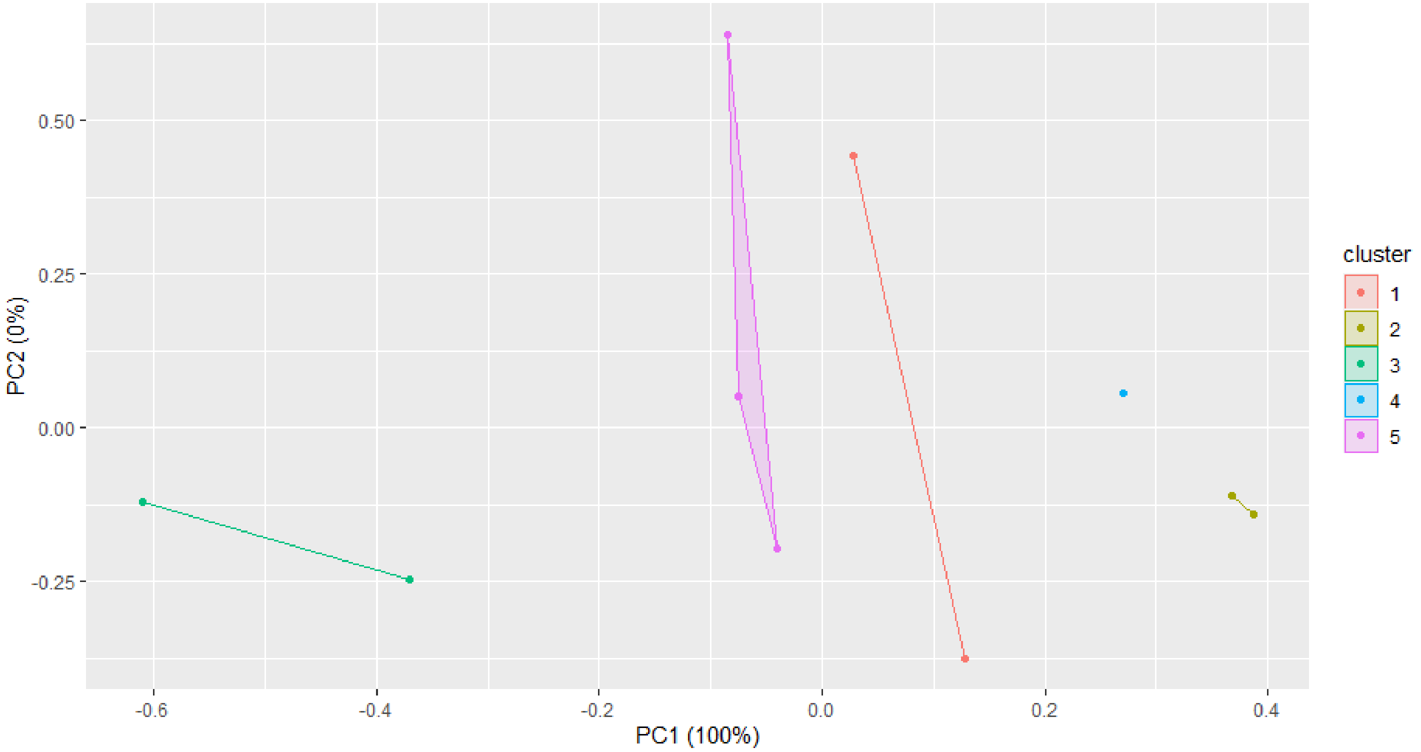
Clustering Algorithms:

K-means: Below are the plots for the K-means clustering algorithm. They were created by using the kmeans() function after cleaning the data to make sure it was entirely numeric. Here we provided 5 as my number of desired clusters and after performing this with 5000 genes, repeated this for 10, 100, 1000, and 10000. When we decreased the number of clusters, each cluster became larger and more genes were encompassed in each cluster. Rerunning the k-means clustering algorithm with fewer genes produced different and smaller clusters since fewer genes results in fewer data points for the clustering algorithm to work with.

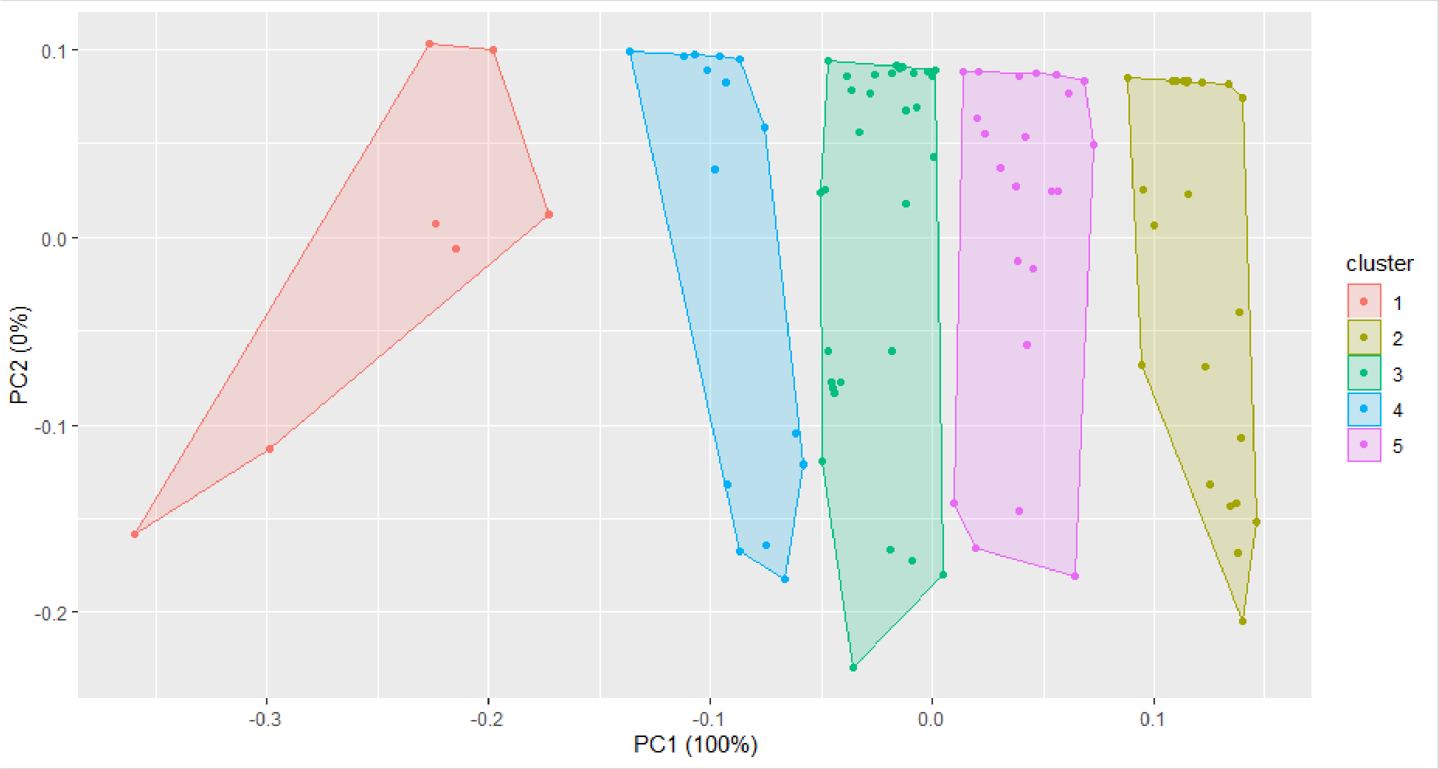
5000:


0.00
-0.02
-0.04
-0.06
-0.08
PC2 (0%)
-0.06
-0.04
-0.02
0.00
0.02
PC1 (100%)
cluster
1
2
3
4
5

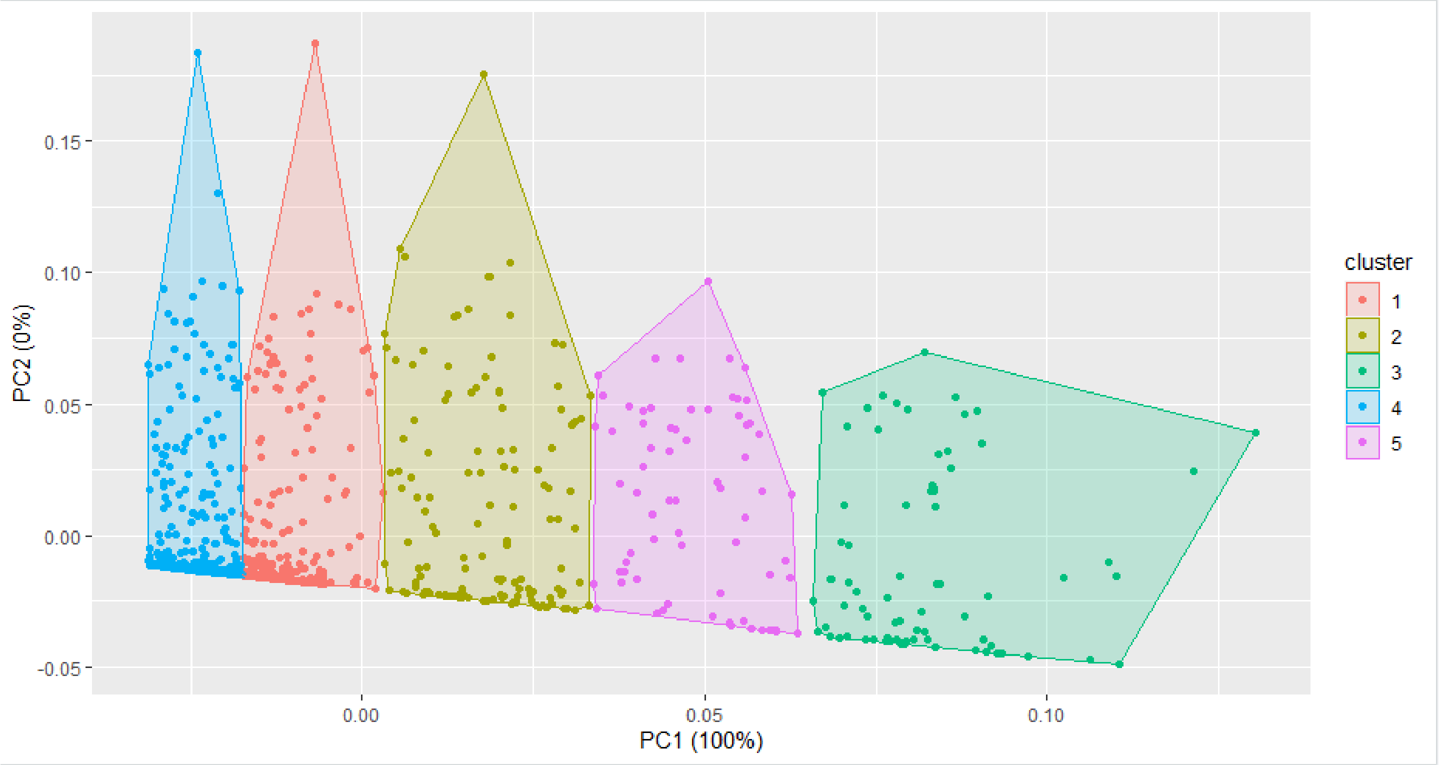
10:


0.50
0.25
0.00
-0.25
PC2 (0%)
-0.6
-0.4
-0.2
0.0
0.2
0.4
PC1 (100%)
cluster
1
2
3
4
5

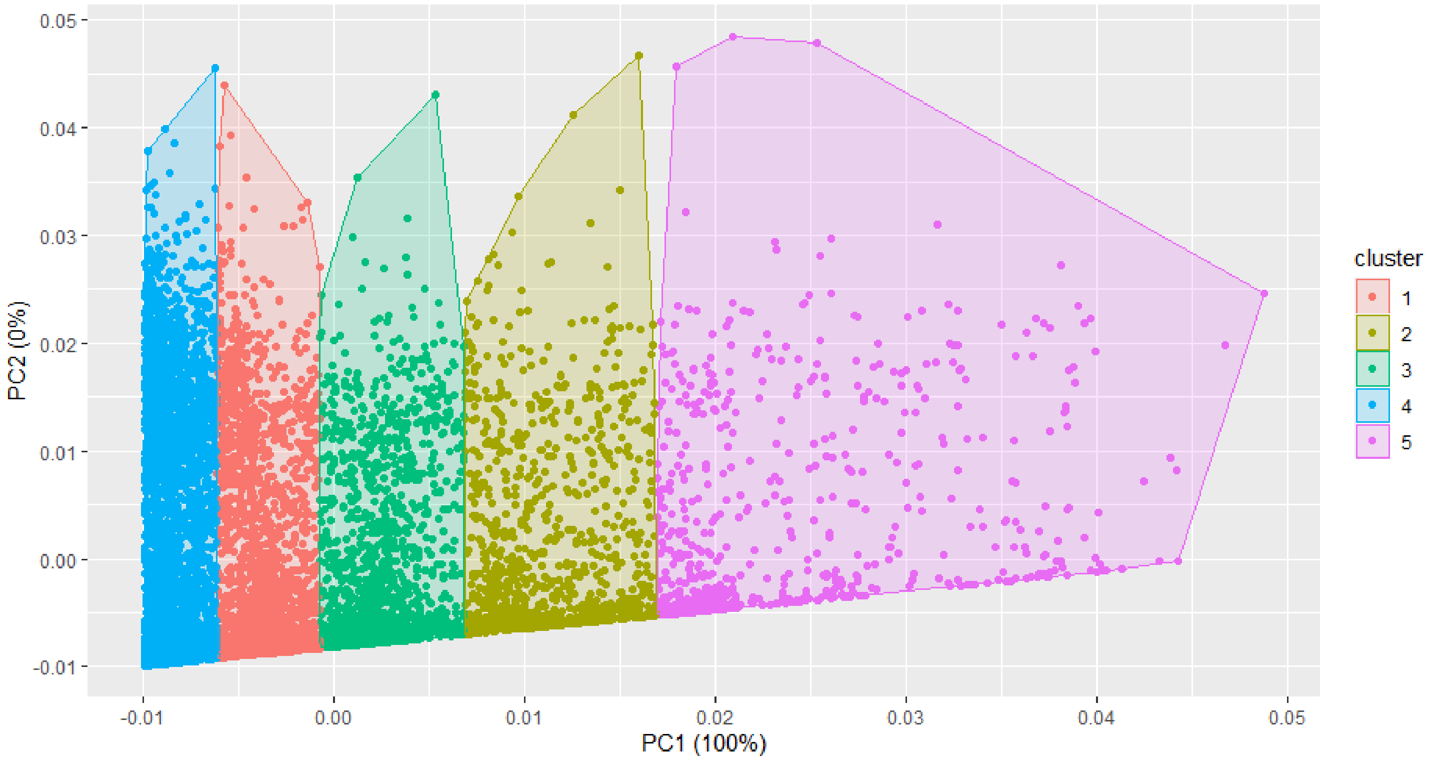
100:


0.1
0.0
-0.1
-0.2
PC2 (0%)
-0.3
-0.2
-0.1
0.0
0.1
PC1 (100%)
cluster
1
2
3
4
5

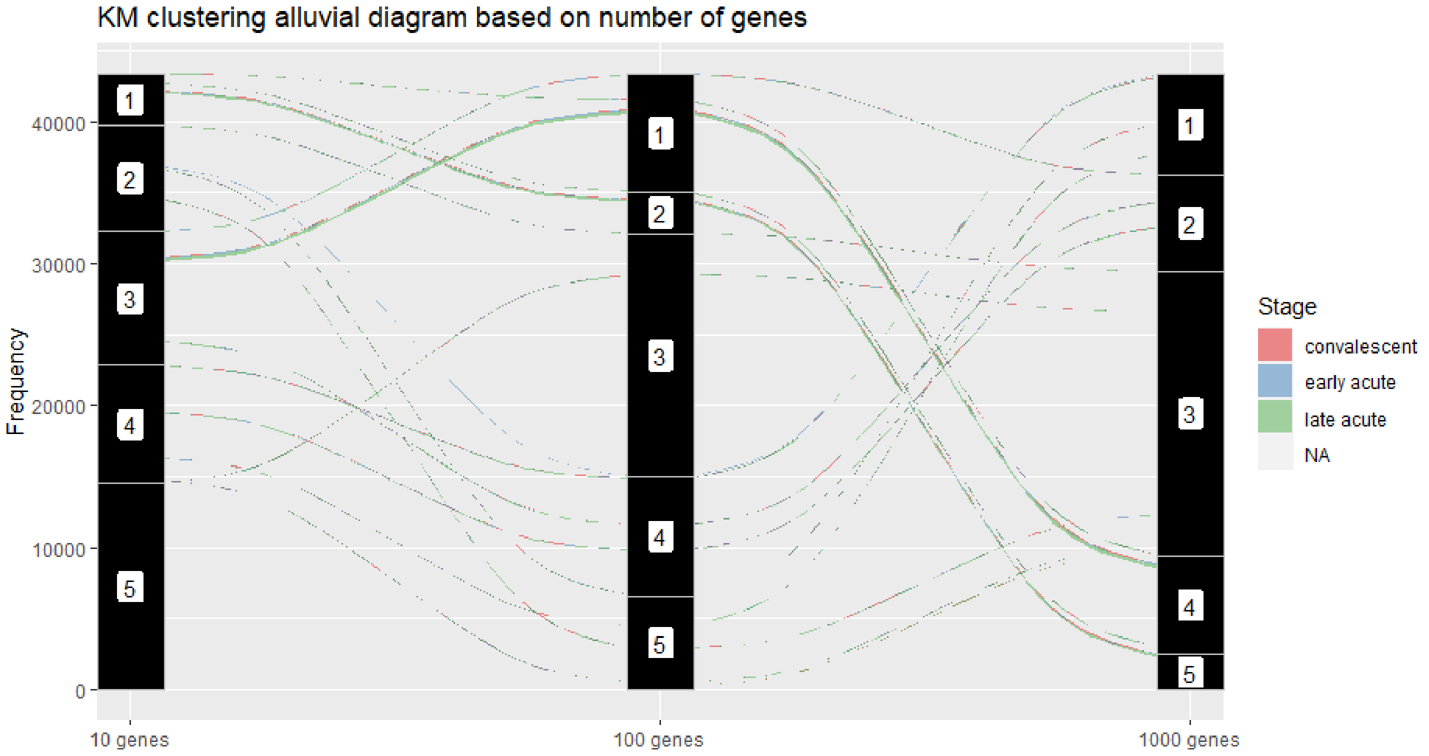
1000:


0.15
0.10
0.05
0.00
-0.05
PC2 (0%)
0.00
0.05
0.10
PC1 (100%)
cluster
1
2
3
4
5

10000:

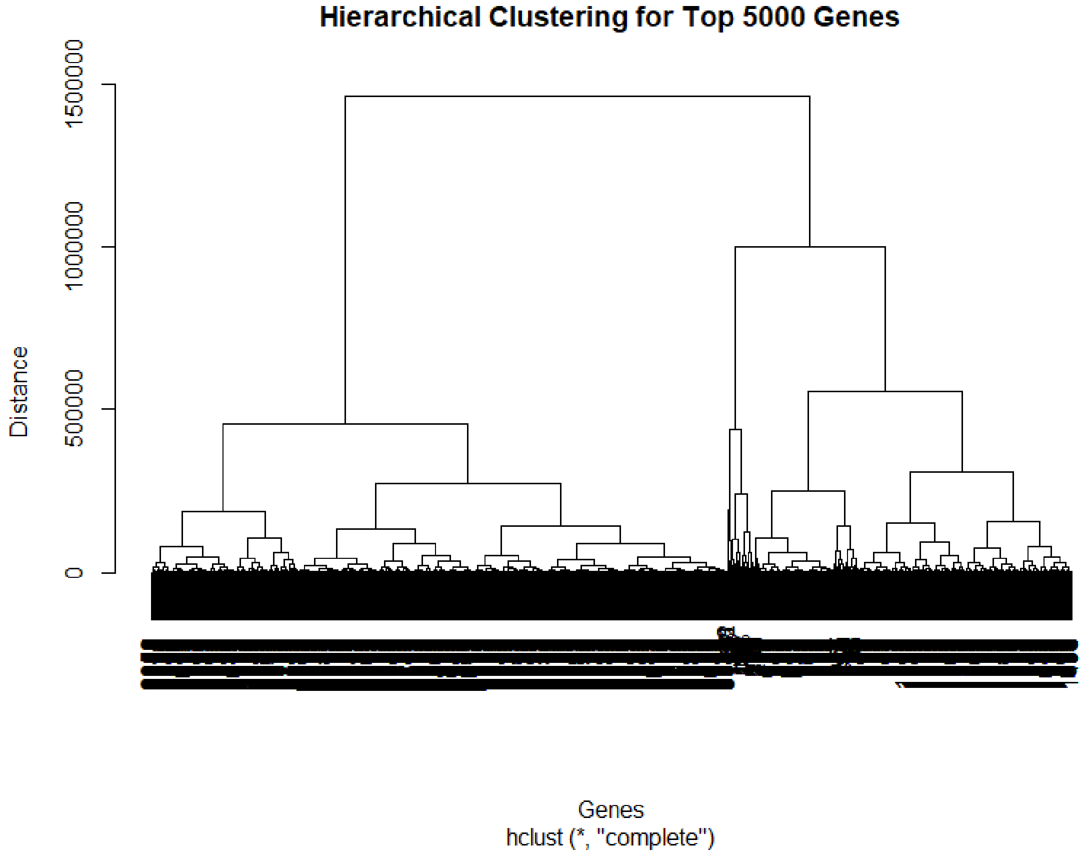


The below alluvial diagram was generated to visualize the clustering setups with varying gene numbers: 10, 100, and 1000. Clustering results were stored in separate data frames, while the stage metadata was collected. The diagram shows the distribution of samples across different stages for each clustering setup, which is intended to help gain insights into the algorithm's performance and how it relates to the metadata (particularly, the corresponding stages).

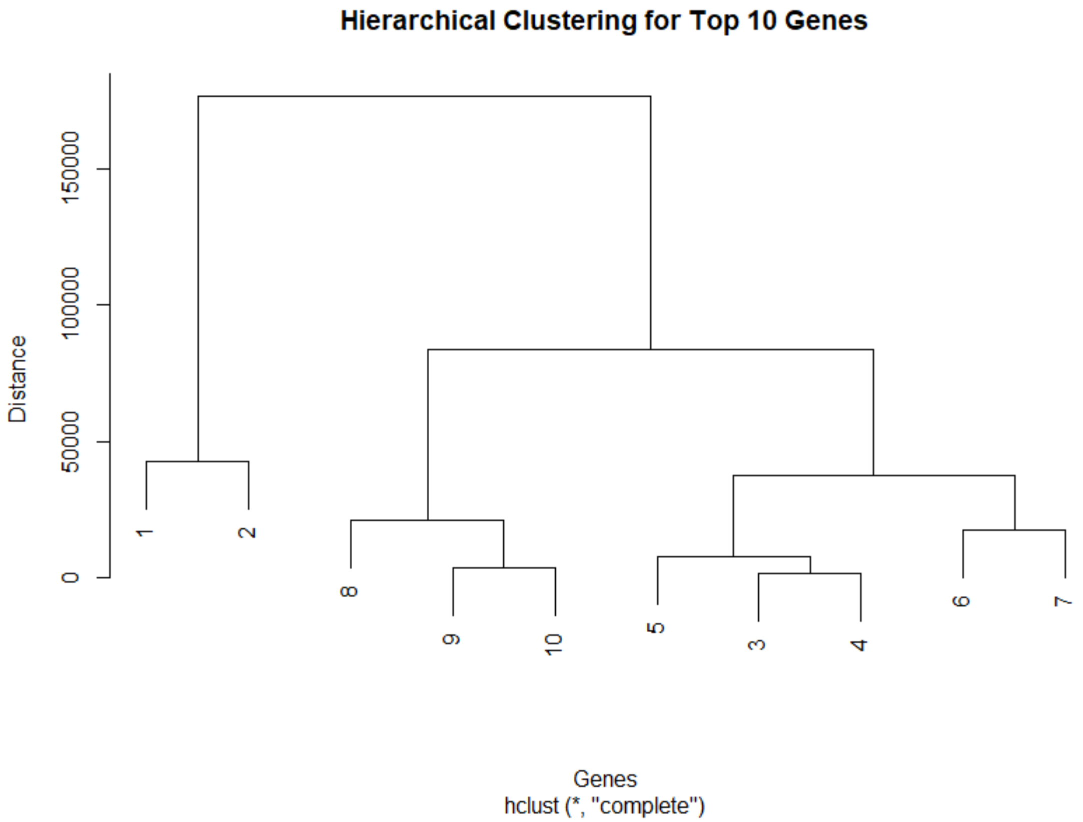


Hierarchical clustering (hclust): Below are the plots for the hierarchical clustering. We ran the analysis with 12 clusters because there were three sample groups, and we wanted a multiple of three, but we also wanted to capture subgroups within the sample groups. First, the Euclidean distance between genes was calculated and put into a matrix. Then, using the hclust() function with the complete method, we were able to perform hierarchical clustering and create a tree based on the distances. Finally, it was plotted for 5000, 10, 100, 1000, and 10000.

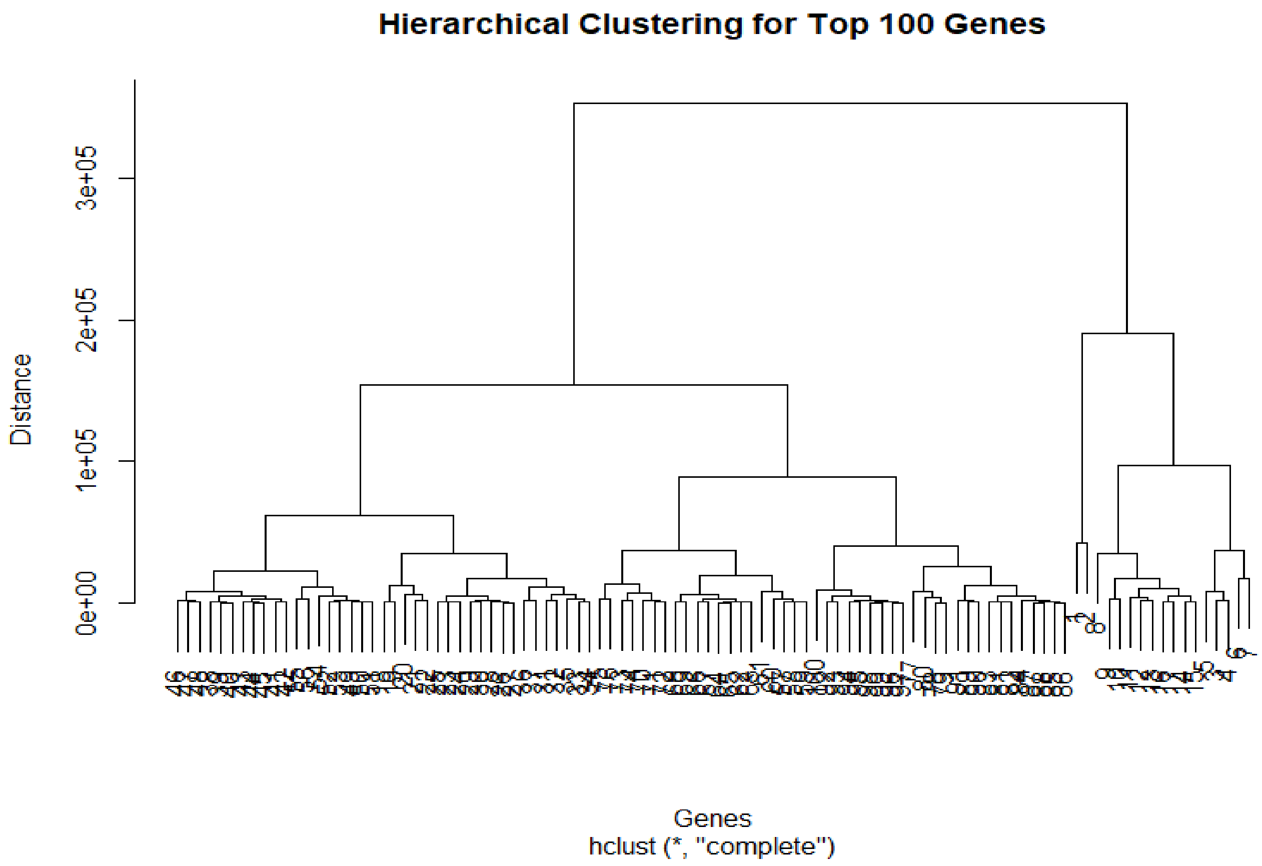
5000:



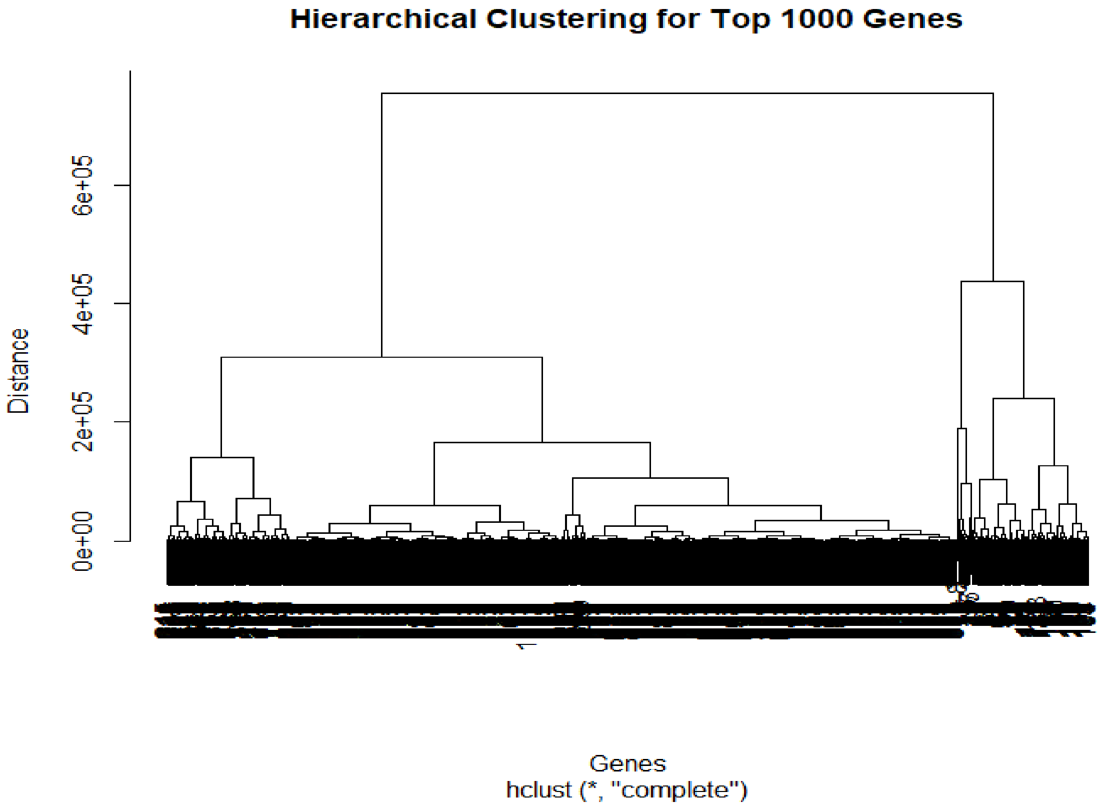
10:


Hierarchical Clustering for Top 10 Genes
Distance
0
50000
100000
150000
1
2
8
9
10
5
3
4
6
7
Genes
hclust (*, "complete")

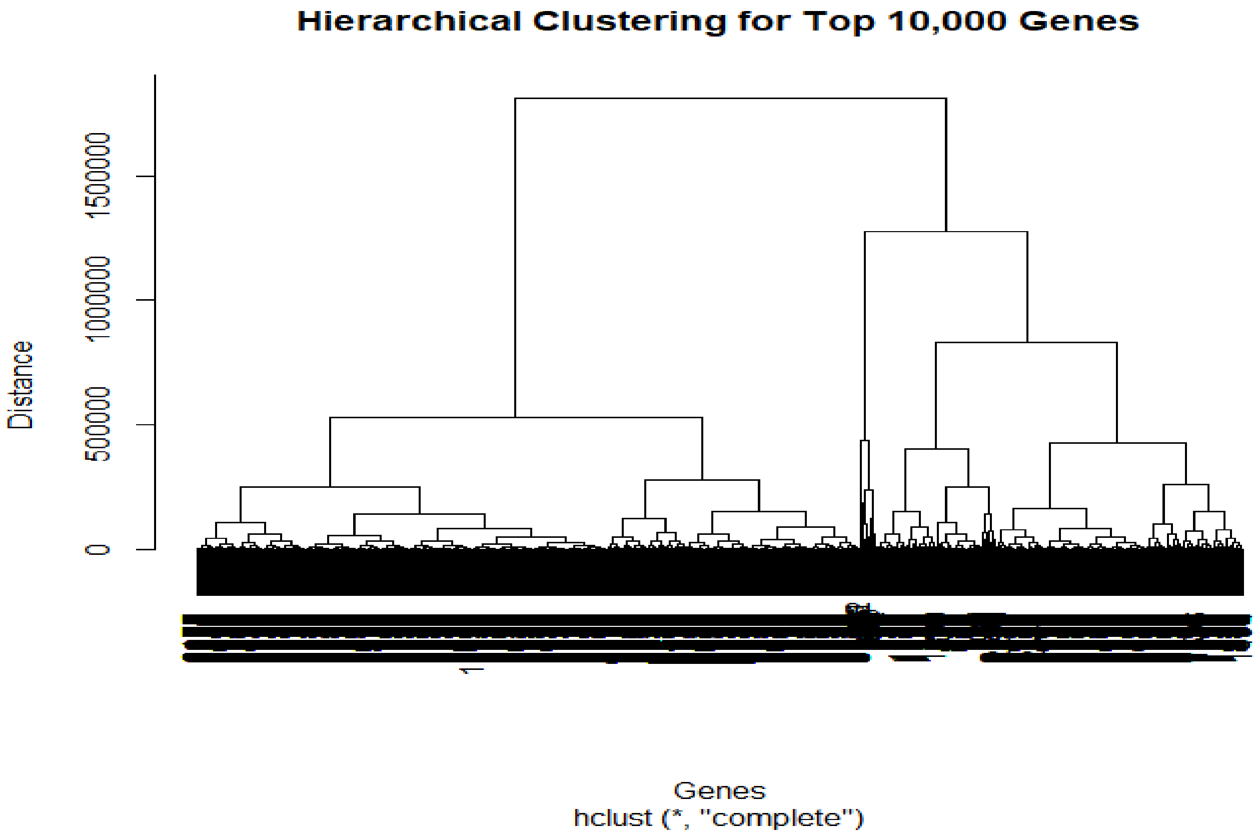
100:


Hierarchical Clustering for Top 100 Genes
Distance
3e+05
2e+05
1e+05
0e+00
Genes
hclust (*, "complete")

1,000:

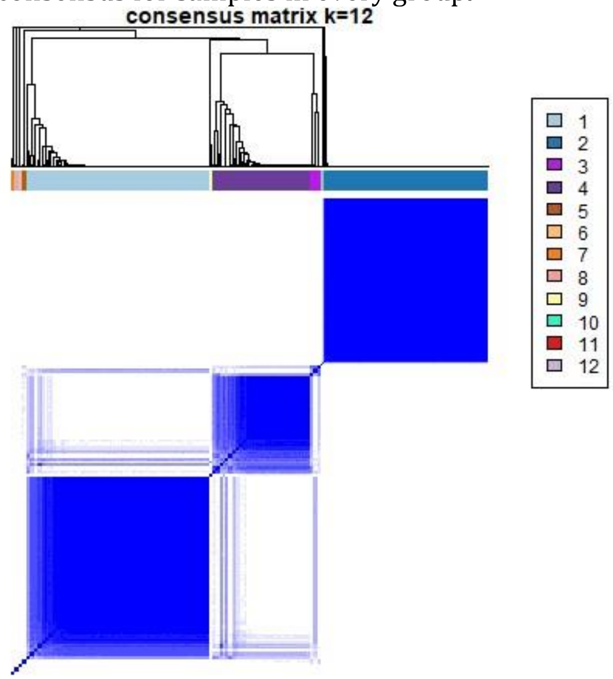

Hierarchical Clustering for Top 1000 Genes
Distance
0e+00
2e+05
4e+05
6e+05
Genes
hclust (*, "complete")

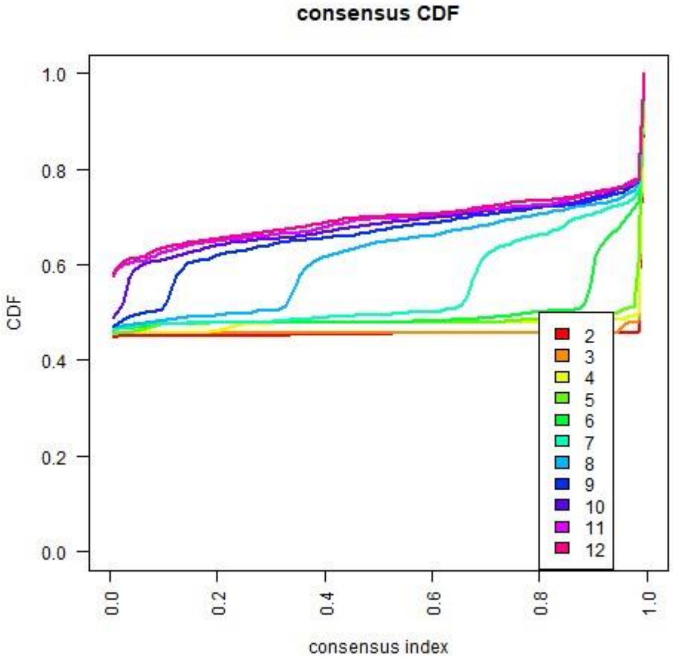
10,000:

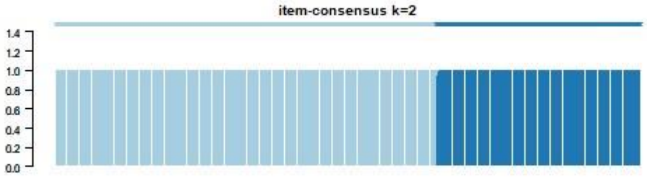


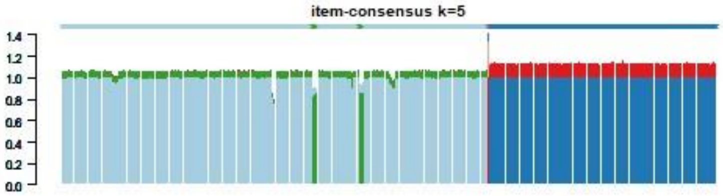
Consensus Cluster Plus (5000 genes):
To generate this plot, we first calculated the variance of each gene in the sample, sorted them in descending order of variance, and removed the top x genes for use in generating the plot. Before analysis, we called the inbuilt sweep function to normalize the data. we chose to run the analysis with 12 clusters because there were three sample

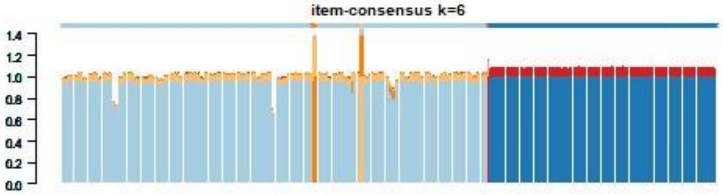
groups, and we wanted a multiple of three, but we also wanted to capture subgroups within the sample groups. Interestingly, the optimal number of clusters was two. Adding more clusters weakened the “stairstep pattern” of the CDF plot and reduced the item consensus for samples in every group.

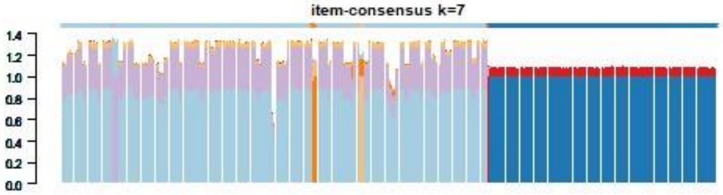


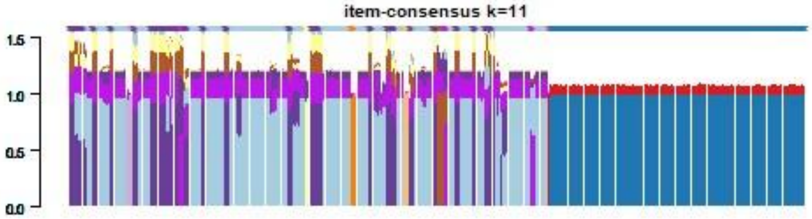


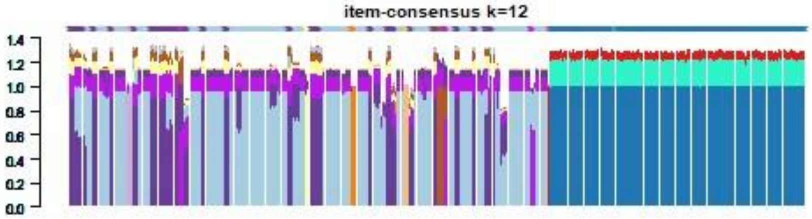




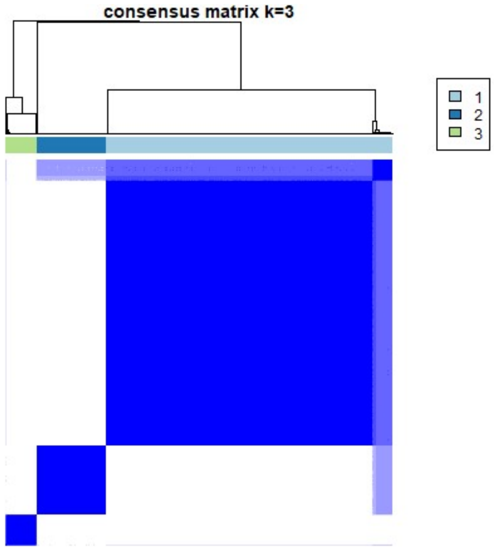


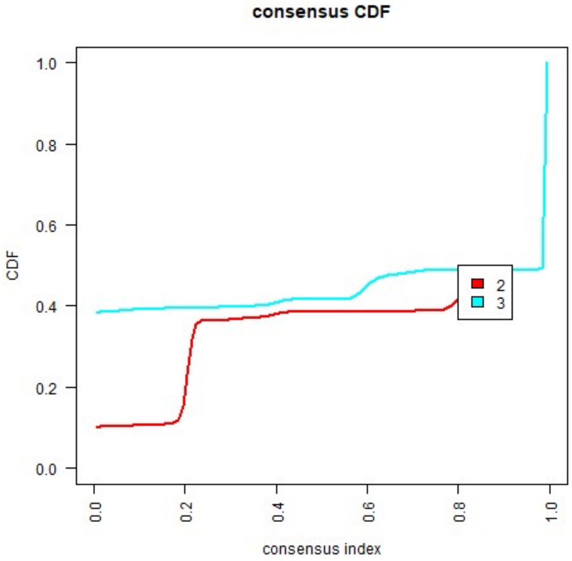




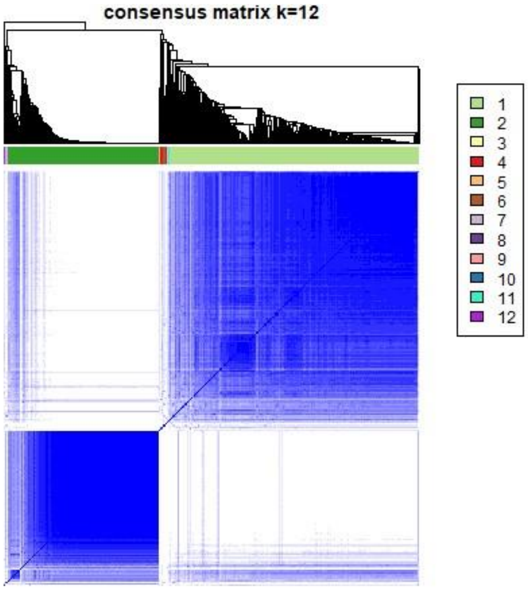


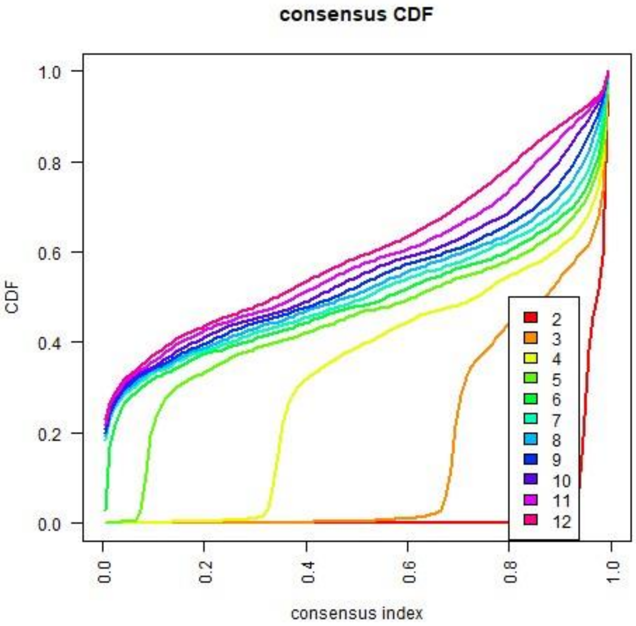
Top 10 (Ran with 3 clusters):


consensus matrix k=3
1
2
3

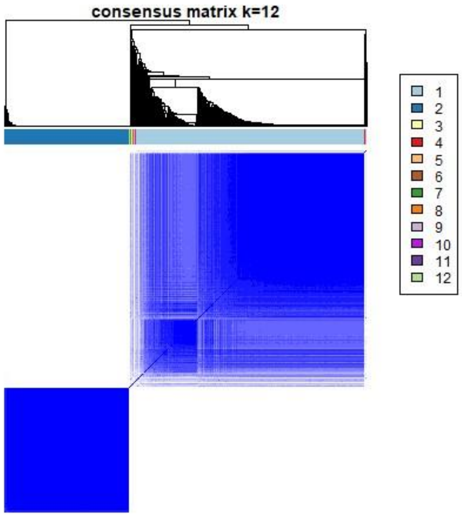

consensus CDF
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
consensus index
2
3

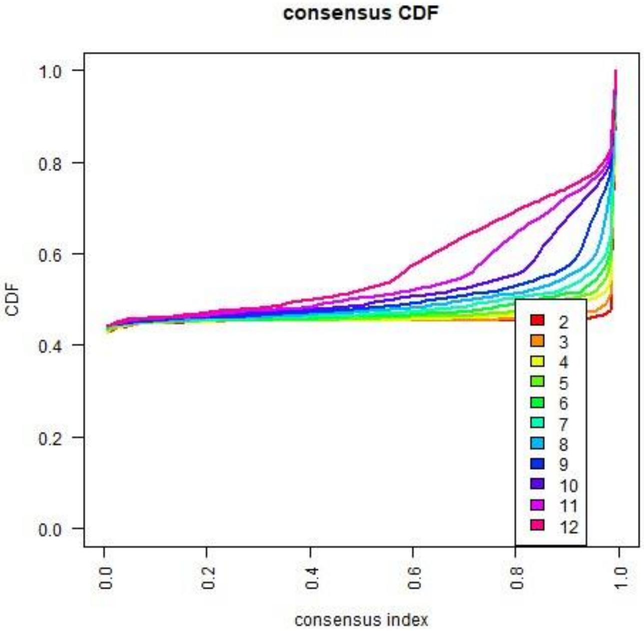
Top 100:


consensus matrix k=12
1
2
3
4
5
6
7
8
9
10
11
12

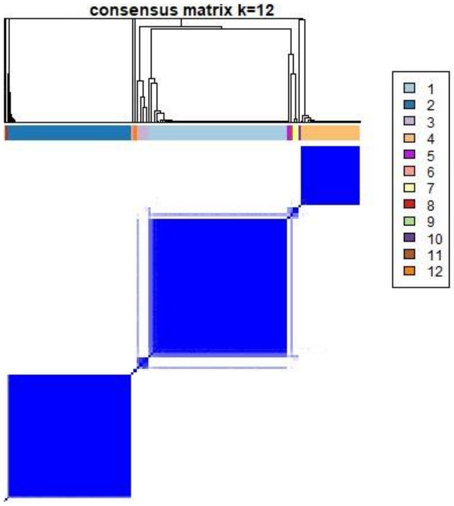

consensus CDF
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
consensus index
2
3
4
5
6
7
8
9
10
11
12

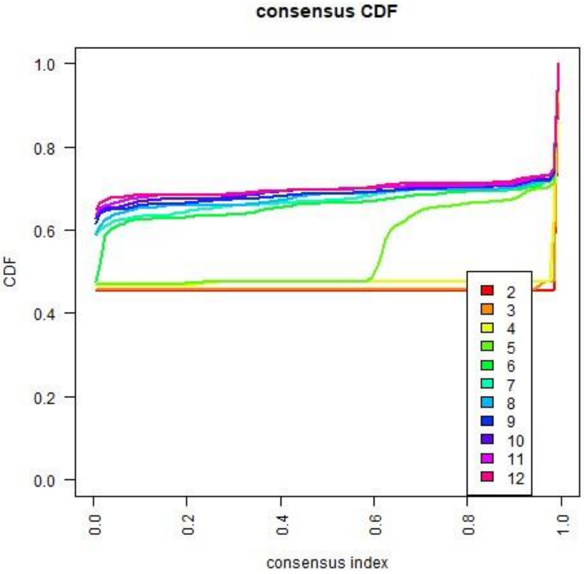
Top 1000


consensus matrix k=12
1
2
3
4
5
6
7
8
9
10
11
12


consensus CDF
CDF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
consensus index
2
3
4
5
6
7
8
9
10
11
12

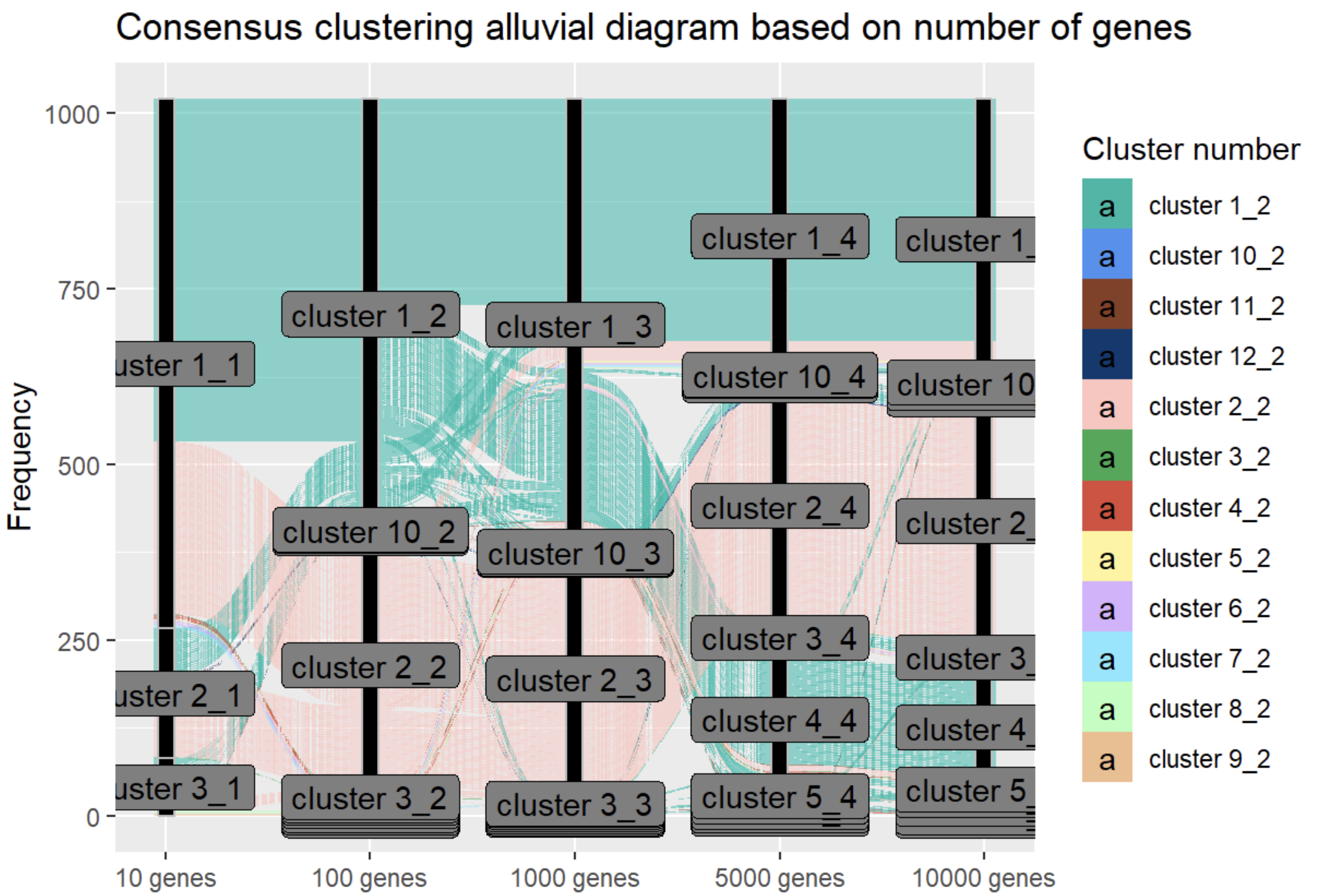
Top 10000





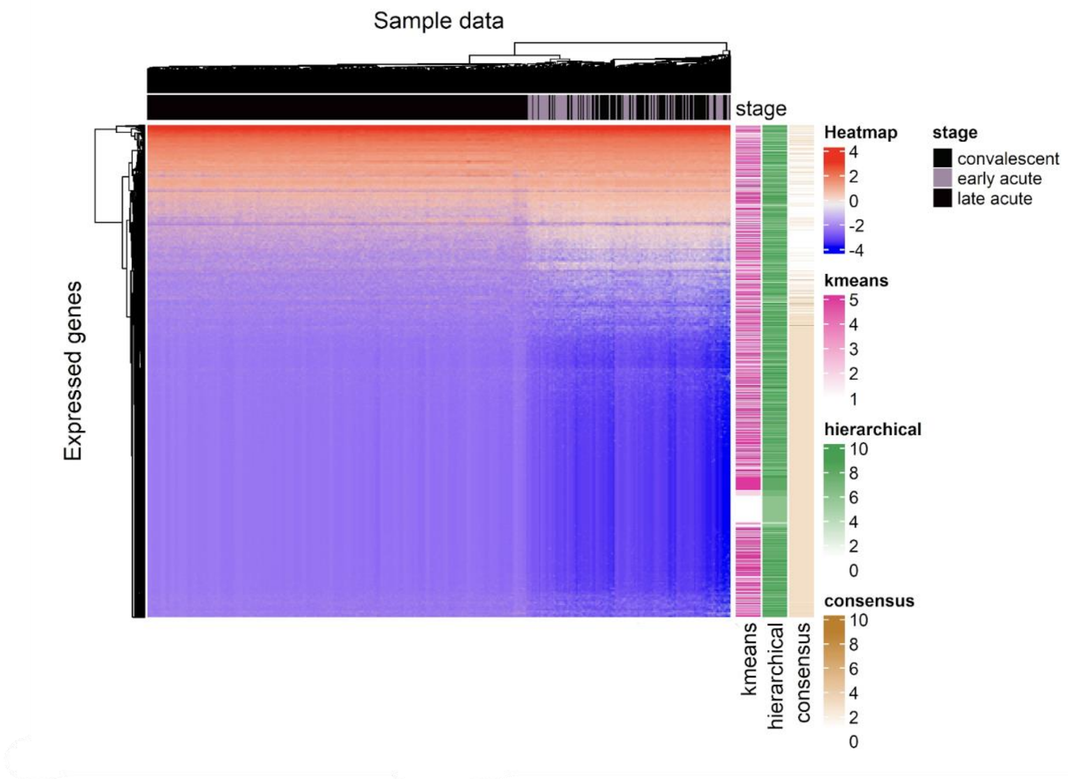
Analysis: Increasing the number of genes did not change the optimal number of clusters (2), but it did generate a flatter consensus CDF, which indicates a higher degree of certainty.

Alluvial diagram



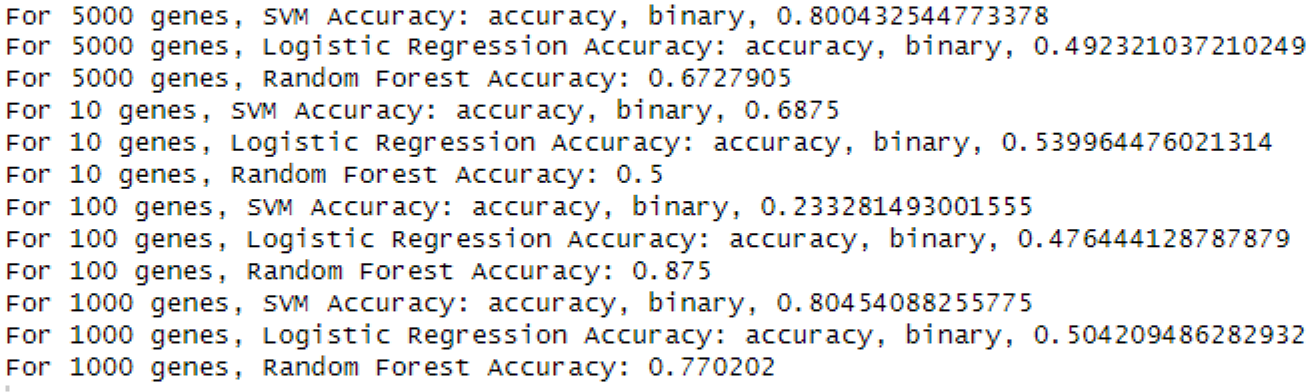
To generate this plot, we first calculated the frequency table of each cluster with the table() function in R. We then combined the frequency tables of each clustering into a single table where rows correspond to genes and columns correspond to different clustering runs. After that, we added a single frequency column where each cell was 1. We used each column of this combined matrix as a single axis in the alluvial plot extension of ggplot.

Clustering Heatmap:

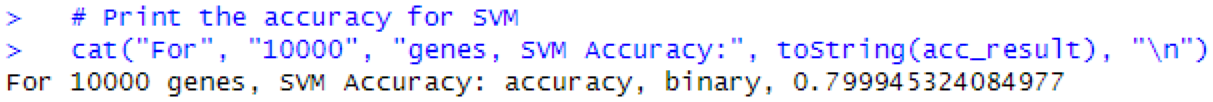


To generate this heatmap, we first used the cutree method to convert the hierarchical cluster tree to the equivalent of ten clusters. We then used the returned object in addition to the data vectors of the other clustering methods to produce a rowAnnotation, which contains all three sidebars. The Heatmap method was used to ultimately draw the heatmap.

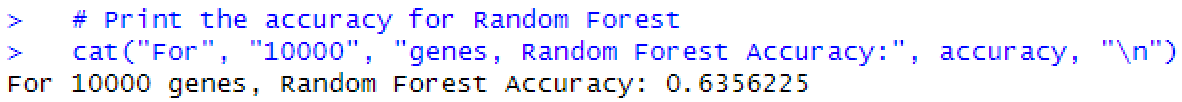
Predictive Methods:



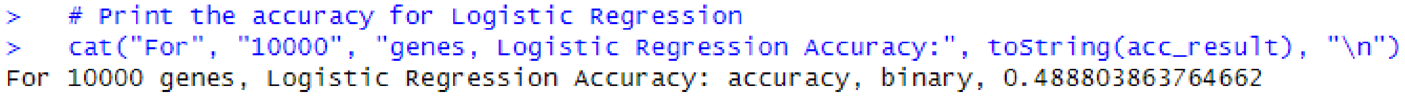
```
For 5000 genes, SVM Accuracy: accuracy, binary, 0.800432544773378
For 5000 genes, Logistic Regression Accuracy: accuracy, binary, 0.492321037210249
For 5000 genes, Random Forest Accuracy: 0.6727905
For 10 genes, SVM Accuracy: accuracy, binary, 0.6875
For 10 genes, Logistic Regression Accuracy: accuracy, binary, 0.539964476021314
For 10 genes, Random Forest Accuracy: 0.5
For 100 genes, SVM Accuracy: accuracy, binary, 0.233281493001555
For 100 genes, Logistic Regression Accuracy: accuracy, binary, 0.476444128787879
For 100 genes, Random Forest Accuracy: 0.875
For 1000 genes, SVM Accuracy: accuracy, binary, 0.80454088255775
For 1000 genes, Logistic Regression Accuracy: accuracy, binary, 0.504209486282932
For 1000 genes, Random Forest Accuracy: 0.770202
```



```
>   # Print the accuracy for SVM
>   cat("For", "10000", "genes, SVM Accuracy:", toString(acc_result), "\n")
For 10000 genes, SVM Accuracy: accuracy, binary, 0.799945324084977
```



```
>   # Print the accuracy for Random Forest
>   cat("For", "10000", "genes, Random Forest Accuracy:", accuracy, "\n")
For 10000 genes, Random Forest Accuracy: 0.6356225
```



```
>   # Print the accuracy for Logistic Regression
>   cat("For", "10000", "genes, Logistic Regression Accuracy:", toString(acc_result), "\n")
For 10000 genes, Logistic Regression Accuracy: accuracy, binary, 0.488803863764662
```

Support vector machine

Overall, this seems to be the most accurate, around 80% accuracy.

Logistic regression

Least accurate, just around 50% accuracy. This is not much better than predicting at random as there are only two options to choose from (late acute vs. early acute).

Random forest

2nd most accurate. Around 65% accuracy. Not that great but some use here could be found. Much faster than Support vector machine.

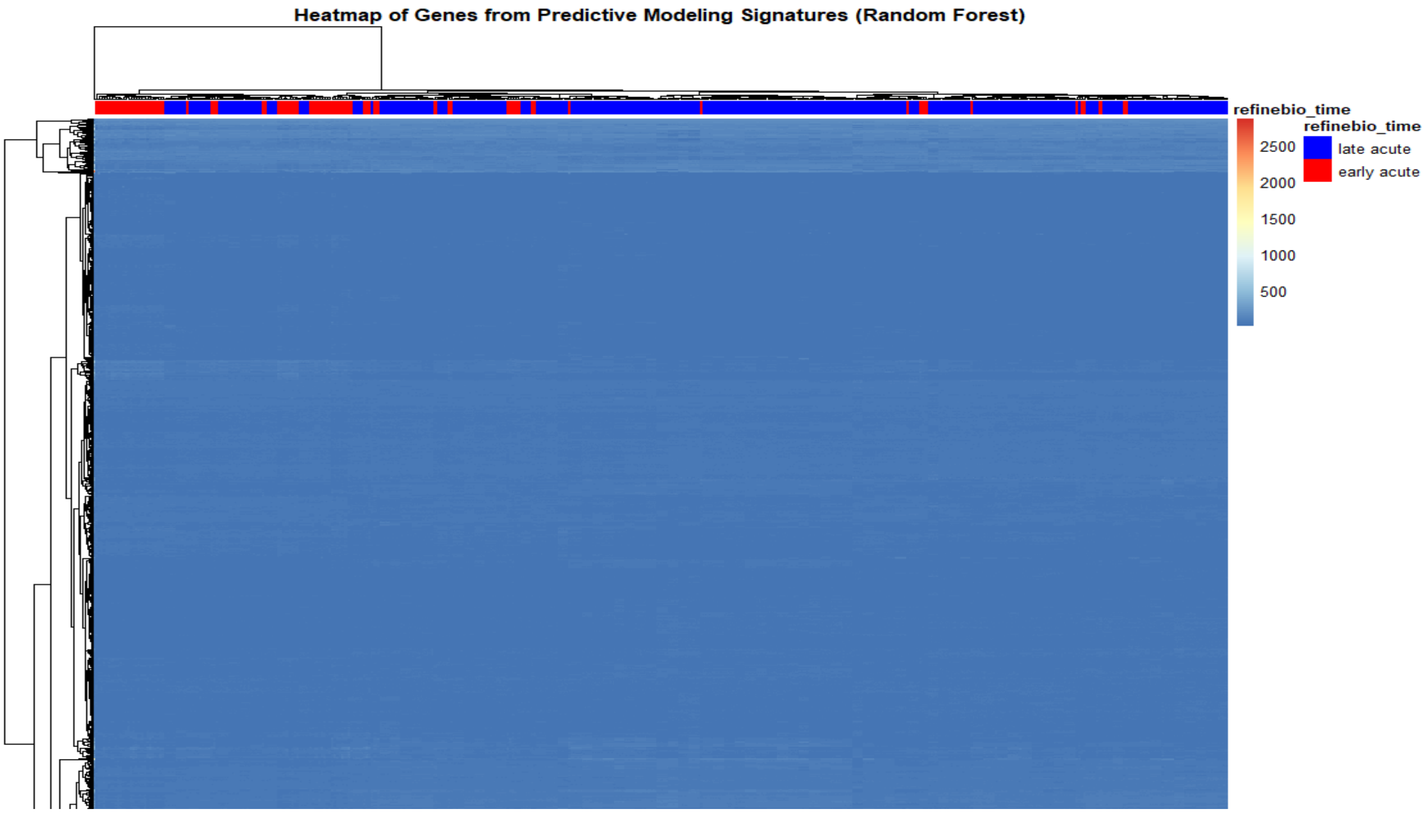
How did the number of genes affect the results?

*For the **support vector machine** method, the more genes used to train the model, the accuracy seemed to progressively increase except for at 100 genes. Something unknown occurred here where the accuracy shot down to about 23%. For **logistic regression**, there seemed to be no dependency on how many genes were used. Perhaps this method for modeling did not work well with our specific Zika data. The accuracy was always around 50%.*

*For the **random forest** method, we reached peak accuracy at 100 genes, then the accuracy seemed to dip again. Overall the model seemed to settle somewhere around the mid-60s in terms of accuracy.*

Are the same genes being included in the models? How much overlap is there between the signatures?

Yes, it uses the same training set for each model (SVM, Logistic Regression, and Random Forest). That being said, the exact genes that are selected or given importance in the model differ due to the differences in the algorithms and how they select features. Extracting feature importance or coefficients from each trained model and comparing the lists of genes, you can see that the genes are still basically the same.

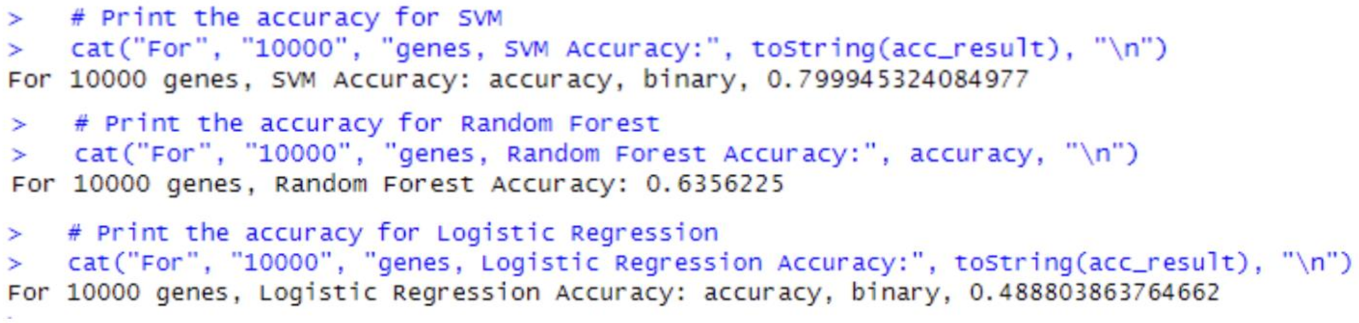


In the heatmap we created, the genes identified were visualized across samples classified as either "late acute" or "early acute"'. The heatmap was generated using the pheatmap package in R, with samples as columns and genes as rows. An annotation sidebar was added to indicate the sample groups, with 'late acute' samples colored in blue and 'early acute' samples colored in red. The top part of the heatmap showed more red (indicating 'early acute' samples) on the left and gradually transitioned to more blue ('late acute' samples) on the right. In contrast, the middle of the heatmap was predominantly blue. This implies that these genes are highly expressed in "late acute" samples.

All code and analysis for these methods are available at the team's GitHub repository:
https://github.com/Donovan55/BioinformaticsProject

4. Results & Discussion

We were able to partially answer our research question, as each of our models outperformed random chance. The support vector machine was the most accurate, with 80% accuracy. Logistic and Random Forest models yielded 50% and 65% accuracy, respectively. Although these models are not accurate enough to be clinically useful, a much more accurate model could be trained if the convalescent and late acute stages were merged.



```
>   # Print the accuracy for SVM
>   cat("For", "10000", "genes, SVM Accuracy:", toString(acc_result), "\n")
For 10000 genes, SVM Accuracy: accuracy, binary, 0.799945324084977
>   # Print the accuracy for Random Forest
>   cat("For", "10000", "genes, Random Forest Accuracy:", accuracy, "\n")
For 10000 genes, Random Forest Accuracy: 0.6356225
>   # Print the accuracy for Logistic Regression
>   cat("For", "10000", "genes, Logistic Regression Accuracy:", toString(acc_result), "\n")
For 10000 genes, Logistic Regression Accuracy: accuracy, binary, 0.488803863764662
```

Fig. 1. Accuracy for each supervised analysis method

An unexpected finding that affected our ability to train an accurate model was the presence of two clusters in the data, instead of three. This result was strongly supported by consensus clustering, and we found that the late acute and convalescent stages likely belonged to the same cluster through our heatmap of the most significant genes. Our alluvial diagram also strongly supports this conclusion, as this two-cluster pattern does not change when the number of significant genes is increased or decreased, though some of the genes seem to change cluster membership as more genes are included.

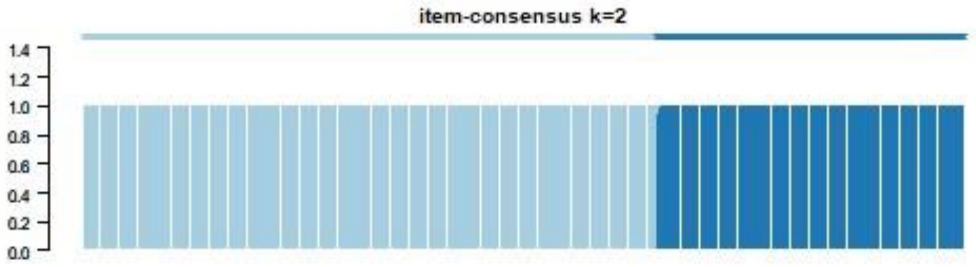


Fig. 2 Item consensus for each sample with 5000 most variable genes

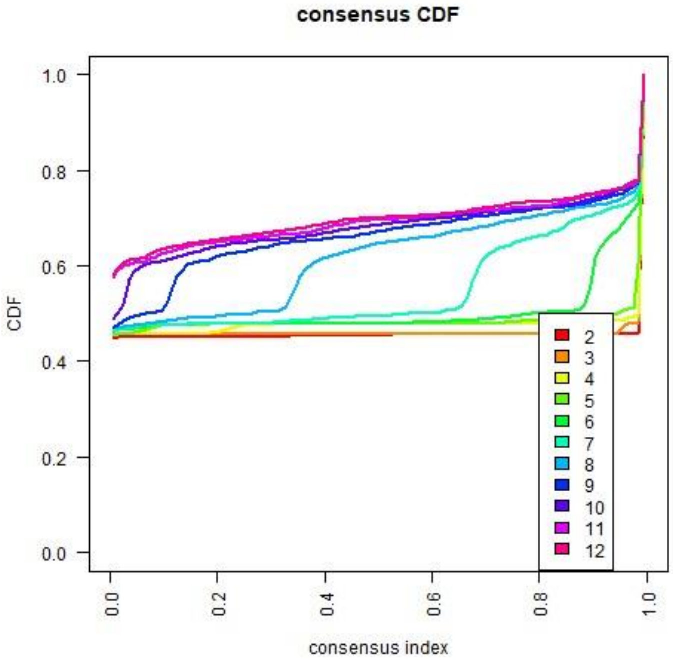


Fig. 3 Consensus CDF graph of the same clustering

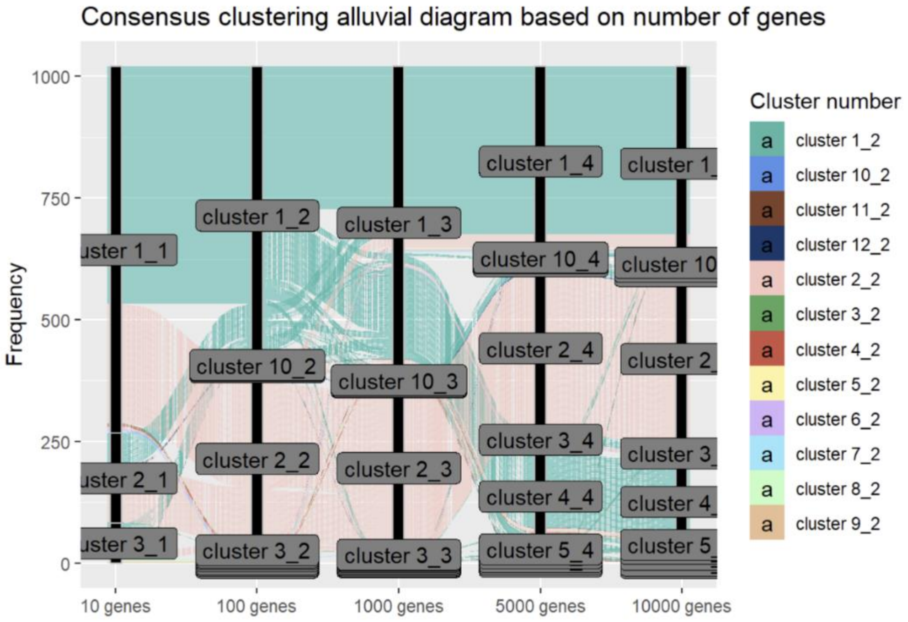


Fig 4. Alluvial diagram of previous consensus clustering with various numbers of genes

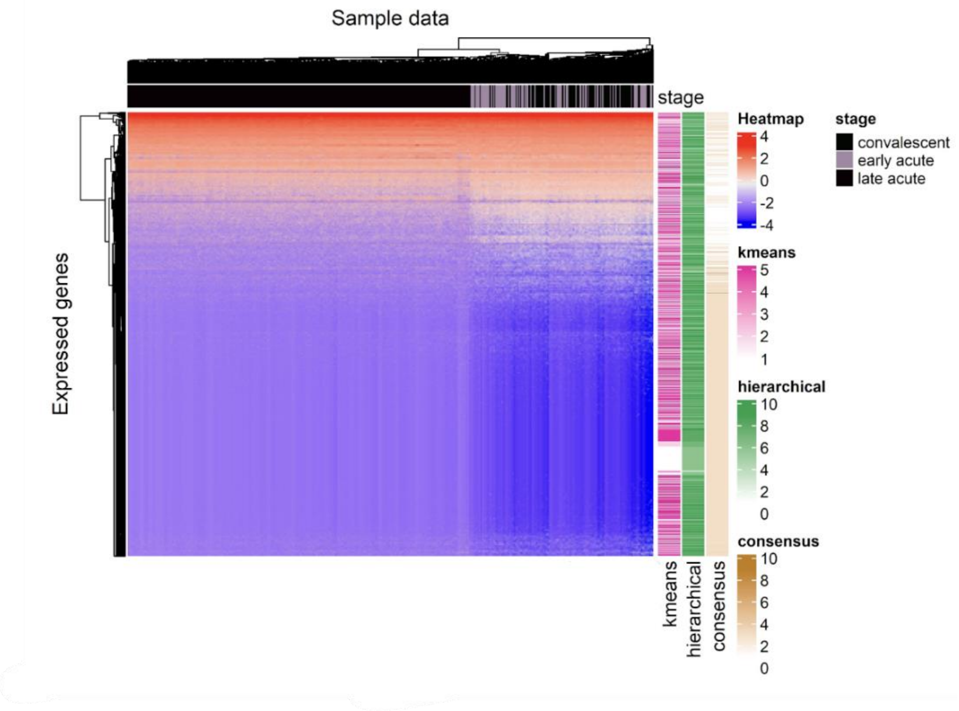


Fig. 5 Heatmap displaying top 5000 most variable genes and each method of clustering

Focusing on gene expression more broadly, our PCA plot also only reveals two focused clusters, while late acute gene expression is much more diffuse and overlaps with the other clusters.

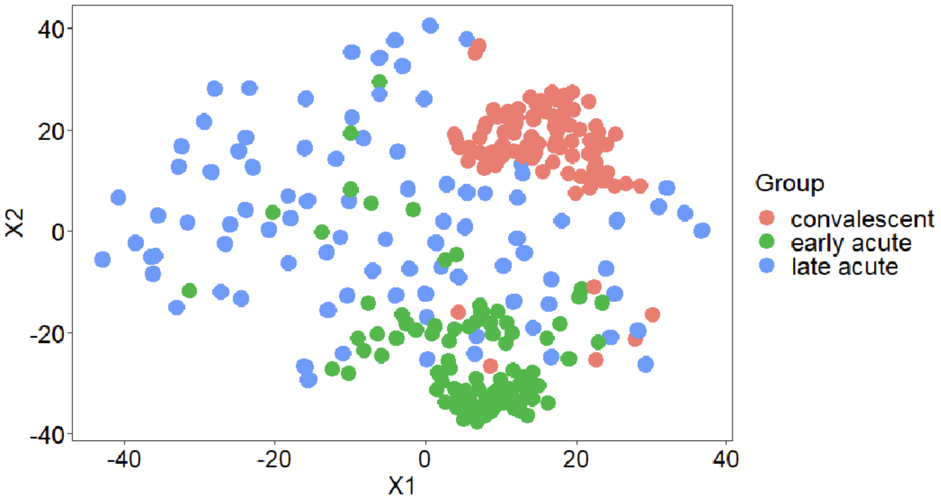


Fig. 6 PCA plot generated with every gene in every sample

5. Conclusions

Our hypothesis was that we could determine the current stage of Zika infection of an individual based on gene expression under the assumption that each stage would have significant differences in gene expression from every other stage. This assumption was incorrect. Through whole-genome analysis, we were able to discover substantial variation in gene expression across the samples in our dataset. Enrichment analysis revealed that the most differentially expressed genes were related to immune function, such as antigen binding and the IgM immunoglobulin complex. Consensus clustering conclusively identified two clusters of gene expression, and we were unable to train a model to identify the stage of Zika infection through RNA-seq data alone that was accurate enough to be clinically useful. If we were to do this project again, we would combine convalescent and late acute patients into a single category in every part of the project, which would likely result in considerably more useful plots and greater model accuracy. We would also choose a dataset that had metadata describing whether each sample was immunologically naïve to Zika, possibly eliminating a key confounding variable. An additional analysis that we would have liked to do is investigating whether there are differences in gene expression between patients in the same stage of infection based on another biological characteristic, such as age or sex.

6. References

[1] de Noronha, Lucia, et al. "Zika Virus Infection at Different Pregnancy Stages: Anatomopathological Findings, Target Cells and Viral Persistence in Placental Tissues." *Frontiers in Microbiology*, vol. 9, 2018, pp. 2266–2266, https://doi.org/10.3389/fmicb.2018.02266.