CGS4144: Introduction to Bioinformatics

Technical Report
Project Title: Mus Musculus Gene Sequencing

Professor Graim

Matheus Maldaner Rama Janco Nathan Gilman Ethan Fan

Abstract*—This study uses bioinformatics to investigate the relationships between gene expressions and various attributes of neurons in Mus Musculus, aiming to understand more about how neuronal networks operate. By leveraging the R programming language, the project explores a specialized RNA-seq dataset to clean and visualize the data, highlighting patterns and correlations within the genetic expressions and neurons' structural and functional properties. Various methodologies, including differential and enrichment analyses, were applied to discern significant genes and underlying biological themes. The insights garnered from this exploration showcase the vital role of bioinformatics in understanding the intricate workings of neuronal networks and the broader implications in neuroscience research.

I. Introduction

Understanding the diversity and complexity of the nervous system components necessitates a multifaceted approach, integrating gene expression with individual neurons' anatomical and functional properties. This project is grounded on the research question, "How can the transcriptional profiles of individual neurons, characterized by their anatomical and functional properties, be used to model and predict neuronal network behavior in a neurosymbolic framework?" It aims to construct a predictive model for neuronal network behavior by performing comprehensive analysis and visualization of the RNA-seq data using R on a dataset derived from individual neurons of Mus musculus, detailing their distinctive anatomical and functional attributes alongside their gene expression patterns. The dataset is titled "Correlating anatomy and function with gene expression in individual neurons by combining in vivo labeling, patch clamp and single cell RNA-seq," which is publicly available.

II. Implementation

The project is propelled by an intricate RNA-seq data analysis, melding diverse R packages to conduct extensive data visualizations, differential analysis, and insights integration, focusing on individual neurons characterized by their distinctive anatomical and functional attributes.

1. Data Preprocessing and Exploration

Because the initial dataset employed Ensembl IDs rather than Hugo IDs, preprocessing had to be done in order to convert the gene variable to an acceptable format. Due to the size of the processed database, it was unable to be pushed to github and a script was then made so that data could be easily converted from Ensembl to Hugo [2].

Providing clear documentation for users, the program defines the working directory, respective file paths and then proceeds to perform the bulk of conversions. A key step in assigning Hugo IDs was ensuring their uniqueness, as when naming rows in a dataframe there must be no duplicate names. This tricky task at first, as different Ensembl IDs may map to the same Hugo ID, was dealt with by combining duplicate Hugo names by averaging them out to keep one unique entry per gene and removing incomplete values (NA values).

	SRR5071780	SRR5071781
0610005C13Rik	0.5174812	0.5122866
0610009B22Rik	10.3057429	11.2011433
0610009E02Rik	0.5174812	0.5122866
0610009L18Rik	0.5174812	4.4519819

Table 1 - Expression Matrix Dimension and Gene Count

Upon preprocessing, the data was then log-scaled, not having to worry about whether the data was numerical or not, as the only categorical column had been made into the row names., Ensuring to add 1 to every entry to tackle the normalization issue before log scaling the values, a density plot was created through the usage of the ggplot library. The median range for specific genes can be seen visualized below and the full image may be found in the github repository [3].

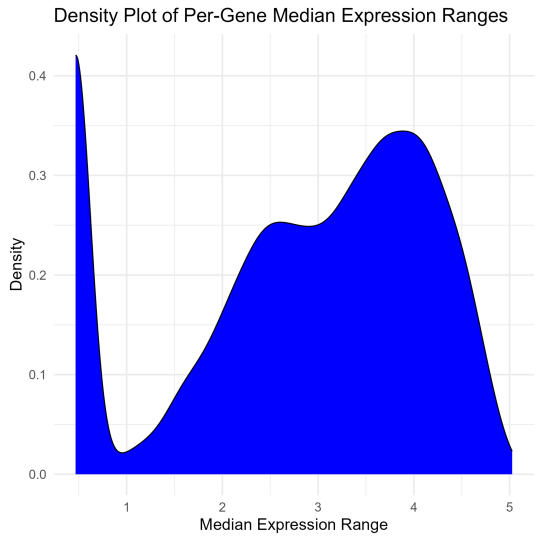


Figure 1 - Density Plot

Though possessing an unusual shape, the above density plot depicts certain patterns in the data, with median expression genes being found at large in the lower region of the graph. There is a clear variation in the observed data, which shall be explored further in the plots that follow.

2. PCA and UMAP Plot Generation

Following the preprocessing of the data, PCA and UMAP plots were developed to visually decipher the underlying structures and relationships in the dataset.

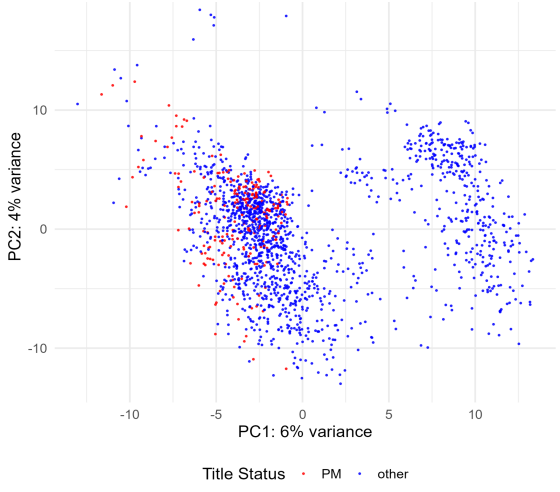


Figure 2 - PCA Plot

Looking at the graph we see that the x-axis and y-axis are labeled PC1 and PC2. The x-axis and y-axis account for 6% and 4% of the total variance in the dataset respectively. The values of 6% and 4% are relatively low, suggesting that the dataset's variability is spread across many dimensions, and the first two principal components only capture a small portion of it. The largest cluster in our data is centered from -5 to 0 in the x-axis and 0 to 5 on the y-axis. The second cluster mimics the first one in shape but is centered around 10 on the x-axis and has significantly fewer points. The biggest thing we noticed is that the majority of the PM plots are in the center of the first cluster while the other points are in both clusters. The title statuses that were not PM were all grouped into another umbrella even though they could be very different in their gene expression profiles. This is likely the cause of the blue dots being very widely spread out.

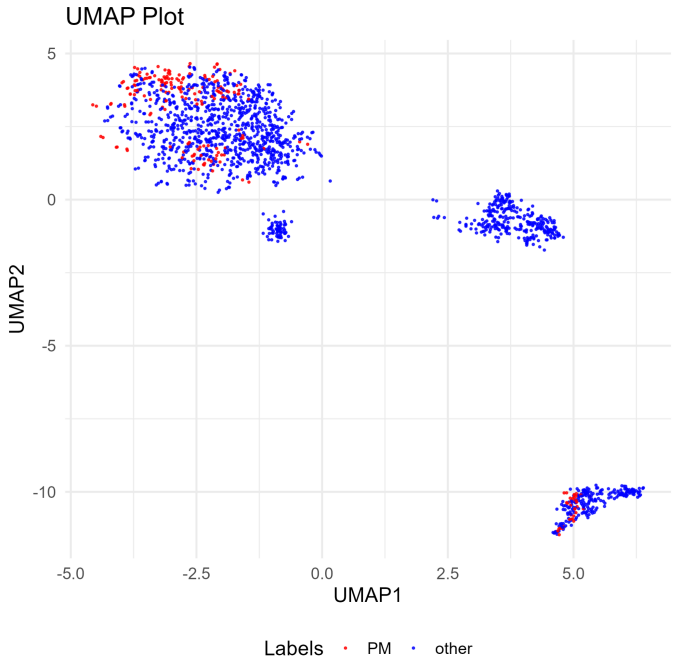


Figure 3 - UMAP Plot

Looking at the UMAP plot we can see the x-axis and y-axis labeled UMAP1 and UMAP2 respectively. The axes "UMAP-1" and "UMAP-2" are two arbitrary dimensions in a lower-dimensional space that UMAP has mapped our data to. Looking at the data points is more important in a UMAP. We can see that the PM data points are focused in two main clusters. The PM points are also around some “other” points which likely means that they are similar in their gene expression profiles. There are also clusters of “other” points that are not near the PM points. These points are similar to each other but not to the PM points.

3. Differential Analysis

A nuanced differential analysis was conducted to elucidate the intricacies of varied gene expressions in correlation with their anatomical and functional characteristics, providing profound insights into the interaction and variance in the studied neurons.

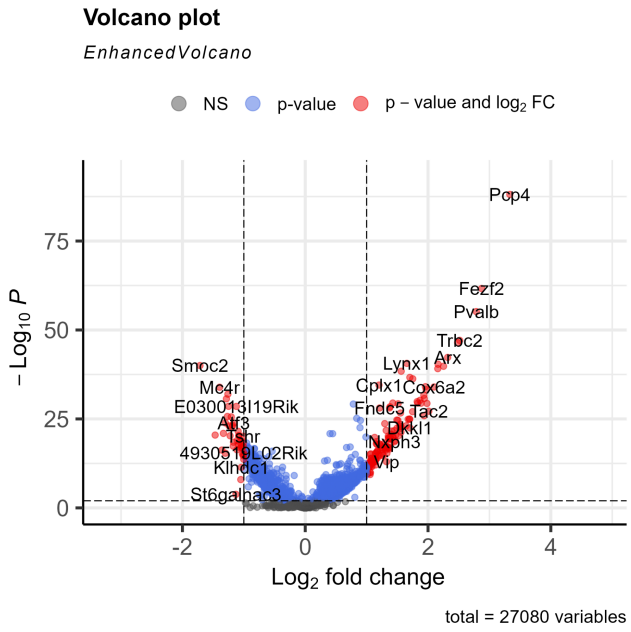


Figure 4 - Volcano Plot

Figure 4 was created utilizing the tutorial provided in the assignment tutorial. In order to create the volcano plot the dataset first needed to be run under a differential expression analysis using the DESeq2 package in RStudio. After organizing the dataset by the metadata and splitting it up into two separate groups by refinebio_title as the demarcator. The two groups were PM, at 211 samples, and other, with 1481. Before running the differential expression analysis I defined a minimum counts cutoff to filter the data. After running I plotted and saved the data as a .tsv. One interesting conclusion from the data is that a lot of samples are located in the upper right corner indicating that they have high statistical significance and magnitude of change meaning smaller p-values and an increase of gene expression or upregulation relative to others.

Hugo	baseMean	log2FoldChange	lfcSE	pvalue	padj	threshold
Pcp4	15.16	3.33	0.1645	4.20×10^{-93}	8.07×10^{-89}	TRUE
Fezf2	7.02	2.87	0.1687	2.29×10^{-66}	2.20×10^{-62}	TRUE
Pvalb	7.57	2.79	0.1730	1.06×10^{-59}	6.79×10^{-56}	TRUE
Trbc2	5.66	2.52	0.1689	1.73×10^{-51}	8.30×10^{-48}	TRUE
Rpp25	4.81	2.48	0.1681	9.27×10^{-51}	3.56×10^{-47}	TRUE
Arx	4.64	2.32	0.1646	1.59×10^{-46}	5.10×10^{-43}	TRUE
Cort	4.27	2.25	0.1651	8.21×10^{-44}	1.57×10^{-40}	TRUE
Rnd2	12.25	2.16	0.1577	1.75×10^{-44}	4.21×10^{-41}	TRUE

Table 2 - Differentially Expressed Genes

Table 2 was created during the process for the volcano plot in Figure 4. It ran under a differential expression analysis and organized using two groups, PM and other, for refinebio_title. It was then ranked in order of refinebio_ascension code to ensure it was in the same order as the metadata. After defining a minimum sample cutoff, we formatted it into a DESeqDataSet object and ran using the titles as the variable. It was then run through lfcShrink to help decrease noise and preserve large differences. Then we sorted it to be readable and saved it as a .tsv. The most significant gene was the PCp4, or the Purkinje cell protein 4, in the *Mus Musculus*, or house mouse. It is responsible for protein coding and is predicted to be in the axon and neurofilament and active in the cytoplasm as part of the protein-containing complex.

4. Heatmap Generation

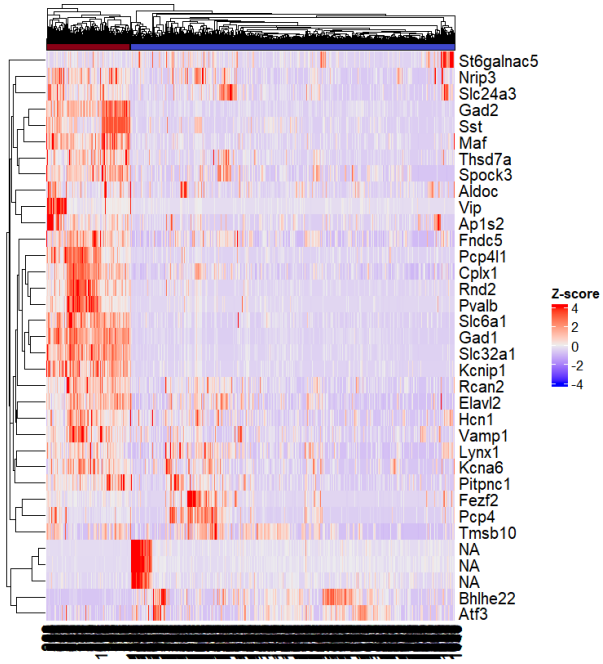


Figure 5 - HeatMap

This heatmap was created using the ComplexHeatmap package provided from jokergoo/ComplexHeatmap repository hosted on github [1]. This heat map was created by calculating the z-score of the counts values for each Hugo gene that we deem to be significant. The filters that we used to establish significant genes were an adjusted p-value less than 0.05, a base mean greater than seven, and an absolute log fold change value greater than or equal to 1. These values were chosen to narrow down our data set from 30,000 possible genes and create a more visible heat map.

When creating the heatmap, the cluster_rows and cluster_columns parameters were set to true as a way of

grouping our genes in a visually sensible way. The heatmap clustered into two main groups: PM and Other.

	Hugo	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	threshold
1	Pcp4	15.158855	3.348482	0.16360057	20.467422	4.202703e-93	8.076335e-89	TRUE
2	Fezf2	7.020452	2.893097	0.16809090	17.211501	2.177156e-66	2.091920e-62	TRUE
3	Pvalb	7.568534	2.806859	0.17225784	16.294520	1.079471e-59	6.914729e-56	TRUE
8	Rnd2	12.246864	2.186667	0.15626010	13.993764	1.701634e-44	4.087537e-41	TRUE
9	Slc32a1	11.150640	2.181824	0.15851600	13.764062	4.192907e-43	7.325008e-40	TRUE
15	Gad1	18.080927	1.969326	0.15308429	12.864328	7.146105e-38	8.582919e-35	TRUE
20	Kcnip1	8.033220	1.895300	0.15623373	12.131184	7.219803e-34	5.780956e-31	TRUE
27	Slc24a3	10.265045	1.722474	0.12918411	13.333479	1.478400e-40	2.185416e-37	TRUE
30	Lynx1	9.303682	1.670746	0.11899792	14.040129	8.855179e-45	2.431000e-41	TRUE
32	Pcp4l1	10.226627	1.584553	0.15167819	10.446808	1.515404e-25	4.622462e-23	TRUE
34	Rcan2	11.801997	1.579626	0.11583127	13.637300	2.402927e-42	3.848087e-39	TRUE
39	Gad2	13.209136	1.534730	0.14009965	10.954563	6.318238e-28	2.529533e-25	TRUE
40	Nrip3	8.301223	1.533764	0.12857465	11.928973	8.359877e-33	5.355059e-30	TRUE
42	Maf	9.468170	1.517738	0.13824889	10.978302	4.859748e-28	1.987016e-25	TRUE
49	Pitpnc1	9.579955	1.446267	0.12074605	11.977761	4.647074e-33	3.434724e-30	TRUE
59	Elavl2	9.836369	1.401709	0.11923161	11.756188	6.563119e-32	3.709513e-29	TRUE
61	Hcn1	9.745783	1.395121	0.11932323	11.691947	1.401368e-31	7.694309e-29	TRUE
62	Slc6a1	10.516700	1.387281	0.14085550	9.848963	6.925470e-23	1.584366e-20	TRUE
66	Vip	8.589300	1.367358	0.17065794	8.012270	1.126101e-15	1.015976e-13	TRUE
75	Tmsb10	8.047393	1.317101	0.12250128	10.751730	5.816069e-27	2.149373e-24	TRUE
76	Sst	26.003211	1.305960	0.15323945	8.522346	1.563529e-17	1.810020e-15	TRUE
83	Ap1s2	7.799840	1.247616	0.13438827	9.283670	1.637395e-20	2.940731e-18	TRUE
84	Thsd7a	11.379992	1.237286	0.13165796	9.397727	5.575435e-21	1.050423e-18	TRUE
86	Fndc5	8.701770	1.228816	0.10517450	11.683596	1.546130e-31	8.253329e-29	TRUE
87	Cplx1	29.047729	1.215463	0.09384952	12.951189	2.313435e-38	2.963819e-35	TRUE
94	St6galnac5	7.611230	1.188784	0.14465709	8.217948	2.070144e-16	2.061241e-14	TRUE
96	Spock3	7.204494	1.185453	0.12400293	9.559876	1.178980e-21	2.410261e-19	TRUE
99	Kcna6	9.811984	1.182116	0.12566024	9.407242	5.093255e-21	9.690800e-19	TRUE
105	Vamp1	11.133824	1.153469	0.11704434	9.854976	6.523231e-23	1.510325e-20	TRUE
117	Aldoc	13.811155	1.106127	0.13330559	8.297679	1.061651e-16	1.087700e-14	TRUE
152	Rph3a	11.827629	1.007579	0.10201113	9.877143	5.230290e-23	1.225738e-20	TRUE
27076	Bhlhe22	9.502190	-1.132708	0.11752820	-9.637755	5.538598e-22	1.156905e-19	TRUE
27079	Atf3	9.268454	-1.183840	0.10967990	-10.793593	3.690713e-27	1.390675e-24	TRUE

Table 3 - Significantly Differentially Expressed Genes used in Heatmap.

III. Experiments

Implementing these methodologies provided an extensive understanding and visualization of gene expression's intricate interaction and variance in correlation with their respective anatomical and functional properties, addressing the underlying research question and elucidating the diversity and intricacy of the neuronal networks.

Following up with the previously extracted list of differentially expressed genes, we proceeded to run enrichment analysis for the following four methods:

i. topGO
ii. clusterProfiler
iii. gProfiler2
iv. GenomicSuperSignature

Below you may find the results after performing enrichment analysis using gene ontology for each of the described techniques..

i. topGO, BP Ontology

GO.ID	Term	Annotated	Significant	Expected
GO:0008150	biological_process	22460	4893	4885
GO:0008285	negative regulation of cell population p...	742	236	161.38
GO:0042776	proton motive force-driven mitochondrial...	65	27	14.14
GO:0031398	positive regulation of protein ubiquitin...	120	47	26.1
GO:0045944	positive regulation of transcription by ...	1257	342	273.39
GO:0043065	positive regulation of apoptotic process	619	188	134.63
GO:0000226	microtubule cytoskeleton organization	650	182	141.37
GO:1990830	cellular response to leukemia inhibitory...	292	44	63.51
GO:0065003	protein-containing complex assembly	1474	430	320.59
GO:0000122	negative regulation of transcription by ...	1003	264	218.15

Term	classicFisher	classicKS	elimKS
biological_process	0.98622	0.00033	< 1e-30
negative regulation of cell population p...	9.00E-11	5.20E-15	1.30E-11
proton motive force-driven mitochondrial...	0.00027	2.40E-10	2.40E-10
positive regulation of protein ubiquitin...	1.20E-05	2.80E-09	2.80E-09
positive regulation of transcription by ...	1.90E-06	1.40E-09	7.60E-09
positive regulation of apoptotic process	3.00E-07	3.50E-14	9.60E-09
microtubule cytoskeleton organization	9.70E-05	9.10E-09	7.60E-08
cellular response to leukemia inhibitory...	0.99862	1.70E-07	1.70E-07
protein-containing complex assembly	4.40E-12	< 1e-30	3.70E-07
negative regulation of transcription by ...	0.0003	6.50E-07	6.70E-07

Table 4 - topGo gene enrichment, BP Ontology

Table 4 showcases the topGO gene enrichment utilizing three tests: the classic fisher, the Kolmogorov-Smirnov, and the Kolmogorov-Smirnov elimination test. The data used was the differentially expressed genes. To prepare the data, we found the intersection of valid genes in the Mus Musculus database and the ones from our dataset. We also selected genes with a p-value less than 0.01 to show statistically significant values to perform the enrichment tests alongside the list of valid genes. We then created the topGOdata object with a BP, or biological process, ontology with a node size of 10 using Symbol ID or Hugo ID. Afterward, the three enrichment tests were run on the topGO data. The Fisher test is based on gene counts, while the Kolmogorov-Smirnov test computer enrichment is based on gene scores, representing how differentially expressed a gene is. An interesting observation is that GO:0008285, or negative regulation of cell population proliferation, scores well across all three tests, showing that the distribution of genes is statistically significant, meaning it is highly enriched in our dataset.

ii. clusterProfiler

Following up with another method commonly used for gene enrichment analysis, the list of differentially expressed genes extracted in previous steps had to be altered due to the nature of clusterProfiler. Upon dealing with the error stating that no matches had been found, online resources and tutorials were sought after and hinted that clusterProfiler could be treating our SYMBOL IDs as ENTREZ IDs. Upon making the conversion and performing the mapping from Hugo to Entrez, a new error was encountered, stating that the data must be in descending order for this specific method. A plot was then generated depicting relationships and dependencies among the studied genes.

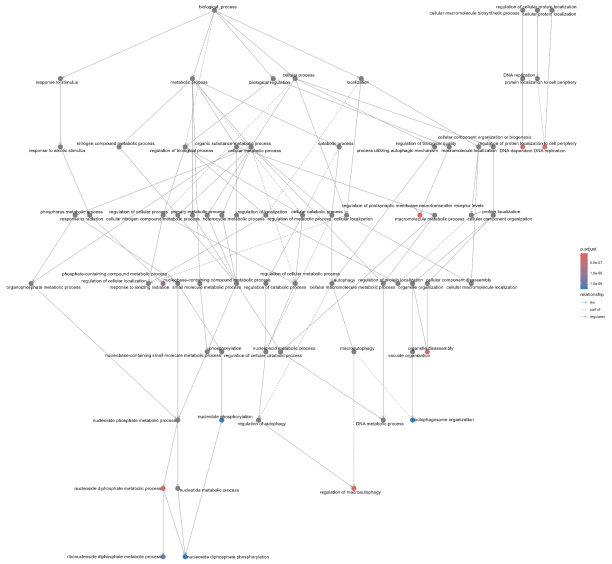


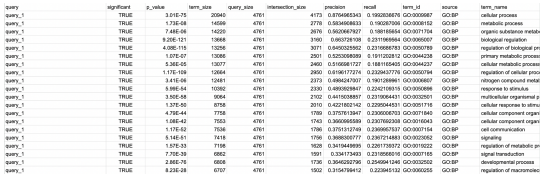
Figure 6 - ClusterProfiler Direct Acyclic Graph

Though too small to read the labels, an enlarged copy of the plot may be found in the github repository associated with the project. Additionally, the plot helps illustrate the hierarchical structure in gene relations or pathways, which is crucial for understanding gene functions and interactions. Furthermore, a gene enrichment analysis table was created, displaying the p value, q value and enrichment score associated with each term.

	Term	P_Value	Q_Value	Enrichment_Score
1	regulation of postsynaptic membrane neurotransmitter rece...	1.000000e-10	1.600000e-08	0.6125115
2	regulation of macroautophagy	7.796336e-10	3.118534e-08	0.6087424
3	regulation of protein localization to cell periphery	5.500992e-10	3.118534e-08	0.5871225
4	nucleoside diphosphate metabolic process	6.609088e-10	3.118534e-08	0.5863429
5	DNA-templated DNA replication	2.382049e-09	7.622556e-08	0.5820555

Table 5 - clusterProfiler Gene Enrichment Analysis

iii. gProfiler2



term_name	term_size	intersection_size	p_value	p_adjusted	effective_domain_size
biological regulation	13668	3160	9.20E-121	1.97E-117	26856
regulation of biological process	13256	3071	4.08E-115	4.37E-112	26856
regulation of cellular process	12664	2950	1.17E-109	8.35E-107	26856
multicellular organism development	4828	1356	2.16E-84	1.16E-81	26856
positive regulation of biological process	6614	1713	1.14E-80	4.32E-78	26856
system development	4076	1182	1.21E-80	4.32E-78	26856
anatomical structure development	6222	1629	2.12E-79	6.48E-77	26856
developmental process	6808	1736	2.86E-76	7.66E-74	26856
cellular process	20940	4173	3.01E-75	7.16E-73	26856
positive regulation of cellular process	6091	1572	1.03E-70	2.20E-68	26856
localization	5386	1419	3.53E-68	6.88E-66	26856
nervous system development	2517	779	2.08E-62	3.72E-60	26856
multicellular organismal process	9064	2102	3.50E-58	5.77E-56	26856
cell population proliferation	3413	964	3.47E-57	5.30E-55	26856
establishment of localization	4723	1241	8.65E-57	1.23E-54	26856
transport	4547	1200	1.31E-55	1.75E-53	26856
organonitrogen compound metabolic proces	6389	1570	1.38E-54	1.74E-52	26856
response to stimulus	10392	2330	5.99E-54	7.12E-52	26856
regulation of signaling	3588	988	4.02E-53	4.53E-51	26856
regulation of cell communication	3591	987	1.16E-52	1.19E-50	26856

Table 6 - gProfiler2 Gene Enrichment Analysis

Table 6 presents the 20 most statistically significant gene ontology terms derived from the enrichment analysis of our gene set. The goal of this analysis was to identify biological processes, cellular components, or molecular functions that are over-represented in our set of genes, hinting at the predominant biological themes at play in the experimental condition. The term_name is the name or description of the gene ontology (GO) term, providing insight into the biological theme it represents. The term_size represents the total number of genes that are associated with a particular Gene Ontology (GO) term in the reference database (e.g., the total number of genes known to be involved in a specific biological process or molecular function in the organism of interest). The p_value is the raw statistical significance value from the enrichment analysis for each term. A smaller p-value indicates higher significance. The p_adjusted is the p-value adjusted for multiple testing, typically using the Benjamini-Hochberg procedure or another correction method. Adjusted p-values provide a more stringent criterion for significance by accounting for the multiple terms tested in the analysis.

iv. topGO, MF Ontology

Annotated	Significant	Expected	Rank in classicFisher
molecular_function	22124	4944	4934.85
protein binding	9361	2549	2088.01
identical protein binding	2373	664	529.31
metal ion binding	3607	1000	804.56
ATP binding	1402	388	312.72
protein-containing complex binding	1609	453	358.89
protein domain specific binding	833	258	185.8
catalytic activity, acting on a protein	2256	582	503.21
GTPase activator activity	241	82	53.76
protein kinase binding	780	244	173.98

Annotated	classicFisher	classicKS	elimKS	
molecular_function	1322	0.99	0.00065	< 1e-30
protein binding	1	< 1e-30	< 1e-30	< 1e-30
identical protein binding	11	0	< 1e-30	< 1e-30
metal ion binding	6	0	< 1e-30	< 1e-30
ATP binding	33	0.0000009	0	0
protein-containing complex binding	24	0.00000001	0	0
protein domain specific binding	21	0.0000000031	0	0.00000000049
catalytic activity, acting on a protein	49	0.000028	0	0.0000013
GTPase activator activity	48	0.000021	0.0000015	0.0000015
protein kinase binding	22	0.0000000031	0.0000000007	0.0000017

Table 7 - topGo gene enrichment, MF Ontology

This table showcases a use of the topGO gene enrichment to perform a molecular function ontology. Similarly to Table 4, the classFisher, classicKS and elimKS columns can be observed in the table as these tests are reliable options for dealing with statistical analysis in a variety of scenarios. We look for genes that perform well across these three tests to make definitive interpretations of our data.

Molecular Function ontology is useful with statistical analysis to uncover which molecular functions are most relevant or significant to our scientific question of predicting the properties of neuronal behavior. By mapping neuronal genes to their molecular function, we can better understand the complex system that makes up the brain.

IV. Unsupervised Analysis

Embarking on this scientific journey to decipher intricate genetic interplays, our focus shifted towards clustering algorithms, promising insights into patterns and potential clusters within gene expression data. Building upon the previously isolated 5,000 most variable genes, multiple clustering paradigms were embraced, casting light on distinct facets of our complex dataset by employing the following methods:

i. K-means
ii. Hierarchical clustering (hclust)
iii. ConsesusClusterPlus
iv. Gaussian Mixture Models

Below you may find the results after running different clustering algorithms on the subset of our original data.

i. K-means

The first cluster analysis we chose was k-means which is an unsupervised learning algorithm used to identify groups (clusters) of data points in a dataset that have similar characteristics. We used the k-means clustering method on the 5000 most variable genes with an initial k set to 3 and this was the result:

kmeans_cluster_assignments	Freq
1	109
2	3969
3	922

Table 8 - initial k value

As shown by the data we can see that a majority of the genes are clustered at cluster 2. We then ran the k-means clustering method with k set to 2, 4, 5, and 6 and we compared the results using Silhouette Analysis. This method calculates the silhouette score for each sample, which is a measure of how similar an object is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

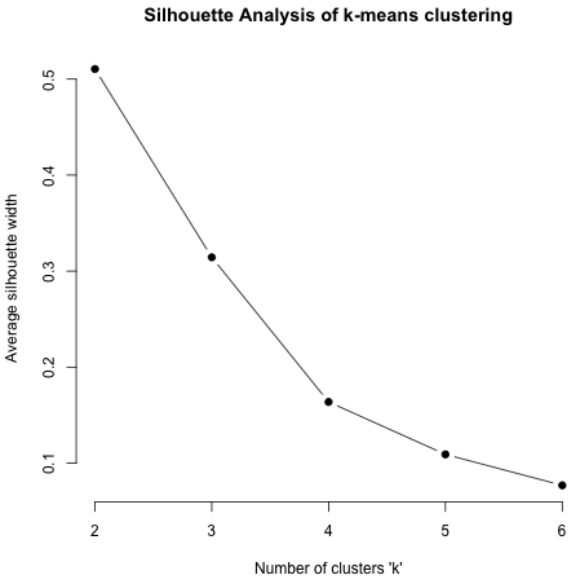


Figure 7 - K-Means Silhouette Analysis Graph

The results in the table indicate that the lower k values had a better average silhouette score. This indicates that the lower k values create more accurate clusters. For the final part of our k-means analysis, we calculated the analysis using k = 5 and 10, 100, 1,000, and 10,000 genes. We then used an alluvial diagram to visualize how the different clustering setups changed cluster memberships for each sample.

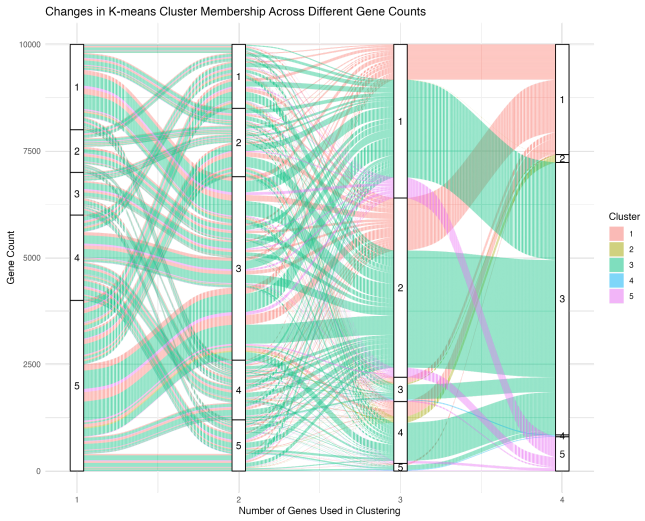


Figure 8 - Alluvial Diagram for K-Means

The alluvial diagram is used to visualize how data points move from one state to another or how clusters of data points change when certain parameters are adjusted. The x-axis represents the number of genes used in the clustering, 10, 100, 1,000, 10,000, with each vertical bar corresponding respectively. On each bar is the cluster number. The y-axis has the gene count. The individual streams represent each group of clusters. Looking at the diagram we can see how certain clusters move from one cluster group to another when the gene sample changes. We can also see how the cluster streams consolidate from 1,000 to 10,000.

ii. Hierarchical clustering (hclust)

Using hierarchical clustering, I had to either select a k-value for the number of clusters or h-value for the height at which to group in the generated hierarchy. I experimented with multiple heights and k-values, but found that increasing or decreasing this value outside of the range **30 < k < 50** made it harder to interpret the data. For example, a **k-value** of **8** yielded the following clustering results:

1	2	3	4	5	6	7	8
5	1	1	44	4909	26	3	11

Table 9 - # of clusters of Hierarchical clustering

This made me believe it was necessary to greatly increase the k-value since I had **98%** of the data in a single cluster. Even with **k=50**, **87%** of the data was still in a single cluster. I decided to proceed with this k-value, as a majority of our data was collected from a similar region of the brain.

In addition to experimenting with the number of clusters, we wanted to experiment with the number of samples that we ran the clustering algorithm on.The following plot represents how running the hclust clustering algorithm with sample sizes of 10, 100, 1000 and 10000 affected the cluster membership for each sample. We can observe that a

majority of the samples belong to the same cluster when the algorithm is applied to 10,000 genes. When experimenting with the different numbers of genes, I selected a height of **3000** to cut at for **n=10, 100,** and **1000**. From visual inspection of the dendrograms for the respective sample sizes, it appeared to me that this arbitrary height was a point where many clusters formed allowing us to see the distribution and relationship of the data. For **n=10000**, I chose a k-value of **50** to get a more representative view of the data. Most of the data belongs to a single cluster, so it was necessary to choose a high k in this case.

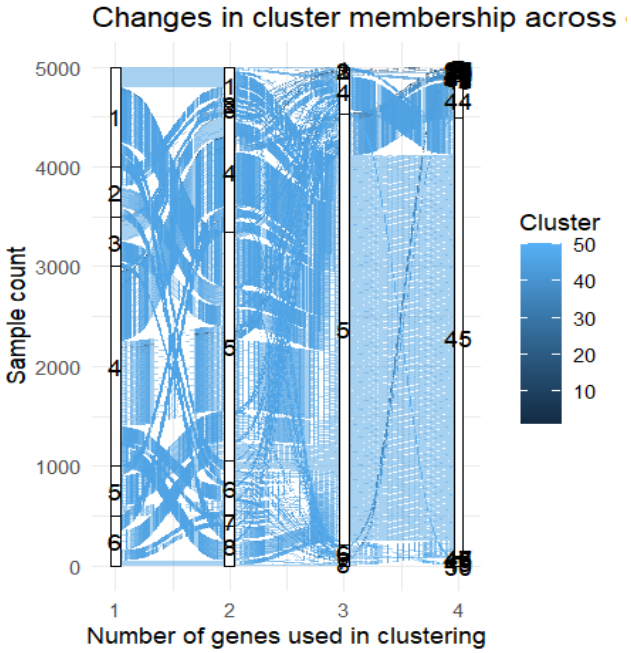


Figure 9 - Alluvial Diagram for Hclust

iii. ConsensusClusterPlus

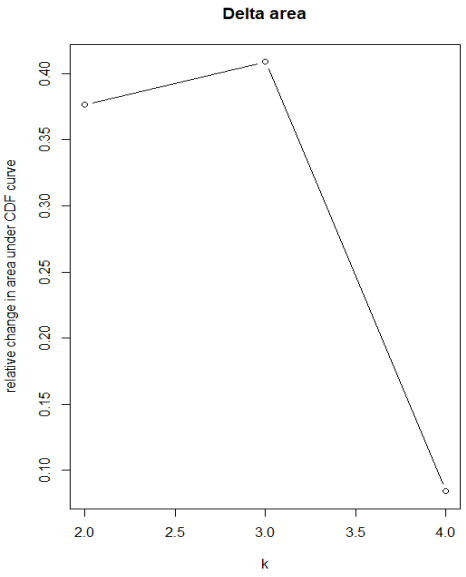


Figure 10 - Delta Area for ConsensusClusterPlus

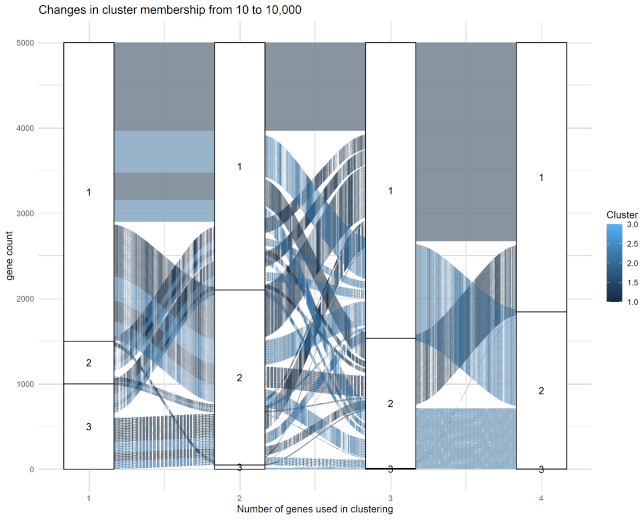


Figure 11 - Alluvial for ConsensusClusterPlus

I ran the ConsensusClusterPlus method that defined a maximum k number of clusters, and I ran the cluster analysis with a k-value of 3. Changing the k-value determines the number of clusters the algorithm will attempt to fit the data into. Above is a delta area plot, which measures cluster stability, showing the relative change in area under the CDF curve. I chose three since it shows a drop from 3 to 4 clusters, showing that more clusters will likely not provide more information and may lead to less meaningful divisions.

All tests were run through ConsensusClusterPlus with a max K-value of three due to the delta area, 100 reps, and 80% item and gene resampling due to its extensive data size. The number of genes affected clustering, but its distribution stayed the same. For instance, it was consistently in the second cluster across the various gene amounts of 10, 100, 1000, and 5000. I left out 10,000 due to the complexity of its calculation. The first cluster's membership remained similar throughout, but the third diminished with size.

iv. Gaussian Mixture Models

```
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEI (diagonal, equal shape) model with 6 components:

 log-likelihood    n    df       BIC       ICL
      -7611735 1000 11854 -15305355 -15305355

Clustering table:
  1   2   3   4   5   6
 58 104 152 276   7 403
```

Table 10 - GMM Cluster Analysis

The Mclust function, leveraging Gaussian Mixture Models (GMM), is adept at autonomously determining the optimal number of clusters for a given dataset. It achieves this through the Bayesian Information Criterion (BIC), which weighs model fit against its complexity. For the dataset under examination, BIC indicated that 6 clusters

were the optimal choice. Recognizing the potential influence of gene count, we attempted to adjust this number to see how it would influence the resulting clusters. We worked with gene counts of 10, 100, and 1,000.

However, we encountered a complication during our analysis. Upon attempting to use 10,000 genes, the Mclust function experienced difficulties processing the data. Specifically, the method would stall at the loading stage, lingering at 11% for over 5 hours without progress. Given this challenge, we decided to cap our gene count at 1,000 for the GMM analysis. So, for the purpose of this study and any related queries, it's essential to note that our GMM analysis is based on a maximum of 1,000 genes.

```
$`10_genes`   $`100_genes`   $`1000_genes`
[1] 9         [1] 5         [1] 6
```

Figure 12 - Cluster Variation with Gene Counts in GMM

The figure provides a clear depiction of how the clusters change with varied gene counts. With just 10 genes, the clusters appear more generalized, possibly missing out on some specific details. As the number of genes increases to 100 and then 1,000, we begin to see more distinct clusters. However, with 1,000 genes being our cap due to processing limitations, we cannot comment on the behavior at higher gene counts, like 10,000.

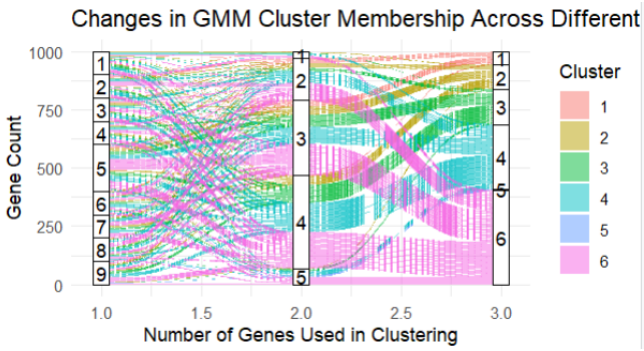


Figure 13 - GMM Alluvial Diagram

This diagram is a straightforward representation of how the cluster assignments shift based on gene counts. As the number of genes used in the clustering changes, we can clearly observe variations in cluster memberships, emphasizing the importance of the gene count parameter. The diagram reveals the interplay between gene counts and how they result in different cluster formations, painting a comprehensive picture of our data.

In summary, our exploration with the Mclust function revealed its capabilities and potential limitations, especially in the context of larger datasets. The importance of gene count became evident in our study, both in terms of its impact on clustering and the computational challenges it presented.

V. Heatmaps and Dendrograms

Below we have a heatmap of the 5,000 most variable genes captured from our expression data. We have annotation data for the four clustering algorithms. As an unfortunate result of the number of samples in our data, we have a hard to interpret heatmap.

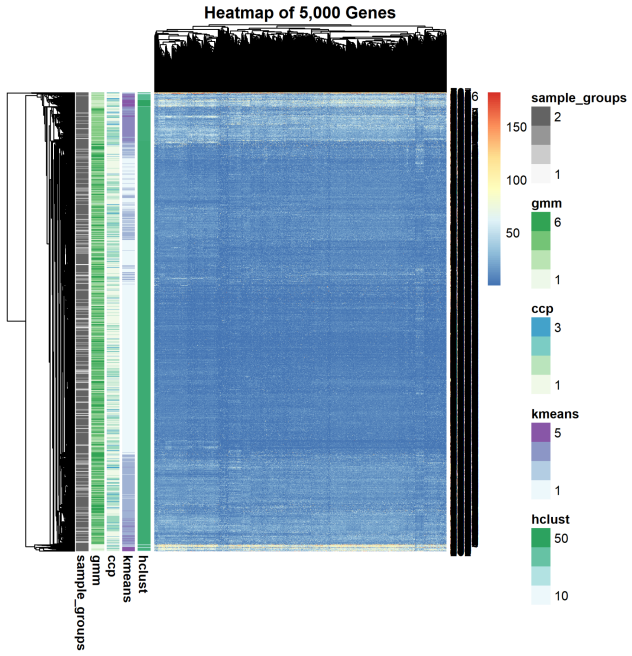


Figure 14 - Heatmap

VI. Statistics

i. K-means

	Unadjusted	Adjusted
kmeans_10	9.927168E-01	9.927168E-01
kmeans_100	9.740343E-01	9.927168E-01
kmeans_1000	4.246978E-08	8.493956E-08
kmeans_10000	5.145756E-09	2.058302E-08

Table 11 - p values for k means

Looking at the table we can see that the kmeans_10 and kmeans_100 are mostly independent from the clusters found in Assignment 1. The kmeans_1000 and kmeans_10000 show a correlation between them and the clusters from part 1 with kmeans_10000 having a slightly better correlation.

ii. Hierarchical clustering (hclust)

```
            Unadjusted   Adjusted
hclust_10   0.99842997 0.99842997
hclust_100  0.09145313 0.18290626
hclust_1000 0.01472204 0.05888816
hclust_all  0.63553277 0.84737703
```

Table 12 - p values for hclust

The clustering results from hclust_10 indicate a very high p-value, suggesting the clusters from this method are largely independent and show no significant association with the groups from Assignment 1.

For hclust_100 and hclust_1000, their respective p-values, especially when unadjusted, are lower. This implies that these clustering results have a stronger association with the groups identified in Assignment 1. It's evident that as the cluster count changes (or perhaps due to other distinct characteristics of the hclust_100 and hclust_1000 methods), different patterns or structures emerge that align more closely with your Assignment 1 groupings.

Conversely, hclust_all, with its moderate p-value, suggests a degree of independence from the Assignment 1 groups, though not as pronounced as hclust_10. As always, when interpreting these results, it's important to be aware of other influencing factors that might not be evident in the dataset or method.

iii. Gaussian mixture model

	Unadjusted	Adjusted
gmm_10	9.999666e-01	9.999666e-01
gmm_100	9.710842e-01	9.999666e-01
gmm_1000	3.612439e-11	1.083732e-10

Table 13 - p values for gaussian mixture model

The clustering outcomes from gmm_10 and gmm_100 appear to be mostly distinct from the groups in Assignment 1, indicating the clusters are almost entirely independent from the ones found in the first part of the assignment.

However, the gmm_1000 clustering output shows a notable correlation with them. Clearly, the cluster count (or perhaps other features unique to the gmm_1000 clustering) encapsulates certain trends or configurations consistent with your Assignment 1 classifications. Once more, it's crucial to note that the trial couldn't be conducted with larger gene counts because of computational constraints.

iv. ConsensusClusterPlus

	Variation	Original_P_Values	Bonferroni_Adjusted	Holm_Adjusted	BH_Adjusted
1	10	1	1	1	1
2	100	1	1	1	1
3	1000	1	1	1	1
4	5000	1	1	1	1

Table 14 - p values for ConsensusClusterPlus

The clustering outcomes from ConsensusClusterPlus all show, even when adjusted, a p-value of 1, showing that it is entirely independent of the two groups identified in Assignment 1. The two groups identified were based on the title status as were PM-# and others. I calculated these p-values using Chi-squared testing by creating a data frame from the cluster results containing the gene names and their respective clusters. I then merged it with the raw data by mapping it with Hugo IDs. I then gathered it by refinebio_accession_code, which was collected in a data frame with columns of cluster, gene, and title status. I then made a contingency table of the cluster and title status and performed the Chi-squared test, resulting in a p-value of 1. The data shows that it is perfectly independent, and our cluster membership isn't associated with the title status, but it may also imply the data processing was incorrect or merged incorrectly.

VII. Supervised Analysis

The field of bioinformatics presents several algorithms and methodologies designed to dissect and comprehend the vast complexities of gene expression data. By harnessing these sophisticated techniques, researchers can uncover the nuanced patterns and relationships embedded within datasets, shedding light on the intricate interplay of genes and their potential roles in diverse biological processes.

Building on the foundational work from earlier assignments, the group delved deeper into the gene expression data by employing four supervised machine learning algorithms. These algorithms were selected based on their robustness, adaptability, and proficiency in managing high-dimensional data, attributes that are essential when navigating the intricacies of gene expression datasets. The methods chosen for this comprehensive analysis are:

i. Support Vector Machine (SVM)
ii. Logistic Regression
iii. Random Forest
iv. Naive Bayes

In the sections that follow, a thorough exploration of each method is presented, emphasizing their distinct advantages, methodologies, and the outcomes they produced when applied to the dataset.

i. Support Vector Machine

Support Vector Machine, commonly known as SVM, is a supervised machine learning algorithm that can be used for both classification and regression tasks. It works by finding the hyperplane that best divides a dataset into classes. The strength of SVM lies in its ability to handle high-dimensional data, making it suitable for gene expression datasets.

In this project, the team utilized SVM to classify samples into the two groups identified in assignment 1. Code was employed to load and preprocess the gene expression data and a SVM model was then trained on a subset of this data, with its performance being subsequently evaluated on a test set. The confusion matrix, which will be presented further below, offers insights into the model's accuracy and potential misclassifications.

Additionally, the team investigated the influence of the number of genes on the SVM model's performance. By training the model with different gene counts (10, 100, 1000, 10000), variations in the model's accuracy were observed. Detailed results from these experiments, such as most influential genes and the model's accuracy across different gene counts will be showcased below.

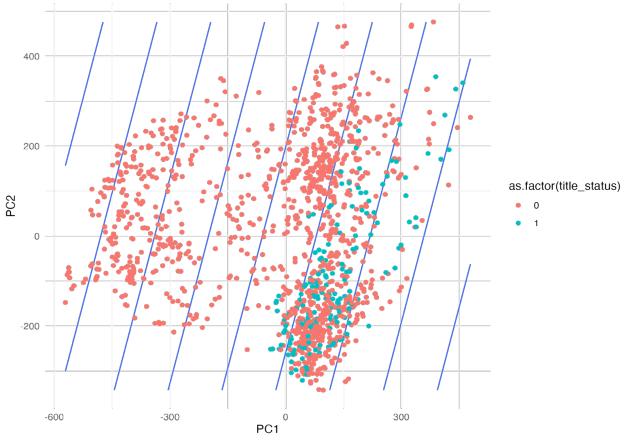


Figure 15 - PCA biplot graph (5000 genes)

Genes	10	100	1000	10000
AUC	0.5164991	0.8687146	0.8324008	0.9955556

Table 15 - AUC Values at Different Gene Counts for SVM

ii. Logistic Regression

Logistic regression is a vital statistical and machine learning technique primarily used for binary classification tasks. It predicts the probability of an event occurring based on input variables, making it ideal for situations where you want to categorize outcomes into two classes. In the context of our research, this would apply to the two groups outlined in assignment 1.

Given the 5,000 most variable genes used in assignment 3, we split the samples into training and test sets at an 80-20 split. This selection process ensures that the model focuses on genes that exhibit significant variability across the samples, optimizing its ability to discriminate between the two defined groups. We chose 80-20 as the train test split due to its frequent use in machine learning. Once the data was separated into the two sets, I trained a logistic regression model with training data of 1362 samples. Then I predicted the results of the test set of 330 samples. The results of the model ran with a gene count of n=5000 can be seen in the following confusion matrix.This model resulted in roughly 50% accuracy when trained with 5000 genes.

	False	True
False	150	22
True	145	22

Table 16 - Log Regression Confusion Matrix (5000 genes)

To evaluate the performance of logistic regression across different gene counts, I retrained the model with gene counts of 10, 100, 1000 and 10000 and examined the AUC values for each.The AUC (Area Under the ROC Curve) summarizes a binary classifier's overall ability to discriminate between classes. A higher AUC indicates better classification performance, making it a vital evaluation metric. The resultant AUC values can be seen varying, rising and falling. Logistic regression yielded the highest AUC value when trained with a gene count of 1000.

	10 genes	100 genes	1000 genes	10000 genes
AUC	0.6126348	0.9178382	0.9228043	0.4802252

Table 17 - Logistic Regression AUC Values

	False	True
False	277	7
True	14	41

Table 18 - Log Regression Confusion Matrix (1000 genes)

In addition to the varying auc values, the gene signatures varied across the different iterations. The most influential genes that appeared were X437, X946, X239, X785 and X648.

	Values
X437	0.33467131
X946	0.21615597
X239	0.20874603
X785	0.19800492
X648	0.19556481
X896	0.19461088
X949	0.19364464
X811	0.19179198
X689	0.18897280
X986	0.18487718
X696	0.18388236
X787	0.18347683
X794	0.17747432
X985	0.17061450
X428	0.16787706
X211	0.16650463
X134	0.16555206
X431	0.16495160
X647	0.15979819
X570	0.15727699

Table 19 - 20 Most Significant Genes in Logistic Regression

While logistic regression was able to predict with auc above 0.9 when trained on a gene count of n=1000, logistic regression is not an ideal prediction algorithm given its struggles with the high dimensionality and multicollinearity in the RNA seq data. These issues may be addressed by dimensionality reduction techniques or other data preprocessing approaches. However, given the scope of this project, we did not choose to implement these ideas.

iii. Random Forest

Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputs the class that is the mode of the types for classification or mean prediction for regression. Its ability to handle large datasets with higher dimensionality makes it apt for gene expression data, and is why it was employed in the current project.

For this research, the team applied the Random Forest algorithm to classify the samples into the two distinct groups mentioned in assignment 1.

In a similar vein to the other algorithms, the team delved into understanding how the number of genes impacted the performance of the Random Forest model. Training the model on varying gene counts (10, 100, 1000, 10000), changes in accuracy could be observed. The following tables delve deeper into these results, highlighting the most influential genes and the model's performance metrics across different gene counts.

	Variable	Importance	index_num	gene_symbol
1	X2486	0.36410965	2486	Enpp2
2	X2424	0.35726457	2424	Atf3
3	X1751	0.34208553	1751	Ccn3
4	X139	0.26178473	139	Lmo4
5	X4509	0.26050231	4509	Gpr88
6	X2468	0.25699002	2468	Hbb-b1
7	X122	0.23730054	122	Tmsb4x
8	X4884	0.23639500	4884	Klf4
9	X1316	0.23186599	1316	Nectin3
10	X352	0.23161586	352	Fam131a
11	X1055	0.22820490	1055	Cacng2
12	X342	0.22557579	342	Lypd1
13	X3161	0.22195255	3161	Palmd
14	X4836	0.21979407	4836	Kctd4
15	X3479	0.21806916	3479	Rbm24
16	X231	0.21785543	231	Lamp5
17	X103	0.21687987	103	Sub1
18	X2884	0.21646157	2884	Pcdh8
19	X237	0.21599444	237	Itm2c
20	X4216	0.20832464	4216	Kcnk2

Table 20 - 20 most Significant Genes in Random Forest

Table 16 showcases the 20 most important genes and their index number within the top 5000 most significant genes. It showcases that the variable with the most significance is Enpp2. It is also interesting to note that there are many indices within the top 1,000.

AUC	Number of Genes
0.8161776	10
0.9637218	100
0.9829431	1000
0.9518566	5000
0.9438989	10000

Table 21 - Random Forest AUC Comparison

The AUC, or Area Under the Curve, values for predicting title_status other shown above indicate that the model with 1,000 genes is the most optimal as it indicates the most robust predictive performance. It also demonstrates diminishing returns at larger datasets, slightly decreasing at 5,000 and 10,000. And the lowest number of genes shows that ten genes do not capture enough information to be practical.

```
common_genes
Itm2c Lamp5  Lmo4  Pcp4
    1     1     1     1
```

Table 22 - Overlapped Genes with Importance > 0.15

The Table above showcases which genes with an importance of more than 0.15 are present in the 1000, 5000, and 10,000 most variable genes. Interestingly, there is 0 overlap between all five variations when limited to 0.15 or high importance level.

iv. Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes theorem. It's particularly suitable for high-dimensional datasets and was therefore a good option to be explored. In this project, the team applied the Naive Bayes algorithm to classify samples into two groups based on gene expression data seen in assignment 1.

By enhancing the current code, the data was first loaded, processed and a Naive Bayes model was then trained on a subset of the data with its performance being evaluated on a test set. The confusion matrix provided insights into the model's accuracy and misclassification rates.

Furthermore, the impact of the number of genes on the model's performance was explored. By training the model on varying numbers of genes (10, 100, 1000, 10000), the team observed changes in the model's accuracy, which will be explored in the tables below.

Predicted	other	PM
other	217	3
PM	80	40

Table 23 - Naive Bayes Confusion Matrix (5000 genes)

	10 genes	100 genes	1000 genes	10000 genes
AUC	0.7005017	0.864053	0.888957	0.92478

Table 24 - AUC Values at different gene counts

AUC values varied by the number of genes included in training. Rising from a value of 0.70 with 10 genes to a value of 0.92 with 10000 genes, this indicates that training with more genes allowed the model to be generalized in a more applicable way.

VIII. Heatmaps and Dendrograms

Heatmaps and dendrograms serve as powerful visualization tools in the realm of bioinformatics, enabling researchers to intuitively grasp the intricate patterns and relationships present within gene expression datasets.

A heatmap offers a visual representation of gene expression levels across samples, where each row typically corresponds to a sample and each column represents a gene. The color intensity within each cell reflects the gene's expression level in the corresponding sample, allowing for a quick and comprehensive overview of the data. By employing heatmaps, the group can gain insights into the predictions and classifications made by each of the four supervised machine learning algorithms.

Dendrograms, on the other hand, provide a hierarchical representation of the relationships between samples or genes. By clustering similar entities together, dendrograms can reveal underlying patterns and groupings within the data, further enhancing the interpretability of the results.

In the process of generating these visualizations, it was deemed essential to log-scale the data values, especially when dealing with a vast range of expression levels. This transformation ensures that the visual representation remains clear and interpretable, even when the dataset contains extreme values.

In the subsequent sections, heatmaps and dendrograms generated from the predictions of the Support Vector Machine, Logistic Regression, Random Forest, and Naive Bayes algorithms will be presented and discussed.

i. Support Vector Machine Heatmap

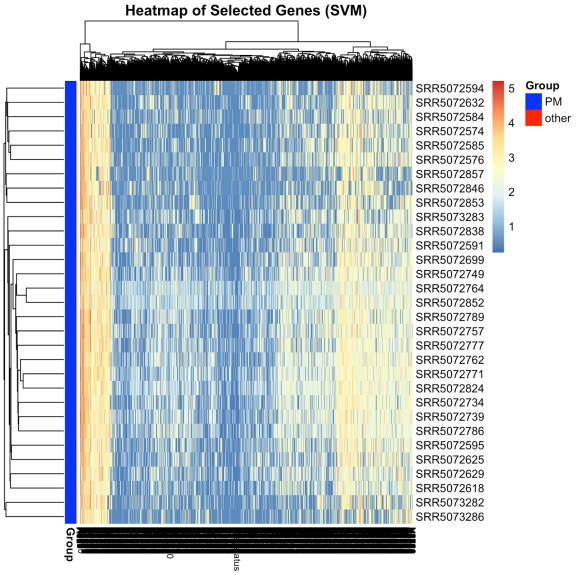


Figure 16 - Support Vector Machine Heatmap

This heatmap visualizes expression patterns of selected genes based on their importance from an SVM model. The genes were selected based on their SVM coefficients, and those with predictions above a set threshold (0.5) were chosen for visualization. The dataset, "test_data", provides the gene expression values which were log-scaled for heatmap visualization. This scaling ensures that the variations in the dataset are visualized effectively, particularly when gene expression values span several orders of magnitude. To supplement the heatmap, an annotation is added to each row, representing the status of the samples. The annotation data is derived from a larger metadata set, which is used to classify samples into two groups: "PM" and "other".

ii. Logistic Regression Heatmap

Heatmaps provide a visual representation of gene expression levels across samples. Each row in the heatmap represents a sample, while each column represents a gene. This allows us to understand the predictions made by the logistic regression model.The generated heat map shows the expression of the 5000 most variable genes across the 127 samples that were predicted to belong to the “PM” group outlined in assignment 1.

When formatting the heatmap, it was important to log scale the values in the data. This made it easier to decipher and understand the data when a large range of values were present.

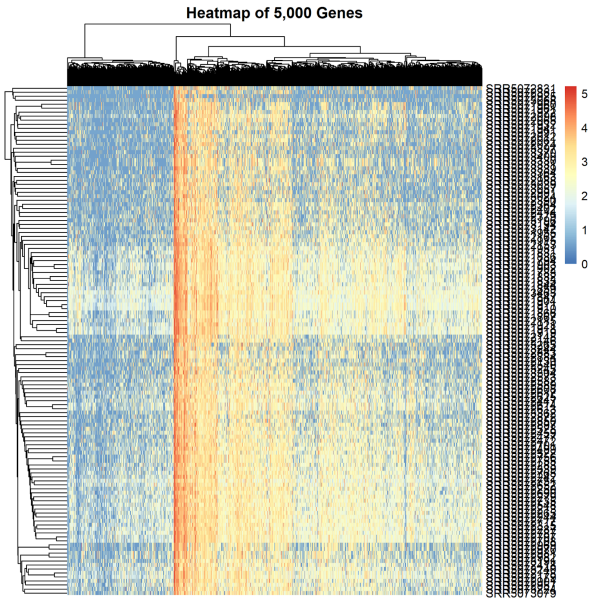


Figure 17 - Logistic Regression Heatmap

iii. Random Forest Heatmap

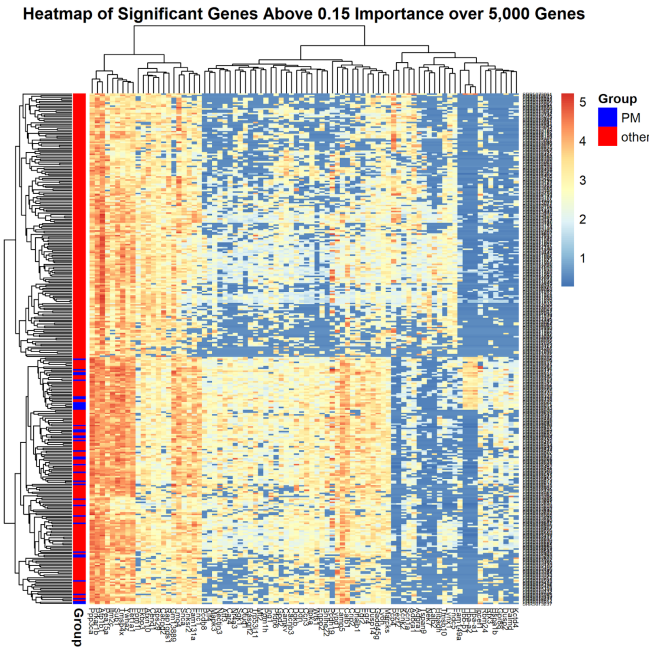


Figure 18 - Random Forest Heatmap

This heatmap showcases significant genes above a 0.15 importance level over the 5,000 most variable genes in the dataset. The random forest algorithm was first run and genes were ranked by their importance.The most significant genes found in the previous experiment were then subsetted, resulting in a subset of 84. Additionally, when looking at Figure 17, the bottom part of the heatmap showcases the gene symbols, and the annotation bar for the groups defined in assignment one is also included in the upper right part of the graph. The heatmap leads to the conclusion that the majority of significant genes identified in this trial belongs to the 'other' title status.

iv. Naive Bayes Heatmap

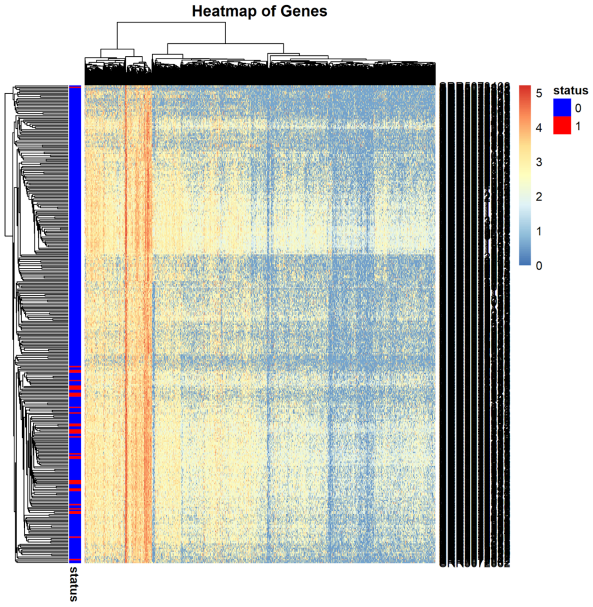


Figure 19 - Naive Bayes Heatmap

The heatmap seen in Figure 18 employs the Naive Bayes classification method to analyze and present the data, highlighting the expression levels through a spectrum of colors where warmer colors indicate higher expression and cooler colors suggest a lower expression. Additionally, a dendrogram clusters the genes and reflects their expression similarities and potential relationships. The significance of each gene is inferred through its expression level, with a focus on distinguishing the genes by their assigned status, which was used to classify genes into two distinct groups.

VII. Conclusions

The integration of the transcriptional profiles with the anatomical and functional properties of individual neurons facilitated a profound understanding of the neuronal networks and their behaviors, underscoring the importance of a multi-disciplinary approach in understanding the components of the nervous system. This multifaceted endeavor exemplified the versatility and efficacy of R in bioinformatics and neurosymbolic studies, highlighting the interconnectedness of genetic, anatomical, and functional domains in the exploration of neuronal diversity.

References

[1] Gu, Z. (2022) Complex Heatmap Visualization, iMeta. DOI: 10.1002/imt2.43

[2]https:/github.com/matheusmaldaner/BioinformaticsProject/blob/main/ensembl-to-hugo.R

[3]https://github.com/matheusmaldaner/BioinformaticsProject