36469 Final Project

Jerry Li (jerryl) and Raymond Li (rli3)

12/12/2021

Introduction

The data set being used for this analysis is the Zeisel data set. The data set contains single cell RNA sequencing data, including gene expression data for thousands of genes and cell type labels. Altogether, the data set has 3005 observations (rows) of 1002 attributes (columns). A preliminary importing task involved setting the row name to the cell name rather than the row indices, making the total number of columns 1001. When performing group-bys, the data set contains seven cell types: astrocytes_ependymal, endothelial-mural, interneurons, microglia, oligodendrocytes, pyramidal CA1, and pyramidal SS. The smallest group is microglia, which contains 98 cells, and the largest, pyramidal CA1, which contains 939 cells. This reference information is useful because when we do our classification, we will have a rough idea of how many separate clusters to expect, as well as the relative size of each cluster.

As mentioned, the Zeisel data set is most compatible with clustering analysis. In this report, two types of clustering will be done. The first is k-means clustering of the seven cell types. With this strategy, both naive and sparse k-means will be done and the results will be compared. Since there are so many genes to cluster within seven cell groups, it is interesting to see if the sparse k-means clustering does a better job dividing the output into seven distinct groups, like we expect it to. An additional wrinkle with k-means is that by default, the euclidean distance is used to find the distances between points. A spherical k-means clustering will also be used and compared in the naive run. However, due to the time it takes to calculate the sparse PCA, it will not be feasible to apply the formula to both the euclidean and the spherical formulas.

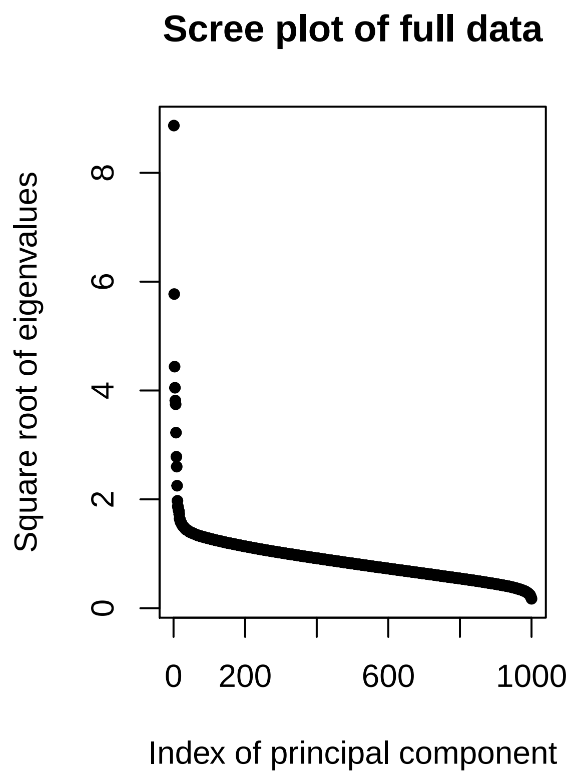
The second type of analysis is using the XGBoost package in R (extreme gradient boosting). The package supports many different types of functions, but the method to be used here will be tree learning for classification purposes. The goal of the second analysis is to view which gene is the most important for various cell groups- by volume of occurrence. In addition, cell types will also be predicted based on the known rate of occurrences of certain genes. The two iterations run here are the basic run (with parameters filled in with commonly accepted values) and a run with an added parameter L2 regularization with a value of 0.1. The theoretical objective of adding this new parameter is to lower variance, while increasing bias. This is a common way to counter overfitting of the model. The results of the two runs are compared.

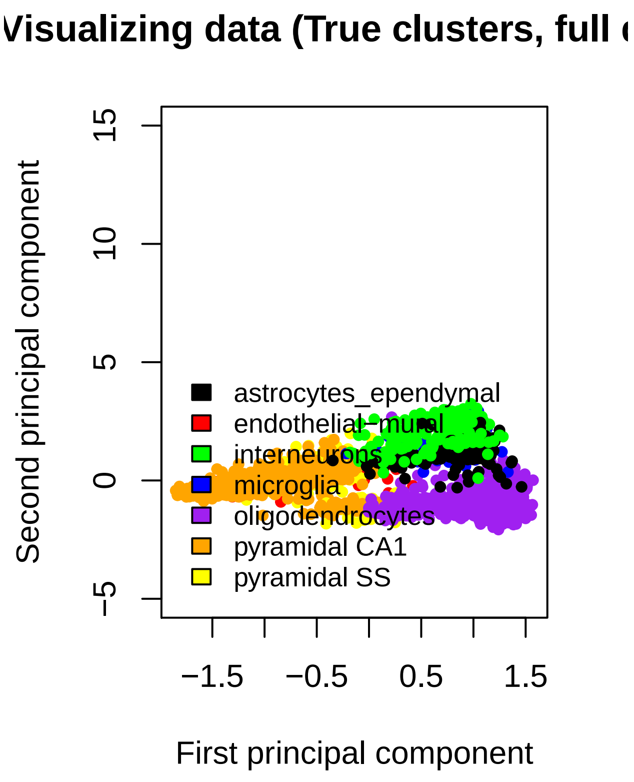
```
## cell_types
## astrocytes_ependymal    endothelial-mural          interneurons
##                  224                  235                   290
##            microglia     oligodendrocytes         pyramidal CA1
##                   98                  820                   939
##        pyramidal SS
##                  399
```

We see that the dataset contains 7 types of labelled cells, with pyramidal CA1 being the most common and microglia being the least common.

Results

PCA Clustering Analysis





The scree plot shows about 10 points that may be considered "outliers", as they are disconnected from the rest of the lines. This indicates we most likely do not need to use beyond the first ten principle components, but we may choose to use fewer than ten.

The cluster graph shows three out of the seven cell types (pyramidal CA1, interneurons, oligodendrocytes) placed in a visible cluster. The astrocytes_ependymol (in black) also appears to be clustered in a group, but is overshadowed by the green-colored group- and it's hard to distinguish the two. The three remaining groups looks to be quite scattered. However, that may just be due to the limitation of the data set. For example, certain cell types of 10x more data points than other cell types.

```
##    cell_types
##     astrocytes_ependymal endothelial-mural interneurons microglia
##   1                    2                 5           23         0
##   2                    0                 0            8         1
##   3                    7               223            4        90
##   4                  214                 3            4         6
##   5                    0                 0            4         0
##   6                    1                 1            0         1
##   7                    0                 3          247         0
##    cell_types
##     oligodendrocytes pyramidal CA1 pyramidal SS
##   1                0           323          279
##   2               89            43           37
##   3               17             4           10
##   4               18            18            7
##   5                0           546           64
##   6              696             0            0
##   7                0             5            2
```

```
## [1] 0.2658902
```

Using the normal euclidean distance k-means on the expression matrix, it seems like cell groups 4-7 are classified really well. However, groups 1-3 have two or three decently large groups of cell types. Groups 1 (pyramidal CA1, pyramdial SS) and 3 (endothelilal-mural, microglia) seems to especially have a hard time clustering one cell type in its grouping. Pyramidal CA1 cells also have very large groupings in both groups 1 and 5. The mis-calculation rate for this analysis is 26.6%.

```
##    cell_types
##     astrocytes_ependymal endothelial-mural interneurons microglia
##   1                    0                 3          267         0
##   2                    0                 0            1         0
##   3                  214                 4            5         6
##   4                    8                227            5        92
##   5                    0                  0            5         0
##   6                    1                  1            2         0
##   7                    1                  0            5         0
##    cell_types
##     oligodendrocytes pyramidal CA1 pyramidal SS
##   1                2             6            2
##   2                0           446            5
##   3               22            25            9
##   4               20            10           12
##   5                0           430           17
##   6              776             9           18
##   7                0            13          336
```

```
## [1] 0.2459235
```

Using the spherical distance k-means on the expression matrix, it seems like all cell groups with the exception of group 4 was clustered relatively well, balancing between endothelial-mural and microglia type cells. Going down the columns, it seems like all cell types were grouped pretty well too, with the exception of pyramidal CA1- splitting its instances between groups 2 and 5. Even though the misclustering rate is 25% (marginal statistical improvement), the eye test shows much improvement when switching over to the spherical distance formula.

```
##    cell_types
##     astrocytes_ependymal endothelial-mural interneurons microglia
##   1                    0                 2            1        93
##   2                    0                 1            4         0
##   3                    7               208            3         3
##   4                  210                 5            7         0
##   5                    1                 0           71         0
##   6                    0                 0            0         0
##   7                    6                19          204         2
##    cell_types
##     oligodendrocytes pyramidal CA1 pyramidal SS
##   1               29             2            6
##   2              151            41           33
##   3               18            15           17
##   4               27            42           13
##   5                0           557          136
##   6              593             0            0
##   7                2           282          194
```

```
## [1] 0.368386
```

Using the default euclidean distance k-means on the expression PCA calculations, it looks like groups 3 and 6 are clustered solidly. All other groups have a decently large proportion of at least two cell-types. And going down the cell types, the first four are grouped well. However, oligodendrocytes, pyramidal CA1, and pyramidal SS cell types are split among multiple cell groups. The misclustering rate for this analysis is 36.8%.

```
##    cell_types
##     astrocytes_ependymal endothelial-mural interneurons microglia
##   1                    1                 6           21        96
##   2                    0                 0          150         0
##   3                  215                 7           17         0
##   4                    0                 0            4         0
##   5                    8               222            7         2
##   6                    0                 0            3         0
##   7                    0                 0           88         0
##    cell_types
##     oligodendrocytes pyramidal CA1 pyramidal SS
## 1                 40            15            21
## 2                  0           199           138
## 3                 40            63            20
## 4                 53            46            33
## 5                 44            32            33
## 6                643             0             3
## 7                  0           584           151

## [1] 0.353411
```

Using the spherical distance k-means on the expression PCA calculations, it looks like only group 6 is closely associate with a cell type- meaning there is a good chance it is being overfit. Other groups have a solid proportion of three to four different cell types. And looking at the cell types, astrocytes_ependymal, endothelial-mural, and microglia cell types looks like they are closely associated with a grouping. The other cell types are not. The miscalculation rate here is 35.3% Compared to the default distance formula, the misclustering rate shows a marginal improvement However the table does not indicate a major benefit in terms of better clustering, and in some aspects, even looks worse than before.

Overall, the PCA calculations hold a higher miscalculation rate from just the matrix, regardless of which distance formula is used. When switching over to the spherical distance formula, there is an improvement in the miscalculation rate. However, the results of the tables are mixed, showing improvements at times and no improvements at other times.

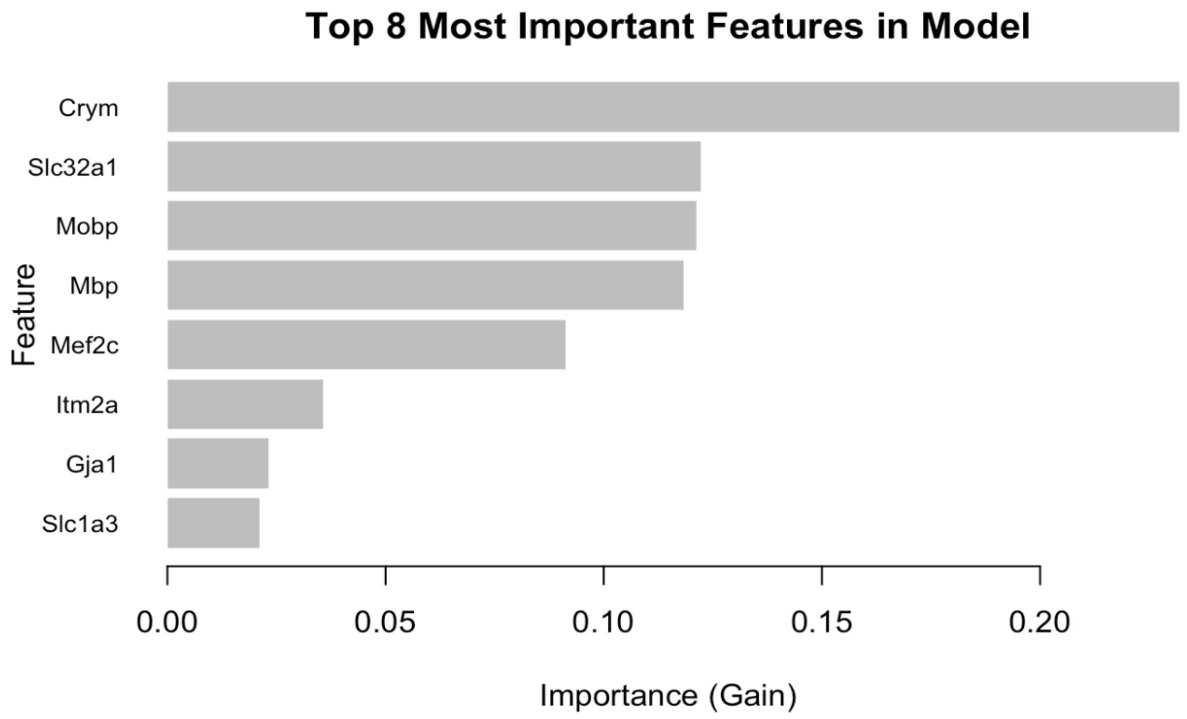
```
##    cell_types
##     astrocytes_ependymal endothelial-mural interneurons microglia
##   1                    0                 2            1        93
##   2                    0                 0            5         0
##   3                    7               198            4         2
##   4                  206                 5            7         0
##   5                    0                 0           82         0
##   6                    0                 4            0         1
##   7                    11                 26          191          2
##    cell_types
##     oligodendrocytes pyramidal CA1 pyramidal SS
##   1               34             4            6
##   2              671            16           23
##   3                4            14           14
##   4               28            45           14
##   5                0           654          160
##   6               81             0            2
##   7                2           206          180

## [1] 0.3294509

##    cell_types
##     astrocytes_ependymal endothelial-mural interneurons microglia
##   1                    0                 4           14        96
##   2                    0                 0          124         0
##   3                  213                 6           22         0
##   4                    0                 0            5         0
##   5                   11               225            7         2
##   6                    0                 0            2         0
##   7                    0                 0          116         0
##    cell_types
##     oligodendrocytes pyramidal CA1 pyramidal SS
##   1               43            10           15
##   2                0           114          101
##   3               40            65           20
##   4               47            44           34
##   5               40            29           33
##   6              650             0            2
##   7                0           677          194
## [1] 0.3281198
```

When using the sparse PCA calculations applied to both the euclidean distance and the spherical distance, the miscalculation rate improves marginally (32.9%, 32.8%). However, this time, there is no noticeable improvement between the two distance formulas. When viewing the tables, in both cases it seems like half of the cell groups are classified well, and the other half is not. Overall the table does not indicate improvement between distance types or sparse PCA vs vanilla PCA. That being said, based on the formula applied in HW6, there seems to be a mathematical improvement when moving to sparse PCA.

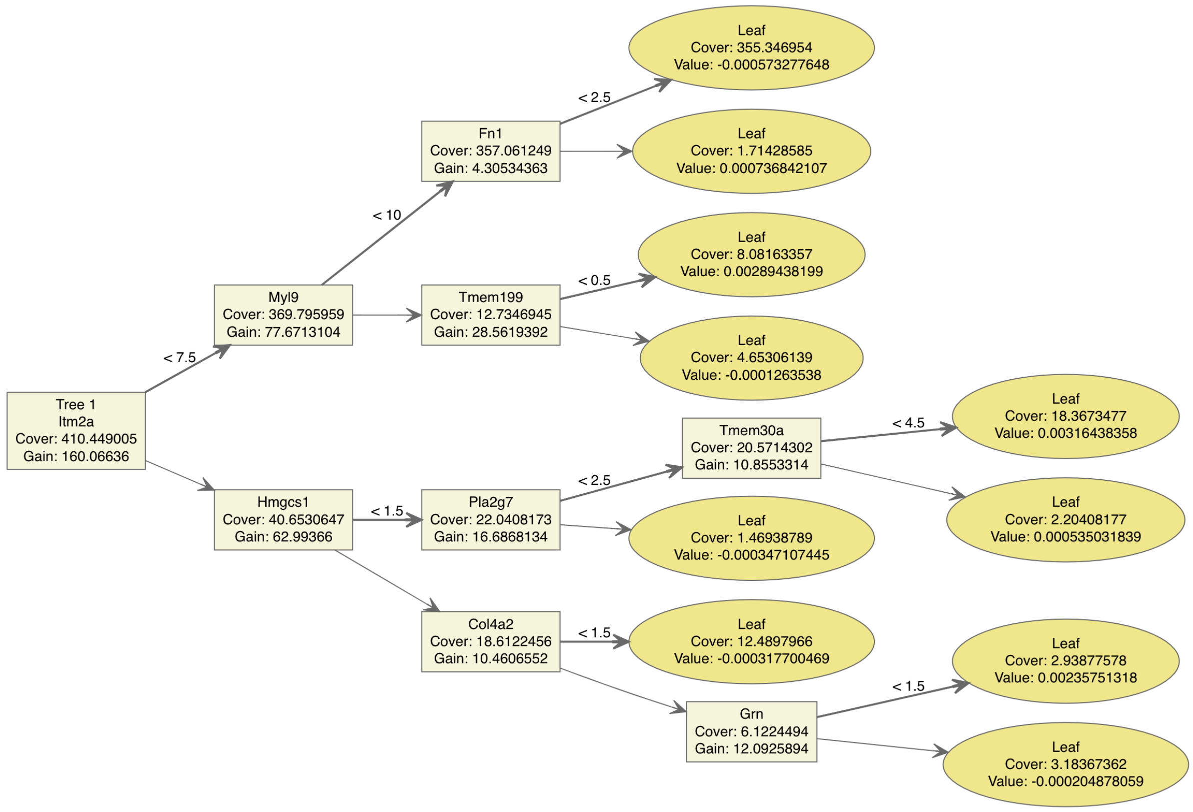
Now viewing the classification plot when using sparse PCA, there does seem to be an improvement from before. The most noticable aspect is the black group (astrocytes_ependymal) clearly being separated from the green (interneurons). Other than that, the orange, purple, and green are still distinct. The blue (microglia) group is still very scattered and not clustered at all. However, that may again be due to its small sample size. Interestingly, the red (endothelial-mural) and yellow (pyramidal SS) groups look like they are clustered, but are hidden behind the orange clusters. In any case, it does not appear that the three groups have a great separation from each other.

XGBoost Classification

As aforementioned, the full Zeisel dataset collects data on 3005 mouse cells. To conduct XGBoost classification, we first randomly split the data into training (75%) and testing (25%), i.e. Leave-One-Out Cross Validation (LOOCV), which will allow us to verify the model's findings.

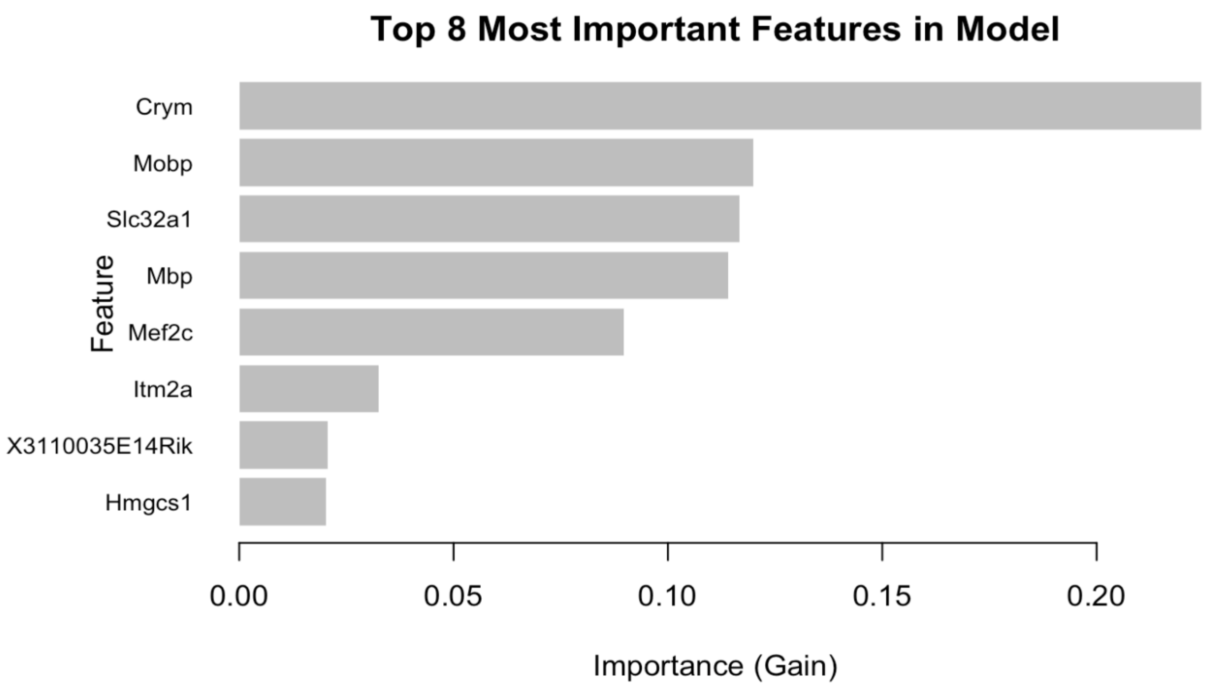


The first model we fit with basic specifications (as mentioned in the **Methods** section) produces the above importance graph, which shows the top 8 most important features in the model. The `Crym`, `Slc32a1`, `Mobp`, `Mbp`, and `Mef2c` genes stand out as the features that produce the highest information gain, suggesting that they are most important in determining cell type. This is reflected in the first XGBoost decision tree shown below, which uses some of the most important features as branching decisions.

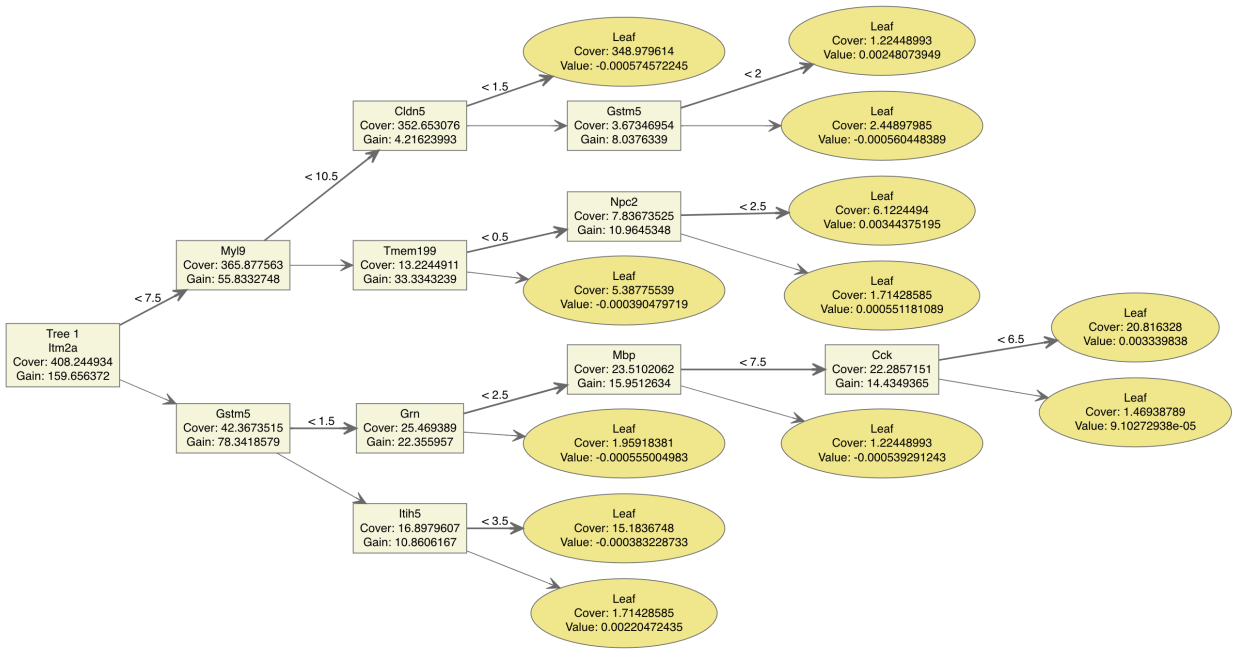


The final accuracy of this model was quite high with a very small training error of 2.09% and testing error of 5.98%, which are both incredibly low. However, despite the fact that the training and testing error are both small and have a marginal difference, the testing error is higher than the training error. In addition, the tree supports this idea that the model may be overfitting, as there seem to be a high number of splits and some that seem almost arbitrary. Thus, we may choose to add an L2 regularization term of 0.1 to smooth parameters toward 0 and avoid overfitting.

Visualizing the most important features, we observe that some of the top 8 have changed between the two models. For instance, the `X3110035E14Rik` gene stands out as one that has now made the top 8 features and has a funny name. This may be because some features that are included at many lower splits which would provide more information are now penalized, thereby reducing their importance in the regularized model. That being said, 6 of the same genes are still in the top 8, such as `Mobp` and `Slc32a1`, with `Crym` being the most important feature across both models.



The first decision tree, as shown below, is also interesting–the number of splits seems to have increased, if anything, as compared to the non-regularized model. However, this may be explained by the coefficients on each feature shrinking toward 0 and not necessarily being altogether eliminated from the model, which would be the case with L1 regularization.



Interestingly, including L2 regularization produced a more significant difference between training and testing error–we see that training error is now 1.46% whereas testing error is 5.98%. This means adding L2 regularization actually seemed to increase the out-of-sample bias, but the small differences in error may be attributable to the random 75-25 train-test split. While this regularization attempt has failed, it is nonetheless interesting to analyze the differences between the two models.

Discussion

From the XGBoost models, we saw that the most important features across both models were `Crym`, `Mobp`, `Slc32a1`, `Mbp`, `Mef2`, `Itm2a`, consistently appearing in the top 8 most important features. These genes, in further detail, are:

Crym: Crystallin Mu

Mobp: Myelin Associated Oligodendrocyte Basic Protein

Slc32a1: Solute Carrier Family 32 Member 1

Mbp: Myelin Basic Protein

Mef2: Myocyte Enhancer Factor 2

Itm2a: Integral Membrane Protein 2A

While at first glance there does not seem to be much association between these genes, we identify that many of them are related to the production of integral membrane proteins. Of these, two are associated with myelin, which is an insulating layer that forms around nerve cells. This may be related to the fact that the mouse cells were all sampled from mouse brains, so the most important differing factors may be associated with cells functions that are specialized for the brain, such as the conduction or maintenance of the central nervous system. This may have important implications for future research, which may delve deeper into how these genes differ by cells.

Aside from this identification of important genes in classifying cells, we have produced a number of models for classification in this paper. Using sparse PCA, the misclassification rate is 32.8%; using XGBoost, both the regularized and non-regularized models have test error rates of around 5%. Thus, we can reasonably conclude that XGBoost may be the more appropriate model for classifying mouse brain cells based on the dataset.

If given more time, we would have liked to conduct cross validation in the XGBoost modelling to select more optimal factors such as `gamma` or tree size. As noted in the **Results** section, our attempt to conduct L2 regularization did not produce notably significant results, so other measures like constraining the size of the decision trees may be attempted to avoid overfitting. Additionally, we would have liked to conduct bootstrapping or resample more data to provide more valid results. Note specifically that the data contained 3005 cell observations across 1000 genes, meaning there were only 3 times more observations than features. While PCA automatically reduces the number of dimensions, models like XGBoost may suffer due to the closeness in dimensionality. Hence, bootstrapping or resampling for more data may improve the validity of our findings.

Methods

KMeans Clustering

To start, the expression matrix was preprocessed by performing three steps. First, each value was normalized through rescaling such that the sum of all scales add up to 10^4. This value is arbitrary but a consistent value is needed for normalization purposes such that analyses between different distance formulas and sparse vs. naive runs can be compared. For every vector y_i, the normalizing step performs $10^4 * \frac{y_i}{\sum_{j=1}^{D} y_j}$ for all i from 1 to D. Next, each cell from the previous step will be log transformed via the transformation $log_2(x+1)$. And lastly, each value will be scaled such that each gene (matrix column) has a mean of 0, and a standard deviation of 1, which effectively normalizes the gene counts.

The output of the processing will replace the values of the original matrix and further visualization and principal component analysis can be done on this result matrix.

XGBoost

XGBoost is a powerful classification model that uses gradient boosting to outperform many other algorithms for classification.

We fit our XGBoost models with a learning rate of 0.001, gamma value of 3, and max depth of 5 to prevent overfitting by making the boosting process more conservative. The booster ran for 20 rounds on the training dataset using the `multi:softprob` model, which predicts probabilities that an observation falls into each of the 7 cell types. This makes use of the softmax objective function, which is described as:

$$\sigma(\overrightarrow{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

As aforementioned, the L2 regularized version of the XGBoost model simply added an L2 lambda term of 0.1 to shrink the coefficients and prevent overfitting.

Note on coding

Please note that some elements of the code created issues when knitting, and therefore we have created images of the plots and inserted them along with code set with `eval=FALSE` to avoid these issues.