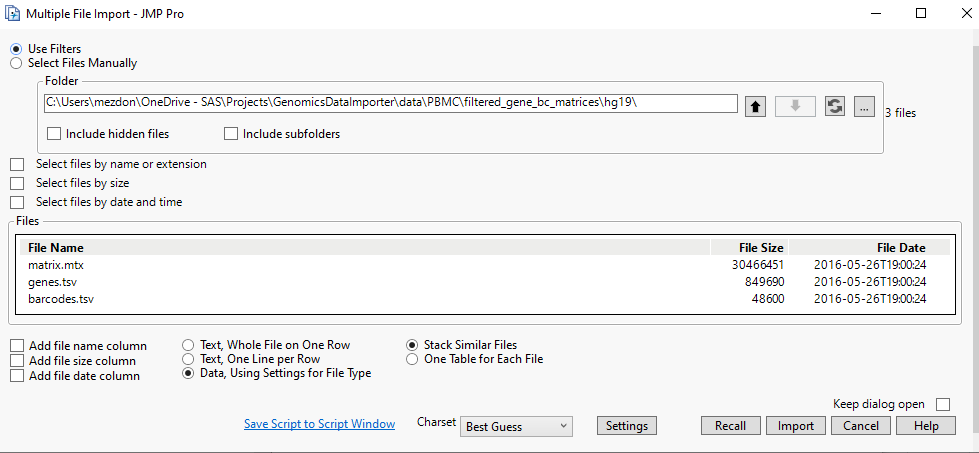
# Single-cell RNA sequencing data analysis in JMP PRO 18.0

## 10X data (<https://satijalab.org/seurat/articles/pbmc3k_tutorial> )

### Import data

File > Import Multiple Files > select the folder where the three files are stored

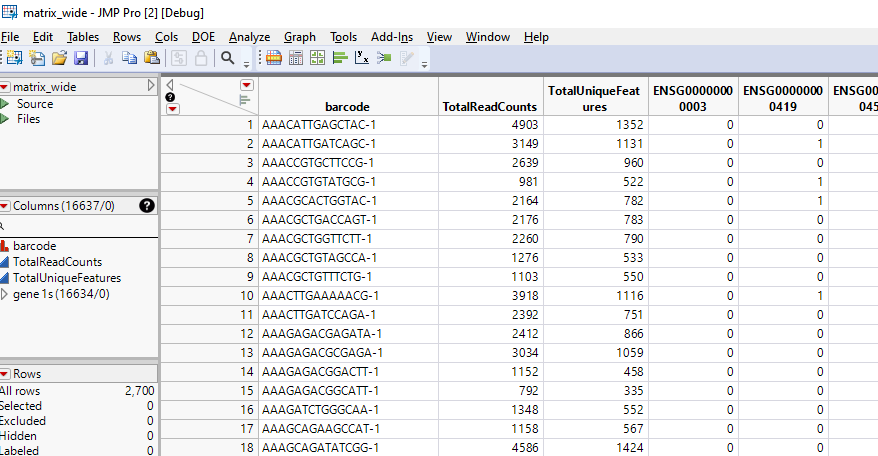
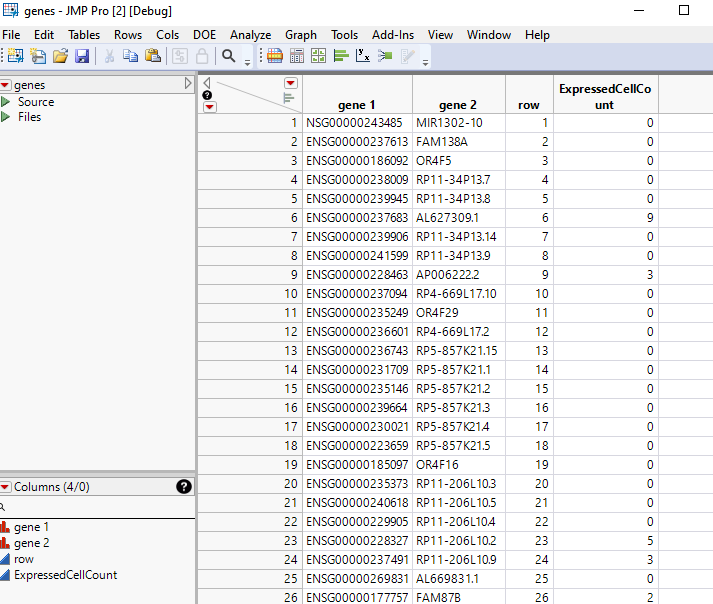


It might take a little time if the data size is large.

Four JMP data tables would show up:

|  |  |  |
| --- | --- | --- |
| JMP table created | Corresponds to | Description |
| Matrix | .mtx file | Sequencing counts in the three-column format. |
| Genes | .tsv file | feature information |
| Barcodes | .tsv file | samples information |
| Matrix\_wide | - | Rows represent samples, and columns represent features. |

### Filter rows (samples) and columns (genes) on matrix\_wide.jmp and genes.jmp

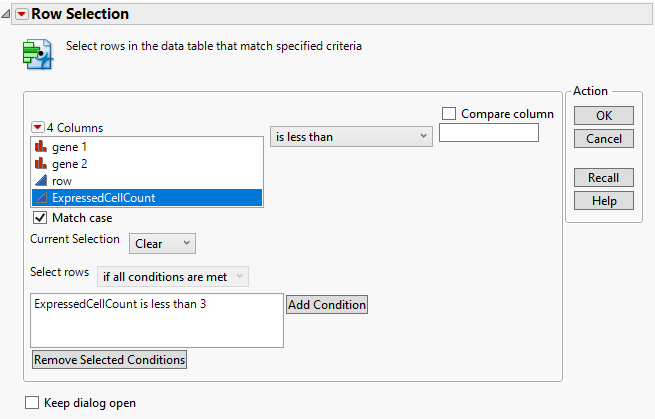
 

#### Update column names

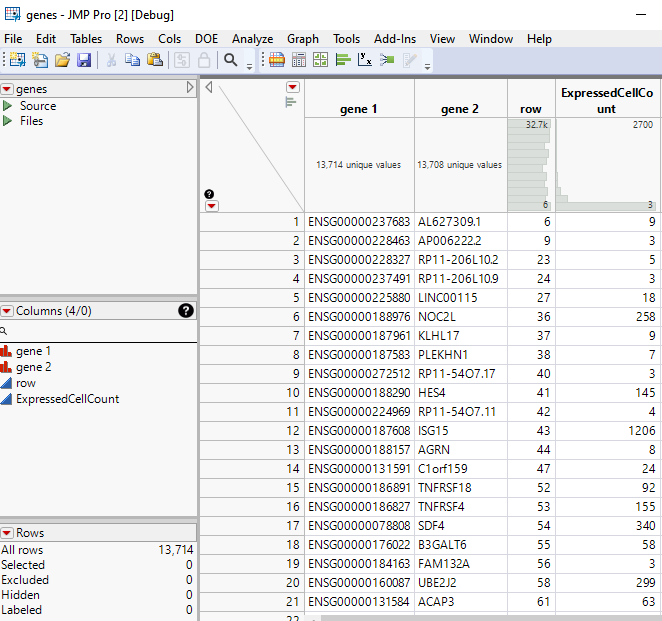
Notice the gene names are in the format of ENSGxxxxxxxx when we first import the data. To perform the downstream analysis, we need to swap the column names using the information from genes.jmp data table.

##### Genes.jmp

Select features that expressed in less than 3 cells: Rows > row selection > select where

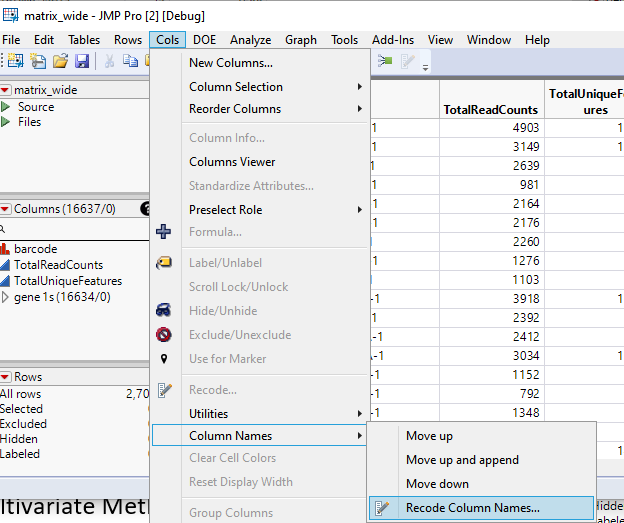


Click OK. Then right click on the selected rows, choose “Delete rows”. 13714 features are left.

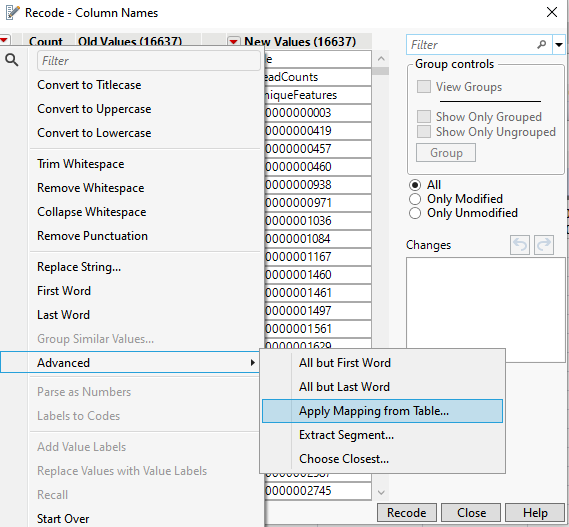
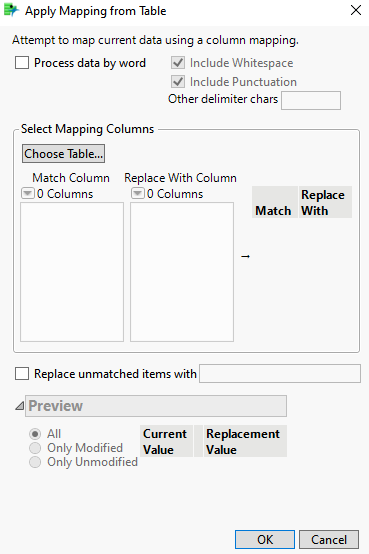


##### Matrix\_wide.jmp

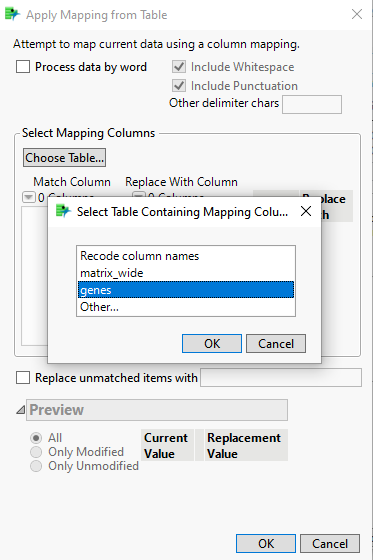
Cols>column names>recode column names.



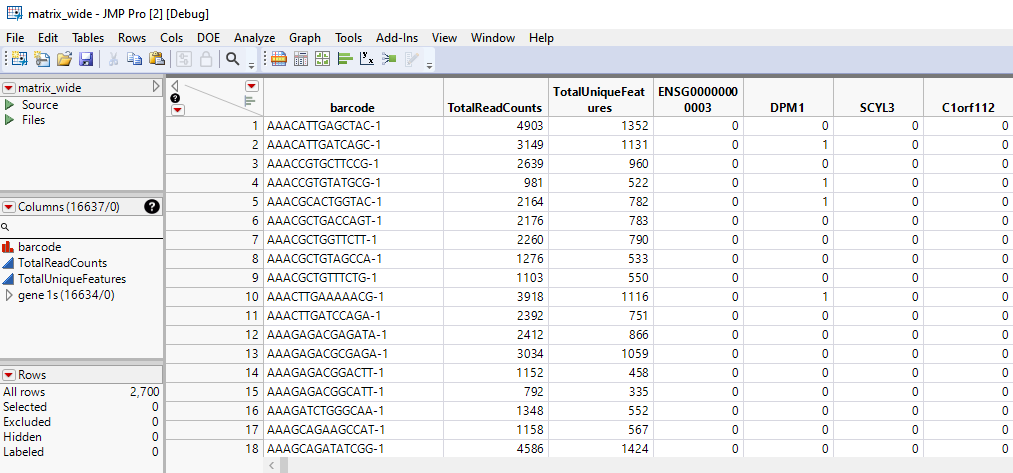
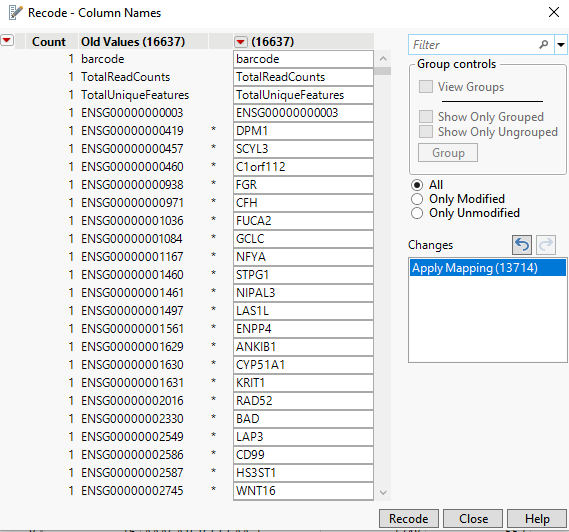
From the pop-up window, click the red triangle from the upper left corner, then advanced>Apply Mapping from Table.

Click choose table and select the data table genes.jmp. Then choose “gene 1” as the Match Column, and choose “gene 2” as the Replace With Column. Click OK.

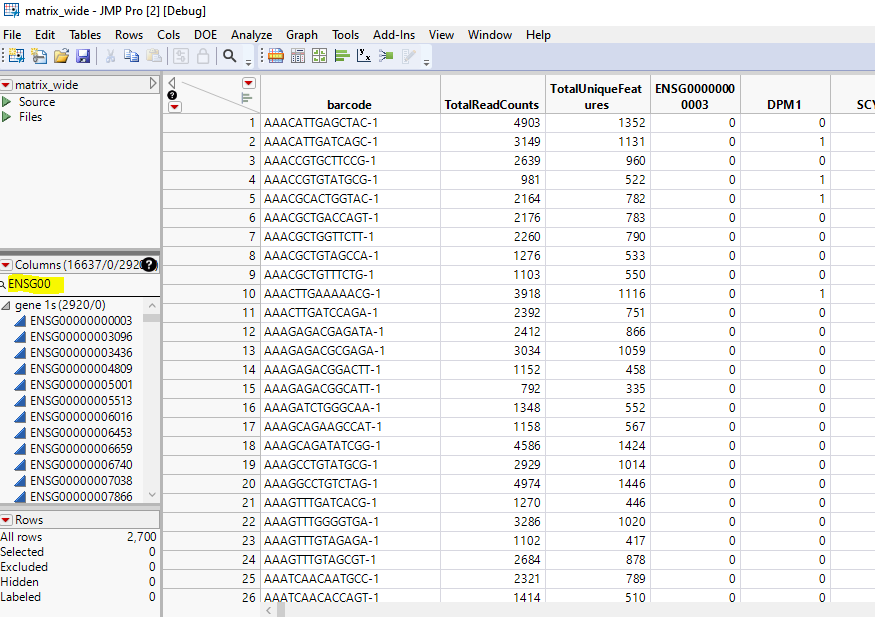
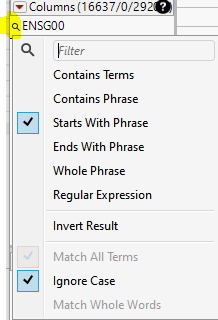
Click recode. The column names are replaced. This process may take several minutes depending on the size of the data.



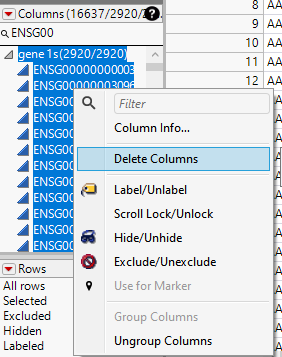
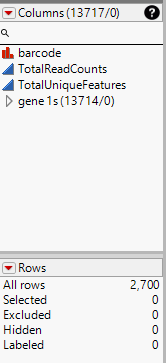
#### Filter out features that expressed in less than 3 cells

Note that we only mapped the gene names where genes expressed in at least 3 cells. Therefore, any features that still have the column name starting with ENSGXXXXXXXXXX will need to be deleted. These genes either have all zeros across samples, or only expressed in a few cells, and thus are non-informative.

From matrix\_wide.jmp, in the columns panel, search for ENSG00, and we see 2920 genes. To make sure we do not accidentally delete other genes, we can click the search sign, and check “Start with phrase”.

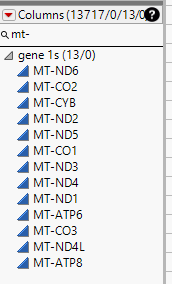
Select all the 2920 columns, and right click on them > Delete Columns. Now we have 13714 genes and 2700 cells left.

#### Calculate Mitochondrial genes percentage

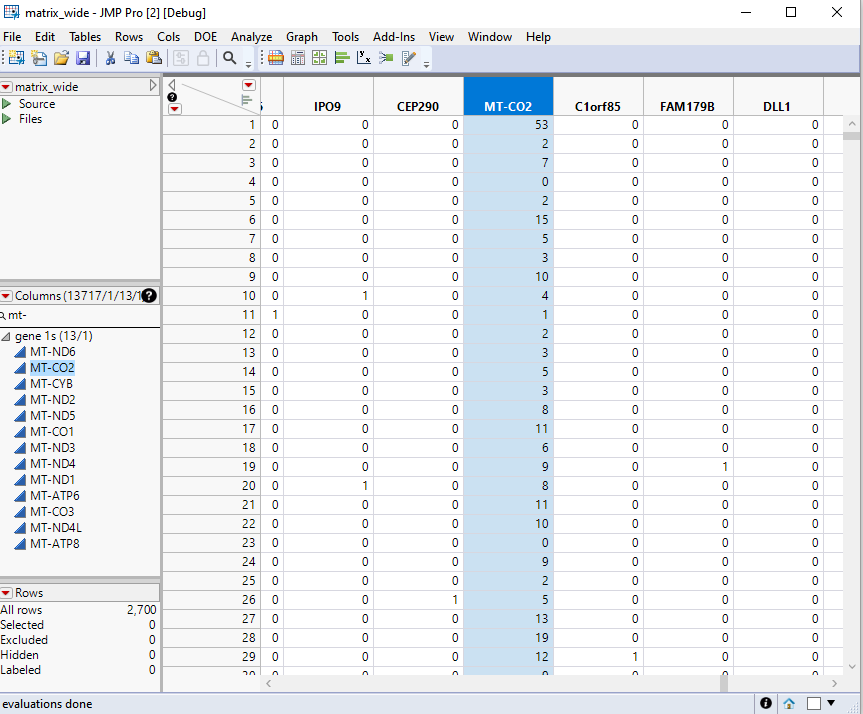
The percentage of mitochondrial genes expression is often used to evaluate the sequencing quality of a cell (sample). The higher the percentage is, the more likely a cell is a poor quality / deceased cell.

Mitochondrial genes usually start with “MT-”. By searching “mt-” in the columns, we derive 13 genes.

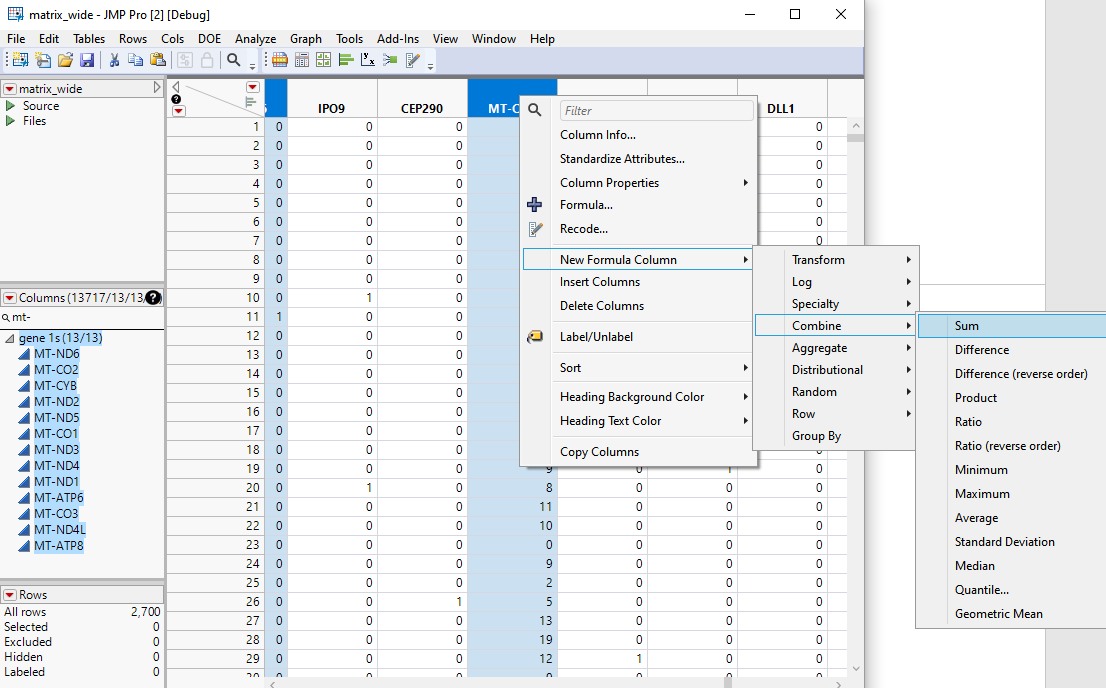


The percentage can be calculated as:

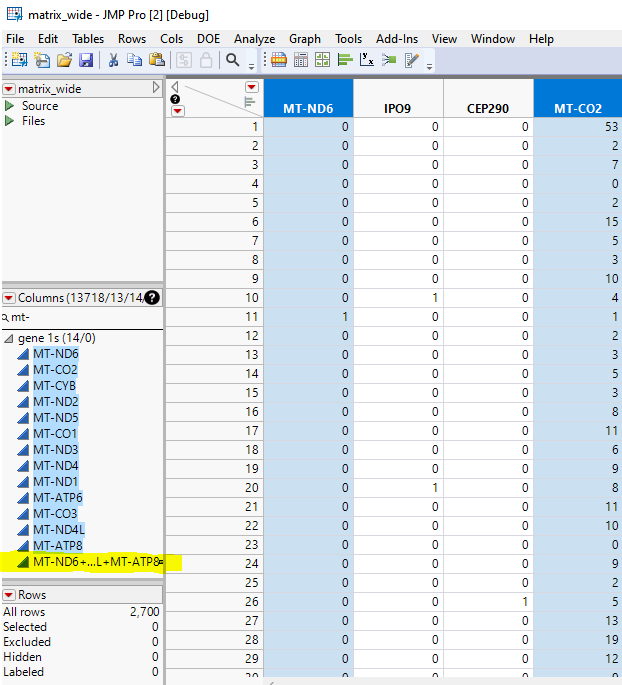
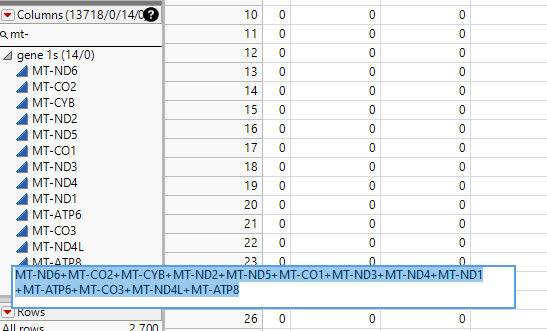
To compute the “sum of MT gene counts”, we first double click one of the MT- genes, for example, “MT-CO2”. Then we will be directed to where “MT-CO2” column is.



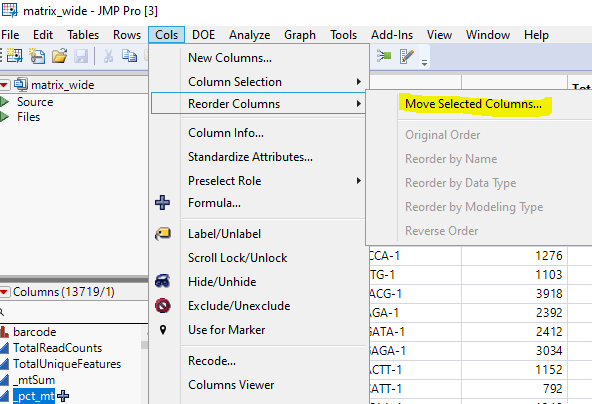
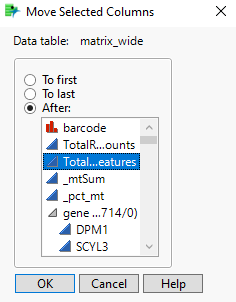
Next, select all the 13 MT- genes, then right click on one of the MT- column names. In this screenshot, we right click on “MT-CO2”. New Formula Column>Combine>Sum.



In this way, we created a new variable that is the sum of all MT- gene counts. Click on this column’s name from the Columns panel, and we can rename it as “\_sum\_mt”, or other names you prefer.

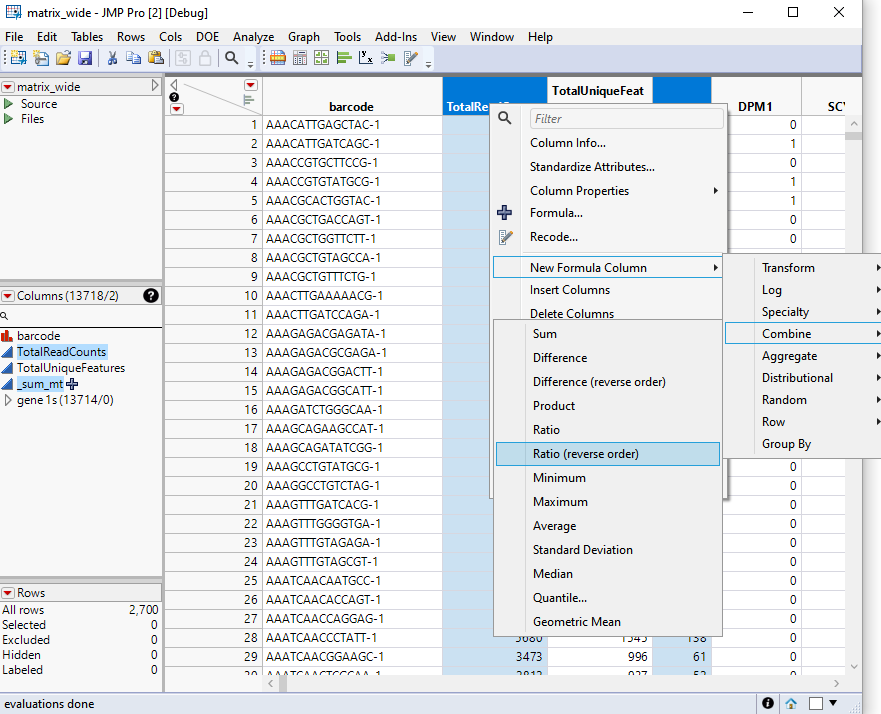
 

Select this “\_sum\_mt”, Cols>Reorder Columns>Move Selected Columns. Move it After “TotalUniqueFeatures”.

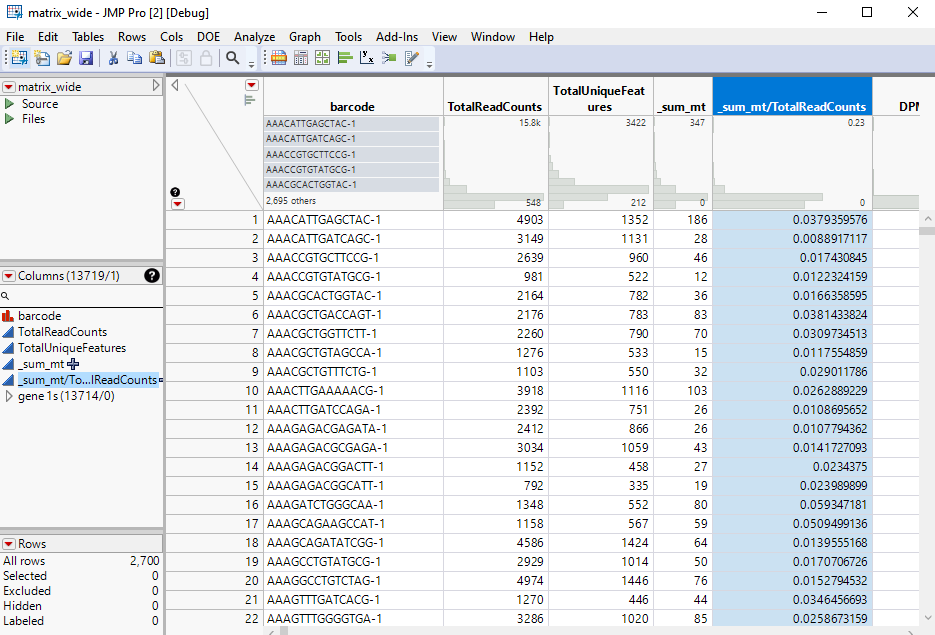
 

(Another way: go to Cols>Ungroup Columns. The “\_sum\_mt” will be separated from the gene groups. You can also drag the column under the “TotalUniqueFeatures” variable, as they are all summary statistics.)

Next, we can create a new column that is the ratio of “\_sum\_mt” and “TotalReadCounts”. Again, we can select the two columns, and right click > New formula column > combine > Ratio (reverse order) (or Ratio, depending on which variable is first listed in your data table).



Now we have created a column “\_sum\_mt/TotalReadCounts”.



#### Filter out samples (rows)

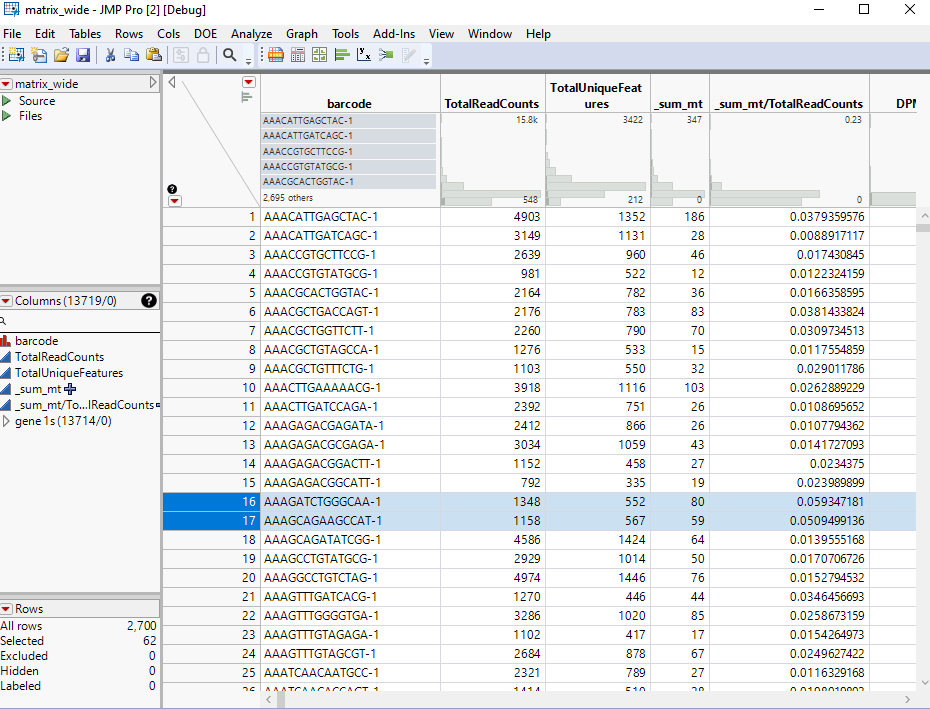
In this example, we filter out samples that satisfy any of the following conditions:

* MT% > 5%
* Total unique features > 2500
* Total unique features < 200

These numbers can be changed according to the research interest.

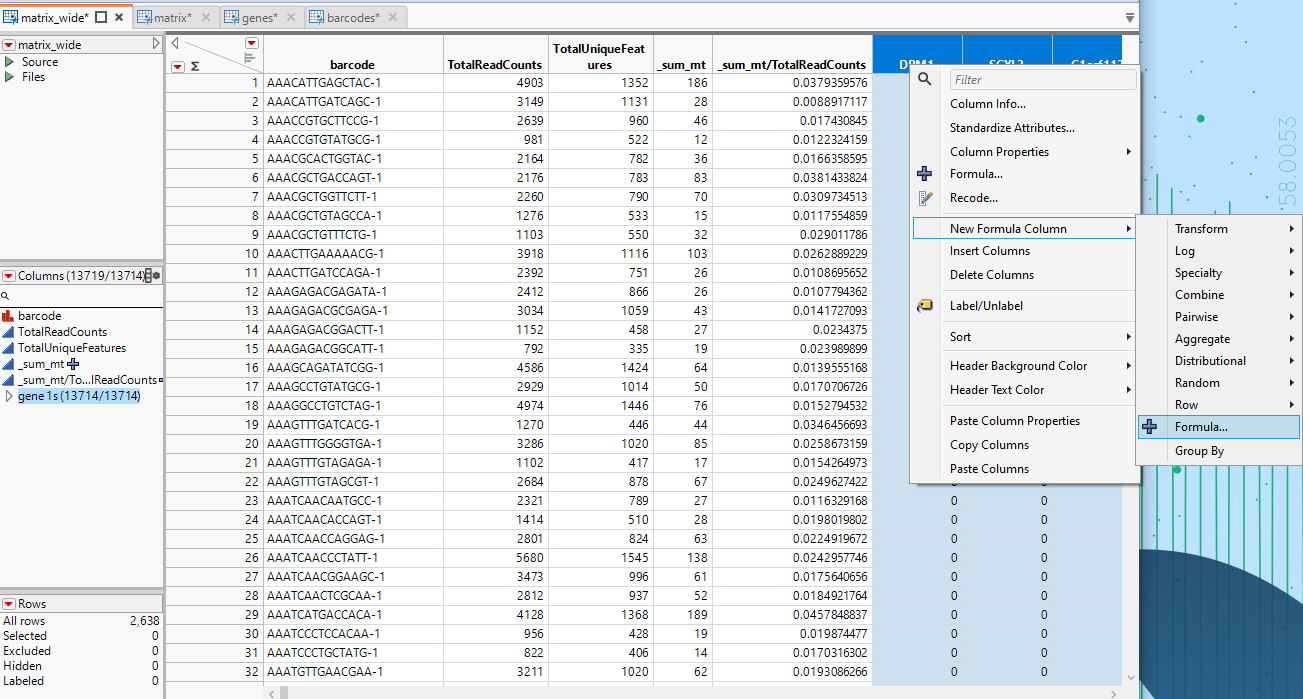


Thus, 62 rows are selected. We can either delete rows or exclude rows.

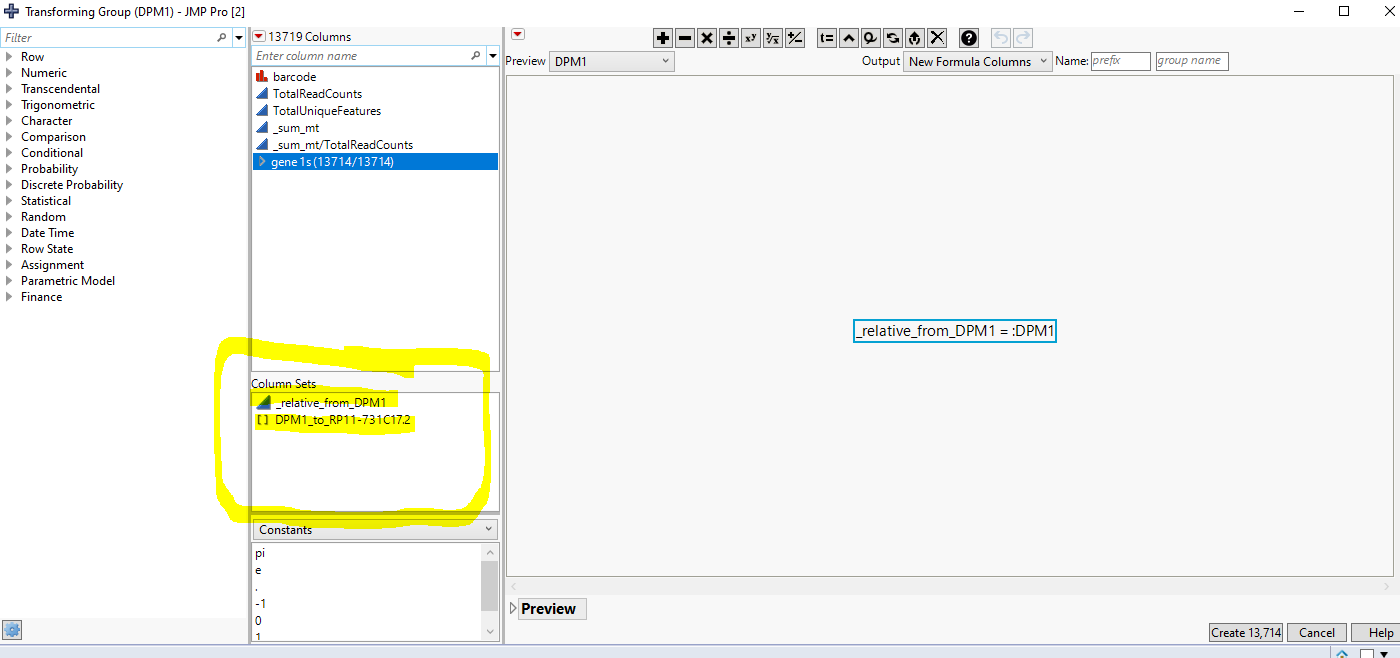


### Normalize data

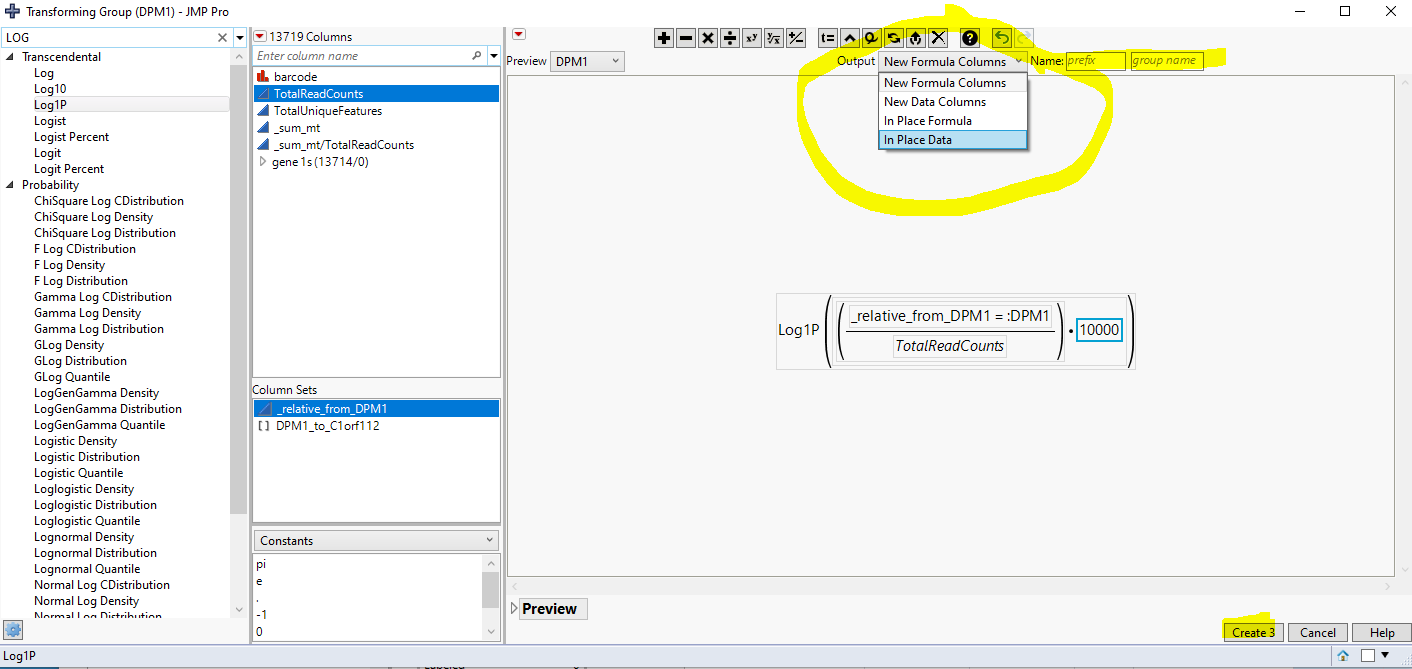
Select all columns of interest, right click on the column names > New Formula Column > Formula.

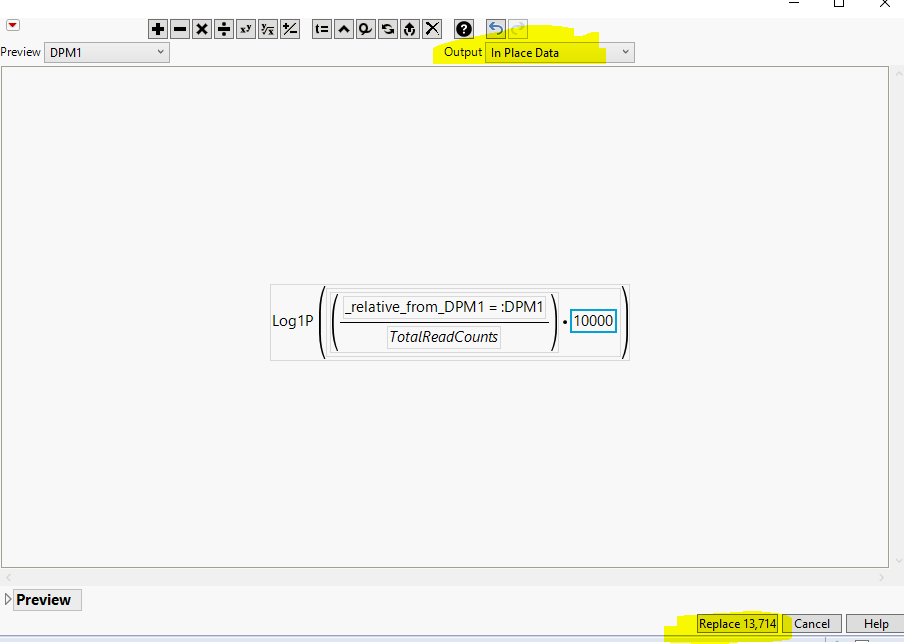


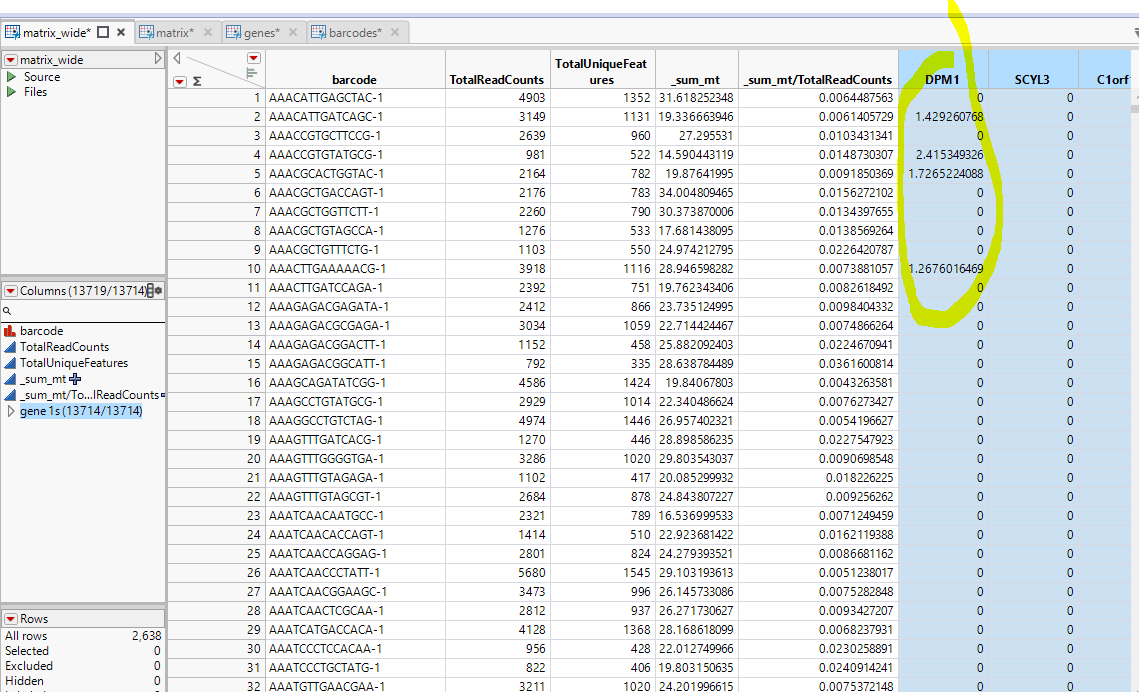
The popup formula window has a section (new feature in JMP18) Column Sets: i) the “\_relative\_from\_DPM1” variable represents a single column from the list of the selected columns; ii) the “[] DPM1\_to\_RP11-731C17.2” is a list of references for all the selected columns.



Edit the desired normalization formula: we can choose to create new columns, or to replace the original numbers in the columns by the normalized numbers.



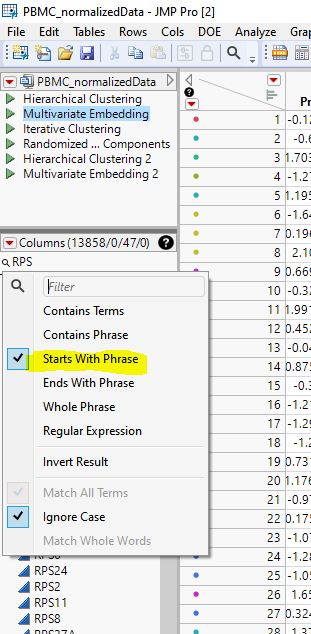
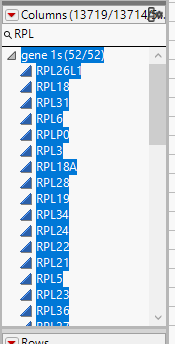
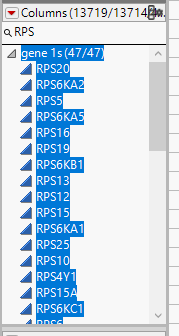
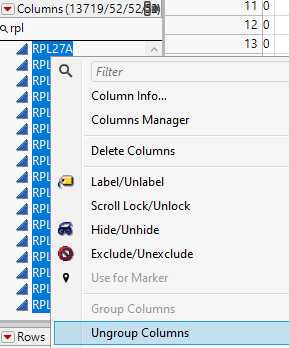


Click “Replace 13714”. This procedure takes several seconds. Now we have the normalized counts: 

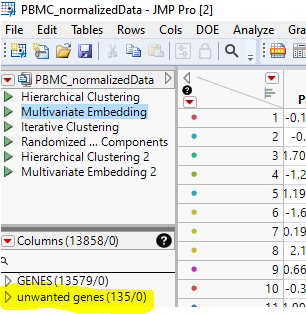
### Filter out genes before downstream analysis

After Normalizing the data, we can remove some unwanted genes, for example, ribosomal genes (**RPLXXX**, **RPSXXX**), mitochondrial genes (**MT-XXX**), heat shock protein genes (**HSPXXX**), etc.

To do this, we can easily find genes which start with “RPS” by inputing RPS in the “columns” panel, and then click on the search sign and check “Starts With Phrase”.

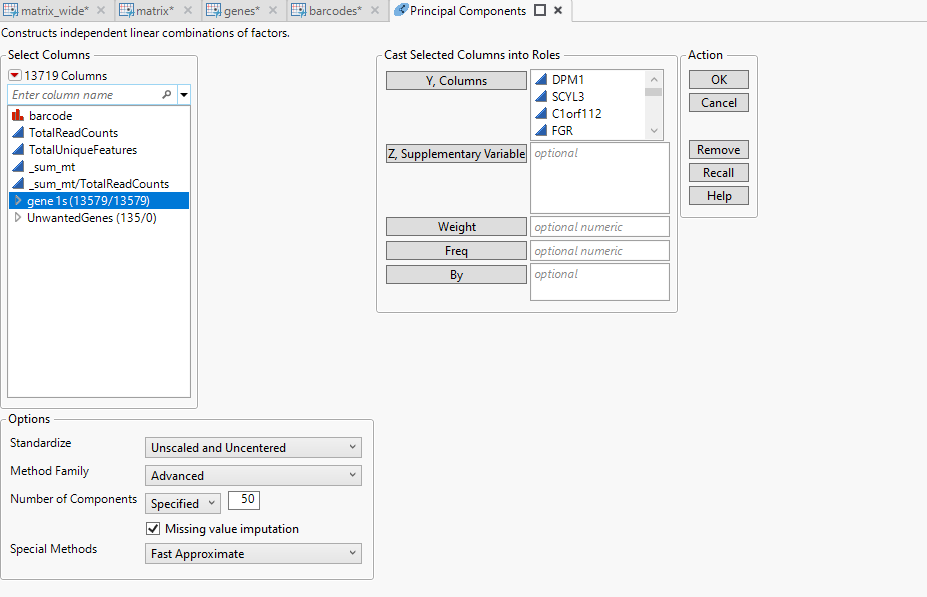
Thus we can group genes into two groups: genes to be used in the downstream analysis, and genes unwanted.



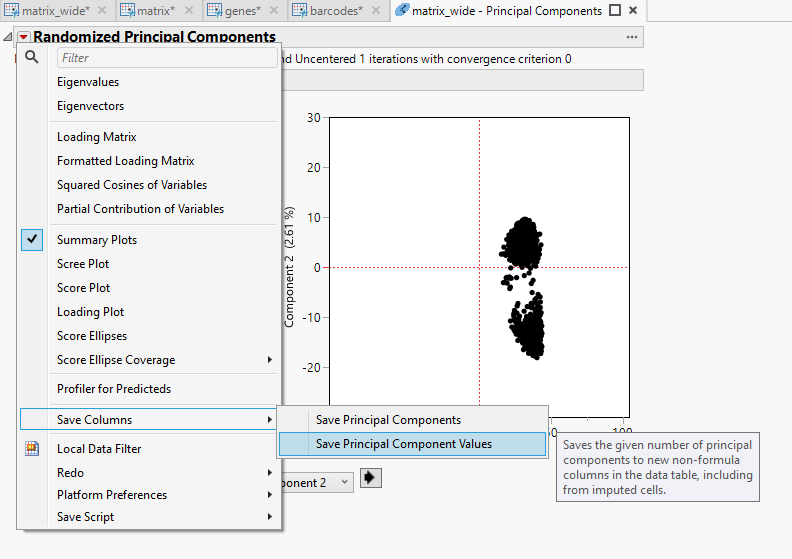
### Dimension Reduction

Run Random PCA: Analyze > Multivariate Methods > Principal Components

* Select 13579 normalized genes into “Y, Columns”.
* In the “Options” panel,
  + select “Unscaled and Uncentered” for Standardize.
  + Select “Advanced” for Method Family
  + Specify 50 for “Number of components”
  + Select the desired method. “Fast Approximate” will run the random SVD algorithm, and is much faster.



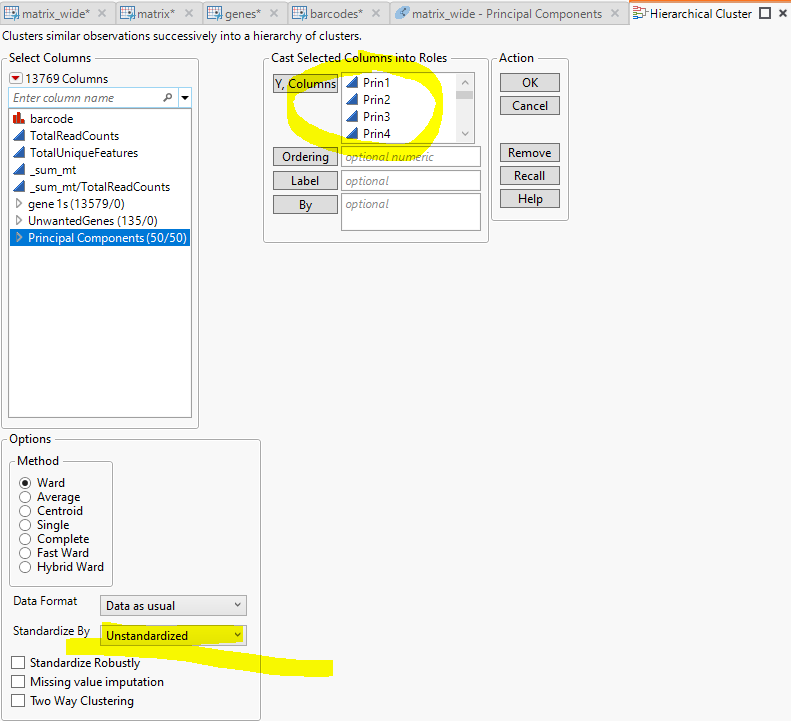
Next, save the PC values:



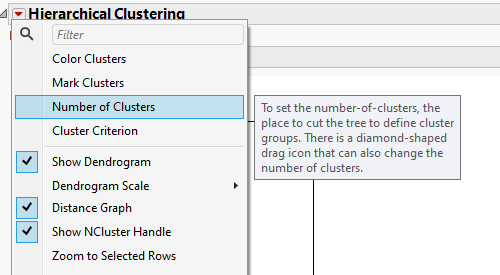
### Clustering

Method choice and clustering parameters setting need to be guided. Otherwise, the results could be very different.

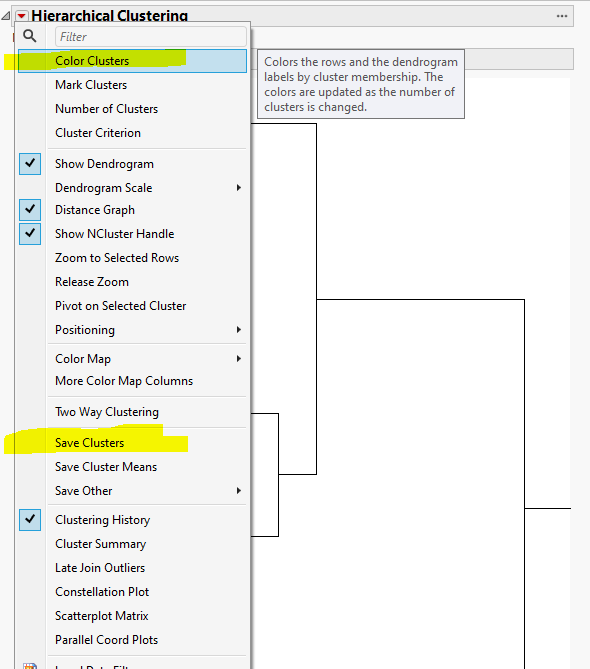
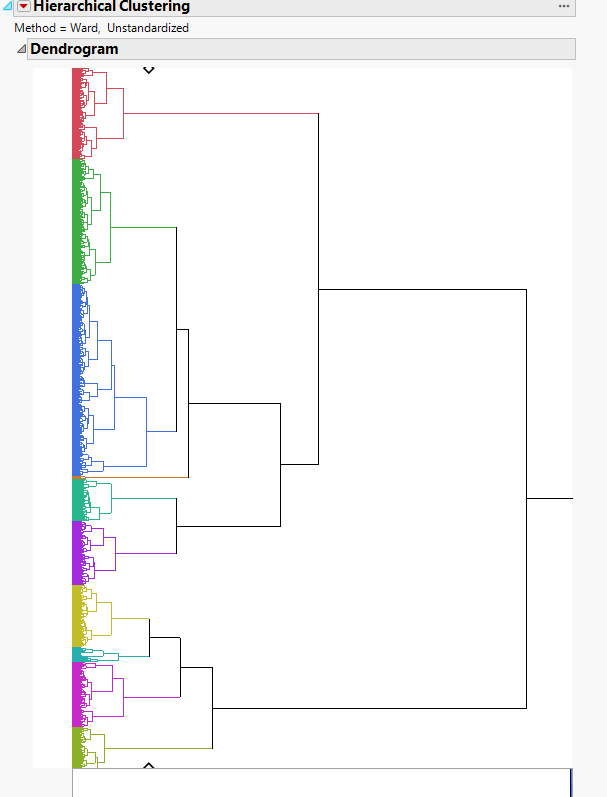
Analyze > Clustering > Hierarchical cluster, 10 clusters



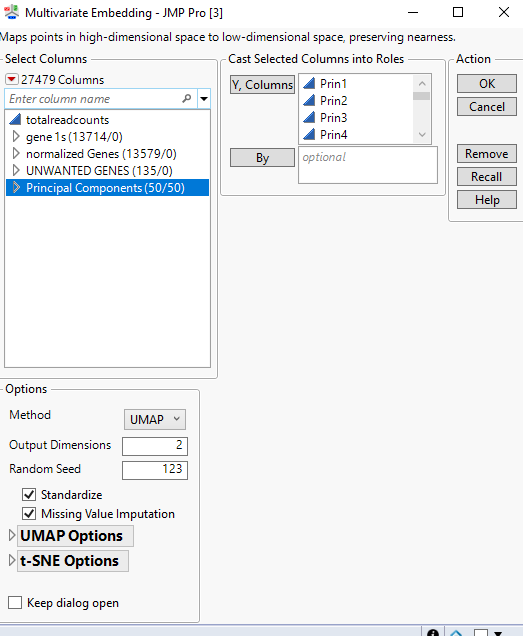
We can set a desired number of clusters: here we choose 10.

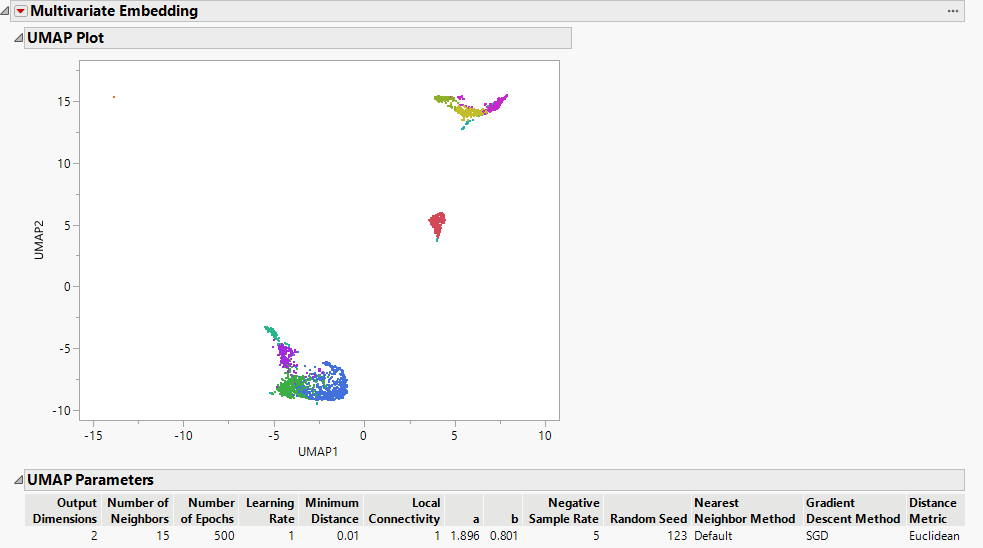


Then we can color clusters, and save the clusters to the data table:

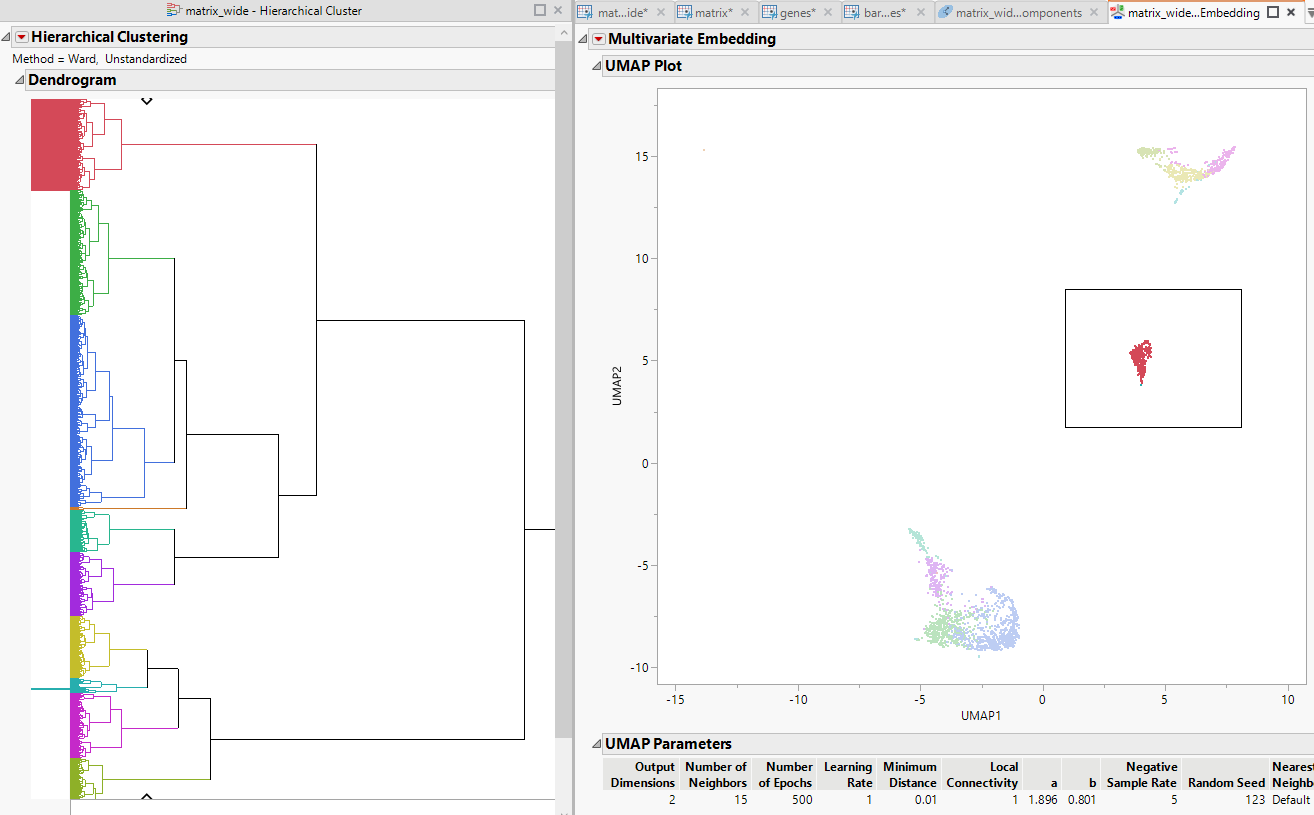
 

We can also visualize the results by Multivariate Embedding: Analyze > Multivariate Methods > Multivariate Embedding





We can put the UMAP plot next to the HC result: select a cluster on UMAP, we can see the corresponding part highlighted in the dendrogram.

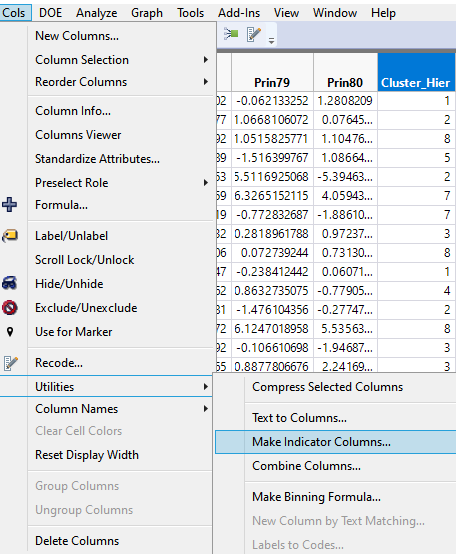
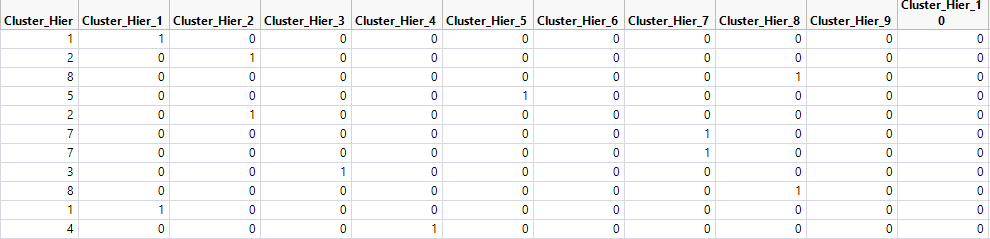


If the results seem reasonable, we can save clustering result into a variable.

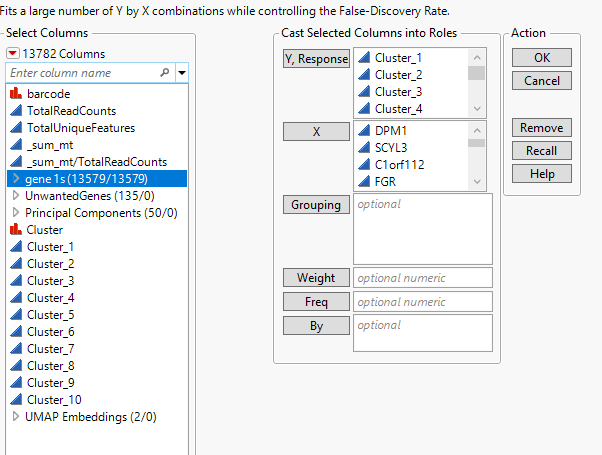
Users can also try other clustering methods, such as Kmeans.

### Find differentially expressed features (genes)

Bin the cluster variable and predict each one. Cols > Utilities > Make Indicator Columns. So each of the new column Cluster\_Hier\_1, … can serve as the outcome variable Y.

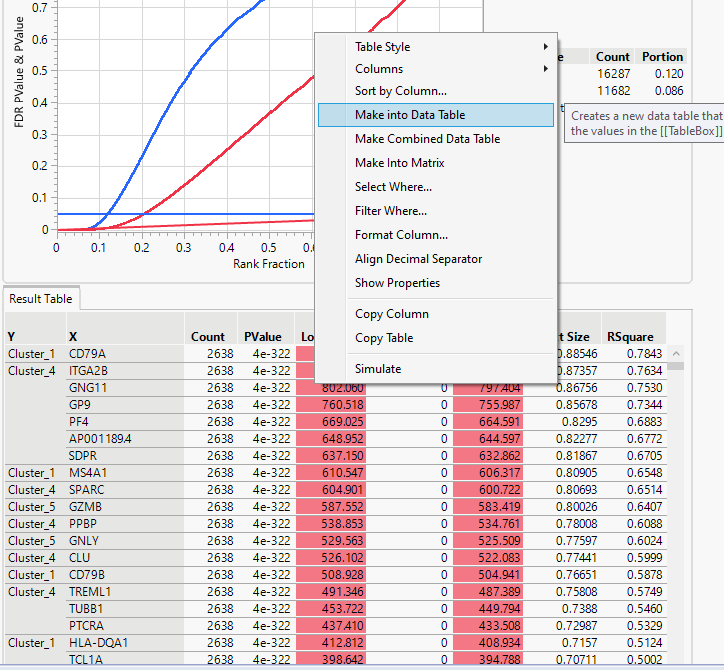
Put all binary cluster variables into Y, and select all normalized genes (13579 in this dataset) into X.



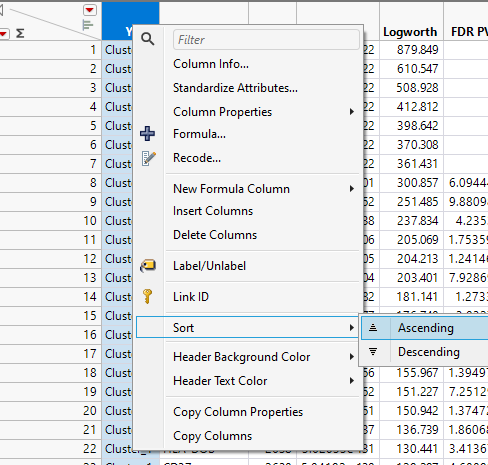
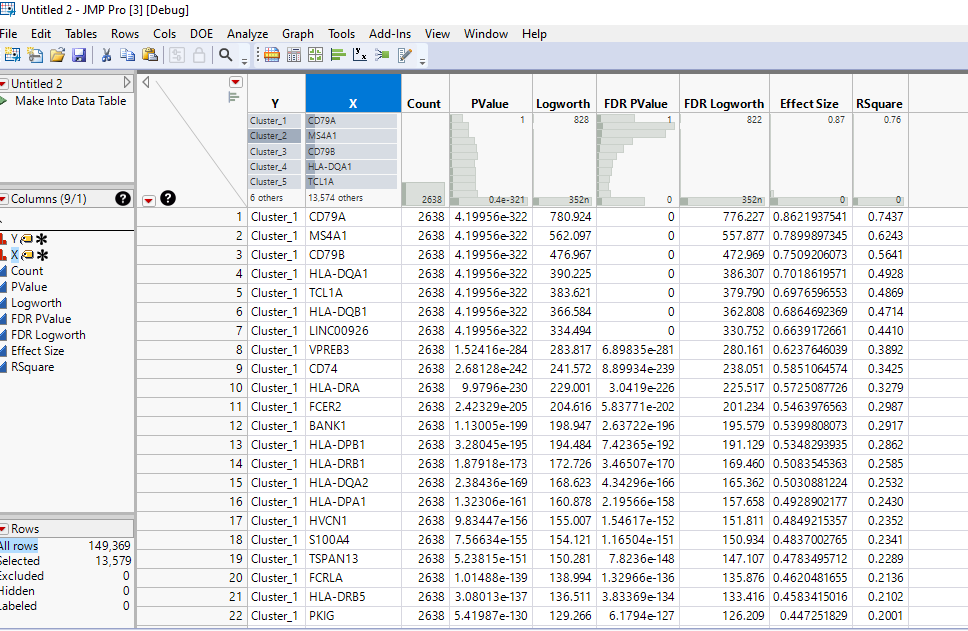
Run the Response screening platform using the default setting – click OK. We get the results in a second.



Make into data table,

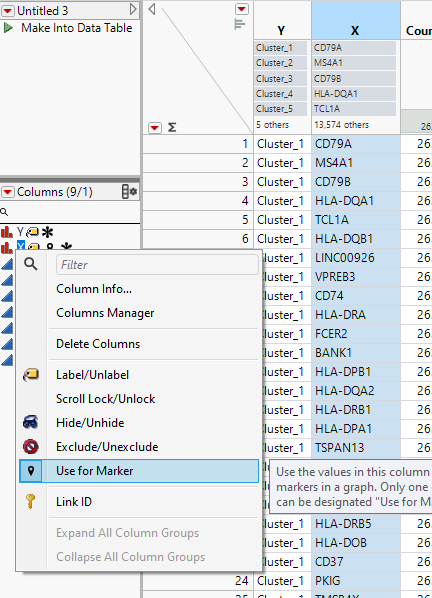


and sort by Y variable.

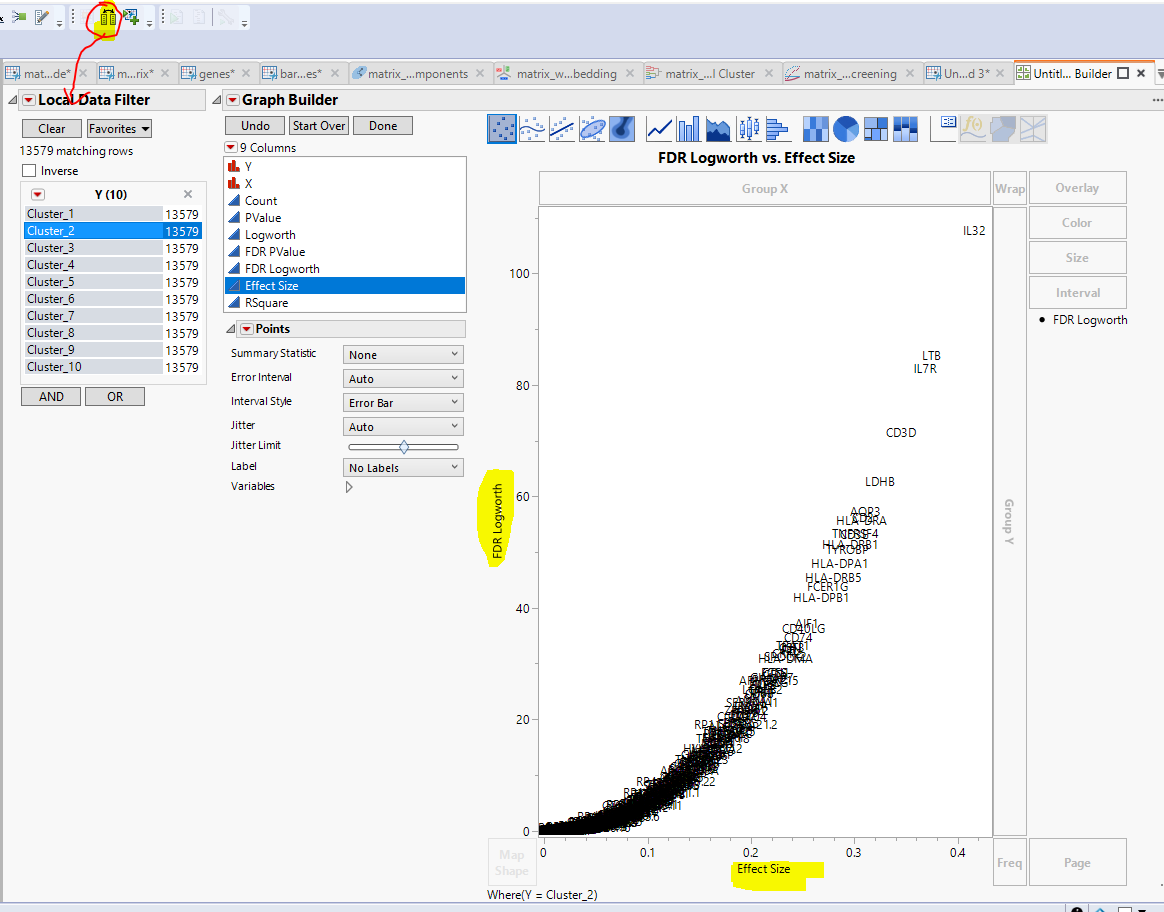
 

We can subset this table to derive the top significant genes in each cluster.

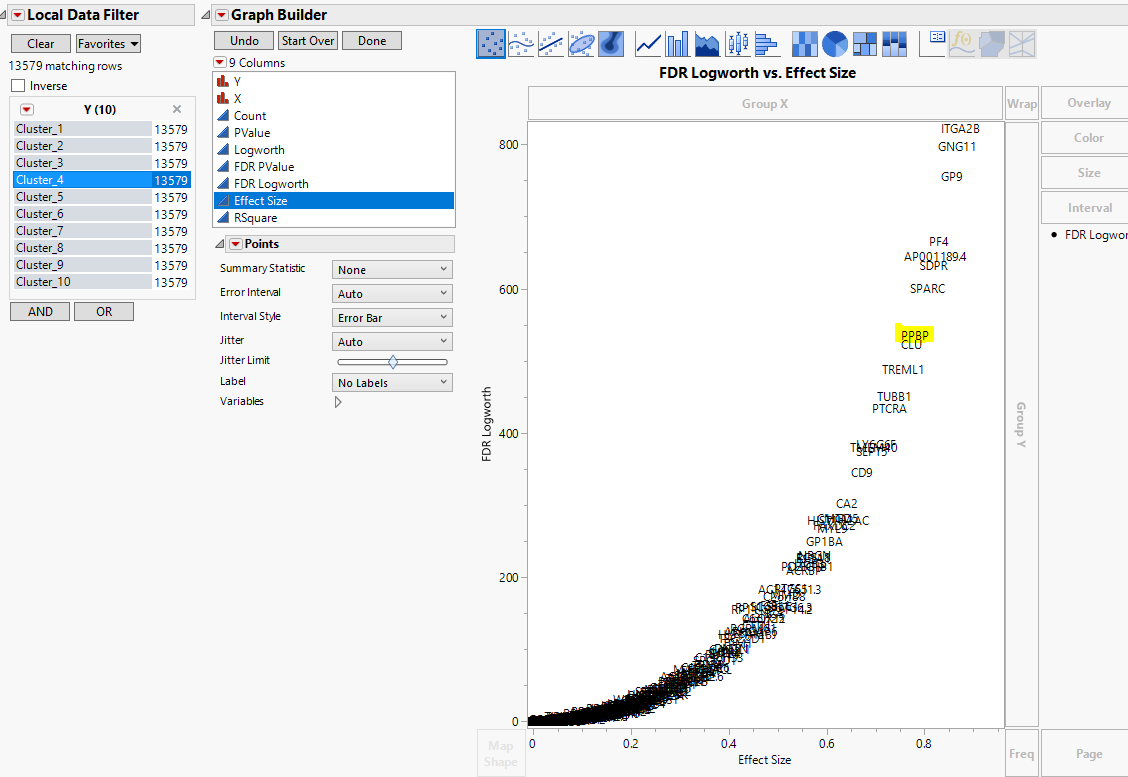
We can right-click on X variable and click on “Use for Marker”, so that when we visualize the results, the gene names would show up on the plot directly.



Go to Graph Builder, drag “FDR Logworth” to Y, and drag “Effect Size” to X. Next, click on the “Local data filter” button , and choose “Y” as the filtering variable:



By clicking on “Cluster\_1” through “Cluster\_10”, we can see the important genes for each cluster. For example, **cluster\_4** includes 11 points, and is far from all the other clusters on the UMAP plot:

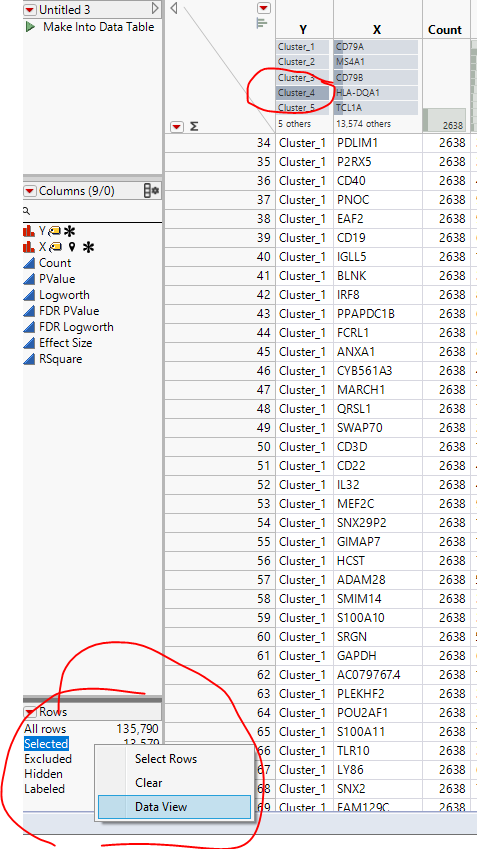
From here, we can identify the clusters by matching the marker genes. For example, “PPBP” is a well known marker gene for “Platelet”. Cluster\_4 is likely Platelet.

### Identify cell types

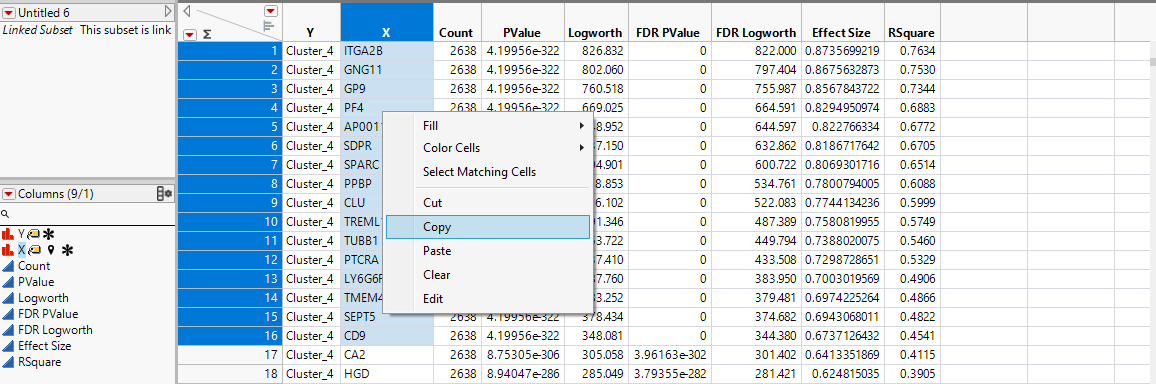
In this step, we will need to refer to biological knowledge to identify a cell type.

Note that some of the top predictors for a cluster could be just house keeping genes, such as ACTB, GAPDH, HPRT1, TBP, RPL13A, RPS18, B2M, PGK1, etc.. When identifying the marker genes, we sometimes need to remove the house keeping genes from the top predictors list.

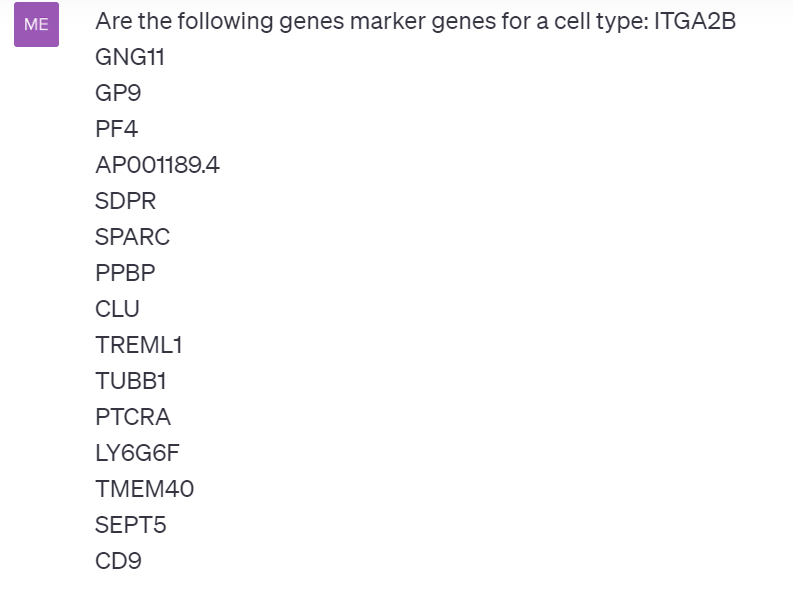
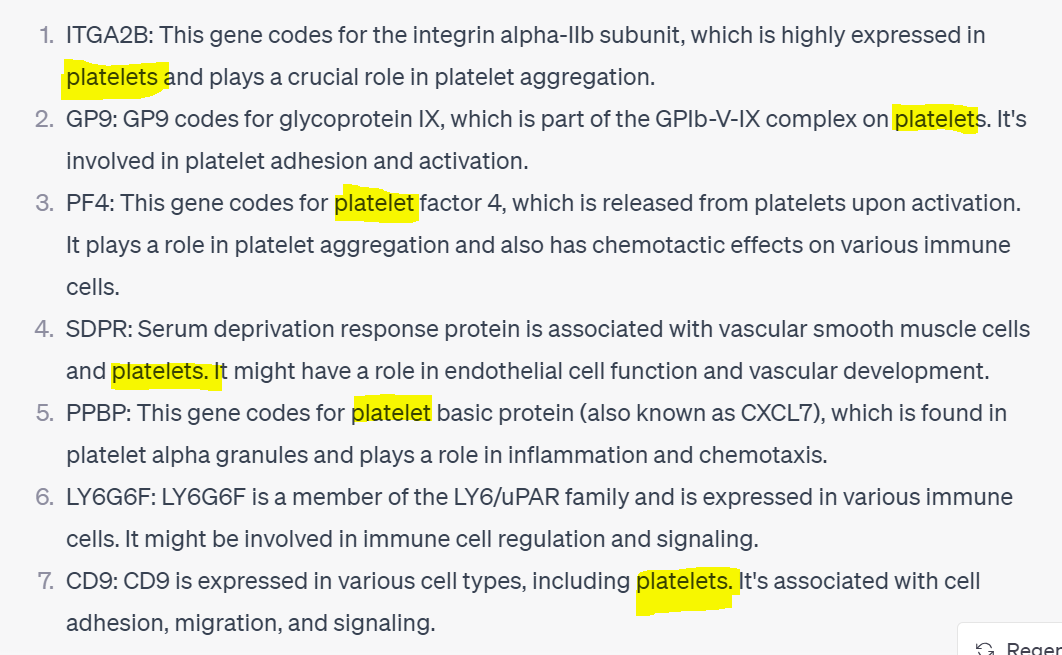
Continue with the Cluster\_4 example above, to take a closer look, we can select “Cluster\_4” from “Y” variable, and get 13579 rows selected. By right clicking on “Selected” at the left bottom panel, we can do a quick “Data View”.



From the data view table, we can easily copy the top genes and try to match them with gene databases.

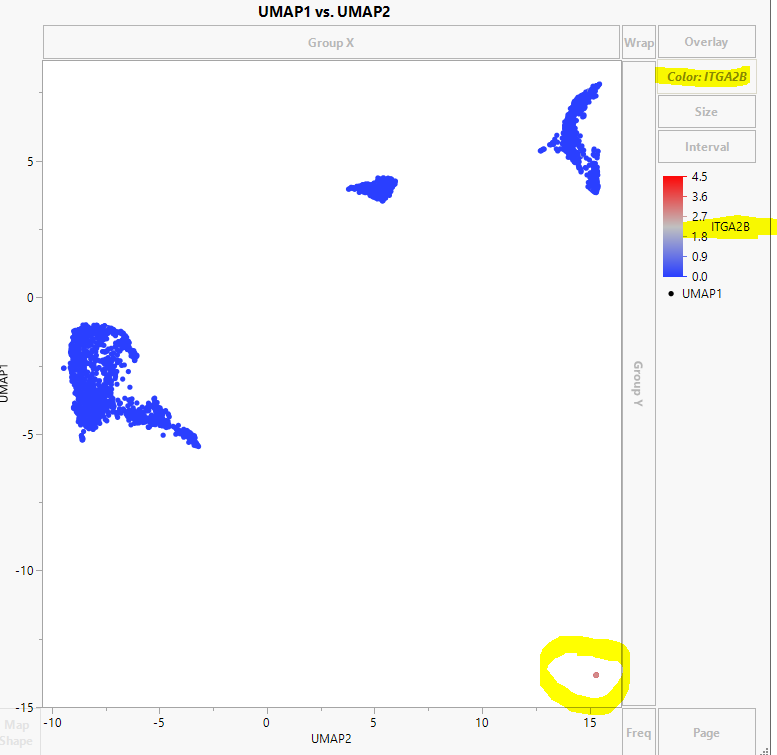


For example, we can ask Chat-GPT:

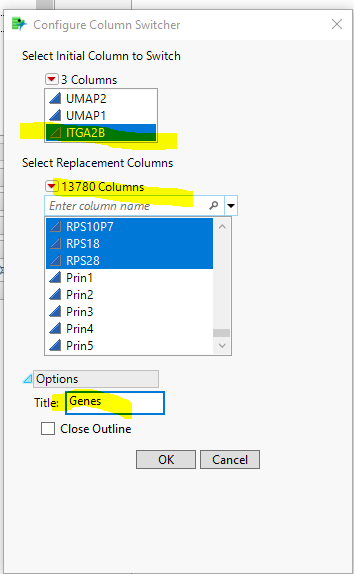
So we would mark Cluster\_4 as “Platelet”.

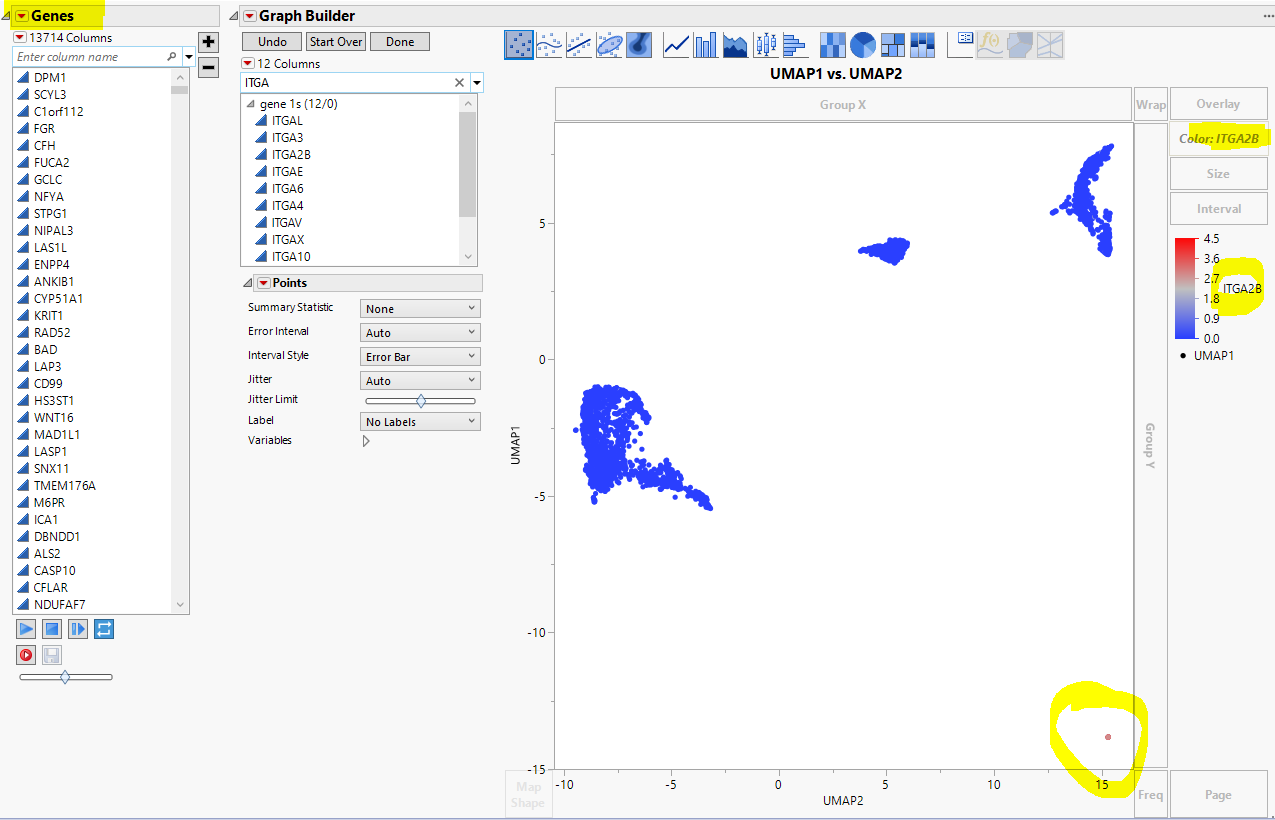
Meanwhile, we can save the UMAP or TSNE embedding components, and visualize expression levels for each cluster. According the to above results, we can use “ITGA2B” as a “color” variable.



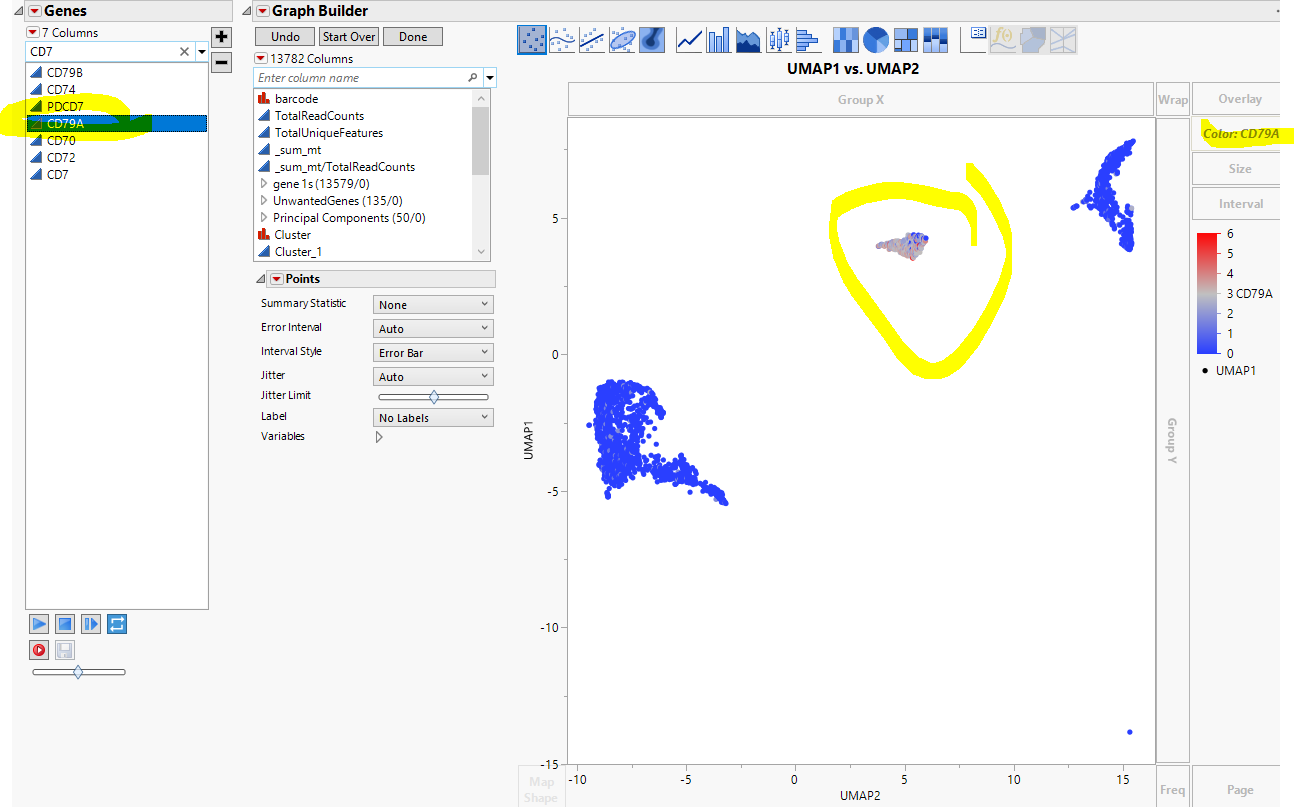
The ITGA2B level is high in Cluster\_4 as colored in red, and low in the other clusters, as colored in Blue.

Similarly, we can use column switcher  to view gene levels quickly:

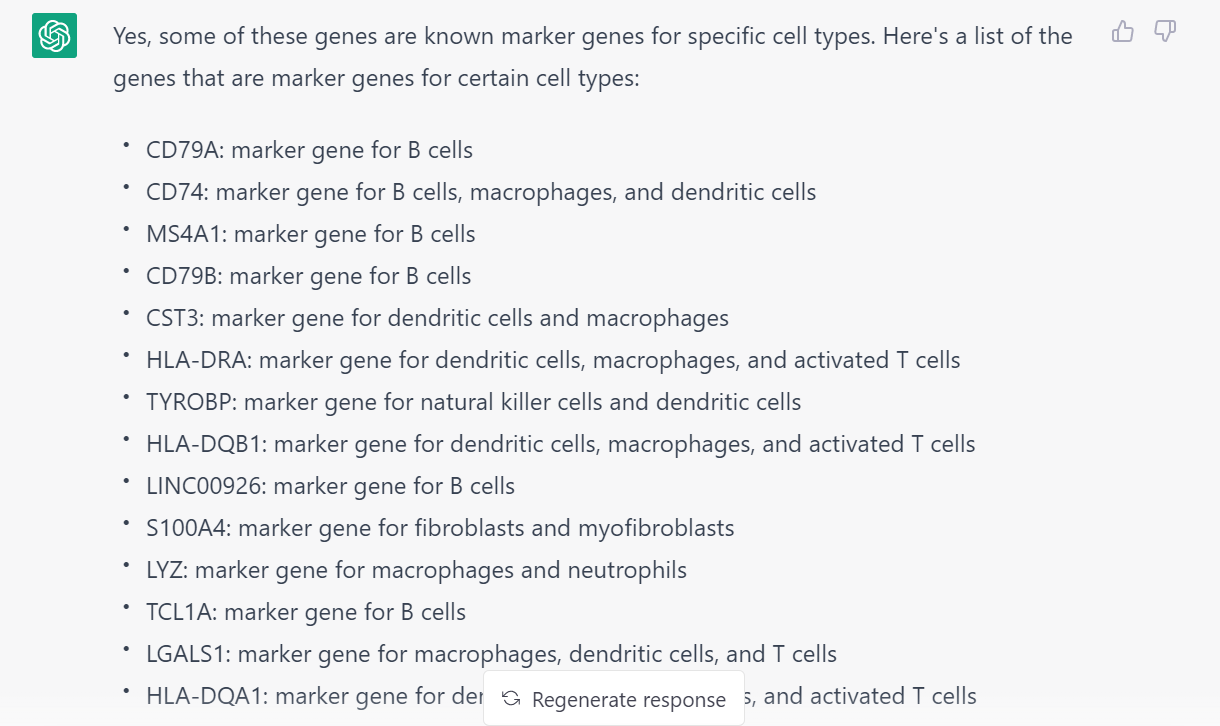




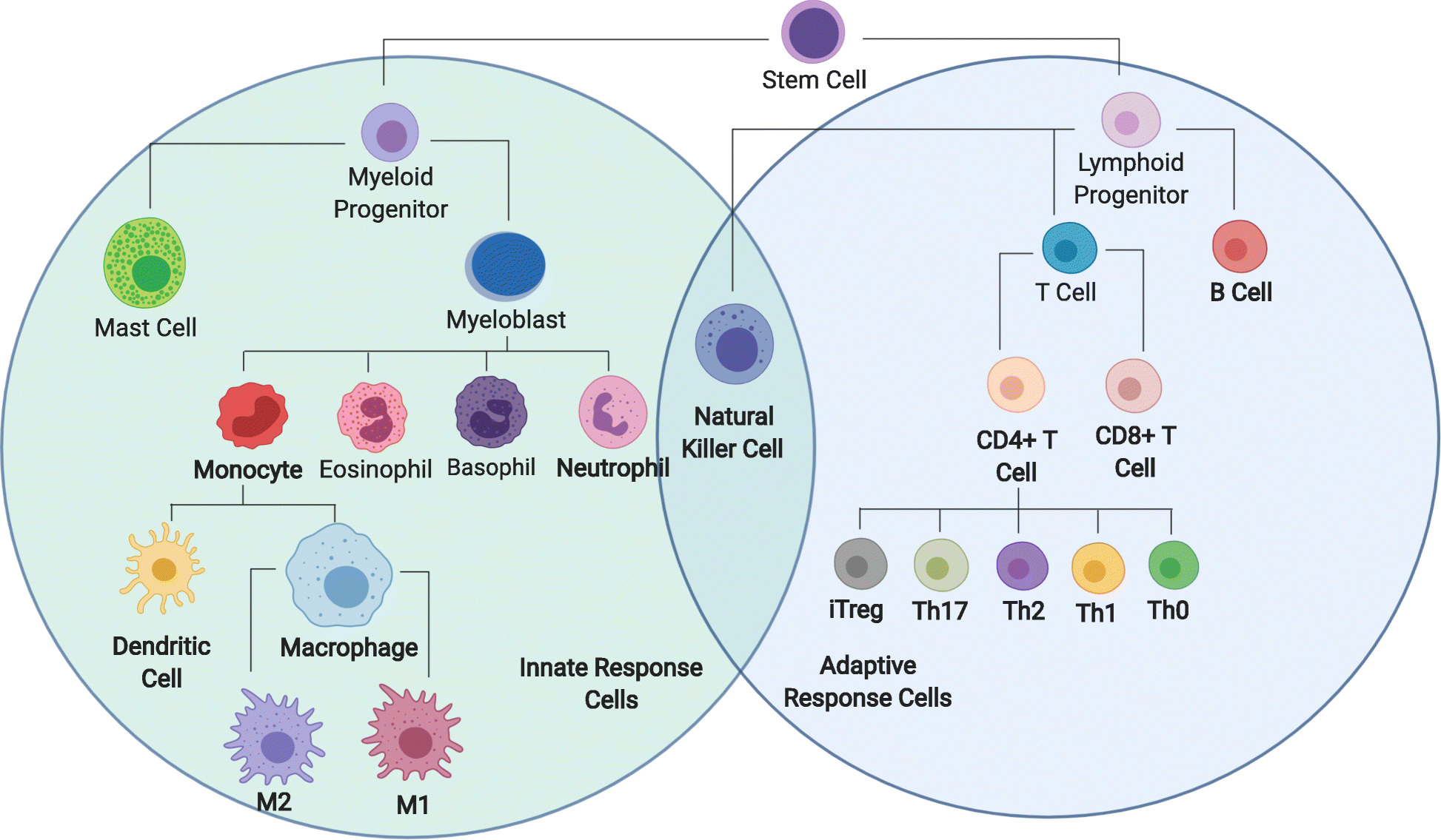
Then we are able to search and swap the coloring variables quickly: search for CD79A and click on it from “Genes” window, we see the UMAP plot shows that “Cluster\_1” has a higher expression level colored by Grey.



Like we did for Cluster\_4, for Cluster\_1, we take the top 20 genes of a cluster, and ask chatGPT:



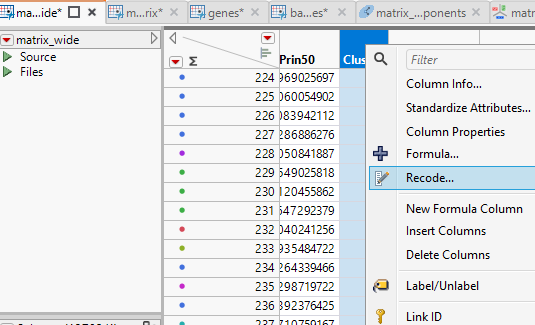
Then we define this cluster as B cells.

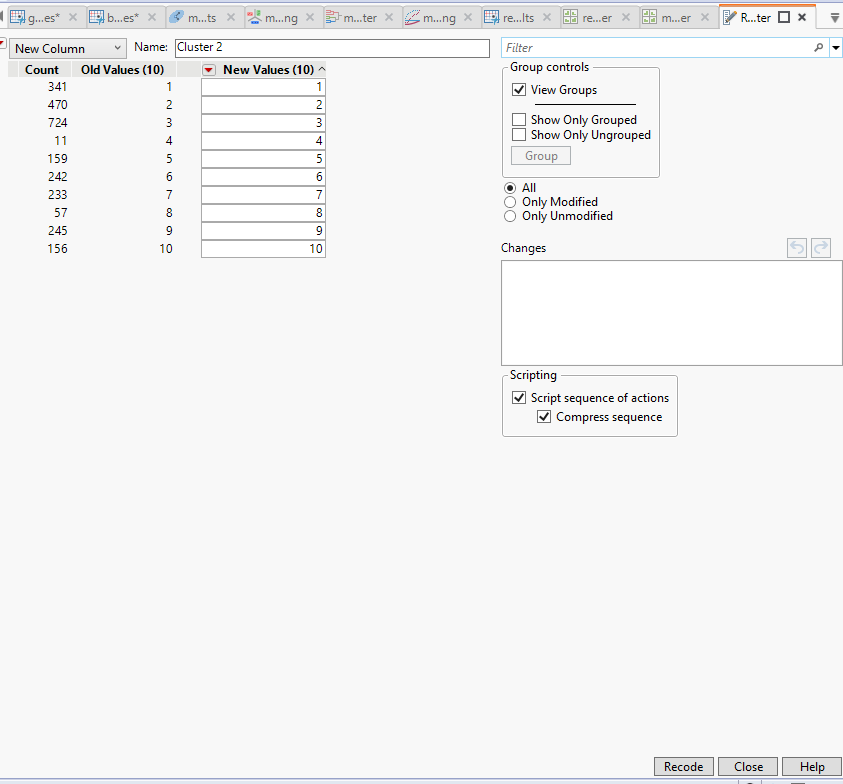


<https://media.springernature.com/full/springer-static/image/art%3A10.1186%2Fs12859-019-2994-z/MediaObjects/12859_2019_2994_Fig1_HTML.png>

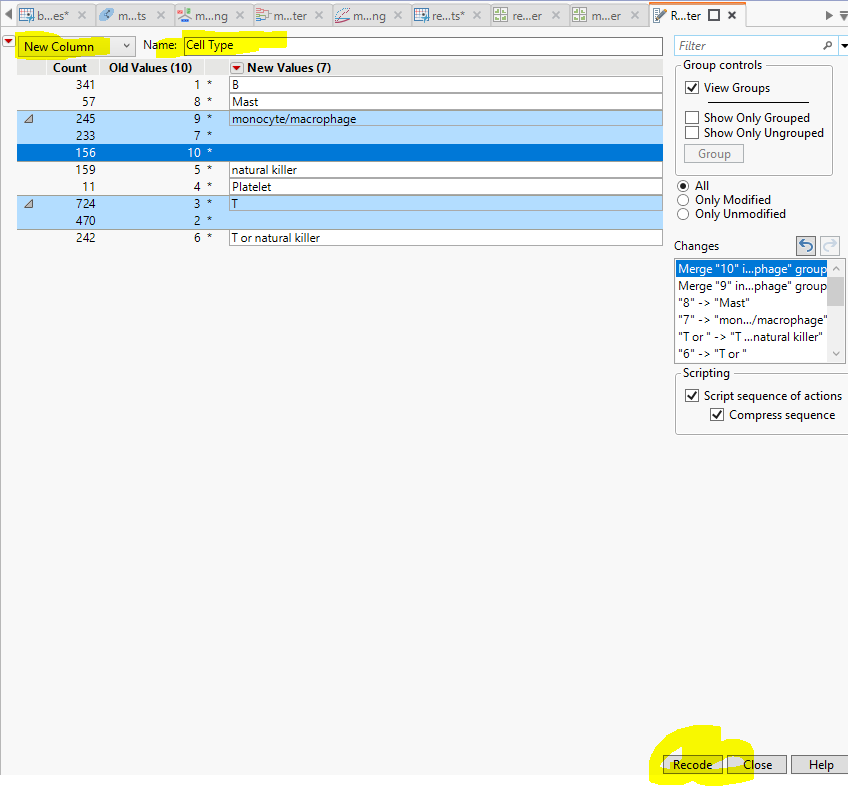
In this case, DC cells, are merged into the monocyte cells clusters.

Repeat this procedure for all the potential clusters. Then we can recode the cluster names. Go back to the Matrix\_wide file, right click on the “Cluster” variable name, and click “Recode”:

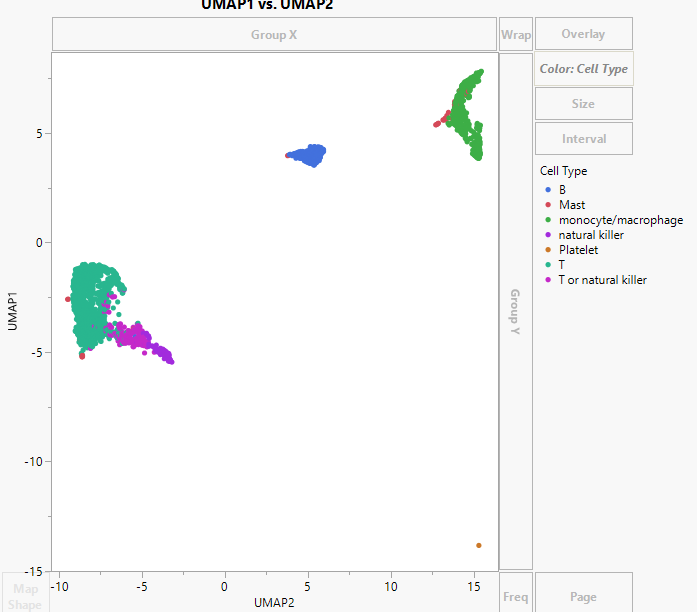




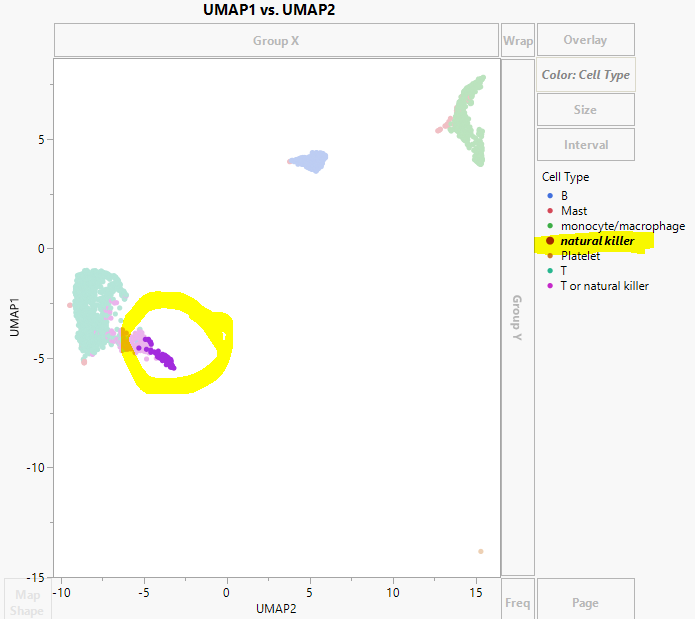
My experience is that Bard by Google is actually better than ChatGPT in terms of identifying the marker genes for different cell types.



We can visualize the annotated clusters on the UMAP plot:



Highlight one cell type by clicking on the “Cell Type” categories, for example, “natural killer”:



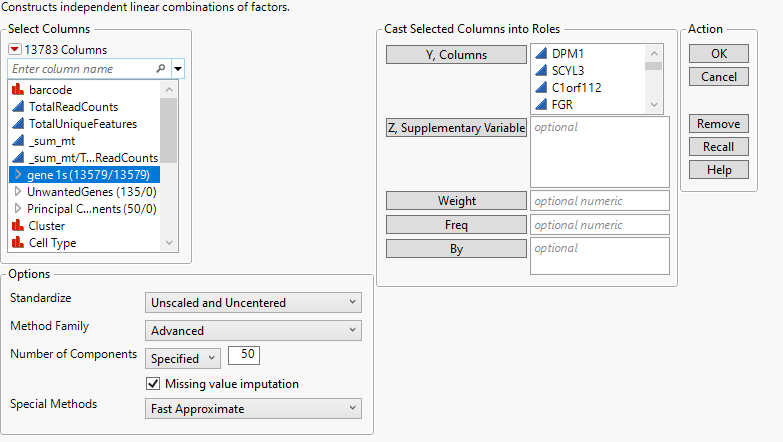
### Refine the clustering results

The above clustering results are our initial annotations. Users can always fine-tune the clusters / annotations. We can further classify the “monocyte/macrophage” into monocyte/macrophage/DC; “T” cells can be also classified into subtypes.

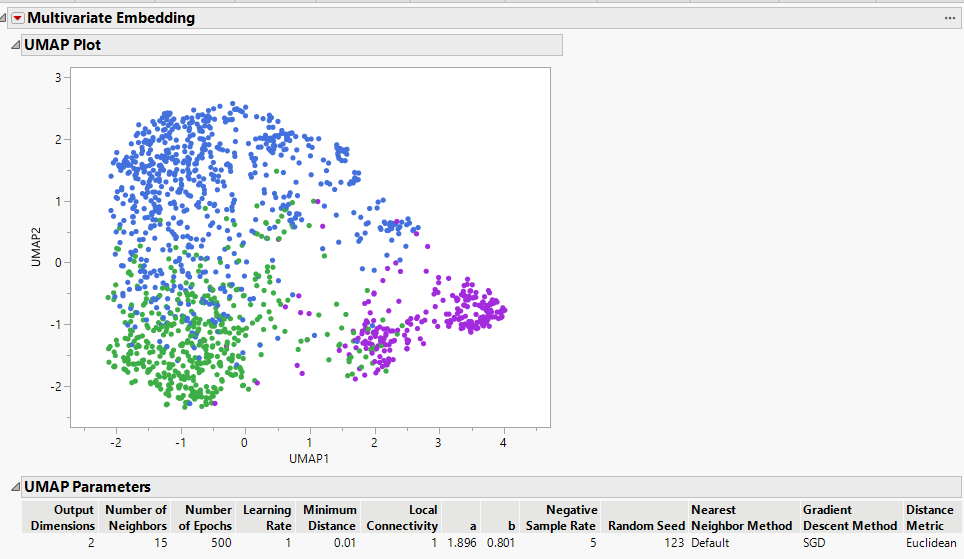
Let’s take the T-cell cluster as an example. Select “T” or “T or natural killer” rows into a subset.

Repeat the major steps:

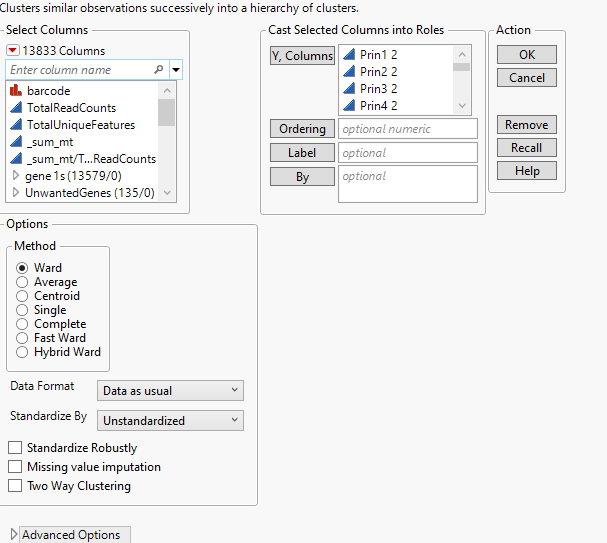
1. Run PCA, and save the 50PCs.



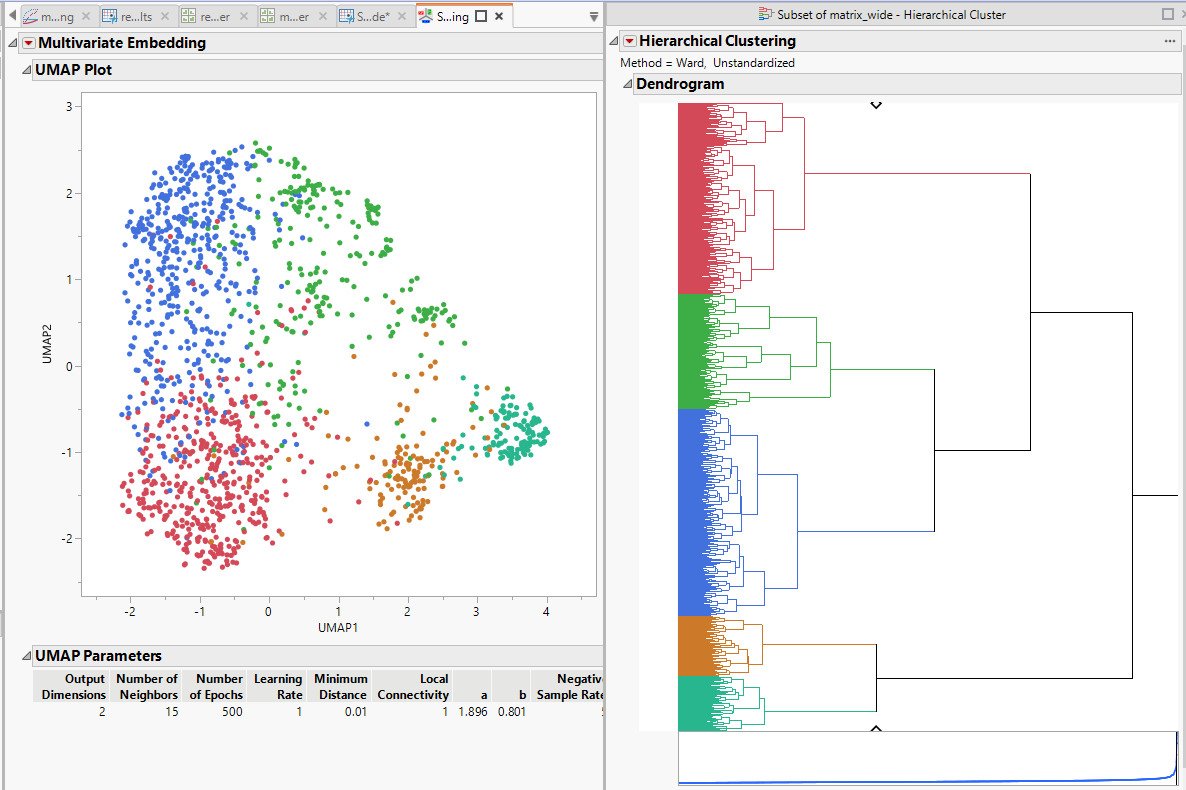
1. Run Multivariate Embedding – UMAP.



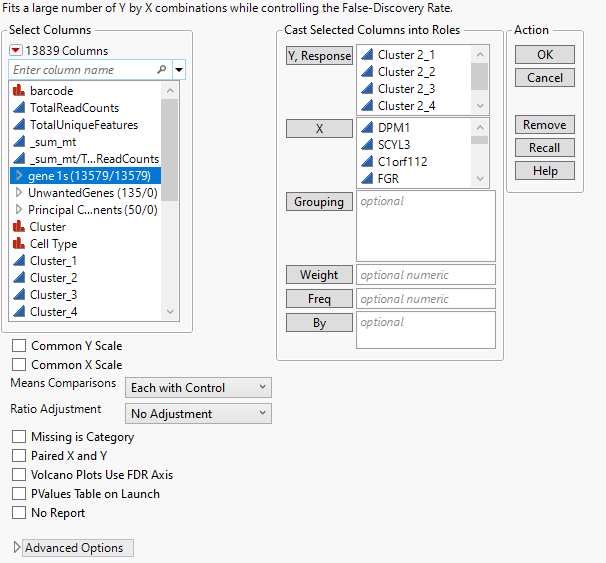
1. Run hierarchical clustering. Set number of clusters = 5, save cluster, and color cluster.

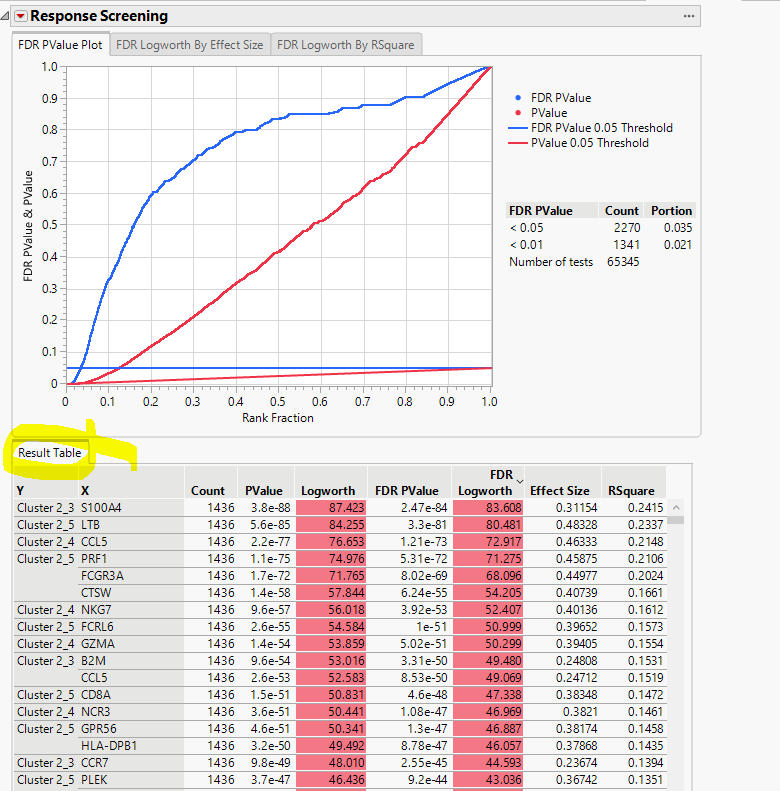


We can visualize the UMAP plot and the HC Dendrogram side by side;

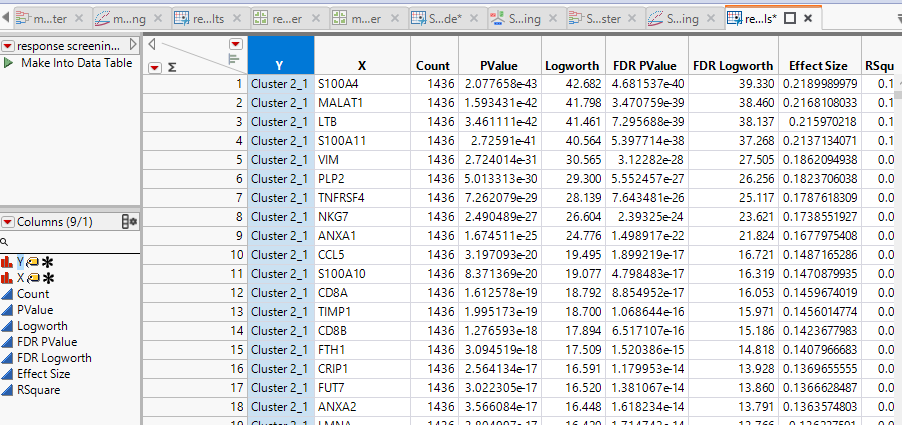


1. Make indicator columns for Cluster 2. Run response screening.

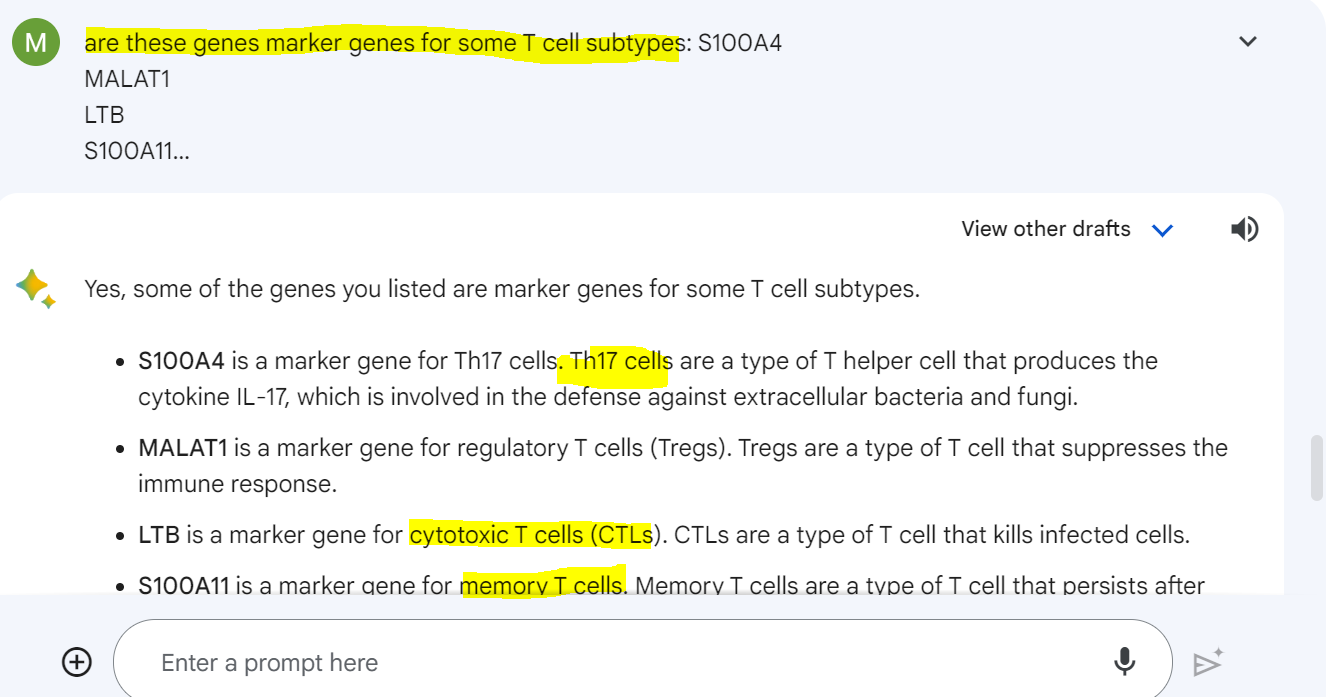




1. Save the results table into data table. Sort by Y.



1. Identify sub-clusters. I ask Bard:



1. Recode names and visualize.