Methods for quality control, normalization, and gene selection on single-cell RNA-seq data

Evan Lavelle

Abstract: The project serves as a follow-up to gene inference on the raw data of the mouse nervous system cells. This program will take the same data as input, but perform a sequence of QC steps followed by normalization. Two methods are used to perform this for each dataset, and the one judged superior by the user is to be taken forward for gene selection. Two functions from the M3Drop package are used for gene selection. For practicality, the combination of these gene lists is further reduced by quantitative measure of expression across all samples. Inference is then conducted on the normalized data of the selected genes. The identities of the top results are compared to the previous project and the pairs whose interactions rated most significant have their normalized expression values plotted against one another.

Introduction

The initial quality control step is removing all gene without any cell counts. This is sensible in that it will make the next steps more memory/time efficient, and these genes are not suitable for inference anyway. Secondly, a histogram is generated of column sums (gene totals of cells). The user must then determine where to apply a cutoff and change this value in the program script remove all cells under it from the matrix. Next, a different histogram is created of unique genes for each cell sample. Again, those with a number of genes lesser than the threshold set by the user can be manually removed. While both are reasonable steps to bring forward a clearer overall picture of relationships between prominent genes, there exists some potential here of eliminating relevant (if only marginally important) biological data. Finally, Ercc and mt- "spike-in" genes are located by search parameters with the respective strings. Cells the total gene expression values of which either contributes an excess of 10% are automatically removed. Because mitochondrial genes were strongly represented in the former gene inference, this step will likely set apart the results considerably.

Methods

The first preprocessing method of removing zero count genes is formatted as:
data <- data[*indeces of rows with sums above 1000*,]

For proper quality control, the histogram must be plotted: hist(sampleTotals, breaks = 100) and the argument "v" in the following abline function set accordingly. Similarly to the first step, columns then will be removed based on the decided value:
data <- data[,*indeces of columns with sums above whatever v was deemed satisfactory*]

The previous step is repeated with a different histogram: hist(originalTotals, breaks = 100). Then:
data <- data[,*indeces of columns with a number of unique genes >= to desired v value*]

The program separately searches indeces with "Ercc" and "mt-" in the row names, then inputs these vectors into a function along with the data to eliminate the column indeces which sums are comprised of a fraction of spike gene counts greater than 0.1:

```
for all columns:
        get the sum
        get the sum of spike genes
        if spike sum/sum > .1:
                add this column to deletion list
return the data without the columns on the deletion list
```

After pre-processing is complete, PCA and TSNE plots are created to compare to their counterparts made prior to the QC process. Then, the data separately normalized by counts per million and pooling. PCA and TSNE plots are generated once again for the user to decide which normalized data to proceed with. Because of this (and the visual judgment required for the earlier quality control measures involving the histograms), the program must be ran from an integrated developer environment such as RStudio. This way, the variables may be deliberately chosen based on the output of the intermediary graphs. As such, runtimes reported in the results for the different cells were computed in blocks between when user input was required and summed at the end of the program.

The program script proceeds to normalization, for which two methods (BrenneckGetVariableGenes and M3DropDifferentialExpression) are implemented, both available through the M3Drop library. Akin to the search for spike genes, the genes selected by these functions are sought for in the matrix of normalized data (whether by cpm or pooling), and a new matrix is made from their data. However, the number of genes may be too large to practically run inference on. Again, user input is necessary to decide which genes to take forward. Rather than basing this on row counts (which would be discriminatory against genes with low but consistent counts across samples), a function was implemented to include only genes with a percentage of sample counts of zero under a specified input value:

for all rows:
    get the sum of samples wherein the count was 0
    if the 0 sum/sum > input percentage expressed as a decimal:
        add this column to deletion list
return the data without the rows on the deletion list

With an acceptable number of rows in the new matrix, GENIE3 is ran and the top 50 interactions presented in the results appendix for the according cell type. A network is plotted and the inference repeated on shuffled data in order to establish an estimation of false positives. Scatterplots of the top 5 interactions are created in a loop, followed by one for the absolute bottom-ranked interaction for comparison. These steps are repeated with CLR. All output is saved in the results appendices files.

Results

Fibroblast genes: 32307 -> 17861 after zero count removal
Fibroblast cells: 1681 -> 1640 after read count QC
Fibroblast cells: 1640 -> 1595 after gene distribution QC
Fibroblast cells: 1595 -> 1595 after Ercc spike QC
Fibroblast cells: 1595 -> 1253 after mt- spike QC
Pool normalized data selected
Brennek-selected genes (fdr = 0.01, minBiolDisp = 0.5) : 2340
Dif Exp-selected genes (fdr = 0.01) : 0
Union of selected genes: 2340
After removing genes with zero counts in excess of 80%: 76
Total runtime: 8 min. 45 sec.

Microglia genes: 32307 -> 17927 after zero count removal
Microglia cells: 2577 -> 2368 after read count QC
Microglia cells: 2368 -> 2194 after gene distribution QC
Microglia cells: 2194 -> 2194 after Ercc spike QC
Microglia cells: 2194 -> 1946 after mt- spike QC
Pool normalized data selected
Brennek-selected genes (fdr = 0.01, minBiolDisp = 0.5): 2396
Dif Exp-selected genes fdr = 0.01) : 69
Union of selected genes: 1139
After removing genes with zero counts in excess of 85%: 86
Total runtime: 12 min. 7 sec.

Mural genes: 32307 -> 18659 after zero count removal
Mural cells: 5357 -> 4238 after read count QC
Mural cells: 4238 -> 4152 after gene distribution QC
Mural cells: 4152 -> 4152 after Ercc spike QC
Mural cells: 4152 -> 3664 after mt- spike QC
Pool normalized data selected
Brennek-selected genes: 2120
Dif Exp-selected genes: 0
Union of selected genes: 2120
After removing genes with zero counts in excess of 75% : 89
Total runtime: 1hr. 11 min. 57 sec.

Discussion

The most notable effect the quality control measures had on the outcome of the inference results was the omission of cells with high counts mt- spike genes, which dominated the top-ranks of the weighted adjacency matrix in the previous project. However, although tested to ensure functionality, the Ercc spike removal step did not remove any cell samples for any of the cell types. Ignoring the mitochondrial genes from the weighted adjacency matrix of the previous project, many of the same top-ranked genes are also placed highly in the weighted adjacency matrix produced by the controlled and normalized data, further corroborating some of the known interactions addressed in the scientific literature referenced in the prior project.

For all datasets, pool-normalized data was selected over the cpm-normalized data, due to what seemed clearer clusters in the PCA and TSNE graphs (see appendices). As described in the methods section, the number of genes to be carried to inference after normalization was cut by means of a written function. This approach was chosen over adjusting the .fdr parameter to the selection methods because the latter eventually resulted in mistaken inferences with overwhelming numbers of zero count values. Moreover, significantly reducing the returned genes with this parameter requires a decrease of several orders of magnitude.

Interestingly, objective measures of gene interaction as represented by the weight scores produced by the GENIE3 and CLR algorithms were largely consistent to those produced by the raw data in the prior project, in spite of many of the genes involved being different, particularly for the CLR results. Furthermore, the similarity in results between the two inference methods was tighter than seen before. The GENIE3 algorithm seems remarkably more capable of producing statistically significant results, judging by both the control scatterplot and the numbers determined by false positive value calculation. Additionally, subjective observation of the patterns elucidated by the plots show correlation between the reported genes. In some instances, the trends are obfuscated by clouds of datapoints and zero counts along the axes. In spite of this, clear linear arrangements can be seen, most prevalently in the mural results, where a large number of sample data is shown. Also worthy of note is the fact that for all datasets, the list of genes selected by the Brennek function entirely overlapped that produced by the DropExpression function, rendering the later redundant. The drastically increased runtime for the mural datasets is likely due to the necessity of manually increasing RStudio's random access memory allotment, which frequently brought the processes to a standstill. Unfortunately, this constraint was prohibitive for analyzing any larger datasets.