Contents

1. R programs derived from http://www.bioconductor.org/packages/release/bioc/vignettes/muscat/inst/doc/analysis.html
2. Python programs
3. Visualizing consistency
4. Visualizing inconsistency
5. Audit / debug .txt files
6. Plots
7. References

This document outlines preliminary results of analysis of the Kang data studied by Crowell, et al. as reported in [1] and [2]

Results descibed here are for the B cell cluster only.

Results produced by the method proposed in this posting are compared with output from Crowell’s program, muscat.

Programs, described briefly in sections 1 and 2, are posted in the folder muscat\_Kang\_data\_programs

Output files, described in sections 5 and 6, are posted in the folder muscat\_Kang\_data\_outputs

1. **R programs derived from**

**http://www.bioconductor.org/packages/release/bioc/vignettes/muscat/inst/doc/analysis.html**

data\_prep.r

* Vignette through section 2.5 Data overview
* Export SCE, counts, row and column data

pseudo bulk analysis and results export.r

* Vignette sections
  + 3.1 Aggregation of single-cell to pseudobulk data
  + 3.3 Sample-level analysis: Pseudobulk methods
* Analyze for each of 4 methods: edgeR, DESeq2, limma-trend, limma-voom
* Export results

1. **Python programs**

extract\_cluster\_data.py

* Select counts for a specific cluster – here, B cells
* Exclude lowly expressed genes

MLE\_cell\_clusters\_group\_id.py

* Compute pi-hat values for each gene – segmenting by group ID

randomized\_MLE\_cell\_clusters\_group\_id.py

* Prepare several sets of synthetic data by randomization
* For each randomized set, compute pi-hat values, by group ID, as for genuine data

compute\_pi\_hat\_range\_randomized\_cell\_clusters\_group\_id.py

* For each randomized set, compute the range of pi-hat values for each gene
* These will be used as baseline statistics to evaluate the genuine data for differential expression

analyze\_pi\_hat\_range\_randomized\_cell\_clusters\_group\_id.py

* For randomized data, compute mean and standard deviation of pi-hat for each gene
* For genuine data, compute the range of pi-hat values for each gene
* For each gene use the means and standard deviations of randomized data to compute z-scores for genuine data

compare\_z\_scores\_to\_muscat\_group\_id.py

* Compare differential expression results from the proposed method with muscat pseudobulk analysis
* Three sets of comparisons are performed
  + Scatter plots
  + Compare ranks, to visualize
    - consistency and
    - inconsistency between methods
* Note on comparing results from the five methods
  + DESeq2 and limma report differential expression with signed statistics (which presumably indicate the direction of the difference between ***ctrl*** and ***stim*** group IDs)
  + edgeR returns an F-statistic, which is always positive; it reports the degree of differential expression, not the sign
  + to consistently compare the five methods, absolute values are used for the statistics computed by DESeq2 and limma

1. **Visualizing consistency**

Consider two methods, B (base) and C (comparison) that estimate differential expression.

* For each gene, each method provides a statistic. Consider the pairs of statistics (b,c) ranging over all genes
* If B and C are consistent across all genes
  + the Spearman correlation of the results is near 1
  + a scatter plot of (b,c) resembles a curve, not a blob
* ***We do not require that ranks be highly correlated across all genes to conclude that B and C are consistent***
  + Suppose that 1000 genes are studied, and that B and C agree on the 100 with highest differential expression, but that for the remaining 900 genes the estimates are uncorrelated
  + We want to identify this consistency
* Intuition / example
  + For each method, calculate the ranks of differential expression estimates – assign ranks in descending order: the gene with greatest differential expression has rank 0
  + Select the 100 (of 1000) genes with B-rank < 100 – these are the genes with greatest differential expression, according to B
  + Consider the C-ranks for these 100 genes
    - If all C-ranks are < 100, then B and C agree exactly – for these 100 genes
    - Note that they may not agree exactly on the genes with B-rank < 50
    - Suppose instead that only 90 of the 100 C-ranks are < 100. Clearly there is high agreement between B and C
* In general
  + For each method, calculate the ranks of differential expression estimates – assign ranks in descending order: the gene with greatest differential expression has rank 0
  + For each rank **r**, 0 ≤ **r** < G, where G is the total number of genes evaluated for differential expression
    - Select **S**: the set of genes with B-rank ≤ **r**
    - Consider the C-ranks for genes in **S**
    - Calculate **n(r)** the number of genes in **S** with C-rank ≤ **r**
      * This is the number of genes in **S** for which B and C are consistent
      * Normalize by treating this as a fraction: compute f**(r)** = **n(r)** / **r**
* In this posting, the program compare\_z\_scores\_to\_muscat\_group\_id.py
  + Considers each of the five methods – z-scores, then edgeR, DESeq2, limma-trend, and limma-voom – as base, B
  + Then plots the curves **( r, f (r) )** for the remaining four methods

1. **Visualizing inconsistency**

The objective of the “visualizing consistency” analysis is to identify the agreement between methods for each number of genes.

Here the objective is to characterize how badly methods disagree. This preliminary suggestion may be too stringent.

* Using the same language as above
  + For each method, calculate the ranks of differential expression estimates – assign ranks in descending order: the gene with greatest differential expression has rank 0
  + For each rank **r**, 0 ≤ **r** < G, where G is the total number of genes evaluated for differential expression
    - Select **S**: the set of genes with B-rank ≤ **r**
    - Consider the C-ranks for genes in **S**
    - Report **M(r)**  – the maximum C-rank. This measures how much C differs from B
      * If B and C agree exactly, M(r) = r
      * Now consider a pathological example
        + Suppose that the gene ranked by B with the greatest differential expression is found by C to have the least differential expression. Its C-rank equals G-1
        + Then **M(r)** = G-1 for all **r**
* As above, the program compare\_z\_scores\_to\_muscat\_group\_id.py
  + Considers each of the five methods – z-scores, then edgeR, DESeq2, limma-trend, and limma-voom – as base, B
  + Then plots the curves **( r, M (r) )** for the other four methods

1. **Audit / debug .txt files**

Each python program includes print statements to document computations.

These are included, with the python source code at the top of each file.

extract\_cluster\_data B cells\_.txt

MLE\_cell\_clusters\_group\_id B cells\_.txt

randomized\_MLE\_cell\_clusters\_group\_id B cells\_41\_randomizations\_.txt

compute\_pi\_hat\_range\_randomized\_cell\_clusters\_group\_id B cells\_41\_randomizations\_.txt

analyze\_pi\_hat\_range\_randomized\_cell\_clusters\_group\_id B cells\_41\_randomizations\_.txt

compare\_z\_scores\_to\_muscat\_group\_id B cells\_.txt

1. **Plots**

The file

compare\_z\_scores\_to\_muscat\_group\_id B cells.pdf

contains 20 plots

* 10 scatter plots – comparing differential expression statistics for each pair of the five methods
* 5 plots to visualize consistency
* 5 plots to visualize inconsistency

These show high agreement between limma-voom, edgeR, and DESeq2

The “inconsistency” plot with the z-score method as base shows considerable disagreement with the pseudo-bulk methods (page 16).

* At least one gene among the 100 most highly expressed according to the z-score method is ranked between 500 and 700 by pseudobulk methods with limma-voom, edgeR, and DESeq2
* This may merit scrutiny

1. **References**

1. Crowell HL, Soneson C, Germain P, et al. On the discovery of subpopulation-specific state transitions from multi-sample multi-condition single-cell RNA sequencing data. bioRxiv; 2019. DOI: 10.1101/713412.

2. http://www.bioconductor.org/packages/release/bioc/vignettes/muscat/inst/doc/analysis.html