Introduction to M3Drop: Michaelis-Menten modelling of dropouts in scRNASeq

Tallulah Andrews

April 24, 2017

Introduction

Single-cell RNA sequencing is able to quantify the whole transcriptome from the small amount of RNA present in individual cells. However, a consequence of reverse-transcribing and amplifying small quantities of RNA is a large number of dropouts, genes with zero expression in particular cells. The frequency of dropout events is strongly non-linearly related to the measured expression levels of the respective genes. M3Drop posits that these dropouts are due to failures of reverse transcription, a simple enzyme reaction, thus should be modelled using the Michaelis-Menten equation as follows:

$$P_i = 1 - \frac{S_i}{S_i + K}$$

Where P_i is the proportion of cells where gene i dropouts out, S_i is the mean expression of gene i and K is the Michaelis constant.

Example Workflow

We'll be using a portion of the Deng et al. (2014) dataset in this example. You can download the R-package containing this data (M3DExampleData) from Bioconductor using biocLite().

```
> library(M3Drop)
> library(M3DExampleData)
```

QC and Normalization

The first step is to clean the data by remove cells with too few detected genes, genes that with very low expression, and to normalize the data. This can be done using any method but M3Drop includes a simple function that will clean the expression matrix and convert raw counts to counts per million (CPM). If alternative normalization methods are used the input expression matrix must not be log-transformed, nor contain negative values. If normalization adjusts zero values then M3Drop will use the minimal expression value in the entire matrix as the value for dropouts.

```
> Normalized_data <- M3DropCleanData(Mmus_example_list$data,
+                    labels = Mmus_example_list$labels,
+                    is.counts=TRUE, min_detected_genes=2000)
> dim(Normalized_data$data)

[1] 17278   133

> length(Normalized_data$labels)

[1] 133
```

Fitting the Michaelis-Menten

Next, we can compare the fits of different possible functions relating the proportion of dropouts to the average expression for each gene.

```
> fits <- M3DropDropoutModels(Normalized_data$data)
```

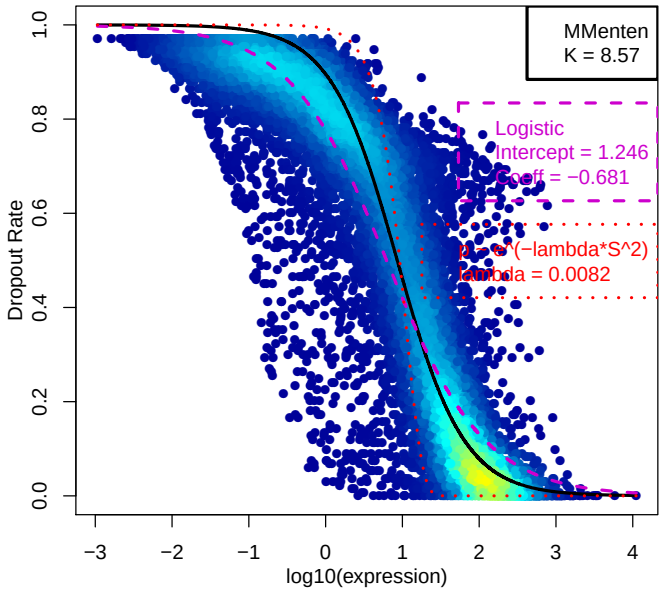

Figure 1: Fits of three different dropout models.

Visual inspection of the resulting plot (Figure 1) shows that the Michaelis-Menten equation is the best fit to the data. However, we can also examine some statistics to confirm this:

```
> # Sum absolute residuals
> data.frame(MM=fits$MMFit$SAr, Logistic=fits$LogiFit$SAr,
+            DoubleExpo=fits$ExpoFit$SAr)
```

```
    MM Logistic DoubleExpo
1 1632     1729       2731
```

```
> # Sum squared residuals
> data.frame(MM=fits$MMFit$SSr, Logistic=fits$LogiFit$SSr,
+            DoubleExpo=fits$ExpoFit$SSr)
```

```
   MM Logistic DoubleExpo
1 373      345        825
```

Here we see that the sum of squared residuals favours the flatter logistic curve due to the noise in the data, where as sum of absolute residuals shows the Michaelis-Menten is the best fit to the data.

Identifying Differentially Expressed (DE) Genes

Since the Michaelis-Menten equation is concave, averaging across a mixed population forces differentially expressed genes to be shifted to the right of the Michaelis-Menten curve. DE genes are identified by comparing the local K calculated for a specific gene to the global K fitted across all genes using a Z-test followed by multiple-testing correction. Here we find 1,248 DE genes at 1% FDR.

```
> DE_genes <- M3DropDifferentialExpression(Normalized_data$data,
+                          mt_method="fdr", mt_threshold=0.01)
```

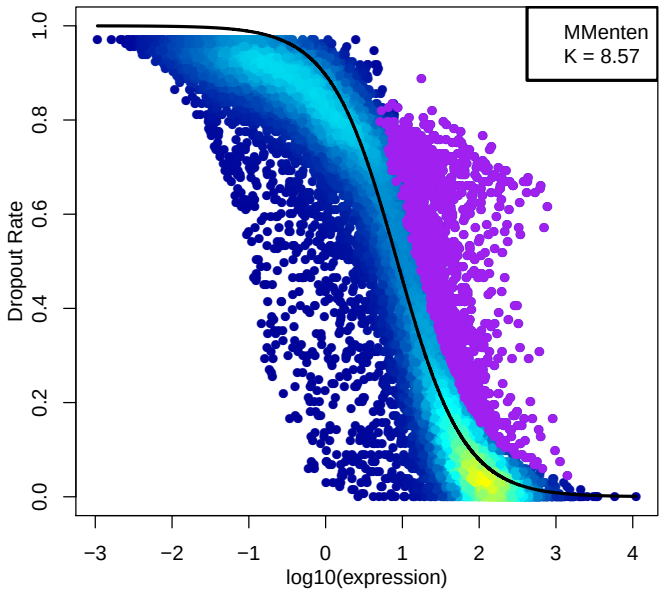

Figure 2: Differentially expressed genes at 1% FDR (purple).

Note that this function runs directly from the expression matrix, hence one could skip straight to identifying DE genes without comparing models and any external normalisation method can be applied to the raw counts prior to DE analysis.

Examining DE Genes and Identifying Subpopulations of Cells

To check that the identified genes are truly differentially expressed we can plot the normalised expression of the genes across cells.

```
> heat_out <- M3DropExpressionHeatmap(DE_genes$Gene, Normalized_data$data,
+                                     cell_labels = Normalized_data$labels)
```

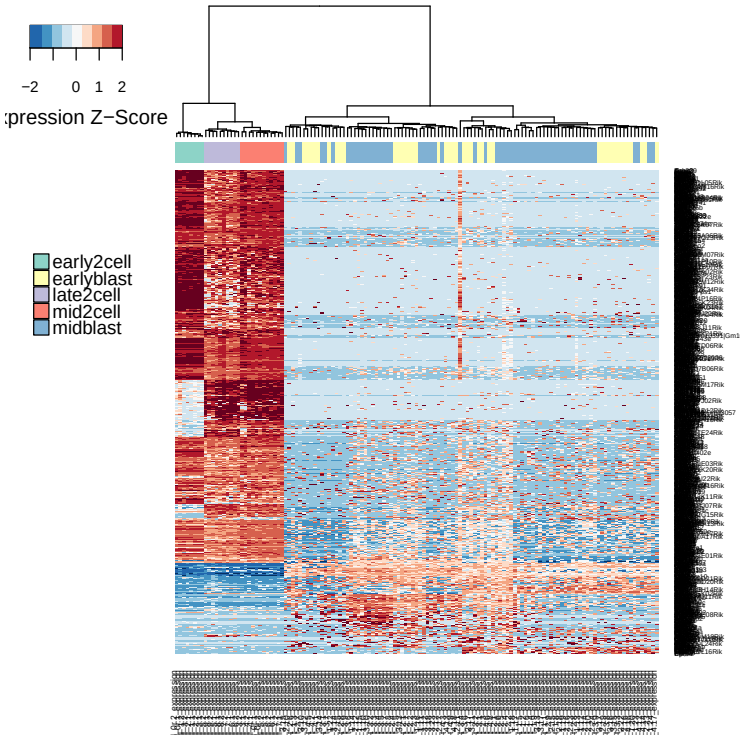

Figure 3: Heatmap of expression of DE genes.

The heatmap (Figure 3) shows that the identified DE genes are differentially expressed across timepoints. Furthermore, it shows that the blastocysts cluster into two different groups based on the expression of these genes. We can extract these groups and identify marker genes for them as follows:

```
> cell_populations <- M3DropGetHeatmapCellClusters(heat_out, k=4)
> library("ROCR")
> marker_genes <- M3DropGetMarkers(Normalized_data$data, cell_populations)
```

The first function cuts the dendrogram from the heatmap to produce k clusters of cells. These labels are stored in cell_populations. The second function tests all genes as marker genes for the provided clusters.

Marker genes are ranked by the area-under the ROC curve (AUC) for predicting the population with the highest expression of the gene from the other groups. Significance is calculated using a Wilcox-rank-sum test. Now we can examine the marker genes of the two clusters of blastocyst cells more closely.

```
> head(marker_genes[marker_genes$Group==4,],20)
```

```
                    AUC Group         pval
Col4a1        0.9650350     4 7.798215e-20
Tdgf1         0.9543124     4 3.895741e-19
Upp1          0.9491841     4 1.337152e-18
Sat1          0.9473193     4 1.861266e-18
Uhrf1         0.9473193     4 1.861266e-18
Sik1          0.9433566     4 3.677662e-18
Ckb           0.9400932     4 6.479222e-18
E130012A19Rik 0.9286713     4 3.415802e-17
Fabp5         0.9277389     4 5.220991e-17
Ahcy          0.9275058     4 5.410417e-17
Spp1          0.9272727     4 4.802140e-17
Etv5          0.9177156     4 2.653114e-16
Rnf130        0.9160839     4 2.976160e-16
Slc12a7       0.9135198     4 5.328023e-16
Pmm1          0.9107226     4 8.410373e-16
Npm1          0.9107226     4 8.410373e-16
Pecam1        0.9102564     4 8.971274e-16
Ephx2         0.9095571     4 1.354118e-17
Serpinh1      0.9074592     4 3.505876e-16
Sox2          0.9074592     4 1.043932e-15
```

```
> marker_genes[rownames(marker_genes)=="Cdx2",]
```

```
           AUC Group         pval
Cdx2 0.8166667     3 8.188436e-10
```

This shows that the inner cell mass (ICM) marker Sox2 is one of the top 20 markers for group 4 and that the trophectoderm (TE) marker Cdx2 is a marker for group 3, suggesting these two clusters coorespond to ICM and TE cells within each blastocyst.

Comparing to Other Methods

For comparison purposes, I have also included a function which implements the method to identify highly variable genes presented in Brennecke et al. (2013) with the added option to run it without providing spike-ins, in which case all genes are used to fit the function between CV2 and mean expression.

```
> HVG <- BrenneckeGetVariableGenes(Normalized_data$data)
```

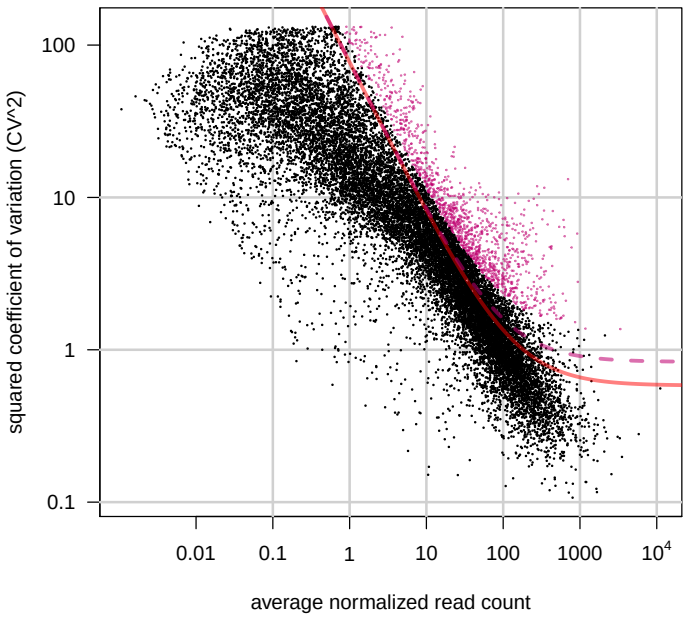

Figure 4: 1,236 Significantly highly variable genes (pink) using all genes as spike-ins (10% FDR).

This method is more sensitive to lowly expressed genes, and frequently picks up genes with fewer than 10 reads per million. In addition, the quadratic model it uses frequenly over estimates the expected variability of highly expressed genes thus only one gene with more than 1000 reads per million was detected as highly variable (Figure 4). This is in contrast with M3Drop (Figure 2) which recognizes the low information available for lowly expressed genes thus identifies few genes with expression < 10 reads per million as differentially expressed.

This difference can also be seen by comparing the heatmaps for the respective genes (Figure 3,5). The highly variable genes contains many more genes exhibiting just noisy expression, whereas nearly all genes detected by M3Drop are clearly differentially expressed across the different cell populations.

```
> heat_out <- M3DropExpressionHeatmap(HVG, Normalized_data$data,
+                 cell_labels = Normalized_data$labels)
```

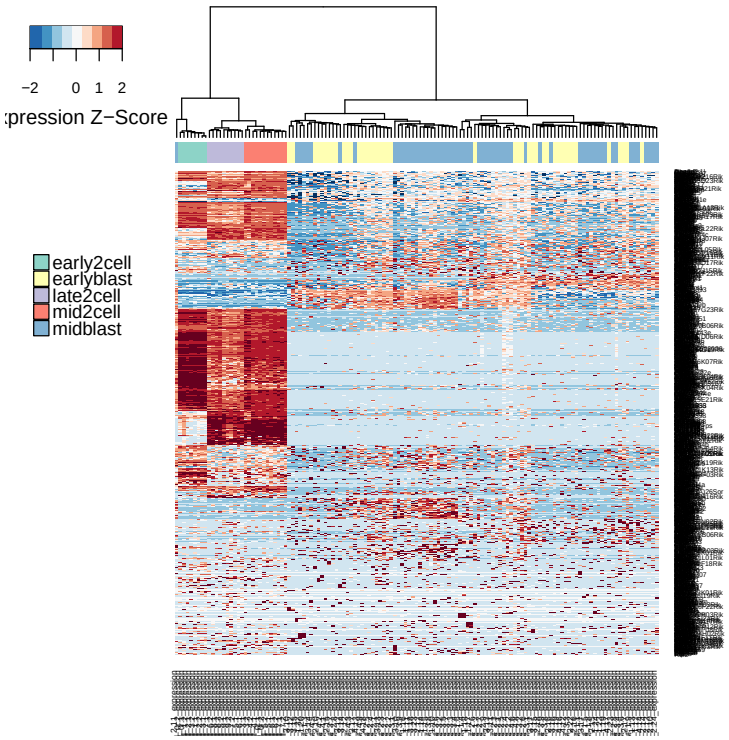

Figure 5: Heatmap of expression of highly variable genes across cells.

References

Tallulah Andrews, Martin Hemberg. Modelling dropouts allows for unbiased identification of marker genes in scRNASeq experiments. *bioRxiv*, 2016. doi:10.1101/065094

Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah A Teichmann, John C Marioni and Marcus G Heisler. Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, 10:1093-1095, 2013. doi:10.1038/nmeth.2645, PMID: 24056876