These programs estimate differential expression (DE) for single cell RNA-Seq data represented by UMI counts.

The approach closely follows the work of Townes et al. [1] by using

* multinomial models and
* binomial deviance to filter genes for clustering

Consider a cell clustering with K clusters

* For each cluster *k*, the multinomial model gives maximum likelihood estimates of the relative abundance of each gene
* Let π̂kj denote the estimate for gene *j* in cluster *k*

Just as each MLE estimate π̂j  for the null model equals the sum of cell counts for gene *j* divided by the total of all cell counts, π̂kj equals the sum of cell counts for gene *j* in cluster *k* divided by the total of all counts for cells in cluster *k*.

For the examples discussed in this posting, the range of values π̂kj for each gene

rj = maxk ( π̂kj ) - mink ( π̂kj )

measures differential expression and is correlated with binomial deviance.

Results are given here for simulated data and for the Zhengmixeq data sets [2] studied in [1].

The Zhengmixeq data sets were analyzed as proposed in [1]

* dimension reduction with GLM-PCA
* clustering with mclust [3]

This summary has 5 sections:

1. Outline of approach
2. Results for Zhengmixeq data sets
3. Results for a simulated data set
4. Example programs for the Zhengmix8eq data set
5. Example programs for simulated data

**1. Outline of approach**

Filter genes using binomial deviance

Genes were ranked by binomial deviance following Townes et al. [1].

For the Zhengmixeq data sets, the randomization scheme described in the Appendix, below, was used to select filtering thresholds:

* 164 genes – Zhengmix4eq
* 195 genes – Zhengmix8eq

Cluster cells

Following [1]

* GLM-PCA was used for dimension reduction. For the Zhengmixeq data sets, Townes reported excellent results using a Poisson model with 10 factors [4]. These parameters were used here.
* Cells were clustered with mclust. Clusterings with 2 to 20 clusters were generated

Fit clustering-specific multinomial models

For each clustering, calculate

* MLEs π̂kj
* negative log likelihood

These calculations were also performed for the null and saturated models.

**2. Results for Zhengmixeq data sets**

* For the clusters calculated with mclust, negative log likelihood and the likelihood ratio statistics (with respect to the null model) are listed in Tables 1 and 2.
* For the Zhengmix4eq data, differential expression was studied for 4 clusters; for Zhengmix8eq, 8 clusters. Please refer to Figures 1 and 2.
* For each gene, the range rj = maxk ( π̂kj ) - mink ( π̂kj ) was plotted against binomial deviance
* Genes selected for filtering are highlighted in red

**3. Results for a simulated data set**

A simulated data set with 2 clusters was analyzed.

Random counts were generated from a Poisson distribution.

* 2 clusters of 500 cells each
* 10,000 genes; 200 differentially expressed in each cluster
* Count distribution is Poisson ( N \* 1e-4 \* DE\_parameter)

where

* Log10 ( N ) is uniformly spaced between 4.5 and 6.0 for the 500 cells in each cluster
* Genes 0-199 are differentially expressed
  + In cluster 0 their DE\_parameter = 0.005
  + In cluster 1, it is 0.0002
* Genes 200-399 are also differentially expressed – the other way around
  + In cluster 0 their DE\_parameter = 0.0002
  + In cluster 1, it is 0.005
* In all other cases, the DE\_parameter = 0.0002

Results are illustrated in Figure 3.

**4. Example programs for the Zhengmix8eq data set**

These 6 programs in the folder example\_programs\_Zhengmix8eq perform a complete analysis:

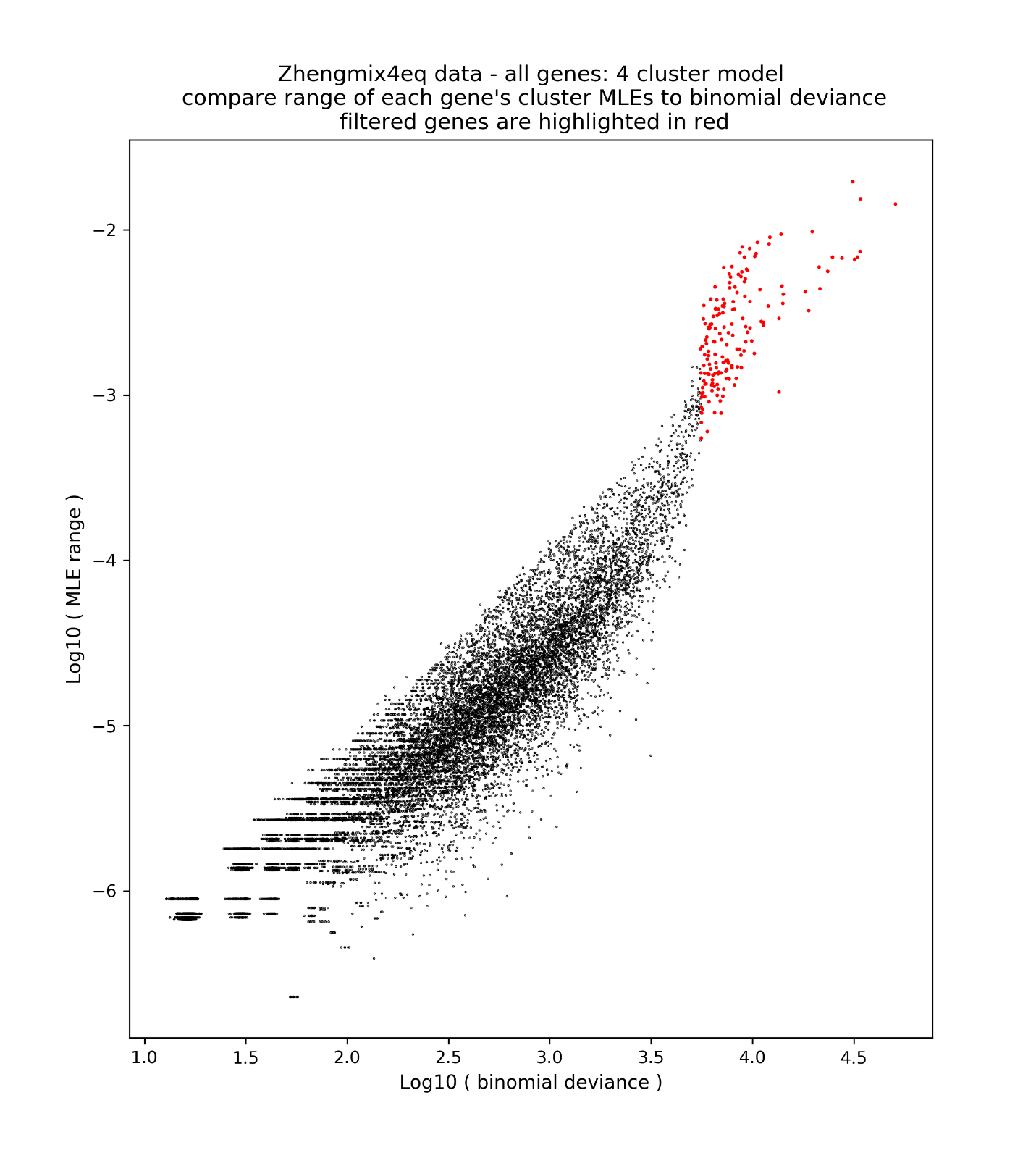
1. extract\_data\_from\_package\_DuoClustering2018.r
2. compute\_binomial\_deviance.py
3. filter\_UMI\_counts.py
4. glmpca\_mclust.r
5. MLE\_cell\_cluster\_models.py
6. compare\_gene\_range\_of\_MLE\_to\_binomial\_deviance.py

**5. Example programs for simulated data**

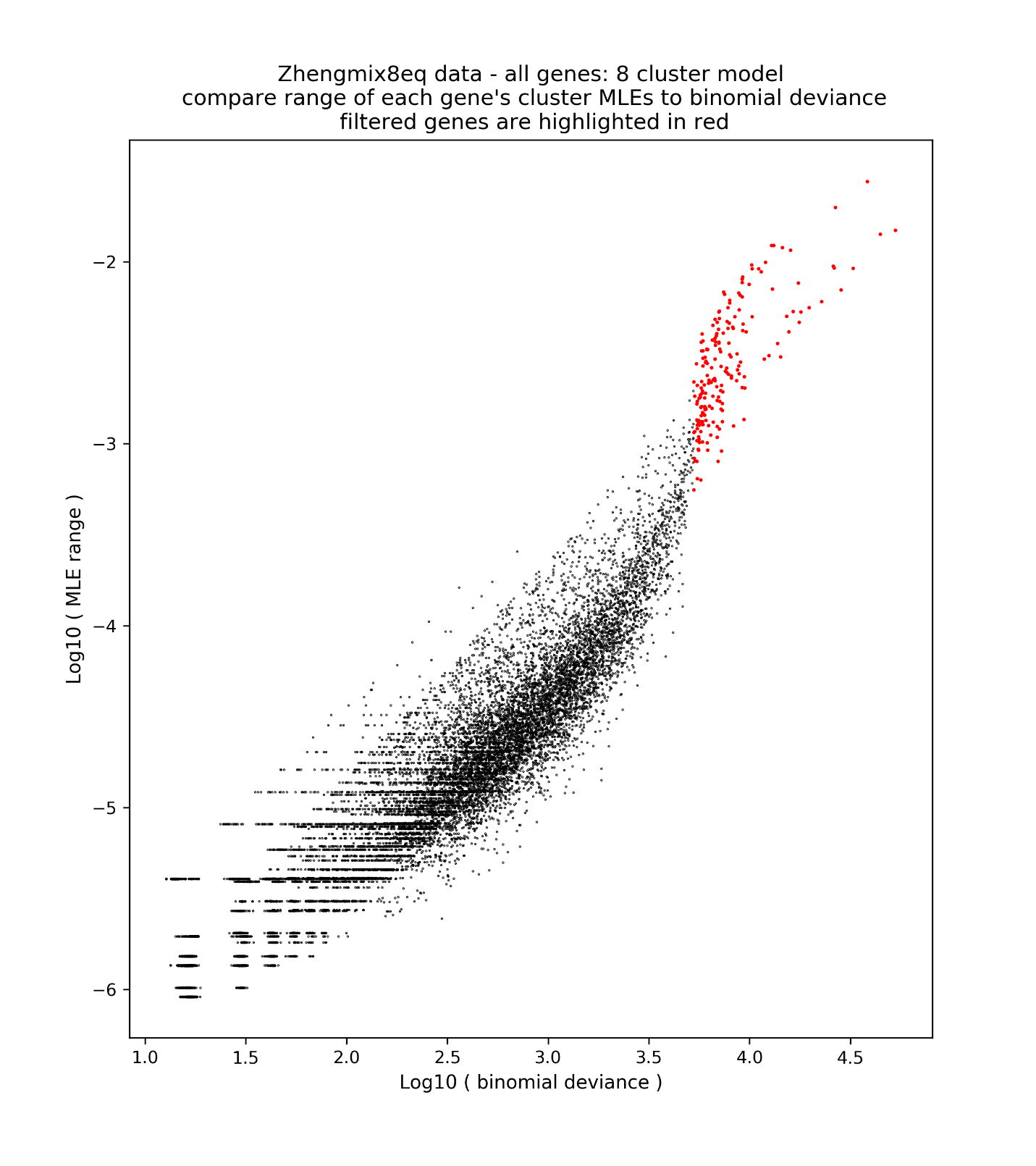
These 4 programs in the folder example\_programs\_simulated prepare and analyze a simulated data set

1. prepare\_synthetic\_data.py
2. compute\_binomial\_deviance.py
3. MLE\_cell\_cluster\_models.py
4. compare\_gene\_range\_of\_MLE\_to\_binomial\_deviance.py

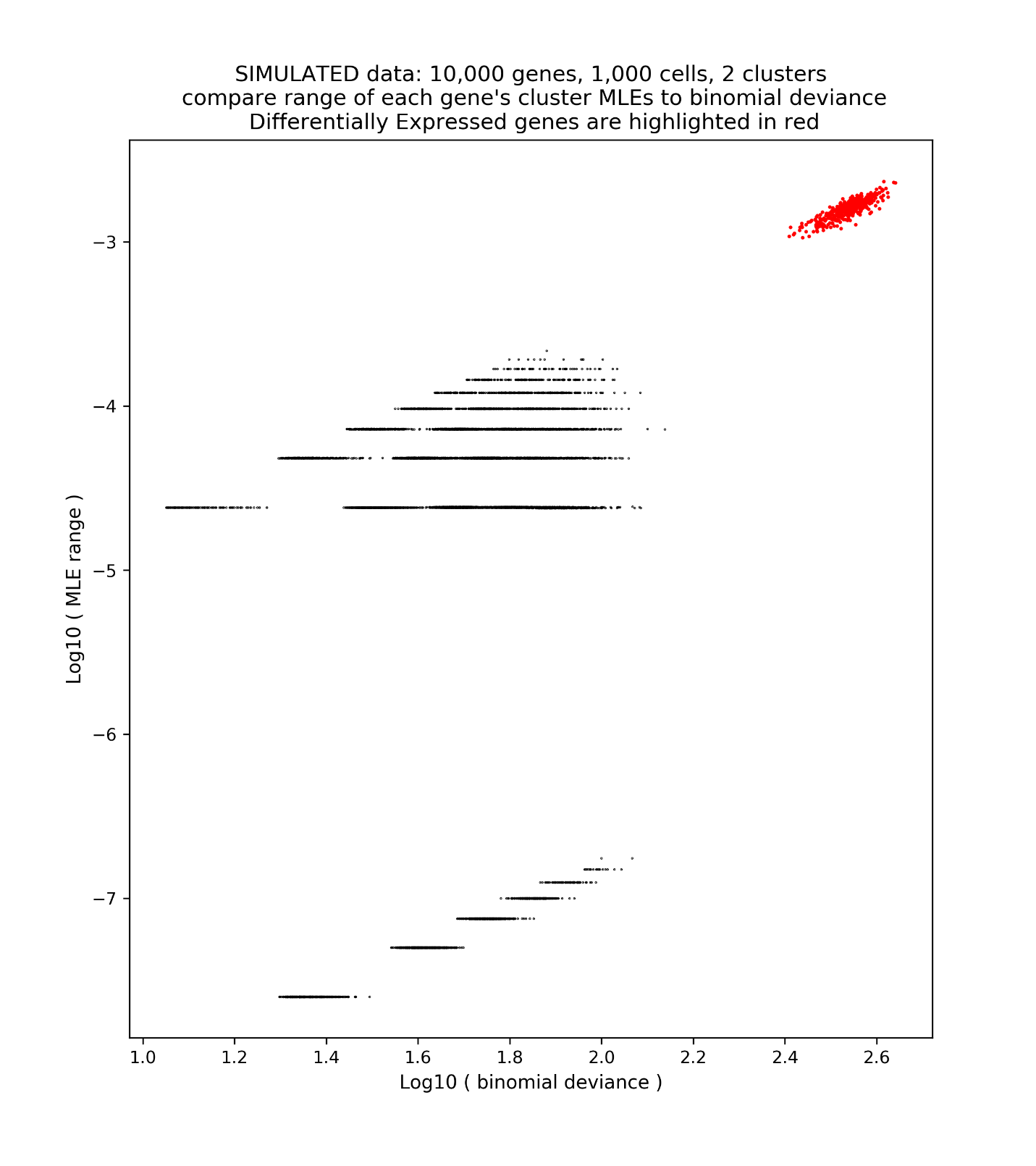
**Figure 1**

****

**Figure 2**



**Figure 3**

****

**Table 1**

Zhengmix4eq data: negative log likelihood and likelihood ratio statistics (with respect to the null model)

|  |  |  |
| --- | --- | --- |
| **number of clusters** | **negative log likelihood** | **likelihood ratio statistic** |
| 1 (null model) | 36,335,630 |  |
| 2 | 36,046,090 | 579,079 |
| 3 | 35,807,741 | 1,055,777 |
| 4 | 35,726,974 | 1,217,311 |
| 5 | 35,673,798 | 1,323,664 |
| 6 | 35,655,843 | 1,359,573 |
| 7 | 35,647,516 | 1,376,228 |
| 8 | 35,611,059 | 1,449,141 |
| 9 | 35,604,618 | 1,462,024 |
| 10 | 35,591,959 | 1,487,342 |
| 11 | 35,586,403 | 1,498,453 |
| 12 | 35,570,655 | 1,529,950 |
| 13 | 35,562,389 | 1,546,482 |
| 14 | 35,552,821 | 1,565,618 |
| 15 | 35,557,454 | 1,556,351 |
| 16 | 35,555,819 | 1,559,620 |
| 17 | 35,548,356 | 1,574,547 |
| 18 | 35,516,618 | 1,638,023 |
| 19 | 35,511,671 | 1,647,917 |
| 20 | 35,510,382 | 1,650,496 |
| 3994 (saturated) | 31,081,322 | 10,508,616 |

**Table 2**

Zhengmix8eq data: negative log likelihood and likelihood ratio statistics (with respect to the null model)

|  |  |  |
| --- | --- | --- |
| **number of clusters** | **negative log likelihood** | **likelihood ratio statistic** |
| 1 (null model) | 37,983,337 |  |
| 2 | 37,725,856 | 514,962 |
| 3 | 37,654,395 | 657,883 |
| 4 | 37,393,837 | 1,178,999 |
| 5 | 37,358,758 | 1,249,158 |
| 6 | 37,327,931 | 1,310,811 |
| 7 | 37,304,176 | 1,358,322 |
| 8 | 37,279,288 | 1,408,097 |
| 9 | 37,269,111 | 1,428,451 |
| 10 | 37,260,380 | 1,445,914 |
| 11 | 37,246,706 | 1,473,262 |
| 12 | 37,226,959 | 1,512,756 |
| 13 | 37,227,131 | 1,512,412 |
| 14 | 37,212,026 | 1,542,622 |
| 15 | 37,207,550 | 1,551,573 |
| 16 | 37,202,579 | 1,561,516 |
| 17 | 37,188,136 | 1,590,402 |
| 18 | 37,172,210 | 1,622,254 |
| 19 | 37,168,933 | 1,628,807 |
| 20 | 37,162,117 | 1,642,439 |
| 3994 (saturated) | 32,499,061 | 10,968,552 |

**Appendix: Randomization to select a filtering threshold**

Genes are ranked by binomial deviance (of the saturated model with respect to the null model) following Townes et al. [1]. This suggests a principled method to select genes using randomization:

* Compute binomial deviance for all genes.
* Compute binomial deviance for randomized data – since randomized data are biologically meaningless, these values should lower-bound the appropriate threshold:
  + Permute each cell’s data – this corrupts gene data.
  + Compute the binomial deviance for the randomized data set.
  + Use the **maximum binomial deviance** for the randomized data as a filtering threshold.
  + Alternatively, perform multiple permutations, calculate binomial deviance for each permuted data set, and use the **median of the maxima.**
* There is no *a-priori* guarantee that this approach will yield **any** genes for analysis. As an example, for the ERCC data set studied by Townes et al.
  + The maximum binomial deviance equals 1,927.
  + For permuted data, the **minimum** exceeds 15,000.
  + **No genes** satisfy the proposed filtering criterion.
  + This is appropriate: Townes et al. write that “We refer to this dataset as the technical replicates negative control as there is no biological variability whatsoever and, in principle, each expression proﬁle should be the same.”

Remarks

* The criterion to use the **median of the maxima** of binomial deviances for randomized data as a filtering threshold works well for the Zhengmix data sets
* For simulated data with low signal to noise ratio, too few genes satisfy the criterion.
* More work may be needed to evaluate competitive thresholds.

**References**

1. Townes F W, Hicks S C, Aryee M J et al.: Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biol* 20, 295 (2019). <https://doi.org/10.118s13059-019-1861-6>
2. Duò A, Soneson C (2020). DuoClustering2018: Data, Clustering Results and Visualization Functions From Duò et al (2018). R package version 1.6.0.
3. Scrucca L, Fop M, Murphy TB, Raftery AE (2016). “mclust 5: clustering, classification and density estimation using Gaussian finite mixture models.” The R Journal, 8(1), 289–317. https://doi.org/10.32614/RJ-2016-021.
4. https://github.com/willtownes/scrna2019 “Downloadable table of results from assessments”