CountSplitting

November 15, 2022

1 Introduction

In the original paper of count-splitting(Poisson thinning). Counts for cell i gene j: X_{ij} are split into two parts: X_{ij}^{train}, and X_{ij}^{test}. For each count, with chance ϵ, it is sent to the train set. Latent variables $L \in \mathbf{R}^{n \times k}$ (pseudotime or cell type labels) are learned in X^{train} and statistical testing is done on X^{test}, thus separating the estimation of $\hat{L}$ and statistical testing on $\hat{L}$.

Correlation between train portion and full dataset has the following expression

$$Cor(X_{ij}^{train}, X_{ij}) = \sqrt{\epsilon}$$

Thus, as ϵ increases, we expect $\hat{L}(X^{train})$ and $\hat{L}(X)$ to look more similar, yet at the same time have less power when performing DE test in the test portion.

2 DE testing

While any DE testing framework could be applied, the author used GLM based approach with Poisson distribution and an exponential link function as a running example. It is easy to estimate overdispersion parameter under this framework as we show later.

For each gene j, we fit the following GLM across all cells

$$\begin{aligned} X_{ij} \sim Poisson(\gamma_i \Lambda_{ij}) &= Poisson(\gamma_i \cdot exp(\eta_{ij})) \\ &= Poisson(\gamma_i \cdot exp(\beta_{0j} + \beta_{1j}^T \hat{L}_i)) \\ &= Poisson(exp(\log \gamma_i) \cdot exp(\beta_{0j} + \beta_{1j}^T \hat{L}_i)) \\ &= Poisson(exp(\log \gamma_i + \beta_{0j} + \beta_{1j}^T \hat{L}_i)) \end{aligned}$$

γ_i denotes the library size (empirically, it takes the value of the total counts of cell i across all genes, normalized by the average total counts of all cells). $\hat{L}_i$ denotes latent space (for example pseudotime or celltype cluster estimates) of cell i. We assume both γ_i and $\hat{L}_i$ are known/fixed or already estimated.

DE test is performed per gene independently on vector β_1 with a Wald test. Large values in β_1 indicate the corresponding column in estimated latent space $\hat{L}$ has non-trivial contribution to the expected counts.

Note that when fitting the GLM with Poisson or with Negative Binomial later we add the (Log) library size as an offset, which is a component of a linear predictor that is known in advance (no need to learn coefficient associated with it). Offset and intercept are different!

3 Overdispersion

One empirical question we might want to ask is what if the process is not well characterized by Poisson? Over-dispersion issue can arise from the zero inflation property of single cell data. The author mentioned that only when the process is well modeled by Poisson do we get the theoretical guarantee that $cor(X_{ij}^{train}, X_{ij}^{test}) = 0$. Correlation between train portion and test portion is proportional to $\frac{\Lambda_{ij}}{b_j}$, which becomes the key coefficient the rest of the analysis will focus on.

$$Cor(X_{ij}^{train}, X_{ij}^{test}) = \frac{\sqrt{\epsilon(1-\epsilon)}}{\sqrt{\epsilon(1-\epsilon) + (\frac{b_j}{\Lambda_{ij}})^2 + \frac{b_j}{\Lambda_{ij}})}}$$

3.1 Negative Binomial vs Poisson

Here we attempt to model it by the negative binomial distribution, which use additional parameter to characterize the variance, allowing it be larger than mean.

$$X \sim Poisson(\Lambda)$$

$$\mathbf{E}(X) = \mathbf{Var}(X) = \Lambda$$

$$X \sim NB(\Lambda, b)$$

$$\mathbf{E}(X) = \Lambda$$

$$\mathbf{Var}(X) = \Lambda + \frac{\Lambda^2}{b}$$

Λ denotes the expected mean of the negative binomial and b denotes the dispersion (paper notation: b, GLM tutorial notation: k, MASS package notation: θ). if b goes to infinity then the NB reduces to Poisson. Specifically:

$$\mu = k * \frac{p}{1-p}$$

$$\sigma^2 = \mu + \frac{\mu^2}{k}$$

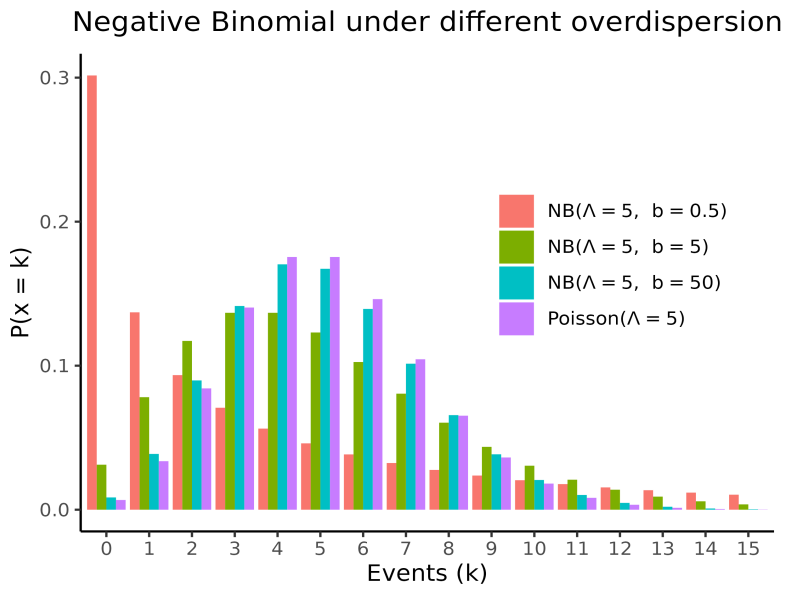

Figure 1: Poisson vs NB with various overdispersion parameters. All have expected occurence of 5

3.2 DE Simulation Result under NULL

if we have overdispersion, in general we can expect count-splitting wont do worse than the double-dipping approach at controlling false positive but less effective at controlling the inflation/deflation in the null model than if the counts are well approximated by Poisson. Data are simulated with no real structure in the expected counts. Yet, pseudotime latent space L is still estimated with top PCs. DE test is performed on the β_1 with Wald test under the GLM framework with NB.

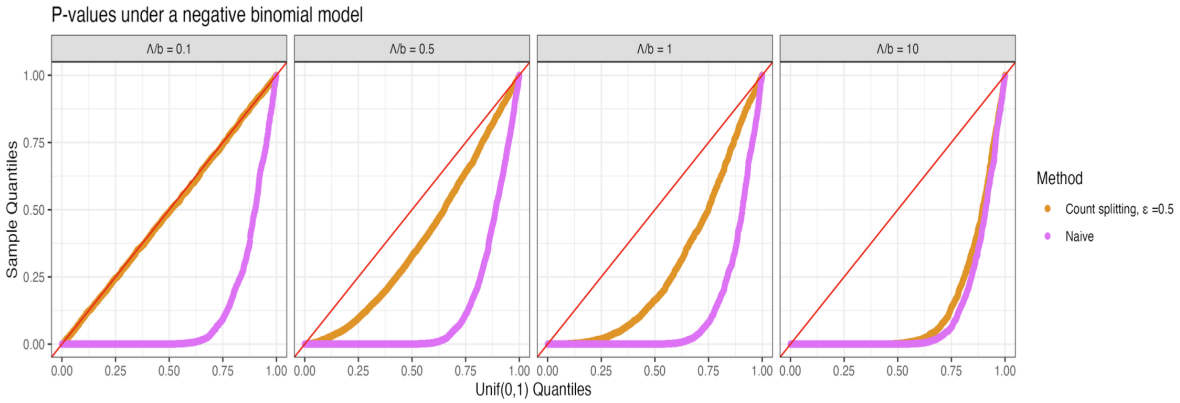

Figure 2: Controlling FP under NB distribution with various degree of overdispersion

3.3 Estimate the overdispersion coefficients in real dataset

Previously, CountSplit method proposes to fit the following model on the test portion of the counts, using the latent space learned from the train.

$$X_{ij} \sim Poisson(\Lambda_{i,j})$$

$$\log(\Lambda_{i,j}) = \log \gamma_i + \beta_{0j} + \beta_{1j}^T \hat{L}_i$$

Here, for each gene j we fit a GLM with NB and (log) library size as an offset to estimate β_{0j}, β_{1j} and b_j simultaneously. Then we can calculate the expected mean for each cell i: Λ_{ij} and then get the key overdispersion coefficient we are interested in $\frac{\Lambda_{ij}}{b_j}$. In practice, when estimating the overdispersion coefficient, the author doubledipped the entire data (X is the entire count matrix, and $\hat{L}$ is also learned using the full count matrix), and justified doing so by claiming they are simply trying to have a rough estimate of this coefficient.

$$X_{ij} \sim NB(\Lambda_{i,j}, b_j)$$

$$\log(\Lambda_{i,j}) = \log \gamma_i + \beta_{0j} + \beta_{1j}^T \hat{L}_i$$

3.4 Estimate the overdispersion coefficients in PBMC3k

$L \in R^{n \times k}$: Latent variable is obtained similar to what the original count splitting paper did but just done with "Seurat" package instead of "scran" and "monocle3". Both packages perform CPM, log transformation, select Highly Variable Genes, project to PCA space based on normalized and scaled data on HVG, clustering and UMAP visualization. Since we are interested in DE genes between celltypes, we one-hot encoded $\hat{L}$. To avoid co-linear/singular issue, we drop the first big cluster column. Thus, $L \in R^{n \times (nClusters-1)}$ We also try just fitting the raw PC scores directly as $\hat{L}$ and the results are quantitatively similar.

Red vertical line denotes $\frac{\Lambda_{ij}}{b_j} = 1$ which still outperforms the naive approach based on simulation results from the original paper. The cyan vertical line denotes $\frac{\Lambda_{ij}}{b_j} = 10$ at which point count splitting performs indistinguishably from the naive double-dipping approach.

Overdispersion coefficients for all cells on top 2000 Highly Variable Genes selected by Seurat are shown on the left. Results for all cells on a set of randomly selected 2000 genes are shown on the right. HVG set in general shows a stronger deviation from Poisson.

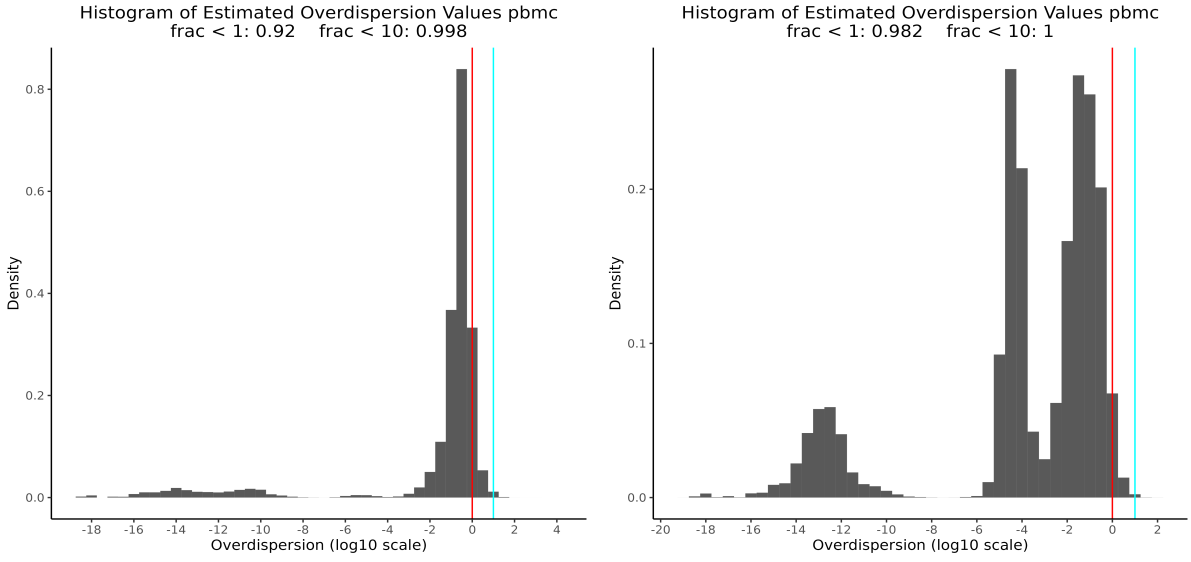

Figure 3: dispersion on RNA pbmc

3.5 Estimate the overdispersion coefficients in Mouse Brain Data: HPC

Exact same procedure but performed on a mouse brain dataset (Hippocampus region)

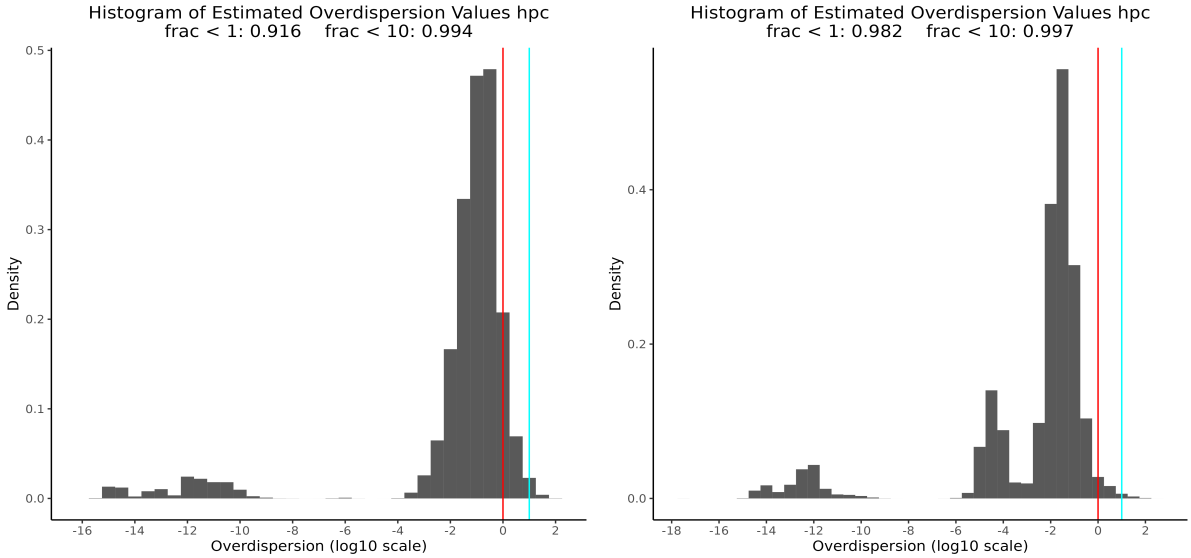

Figure 4: dispersion on RNA hpc

3.6 Estimate the overdispersion coefficients in Mouse Brain Data: AMY

Exact same procedure but performed on a mouse brain dataset (Amygdala region)

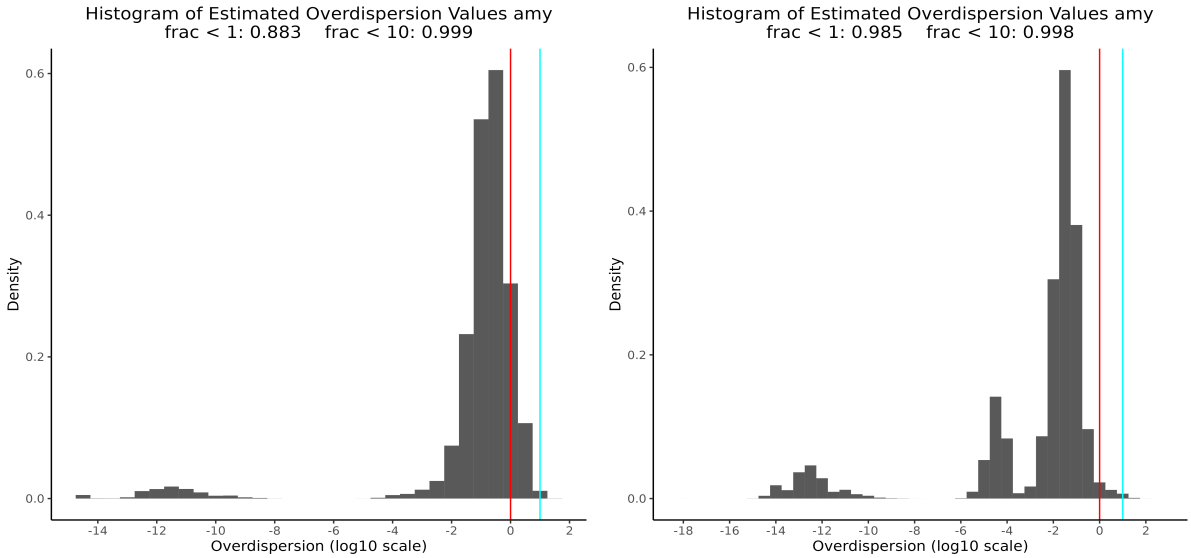

Figure 5: dispersion on RNA Amy

4 Differential expression between cell types

Here, we perform a simple DE test (wilcox ranksum) between cell-type clusters. "Full" denotes learning $\hat{L}$ using the entire dataset and subsequently DE test on entire dataset, hence the naive/double-dipped approach. For the CountSplit version, we send 50% of the counts to Train set. "Train" denotes learning $\hat{L}$ using X^{train} and subsequently DE test on X^{train}. "Test" denotes directly using $\hat{L}$ estimated under X^{train} and perform DE test on X^{test}. Finally, we also calculate "Test denovo" which learns $\hat{L}$ and DE test on X^{test}.

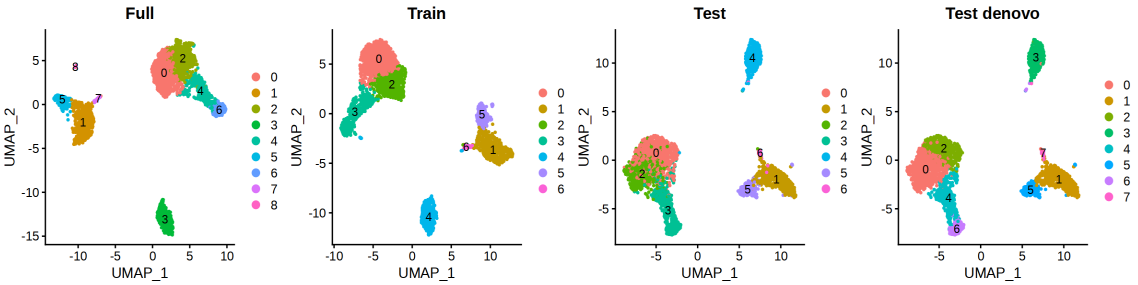

Figure 6: UMAP

We focus on 2 major cases: (1) clear distinct celltype cluster that shows up in train and is well preserved in test (in X^{test} low dim UMAP embedding, coloring each cell with train's label show high concordance, aka clear cluster/color separation boundary and same color within test cluster. train's label should also be consistent with denovo clustering in X^{test}, indicated by high rand-index) (2) a more tricky separation of subtypes within a major cell-type that shows up in train but less well preserved in test. (in X^{test} low dim UMAP embedding, coloring each cell with train's label shows poor cluster boundary and mixed colors)

4.1 B cells

For the first case, we pick B cells as an example. This corresponds to cluster 3 in "Full", cluster 4 in "Train" and cluster 3 in "Test denovo".

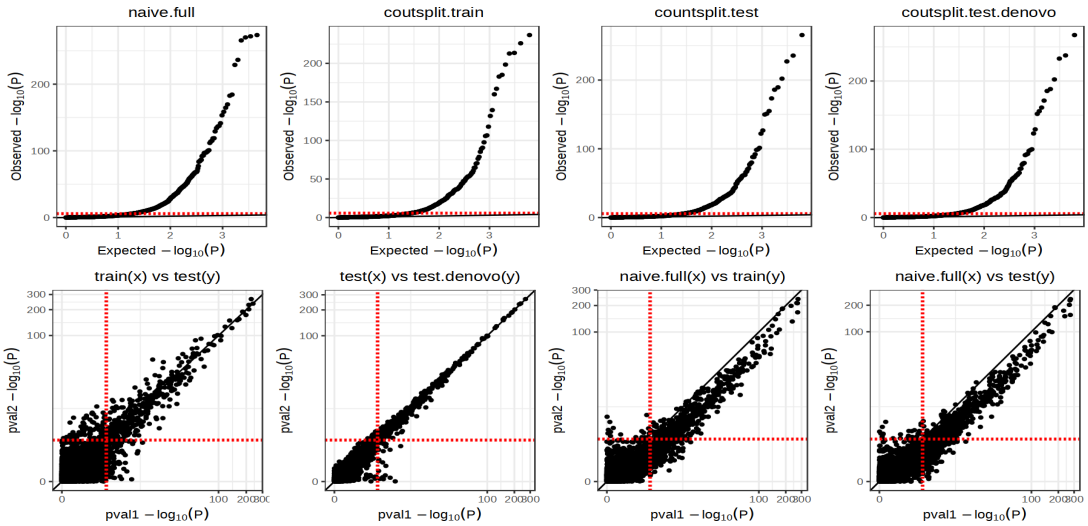

Figure 7: Bcells. First roll indicates that qq plots are still inflated. First panel in the second roll shows the head to head comparison between double-dipping train half vs CountSplit test DE stats. pvals are on the same scales, indicating the cluster is real. Second panel in the second roll shows the marginal calibration on top of DE test pvals obtained via double dipping in test. There is almost no difference.

4.2 CD4 subtypes

For the second case, we pick CD4 memory T cell as an example. This corresponds to cluster 0 in "Full", cluster 0 in "Train" and cluster 2 in "Test denovo". We first check it against its neighbour CD4 naive T cells, then we check it against the rest of the cells.

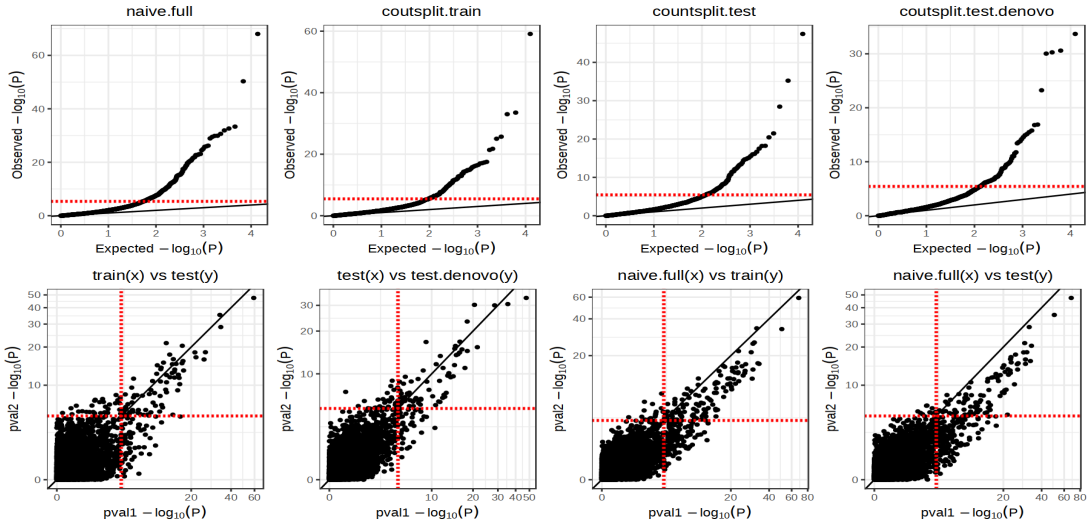

Figure 8: CD4-memory-vs-CD4-naive. First panel in second row shows slightly better calibration of count-split test DE stats over double-dipped train DE stats. It is expected as observed in the UMAP, the boundary between these 2 closely related cell types are not as well defined as in the train. Second panel in the second row shows minimal marginal calibration on top of DE test pvals obtained via double dipping in test.

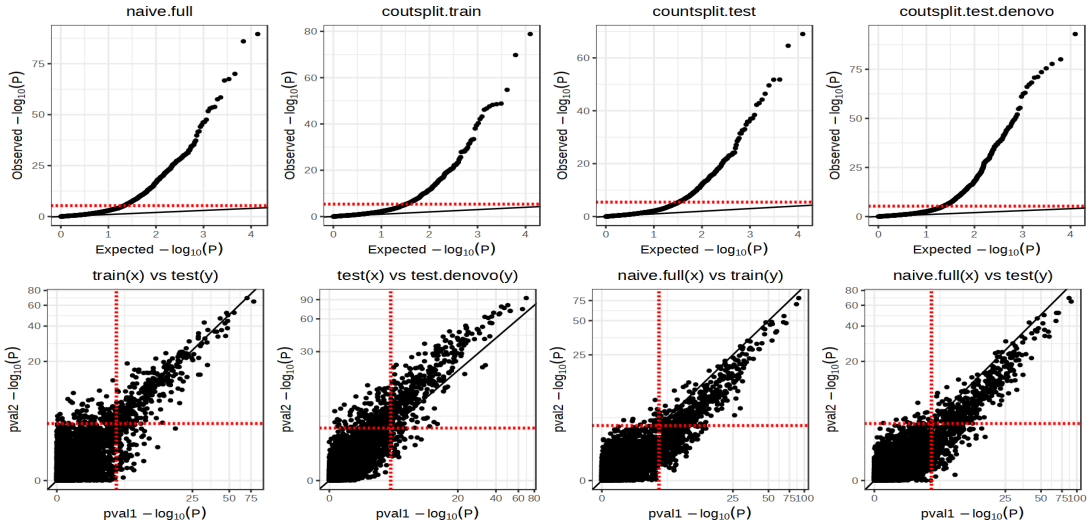

Figure 9: CD4-memory-rest. First panel in second row shows almost no additional calibration of count-split test DE stats over double-dipped train DE stats. Second panel in second row shows relative strong marginal calibration on top of DE test pvals obtained via double dipping in test.

5 Discussion

The general idea of CountSplitting is that once the counts are split into train set and test set, user can get better calibrated test statistic in the test portion of the data using the labels derived from the train set. There is no restriction on the exact type of DE test you want to perform.

This paper mostly focuses on "negative controls": data are simulated with no real structure. CountSplitting is very effective at controlling for the FP here: artificial clusters labeled in the train set will have uniform DE p-values in the test set. However, CountSplitting is going to be less effective if the structure observed in train is real. Specifically, if denovo clustering performed in test portion agrees with the clustering labels derived from the train portion of the data. Then since we are performing the same statistical DE testing. Over-dipping version of the DE test statistics (denovo clustering and DE in test portion) will agree with the CountSplitting version of the DE test statistics (cluster in train and DE in test portion). As we never change the type of DE test we perform, QQ plots will be as inflated as before.

Empirically, highly variable genes that supposedly are more informative marker genes for one cell type or a set of cell types have larger deviation from Poisson distribution, than a set of randomly selected genes. This is the case for the top 2000 HVG compared to a set of randomly selected 2000 genes. The difference is larger if we become more strict and focus on the top 100 HVG genes. Hence, the test statistics even under null might be less well calibrated in this strata of more interesting genes.

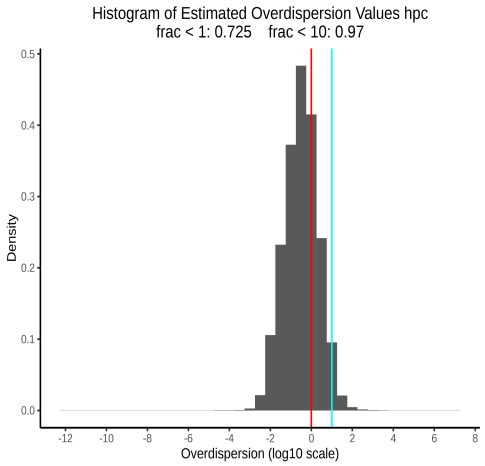

Figure 10: overdispersion in hpc top200-HVG

6 Useful resource

CountSplitting Paper
CountSplitting Tutorial
CountSplitting Figure Codes
Tutorial on GLM with negative binomial
Explain the parameter estimated from MASS glm.nb