**Benchmarking computational demultiplexing tools**

Timothy Bi

BMI 5332 Final Project

**Problem Statement**

Sample multiplexing is often used in single-cell sequencing experiments to cut costs and reduce the effect of technical artifacts, such as batch effect or multiplets, on downstream analyses. However, this often necessitates the use of computational demultiplexing tools after sequencing in order to accurately recapitulate the original populations. There exist many such different tools, each with different underlying statistical assumptions, yet little is known about how tool performance compares across different experimental conditions. This study aims to formally benchmark the performance of six demultiplexing tools / algorithms using simulated datasets representing conditions of different sample sizes and staining qualities.

**Significance and Background**

Single-cell sequencing technologies have revolutionized our understanding of human physiology and disease1–3. Nevertheless, the relative cost of sequencing remains high and throughput remains low for most applications4. This is especially true when we account for recent technological advancements which have enabled simultaneous profiling of additional modalities such as surface protein expression5 and chromatin accessibility6, as well as various combinations7–9. Sample multiplexing, a technique which has been used in a variety of applications ranging from mass10 to flow cytometry11,12 and quantitative proteomics13, has also seen extensive application in the single-cell space in an attempt to reduce costs and increase throughput14–17. Given the high cost-per-cell of single-cell experiments, however, and the rarity of cell subsets which are commonly studied, accurate sample demultiplexing and high cell recovery is a priority. Furthermore, whereas most multiplexing techniques in other single-cell applications like flow or mass cytometry involve direct staining with a cell dye, single-cell multiplexing is carried out either by staining cells with an antibody against a ubiquitous surface protein (such as β2-microglobulin, which is expressed on all nucleated cells in humans) conjugated to a short oligonucleotide tag18 or by oligonucleotide-conjugated lipids, which can incorporate directly into cell plasma membranes19. Cells can also be demultiplexed if they are genetically distinct by using single nucleotide polymorphisms20,21, though this can be complicated by the relative sparsity of single cell reads and sequencing errors. All these factors have motivated the development of many different bioinformatic tools to computationally demultiplex single cell data with greater accuracy19,22–28.

Given the abundance of tools available, there is the need for a systematic comparison of performance under different experimental conditions. One such benchmarking study was carried out recently by researchers at the University of Melbourne29, which served as the inspiration for this project; nevertheless, exploration of other experimental parameters would necessitate the acquisition of another dataset, which may prove costly and impractical. Therefore, in this project I will develop a framework for generating simulated count datasets, which can be modified in the future to perturb the experimental condition of choice. In addition, I will benchmark six computational algorithms with different underlying statistical assumptions: 1) bimodal flexible fitting raw (BFF\_raw)22, which assumes an inherently bimodal distribution and sets a threshold which divides the data into positive and negative distributions; 2) BFF\_cluster, which operates under a similar assumption as BFF\_raw but with an additional bimodal quantile normalization step before determination of a threshold; 3) GMM-Demux25, which also assumes a bimodal distribution but performs threshold determination on centered log ratio transformed values instead of on count values directly; 4) HashedDrops24, which decides a threshold based on a simple log-fold change cutoff; 5) HTODemux23, which assigns cells based on k-means clustering on count data; and 6) demuxmix26, which assumes an underlying negative binomial distribution and assigns each cell a probability of belonging to either a positive or a negative distribution.

**Aims and Rationale**

*Aim 1: Generation of simulated datasets.* As mentioned previously, generating single cell sequencing datasets is expensive and is impractical if done purely for benchmarking purposes. Simulated datasets enable us to test the effects of a variety of different conditions, including but not limited to sample size, staining quality, multiplet rate, etc. To accurately model count data, I chose to use a negative binomial distribution30. In this specific proposal, I chose three sample sizes (12,000, 8,000, and 4,000 cells) and three staining qualities (high, medium, and low) for a total of nine test samples, with each sample containing cells evenly stained by eight different barcodes.

*Aim 2: Benchmark tool performance on simulated datasets.* Determining tool performance under differing conditions of sample size and staining quality is important for researchers to determine which tool is most appropriate for their data. Depending on the benchmarked findings, it may be more appropriate to select one tool which offers the best performance across a variety of staining qualities and sample sizes, or to choose specific tools which outperform the rest under very specific circumstances. I hypothesize that HTODemux, a clustering-based algorithm, will have higher error rates as sample size decreases due to decreasing cluster size and confidence. I also hypothesize that all tools will perform worse as staining quality decreases.

**Methods**

***Dataset simulation.*** Cells from each sample were divided into eight even groups and designated as positive for one barcode (ground truth). Count values were simulated for each cell by sampling from either a higher negative binomial distribution for positive barcodes or from a lower negative binomial distribution for negative barcodes using the rnbinom() function in R. The lower negative binomial distribution has parameters mu = 20, size = 10. The upper negative binomial distribution has parameters mu = 100, 80, or 60 (corresponding to a staining quality of “high”, “medium”, or “low”, respectively).

To simulate doublets in the data, I first determined doublet rate using a calculator available at <https://satijalab.org/costpercell/>, defining the doublet rate as 0.43%, 0.91%, or 1.47% for sample sizes 4,000, 8,000, and 12,000, respectively. For each doublet, I used the sample() function in R to randomly choose 2 of 8 barcodes (replace = T to allow for doublets containing two cells with the same barcode). Counts were simulated for each cell in the doublet as above before summing together. See **Fig. 1** for schematic.

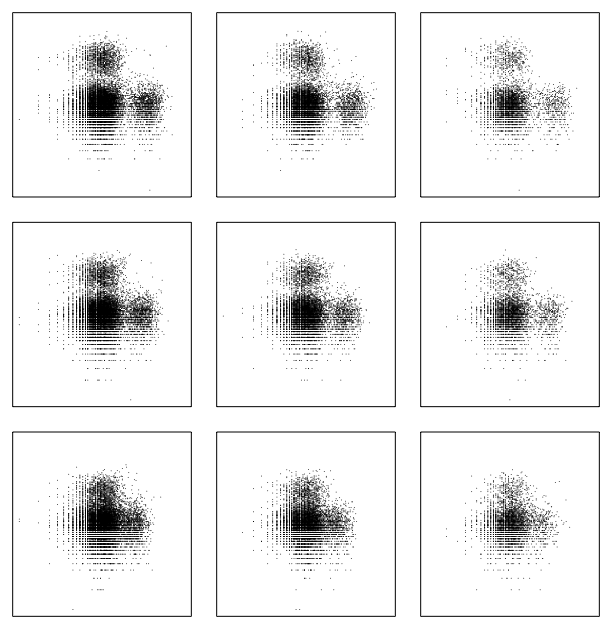
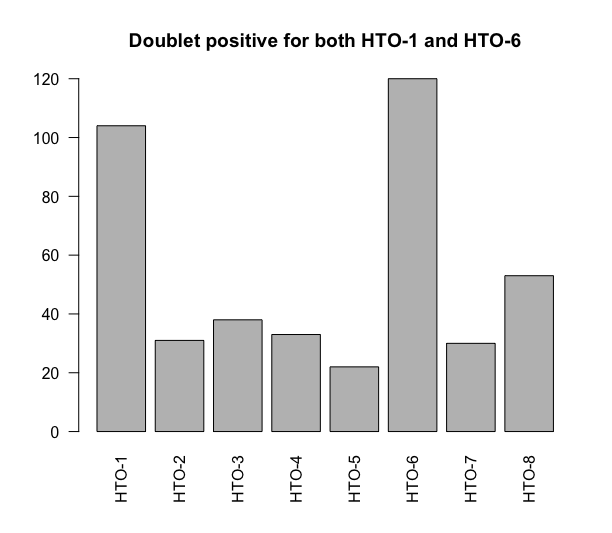
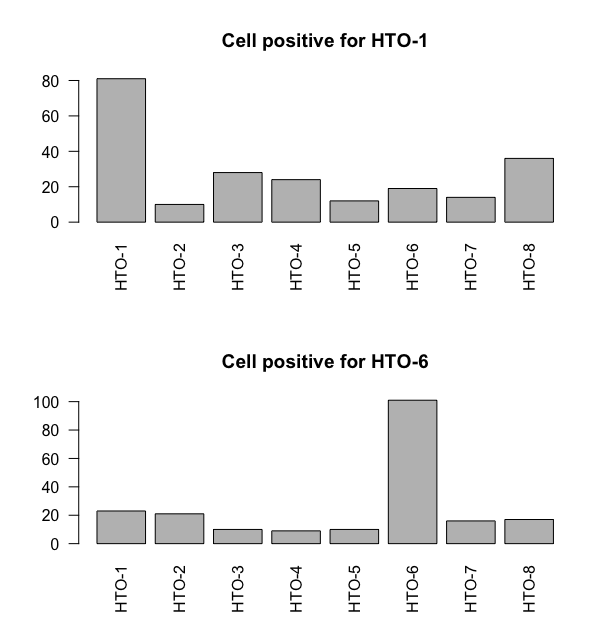
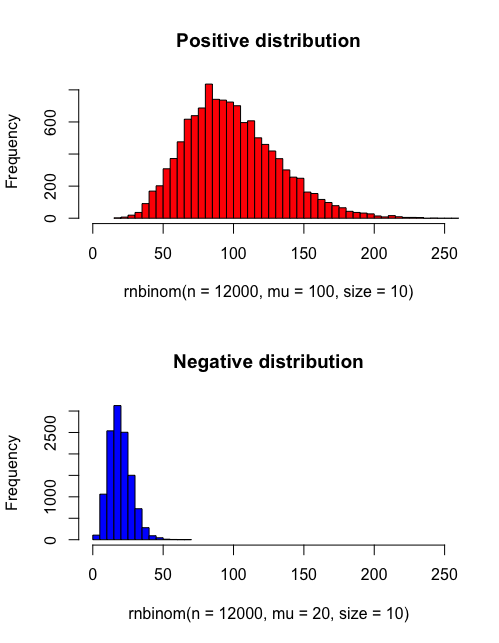
***Implementation of callers.***This benchmarking experiment includes six callers: BFF raw, BFF cluster, HashedDrops, HTODemux, demuxmix, and GMMDemux. BFF raw, BFF cluster, HTODemux, and GMMDemux were all implemented using the cellhashR package. All default parameters were used except setting positive.quantile to 0.99 for HTODemux. HashedDrops and demuxmix were implemented separately using their respective R packages due to difficulties with cellhashR (demuxmix in particular cannot be run in cellhashR without a gene expression covariate matrix). For HashedDrops, cutoff parameters were set to confident.min = 0.5 and doublet.min = 0.5. Default parameters were untouched for demuxmix. Each caller was run on all nine simulated samples generated above.

(3)

(2)

(4)

(1)



HTO-1

HTO-6

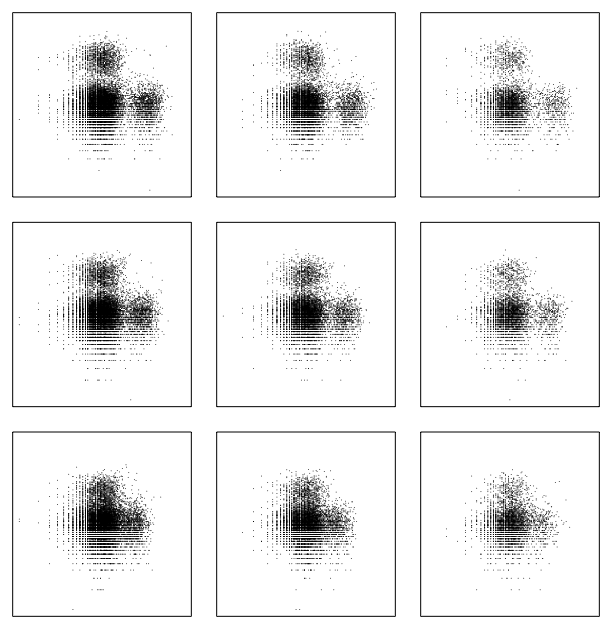
**Figure 1. Schematic for simulation of count data.** Counts for singlets are sampled from a “positive” (red) or “negative” (blue) negative binomial distribution (1). Of these simulated cells (2), a small number (determined by multiplet rate) are combined to form doublets (3), which are included in the final cell matrix (4).

***Tool benchmarks.*** In this project, we considered four major benchmarks: comparison of overall classification to ground truth, singlet assignment accuracy, doublet assignment accuracy, and F-score calculation (overall performance metric).

* Overall classification for each tool was determined by proportions of singlet, doublet (all multiplets are assumed doublets), negative, and uncertain calls (only for demuxmix).
* Singlet assignment accuracy was measured by the percentages of singlets which were assigned the wrong barcode, singlets inaccurately defined as doublets, and singlets which the algorithm could not make a call for.
* Double assignment accuracy was determined by comparing proportions of doublets assigned correctly as doublets versus incorrectly assigned as negative or as singlets. Comparison was also made of doublet assignment rates between doublets consisting of two cells with the same barcode versus two cells with different barcodes.
* The F score for a tool is calculated by the following equation:

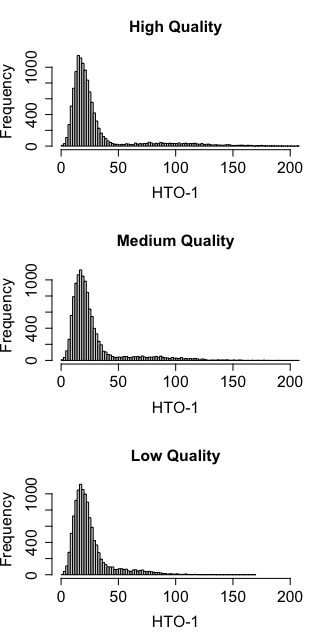
where TP is defined as the number of cells correctly assigned by the tool, FP as the number of cells where an incorrect assignment is made (in the case of singlets, this includes either singlets assigned the wrong barcode or singlets assigned as doublets; in the case of doublets, this includes doublets assigned as singlets), and FN as the number of cells where no call is made.

**Figure 2.** **Plots of counts from simulated datasets.** a) Representative histograms of HTO-1 counts at high, medium, and low staining qualities for a sample of 12,000 cells. b) Log-transformed plots of HTO-1 versus HTO-2. From left-to-right: 12,000, 8,000, and 4,000 cells. From top-to-bottom: high, medium, and low staining qualities.



HTO-1

HTO-2



**B.**

**A.**

High (mu = 100)

Med (mu = 80)

Low (mu = 60)

12,000 cells

8,000 cells

4,000 cells

**Results**

***Dataset simulation results***

Simulation of datasets successfully generated expected relative distributions (**Fig. 2a**). As staining quality decreases, it is apparent that the separation between “positive” and “negative” distributions decreases. Nevertheless, even at low staining qualities, there still appears to be a positive population that can be roughly identified by eye. Log-transformed plots (**Fig. 2b**) of hashtag 1 (HTO-1) versus HTO-2 show a distinct population of singlets positive for HTO-1, a separate population of singlets positive for HTO-2, a large population negative for both HTO-1 and HTO-2 (containing singlets of cells positive for the other six HTOs), and a few doublets positive for both HTO-1 and HTO-2.

***Demultiplexing benchmarks***

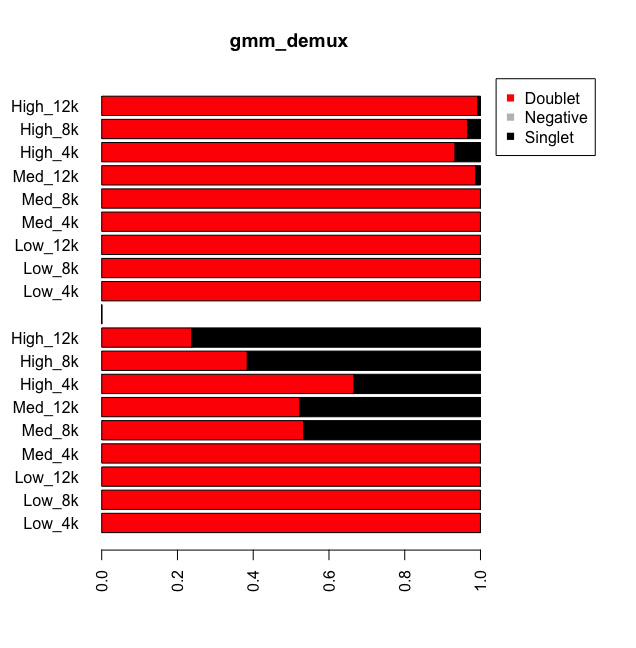
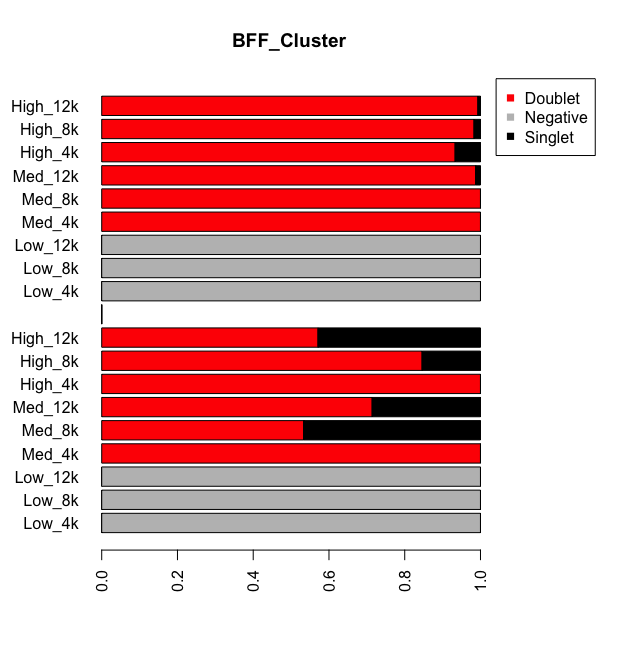
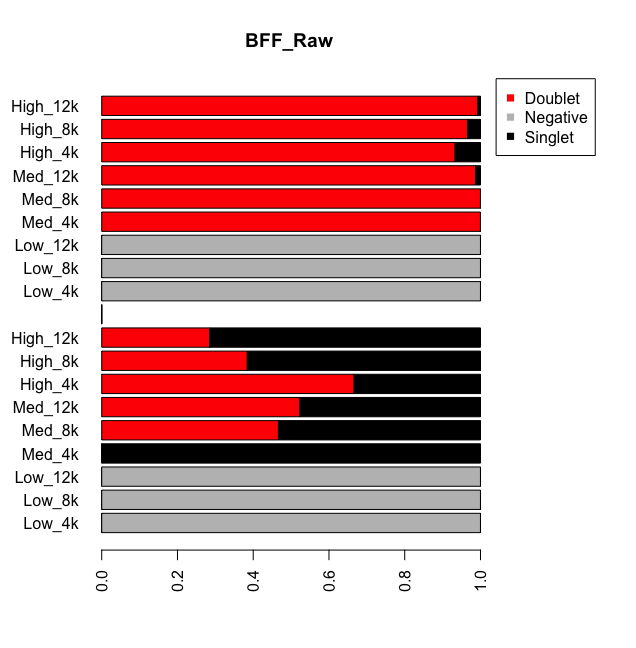
*Comparison of overall calling*

**Fig. 3a** shows the comparison of overall calls made by each caller to ground truth. As we hypothesized, every caller performs more poorly as sample size decreases. In particular, we note that BFF cluster appears to have the closest assignment accuracy at high and medium staining qualities, but both BFF raw and BFF cluster completely fail to generate any calls at low staining qualities. HTODemux, being the only clustering-based algorithm we benchmarked, appears to generate roughly the same percentage of doublets regardless of changes to staining quality. GMM-Demux interestingly calls very few doublets until low staining quality is defined, after which it calls an extremely high proportion of doublets. Nevertheless, it is also apparent that even at high staining qualities, no tool is exactly perfect. Note that while we had proposed to benchmark

**C.**

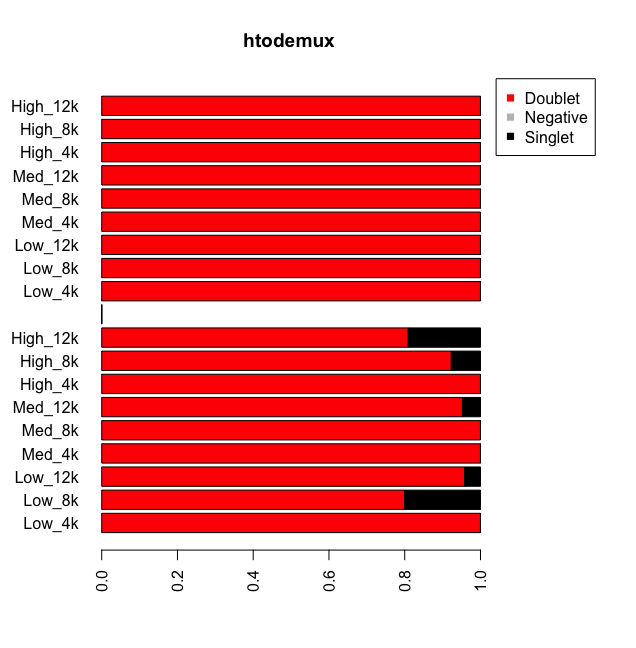
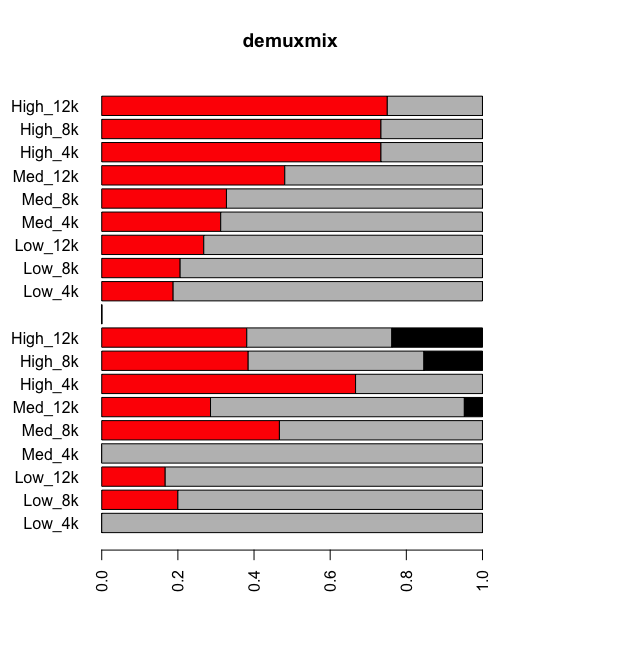
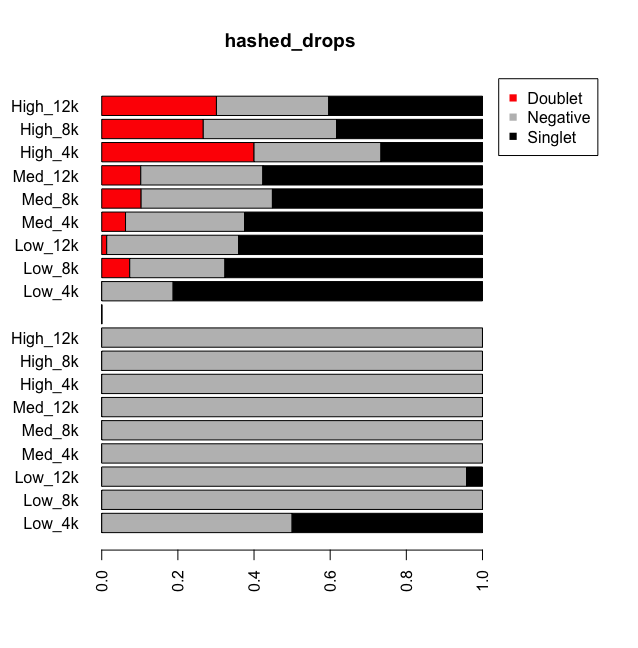
**B.**

**A.**



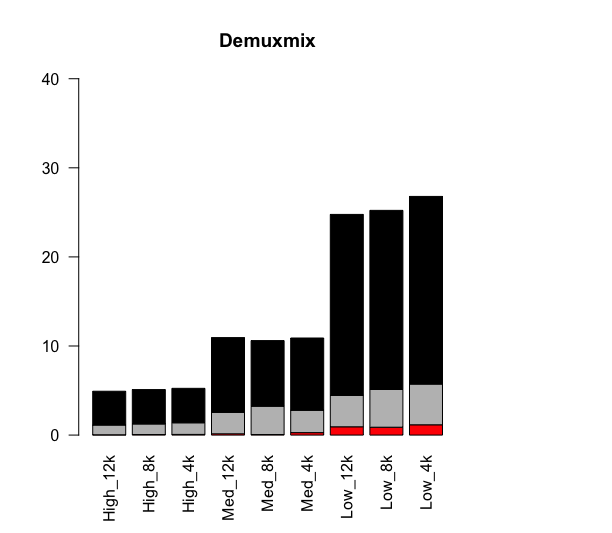
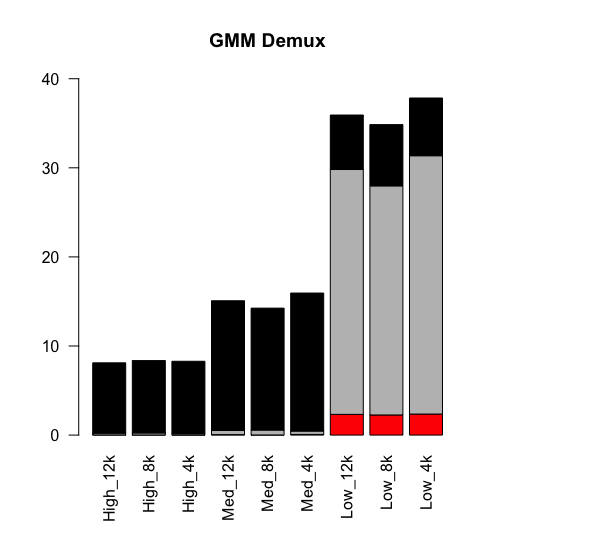
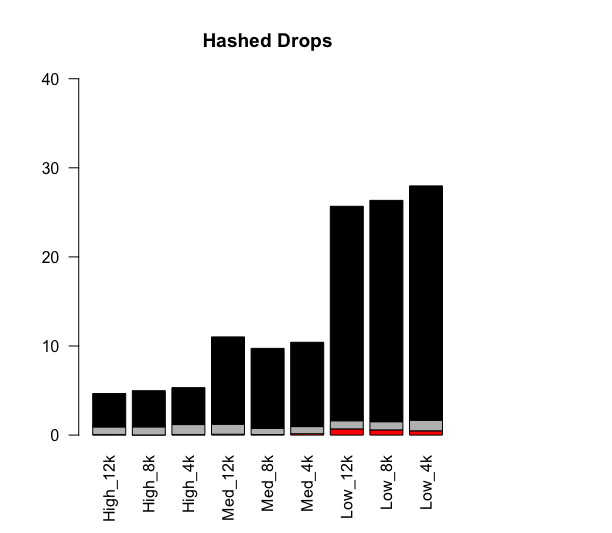
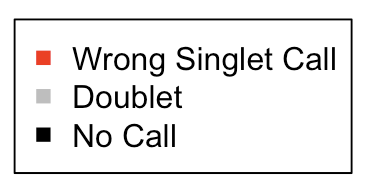
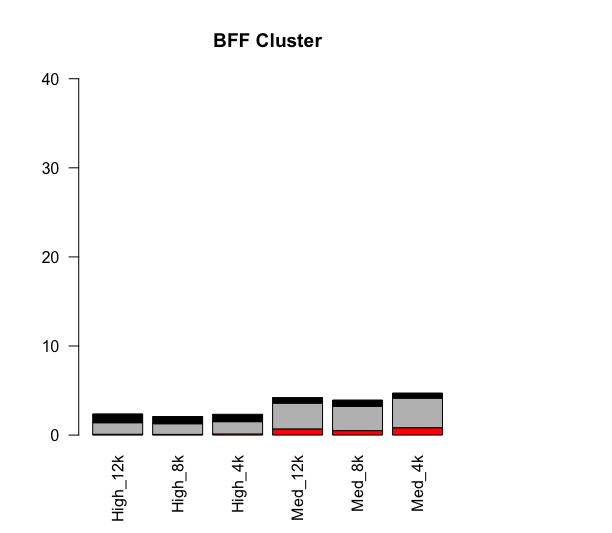
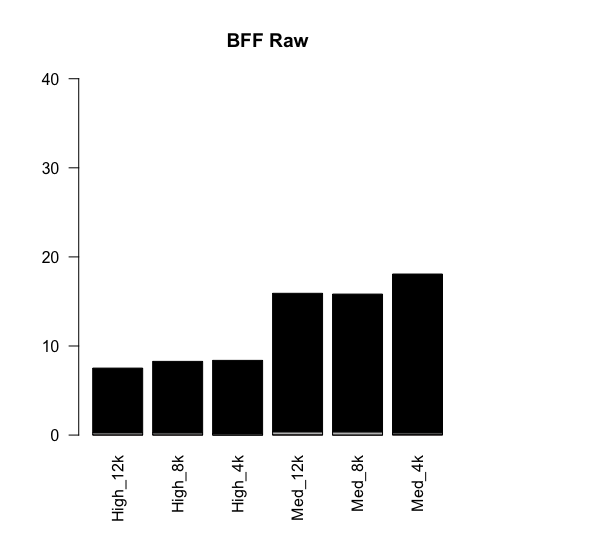
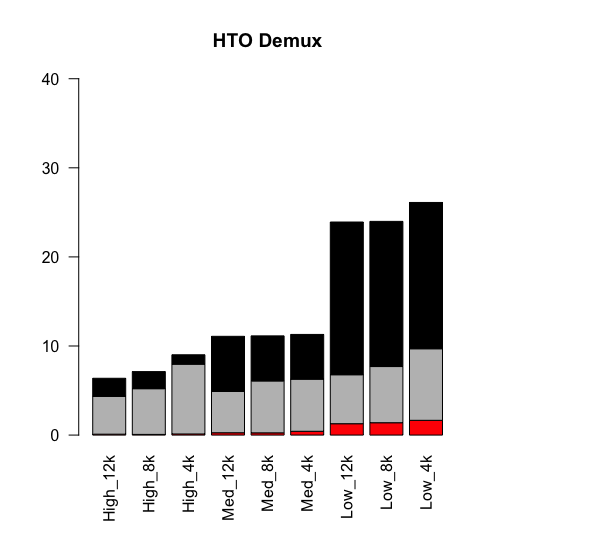
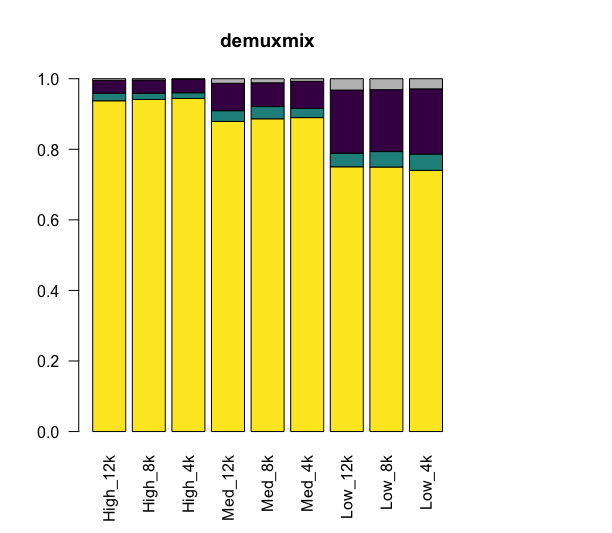
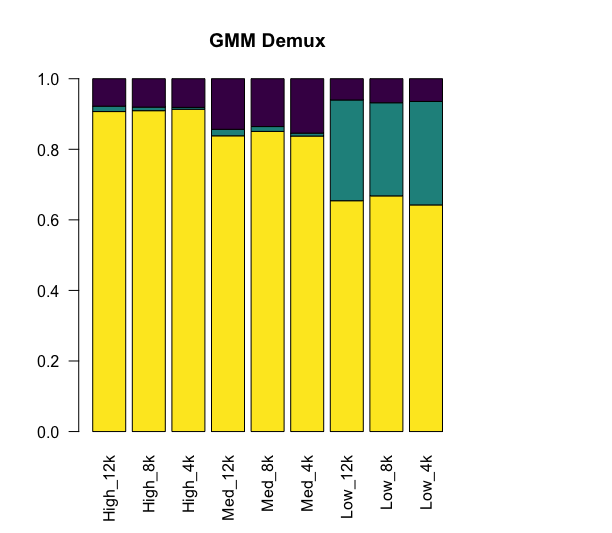
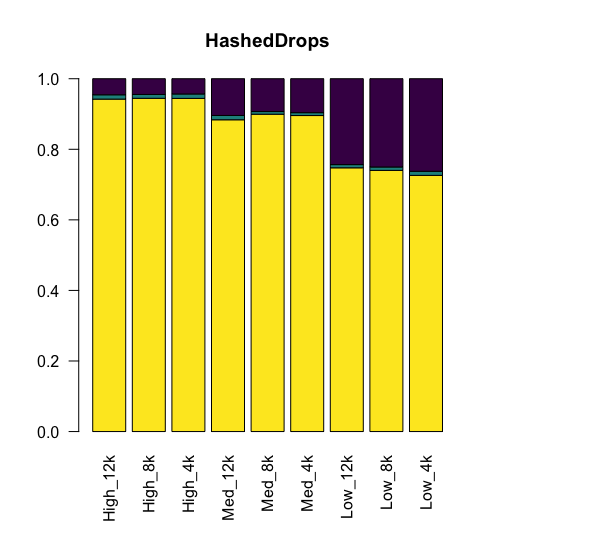
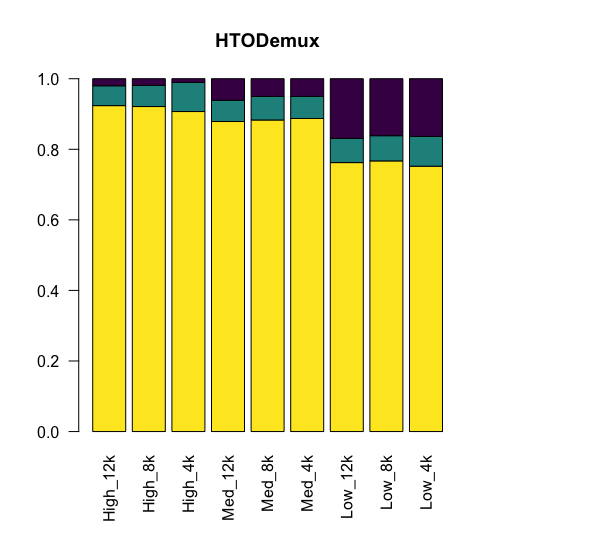
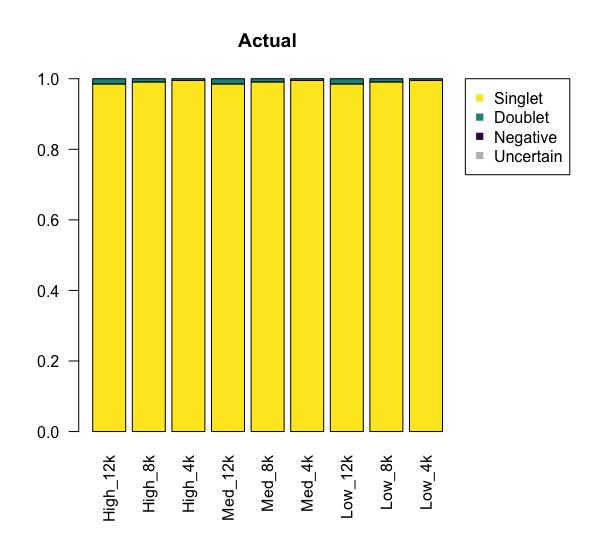
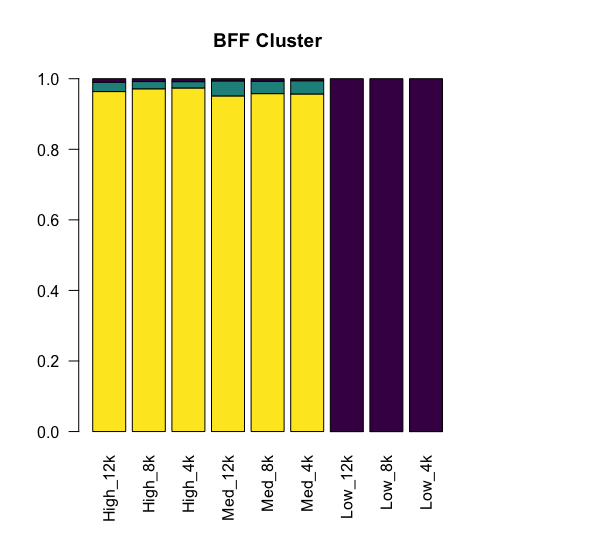
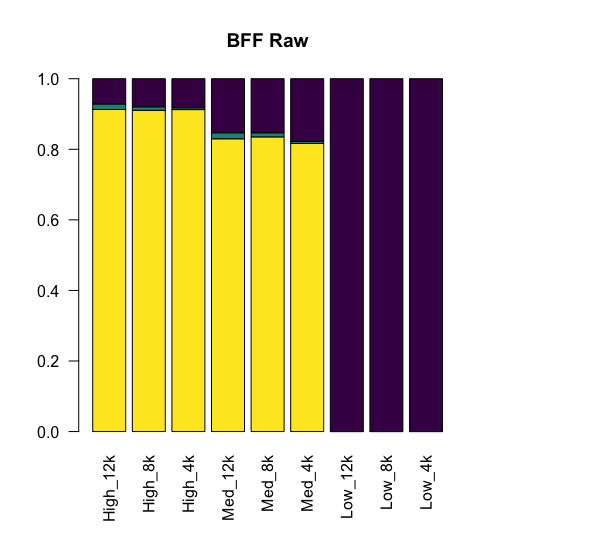
Different barcode

Same barcode



Different barcode

Same barcode



**Figure 3. Demultiplexing benchmarks for each tool, including overall assignment, singlet assignment error rates, and doublet assignment.** a) Comparison of the proportion of overall assignments by each demultiplexing tool with actual ground truth (leftmost panel). “Uncertain” is a unique call returned by demuxmix due to its calculation of posterior probability for the assignment of barcodes and is not returned by any other caller. b) Comparison of singlet assignment error rates across experimental conditions. Only high and medium staining quality conditions were plotted for BFF raw and BFF cluster due to complete failure to generate any calls at low staining qualities. “Wrong singlet call” refers to singlets which were correctly identified as singlets but were assigned the wrong hashtag. “Doublets” refers to singlets which were incorrectly identified as doublets. “No call” refers to singlets which were incorrectly determined to be negative for all hashtags. c) Comparison of doublet assignment accuracy. “Different barcode” refers to doublets with two cells positive for two different barcodes. “Same barcode” refers to doublets with two cells positive for the same barcode.

DemuxEM previously, its algorithm could not be run without a gene expression matrix covariate. As such, we ultimately proceeded without including it in any further analyses. Finally, contrary to what we had hypothesized, we observed very little changes in calling proportions or accuracy as a result of changes to sample size.

*Comparison of singlet and doublet calling accuracy*

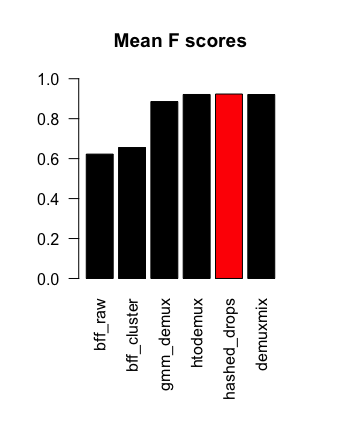
**Fig. 3b** shows the overall singlet assignment errors. As staining quality decreased, we observed that the most common error made by all algorithms (except GMM-Demux, which we previously noted had a propensity to call large proportions of doublets at low staining qualities) was to assign no call to the singlet. We also observe, strikingly, that HashedDrops, the most simplistic algorithm, rarely assigns singlets as false doublets, although that may be due to manually setting the cutoff for doublets to 0.5. Finally, we can see in the case of HTODemux that at high staining qualities, decreasing the sample size affects calling accuracy. This is the only algorithm and the only condition in our benchmarking experiment which demonstrated this trend.

We also benchmarked each algorithm’s ability to accurately assign doublets. Specifically, it is possible for doublets to contain two or more cells with either different or similar hashtags. Intuitively, we can understand that it is much more trivial for algorithms to detect doublets with different hashtags (especially at higher staining qualities) than to detect doublets with the same hashtag, which may appear to be merely singlets with particularly high hashtag counts. We can indeed see in **Fig. 3c** that this is the case. While most algorithms perform admirably in assigning doublets with different hashtags, almost all algorithms perform far worse in assigning doublets with the same hashtag, with the exception of HTODemux, which we note does an excellent job in identifying doublets in both cases. This suggests that HTODemux is a tool with particularly high sensitivity (albeit relatively low specificity) for doublets.

We note here the comparatively poor performance by both demuxmix and HashedDrops in calling doublets accurately. In the case of demuxmix, unlike BFF raw, BFF cluster, GMM-Demux, and HTODemux, demuxmix can actually assign more than two barcodes to a multiplet. As such, an “uncertain” call in these cases can simply mean that demuxmix is unsure of exactly how many barcodes are in this multiplet. HashedDrops, as the simplest multiplet caller based on log-fold change thresholds, displays the worst ability to assign doublets in either case. Because of its sensitivity to parameter optimization, it is likely that doublet assignment accuracy can be improved in the future by modifying manual thresholds for doublets.

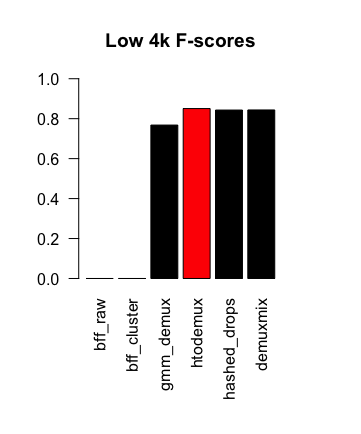
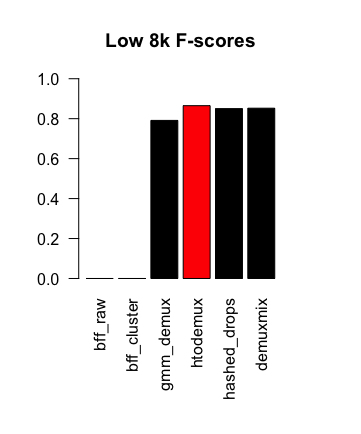
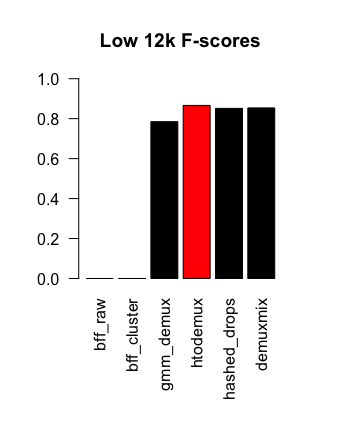
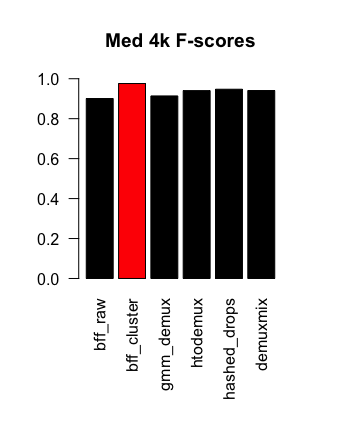
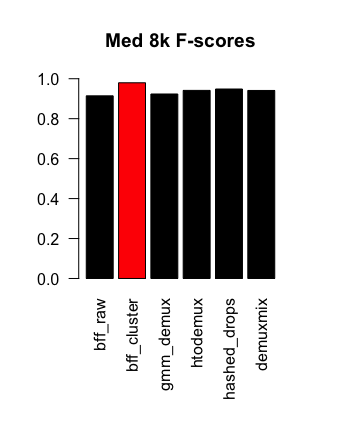
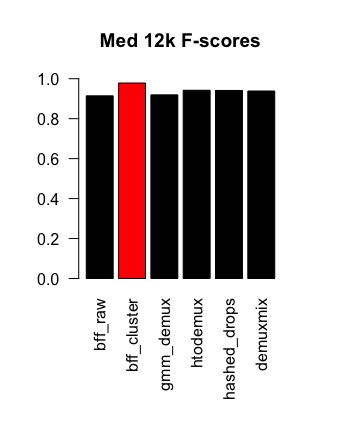
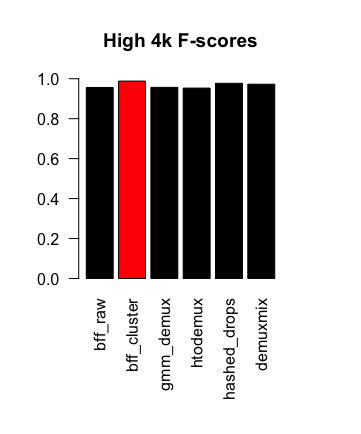
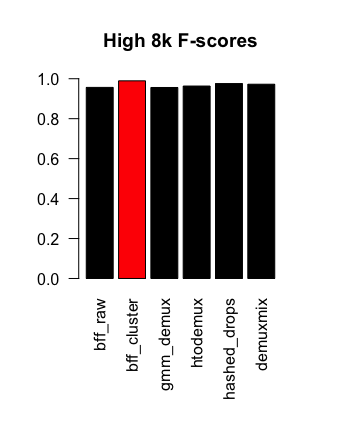
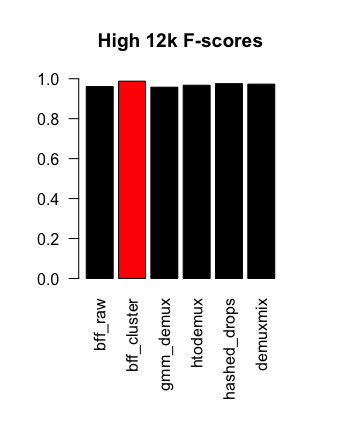
**Computation of overall F-score**

In order to conceptualize an overall score which can summarize the performance of each tool in different conditions, I turned to the F-score, which represents the harmonic mean of precision and recall, with higher F-scores representing better performance. As we can see in **Fig. 4a**, at high and medium staining qualities, BFF\_cluster has the best performance, but almost all algorithms

**Figure 4. F-scores for each algorithm under various experimental conditions.** a) shows F-scores for each individual condition, with the algorithm with the highest F-score for that category colored in red. b) shows the mean F-scores for each algorithm across all conditions. Highest F-score colored in red.

**B.**

**A.**



perform well in these more ideal cases. In cases of low staining quality, however, failure of both BFF algorithms gives them an F-score of 0. In comparison, HTODemux has the best performance. When we calculate the mean F-score across all conditions for each algorithm (**Fig. 4b**), we find that very surprisingly, HashedDrops displays the best overall performance despite the fact that it never performed the best in any experimental conditions. However, we can also see that both HTODemux and demuxmix, high-performing algorithms in their own right, have comparable F-scores.

**Limitations and Alternative Approaches**

There are quite a few limitations to this project, some of which are nontrivial and were only discovered as I proceeded with the analysis. First, the parameters of certain algorithms may not have been ideal. While Howitt et al.29 reported that changing default parameters of most algorithms did not result in appreciable changes to calling accuracy, my experience while attempting to implement calling algorithms through cellhashR was different. For instance, I initially found that HTODemux called an incredibly large number of doublets (30-50%). Eventually, I discovered that the default value for positive.quantile in cellhashR was 0.95, while the default value for positive.quantile in the Seurat vignette available from the Satija lab website directly was 0.99. Setting the value to 0.99 produced the output reported here. It is entirely possible that there are other parameters which are not coded in accurately in cellhashR (or have since been updated) that could improve the performance of GMM-Demux.

Secondly, demuxmix is intended to function with a gene expression matrix as a covariate to further refine accuracy of calls. Unlike DemuxEM, however, it can be run in a naïve fashion without this matrix – at the expense of accuracy. In single-cell sequencing experiments, you will almost always have the gene expression matrix available, which means that demuxmix may actually be the best overall performing tool in real-life scenarios.

Thirdly, the performance of the tool most affected by parameter optimization – HashedDrops – could have been improved further. I used the value which Howitt et al. reported gave them the best performance, but it is entirely possible that a different value would have been better suited to the simulated dataset. Unfortunately, I ran out of time to implement this.

Finally, in comparing my simulated data to real-life hashtag data that I and others in my lab have generated in the past, I have noticed that while the negative binomial distribution method generates count distributions with roughly the same shape, the actual count values tend to be much larger. It is entirely possible that some tools may have performed with better accuracy if the count magnitudes were amplified to better match real-life data.

**References**

1. Baysoy A, Bai Z, Satija R, Fan R. The technological landscape and applications of single-cell multi-omics. *Nat Rev Mol Cell Biol*. 2023;24(10):695-713. doi:10.1038/s41580-023-00615-w

2. Free T. Single-cell Sequencing: The Technological Revolution Behind a New Wave of Multiomic Studies in Basic and Cancer Research. *BioTechniques*. 2021;71(5):539-542. doi:10.2144/btn-2021-0088

3. Aldridge S, Teichmann SA. Single cell transcriptomics comes of age. *Nat Commun*. 2020;11(1):4307. doi:10.1038/s41467-020-18158-5

4. Ziegenhain C, Vieth B, Parekh S, et al. Comparative Analysis of Single-Cell RNA Sequencing Methods. *Mol Cell*. 2017;65(4):631-643.e4. doi:10.1016/j.molcel.2017.01.023

5. Stoeckius M, Hafemeister C, Stephenson W, et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat Methods*. 2017;14(9):865-868. doi:10.1038/nmeth.4380

6. Grandi FC, Modi H, Kampman L, Corces MR. Chromatin accessibility profiling by ATAC-seq. *Nat Protoc*. 2022;17(6):1518-1552. doi:10.1038/s41596-022-00692-9

7. Mimitou EP, Lareau CA, Chen KY, et al. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nat Biotechnol*. 2021;39(10):1246-1258. doi:10.1038/s41587-021-00927-2

8. Lareau CA, Ludwig LS, Muus C, et al. Massively parallel single-cell mitochondrial DNA genotyping and chromatin profiling. *Nat Biotechnol*. 2021;39(4):451-461. doi:10.1038/s41587-020-0645-6

9. Swanson E, Lord C, Reading J, et al. Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using TEA-seq. *eLife*. 10:e63632. doi:10.7554/eLife.63632

10. Muftuoglu M, Li L, Liang S, et al. Extended live-cell barcoding approach for multiplexed mass cytometry. *Sci Rep*. 2021;11(1):12388. doi:10.1038/s41598-021-91816-w

11. Krutzik PO, Clutter MR, Trejo A, Nolan GP. Fluorescent cell barcoding for multiplex flow cytometry. *Curr Protoc Cytom*. 2011;Chapter 6:6.31.1-6.31.15. doi:10.1002/0471142956.cy0631s55

12. Krishhan VV, Khan IH, Luciw PA. Multiplexed microbead immunoassays by flow cytometry for molecular profiling: Basic concepts and proteomics applications. *Crit Rev Biotechnol*. 2009;29(1):29-43. doi:10.1080/07388550802688847

13. Pappireddi N, Martin L, Wühr M. A Review on Quantitative Multiplexed Proteomics. *Chembiochem Eur J Chem Biol*. 2019;20(10):1210-1224. doi:10.1002/cbic.201800650

14. Chen J, Schwarz E. Opportunities and Challenges of Multiplex Assays: A Machine Learning Perspective. In: Guest PC, ed. *Multiplex Biomarker Techniques: Methods and Applications*. Springer New York; 2017:115-122. doi:10.1007/978-1-4939-6730-8\_7

15. Auer PL, Doerge RW. Statistical design and analysis of RNA sequencing data. *Genetics*. 2010;185(2):405-416. doi:10.1534/genetics.110.114983

16. Zhang Y, Xu S, Wen Z, et al. Sample-multiplexing approaches for single-cell sequencing. *Cell Mol Life Sci*. 2022;79(8):466. doi:10.1007/s00018-022-04482-0

17. Brown DV, Anttila CJA, Ling L, et al. A risk-reward examination of sample multiplexing reagents for single cell RNA-Seq. *Genomics*. 2024;116(2):110793. doi:10.1016/j.ygeno.2024.110793

18. Stoeckius M, Zheng S, Houck-Loomis B, et al. Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome Biol*. 2018;19(1):224. doi:10.1186/s13059-018-1603-1

19. McGinnis CS, Patterson DM, Winkler J, et al. MULTI-seq: sample multiplexing for single-cell RNA sequencing using lipid-tagged indices. *Nat Methods*. 2019;16(7):619-626. doi:10.1038/s41592-019-0433-8

20. Huang Y, McCarthy DJ, Stegle O. Vireo: Bayesian demultiplexing of pooled single-cell RNA-seq data without genotype reference. *Genome Biol*. 2019;20(1):273. doi:10.1186/s13059-019-1865-2

21. Kang HM, Subramaniam M, Targ S, et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat Biotechnol*. 2018;36(1):89-94. doi:10.1038/nbt.4042

22. Boggy GJ, McElfresh GW, Mahyari E, et al. BFF and cellhashR: analysis tools for accurate demultiplexing of cell hashing data. *Bioinforma Oxf Engl*. 2022;38(10):2791-2801. doi:10.1093/bioinformatics/btac213

23. Hao Y, Hao S, Andersen-Nissen E, et al. Integrated analysis of multimodal single-cell data. *Cell*. 2021;184(13):3573-3587.e29. doi:10.1016/j.cell.2021.04.048

24. Lun ATL, Riesenfeld S, Andrews T, Dao TP, Gomes T, Marioni JC. EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol*. 2019;20(1):63. doi:10.1186/s13059-019-1662-y

25. Xin H, Lian Q, Jiang Y, et al. GMM-Demux: sample demultiplexing, multiplet detection, experiment planning, and novel cell-type verification in single cell sequencing. *Genome Biol*. 2020;21(1):188. doi:10.1186/s13059-020-02084-2

26. Klein HU. demuxmix: Demultiplexing oligonucleotide-barcoded single-cell RNA sequencing data with regression mixture models. *BioRxiv Prepr Serv Biol*. Published online January 29, 2023:2023.01.27.525961. doi:10.1101/2023.01.27.525961

27. Gaublomme JT, Li B, McCabe C, et al. Nuclei multiplexing with barcoded antibodies for single-nucleus genomics. *Nat Commun*. 2019;10(1):2907. doi:10.1038/s41467-019-10756-2

28. Bernstein NJ, Fong NL, Lam I, Roy MA, Hendrickson DG, Kelley DR. Solo: Doublet Identification in Single-Cell RNA-Seq via Semi-Supervised Deep Learning. *Cell Syst*. 2020;11(1):95-101.e5. doi:10.1016/j.cels.2020.05.010

29. Howitt G, Feng Y, Tobar L, et al. Benchmarking single-cell hashtag oligo demultiplexing methods. *NAR Genomics Bioinforma*. 2023;5(4):lqad086. doi:10.1093/nargab/lqad086

30. Plan EL. Modeling and Simulation of Count Data. *CPT Pharmacomet Syst Pharmacol*. 2014;3(8):e129. doi:10.1038/psp.2014.27