

Donor Assignment Computational Cookbook

version 1.0.0 (3/25/24)

James Nemesh

Steve McCarroll's lab, Harvard Medical School

The purpose of this cookbook is to help you run the dropseq tools responsible for determining the donor of origin of cells, and detect doublet cell barcodes where cells from two donors have been co-encapsulated.

This software supports both data processed by DropSeq software tools, as well as CellRanger and STARSolo tools. With minor modifications many other software pipelines may be accommodated.

[Preprocessing data from CellRanger or StarSolo](#)

[TagReadWithGeneFunction](#)

[Donor Assignment](#)

[Example Invocation](#)

[10x scRNASeq](#)

[10x scATAC](#)

[Additional Program Options](#)

[Output File Details](#)

[Doublet detection](#)

[Example Invocation](#)

[10x scRNASeq](#)

[10x scATAC](#)

[Mitigating the effects of cell free RNA](#)

[MAX_ERROR_RATE](#)

[Per-cell correction with Cellbender](#)

[Output File Details](#)

[Math](#)

[Analysis and QC](#)

[QC Text Reports](#)

[donor_cell_map](#)

[dropulation_summary_stats](#)

[QC Plots](#)

[Nuclei \(low loading\)](#)

[Sample Swap Example](#)

[iPSC cells \(large pool\)](#)

[Nuclei \(high loading\)](#)

[Experimental diffuse contamination rescue](#)

Preprocessing data from CellRanger or StarSolo

CellRanger and [StarSolo](#) emit reads that are very similar to, but not quite identical to DropSeq alignment's standard outputs. The two main differences are that DropSeq tools embed gene annotation information directly on reads as bam tags for scRNASeq data, and the cell / molecular barcode tag names are different. To process this data, we'll apply a pre-processing step to add in gene annotations, then during processing a few additional arguments will be added to use the appropriate tags.

TagReadWithGeneFunction

TagReadWithGeneFunction (part of the DropSeq software distribution) directly adds tags to BAM reads that indicate how each read interacts with gene models. This is useful to later introspect reads and better understand why they were or were not considered for a particular analysis. This tagging process and how the tags are interpreted are detailed in depth in the [Alignment Cookbook](#) if you're interested in further details. In order to make scRNASeq data from other platforms compatible with DropSeq tools, the first step is to add those functional annotations. Given a GTF file, this process is straightforward - the input BAM has the tags added from a GTF or refFlat file, and a new BAM is emitted.

Unset

```
/path/to/dropseq_install/TagReadWithGeneFunction
-ANNOTATIONS_FILE /path/to/organism.gtf
-INPUT /path/to/cellranger_output/outs/possorted_genome_bam.bam
-OUTPUT /path/to/cellranger_output/outs/possorted_genome_bam_tagged.bam
```

This new tagged BAM will be used for all downstream processing. If you're running the standard DropSeq pipeline, there's no need to repeat this step as you performed it before running DigitalExpression.

This step can be skipped if you are using ATAC data, which does not rely on gene annotations for quantification.

CellRanger and STARSolo use different cell barcode (CB) and molecular barcode (UB) tags than the DropSeq Pipeline defaults. When using donor assignment tools with data from these pipelines, it's important to remember to explicitly set these tags.

For reference, the CellRanger documentation for BAM Tags:

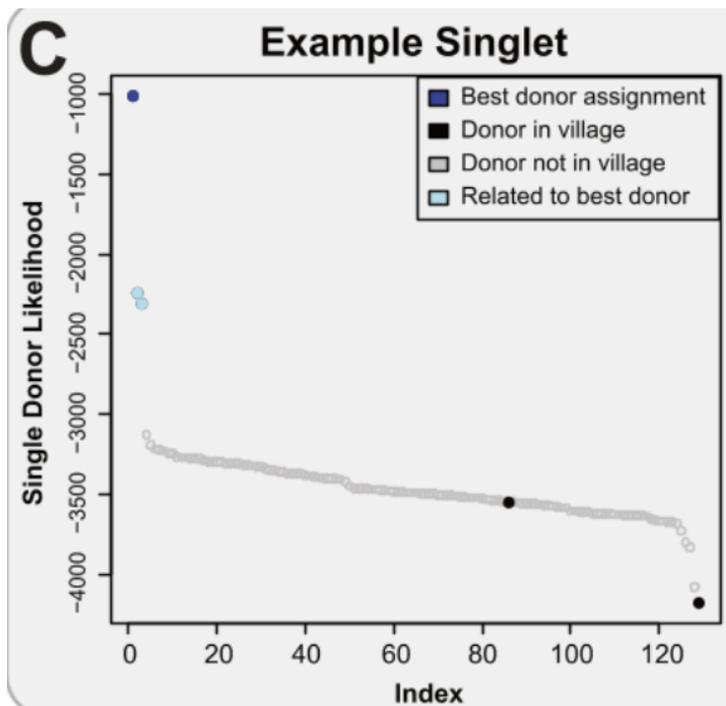
<https://support.10xgenomics.com/single-cell-multiome-atac-gex/software/pipelines/latest/output/bam-gex>

<https://support.10xgenomics.com/single-cell-multiome-atac-gex/software/pipelines/latest/output/bam-atac>

Donor Assignment

Donor Assignment evaluates each cell of an RNASeq data set independently, leveraging prior knowledge of donor genotypes to construct a likelihood score for each donor. The donor that best explains the observed transcribed SNPs in the scRNASeq data is selected as the best candidate. This likelihood score is then compared to all other donor likelihood scores to determine the confidence of the assignment.

In the donor assignment file, there is one column emitted for each potential donor's likelihood. In the example below, one row of the donor assignment output is plotted. The donor with the maximum score is selected as the donor most likely to have generated the observed data. While donors related to the main donor share haplotypes, related donors are still less likely to explain the observed data. This remains true even for parent/child relationships, where the child shares the haplotypes of both parents, but in different combinations that can be distinguished without explicitly leveraging these differences or prior knowledge of these relationships.



Example Invocation

Example invocations for 10x CellRanger (StarSolo should be very similar)

10x scRNASeq

Unset

```
/path/to/dropseq_install/AssignCellsToSamples -m 24g
-INPUT_BAM /path/to/cellranger_output/outs/possorted_genome_bam_tagged.bam
-CELL_BC_FILE
/path/to/cellranger_output/outs/filtered_feature_bc_matrix/barcodes.tsv.gz
-VCF /path/to/project_vcf/project.vcf.bgz
-OUTPUT /path/to/output/possorted_genome_bam.donor_assignments.txt
-VCF_OUTPUT /path/to/output/possorted_genome_bam.donor_assignments.vcf.gz
-CELL_BARCODE_TAG CB -MOLECULAR_BARCODE_TAG UB
-LOCUS_FUNCTION_LIST INTRONIC
-IGNORED_CHROMOSOMES null -IGNORED_CHROMOSOMES chrX
-IGNORED_CHROMOSOMES chrY -IGNORED_CHROMOSOMES=chrM
```

The inputs are a set of scRNASeq aligned reads in a BAM file, a list of cell barcodes that are likely cells (in this case as determined by 10x software) and a VCF file containing genotypes expected to be in the experiment. Providing a cell barcode file limits the scope of which cell barcodes in the BAM are tested. Memory and computation scales linearly with the number of donors in the VCF and number of cell barcodes, so it is useful to at least somewhat limit the cell barcodes included in analysis to those you would want to further analyze downstream.

The VCF file can be a superset of the donors expected in the experiment. Our best practice is to include all donors that are currently being used in the lab, as donor assignment can then detect sample swaps - and unfortunately this happens more often than you might expect. The VCF can be in either VCF format (which is a text based format) or in BCF format (binary). The advantage of the BCF format is that it parses much more quickly than VCF (~5x faster) which significantly decreases the run time of donor assignment, as VCF parsing can be quite expensive for large numbers of donors. To create a binary file, please use Picard's [VcfFormatConverter](#).

Other arguments include the list of locus functions to evaluate and the chromosomes to not be included in the analysis. The locus function defines which RNASeq reads are included in the analysis - in old versions of gene expression software only coding regions were used, but more recent versions include intronic UMIs in analysis. Nuclei sequencing data in particular has a large number of intronic UMIs, which greatly increases the power of donor assignment. We suggest you always leave this on for scRNASeq unless you have strong priors. The contigs that are excluded in the above list are the sex and MT contigs, which are all a bit tricky to work with and require different models than the autosomes. Since donor assignment tends to be well powered in a typical sequencing experiment we have not implemented those additional models, and suggest you always ignore those contigs.

Emitted are the main OUTPUT file containing the information about each cell in the input cell barcode list, as well as a VCF containing the subset of SNPs that passed QC and were transcribed in the sequence data. This new much smaller VCF can then be used downstream

in doublet detection, which significantly reduces the computational time and memory requirements.

An argument you might be tempted to use at this point in the analysis is **SAMPLE_FILE**. This restricts the VCF to the subset of donors included in the file. However, this reduces your ability to detect sample swaps in experiments. Donor assignment should select the correct subset of donors regardless of how many potential donors are in the VCF. Limiting the VCF subset of donors that are in the experiment will force the algorithm to select one of those donors as the maximum likelihood answer, even if that answer is of poor quality.

We suggest not using this argument unless you have a specific use case - for example, you might have a VCF with 1000's of donors and you want to restrict to all donors used in your lab in the last 2 years (500). Or, you might want to simulate how a donor assignment responds if a donor is present in the pool, but you do not have reference information for the donor - in this case, make a donor list that includes all but one donor from the VCF. Ultimately, you should not need to give the program hints as to which donors are in your pool.

10x scATAC

Unset

```
/path/to/dropseq_install/AssignCellsToSamples -m 24g
-INPUT_BAM /path/to/cellranger_output/outs/atac_possorted_bam.bam
-CELL_BC_FILE
/path/to/cellranger_output/outs/filtered_feature_bc_matrix/barcodes.tsv.gz
-VCF /path/to/project_vcf/project.vcf.bgz
-OUTPUT /path/to/output/atac_possorted_bam.donor_assignments.txt
-VCF_OUTPUT /path/to/output/atac_possorted_bam.donor_assignments.vcf.gz
-CELL_BARCODE_TAG CB -DNA_MODE true
-IGNORED_CHROMOSOMES null -IGNORED_CHROMOSOMES chrX
-IGNORED_CHROMOSOMES chrY -IGNORED_CHROMOSOMES chrM
```

The scATAC invocation is quite similar to scRNASeq. The main difference is due to differences in the chemistry of the experiment - ATACSeq does not have UMIs, and does not measure expression, so the UMI and LOCUS_FUNCTION arguments are dropped. Instead, the DNA_MODE is enabled, which instead uses pcr duplicate flags instead of UMIs for read deduplication.

In our somewhat limited testing of 10x multiome data, both scATAC and scRNASeq cells are assigned to donors with similar error rates. What is most important with multiome data sets is that both modalities do not have equal amounts of data for each cell, so it may be useful to use the donor assignment for the data set with the highest number of reads/UMIs if you wish to maximize the number of cells you can assign.

Additional Program Options

Unset

```
-VALIDATION_STRINGENCY SILENT  
-TMP_DIR /path/to/TMP  
-BAM_OUTPUT /path/to/output/possorted_genome_bam.donor_assignments.bam
```

Setting the validation stringency to silent will disable the HTSJDK sequencing read validation, which can marginally improve run time. If your data has been generated by some consistent process, turn this on to save a few cycles.

The TMP_DIR option allows you to manually set a temp directory for intermediate file generation. This directory is used as the input BAM is sorted as part of processing. In certain cases, users have reported errors where java is unable to correctly determine their temp directory, and strange looking errors occur. For example:

Unset

```
Exception in thread "main"  
htsjdk.samtools.util.RuntimeIOException:  
java.nio.file.NoSuchFileException:  
/scratch/tmp/sortingcollection.10853389501593454727.tmp
```

Setting the TMP_DIR explicitly should fix this issue.

Setting the BAM_OUTPUT parameter will cause AssignCellsToSamples to emit a BAM file containing all of the informative reads for the analysis. This can then be used as the input BAM for doublet detection, which will improve runtime performance of DetectDoublets.

Output File Details

The key output from AssignCellsToSamples is OUTPUT, which contains one row for each cell barcode evaluated where the cell has at least one informative UMI.

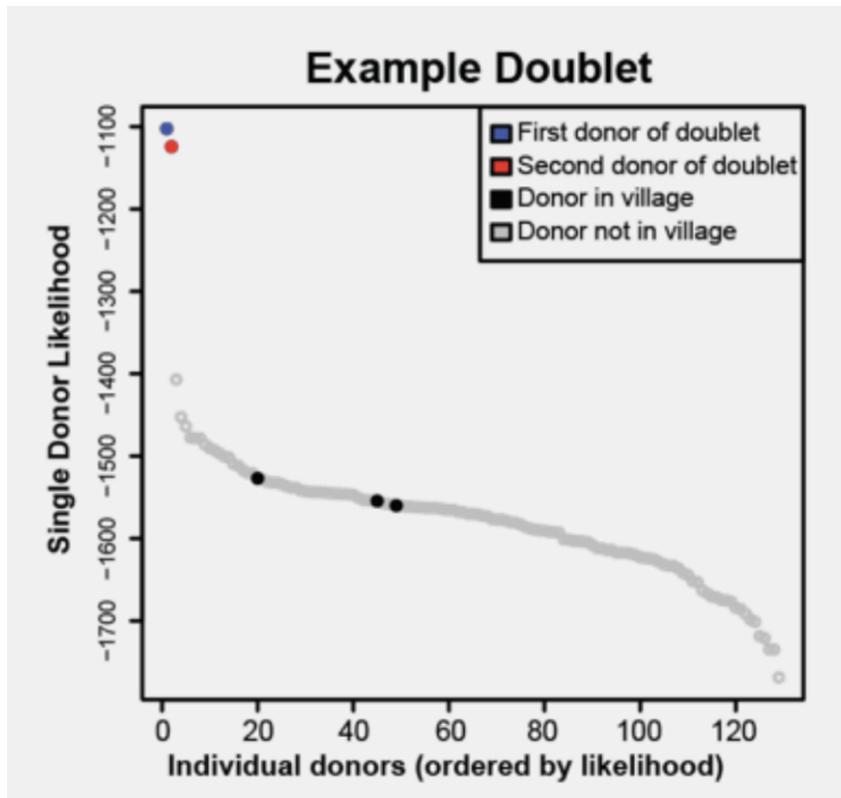
- **cell** - The cell barcode analyzed. Each cell in the input CELL_BC_FILE file will appear in the output, unless the cell has no informative data for analysis

- **num_snps** - The number of informative SNPs for this cell. Informative SNPs pass both VCF backbone filters, and are transcribed in the sequencing data of the cell.
- **num_umis** - The number of informative UMIs for this cell. These are distinct allelic observations (which may be supported by one or more reads), and their total count should always be greater than or equal to the number of snps.
- **ratio** - **The likelihood ratio of best / second best donor.**
- **pvalue** - **The likelihood of the best donor divided by the sum of likelihood ratios of all donors. This is reported as $1 - (\text{best}/\text{all})$, such that smaller values indicate higher confidence in the assignment. Due to numeric issues, the most confident value is $\sim 5e-324$.**
- **FDR_pvalue** - The pvalue corrected for the number of cells tested via benjamini hochberg correction. This value is used to select cells that are confidently assigned to a single donor, with a threshold value of 0.05.
- **bestLikelihood** - The maximum donor assignment likelihood for a single cell.
- **bestSample** - The donor label with the best likelihood.
- **median_likelihood** - The average likelihood for all donors
- **population_average_likelihood** - For each UMI, we calculate the likelihood of observing a uniform mixture of all donors. This is used as the likelihood when a donor does not have a high quality genotype at the site. This column contains that value aggregated across all sites considered.
- Additional columns represent individual donor likelihood scores.

Doublet detection

At this point, each cell barcode has been assigned to its most likely donor. However, a cell barcode might contain more than one physical cell. If those cells arose from two different donors, the transcribed alleles can be interpreted as a mixture of alleles from both of those individuals.

What do the donor assignment likelihoods look like for a doublet? Below is plotted the likelihoods for all donors for a doublet. The “best” donor the cell is assigned to has the highest likelihood, but there’s a second donor that also has quite a high likelihood, compared to the rest of the distribution. In this situation, both donors do a good job of explaining the data on their own, but we can also test a model where both donors contribute transcribed alleles, and the combination of the two donors better explains the data than either of the two donors alone. When a mixture of two donors is more likely to explain the data than a single donor, we label that cell a doublet.



Doublet detection works as a series of hierarchical models, one per donor pair. The initial donor assignment selects the donor most likely to explain the observed UMIs, and this becomes the first donor of the pair. The second donor of the pair comes from any donor that is observed at least once in the initial donor assignment.

For each pair of donors a joint likelihood of the two donors is calculated along with the relative proportion of the two donors in the cell. The likelihood is defined as the summed weighted average of the two donors' likelihoods at each UMI. The relative proportion of each donor in the cell is unknown, but can be determined by maximizing the likelihood of the data given a particular weight.

To determine this proportion, we optimize the mixture component of donor 1 to donor 2 to maximize the likelihood of that donor pair. The mixture score is the fraction of the data that arises from the first donor of the pair and is bounded to $[0.8-0.2]$. If the mixture score is unbounded, sequencing errors, ambient RNA, and genotyping errors will almost always generate mixtures of two donors that are very close to one, with a higher likelihood than the single donor likelihood, resulting in most cells being classified as doublets.

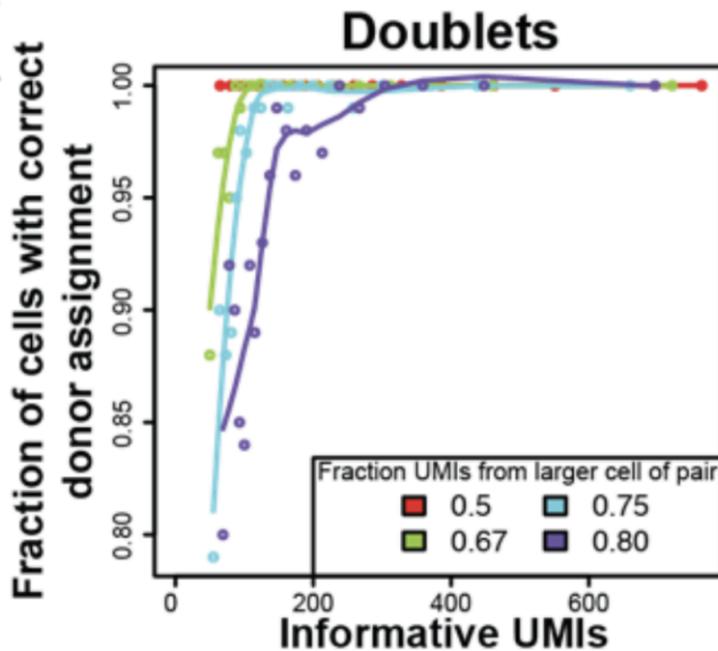
Now we have a likelihood model for each possible pair of the best donor plus another donor expected to be in the experiment. All models are adjusted such that they have the same number of observations so their likelihoods are comparable. To select the donor pair that best explains the data, we first calculate the maximum likelihood each donor pair by selecting the

maximum likelihood of the optimal mixture, the likelihood of the pair with a mixture of 1 (all data arises from donor 1) and the likelihood of the pair with a mixture of 0 (all data arises from donor 2.) The donor pair with maximum likelihood is then selected as the best pair.

With just a single donor pair remaining, we can finally test if one donor or two best explains the data. To classify the pair as a singlet or doublet, we calculate the probability of the data being a doublet as the doublet likelihood divided by the sum of the doublet likelihood and mixture=1,0 likelihoods. We classify cells as doublets if their probability is greater ≥ 0.9 . The vast majority of doublet probabilities are bimodally distributed at approximately 1 and 0.

To cover strange edge cases where a donor's cells might be in the pool but are extremely rare and small and are never seen as the most common donor of a pair, a donor list can be supplied that contains all donors expected. Even if a donor was not observed as the primary donor for a cell, this list will add those donors as possible pairs for testing. This should catch that odd edge case (that we have yet to see in a real experiment, despite having run 100's of experiments.)

One question that frequently arises is the amount of UMIs needed to confidently distinguish between singlets and doublets. When Doublet detection is underpowered, it may be unable to distinguish between the two models and erroneously flag cells as doublets. From [in-silico mixing experiments](#), a reasonable estimate for the number of informative UMIs needed is approximately 100-200. Doublets where both cells have relatively equal sizes are easier to detect with a given number of UMIs, while unequal sized doublets where one donor's cell is much larger than the other (0.8) require more UMIs.



If you think the number of doublets detected is much higher than you expected, check to make sure your cells have captured a sufficient number of UMIs. This can be increased by sequencing deeper, or by using a SNP backbone (VCF file) with more SNPs. We have found that SNP array data + high quality imputation (filtered to $R^2 \geq 0.9$ sites) works well, and is significantly better than SNP array data without imputation.

Example Invocation

10x scRNASeq

```
Unset
/path/to/dropseq_install/DetectDoublets -m 16g
-INPUT_BAM /path/to/output/possorted_genome_bam.donor_assignments.bam
-CELL_BC_FILE
/path/to/cellranger_output/outs/filtered_feature_bc_matrix/barcodes.tsv.gz
-VCF /path/to/output/possorted_genome_bam.donor_assignments.vcf.gz
-SINGLE_DONOR_LIKELIHOOD_FILE
/path/to/output/possorted_genome_bam.donor_assignments.txt
-SAMPLE_FILE /path/to/donor_list.txt
-OUTPUT /path/to/output/possorted_genome_bam.doublets.txt
-CELL_BARCODE_TAG CB - MOLECULAR_BARCODE_TAG UB -LOCUS_FUNCTION_LIST INTRONIC
-IGNORED_CHROMOSOMES null -IGNORED_CHROMOSOMES chrX
-IGNORED_CHROMOSOMES chrY -IGNORED_CHROMOSOMES chrM
-MAX_ERROR_RATE 0.05
```

The outputs from AssignCellsToSamples become the inputs to DetectDoublets. In the above example, we've taken advantage of AssignCellsToSamples ability to emit both more minimal BAM and a smaller VCF to speed up computation time and reduce memory usage. As with AssignCellsToSamples, we keep the same settings for the functional annotations considered, and filter the same contigs.

10x scATAC

```
Unset
/path/to/dropseq_install/DetectDoublets -m 24g
-INPUT_BAM /path/to/output/atac_possorted_bam.donor_assignments.bam
-CELL_BC_FILE
/path/to/cellranger_output/outs/filtered_feature_bc_matrix/barcodes.tsv.gz
-VCF /path/to/output/atac_possorted_bam.donor_assignments.vcf.gz
```

```
-SINGLE_DONOR_LIKELIHOOD_FILE
/path/to/output/atac_possorted_bam.donor_assignments.txt
-SAMPLE_FILE /path/to/donor_list.txt
-OUTPUT /path/to/output/atac_possorted_bam.doublets.txt
-CELL_BARCODE_TAG CB -DNA_MODE true
-IGNORED_CHROMOSOMES null -IGNORED_CHROMOSOMES chrX
-IGNORED_CHROMOSOMES chrY -IGNORED_CHROMOSOMES chrM
-MAX_ERROR_RATE 0.05
```

Using scATACSeq has the same changes to parameters for DNA data as AssignCellsToSamples - No UMI barcodes, and DNA_MODE true.

Mitigating the effects of cell free RNA

MAX_ERROR_RATE

One new parameter is MAX_ERROR_RATE. The goal of this parameter is to put a cap on the maximum penalty to the likelihood calculation when a confident observation of an allele in the sequencing data conflicts with the expected allele. Alleles that are observed can come both from the donor(s) that are captured by the cell, as well as other error modes like cell free RNA. Without this parameter being set, one or more of the pairwise models may observe an error allele as coming from the second donor of the pool. With enough cell free RNA, this can result in false positive doublet calls. In practice, the value of 0.05 mitigates these errors enough to keep the false positive rate low.

Per-cell correction with Cellbender

If you don't like setting an arbitrary threshold, we have an alternate set of parameters that model contamination at a per-cell level and perform as well or better than the simple threshold. Instead of supplying a global threshold to all cells, you can instead supply two parameters at a per-cell level.

The first parameter is the amount of cell free RNA captured by each cell, which can be generated by Cellbender [remove-background](#). Cellbender models how much cell free RNA and UMI chimeras are present in each cell, and generates a new expression matrix after those error counts are removed. To supply this parameter to DetectDoublets, calculate the total number of UMIs captured by each cell before running Cellbender, and then again after CellBender is run. The contamination estimate is the fraction of UMIs that were removed. For example, if a cell had 500 UMIs before CellBender was run, and 400 after, 100 of those UMIs would have come from cell free RNA, and the contamination fraction would be $(500-400)/500=0.2$. The input file is tab separated, and contains two columns: cell_barcode and frac_contamination.

The second parameter is the allele frequency at each site. This can be calculated via another program we distribute, GatherDigitalAlleleCounts. The inputs required for this program are inputs you've already used - the VCF file, the BAM, and the donor list.

Here's an example invocation for 10x RNASeq data:

```
Unset
/path/to/dropseq_install/GatherDigitalAlleleCounts -m 16g
-INPUT /path/to/cellranger_output/outs/possorted_genome_bam_tagged.bam
-CELL_BC_FILE
/path/to/cellranger_output/outs/filtered_feature_bc_matrix/barcodes.tsv.gz
-VCF /path/to/project_vcf/project.vcf.bgz
-SAMPLE_FILE /path/to/donor_list.txt
-ALLELE_FREQUENCY_OUTPUT /path/to/output/allele_freq.txt
-LOCUS_FUNCTION_LIST INTRONIC
-IGNORED_CHROMOSOMES null -IGNORED_CHROMOSOMES chrX
-IGNORED_CHROMOSOMES chrY -IGNORED_CHROMOSOMES chrM
-SINGLE_VARIANT_READS false -MULTI_GENES_PER_READ false
```

GatherDigitalAlleleCounts is a multi-function program that generates allelic pileups with both UMI and read counts at requested SNP sites and can be used for both scRNASeq and DNA data. An example of the key columns in the output:

chromosome	position	ref_allele	alt_allele	maf_umi
chr1	730177	G	A	0.000
chr1	791101	T	G	1.000
chr1	796652	A	C	0.167
chr1	798969	T	C	0.068
chr1	802843	T	C	0.111

With these two sets of features, Doublet detection can calculate the likelihood of an observed allele being drawn from the cell, vs being drawn from the cell free RNA that was co-captured, and modify the doublet detection likelihoods appropriately without setting an arbitrary threshold.

To use these features in DetectDoublets remove the MAX_ERROR_RATE argument and add:

```
Unset
-ALLELE_FREQUENCY_ESTIMATE_FILE /path/to/output/allele_freq.txt
-CELL_CONTAMINATION_ESTIMATE_FILE /path/to/output/cell_contamination.txt
```

This adaptation for doublet detection has only been implemented and tested for scRNASeq data. scATAC data in our limited testing seems to be a bit more robust to errors, and a `MAX_ERROR_RATE=0.05` produces reasonable results.

Output File Details

In this output, each line contains the pair of donors that best explains the data for an input cell. It's important to remember that even though the best pair is emitted, one donor of that pair may be a far better explanation of the observed alleles than a mixture of the two donors. The restriction on the donor mixture to `[0.8-0.2]` controls the minimum difference between a droplet that captured a single donor and a mixture of two donor cells of unequal sizes.

- **cell** - The cell barcode analyzed. Each cell in the input `CELL_BC_FILE` file will appear in the output, unless the cell has no informative data for analysis
- **sampleOneMixtureRatio** - The estimated proportion of alleles that are contributed by the first donor of the donor pair. This value is by default restricted to values between 0.8 and 0.2.
- **sampleOne** - The first donor of the pair.
- **sampleOneLikelihood** - The likelihood of the first donor of the pair. This is the likelihood of the data with the mixture parameter set to 1.
- **sampleTwo** - The second donor of the pair.
- **sampleTwoLikelihood** - The likelihood of the second donor of the pair. This is the likelihood of the data with the mixture parameter set to 0.
- **mixedSample** - The concatenated label for **sampleOne:sampleTwo**.
- **mixedSampleLikelihood** - The likelihood of the data given a mixture of the two donors at the mixture ratio.
- **num_paired_snps** - The number of SNPs that were transcribed and genotype in both donors of the pair.
- **num_inform_snps** - The number of paired snps that had different genotypes for the two donors.
- **num_umi** - The number of UMIs captured by paired SNPs.
- **num_inform_umis** - The number of UMIs captured by informative SNPs
- **lr_test_stat** - The likelihood ratio of the doublet likelihood / max(sample one likelihood, sample two likelihood).
- **sampleOneWrongAlleleCount** - The number of transcribed alleles that could not have been generated by the sampleOne. For example, if sampleOne had a genotype of A/A and a non-A allele was observed, this count would be incremented once per UMI observed.
- **sampleTwoWrongAlleleCount** - The number of transcribed alleles that could not have been generated by sampleTwo.
- **bestLikelihood** - The maximum of the doublet and single donor likelihoods.
- **bestSample** - The sample with the best likelihood - this may sampleOne, sampleTwo, or mixedSample.

- **doublet_pval** - The probability that the cell is a doublet in the range [0-1].
- **best_pair_pvalue** - The probability that one pair of donors best explains the data, in the range of [0-1]. The initial model tests all possible combinations sampleOne with all other observed donors as sampleTwo. This tests that one of those pairs explains the observed data much better than any of the other pairs by comparing the best likelihood / sum (all likelihoods). In cases where a single pair of donors best explains the data ($\text{best_pair} \geq 0.9$), we label that a confident doublet. In cases where many pairs of donors explain the data with similar likelihoods, that is more likely the result of cell free RNA contributing many alleles to the cell barcode.

Math

If you'd like to dig deeper into the likelihood calculations and how missing genotype data is handled for both donor assignment and doublet detection, you can dig into the methods section of Wells MF, et al [Natural variation in gene expression and viral susceptibility revealed by neural progenitor cell villages](#). Cell Stem Cell. 2023 Mar 2;30(3):312-332.e13. doi: 10.1016/j.stem.2023.01.010. Epub 2023 Feb 15. PMID: 36796362; PMCID: PMC10581885.

The methods section covers the math, and some additional validation of the method is covered by the first supplemental figure.

Analysis and QC

We have released an accompanying R package DropSeq.dropulation that can further analyze the outputs of AssignCellsToSamples and DetectDoublets to produce a number of useful outputs. The function takes in at a minimum the single donor assignment and doublet files along with a file containing a list of donors expected in the experiment. There are some additional input files that will generate other plots, but you can skip those inputs if they are onerous to generate. The main outputs are a text file mapping the cell barcodes to donors of origin for singlet cells, a set of summary metrics, and a PDF containing a number of different QC plots to examine the results in more detail.

As of DropSeq V3, R packages are released as stable binary freezes. You can also install the latest (possibly) unstable version of the code we're using for analysis directly from source code. See the github main page [readme](#) for installation instructions.

QC Text Reports

donor_cell_map

This is a tab delimited text file that maps cell barcodes to the donor of origin for cell barcodes that can confidently be mapped to a single donor. This is done by excluding all cell barcodes assigned to doublets, then filtering on single donor FDR to those cells with a value ≤ 0.05 . There's a final filter to remove cells from donors that are very rare in the pool ($< 0.2\%$), which typically removes zero to a few spuriously assigned cells. This is the subset of cell barcodes that we take into downstream analysis.

Example:

cell	bestSample
AAAGAACGTGCACGCT	donor1
AAAGGATAGAGAACCC	donor2
AAAGGTATCGGTAGGA	donor3

dropulation_summary_stats

A tab delimited text file that contains some metrics about the experiment. This is a companion to the visual report below. When running multiple experiments, this file is a convenient way to aggregate quality control results across many experiments to look for outliers. If some of the optional inputs are not supplied, some of these columns will not have results.

- **expName** - The data set name
- **total_cells** - The total number of cells tested. Cell barcodes show up in this list only if they contain at least one informative UMI.
- **pct_all_doublets** - The percentage of cells that are flagged as either confident doublets (where a single donor pair is much more likely than any other donor pair) or diffuse contamination doublet (where many donor pairs have similar likelihoods.)
- **pct_diffuse_contam_doublets** - The percentage of cells that are flagged as having a higher amount of cell free RNA, such that many pairs of donors could explain the data.
- **pct_confident_doublets** - The percentage of cells where a single donor pair best explains the data.
- **pct_impossible_donors** - The fraction of cells are assigned to donors that are not expected in the pool, before any filtering takes place.
- **pct_fdr_impossible_donors** - The fraction of cells are assigned to donors that are not expected in the pool after FDR filtering.
- **pct_doublet_filtered_impossible_donors** - The fraction of cells are assigned to donors that are not expected in the pool, after both FDR filtering and doublet filtering. This number should be close to 0.
- **singlets** - The number of cells in the pool not flagged as a doublet - not all of these cells may pass the FDR threshold.

- **assignable_singlets** - The number of cells in the pool not flagged as a doublet and pass the FDR threshold
- **cell_equitability** - A measure of how uniform the donor pool is. This metric looks at how many cells are assigned to each donor, and calculates the [diversity index](#), normalized to the range of 0-1, where numbers closer to 1 are closer to a uniform distribution.
- **diversity** - A measure of how uniform the donor pool is. This metric looks at how many total UMIs are assigned to each donor by summing the number of UMIs across all cells assigned to the donor. This is the unnormalized diversity index.
- **equitability** - The diversity index normalized to a range of 0-1.
- **totalUMIs** - The total number of UMIs assigned to singlet cells.
- **reads_per_umi** - The average number of reads per UMI. This can be useful to determine if a library has been under or over sequenced. We typically see this value in the range of 2-4 for pools with appropriate amounts of sequencing.

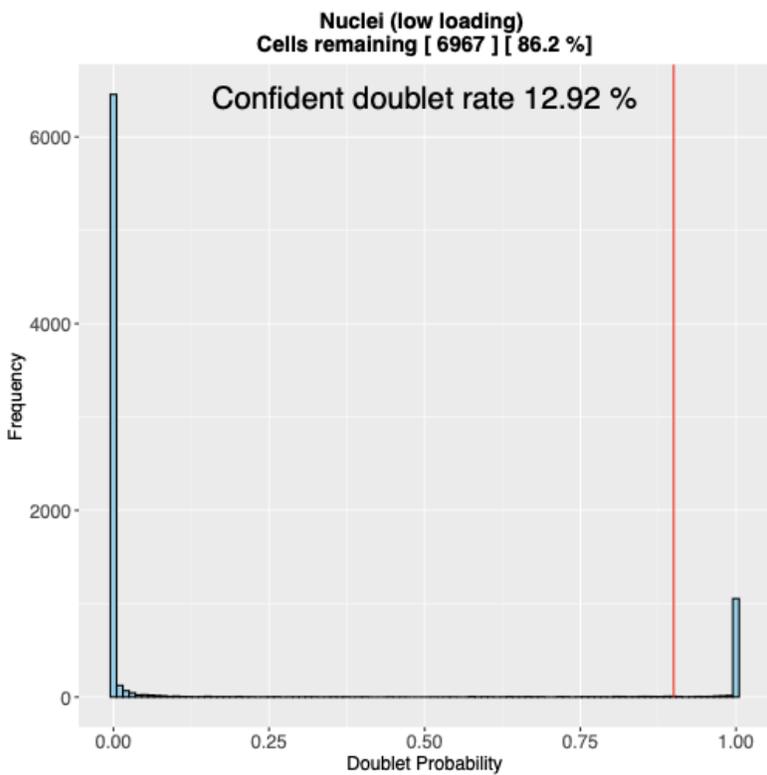
QC Plots

We use a number of QC plots to evaluate the quality of each experiment. Because some experiments are inherently more or less difficult to analyze, we have included a few different data sets to demonstrate those differences.

The R package that generates these plots is available from our [git repository](#). Most of the plots below will be generated by including the outputs from donor assignment and doublet detection. Some additional optional plots can be generated by including a few additional files that contain measurements of the total number of reads and UMIs captured per cell.

Nuclei (low loading)

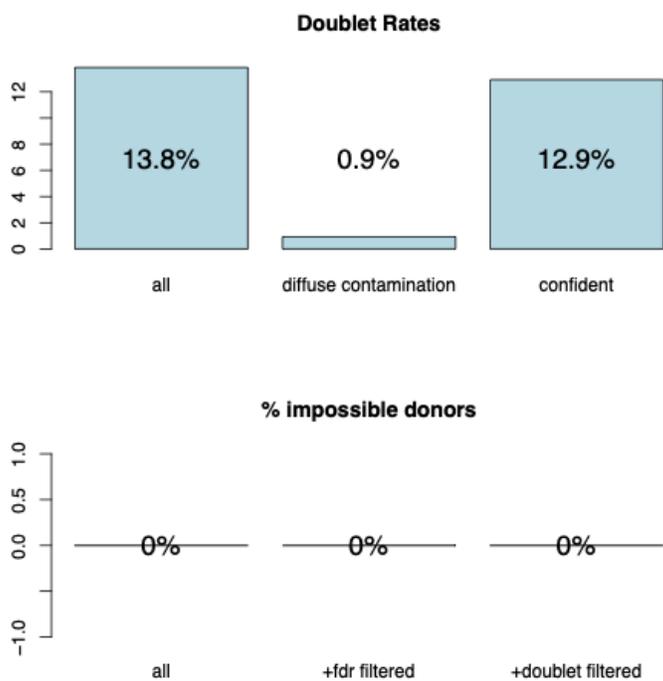
The first data set comes from primary tissue from the brain that was previously frozen and prepared as nuclei. The loading of nuclei to beads in this data set is fairly low, leading to relatively low numbers of total nuclei, lower numbers of doublets and lower levels of cell free RNA..



The distribution of doublet probability scores. In a well behaved experiment this distribution is extremely bimodal and insensitive to where the doublet threshold is set.

The confident doublet rate of 12.92% should approximate the expected doublet rate from poisson loading.

The total fraction of cells remaining after all doublet filtering is 86.2, which includes filtering of both confident doublets and diffuse contamination cells.



The top bar plot lists the fraction of cells filtered by both diffuse contamination and confident doublets.

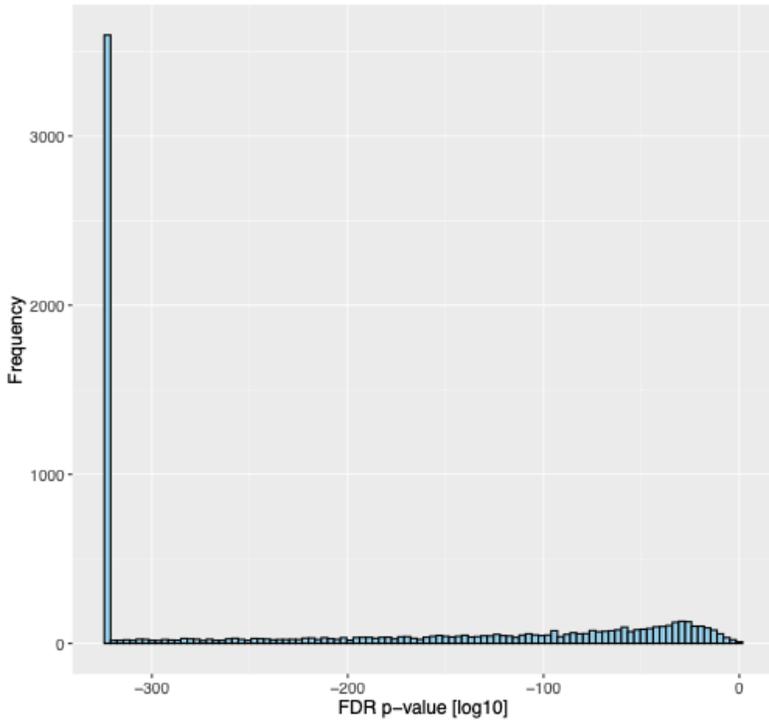
The second barplot looks at the percentage of cells that are assigned to a donor that is unexpected in the pool. If the data set is noisy, it's possible for cells to be assigned to an unexpected donor.

The leftmost bar has no filtering. The center bar adds FDR filtering, and the right bar uses both FDR filtering and removes doublets.

If the % of impossible donors is non-zero but low in the fully filtered data, it's possible that a small number of cells with low numbers of observations are mis-assigned to an incorrect donor. A further filtering step will be applied by a later QC step that may remove these.

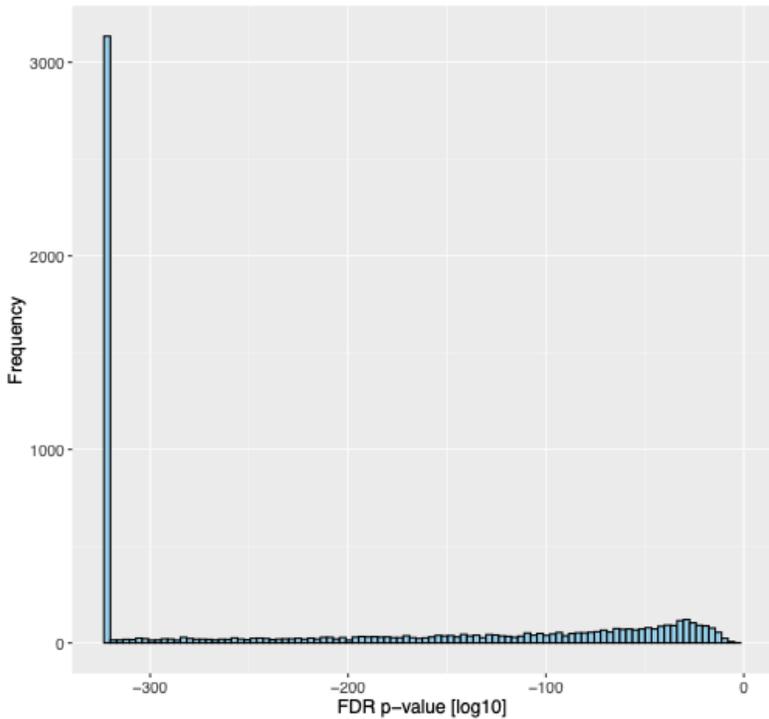
If the % of impossible donors after doublet is high in the right most column, it's possible the expected list of donors is incorrect - either due misspecification, or a sample swap in the experiment. Donor assignment doesn't rely on a sample list to run, so if the correct sample is in the VCF these errors can be detected and corrected later in the report.

All FDR corrected pvalues
99.9 % of cells with FDR < 0.05



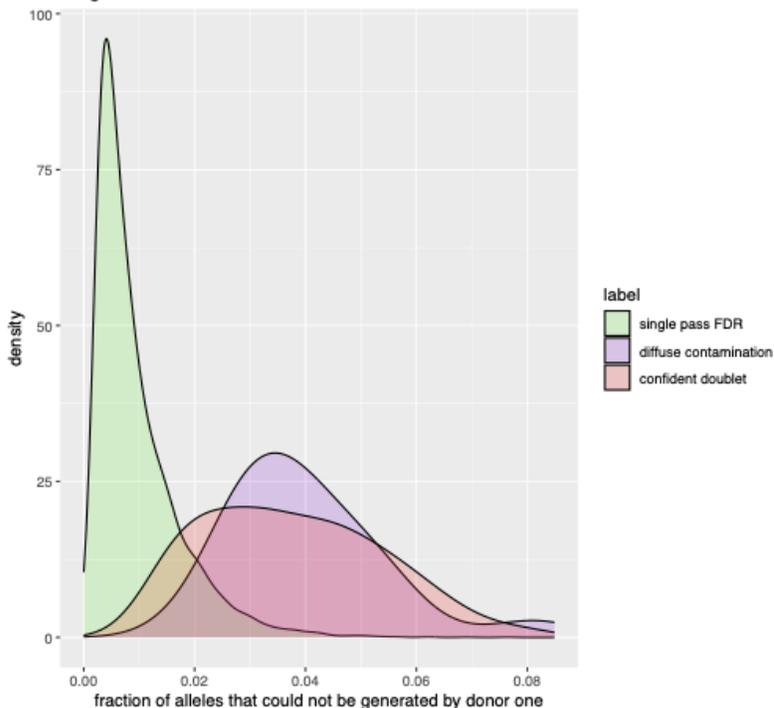
The distribution of FDR corrected p values. In a well powered data set the majority of cells will have very small values. Cells that have values closer to 0 are either cells with few observations, or cells that are doublets where multiple donors had very similar likelihoods.

All FDR corrected pvalues after filtering doublets
86.2% of cells [6967 / 6967] with FDR < 0.05



The FDR distribution after removing doublets.

Nuclei (low loading)
 Singlets (FDR<=0.05) + doublets
 Singlet Error Rate 0.7%

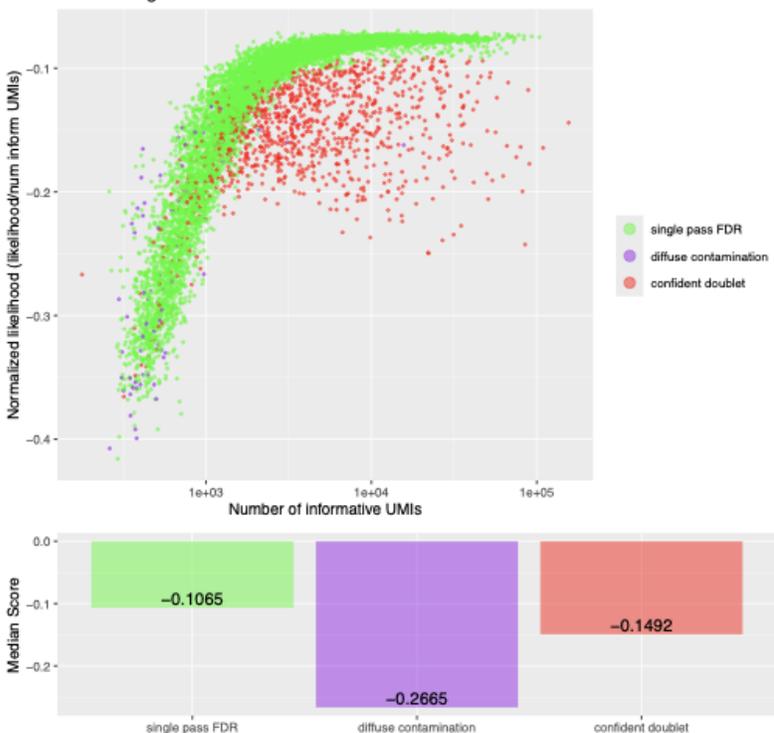


Each cell is assigned to a donor, giving us an expectation of what alleles we should see in the sequencing data. For each cell we can calculate the fraction of observed alleles that could not be generated by the donor - for example if the donor is A/A at a SNP site and we observe a non-A allele, then that allele could not have come from the donor.

The x-axis of the plot captures this error rate. The expectation is that for singlets, the error rate should be low as the observed alleles come from a mixture of the donor cell and cell free RNA that was co-captured in the droplet and assigned to the cell. The error rate in the header is the median error rate of the singlets.

For cells that are doublets, both donors contribute alleles to the data. At sites where the genotypes of the two donors differ, the second donor can contribute alleles that could not be generated by the first donor. As expected, this leads to the confident doublets having a higher error rate than the singlets. The diffuse contamination cells also have a higher error rate, though this may be due to more cell free RNA being captured by these cells.

Donor assignment normalized likelihood

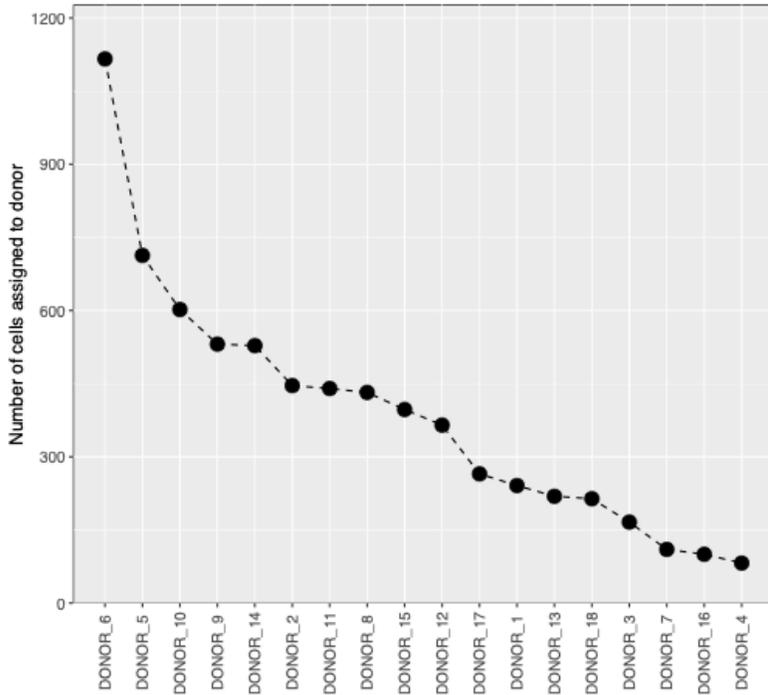


Another way to look at error rates is to calculate the average penalty score for each cell. This is the likelihood of the donor divided by the total number of UMIs observed, which gives an average penalty per UMI. Penalties closer to 0 have fewer errors.

In this plot, each point is a cell. The average likelihood is plotted on the Y axis, and the number of informative UMIs on the X. There's a few interesting features to this plot:

1. The cells labeled as doublets tend to have higher penalty scores for a given number of UMIs. This makes sense, as a second donor contributes alleles that could not have come from the donor.
2. Smaller cells have higher penalty scores on average. A smaller cell will contribute fewer UMIs to the droplet compared to a large cell, but will capture a larger proportion of cell free RNA. The cell free RNA will contain a mixture of alleles from all donors, some of which will be errors.
3. Some diffuse contamination cells have fewer UMIs relative to the observed penalty score. They also tend to be some of the smaller cells captured.

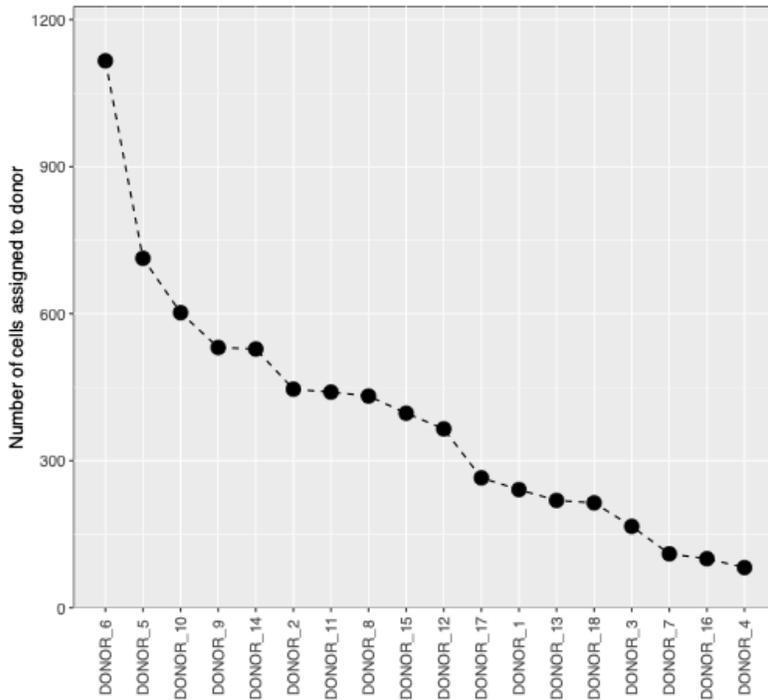
All Donor Assignments seen at least once
[FDR passing cells] [6967]



The number of cells assigned to each donor. The cells included in this list have been filtered by both FDR and doublets have been removed.

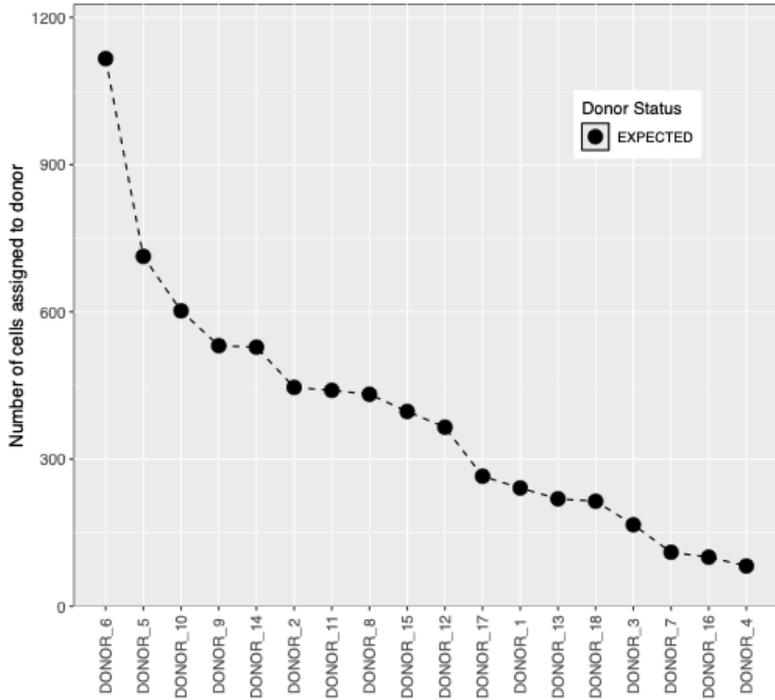
If cells are assigned to unexpected donors (see page 2 plot) those cells will appear here. It's possible in an experiment with a high amount of cell free RNA or with very small cells for a cell to be misassigned to the wrong donor. Those mis assignments would appear in this plot as donors with very few cells assigned.

Common Single Donor Assignment
[FDR passing cells] [6967]



The number of cells assigned to each donor after FDR and doublet filtering. A further filter is applied to remove cells from donors that are less than 0.2% of the total pool. If there are a few spurious assignments, those will be removed in this plot. The filtering threshold [minimumFractionDonor] can be changed by the user depending on the number of donors in the pool from the default, which can be useful if the pool is very large and it is reasonable for a donor to only contribute a small number of cells.

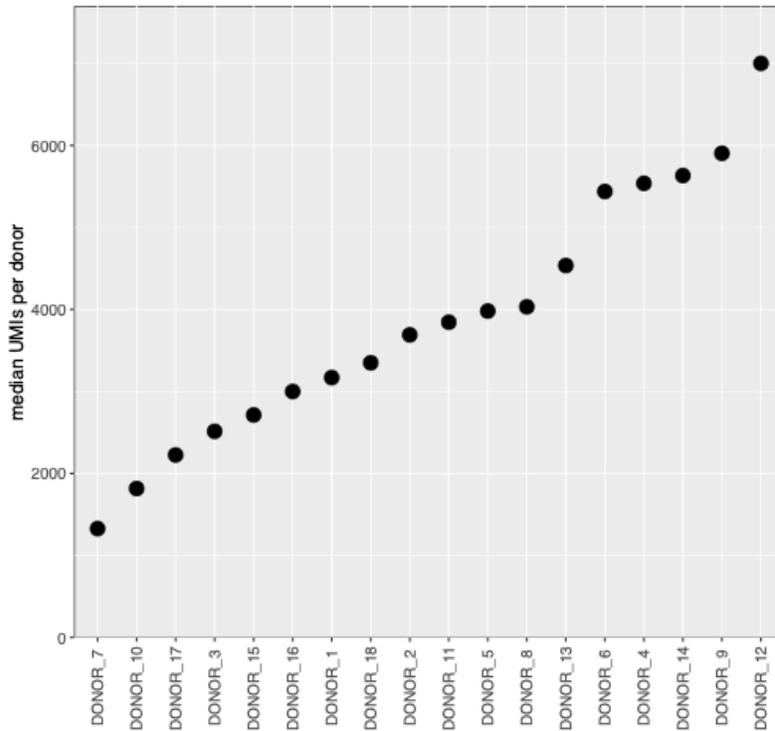
Nuclei (low loading)
SW Div: 2.69; SW Eq: 0.93



This plot has the same filters as the previous plot. This plot includes the priors of which donors should appear in the pool. If donors appear in this plot but are not expected, they will be colored red. If the supplied list of donors contains identifiers that were not detected in the data set, those donors will be included in this output and their count will be set to 0.

It's useful to check this plot both to make sure that only the donors you expected were observed, and all donors you expected were detected.

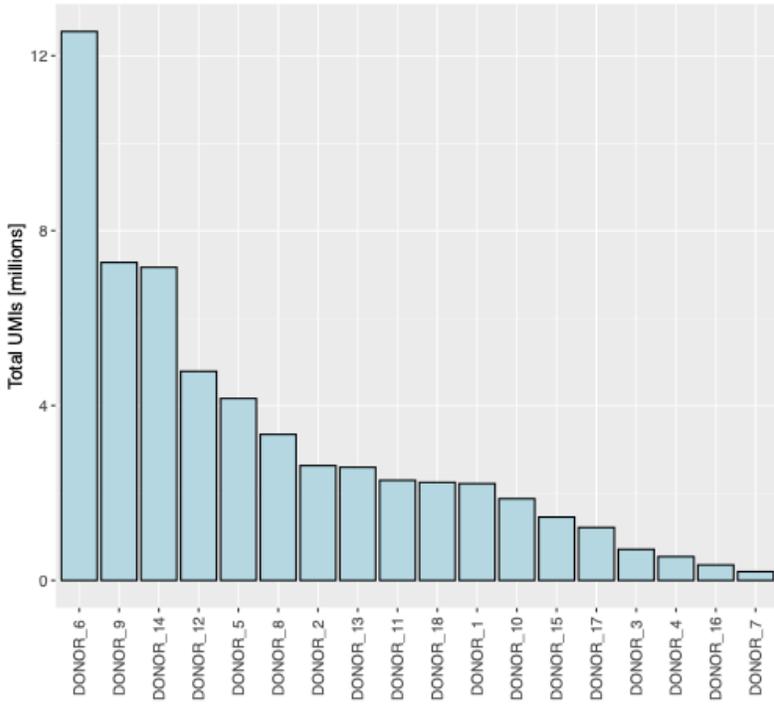
Nuclei (low loading)



The median number of UMIs per donor. This data comes from primary brain nuclei that have been previously frozen.

Differences in tissue quality and preparation may lead to some donors having far more UMIs assigned to them than others.

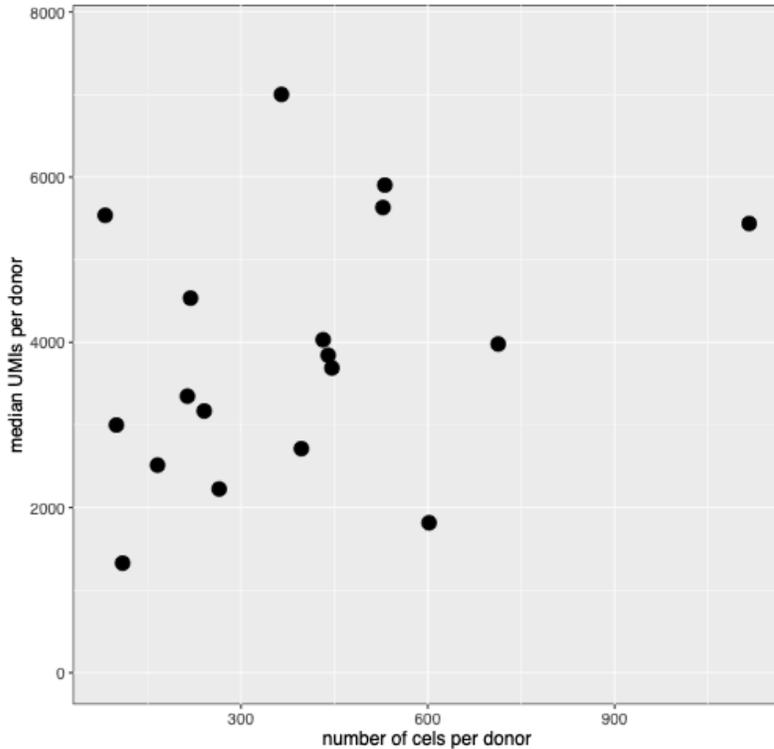
Distribution of UMIs across donors
18 donors; 57.6M UMIs; SW Div: 2.51; SW Eq: 0.87



The distribution of total UMIs per donor. For each donor, the UMIs across all cells are summed together. This metric is useful for downstream processes where analysis is performed on pseudobulk data - for example, standard eQTL analysis combines expression across all cells for each donor into a single measurement.

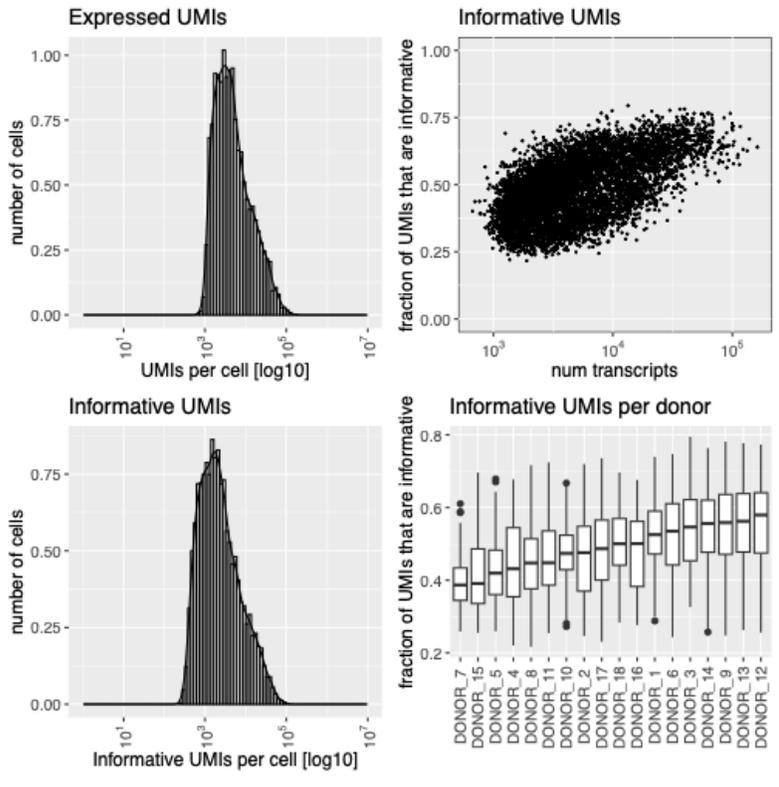
In a perfect experiment, all donors would capture the same number of UMIs. To evaluate how uniform the distribution is, the [Shannon diversity index](#) is calculated. This is then normalized to a scale of 0-1 as Shannon's equitability, with a score of 1 being a completely uniform distribution. This second metric is useful to compare pools to each other when the number of donors in the pool varies.

Nuclei (low loading)



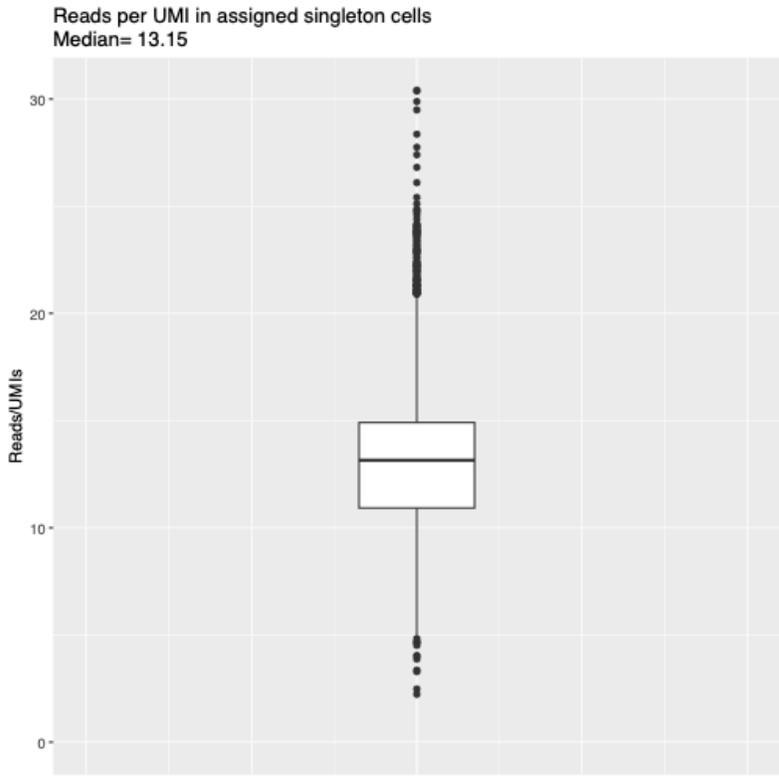
Since the number of UMI per donor scales with both the number of cells captured per donor and the number of UMI captured, it can be useful to look at the relationship of the two parameters.

In particular, it may be helpful to look for donors that have both small numbers of cells and relatively few UMIs captured on average, as the tissue quality of the donor may be low.



These four panels look at the number of informative UMIs observed compared to the number of expressed UMIs. These plots can give you a sense of how well your measured expression converts into power to assign the cells to donors.

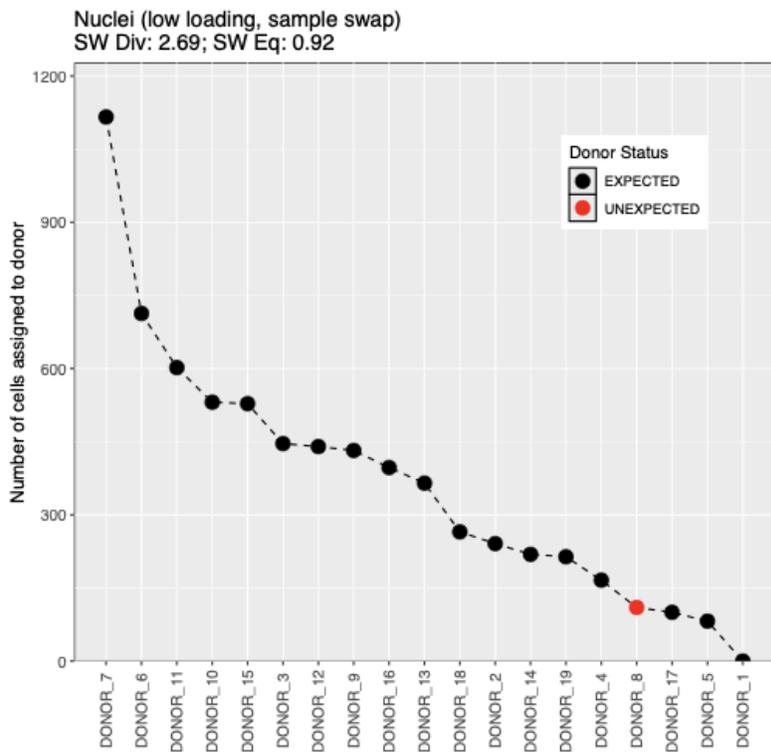
This distribution is typical for what we see in many populations - one would not expect all transcripts to contain a transcribed SNP, but a result in the range of 0.25 - 0.75 is reasonable. If this number is lower than 0.25, then it may be worth investigating the density of your SNP backbone (VCF) file. Increasing the total number of SNPs in the VCF may be useful, especially in cases where the data comes from a SNP array. Imputation of the VCF and filtering to high quality SNPs ($R^2 \geq 0.9$) is recommended in these cases.



This measures the number of reads per UMI for cells that are confidently assigned to single donors.

In general, you want this metric to be in the 2-4 range. This experiment had a low nuclei loading, and may have been sequenced more than necessary.

Sample Swap Example



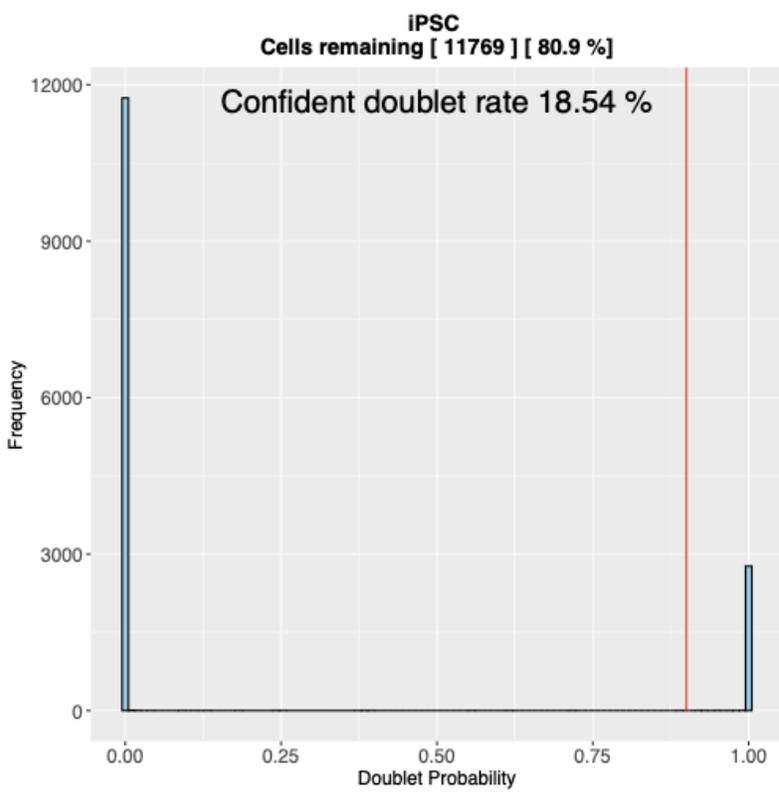
This plot was generated from the same data as the low loading Nuclei data set, but the donor list was altered to simulate sample swap.

In the case of a single sample swap, the plot highlights the unexpected donor in red. This donor was in the VCF file which allowed the cells to be assigned to that donor, but the donor was not in the donor list so was not expected to be in the pool. The donor that was not seen in the data but was in the expected file is added to the plot with 0 counts indicating the donor is expected but absent.

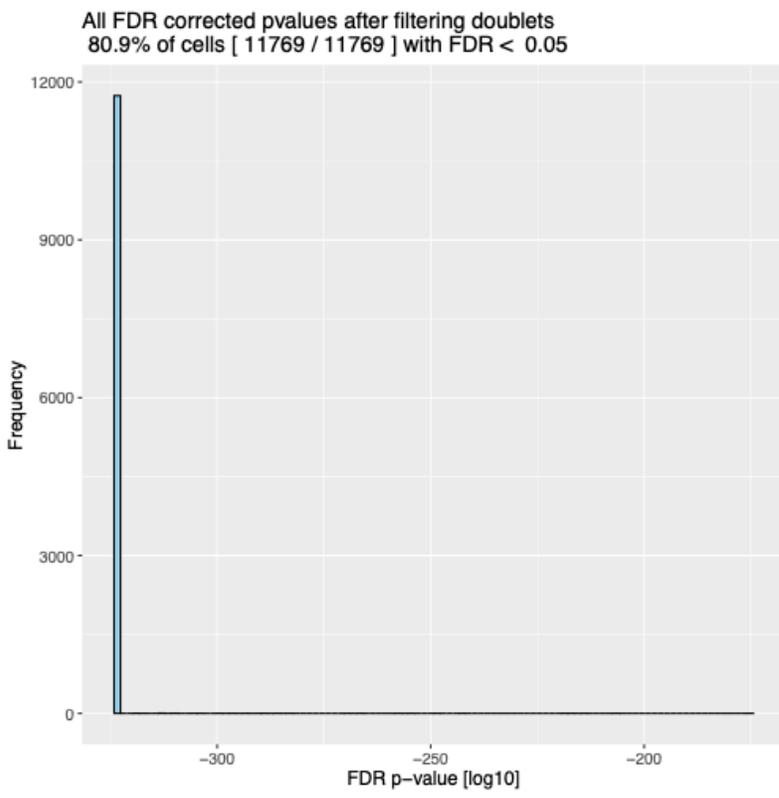
iPSC cells (large pool)

Cultured cells like iPSCs also tend to be uniformly shaped and travel through nanofluidic devices more easily, creating less cell free RNA. Cultured cells also tend to have less donor to donor variability in quality and thus power to the cells to their donor of origin. In this experiment, the 108 donors were added to the pool. This data set had very high UMI yields (median ~ 100,000 UMIs per cell), which provided far more than enough data for donor assignment. Because our implementation of donor assignment and doublet detection scales linearly with the number of cells and donors, this analysis was still straightforward to run on a modest machine with 32G memory.

We'll skip some of the report pages and look at the more interesting ones that may differ from the first example.

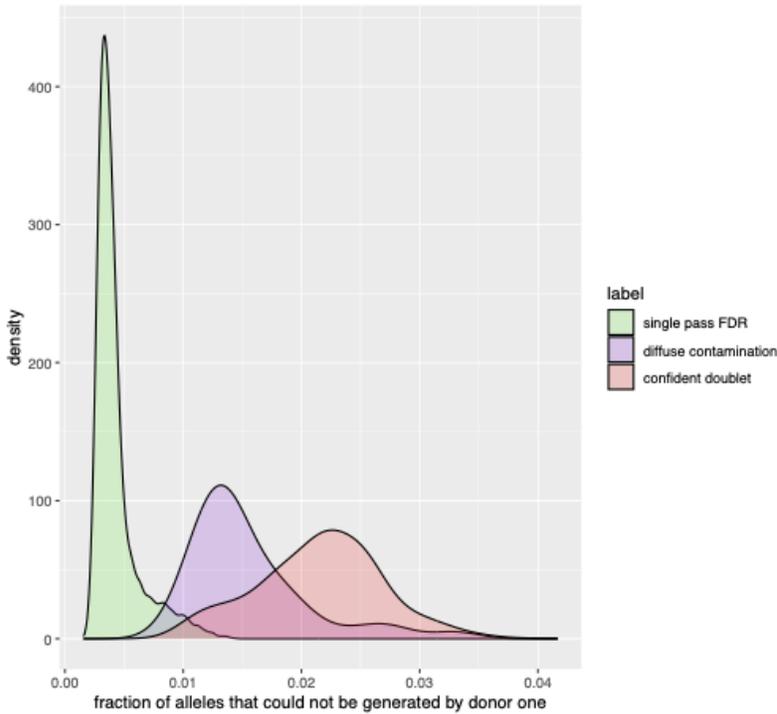


The doublet rate is a bit higher than the first experiment. Due to the average number of UMIs per cell being very high the distribution is even more bimodal than the first example.



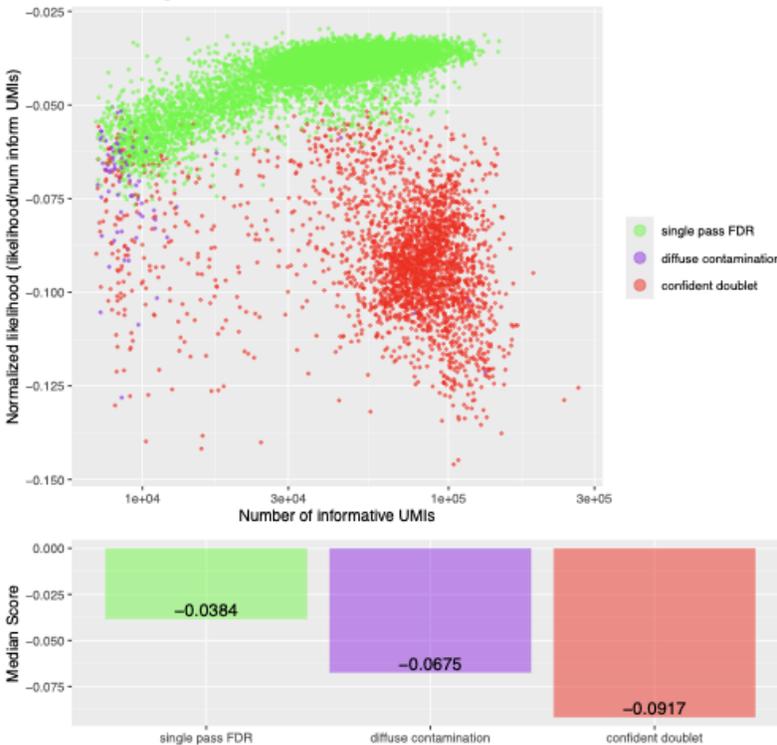
All cells are confidently assigned, there are no cells with FDR scores anywhere near 0.

iPSC
Singlets (FDR<=0.05) + doublets
Singlet Error Rate 0.37%



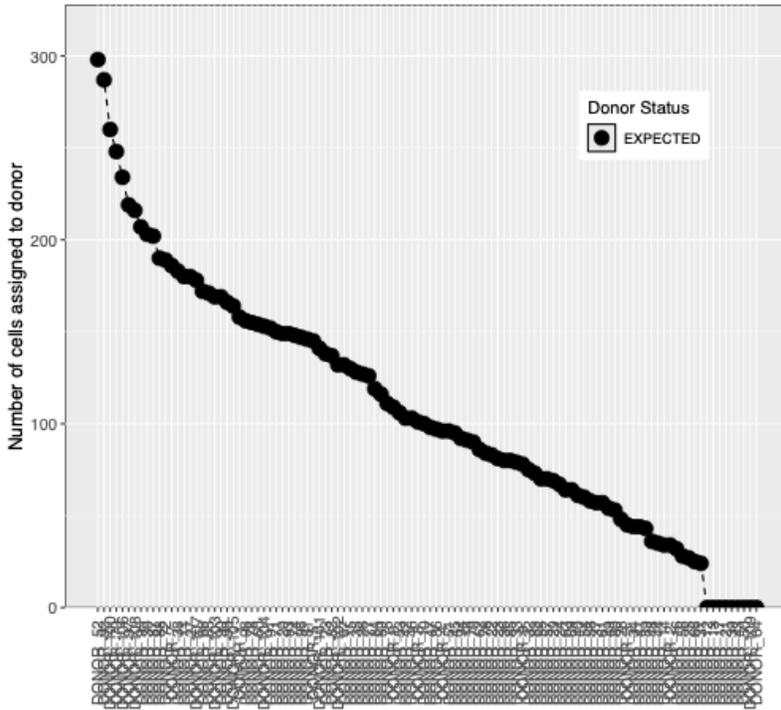
The diffuse and confident doublet penalty scores are better separated from the singlets than the previous example.

Donor assignment normalized likelihood



One striking thing about the iPSC experiment is how much more distinct the doublet class is in this plot than the previous example. Even without cluster labels, it would be possible to properly classify most doublet cells in this data set by finding a cluster of cells with high error rates.

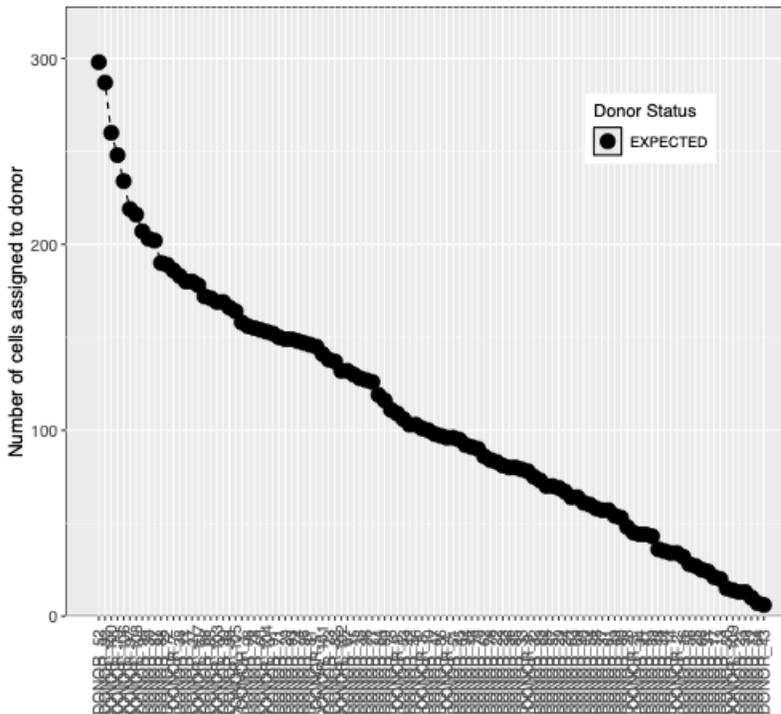
iPSC
SW Div: 4.46; SW Eq: 0.95



Because this pool contains so many donors, it's possible for some donors to have very low representations but be present in the pool.

With the `minimumFractionDonor` threshold set at 0.2%, some donors were removed as being below the representation, then flagged as being missing.

iPSC
SW Div: 4.49; SW Eq: 0.96

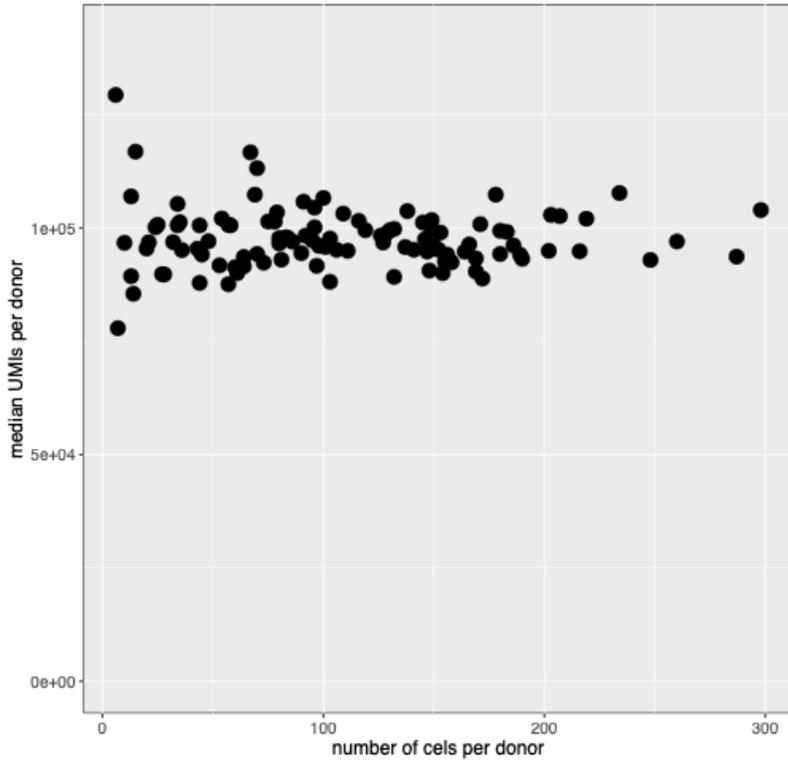


Repeating this plot with a threshold of 0.05%, the donors that were filtered out reappear. With a very large pool size, some donors had very few cells assigned to them, but there were no misclassifications above this threshold.

In fact, there was just 1 cell assigned to a donor not in the expected pool out of 11769 cells. The VCF contained 187 donors, of which 108 were expected in the pool.

Allowing donor assignment to have a chance to try and assign cells to donors outside the expected set further boosts our confidence that donor assignment is working properly, and is capable of detecting sample swaps.

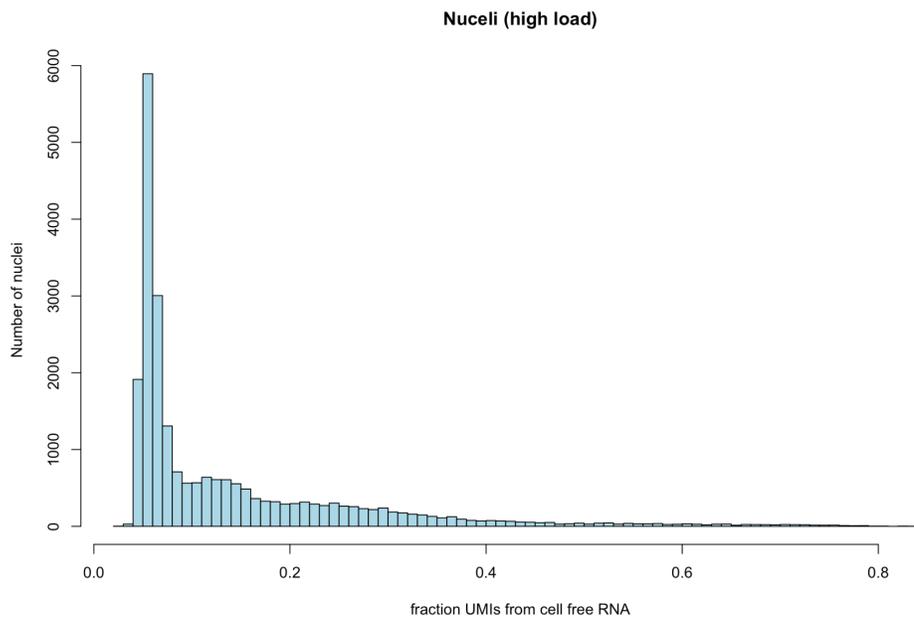
iPSC



Unlike the nuclei data set, the median number of UMIs captured by each donor are very similar. This is an advantage of working with co-cultured cell systems where tissue quality does not play a role.

Nuclei (high loading)

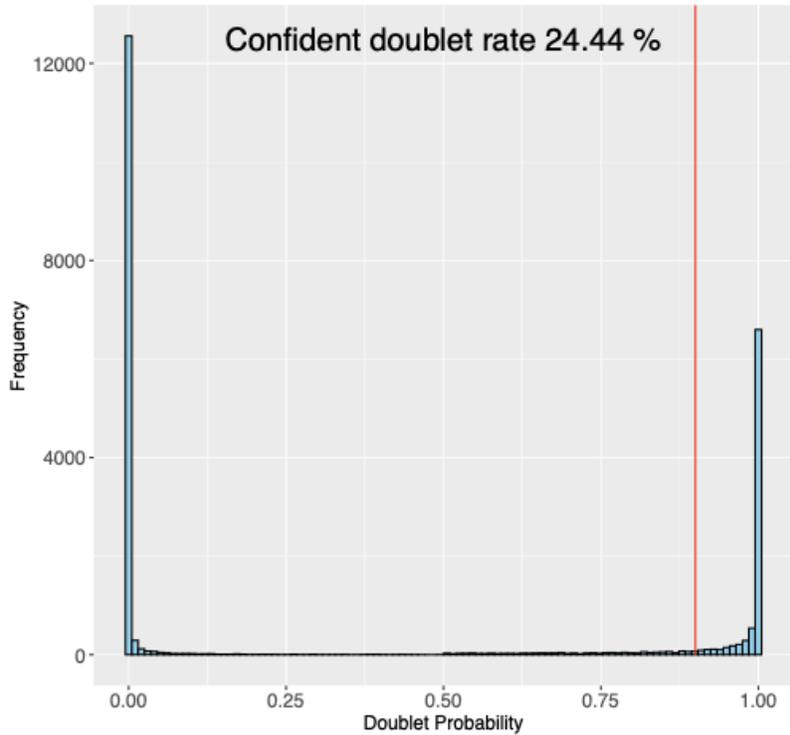
This is a library of nuclei captured from previously frozen primary tissue samples. The pool was a mixture of tissue from 20 donors that was then processed as a single batch. Below is a plot of the fraction of UMIs for each nuclei that are removed by CellBender. 22.2% of nuclei have at least 20% of their UMIs (and thus transcribed alleles) contributed by cell free RNA. CellBender is very helpful to remove these from the expression matrix, but it does not remove specific reads from the BAM file. The amount of cell free RNA is significantly higher in this experiment, which makes donor assignment more challenging.



23374 nuclei were captured in this data set. Donor assignment was run with a SNP backbone containing 1749 donors, which encompassed all donors that had been used in the lab in the last few years, along with a number of other donors that the lab had only received genotypes for, but not physical samples. Running donor assignment on this huge VCF file allowed us to be confident that there were no sample swaps that occurred at any point either during the experiment, or prior to the lab receiving the samples. We ran donor assignment on ~ 2G bam files that contained all reads for a subset of cells, and donor assignment ran successfully with 32G of memory.

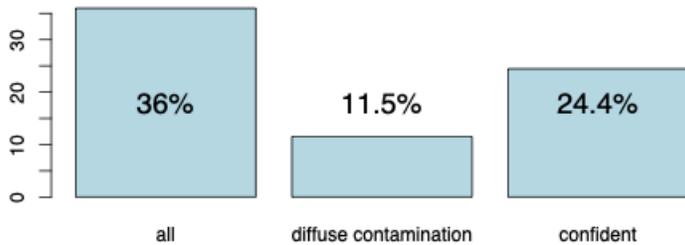
Nuclei (high loading)
Cells remaining [14939] [64 %]

Confident doublet rate 24.44 %

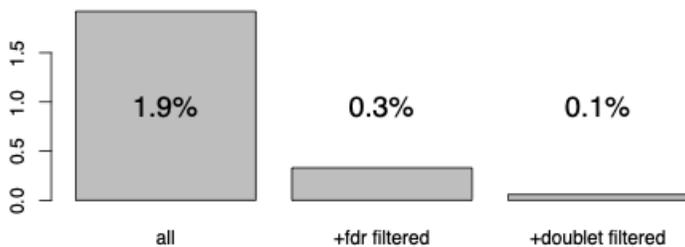


The confident doublet rate is higher in this data set, but even with both confident doublets and diffuse contamination doublets removed, 14939 nuclei of the 23374 tested are confidently assigned. Even with many nuclei discarded, this is a good yield for the experiment.

Doublet Rates



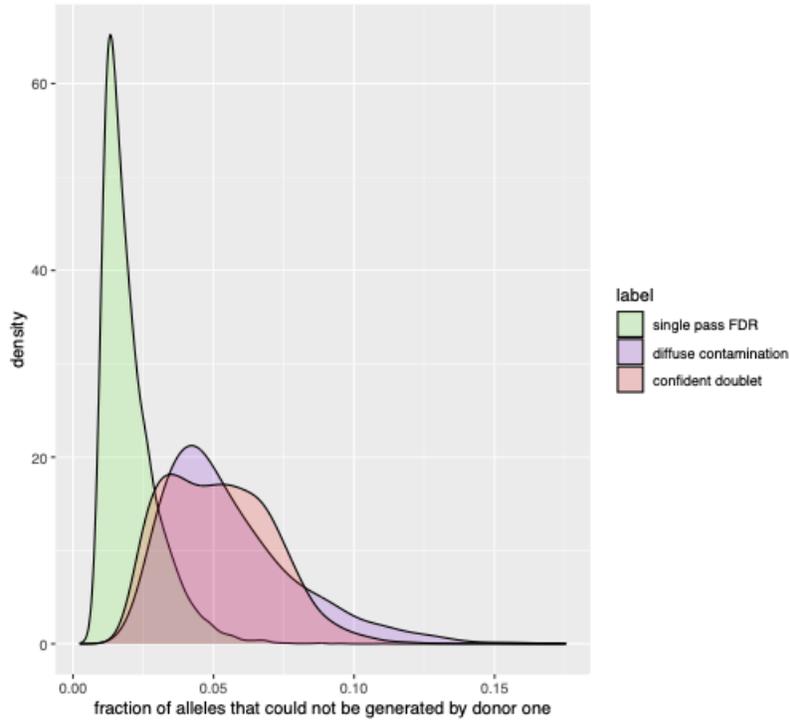
% impossible donors



In the top barplot, the diffuse contamination nuclei class is much larger than the previous two data sets.

In the bottom plot, the filters help remove assignments to donors that are not expected to be in the pool.

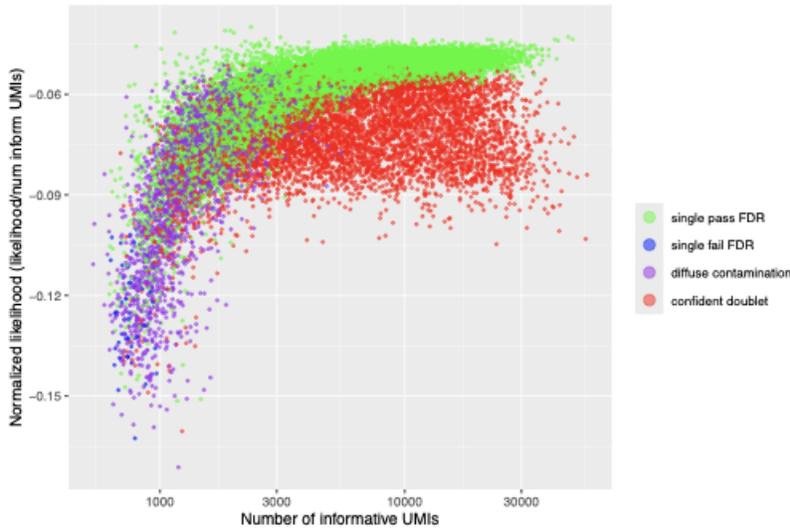
Nuclei (high loading)
 Singlets (FDR≤0.05) + doublets
 Singlet Error Rate 1.75%



The error rate of the singlet cells is much higher in this data set than previous experiments - 1.75% vs 0.7% for the previous nuclei data set. This is a lower bound on the error rate, but useful to check.

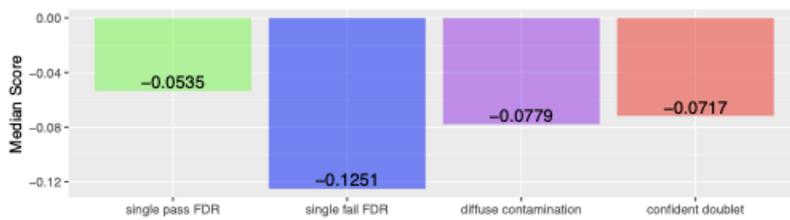
The error rates of the three classes are less well separated with significant overlaps, a hint that doublet detection is more challenging.

Donor assignment normalized likelihood

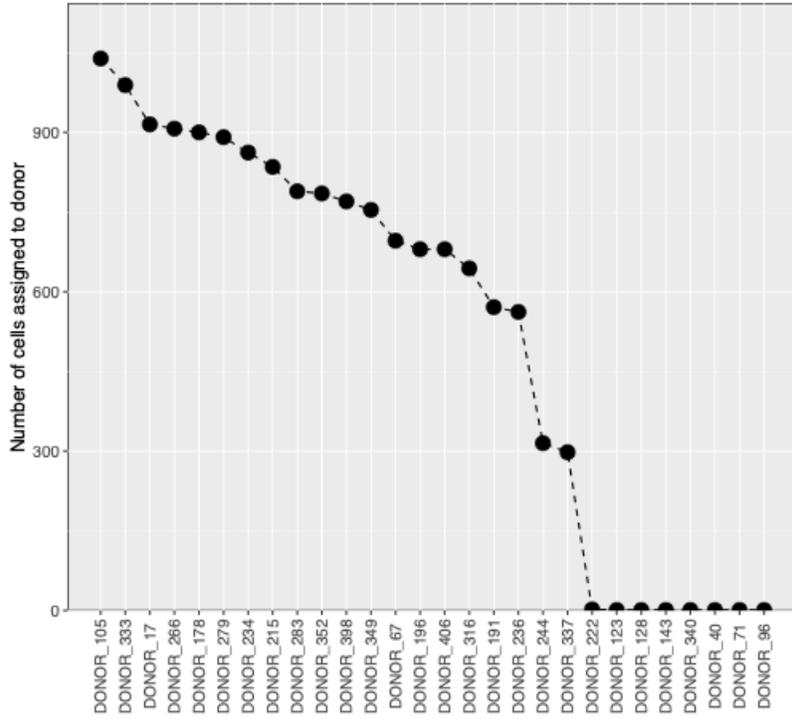


Donor assignment is more difficult for cells with fewer UMIs. Given the CellBender results, it is not surprising that smaller nuclei that capture a larger proportion of cell free RNA are classified as diffuse contamination.

The confident doublets are in a class separate from the singlets, so those labels appear to be reasonable.

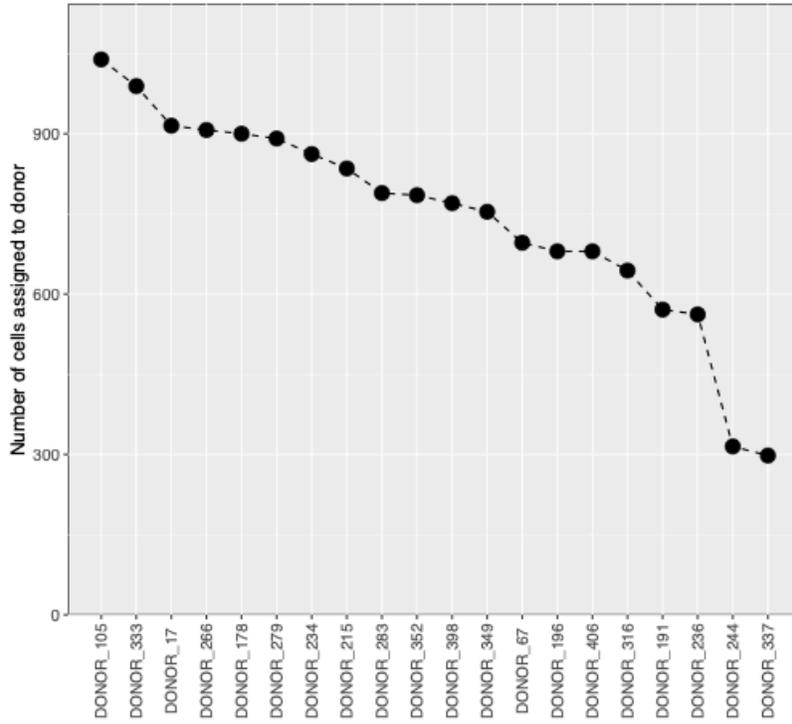


All Donor Assignments seen at least once
[FDR passing cells] [14891]



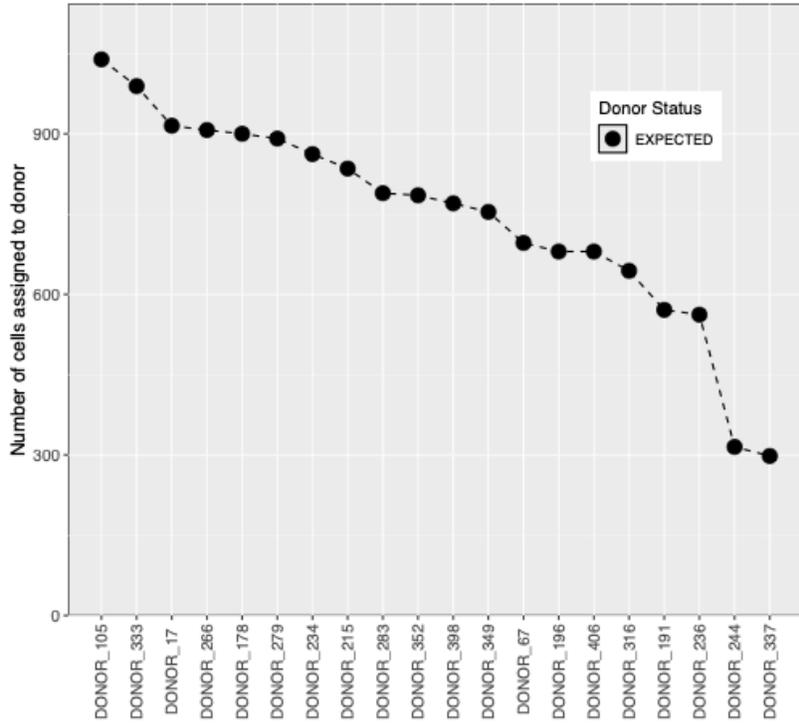
There are 9 cells that pass FDR and are assigned to donors that are not expected in the pool. Those errors are distributed as 1 or 2 nuclei assigned to each of the spurious donors.

Common Single Donor Assignment
[FDR passing cells] [14882]



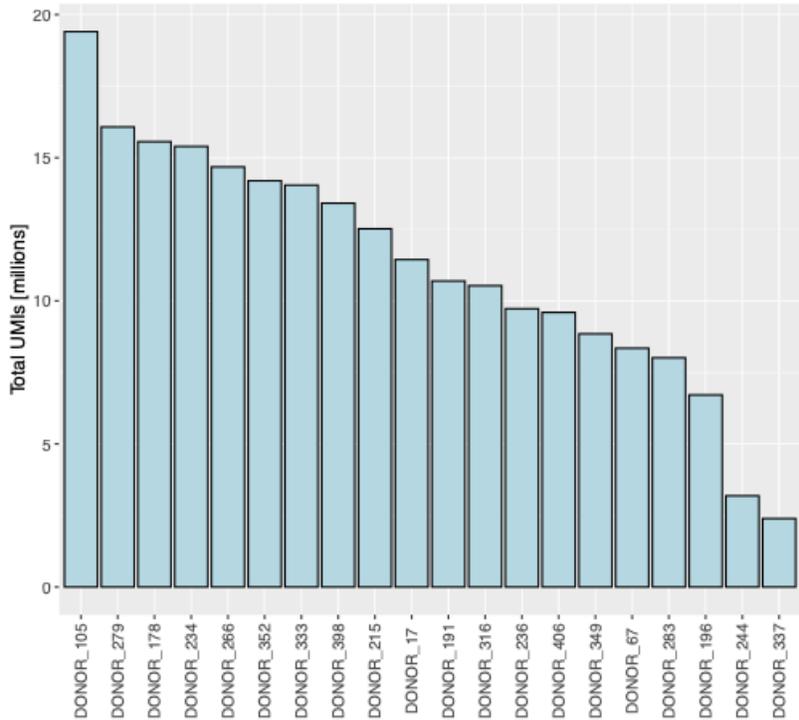
The 9 nuclei that were assigned to spurious donors are removed at this stage, where each donor is expected to contribute at least 0.2% of the total number of nuclei in the experiment.

Nuclei (high loading)
SW Div: 2.96; SW Eq: 0.99



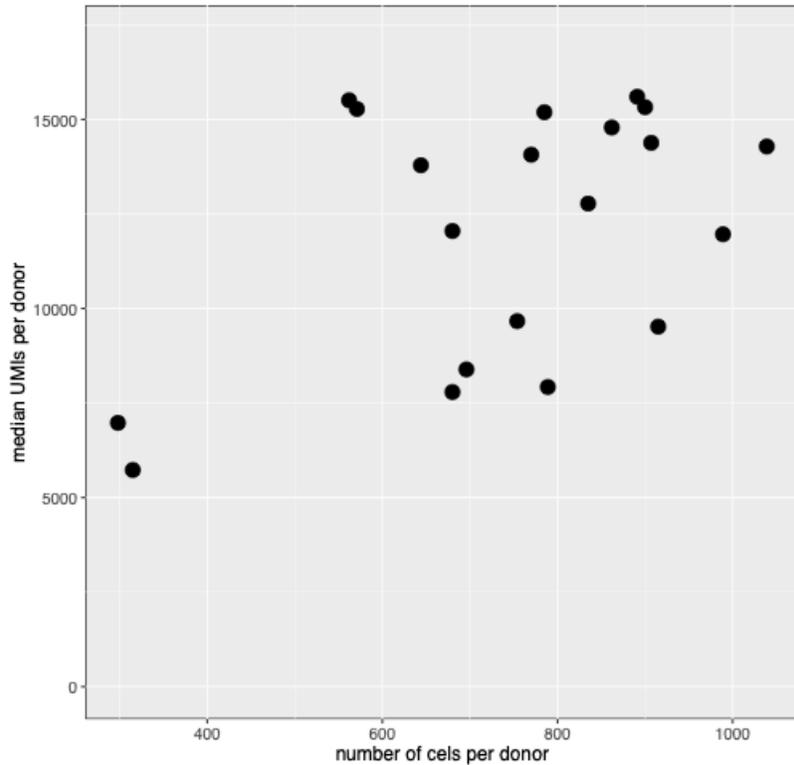
Despite the challenging conditions of the experiment, all remaining nuclei are assigned to the expected set of 20 donors from a possible set of 1749 donors.

Distribution of UMIs across donors
20 donors; 224.8M UMIs; SW Div: 2.92; SW Eq: 0.97



The pool is fairly well balanced, with only two donors having moderately low representation.

Nuclei (high loading)



Those two donors have both fewer cells than expected, and the number of UMIs captured on average for each cell are the lowest of the group. It's possible that the source tissue was not of the same quality as the rest of the experiment.

Experimental diffuse contamination rescue

We have some work in progress that may rescue many of the diffuse contamination cell barcodes and improve yields in these more difficult experiments. Using data from the Nuclei (high load) library, experimental rescue recovers 2051 of the 2692 diffuse contamination cell barcodes (76%). For many data sets this is unnecessary, but may be of use for more challenging sets.

We'd like to design some validation experiments before we move this feature from EXPERIMENTAL to a step we recommend or enable by default. This document will be updated and a new software version will be released if that validation is successful.

This rescue defines two populations - cells that are assigned to a donor that belongs in the pool, and cells that are mistakenly assigned to donors that are not expected in the pool that are more clearly mistakes in assignment. Diffuse contamination cells tend to have smaller numbers of UMIs and contain more cell free RNA than other cells, so are more likely to be assigned to a random donor in the superset. These cells are then rescued by finding linear separations between the two populations that maximize the number of assignments to donors that belong in the pool while controlling the FDR rate to minimize the number of cells that are assigned to a donor not expected in the pool.

If you wish to experiment with this yourself, it's very important that donor assignment has been run on a superset of donors in the experiment pool. We frequently run donor assignment on large VCFs containing hundreds of donors. Review your plot of % impossible donors to see if you have some cells that are assigned to a donor not expected in the pool before attempting rescue.

