Genotype-free demultiplexing of pooled single-cell RNA-Seq

Jun Xu[1], Caitlin Falconer[2], Quan Nguyen[2], Joanna Crawford[2], Brett D. McKinnon[2,5], Sally Mortlock[2], Anne Senabouth[4], Stacey Andersen[1,2], Grant W. Montgomery[1,2], Joseph Powell[3,4], and Lachlan Coin[1,2,*]

[1]Genome Innovation Hub, The University of Queensland, 306 Carmody Road, St Lucia, QLD 4072, Brisbane, Australia
[2]Institute for Molecular Bioscience, The University of Queensland, 306 Carmody Road, St Lucia, QLD 4072, Brisbane, Australia
[3]UNSW Cellular Genomics Futures Institute, School of Medical Sciences, University of New South Wales, NSW 2052, Sydney, Australia
[4]Garvan-Weizmann Centre for Cellular Genomics, Garvan Institute, 384 Victoria St, Darlinghurst, NSW 2010, Sydney, Australia
[5]Department of Obstetrics and Gynaecology, University hospital of Berne, 3010, Berne, Switzerland
[*]Correspoinding Author: l.coin@imb.uq.edu.au

ABSTRACT

A variety of experimental and computational methods have been developed to demultiplex samples from pooled individuals in a single-cell RNA sequencing (scRNA-Seq) experiment which either require adding information (such as hashtag barcodes) or measuring information (such as genotypes) prior to pooling. We introduce scSplit which utilises genetic differences inferred from scRNA-Seq data alone to demultiplex pooled samples. scSplit also extracts a minimal set of high confidence presence/absence genotypes in each cluster which can be used to map clusters to original samples. Using a range of simulated, merged individual-sample as well as pooled multi-individual scRNA-Seq datasets, we show that scSplit is highly accurate and concordant with demuxlet predictions. We also show that multiplexed-scRNAseq can be used to reduce batch effects caused by technical biases. scSplit is ideally suited to samples for which external genome-wide genotype data cannot be obtained (for example non-model organisms), or for which it is impossible to obtain unmixed samples directly, such as mixtures of genetically distinct tumour cells, or mixed infections. scSplit is available at: https://github.com/jon-xu/scSplit

Introduction

Using single-cell RNA sequencing (scRNA-Seq) to cell biology at cellular level provides greater resolution than 'bulk' level analyses, thus allowing more refined understanding of cellular heterogeneity. For example, it can be used to cluster cells into sub-populations based on their differential gene expression, so that different fates of cells during development can be discovered. Droplet-based scRNA-Seq (for example Drop-Seq[1] or 10X Genomics Systems[2]) allows profiling large numbers of cells for sequencing by dispersing liquid droplets in a continuous oil phase[3] in an automated microfluidics system, and as a result is currently the most popular approach to scRNAseq despite a high cost per run. Methods that lower the per sample cost of running scRNA-Seq are required in order to scale this approach up to population scale. An effective method for lowering scRNA-Seq cost is to pool samples prior to droplet-based barcoding with subsequent demultiplexing of sequence reads.

Cell hashing[4] based on Cite-seq[5] is one such experimental approach to demultiplex pooled samples. This approach uses oligo-tagged antibodies to label cells prior to mixing, but use of these antibodies increases both the cost and sample preparation time per run. Moreover it requires access to universal antibodies for organism of interest, thus limiting applicability at this stage to human and mouse. Alternatively, computational tools like demuxlet[6] have been developed to demultiplex cells from multiple individuals, although this requires additional genotyping information to assign individual cells back to their samples of origin. This limits the utility of demuxlet, as genotype data might not be available for different species; biological material may not be avaliable to extract DNA; or the genetic differences between samples might be somatic in origin.

Another issue for droplet-based scRNA-Seq protocols is the presence of doublets, which occurs when two cells are encapsulated in same droplet and acquire the same barcode. The proportion of doublets increases with increasing number of cells barcoded in a run. It is imperative that these are flagged and removed prior to downstream analysis. Demuxlet[6] uses external genotype information to address this issue, and other tools have been developed to solve this issue based on expression

data alone, including Scrublet[7] and Doubletdetection[8].

Here we introduce a simple, accurate and efficient tool, mainly for droplet-based scRNA-Seq, called "scSplit", which uses a hidden state model approach to demultiplex individual samples from mixed scRNA-Seq data with high accuracy. Our approach does not require genotype information from the individual samples to demultiplex them, which also makes it suitable for applications where genotypes are unavailable or difficult to obtain. scSplit uses existing bioinformatics tools to identify putative variant sites from scRNA-seq data, then models the allelic counts to assign cells to clusters using an expectation-maximisation framework.

Results

Our new tool, scSplit for demultiplexing pooled samples from scRNAseq data, only requires the FASTQ files obtained from single cell sequencing, together with a white-list of barcodes, while it does not require either genotype data, nor a list of common variants. All result data are available via `https://github.com/jon-xu/scSplit_paper_data`.

Runs on simulated data shows high accuracy and efficiency of scSplit

We used a single scRNA-seq BAM file from Zheng et al.[2] as a template for simulation. Additionally, we took eight unrelated samples (1043, 1249, 1511, 1493, 1598, 1085, 1079, 1154) from a merged VCF file used in figure 2 supplementary data of Kang et al.[6], as the source of multi-sample genotype likelihoods for simulation (see "Data simulation" in Methods). We ran simulation tests using our scSplit tool and used the distinguishing variants to identify the individual donor for each cluster. We obtained very high overall TPR (0.97) and low FDR (less than 1e-4) for two-, three-, four- and eight-mixed samples, and also very accurate doublet predictions (Table 1). All clusters including doublet cluster match perfectly with known samples (Figure 1A).

scSplit performs similarly well as demuxlet in demultiplexing merged individually-sequenced samples

We then tried running scSplit on a manual merging of three individually-sequenced samples. We merged the BAM files from three individual samples (Methods). In order to create synthetic doublets, we randomly chose 500 barcodes whose reads were merged with another 500 barcodes. We ended up with 9,067 singlets and 500 doublets, knowing the their sample origins prior to merging. Both scSplit and demuxlet[6] pipelines were run on the merged samples, and the results were compared with the known truth. The results showed that scSplit was as accurate in detecting singlets but could be better than demuxlet in doublet detection, if information on read depth was used to detect more doublets from the originally assigned singlets (Table 2).

Comparing scSplit with demuxlet on real pooled samples scRNA-Seq data

Next, we ran our tool on a set of genotyped and scRNA-Seq mixed-sample datasets. According to the scSplit pipeline, distinguishing variants were identified, and P/A matrix was generated to assign the cells to clusters (Methods). We then extracted the reference and alternative allele absence information at these distinguishing variants from the sample genotypes and generated a similar P/A matrix. Both matrices were compared so that clusters were mapped to samples (Figure 2).

Based on cluster-to-sample mapping, we checked the similarity of cell assignments between scSplit and demuxlet, and observed high concordance in singlets prediction (TPR: 0.93-0.94, FDR: 0.06-0.07), although not on doublets (TPR: 0.08-0.52, FDR: 0.45-0.92) if demuxlet was treated as gold-standard (Figure 1D and Table 3).

We also compared the performance of overall P/A genotyping matrices generated based on scSplit and demuxlet predictions against that from the known genotypes ("Alternative allele Presence/Absence genotyping for clusters" section of Methods). The results show that genotypes inferred from scSplit clusters have high concordance with sample genotypes, and higher than those inferred from demuxlet clusters (on average 98.4% vs 95.9%) (Table 4).

Pooled samples show less batch effects than individually sequenced ones

We further checked the gene expression profiles of the three individual samples mentioned previously (Figure 1C and Table 2), in two different scenarios: 1) sequenced individually; 2) pooled the samples followed by demultiplexing after sequencing. The results plotted using Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP[9]); (Figure 3) showed a clear batch effect when we sequenced them in different runs, which could be minimized if we multiplexed them before sequencing. Pooling multiple samples and sequence them together followed by demultiplexing processes could remove the need for normalization and keep as much information as possible for downstream analyses.

Discussion

We developed the scSplit toolset to facilitate accurate, cheap and fast demultiplexing of mixed scRNA samples, without needing sample genotypes prior to mixing. scSplit also generates a minimum set of alleles (as few as the sample numbers), enabling

researchers to link the resulting clusters with the actual samples by comparing the allele presence at these distinguishing loci. When predefined individual genotypes are not available as a reference, this can be achieved by designing a simple assay focused on these distinguishing variants (such as a Massarray or multiplexed PCR assay). Although the tool was mainly designed for droplet-based scRNA-Seq, it can also be used for scRNA-Seq data generated from other types of scRNA-Seq protocols.

We filtered out indels, MNPs and complex rearrangements when building the model and were able to show that SNVs alone provide adequate information to delineate the differences between multiple samples. As an alternative to using allele fractions to model multiple samples, genotype likelihoods could also be used for the same purpose, however more memory and running time would likely be needed, especially when barcode numbers in mixed sample experiments increase. Our tests showed no discernable difference in accuracy between these two methods.

The current version of scSplit assumes that the number of mixed samples are known. It is possible to run the scSplit tool for different sample numbers and compare the model log-likelihoods to select the most likely number of samples being modelled, but this would require significant computational resources and time. Alternatively, the reference and alternative allele counts in different samples and the size of the doublet cluster could be used to determine the sample number. Further optimization of the tool would be needed to effectively implement these options.

Although scSplit was mainly tested on human samples, it can also be applied to other organisms, and is especially useful for those species without dense genotyping chips available. We also expect the application of scSplit in cancer related studies, to distinguish tumor cells from healthy cells, as well as to distinguish tumor sub clones.

Methods

(All relevant source codes available at https://github.com/jon-xu/scSplit/)

Overview

The overall pipeline for the scSplit tool includes seven major steps (Figure 4):

1. Data quality control and filtering: The mixed sample BAM file is first filtered to keep only the reads with a list of valid barcodes to reduce technical noise. Additional filtering is then performed to remove reads that meet any of the following: mapping quality score less than 10, unmapped, failing quality checks, secondary or supplementary alignment, or PCR or optical duplicate. The BAM file is then marked for duplication, sorted and indexed.

2. SNV calling (Figure 4A): Freebayes v1.2[10] is used to call SNVs on the filtered BAM file, set to ignore insertions and deletions (indels), multi-nucleotide polymorphisms (MNPs) and complex events. A minimum base quality score of one and minimum allele count of two is required to call a variant. The output VCF file is further filtered to keep only SNVs with quality scores greater than 30.

3. Building allele count matrices (Figure 4B): The "build_matrices.py" script is run which produces two .csv files, one for each of reference and alternate allele counts as output.

4. Model initialization (Figure 4C): find the distinct groups of cells in the scRNA-Seq and use them to initialize the Allele Fraction Model (SNVs by samples).

5. E-M iterations till convergence (Figure 4D): Initialized allele fraction model and the two allele count matrices are used together to calculate the probability of each cell belonging to the clusters. After each round, allele fraction model is updated based on the probability of cell assignment and this is iterated until overall likelihood of the model reaches convergence.

6. Alternative presence/absence genotypes (Figure 4E): matrix indicating cluster genotypes at each SNV is built in this step.

7. Find distinguishing variants for clusters and use to assign samples to clusters (Figure 4F): In order to assign each model cluster back to the specific sample, distinguishing variants are identified so that genotyping of the least number of loci using the a suitable platform may be performed. Gram-Schmidt orthogonalization[11] is used to get the minimum set of informative P/A genotypes.

Data quality control

Samtools was used to filter the reads with verified barcodes for mapping and alignment status, mapping quality, and duplication (samtools view -S -bh -q 10 -F 3844 [input] >[output]). Duplicates were removed (samtools rmdup [input] [output]) followed by sorting and indexing.

SNV calling on scRNA-Seq dataset

SNVs were called on the scRNA-Seq mixed sample BAM file with freebayes[10], a widely used variant calling tool. The freebayes arguments "-iXu -q 1" were set to ignore indels and MNPs and exclude alleles with supporting base quality scores of less than one. This generated a VCF file containing all SNVs from the mixed sample BAM file.

Building allele count matrices

Allele count matrices were then built from 1) the provided mixed sample BAM file and 2) the VCF file obtained from the SNV calling program. Two allele count matrices were generated, one for the reference alleles and one for the alternate alleles, each with SNVs in rows and barcodes in columns. Each data element in the matrix indicated either the number of reference or alternate alleles detected in one cell barcode at that specific SNV position. This provided a full map of the distribution of reference and alternate alleles across all barcodes at each SNV.

The allele count matrices captured information from all reads overlapping SNVs to reflect the different allele fraction patterns from different barcodes or samples. To build the allele count matrices, pysam fetch[12] was used to extract reads from the BAM file. The reads overlapping each SNV position were fetched and counted for the presence of the reference or alternate allele. In order to increase overall accuracy and efficiency, SNVs whose GL(RA) (likelihood of heterozygous genotypes) was lower than $\log_{10}(1-error)$ where error = 0.01 were filtered out. These were more homozygous and thus less informative for detecting the differences between the multiple samples. The generated matrices were exported to csv files for further processing.

Model initialization by using maximally informative cluster representatives

To initialize the model, initial probabilities of observing an alternative allele on each SNV position in each cluster were calculated. The overall matrix was sparse and a dense sub-matrix with a small number of zero count cells was generated. To do that, cells were first sorted according to their number of zero allele counts (sum of reference and alternative alleles) at all SNVs and SNVs were similarly sorted according to their number of zero allele counts (sum of reference and alternative alleles) across all cells. Next, we selected and filtered out 10% of the cells among those with the most number of zero expressed SNVs and 10% of the SNVs among those where the most number cells had zero counts. This was repeated until all remaining cells had more than 90% of their SNVs with non-zero allele counts and all SNVs had non-zero counts in more than 90% of cells. This subset of matrices was the basis for the seed barcodes to initialize the whole model. The sub-matrix was transformed using PCA with reduced dimensions and then K-means clustering was performed to split the cell subset into expected number of clusters. By using the allele fractions on the subset of SNVs in these initially assigned cells, each cluster of the model could be initialized. Let $N(A_{c,v})$ and $N(R_{c,v})$ be the Alternative and Reference allele counts on SNV v and cell c accordingly, and let $pseudo_A R$ be the pseudo allele count for both Alternative and Reference alleles, and $pseudo_A$ be the pseudo allele count for Alternative alleles, we calculated $P(A_v|S_n)$, the probability of observing Alternative allele on SNV v in Sample n, according to below equation:

$$P(A_v|S_n) = \frac{\left[\sum_c N(A_{c,v}) + pseudo_A)\right]}{\left[\sum_c N(A_{c,v}) + \sum_c N(R_{c,v}) + pseudo_{AR}\right]} \tag{1}$$

Expectation–maximization approach

The Expectation-maximization (EM) algorithm[13] was used to conduct iterations using the full allele count matrices (Figure 4). Each iteration consisted of an E-step to calculated the probablity of seeing cells in all clusters, based on the allele fraction model, and an M-step to use the new probability of seeing cells in all clusters to update the allele fraction model. EM iterations stopped when convergence was reached, so that the overall probability of observing the cells, or the reference/alternative alleles count matrices, was maximized.

During the E-step, the tool first calculated $P(C_i|S_n)$, the likelihood of observing a cell C_i in sample S_n, which was equal to the product of the probability of observing the allele fraction pattern over each SNV, which in turn equaled to the product of probability of having observed the count of alternative alleles and probability of having observed the count of reference alleles. Let c_i be the i-th cell, S_n be the n-th sample, A_v be the Alternative allele on SNV v, and N(A), N(R) be the quantity of Alternative and Reference alleles:

$$P(C_i|S_n) = P(A_{c_i}, R_{c_i}|S_n) = \prod_v \left[P(A_v|S_n)^{N(A_{c_i,v})}\left[1 - P(A_v|S_n)\right]^{N(R_{c_i,v})}\right] \tag{2}$$

And then $P(C_i|S_n)$ was transformed to $P(S_n|C_i)$, the cell-sample probability, i.e. the probability of a cell C_i belonging to sample S_n, using Bayes' theorem, assuming equal sample prior probabilities ($P(S_1) = P(S_2) = ... = P(S_n)$):

$$P(S_n|C_i) = \frac{P(C_i|S_n)}{\sum_{x=1}^{n} P(C_i|S_n)} \tag{3}$$

Next, weighted allele counts were distributed to the different cluster models according to the cell-sample probability, followed by the M-step, where the allele fraction model represented by the alternative allele fractions was updated using the newly distributed allele counts, so that allele fractions at all SNV positions in each sample model was recalculated:

$$P(A_v|S_n) = \frac{\sum_i N(A_{c,v})P(S_n|C_i) + pseudo_A}{\sum_i N(T_{c,v})P(S_n|C_i) + pseudo_{AR}} \tag{4}$$

Let $P(S_n)$ be the probability of seeing the n-th sample, the overall log-likelihood of the whole model[14] was calculated as:

$$\mathscr{L}_{model} = \sum_i \log \sum_n P(C_i|S_n) = \sum_i \log \sum_n \prod_v \left[P(C_{i,v}|S_{n,v})P(S_n)\right] \tag{5}$$

Multiple runs to avoid local maximum likelihood

The entire process was repeated for 30 rounds with the addition of randomness during model initialization and the round with the largest sum of log likelihood was taken as the final result. Randomness was introduced by randomly selecting the 10% of cells and SNVs to be removed from the matrices during initialization from a range of the lowest ranked cells and SNVs as detailed previously.

Cell cluster assignment

Next, probability of a cell belonging to a cluster $P(S_n|C_i)$ was calculated. Cells were assigned to a cluster based on a minimum threshold of P>0.99. Those cells with no $P(S_n|C_i)$ larger than the threshold were regarded as unassigned.

Handling of doublets

During scRNA-Seq experiments, a small proportion of droplets can contain cells from more than one sample. These so called doublets, contain cells from same or different samples sharing the same barcode, which if not addressed would cause bias. Our model took these doublets into consideration. During our hidden state based demultiplexing approach, we included an additional cluster so that doublets could be captured. To identify which cluster in the model was the doublet cluster in each round, the sum of log-likelihood of cross assignments was checked:

$$P(c \models doublet) = \sum_{i \notin c} \prod_v \left[P(C_{i,v}|S_{c,v})P(S_c)\right] \tag{6}$$

The sum log-likelihood of cells from all other clusters being assigned to a specific cluster was calculated for each cluster in turn and compared. The cluster with the largest sum log-likelihood of cross assignment was designated as the doublet cluster. We allow user input on the expected proportion of doublets. If the expected number of doublets was larger than those detected in the doublet cluster, cells with largest read depth were moved from singlet clusters to doublet cluster, so that the total number of doublets meet expectation as input.

Alternative allele Presence/Absence genotyping for clusters

To identify a minimum set of variants, which can distinguish between sample clusters, we generated alternative allele P/A genotype matrix (SNVs by clusters). To do that, sum of reference and alternate allele counts across all cells assigned to each cluster were calculated. And for each SNV and each cluster, "P" was marked if there were more than 10 alternative allele counts, and "A" for more than 10 reference allele counts but no alternative allele count. "NA" was set if neither criterion was met.

Mapping clusters back to individual samples using minimal set of P/A genotypes

Based on the P/A matrix, we started from informative SNVs which had variations of "P" or "A" across clusters and avoid picking those with "NA"s. Then, unique patterns involved in those SNVs were derived and for each unique P/A pattern, one allele was selected to subset the whole matrix. Next, Gram-Schmidt orthogonalization[11] was applied on the subset of P/A matrix, in order to find the variants which can be basis vectors to effectively distinguish the clusters. If not enough SNVs were found to distinguish all the clusters, the clusters were split into smaller groups so that for each group there was enough variants to distinguish the clusters within that group. And to distinguish clusters from different smaller groups, if the selected variants could not be used to distinguish any pair of clusters, additional variants were selected from the whole list of variants where no NAs were involved and P/A was different between the pair of clusters. Ideal situation was N variants for N clusters, but it was possible that >N variants were needed to distinguish N clusters.

As such, the P/A genotyping of each cluster, on the minimum set of distinguishing variants, could be used as a reference to map samples to clusters. After running genotyping on this minimum set of loci for each of the individual samples, a similar matrix based on sample genotypes could be generated, by setting the alternative presence flag when genotype probability (GP) was larger than 0.9 for RA or AA, or absence flag when GP was larger than 0.9 for RR. By comparing both P/A matrices, we could link the identified clusters in scSplit results to the actual individual samples.

In practice, samples can be genotyped only on the few distinguishing variants, so that scSplit-predicted clusters can be mapped with individual samples, while the whole genotyping is not needed. When the whole genotyping is available, we also provide an option for users to generate distinguishing variants only from variants with $R2 > 0.9$, so that they can compare the distinguishing matrix from scSplit with that from known genotypes on more confident variants.

Data simulation

To test the consistency of the model, and the performance of our demultiplexing tool, reference/alternative count matrices were simulated from a randomly selected scRNA-Seq BAM file and a multi-sample VCF file. We assume the randomly selected BAM file had a representative gene expression profile.

First, data quality was checked and the BAM and VCF files were filtered. Second, barcodes contained in the BAM file were randomly assigned to samples in the VCF file, which gave us the gold-standard of cell-sample assignments to check against after demultiplexing. Then, all the reads in the BAM file were processed, that if a read overlapped with any SNV position contained in the merged VCF file, its barcode was checked to get its assigned sample and the probability $P(A_{c,v})$ of having the alternative allele for that sample was calculated using the logarithm-transformed genotype likelihood (GL) or genotype probability (GP) contained in the VCF file. The probability of an allele being present at that position could then be derived so that the ALT/REF count at the SNV/barcode in the matrices could be simulated based on the alternative allele probability. Let $\mathscr{L}(AA)$ and $\mathscr{L}(RA)$ be the likelihood of seeing AA and RA of a certain cell c on a certain SNV v:

$$P(A_{c,v}) = \frac{1}{2}10^{[\log_{10}\mathscr{L}(RA)]} + 10^{[\log_{10}\mathscr{L}(AA)]} \tag{7}$$

Finally, doublets were simulated by merging randomly picked 3% barcodes with another 3% without overlapping in the matrix. This was repeated for every single read in the BAM file.

With the simulated allele fraction matrices, the barcodes were demultiplexed using scSplit and the results were compared with the original random barcode sample assignments to validate. Code for simulation is available in the Github repository (https://github.com/jon-xu/scSplit).

Single cell RNA-seq data used in testing scSplit

For table 2 and table 3, endometrial stromal cells cultured from 3 women and fibroblast cells cultured from 38 healthy donors over the age of 18 years respectively were run through the 10x Genomics Chromium 3' scRNA-seq protocol. The libraries were sequenced on the Illumina Nextseq 500. FASTQ files were generated and aligned to Homo sapiens GRCh38p10 using Cell Ranger. Individuals were genotyped prior to pooling using the Infinium PsychArray.

References

1. Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* (2015). Doi: 10.1016/j.cell.2015.05.002.
2. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8** (2017). Doi: 10.1038/ncomms14049.
3. Zhang, X. *et al.* Comparative analysis of droplet-based ultra-highthroughput single-cell rna-seq systems. *Mol. Cell* **73** (2019). 10.1016/j.molcel.2018.10.020.
4. Stoeckius, M. *et al.* Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. *Genome Biol.* **19**, 224 (2018).
5. Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods* **14**, 865–868 (2017).
6. Kang, H. M. *et al.* Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nat. Biotechnol.* **36**, 89 (2017). Doi: 10.1038/nbt.4042.
7. L.Wolock, S., Lopez, R. & M.Klein, A. Scrublet: Computational identification of cell doublets in single-cell transcriptomic data. *Cell Syst.* (2019). Doi: 10.1016/j.cels.2018.11.005.
8. S.McGinnis, C., M.Murrow, L. & J.Gartner, Z. Doubletfinder: Doublet detection in single-cell rna sequencing data using artificial nearest neighbors. *bioRxiv* (2019). Doi: 10.1016/j.cels.2019.03.003.
9. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using umap. *Nat. Biotechnol.* **37**, 38–44 (2019).
10. E, G. & Marth, G. Haplotype-based variant detection from short-read sequencing. *arXiv preprint* (2012). ArXiv:1207.3907 [q-bio.GN].
11. Cheney, W. & Kincaid, D. *Linear Algebra: Theory and Applications* (Jones and Barlett Publishers, Sudbury, Massachusetts, 2009).
12. Heger, A., Jacobs, K. & contributors. pysam: htslib interface for python.
13. Do, C. B. & Batzoglou, S. What is the expectation maximization algorithm? *Nat. Biotechnol.* **26**, 897–899 (2008). Doi: 10.1038/nbt1406.
14. Borodovsky, M. & Ekisheva, S. *Problems and Solutions in Biological Sequence Analysis* (Cambridge University Press, Sudbury, Massachusetts, 2006).

Author contributions statement

LC and CF initiated the project. LC and JX designed the algorithms. JX implemented the toolset in Python and tested it on multiple datasets. JP and BM provided sample datasets. QN, SM and AS preprocessed the sample datasets. JP, QN, GM, BM, SM, JC and SA participated in important discussions and provided useful suggestions on different issues. JX drafted the manuscript, LC, JP, GM, QN and JC reviewed and revised the manuscript.

Competing interests

The authors declare that they have no competing interests.

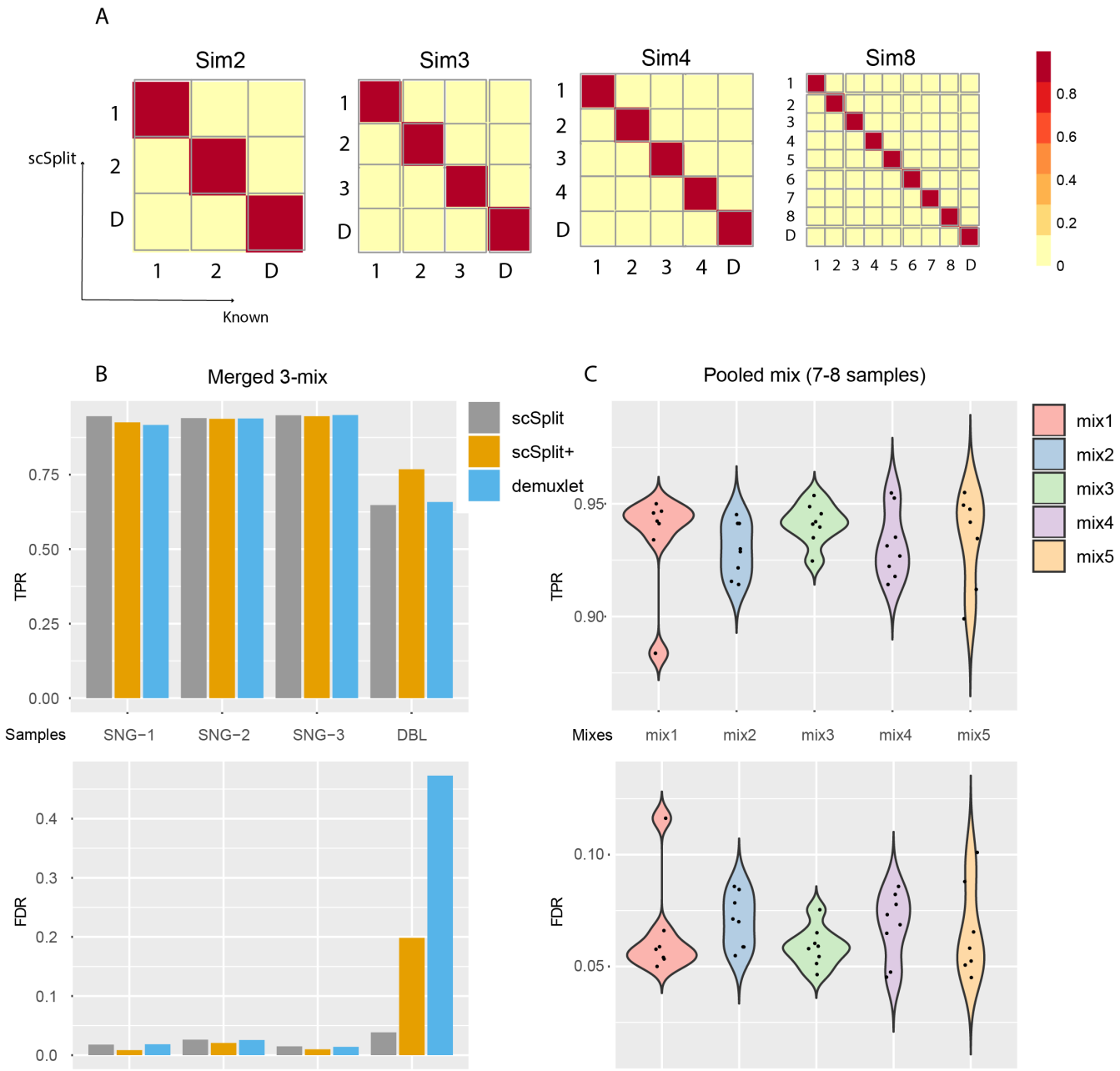


Figure 1. Results on simulated, merged, pooled RNA-Seq datasets confirmed scSplit a useful tool to demultiplex pooled single cells

(A) Confusion matrix showing scSplit demultiplexing results on simulated 2-, 3-, 4- and 8-mix; (B) TPR and FDR compared to known truth before merging for singlets and doublet clusters, among scSplit default, scSplit+ (scSplit run with doublet proportion expectation set to 5%), and demuxlet; (C) Violin plot of singlet TPR and FDR for five 7- or 8-mixed samples based on scSplit vs demuxlet.

mix1	scSplit							imputed vcf						
	1	2	3	4	5	6	7	11	28	53	110	181	195	326
10:135233840	1	0	0	1	0	1	0	0	1	0	0	1	1	0
13:111293915	0	0	1	0	0	1	0	0	0	0	0	0	1	1
17:46103760	0	0	1	0	1	1	0	0	0	0	1	0	1	1
1:151374025	0	1	0	0	1	1	1	1	0	1	1	0	1	0
1:240695255	0	1	1	0	1	0	0	0	0	1	1	0	0	1
4:166264119	0	0	1	0	1	0	1	1	0	0	1	0	0	1
4:1694809	0	1	0	1	0	1	0	0	0	1	0	1	1	0

mix2	scSplit								imputed vcf							
	0	1	2	4	5	6	7	8	104	114	137	145	161	184	212	253
12:57924050	0	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0
14:65542067	1	0	0	1	1	1	0	1	0	1	1	0	1	1	0	1
15:75890612	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	1
19:17430846	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	0
19:6751293	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0
2:228222293	1	1	0	0	1	0	1	0	0	0	0	1	0	1	1	1
2:3483205	0	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0
3:66428282	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0

mix3	scSplit								imputed vcf							
	1	2	3	4	5	6	7	8	77	78	2	31	169	43	124	320
10:3178881	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1
12:69995273	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1
14:71051088	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0
1:33789382	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0
2:99978048	1	0	1	1	0	0	1	0	1	0	1	0	1	1	0	0
3:29539770	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	0
6:143661166	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0
7:2290685	0	1	1	1	0	1	0	0	1	1	0	0	1	0	0	1

mix4	scSplit								imputed vcf							
	0	1	2	3	5	6	7	8	48	173	147	171	205	192	324	76
19:19287928	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	1
1:68170255	0	1	0	1	1	0	1	0	1	0	0	1	1	0	1	0
2:120059921	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0
2:26413817	0	0	0	0	1	0	1	1	1	0	0	1	0	1	0	0
3:52728804	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1	0
7:45115773	1	0	1	0	0	0	1	1	1	1	1	0	1	1	0	0
7:45139892	1	0	1	0	0	0	0	1	0	1	1	0	0	1	0	0
8:11700101	1	0	0	1	0	1	0	1	0	0	1	0	0	1	1	1

mix5	scSplit								imputed vcf							
	0	1	2	3	5	6	7	8	36	122	111	133	180	266	214	291
10:53021802	0	0	0	0	0	1		0	0	1	0	0	0	0	0	0
12:99019469	1	1	0	1	1	1	1	0	1	1	1	1	0	0	1	1
14:39651193	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1
15:81270754	0	1	0	0	0	0		1	1	0	0	0	0	1	0	0
17:4871518	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0
1:20977748	0		1	1	1	0		1		0	1	1	1	1	1	0
1:214837336	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
1:26497340		1	1	1	1	0		1	1	0	1	1	1	1	1	1
2:88326821	0	0	0	1	0	0		1	0	0	0	1	0	1	1	0

mix6	scSplit							imputed vcf						
	1	2	3	4	5	6	7	88	176	235	193	265	5	39
10:95104365	1	0	0	1	0	0	0	1	0	0	0	0	0	1
12:65858010	0	1	1	0	1	0	1	0	1	1	1	1	0	0
14:31647241	0	1	0	0	1	1	0	0	1	0	1	0	1	0
15:81166740	1	0	0	0	1	1	0	1	1	0	0	0	1	0
16:85741257	0	1	0	0	0	0	1	0	0	1	1	0	0	0
2:98372980	1	1	1	0	0	1	0	1	0	0	1	1	1	0
6:38086314	0	0	0	0	0	1	1	0	0	1	0	0	1	0

Figure 2. Illustration of presence absence matrices calculated on pooled scRNA-seq datasets. Within each rectangle we show the P/A genotype for each sample caclulate on scRNAseq data using scSpit, and the corresponding P/A genotype calculated from matched genotype files. The mapping of scRNAseq sample to genotype sample is indicated by colors.

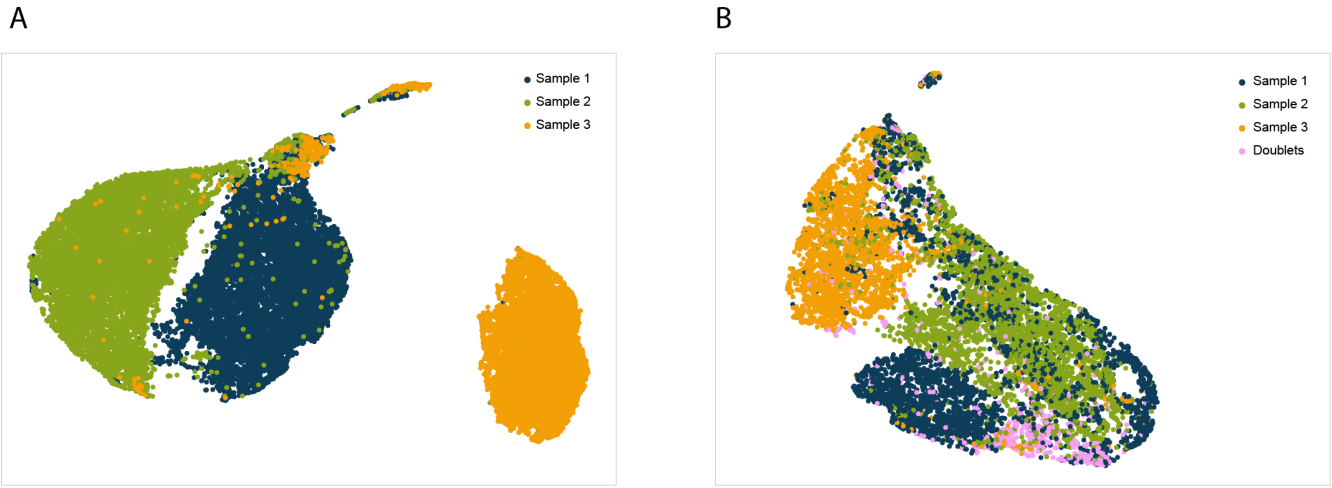


Figure 3. Batch effect during sequencing runs found in comparison of individually and pooled RNA-Seq data

(A) merged UMAP for three individually sequenced samples; (B) UMAP for pooled sequencing of same three individual samples, samples marked based on demultiplexing results using scSplit

Figure 4. The overall pipeline of scSplit tool

(A) SNV identified based on reads from all cells which have similar or different genotypes; (B) Alternative and reference allele count matrices built from each read in the pooled-sequenced BAM at the identified informative SNVs; (C) Initial allele fraction model constructed from the initial cell seeds and their allele counts; (D) Expectation-maximization process to find the most optimized allele fraction model, based on which the cells are assigned to clusters; (E) Presence/Absence matrix of alternative alleles generated from the cell assignments; (F) minimum set of distinguishing variants found to be used to map clusters with samples.

Simulation	sim2	sim3	sim4	sim8
Mixed samples	2	3	4	8
Number of cells	12,383	12,383	12,383	12,383
Reads per cell	1,041	1,041	1,041	1,041
Informative SNVs	34,116	34,116	34,116	34,116
Assign cells	41min	41min	46min	47min
Singlet TPR	0.97	0.97	0.97	0.97
Singlet FDR	0	9E-5	9E-5	9E-5
Doublet TPR	0.997	0.997	0.997	0.997
Doublet FDR	0	0	0	0

Table 1. Overview of accuracy and performance of scSplit on simulated mixed samples, with one CPU and 30GB RAM. We used PBMC donor B[2] and genotype data from demuxlet[6] as simulation templates

Predictions vs Origin	Overall
Mixed samples	3
Number of cells	9,567
Reads per cell	14,495
informative SNVs	63,129
Build matrices	67min
Assign cells	55min
TPR (scSplit)	0.91
FDR (scSplit)	0.04
TPR (scSplit+)	0.94
FDR (scSplit+)	0.01
TPR (demuxlet)	0.92
FDR (demuxlet)	0.05

Table 2. Comparison of scSplit, scSplit+ (scSplit run with doublet proportion expectation set to 5%) and demuxlet performance in demultiplexing merged individually genotyped samples (TPR: True Positive Rate; FDR: False Discovery Rate)

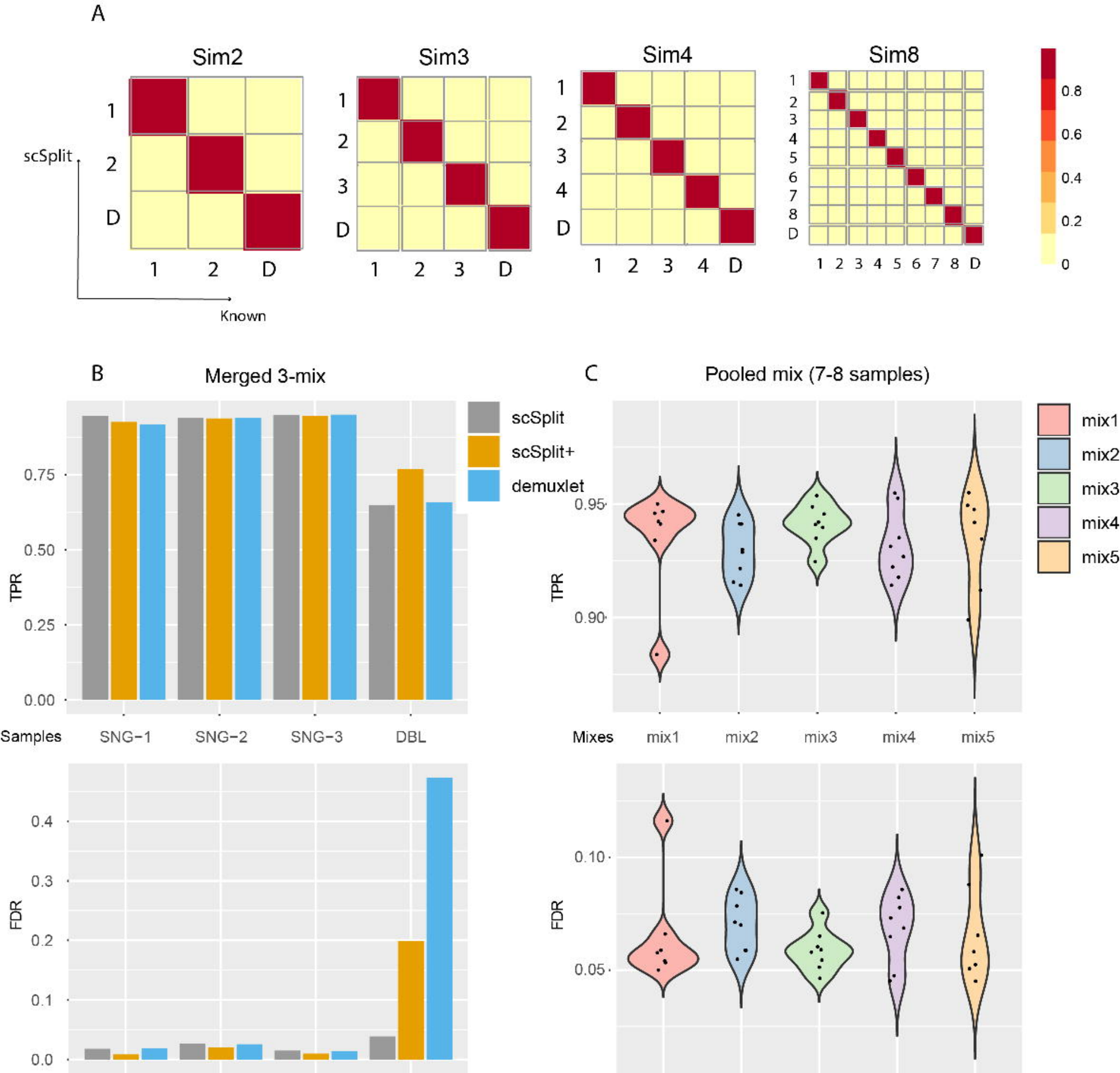
scSplit vs demuxlet	Mix 1	Mix 2	Mix 3	Mix 4	Mix 5
Mixed samples	7	8	8	8	7
Number of cells	914	8,137	5,165	6,977	7,428
Reads per cell	86,148	16,386	21,265	18,572	19,657
informative SNVs	15,848	26,830	26,162	23,224	41,993
Build matrices	10min	23min	18min	21min	35min
Assign cells	4min	47min	23min	45min	50min
Singlet TPR	0.94	0.93	0.94	0.93	0.93
Singlet FDR	0.06	0.07	0.06	0.07	0.07
Doublet TPR	0.52	0.17	0.15	0.17	0.08
Doublet FDR	0.48	0.83	0.85	0.83	0.92

Table 3. Overview of accuracy and performance running scSplit on five genotyped scRNA-Seq datasets, with one CPU and 30GB RAM. (TPR: True Positive Rate; FDR: False Discovery Rate)

Genotyping		Mix 1	Mix 2	Mix 3	Mix 4	Mix 5	merge1
scSplit	Accuracy	0.990	0.984	0.989	0.978	0.978	0.895
	N/A	93%	93%	93%	94%	92%	78%
demuxlet	Accuracy	0.964	0.983	0.908	0.955	0.983	0.913
	N/A	95%	97%	96%	95%	97%	83%

Table 4. Accuracy of alternative allele Presence/Absence genotypes built from scSplit/demuxlet clusters compared with that from sample genotyping, based on pooled scRNAseq dataset.

A
SNV identified in mixed sample
Reference Genome:
...GCATTACAGGCAATTCGTACGATCGTCCGACTACAATGGC...
CAATTCGTACGATCG
ACAGGCAATTCGTA
TTCGTACGATCGTC
CAATTTGTACGATCGTCC
TACAGGCAATTTGTAC
GCAATTTGTACGATCGTC
TACAGGCAATTCGTACG
TCGTACGATCGTCCGAC
CAGGCAATTCGTACGAT
TTTGTACGATCGTCCGA
ATTACAGGCAATTTGTA
reads from cell 1
reads from cell 2
reads from cell 3
reads from cell 4
B
Pooled BAM
barcode tags
alleles on identified SNV positions
REF counts
ALT counts
Informative SNVs P(RA) > 0.99 show allele heterogeneity among samples
C
REF counts
ALT counts
AF V1 V2 V3 V4 V5 ...
1 0.23 0.15 0.92 0.5 0.5 ...
2 0.82 0.73 0.34 0.21 0.08 ...
3 0.45 0.51 0.55 0.61 0.43 ...
4 0.11 0.21 0.15 0.05 0.25 ...
find most informative submatrices
K-mean
Seeds for each cluster
Initialize SNV/cluster Alternative Allele Fraction Model
D
Sample Allele Fractions
S0 S1 S2
SNV-1 0.50 0.57 0.49
SNV-2 0.50 0.36 0.28
SNV-3 0.50 0.11 0.71
Ref. alleles count
Alt. alleles count
barcode-1 barcode-2 barcode-3 ...
(3)
Maximization step
until convergence
(1)
Probability of cell belonging to samples P(S_n|C)
Probability of cell being observed from each sample P(C|S_n)
(2)
Bayesian theorem
Expectation step
E
REF counts
ALT counts
Count A/R alleles for each cluster
P: A>=10
A: A>=0 & R>=10
n/a: otherwise
F
Find distinguishing variants
RA/AA: P
RR: A
n=N
compare


A
Sim2
Sim3
Sim4
Sim8
scSplit
Known
1
2
D
3
4
5
6
7
8
0.8
0.6
0.4
0.2
0
B
Merged 3-mix
scSplit
scSplit+
demuxlet
TPR
0.75
0.50
0.25
0.00
Samples
SNG−1
SNG−2
SNG−3
DBL
FDR
0.4
0.3
0.2
0.1
0.0
C
Pooled mix (7-8 samples)
mix1
mix2
mix3
mix4
mix5
0.95
0.90
Mixes
0.10
0.05

mix1	sziplit								unpulsed vcf							
	1	2	3	4	5	6	7	8	11	28	53	110	161	195	237	
10:131283840	1	0	0	0	0	0	1	0	0	1	0	0	1	1	0	
13:111293913	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	
17:40103700	0	0	1	0	1	1	0	0	0	0	0	1	0	1	1	
1:151374021	0	1	0	0	1	1	1	1	1	0	1	1	0	1	0	
1:240095251	0	1	1	0	1	0	0	0	0	0	1	1	0	0	1	
4:160204119	0	0	1	0	1	0	1	1	1	0	0	1	0	0	1	
4:1084829	0	1	0	1	0	1	0	0	0	0	1	0	1	1	0	

mix2	sziplit								unpulsed vcf							
	0	1	2	3	4	5	6	7	8	104	114	132	145	161	180	212	253
12:57924050	0	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0
14:61542067	1	0	0	1	1	1	0	1	0	1	1	0	1	1	0	1
15:75890112	1	0	1	1	0	0	1	0	1	0	0	0	1	0	1	1
19:17480840	0	0	0	1	1	0	1	0	0	0	0	0	1	1	1	0
19:6751293	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0
2:128222293	1	1	0	1	0	0	1	0	0	0	0	1	0	1	1	1
2:3483205	0	1	0	0	1	1	0	0	0	0	1	1	0	0	0	0
3:10426252	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0

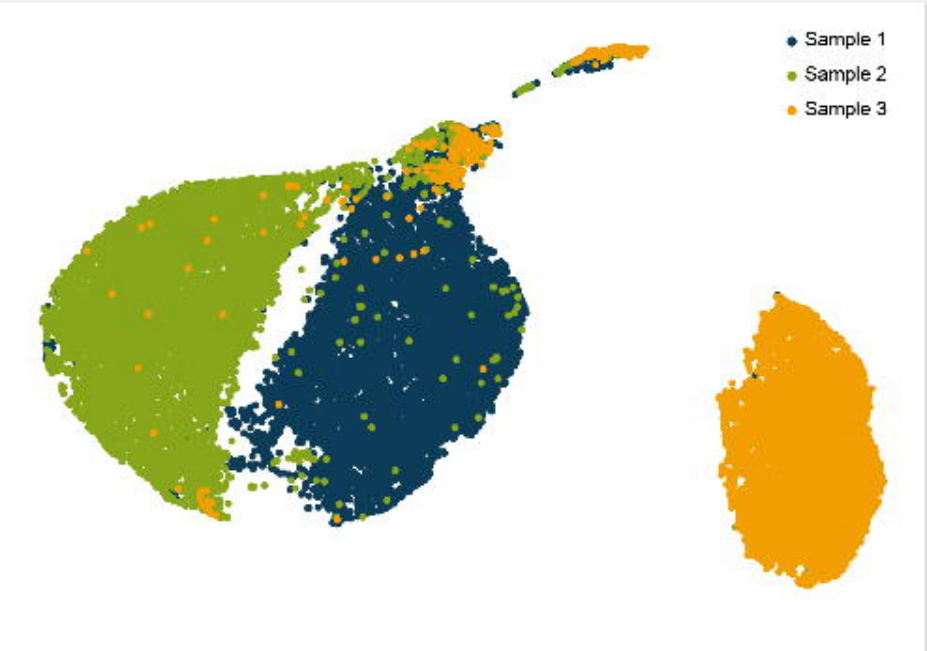
mix3	sziplit								unpulsed vcf							
	1	2	3	4	5	6	7	8	77	78	2	31	109	43	134	220
10:3178851	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1
12:6995273	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1
14:71011085	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0
1:83789352	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0
2:9973048	1	0	1	1	0	0	1	0	1	0	1	0	1	1	0	0
3:29138770	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	0
6:148601160	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0
7:2290685	0	1	1	1	0	1	0	0	1	1	0	0	1	0	0	1

mix4	sziplit								unpulsed vcf							
	0	1	2	3	4	5	6	7	8	46	173	187	191	203	189	204	23
19:19287925	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	1
1:28170255	0	1	0	1	1	0	1	0	1	0	0	1	1	0	1	0
2:120039921	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0
2:26418817	0	0	0	0	1	0	1	1	1	0	0	1	0	1	0	0
3:12728804	0	1	0	1	0	0	1	1	1	0	0	0	1	1	1	0
7:45117773	1	0	1	0	0	0	1	1	1	1	1	0	1	1	0	0
7:45139892	1	0	0	0	0	0	1	1	0	1	1	0	0	1	0	0
8:11700101	1	0	0	1	0	1	0	1	0	0	1	0	0	1	1	1

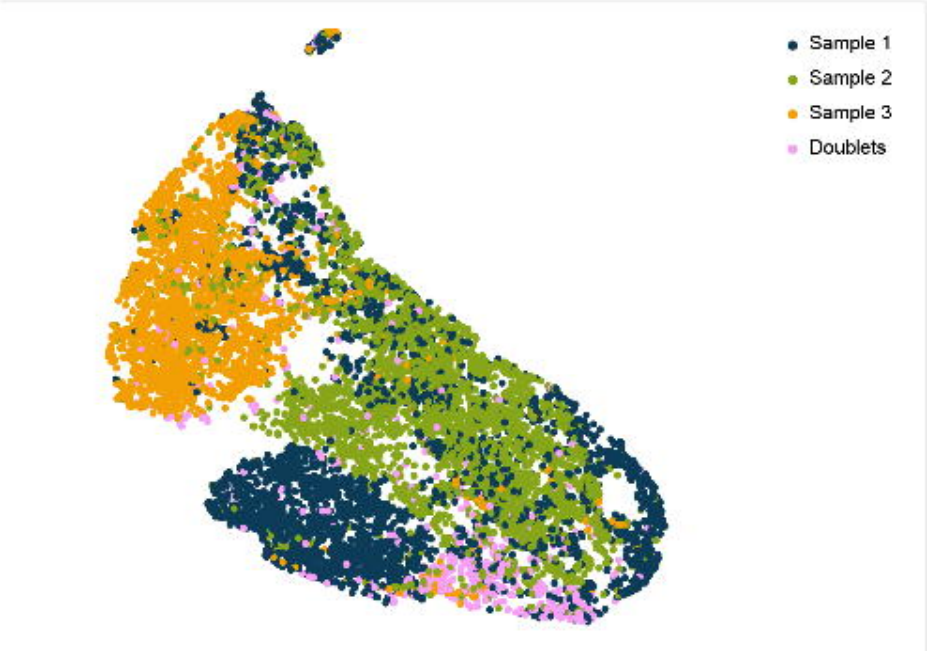
mix5	sziplit								unpulsed vcf							
	0	1	2	3	4	5	6	7	8	56	122	111	133	180	260	314	241
10:53021302	0	0	0	0	0	1		0	0	1	0	0	0	0	0	0
12:99019409	1	1	0	1	1	1	1	0	1	1	1	1	0	0	1	1
14:39011193	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1
15:81270754	0	1	0	0	0	0		1	1	0	0	0	0	1	0	0
17:4571518	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0
1:20977748	0		1	1	1	0		1		0	1	1	1	1	1	0
1:214837930	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
1:26487340		1	1	1	1	0		1	1	0	1	1	1	1	1	1
2:28820821	0	0	0	1	1	0	0	1	0	0	0	1	0	1	1	1

mix6	sziplit								unpulsed vcf							
	1	2	3	4	5	6	7	8	88	176	217	180	217	5	28	
10:91104861	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	
12:63838010	0	1	1	0	1	0	1	0	1	1	1	1	1	0	0	
14:31647241	0	1	0	0	1	1	0	0	0	1	0	1	0	1	0	
15:81100740	1	0	0	0	1	1	0	0	1	1	0	0	0	1	0	
16:81741257	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	
2:98872980	1	1	1	0	0	1	0	0	1	0	0	1	1	1	0	
6:88080314	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	

A


Sample 1
Sample 2
Sample 3

B


Sample 1
Sample 2
Sample 3
Doublets