RESEARCH

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

Jun Xu*, Caitlin Falconer and Lachlan Coin

*Correspondence:
jun.xu@uq.edu.au
Institute for Molecular Bioscience, The University of Queensland, 306 Carmody Road, St Lucia, QLD 4072 Brisbane, Australia
Full list of author information is available at the end of the article

Abstract

A variety of experimental and computational methods have been developed to demultiplex samples from different individuals mixed in a single-cell RNA sequencing (scRNA-seq) experiment. However, these methods all require extra information is either added to samples (such as sample barcode) or measured from samples prior to mixing (such as genome-wide genotypes). We introduce an alternative approach, in which genetic differences between mixed samples are inferred directly from scRNAseq data without extra information and these differences are used to assign single cells to samples. Our method also identifies a minimal set of presence/absence genotypes which can be used to map samples to their biological source, or to track samples between experiments. We tested our tool "scSplit" on different real and simulated datasets and achieved high true positive rate ($> 90\%$) in mapping cells back to their original source. Doublets can also be sensitively detected as an independent group. Our method is ideally suited to samples for which external genome-wide genotype data cannot be obtained (for example for 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.

Keywords: scSplit; scRNA; Demultiplexing; Hidden State Model; Expectation-Maximization; Genotype-free; Allele Fraction; Doublet; Gram-Schmidt Orthogonalization

Background

With its high resolution in analyzing genetic expression at the individual cell level, single-cell RNA sequencing (scRNA-seq) has recently become a popular tool in biological research. Studying cell biology at the single cell level provides much higher detail and resolution than previous bulk level analyses. As an 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. However, the per cell cost for running scRNA-seq remains high. Methods which could lower the per sample cost of running scRNA-seq are required so this new technique can be more widely adopted. An effective method for lowering scRNA-seq cost is by multiplexing samples to run as a single batch, and then sequencing results can be demultiplexed using various techniques.

Recently published experimental methods like SPLiT-Seq [1] subject cells to a combinatorial indexing procedure by randomly arranging multiple samples into different

wells with different barcodes over a number of rounds to achieve unique cell barcodes and which also contain the cells' sample of origin information. After sequencing, the scRNA-seq data can be demultiplexed according to their original samples. Although such a method can achieve lower per cell scRNA-seq cost, this approach adds experimental complexity potentially introducing additional sources of technical errors and is associated with lower numbers of reads per cell than for other scRNA-seq methods. Other methods like Cite-seq [2] can similarly derive sample origin in the sequencing result based on the barcodes used for different samples, but still makes the assumption that original samples are available before mixture.

Alternatively, computational tools like demuxlet [3] have also been developed to demultiplex mixed samples although this currently requires additional genotyping information to assign individual cells back to their samples of origin. However, genotyping arrays might not be always available for different species and it might not work properly if the genetic differences between samples are somatic in origin.

Here we introduce a simple and efficient tool, "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 prior to demultiplexing which also makes it suitable for applications where genotypes are unavailable or difficult to obtain.

Single nucleotide variants (SNVs) are one of the main types of human sequence variation [4]. Being conservative and informative, SNV detection is a popular tool in genetic analyses. In the mixed samples, SNVs can be detected using common SNV calling tools and the presence of alternative alleles at each genomic location may indicate potential differences between the samples. scSplit utilizes differing allele count distributions across informative heterozygous SNV sites to distinguish the different cell sample origins.

To achieve this, the model states are initialized with allele fractions on informative SNV positions and each cell is then compared with the states in the model to derive a probability of a certain cell belonging to a certain state. Next, weighted allele counts at each SNV position for each cell are distributed to each model state based on the cell-state probability. The allele fraction model is then re-calculated based on the newly distributed allele counts. Once overall model likelihood converges after a number of learning iterations the cells can be assigned to the final states in the model. Details are described in Methods.

Methods

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

Overview

Testing on real data was conducted following the below pipeline (Figure 1):

1. Data quality control and filtering: The mixed sample BAM file is first filtered to keep only the reads with white listed barcodes to reduce technical noise. Additional filtered 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: Freebayes v1.2 [5] 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: The "build_matrices.py" script is run which produces two .csv files, one for each of reference and alternate allele counts as output.

4. Execution and verification of demultiplexing: The two allele count matrices are used to demultiplex the cells into their original sample assignments. The output produces n+3 files for n input samples: n .csv files listing the barcodes assigned to each n samples, one file capturing doublet barcodes, one file recording the final allele fraction model and a file recording the probability of each cell belonging to each of the n samples in the model.

5. Distinguishing variants for all samples: To more easily assign each model state back to the specific sample, distinguishing variants are identified so that genotyping of the least number of loci using the Sequenom platform may be performed. Gram-Schmidt orthogonalization [6] is used to achieve this.

6. Output of genotype likelihoods for model states: A VCF file is generated for the logarithm-transformed genotype likelihoods for all sample models.

Results were validated by comparing cell state assignments with known cell sample origins.

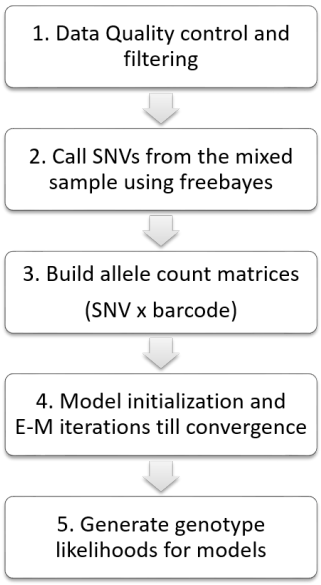
Data quality control

Samtools is 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 are removed (samtools rmdup [input] [output]) followed by sorting and indexing.

SNV calling on scRNA-seq dataset

SNVs are called on the scRNA-seq mixed sample BAM file with freebayes [5], a widely used variant calling tool. The freebayes arguments "-iXu -q 1" are set to ignore indels and MNPs and exclude alleles with supporting base quality scores of less than one. This generates a VCF file containing all SNVs from the mixed sample BAM file.

Figure 1 Overall pipeline of scSplit. The overall pipeline of scSplit tool includes five major steps.



Building allele count matrices

Allele count matrices are 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 are 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 indicates either the number of reference or alternate alleles detected in one cell barcode at that specific SNV position. This provides a full map of the distribution of reference and alternate alleles across all barcodes at each SNV.

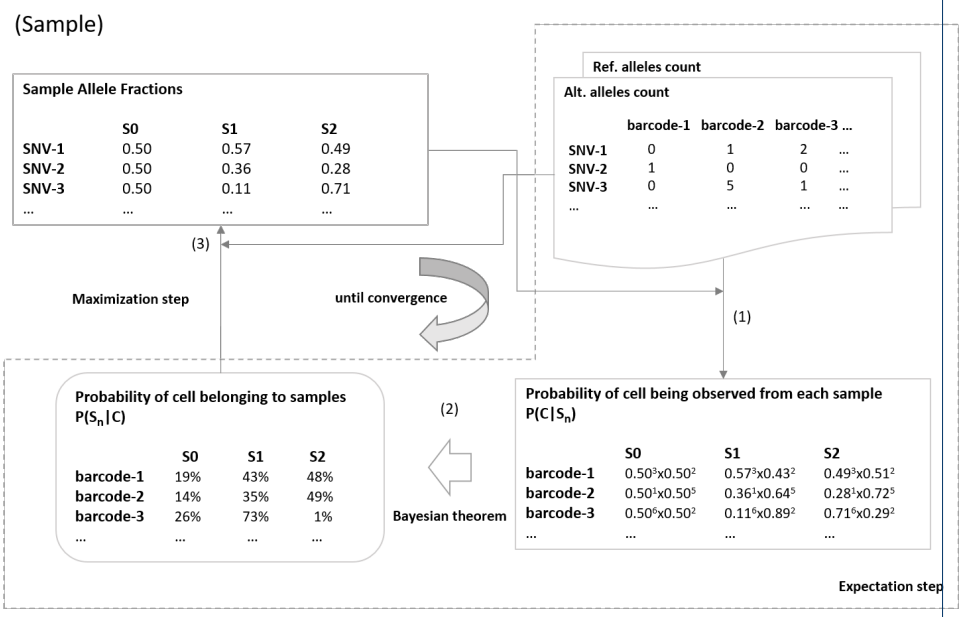
The allele count matrices capture information from all reads at all SNVs that reflects the different allele fraction patterns from different barcodes or samples. To build the allele count matrices, pysam fetch [7] is used to extract reads from the BAM file. The reads overlapping each SNV position are 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) is lower than $\log_{10}(1-error)$ where error $= 0.01$ are filtered out. These are more homozygous and thus less informative for detecting the differences between the multiple samples. The generated matrices are exported to csv files for further processing.

Demultiplexing

Using the built reference and alternative allele counts matrices across all SNVs and barcodes as input, the allele fraction model is initialized using selected seed barcodes by PCA and K-means clustering on informative barcodes and SNVs, followed by

iterations of the Expectation-Maximization (E-M) approach until maximum likelihood converges. In the E-step, the cell-sample likelihoods given the allele fraction model is calculated, while in the M-step, the allele fraction model is updated using the newly calculate cell-sample likelihoods. (Figure 2). This process is repeated for 30 rounds introducing randomness at each initialization and taking the round with the largest overall log likelihood as the final result.

Figure 2 EM algorithm Expectation-Maximization approach to find the best cell assignment.



Identification of maximally informative cluster representatives for model initialisation

To initialize the model, we need initial probabilities of observing an alternative allele on each SNV position in each state. To avoid bias caused by noise in scRNA-seq data, we first look for cells and SNVs with the minimum number of zero allele counts on all SNVs and in all cells. Cells are sorted according to their number of zero allele counts (sum of reference and alternative alleles) at all SNVs and SNVs are similarly sorted according to their number of zero allele counts (sum of reference and alternative alleles) across all cells. Next, we randomly select and filter 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 have zero counts. This is repeated until all remaining cells have more than 90% of their SNVs with non-zero allele counts and all SNVs have non-zero counts in more than 90% of cells. This subset of matrices will serve as the basis for the seed barcodes to initialize the whole model. The data is transformed using PCA with 20 dimensions and then K-means is performed to split the cell subset into the expected number of states, and using

the allele fractions on the subset of SNVs in these selected cells, each state of the model is initialized according to below formula:

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

Expectation–maximization approach

We then run E-M machine learning iterations. Each iteration starts with an E-step that calculates the cell-sample likelihood based on the given allele fraction model, and ends with an M-step to recalculate the allele fraction model based on the most updated cell sample relationship. During the E-step, we first calculate $P(C_i|S_n)$,

the likelihood of observing a cell C_i in sample S_n, which is equal to the product of the probability of observing the allele fraction pattern over each SNV, which in turn equals the product of probability of having observed the count of alternative alleles and probability of having observed the count of reference alleles:

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

And then transform $P(C_i|S_n)$ 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 ($P(S_1) = P(S_2) = ... = P(S_n)$):

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

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

$$P(A_v|S_n) = \frac{\sum_i N(A_{c,v})P(S_n|C_i)}{\sum_i N(T_{c,v})P(S_n|C_i)}$$

The overall log-likelihood of the whole model is calculated as:

$$\mathcal{L}_{model} = \sum_n \sum_i \log(P(C_i|S_n))$$

The E-M iterations continue until sum of log-likelihood reaches convergence.

Cell barcode assignment
After convergence of the E-M iterations, cells are assigned to the singlet or doublet states based on their probability of originating from each state $P(S_n|C_i)$. Cells are assigned to a state based on a minimum threshold of P>0.9. Those cells with no $P(S_n|C_i)$ larger than the threshold are regarded as unassigned.

Multiple runs to avoid local maximum likelihood
The entire process is repeated for 30 rounds with the addition of randomness during model initialization and the round with the largest sum of log likelihood is taken as the final result. Randomness is 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.

Handling of doublets
During scRNA-seq experiments, it is unavoidable for a small proportion of droplets to contain cells from more than one sample and form so called doublets. Our model takes these doublets into consideration. During our hidden state based demultiplexing approach, we have added an additional state so that doublets can be captured. To identify which state in the model is the doublet state in each round, the sum of log-likelihood of cross assignments is checked. The sum log-likelihood of cells from all other states being assigned a specific state is calculated for each state in turn and compared. The state with the largest sum log-likelihood of cross assignment is designated as the doublet state.

Presence/absence genotyping
To identify distinguishing variants, first the sum of reference and alternate allele counts across all cells assigned to each state are calculated. Then for each SNV in each state, the alternative allele is marked 'present' if there are more than 2 alternative allele counts, and 'absent', if there are more than 5 reference allele counts but no alternative allele count, or 'NA' if neither criteria is met.

Identifying minimal discriminatory set of presence/absence genotypes
By applying Gram-Schmidt orthogonalization [6] on the presence/absence genotyping, we can find the fewest number of variants that can be used to distinguish each state. We use the alternative allele presence/absence matrices for each state on the minimum set of distinguishing variants. After running genotyping on these minimum set of loci for each of the individual samples, we can generate a similar matrix based on sample genotypes, by setting the presence flag when genotype probability (GP) is larger than 0.9 for AR or AA, or absence flag when GP is larger than 0.9 for RR. By comparing the two matrices, we can map the identified clusters in scSplit results with the actual individual samples.

Simulation studies
To better test the performance and limit of our demultiplexing tool, we simulate BAM files and reference/alternative count matrices from a randomly selected scRNA-seq BAM file and a multi-sample VCF file. We assume the randomly selected BAM file has a representative gene expression profile.

First, we check data quality and filter the BAM file and VCF file. Second, we randomly assign barcodes contained in the BAM file to samples in the VCF file, and this also gives us the expected 'true' cell-sample assignments to compare with after testing. Then, we read through all the reads in the BAM file. If a read is overlapping with any SNV position contained in the multi-sample VCF file, we check its barcode to get its "sample" and calculate the probability $P(A_{c,v})$ of having the alternative allele using the logarithm-transformed genotype likelihood contained in the VCF file for that sample. We can then determine the probability of an allele being present at that position and use that probability to update for the presence of an allele in that read and assign a count in the appropriate SNV/barcode allele matrix. This is repeated for all reads in the BAM file.

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

With the simulated BAM file and allele fraction matrices, we demultiplex the barcodes using scSplit and compare the results with the original random barcode sample assignments to validate.

Results

(Result data are available at `https://github.com/jon-xu/sc_split_paper_data`)

Higher than 99% accuracy achieved splitting mixed BAM files from real samples

We used publicly available scRNA-seq data from Zheng et al. [8] to test our scSplit tool. The dataset included scRNA-seq results of frozen PBMCs from three donors A (2,900 cells), B (7,783 cells) and C (9,519 cells).

We first tried to split donor A from donor C. For that, we simply merged both BAM files. After data cleansing, we had 122,933 SNVs based on Homo sapiens (human) genome assembly GRCh37 (hg19) using freebayes v1.2 [5]. Based on that we built the reference and alternative allele count matrices, each with 122,933 SNVs and 12,419 barcodes.

Then we ran scSplit, in non-doublet mode, using the two matrices as input, and the result showed us a very high accuracy of demultiplexing the two samples - 99.1% of the cells were correctly assigned to their original samples, 0.5% were identified as doublets, and 0.4% remained unassigned.

Next we tested on a three sample mixed BAM file, which gave us 187,694 SNVs and 20,202 cells. This time we kept 88 doublets in the mixed sample. After running through the pipeline, we achieved 99.2% accuracy in sample assignment, and detected 85.2% of the doublets, with 0.4% cells unassigned (Table 1).

Table 1 Overview of accuracy and performance of scSplit on mixed sequencing data, using high performance cluster with one CPU and enough memory assigned to each job. With 2-sample, 12,419 cells from PBMC donor A and C were mixed, 3-sample involved 20,202 cells from PBMC donor A, B and C

mixed BAM	2-sample	3-sample
Number of cells	12,419	20,202
Reads per Cell	4,958	4,870
informative SNVs	122,933	187,694
Build matrices	1h21m	4h25m
Assign cells	3h59m	11h38m
True Positive rate (singlets)	100.0%	99.1%
distinguishing variants	1:564667	1:564667
	1:135040	1:1246831
		1:879911

Utilising scSplit to demultiplex mixed samples from scRNA-seq data, and compare it with demuxlet [3] predictions based on sample genotypes

We also tried our tool on a set of genotyped and scRNA-seqed mixed-sample datasets (available on request and will be publicly available by end of 2019). Since the datasets were genotyped, we were able to compare our results with demuxlet predictions. The true positive rates of our results are between 85-100%, if we treat demuxlet prediction as the gold-standard (Table 2)

Table 2 Overview of accuracy and performance running scSplit on seven genotyped scRNA-seq datasets, using high performance cluster with one CPU and enough memory assigned to each job.

pooled scRNA	7-sample	8-sample	8-sample	8-sample
Number of cells	914	8,093	5,153	6,939
Reads per Cell	4,086	5,295	7,033	5,364
informative SNVs	15,848	26,830	26,162	23,224
Build matrices	10min	23min	18min	21min
Assign cells	4min	47min	23min	45min
True positive rate *	93.0%	89.3%	85.8%	90.4%
Doublet detected	yes	yes	yes	yes
distinguishing variants	13:114204369	12:120901266	11:110333608	14:24649669
	12:21622051	12:124105925	7:34193952	1:89401613
	12:69995273	10:127950608	12:123806219	7:83590231
	7:150910744	18:12985600	10:90774772	12:124105839
	17:79648527	11:123030845	10:88722865	10:65135137
	16:75690279	2:3502625	3:29689912	7:140706031
	7:158523888	12:48541757	10:121437012	14:20757518
		10:128596257	1:19585207	21:47519274

pooled scRNA	8-sample	7-sample	3-sample
Number of cells	5,671	7,394	9,809
Reads per Cell	11,493	7,361	8,347
informative SNVs	34,103	41,993	61,356
Build matrices	37min	35min	68min
Assign cells	30min	50min	
True positive rate *	93.7%	89.7%	100.0%
Doublet detected	yes	yes	yes
distinguishing variants	10:53021802	15:52315122	1:569591
	11:20139937	10:9510436	1:12474945
	1:120492164	14:74527190	1:564520
	2:177520819	11:102713447	
	15:42190692	12:10854625	
	22:39414766	12:70655240	
	14:56146357	6:144289022	
	1:42965592		

* compared to demuxlet predictions.

Simulation run shows high accuracy and efficiency of scSplit

Finally, we used the mixed BAM file containing two donors (A and C) from the above mentioned data from Zheng et al. [8] as a template for simulation. Additionally, we took eight unrelated samples (1043, 1249, 1511, 1493, 1598, 1085, 1079, 1154) of 32 samples in the merged VCF file, from figure 2 supplementary data from Kang et al. [3], as the source of multi-sample genotype likelihoods for simulation (see "Simulation studies" in Methods). We ran simulation tests using our scSplit tool and got almost 100% true positive rates for two-, three-, four- and eight-mixed samples, as shown in Table 3

Table 3 Overview of accuracy and performance of scSplit on simulated mixed samples, using high performance cluster with one CPU and enough memory assigned to each job. We used PBMC donor B as BAM template, and 32-sample merged VCF provided by demuxlet [3]

Simulation	2-sample	3-sample	4-sample	8-sample
Number of cells	12,383	12,383	7,783	7,783
Reads per Cell	177	142	155	154
informative SNVs	34,116	34,116	34,116	34,116
Simulate matrices	2h26m	1h37m	1h15m	1h10m
Assign cells	41min	41min	46min	47min
True positive rate	99.3%	99.9%	99.7%	99.9%
Doublet detected	yes	yes	yes	yes
distinguishing variants	1:10521158	1:10521158	1:23884661	14:51722690
	1:236713828	1:46112218	1:26901319	19:50162909
		10:12165038	1:26607726	10:3178815
			1:16162092	11:110333762
				3:67425965
				22:42908743
				1:26607726
				21:46269710

Distinguishing variants found and validated to be useful to find sample origin

We generated distinguishing variants for all datasets using our tool (Tables: 1, 2, 3). We then checked these distinguishing variants against the VCF file containing sample genotypes and extracted the alternative allele presence/absence matrix at each distinguishing variant for each sample, and compared this to the sample allele matrices generated from the scSplit cell assignments. Taking one of the 8-sample mixes containing 8,093 cells (second sample in table 2 as an example, there was a one-to-one allele presence/absence mapping between the two matrices, based on which we assigned our clusters from scSplit to samples in VCF file (Figure 3). Now that we know which cluster belongs to which sample in our test datasets, we checked the true positive rate (True positive cell assignments / Total number of cells) for each cluster-to-sample mapping and they are between 85% and 95% (Table 2. In future cases where no sample genotypes are available, it is affordable to genotype the samples using the distinguishing variants we found in scSplit and use them as a reference to map the scSplit clusters to the actual samples.

Discussion

We developed the scSplit toolset to facilitate accurate, cheap and fast splitting of mixed scRNA samples, without needing prior genotype information from the

Figure 3 Assign clusters to samples Assign clusters to samples using Alternative allele presence/absence matrix in one of tested datasets

Clusters in scSplit	0	1	2	3	4	5	6	8
12:120901266	0	1	0	0	0	0	0	1
12:124105925	0	1	0	0	1	0	1	0
10:127950608	0	0	1	0	0	0	1	1
18:12985600	1	0	0	0	1	0	0	1
11:123030845	0	0	0	1	0	0	0	1
2:3502625	1	0	1	0	0	0	0	0
12:48541757	0	0	1	1	0	1	0	0
10:128596257	1	0	0	0	0	1	1	0

Samples in vcf	A	B	C	D	E	F	G	H
12:120901266	1	0	0	0	0	1	0	0
12:124105925	0	1	0	1	0	1	0	0
10:127950608	1	0	0	1	0	0	0	1
18:12985600	1	1	1	0	0	0	0	0
11:123030845	1	0	NA	0	1	0	0	0
2:3502625	0	0	1	0	0	0	0	1
12:48541757	0	0	0	0	1	0	1	1
10:128596257	0	0	1	1	0	0	1	0

1: high confidence for Alternative allele
0: high confidence for Reference allele
NA: neither A or R allele is of high confidence

individual samples. ScSplit can also generate a minimum set of alleles (as few as the sample numbers), so that researchers can link the resulting clusters with the actual samples, by comparing the allele presence at these distinguishing variants.

We filtered out MNPs when building the model and the results show SNVs provide adequate information to capture the difference between multiple samples. Besides using allele fractions to model multiple samples, we could also use genotype likelihoods for same purpose, however, more memory and running time would be expected, especially when barcode numbers in mixed sample experiments increase. And no higher accuracy has been observed based on our tests.

The current version of scSplit assumes that samples are equally mixed, and the number of mixed samples are known. It is possible to run scSplit tool for different numbers of samples and compare the model log-likelihoods to select the most likely or optimal number of samples that is being modelled, but this would require significant computational resources and time. Instead, the reference and alternative allele counts in different samples and the size the doublet state would also be informative for detecting an optimised sample number. Further study will be needed to implement these.

We also expect the application of scSplit in cancer related studies, to distinguish tumor cells from healthy cells.

Although scSplit is 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.

Conclusions

With all the tests we have conducted, we are confident that scSplit is an accurate and efficient tool to conduct genotype-free demultiplexing based on scRNA-seq datasets from mixed samples. We believe this tool would be helpful in different biological or medical research areas, including but not limited to mixed infections, distinguishing tumor subclones and working in non-model organisms.

Competing interests

The authors declare that they have no competing interests.

Authors

Jun Xu, Genome Innovation Hub, the University of Queensland, St Lucia, QLD 4072, Australia Caitlin Falconer, School of Chemistry and Molecular Biology, the University of Queensland, St Lucia, QLD 4072, Australia Lachlan Coin, Institute for Molecular Bioscience, the University of Queensland, St Lucia, QLD 4072, Australia
Correspondence: jun.xu@uq.edu.au

References

1. Rosenberg, A.B.e.a.: Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. Science (2018). doi: 10.1126/science.aam8999
2. Stoeckius M, e.a.: Simultaneous epitope and transcriptome measurement in single cells. Nature Methods **14**, 865–868 (2017)
3. Kang, H.M.e.a.: Multiplexed droplet single-cell rna-sequencing using natural genetic variation. Nature Biotechnology **36**, 89 (2017). doi: 10.1038/nbt.4042
4. Group, T.I.S.M.W.: A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. Nature **409**, 928–933 (2001)
5. Garrison E, M.G.: Haplotype-based variant detection from short-read sequencing. arXiv preprint (2012). arXiv:1207.3907 [q-bio.GN]
6. Cheney, W., Kincaid, D.: Linear Algebra: Theory and Applications. Jones and Barlett Publishers, Sudbury, Massachusetts (2009)
7. Heger, e.a. Andreas: Pysam: Htslib Interface for Python. `https://github.com/pysam-developers/pysam`
8. Zheng, G.X.Y.e.a.: Massively parallel digital transcriptional profiling of single cells. Nature Communications **8** (2017). doi: 10.1038/ncomms14049