Bioinformatics, 37(23), 2021, 4569–4571
doi: 10.1093/bioinformatics/btab358
Advance Access Publication Date: 8 May 2021
Applications Note

Sequence analysis

Cellsnp-lite: an efficient tool for genotyping single cells

Xianjie Huang [1] **and Yuanhua Huang**[1,2,*]

[1]School of Biomedical Sciences, LKS Faculty of Medicine, University of Hong Kong, Hong Kong, China and [2]Department of Statistics and Actuarial Science, University of Hong Kong, Hong Kong, China

*To whom correspondence should be addressed.

Associate Editor: Can Alkan

Received on December 31, 2020; revised on April 12, 2021; editorial decision on May 3, 2021; accepted on May 7, 2021

Abstract

Summary: Single-cell sequencing is an increasingly used technology and has promising applications in basic research and clinical translations. However, genotyping methods developed for bulk sequencing data have not been well adapted for single-cell data, in terms of both computational parallelization and simplified user interface. Here, we introduce a software, cellsnp-lite, implemented in C/C++ and based on well-supported package htslib, for genotyping in single-cell sequencing data for both droplet and well-based platforms. On various experimental datasets, it shows substantial improvement in computational speed and memory efficiency with retaining highly concordant results compared to existing methods. Cellsnp-lite, therefore, lightens the genetic analysis for increasingly large single-cell data.

Availability and implementation: The source code is freely available at https://github.com/single-cell-genetics/cellsnp-lite.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Contact: yuanhua@hku.hk

1 Introduction

Single-cell sequencing has become a powerful technology for disentangling heterogeneity in cell populations at different levels, including genetics, transcriptome and epigenetics, hence has profound implications in basic research and clinical translations. The cellular genotype was primarily studied by single-cell DNA-seq (scDNA-seq) for detecting somatic mutations in tumours, clustering cells into clones and inferring their evolutionary dynamics (Navin, 2014). Recently, more evidence has been found that a subset of somatic mutations can also be observed in other single-cell probes, including scATAC-seq and full-length scRNA-seq (e.g. SMART-seq2), at both nuclear (McCarthy *et al.*, 2020) and mitochondrial genomes (Ludwig *et al.*, 2019). On the other hand, germline variants (a.k.a., single nucleotide polymorphisms, SNPs) are more widely observed in single-cell sequencing data, even in shallow droplet-based platforms, e.g. 10× Genomics, thanks to the large candidate list [around 7 million SNPs in human population with frequency $> 5\%$ (1000 Genomes Project Consortium, 2015)]. Germline SNPs are not only perfect natural barcodes when multiplexing cells from multiple individuals (Huang *et al.*, 2019), but also important in implying functional regulation via cellular eQTL analysis or allele specific expression (Cuomo *et al.*, 2020) and allelic imbalance caused by copy number variation (Fan *et al.*, 2018; Zaccaria and Raphael, 2021).

Genotyping methods for bulk sequencing samples are nearly mature with a decade of efforts and many methods remain effective when applying to single-cell sequencing data (Liu *et al.*, 2019), including the successful BCFtools (Li *et al.*, 2009; Li, 2011). However, there is a lack of good adaption of these methods for single-cell data in terms of computational parallelization and simplified user interface. Here, we develop cellsnp-lite, a htslib-based (Li *et al.*, 2009) tool for genotyping in single cells. Htslib is a well-developed, optimized and maintained package and is the core library used by BCFtools, hence we expect cellsnp-lite to give comparable accuracy as BCFtools in genotyping but higher efficiency and better convenience for single-cell data.

The goal of cellsnp-lite is to provide a user-friendly command-line interface, with achieving high efficiency in both speed and memory. Therefore, it is designed as a light way allelic reads pileup with minimum filtering by keeping most data for customized downstream filtering and/or statistical modelling. We expect cellsnp-lite to be convenient in intermediate processing, e.g. for allelic ratio in copy number variations, and immediately useful for coarse analysis in less sensitive situations, e.g. for sample swap check.

2 Implementation

Cellsnp-lite is implemented in C/C++ and performs per cell genotyping, supporting both with (mode 1) and without (mode 2) given SNPs. In the latter case, heterozygous SNPs will be detected automatically. Cellsnp-lite is applicable for both droplet-based (e.g. 10× Genomics data) and well-based platforms (e.g. SMART-seq2 data).

Table 1. Cellsnp-lite genotype options and example alternatives

Mode	SNPs	Bam files	Platform	Alternative
Mode 1a	Given	Pooled one	Droplet	VarTrix
Mode 1b	Given	Each per cell	SMART-seq	BCFtools mpileup
Mode 2a	To detect	Pooled one	Droplet	N.A.
Mode 2b	To detect	Each per cell	SMART-seq	Freebayes

Note: the two-step approach Mode 2b + 1a is an internal alternative to mode 2a.

See Table 1 for a summary of these four options, and example alternatives in each mode.

Cellsnp-lite requires aligned reads as input, in bam/sam/cram file formats. Cell labels can be coded in the cell tag in a multiplexed bam file (droplet-based platforms) or specified by each per-cell bam file (well-based platforms). This flexibility also allows cellsnp-lite to work seamlessly on bulk samples, e.g. bulk RNA-seq, by simply treating it as a well-based 'cell'.

The pileup is performed per genome position, either for given SNPs (mode 1) or the whole chromosome (i.e. mode 2). All reads covering a query position will be fetched. By default, we discard those reads with low alignment quality, including MAPQ < 20, aligned length < 30 nt and FLAG with UNMAP, SECONDARY, QCFAIL (and DUP if UMI is not applicable). We then assign all these reads into each cell by hashmap for droplet-based sample (mode 1a or 2a) or direct assignment for well-based cells (mode 1b or 2b). Within each cell, we count the UMIs (if exist) or reads for all A, C, G, T, N bases. The REF and ALT alleles are taken from the input SNPs if given (i.e. in mode 1) otherwise by selecting the base with the highest count as REF and the second highest as ALT (mode 2).

When SNPs are given (mode 1), cellsnp-lite will perform parallel computing by splitting the input SNPs in-order and equally into multiple threads. Otherwise, in mode 2, cellsnp-lite will compute in parallel by splitting the listed chromosomes, with each thread for one chromosome.

In all the above scenarios, cellsnp-lite outputs sparse matrices for alternative allele, depths (i.e. REF and ALT alleles), and other alleles. If adding argument '—genotype', cellsnp-lite will perform genotyping with the error model as presented in Table 1 in (Jun *et al.*, 2012) and output in VCF format with cells as samples.

3 Performance

3.1 High accuracy in pileup allelic counts

As discussed above, we aim for a light way pileup of allelic counts in large single-cell sequencing data. Hence, we mainly compared the

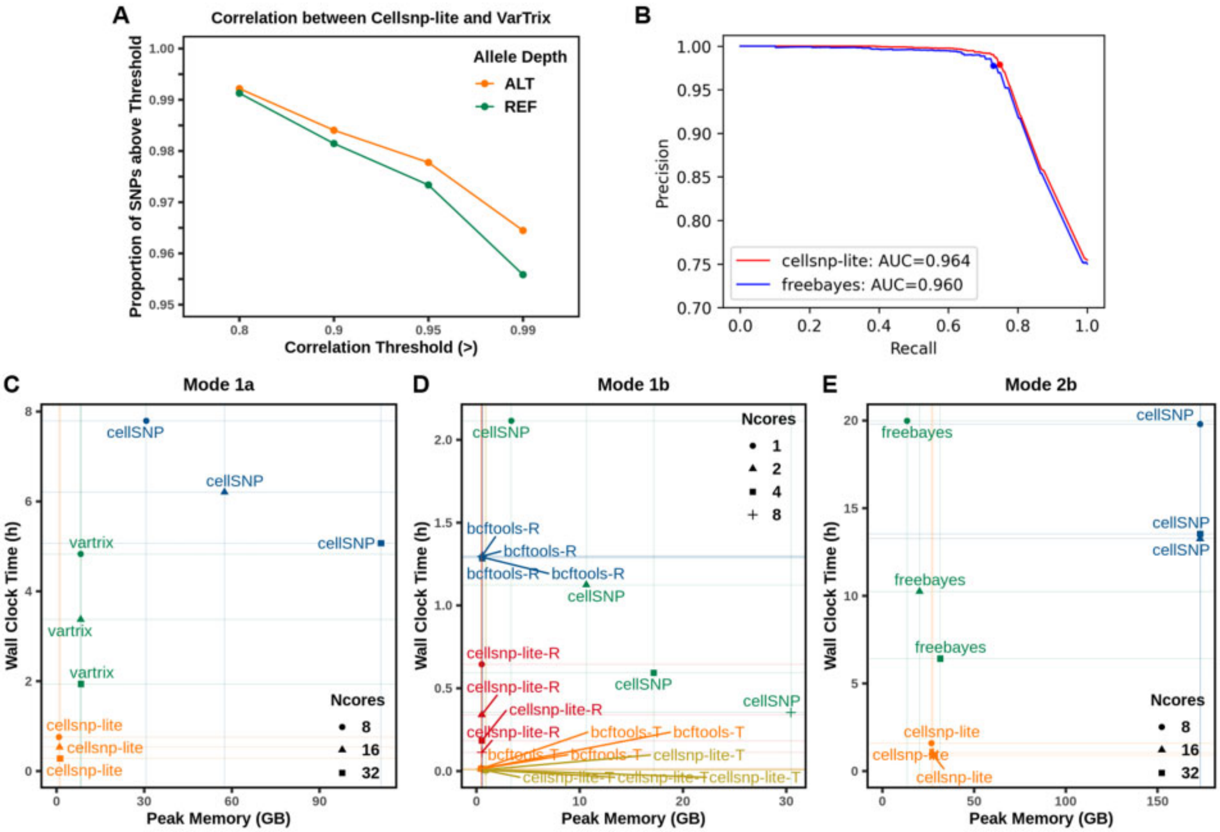

Fig. 1. Cellsnp-lite showed high accuracy in pileup allelic counts and substantial improvement in running speed. CellSNP is the predecessor in Python of cellsnp-lite. (**A**) In mode 1a on the souporcell dataset, pileup REF and ALT allele counts of cellsnp-lite and vartrix were highly concordant (>95% SNPs with Pearson's correlation coefficient >0.99). (**B**) The precision-recall curve for the 6239 heterozygous SNPs shared by cellsnp-lite, freebayes and genotype array, with TGT (Translated Genotype) as labels and GQ (Genotype Quality; converted from PL or GL) as scores. These SNPs were called from the souporcell dataset (droplet-based) which was treated as bulk data in mode 2b, where we treated SNP arrays-based genotype as ground truth. The red or blue dot denoted the result when GQ equals 20 for cellsnp-lite and freebayes, respectively. The curves and GQ20 dots of the two tools almost coincided with each other while cellsnp-lite gave a marginally higher AUC (0.964 versus 0.960). (**C**) Mode 1a on the souporcell dataset. Compared to vartrix, cellsnp-lite was about 6× speedups in wall time and could save up to ~90% memory. Limited by huge memory usage, cellSNP was slower than vartrix on this big dataset. (**D**) Mode 1b on the cardelino dataset. For either *-R* or *-T* option, cellsnp-lite was faster (in wall time about 2× ~ 11× speedups for *-R* and around 1.6× ~ 3× speedups for *-T*) than bcftools mpileup, even with a single thread. Compared to bcftools mpileup, cellsnp-lite used slightly less memory with *-R* option while no more than two times memory with *-T* option. For both tools, the *-T* option was much (>25×) faster than *-R* option on this small dataset. Using large memory, cellSNP could be faster than bcftools mpileup *-R* option with many cores. (**E**) Mode 2b on the souporcell dataset. Compared to freebayes, cellsnp-lite was about 7× ~ 13× speedups in wall time with no more than 2 times memory. The memory usage of freebayes gradually approached and eventually exceeded the one of cellsnp-lite as the number of threads increased. Limited by huge memory usage, cellSNP gained little increase of speed with many threads.

pileup allelic counts between cellsnp-lite, cellsnp-Python, bcftools, vartrix (available at https://github.com/10XGenomics/vartrix; release version 1.1.16) and freebayes (Garrison and Marth, 2012). Unsurprisingly, we found cellsnp-lite gives identical allelic read counts compared to both its Python version and bcftools, as all use htslib for reads fetching (see settings in Supplementary Sections 4.3 and 6.2.3 for mode 1b and 2b on well-based cells, respectively).

In addition, for mode 1a with given SNPs on droplet-based data, we compared cellsnp-lite with vartrix, and found they are highly concordant (∼97% SNPs with Pearson's correlation coefficient >0.99, and >99.9% SNPs with mean absolute error < 0.01; Fig. 1A and Supplementary Tables S2–S3 and S5–S6). When there are no candidate SNPs given (mode 2), cellsnp-lite could call heterozygous SNPs directly. We found that in this setting, cellsnp-lite gives identical allelic counts compared to its mode 1 with given SNPs. Comparing to a commonly used alternative method, freebayes, for calling heterozygous SNPs from droplet-based scRNA-seq data, we found cellsnp-lite gives marginally higher accuracy (area under the precision-recall curve, AUPRC: 0.964 versus 0.960; Fig. 1B and Supplementary Fig. S2), where we treat the SNP arrays-based genotype as ground truth.

3.2 Substantial improvement on running speed

Thanks to well-supported parallel computing, cellsnp-lite substantially outperforms existing methods in all the above settings, with achieving around 6× to 13× speedups in droplet-based data (with large size: 10 to 100 GB per sample). When SNPs are given in mode 1a, cellsnp-lite is around 6× speedups in wall time and could save up to ∼90% peak memory compared to vartrix (Fig. 1C and Supplementary Fig. S1; see CPU time in Supplementary Tables S1 and S4). When lack of known SNPs in mode 2b, where we treat droplet-based data as bulk data, cellsnp-lite is about 7× ∼ 13× speedups in wall time with no more than 2 times memory than freebayes in calling heterogynous SNPs (Fig. 1E; see CPU time in Supplementary Table S9).

Interestingly, cellsnp-lite also clearly outperforms bcftools mpileup on speed (with around 1.6× ∼ 11× speedups in wall time) for well-based samples thanks to its better use of multi-threading, as bcftools only uses multiple threads for writing file (Fig. 1D for given SNPs and Supplementary Fig. S3 for *de novo* genotyping; see CPU time in Supplementary Tables S7 and S10). We also noticed that since the bam files are small in well-based cells (50 to 500 MB per cell), the bcftools '-*T*' option is more efficient to stream the whole bam file instead of to fetch each SNP through index file. In this particular setting with much less computing demand, cellsnp-lite (also with '-*T*' option) is still about 1.6× ∼ 3× speedups in wall time than bcftools with using no more than 2 times memory (Fig. 1D; see CPU time in Supplementary Table S7).

Though cellsnp-lite is able to do joint calling and genotyping in mode 2a, it is substantially (>30×; Supplementary Table S8) slower than calling first in a bulk manner by mode 2b followed by genotyping in mode 1a. On the other hand, the joint strategy could be particularly useful for small chromosomes and mitochondrial genome.

4 Conclusion

Cellsnp-lite aims to pileup the detected alleles in single-cell or bulk sequencing data with simple filtering. It has highly concordant results, but substantially higher speed and less memory usage compared to other methods. Cellsnp-lite also provides a simplified user interface and better convenience that supports parallel computing, cell barcode and UMI tags. On the other hand, cellsnp-lite does not aim to address the technical issues caused by sequencing platforms, e.g. uneven amplification in scDNA-seq and low coverage in scRNA-seq, but rather leaves them to downstream statistical modelling. Taken together, cellsnp-lite is expected to largely boost single-cell genetics analysis, especially considering the increasingly large size of single-cell data.

Acknowledgement

The authors thank members in Oliver Stegle's lab, especially Davis McCarthy and Hana Susak, for using and providing feedbacks on the predecessor CellSNP and current cellsnp-lite.

Funding

The authors acknowledge support from the University of Hong Kong and its Li Ka Shing Faculty of Medicine through a start-up fund.

Conflict of Interest: none declared.

References

Cuomo,A.S. *et al.* (2020) Single-cell RNA-sequencing of differentiating iPS cells reveals dynamic genetic effects on gene expression. *Nat. Commun.*, **11**, 1–14.

Fan,J. *et al.* (2018) Linking transcriptional and genetic tumour heterogeneity through allele analysis of single-cell RNA-seq data. *Genome Res.*, **28**, 1217–1227.

Garrison,E. and Marth,G. (2012) Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*.

Huang,Y. *et al.* (2019) Vireo: Bayesian demultiplexing of pooled single-cell RNA-seq data without genotype reference. *Genome Biol.*, **20**, 1–12.

Jun,G. *et al.* (2012) Detecting and estimating contamination of human DNA samples in sequencing and array-based genotype data. *Am. J. Hum. Genet.*, **91**, 839–848.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Li,H. (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, **27**, 2987–2993.

Liu,F. *et al.* (2019) Systematic comparative analysis of single-nucleotide variant detection methods from single-cell RNA sequencing data. *Genome Biol.*, **20**, 1–15.

Ludwig,L.S. *et al.* (2019) Lineage tracing in humans enabled by mitochondrial mutations and single-cell genomics. *Cell*, **176**, 1325–1339.

McCarthy,D.J. *et al.* (2020) Cardelino: computational integration of somatic clonal substructure and single-cell transcriptomes. *Nat. Methods*, **17**, 414–421.

Navin,N.E. (2014) Cancer genomics: one cell at a time. *Genome Biol.*, **15**, 1–13.

The 1000 Genomes Project Consortium., Auton, A. *et al.* (2015) A global reference for human genetic variation. *Nature*, **526**, 68–74.

Zaccaria,S. and Raphael,B.J. (2021) Characterizing allele-and haplotype-specific copy numbers in single cells with CHISEL. *Nat. Biotechnol.*, **39**, 207–214.