BMC Bioinformatics

SOFTWARE

Open Access

DAESC+: high-performance, integrated software for single-cell allele-specific expression data

Tengfei Cui[1,2] and Guanghao Qi[1*]

*Correspondence:
Guanghao Qi
gqi@uw.edu
[1]Department of Biostatistics, University of Washington, Seattle, WA 98105, USA
[2]Department of Statistics, Texas A&M University, College Station, TX 77840, USA

Abstract

Single-cell allele-specific expression (ASE) provides valuable insights into gene regulatory mechanisms. However, its utility is limited by the lack of dedicated computational tools. We present DAESC+, a dual-module end-to-end software package for the processing and analysis of single-cell ASE. The preprocessing module, DAESC-P, is a user-friendly bioinformatics pipeline to obtain ASE counts from multiplexed scRNA-seq data. The analysis module, DAESC-GPU, is a scalable tool for differential ASE analysis powered by graphics processing units (GPUs). We demonstrated that DAESC-P is more accurate than the existing SALSA pipeline. DAESC-GPU is dozens of times faster than our previous method (DAESC) and scalable to over a million cells. Applying DAESC+to a subset of the OneK1K cohort, we identified 15 genes exhibiting differential regulatory patterns between naïve and central memory CD4+T cells, and 2 genes between naïve and memory B cells.

Keywords Single cell, Allele-specific expression analysis, GPU, Bioinformatics

Background

Allele-specific expression (ASE) is a powerful signal to study multiple regulatory mechanisms [1–4]. ASE is typically quantified from RNA-sequencing by the number of reads from two parental alleles at a transcribed single-nucleotide polymorphism (referred to as tSNP). In the past decade, single-cell RNA-seq (scRNA-seq) has advanced the quantification of ASE to single-cell resolution. Single-cell ASE was used to study a range of biological effects, such as transcriptional bursting [5–7], genomic imprinting8, cis-regulatory effects [9–12], and cancer clonal evolution [13]. With the growth of single-cell data and emerging technologies (e.g., long-read sequencing), ASE has the potential to lead to many novel discoveries [4,14,15]. However, there is an urgent lack of computational tools to process and analyze this type of data.

While total gene expression counts are usually provided by the read mapping software (e.g., STAR [16,17], Cell Ranger), ASE counts are not readily available. To retrieve ASE counts from scRNA-seq reads, researchers must conduct a series of bioinformatics processing: mapping reads to the reference genome [18], genotype calling from scRNA-seq

[19], haplotype phasing [20], removing mapping bias for allele-specific reads [21], and read counting for reference and alternative alleles [19]. Previous studies typically relied on fragmented tool chains that address only individual components of the workflow, requiring extensive manual integration, complex software installation, and ad-hoc data preparation [5,7,9,12,22,23]. SALSA [24] is a combined implementation of existing software and currently the only integrated pipeline to generate single-cell ASE from 10x Genomics data. However, SALSA assumes that all reads from a FASTQ file are from the same individual and does not account for multiplexed design, where multiple samples are sequenced together. This can lead to erroneous genotype calls and ASE quantification. In addition, SALSA is not fully end-to-end: for the large volume of data in public repositories such as the Sequencing Read Archive (SRA) [25,26], users must manually download raw data and preprocess it into specific formats prior to analysis. The pipeline also depends on multiple external reference datasets and legacy software packages [24], which require careful configuration. As a result, users must frequently consult the documentation of multiple third-party tools to resolve compatibility issues and adjust parameters, creating a substantial barrier particularly for researchers without extensive bioinformatics expertise.

Software for downstream statistical analysis is equally lacking. Only a few methods have been developed to date [4,5,8,10–12,27,28]. An important application of ASE is to study gene-environment interaction by comparing allelic imbalance across conditions, known as differential ASE (D-ASE). Although two other methods provides functions for D-ASE analysis [10,11], our previous work, DAESC [12], is the only method for single-cell D-ASE using multiple samples. DAESC can be applied to any comparison of interest, such as discrete cell types, continuous cell states, and case–control disease status. Through simulation studies and read data applications, DAESC demonstrated robust type I error and high power, and identified hundreds of genes that are dynamically regulated during endoderm differentiation [12]. However, the variational EM algorithm adopted by DAESC is computationally intensive. The original R implementation does not scale well to large datasets. With rapid expansion of single-cell RNA-seq data [29], DAESC has the potential for many new applications, but its utility is limited by lack of computational efficiency.

We present DAESC + , an integrated software package for the processing and analysis of single-cell ASE data. DAESC+ has two modules: DAESC-P and DAESC-GPU. The preprocessing module, DAESC-P, is an end-to-end bioinformatics pipeline to obtain ASE counts from raw reads of 10 × Genomics scRNA-seq data. DAESC-P allows multiplexing, requires minimal user input, and reduces workload of environment configuration. The analysis module, DAESC-GPU, is a high-performance tool for single-cell D-ASE analysis implemented using graphics processing units. DAESC-GPU is dozens of times faster than DAESC and can be applied to over 1 million cells. Application of DAESC + to the scRNA-seq data of the OneK1K cohort identified genes that have distinct patterns of allelic imbalance between naïve and memory T cells and B cells.

Results

Overview of DAESC +

DAESC + is comprised of two modules: DAESC-P for preprocessing and DAESC-GPU for differential analysis. DAESC-P begins with an automated workflow to download

sequencing data from SRA [25, 26] and convert SRA files to FASTQ format. Because the FASTQ file names generated from SRA are not compatible with Cell Ranger, DAESC-P automatically detects read 1, read 2, and when present, index files based on read length, and renames them according to 10x naming convention. DAESC-P requires only SRA run IDs and a list of cell barcodes with donor information and automatically generates input files in the appropriate format, thereby eliminating the need for manual data processing. Next, if the scRNA-seq experiments are multiplexed, i.e., multiple samples are sequenced together, DAESC-P demultiplexes FASTQ files by individual and merges individual-specific FASTQ files across sequencing runs (Fig. 1). This step is essential for variant calling but is not supported by SALSA. By handling multiplexed data, DAESC-P is applicable to a broader range of datasets. All FASTQ files are then aligned to the reference genome using Cell Ranger (version 9.0.1, 10 × Genomics).

DAESC-P subsequently executes an integrated implementation of the SALSA pipeline [24], including genotype calling, haplotype phasing, alignment bias removal by WASP [21], and ASE read counting, as well as a few other intermediate steps. In contrast to SALSA, DAESC-P consolidates all modules into a few scripts and distributes all external reference files with the software. This design eliminates the need for users to navigate multiple databases and software documentations to identify compatible components. DAESC-P also incorporates technical optimizations to improve computational performance, including increased memory allocation, an updated software version for faster haplotype phasing, and straightforward parallel computing enabled through pre-included Slurm scripts (Fig. 1). See "Methods" for details.

DAESC-GPU is a scalable GPU-based re-implementation of DAESC designed to accelerate differential ASE analysis. The original DAESC employs a beta-binomial regression model with ASE counts as the outcome and experimental and biological

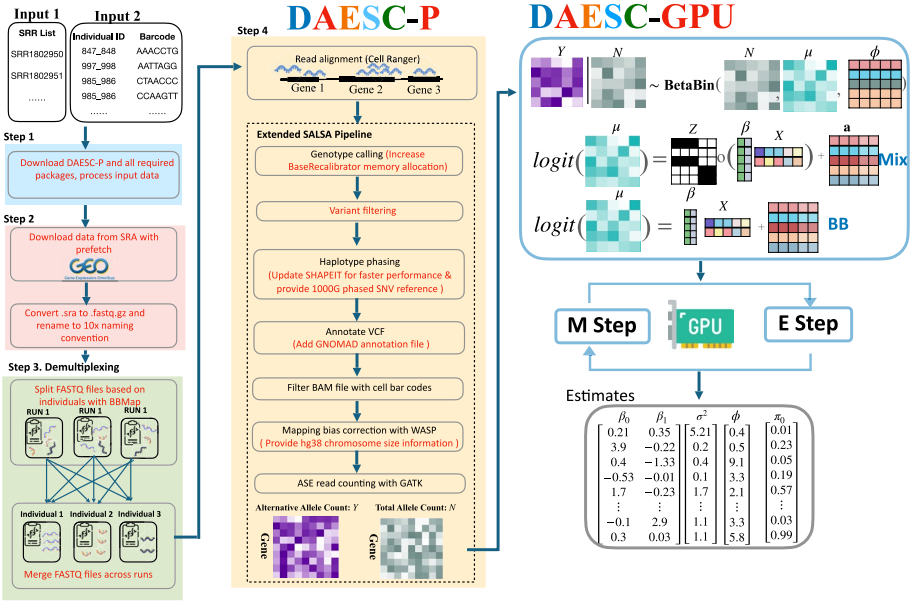

Fig. 1 Schematic of DAESC + . DAESC + is an end-to-end package for processing and analysis of single-cell ASE data, comprised of two modules. 1) DAESC-P is a preprocessing module to obtain single-cell ASE counts from raw reads of multiplexed scRNA-seq data. The processing steps and technical optimizations that are incorporated into our pipeline but not in SALSA are colored in red text. 2) DAESC-GPU is a scalable tool for single-cell differential allele-specific expression (D-ASE) analysis, leveraging large matrix computation and massive parallel computing using graphics processing units (GPUs)

conditions as predictors (see "Methods" for summary) [12]. It incorporates individual-specific random effects to account for sample repeat structure arising from multiple cells per individual, as well as latent variables to enable implicit haplotype phasing. Whereas DAESC analyzes one gene at a time, DAESC-GPU performs analysis of all genes simultaneously using large matrix operations (Fig. 1). These operations are implemented in CuPy, a Python library using CUDA Toolkit, which dramatically accelerates computation using GPUs through massive parallel processing. Because CuPy does not provide an efficient built-in optimizer for this setting, we developed a custom, fully vectorized optimization routines that implement the BFGS algorithm across all genes in parallel. To further improve efficiency, we implemented custom CuPy kernels that execute directly on GPUs and fuse multiple mathematical operations into a single specialized function, yielding substantially higher performance than built-in CuPy functions. DAESC-GPU can be executed on freely available GPUs through Google Colab or on dedicated GPUs for enhanced performance. See "Methods" for details.

Processing and analysis of the OneK1K data

We applied DAESC + to the OneK1K cohort [29] which was comprised of 982 donors of Northern European ancestry. Multiplexed scRNA-seq data were collected using 10 × Genomics Chromium for ~ 1.27 million peripheral blood mononuclear cells (PBMCs). For benchmarking purposes, we restricted our analysis to pools 1–10 (out of 77) comprised of 105 individuals. We used DAESC-P to download and process raw reads from SRA and obtained ASE read counts. For comparison, we also conducted the same processing with SALSA [24] in the default setting, which processed each sequencing run individually and summed up ASE counts across runs for each variant-cell pair. SALSA detected non-zero ASE counts in all variant-individual pairs among the top 30 highly expressed variants, even though some of the variants were homozygous in some individuals (Fig. 2). This artifact arised because SALSA conducted genotype calling using multiplexed FASTQ files, thereby mixing reads from multiple individuals and falsely calling homozygotes as heterozygotes. This error led to about 95% false monoallelic expression for many individuals, severely biasing the results (Fig. 2 and Supplementary Figs. 1 and 2). In contrast, DAESC-P correctly identified zero ASE counts for homozygous genotypes by splitting multiplexed FASTQ files by individuals and recombining individual-specific FASTQ files across sequencing runs (Fig. 2). In addition, our technical optimization improved pipeline efficiency. Driven by upgrading the haplotype phasing software from SHAPEIT 4 to SHAPEIT 5, DAESC-P was > 1.6 times faster than SALSA for processing chromosomes 1–3 of donor 847,848 (Pool 11, Supplementary Fig. 3).

After preprocessing, we obtained single-cell ASE data of 160,709 cells (See "Methods" for details) from OneK1K. The dataset contained 29 cell types (Supplementary Fig. 4). We used DAESC-GPU to test differential ASE for three immune cell types between naïve versus memory subtypes: CD4 + T cells (naïve and central memory, 69,713 cells), CD8 + T cells (naïve and central memory, 7,029 cells), and B cells (naïve and memory, 11,534 cells). They were among some of the most abundant cell types in the sample (Fig. 3a). Despite the sparsity of the data, DAESC-GPU was over 10 times faster than DAESC for both the BB and Mix models in most scenarios (Fig. 3b). For example, DAESC-GPU-Mix completed D-ASE analysis between naïve versus memory CD4 + naïve T cells in 7.9 min (A100) and in 43.8 min (T4), while the R package DAESC-Mix finished the same

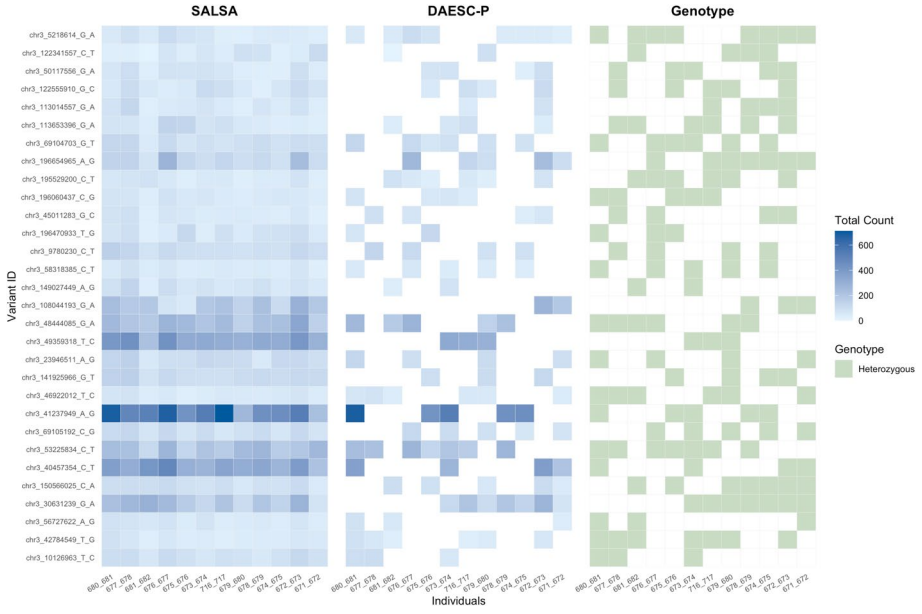

Fig. 2 Comparison of ASE counts quantified by DAESC-P and SALSA. The Y-axis shows the top 30 variants of highest total allele-specific read depth quantified by DAESC-P, and the X-axis shows the 12 individuals in Pool 4. Left: total allele-specific read counts quantified by SALSA. Middle: total allele-specific read counts quantified by DAESC-P. White squares represent zero counts (left and middle). Right: heterozygosity of the variants for each individual. Heterozygous genotypes are colored green. Homozygous or uncalled genotypes are colored white

task in 168.8 min (UW Server). For CD8 + T cells, DAESC-GPU-Mix completed the same analysis in 2.3 min (A100) and 6.1 min (T4), whereas DAESC-Mix used 270.8 min (UW Server). For DAESC-GPU, the runtime for CD4 + T cells was longer than that for CD8 + T cells or B cells, which was consistent with expectation due to larger number of cells. However, the runtime of DAESC is shorter for CD4 + T cells than CD8 + and B cells (Fig. 3b). Upon further investigation, we found that the original DAESC required more iterations to converge for CD8 + T cells and B cells (Supplementary Fig. 5). This phenomenon was likely due to the smaller number of cells for CD8 + T cells and B cells, which made the convergence less stable. The different optimizers used by DAESC and DAESC-GPU (R's built-in BFGS versus the vectorized L-BFGS) adopted different line search strategies, step-length determination, and termination criteria, which could have partly led to the differences between the two methods.

DAESC-GPU-Mix identified 15 D-ASE genes between CD4 + naïve vs. central memory T cells (FDR < 0.05, Fig. 3 and Supplementary Table 1). DAESC-GPU-BB identified 6 genes, among which 5 genes overlapped with DAESC-GPU-Mix (Fig. 3c and d). For B cells, DAESC-GPU-Mix and DAESC-GPU-BB identified the same two genes (Fig. 3c and Supplementary Fig. 6). Neither DAESC-GPU-BB nor DAESC-GPU-Mix identified significant D-ASE genes between CD8 + naïve vs. central memory T cells. Next, we sought to validate our findings by examining the overlap between the D-ASE genes identified by our methods and the eGenes reported in the original study [29]. For CD4 + T cells, 4 out of 6 genes detected by DAESC-GPU-BB overlapped with the list of eGenes. DAESC-GPU-Mix had the same rate of overlap despite more significant genes, with 10 out of 15 (Fig. 3e). This finding was consistent with our previous recommendation to use DAESC-GPU-Mix to analyze data with many individuals (e.g., N > 20). For B cells, it was

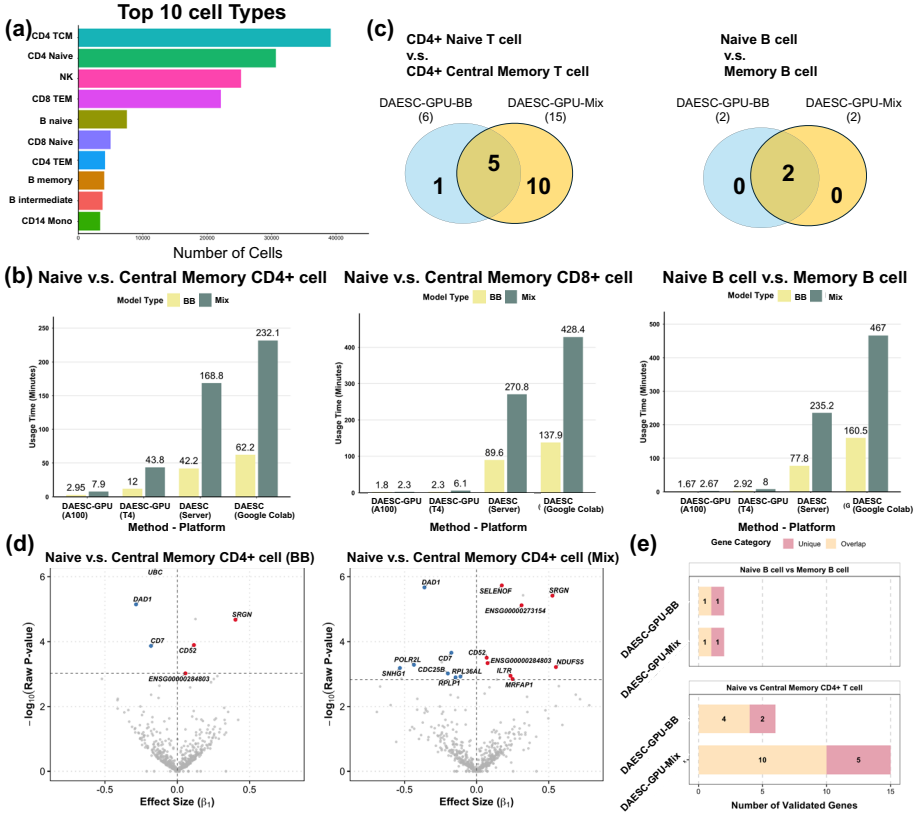

Fig. 3 The performance of DAESC-GPU on the OneK1K cohort. (a) The number of cells for top 10 most abundant cell types. (b) Runtime comparison between DAESC-GPU (on A100 and T4 GPUs) and the original DAESC R package for differential ASE analyses. (c) Number of significant genes identified by the DAESC-GPU-BB and DAESC-GPU-Mix for CD4+ T cells and B cells (FDR<0.05). No significant genes were found for CD8+ T cells. (d) Volcano plot of differential ASE analysis for CD4+ T cells. (e) Overlap between genes identified by DAESC-GPU and cell-type-specific eGenes reported in the original study as validation evidence. A gene is considered overlapped if it is an eGene in either the naïve or memory cell type.

difficult to compare the BB and Mix models due to small number of genes (2 genes by both DAESC-GPU-BB and DAESC-GPU-Mix).

Several significant D-ASE genes have functions related to immunity [30]. *SRGN* encodes serglycin, a protein best known as a hematopoietic cell granule proteoglycan, which is crucial for immune and secretory functions [1]. *CD7* encodes a transmembrane protein which is a member of the immunoglobulin superfamily [30]. It plays an essential role in T-cell interactions and in T-cell/B-cell interaction during early lymphoid development. *CD52* is involved in positive regulation of cytosolic calcium ion concentration [30]. A previous study found that soluble *CD52* exerts a concerted immunosuppressive effect on human T cell function [31]. *IL7R* encodes a receptor for interleukin 7, which has been shown to play a critical role in V(D)J recombination during lymphocyte development [30, 32]. Defects in this gene may be associated with severe combined immunodeficiency [30].

Performance of DAESC-GPU on endoderm differentiation data

To evaluate the performance of DAESC-GPU on a different sequencing technology, we conducted differential ASE analysis along pseudotime using single-cell RNA-seq data (Smart-seq2) from an endoderm differentiation experiment [9] (see "Methods"

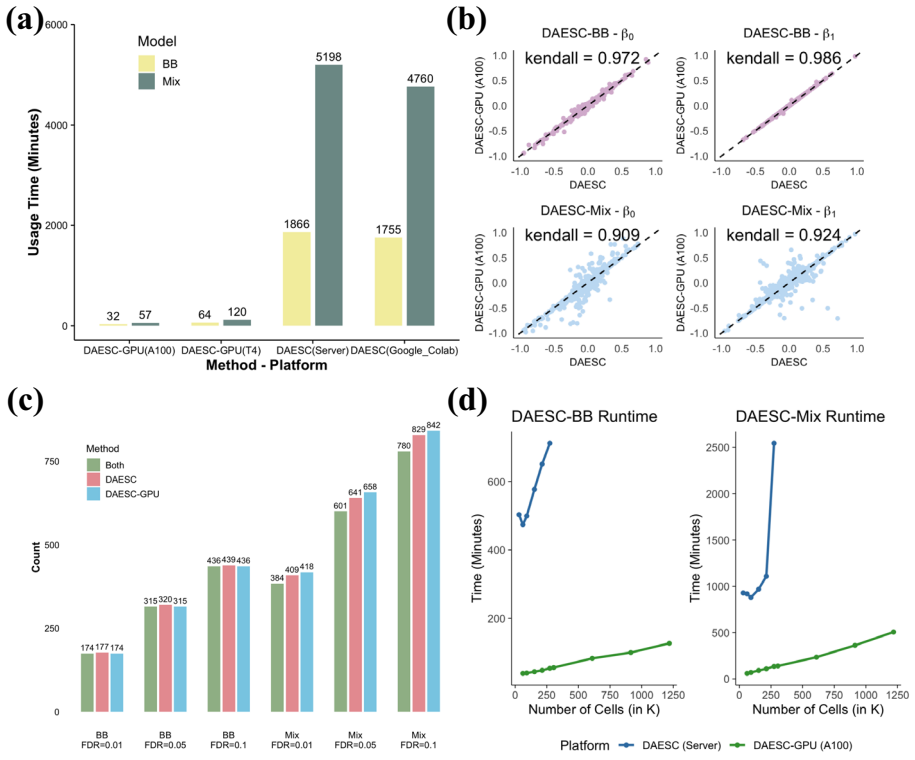

Fig. 4 The performance of DAESC-GPU on the endoderm differentiation data. **a** Runtime of DAESC-GPU (A100 and T4) vs DAESC R package (UW Server and Google Colab, 8 CPUs) for D-ASE along pseudotime for endoderm differentiation data. **b** Comparison of the estimates of main model parameters between DAESC-GPU and DAESC. Each dot is a gene. Black dashed line is $y = x$. Correlation (Kendall's τ) of the estimate between two methods is marked in the figure. **c** Total number of significant genes identified by DAESC, DAESC-GPU, and their intersection (both) under varying FDR thresholds. The intersection constitutes the majority of detected genes, indicating high level of consistency between DAESC and DAESC-GPU. **d** Runtime of DAESC-GPU vs DAESC (R package, 8 CPUs) with varying number of cells in simulation studies

for details). ASE count data were provided by the original study and hence DAESC-P did not need to be applied. After processing as described in our earlier paper [12], we obtained allele-specific read counts for 4,102 genes and 30,474 cells collected from 105 individuals. DAESC-GPU was 30 ~ 90 times faster than DAESC for analyzing this dataset (Fig. 4a). DAESC-GPU-BB used 64 min on a T4 GPU and 32 min on an A100 GPU, while DAESC-BB used 1,755 min on CPUs from Google Colab and 1,866 min on the UW Biostatistics Computing Cluster (referred to as "UW Server"). The patterns for the DAESC-GPU-Mix and DAESC-Mix were similar, though both methods used longer time than the BB version due to the extra computational complexity (Fig. 4a). In order to quantify the individual contributions of proposed L-BFGS optimizer and Cupy kernels, we designed a simplified version, referred to as DAESC-GPU (T4, R optim), by only replacing our vectorized L-BFGS optimizer with the standard R built-in BFGS optimizer. This version retained all other GPU operations and parameters. For both BB and Mix model, DAESC-GPU (T4, R optim) was slower than DAESC-GPU but still faster than DAESC R package (Supplementary Fig. 7). DAESC-GPU produced nearly identical estimates to DAESC. The Kendall correlation between DAESC-GPU and DAESC estimates were higher than 0.9 for β_0, β_1, σ^2, and ϕ for both the BB and Mix model (Fig. 4b, Supplementary Figs. 8 and 9). The correlation was weaker for π_0 for Mix ($r = 0.69$, Supplementary Fig. 9). However, this parameter is of secondary importance and does not

directly impact the interpretation of the model. For hypothesis testing of differential ASE, DAESC-GPU and DAESC detected nearly identical sets of dynamic ASE genes at varying FDR thresholds (Fig. 4c), especially for the BB model. There is slight discrepancy in the sets detected by DAESC-Mix and DAESC-GPU-Mix, but the overlap is still above 90%. The difference is likely due to increased model complexity of the Mix model, which makes it more challenging for different optimization algorithms converge to the same estimates. These results demonstrated high fidelity of DAESC-GPU to the original method despite using different computational techniques.

Next, we tested the scalability of DAESC-GPU in a simulation study (see "Methods" for details). The runtime of DAESC increased rapidly as the number of cells increased, while that of DAESC-GPU increased only moderately and did not present major challenges for analyzing > 1 million cells and 2,000 genes (Fig. 4d). For example, DAESC-BB required 757 min to analyze 0.27 million cells, whereas DAESC-GPU-BB completed the analysis of 1.2 million cells in only 127 min. Similarly, DAESC-Mix used 2545 min (1.77 days) to analyze 0.27 million cells, while DAESC-GPU-Mix needed only 457 min to analyze 1.2 million cells. These results demonstrated that DAESC-GPU is many times faster than DAESC and scalable to large single-cell ASE datasets.

Discussion

We presented DAESC + , an integrated software package for single-cell ASE analysis. DAESC + only requires SRA run IDs and cell barcodes as input and automatically downloads, processes and analyzes the data to generates differential ASE results with a few commands. The preprocessing module, DAESC-P, automates many data handling steps and requires minimal configuration overhead. In contrast to SALSA, DAESC-P supports the processing of multiplexed scRNA-seq data, which is widely available in public repositories. Although these additions may be manageable for experienced bioinformaticians to implement independently, many labs primarily focus on data generation but lack bioinformatics expertise. For these groups, configuring multiple software packages and executing each step of the workflow can be both challenging and time-consuming, potentially delaying research progress. Although DAESC-P is not a "one-click" solution, it streamlines ASE processing within a unified framework, substantially lowering technical barriers and accelerating scientific discovery.

The differential analysis module, DAESC-GPU, significantly accelerates computational and scales well to large datasets on an entry-level GPU. Further scalability can be achieved when additional dedicated GPUs are available. Although DAESC-GPU does not modify the statistical model underlying DAESC, the computational acceleration it provides is essential for practical applications to real datasets. Based on the observed runtime trend (Fig. 4d), applying the original DAESC R package to datasets containing millions of cells would be computationally prohibitive. In contrast, DAESC-GPU keeps the runtime manageable, enabling analyses at the scale of modern single-cell datasets and meeting the demands of rapidly growing single-cell data.

Traditionally, Smart-seq [33, 34] protocols was commonly used for single-cell ASE quantification due to its full-length coverage [6, 7, 9]. However, it is cost-prohibitive to use Smart-seq to measure many cells and individuals. In contrast, 10 × Genomics Chromium has been one of the most popular scRNA-seq technologies due to scalability. However, 10 × has a strong bias toward the 3' or 5' end of genes and cannot capture the

exonic SNPs further inside thegene body. This leads to fewer allele-specific reads and higher level of sparsity. Nevertheless, the data we obtained from 10 out of 77 pools of OneK1K were sufficiently powered to detect differential ASE between naïve and memory T cells and B cells. This indicates that 10x can still yield valuable ASE data. While Smart-seq data can be processed with a standard GATK pipeline, 10 × data does not have a mature pipeline for ASE extraction. DAESC-P will fill this important gap.

Although DAESC-P depends on the sequencing technology, DAESC-GPU is a generic software that can be used on any single-cell ASE data, if reference and alternative allele read counts are available. We demonstrated that DAESC-GPU improved computational speed many times compared to DAESC for both Smart-seq and 10 × data (Figs. 3 and 4). The improvement appeared to be more pronounced for the Smart-seq2 dataset, likely due to higher sequencing depth and reduced sparsity. By applying DAESC-GPU to a subset of the OneK1K cohort, we identified genes with cell-type-specific cis-regulatory effects. Those genes were further validated by comparing with previously identified eGenes. These observations indicate that DAESC-GPU can be a candidate tool for analyzing single-cell ASE from other technologies (e.g., long-read sequencing). Long-read sequencing can capture exonic SNPs inside the gene along with isoform information. Comprehensive testing of the software on long-read sequencing is beyond the scope of this paper but remains a promising direction for future work.

Our study has a few limitations. First, due to the computational demand to process raw sequencing data, we only analyzed 10 pools from the OneK1K dataset. With more computational resources available in the future, we aim to process the entire dataset and conduct a well powered study of differential ASE for immune cells. Second, although DAESC-GPU significantly accelerates the computational speed of DAESC, it does not improve the underlying statistical model. A limitation of this model is its inability to integrate information across multiple discrete cell types. Nevertheless, it remains valuable as a high-performance software package for generic differential ASE analyses. Third, the runtime advantage of DAESC-GPU in differential analyses between naïve andcentral memory CD4 + T cells appeared to be less pronounced due to the sparsity of single-cell counts (Fig. 3b).In such cases, DAESC-GPU organizes the data into large sparse matrices that include all cell-gene combinations with zero counts, while DAESC R package analyzes each gene individually and excludes cells with zero total read counts. This difference narrows their performance gap, but DAESC-GPU remains substantially faster. Fourth, we quantified gene-level ASE using the top variant (i.e., the SNP with the highest read depth) in the OneK1K analysis. This approach was adopted in a previous paper from the GTEx Consortium [3]. A more powerful and unbiased approach is to use haplotype information to aggregate ASE counts across multiple exonic variants. However, this approach is challenging because inferred haplotypes are often noisy for variants with low read depth. In addition, unlike individual variants, which naturally have reference and alternative alleles, it is not straightforward to define reference and alternative haplotypes. However, this issue does not affect DAESC-P processing as it occurs downstream. While it may affect the results produced by DAESC-GPU, it does not affect the relative performance comparison between DAESC-GPU and DAESC. We believe this issue is beyond the scope of this paper and leave it as future work.

Conclusions

In conclusion, DAESC + is a powerful software package for single-cell ASE processing and differential analysis. Application of the software will reveal novel biological insights from the rapidly growing allele-specific datasets.

Methods

DAESC-P: preprocessing module of DAESC +

DAESC-P is an open-source, user-friendly bioinformatics pipeline designed to generate single-cell ASE counts from multiplexed scRNA-seq data. The pipeline is conceptually described as the following steps.

1. In the first stage, an initialization script retrieves the DAESC-P distribution from Zenodo and generates the following intermediate files in a temporary directory to support downstream analysis. First, DAESC-P identifies all unique individual IDs in the input file and generates a configuration file that maps individual IDs to the target chromosomes (all 22 autosomes by default). Second, DAESC-P extracts individual-specific cell barcode information to produce both a CSV and a plain-text file for each individual. These two files contain barcodes in distinct formats to ensure compatibility with different functional modules of the pipeline. All partitioned files are automatically detected and integrated into the subsequent analytical workflow.
2. Raw sequencing data (.sra files) are downloaded from the SRA database using SRA-Toolkit. Users only need to supply the list of SRA run IDs as input. DAESC-P further converts.sra files into compressed FASTQ files (.fastq.gz) using the fastq-dump command from SRA-Toolkit. To ensure compatibility with Cell Ranger, DAESC-P renames FASTQ files based on the naming conventions required by 10 × Genomics (e.g., [Sample]_S1_L001_R1_001.fastq.gz). Each sequencing run includes read 1 (R1), read 2 (R2), and potentially index (I1). This step is conducted for all sequencing runs in parallel.
3. DAESC-P then performs the following demultiplexing operations. For each sequencing run in each pool (of individuals), DAESC-P uses BBMap program [35] to split multiplexed FASTQ files into individual-specific FASTQ files based on lists of cell barcodes provided by Gene Expression Omnibus (GEO). Single-run FASTQ files are merged across all runs for each individual to generate individual-specific FASTQ files for downstream alignment and genotype calling. This step is performed in parallel across individuals. Cell Ranger is used to align each FASTQ file to the reference genome.
4. The remaining steps are adapted from the SALSA pipeline [24], which comprises variant calling using GATK (4.2.0.0) best practices workflow, haplotype phasing using SHAPEIT 5 [36], VCF file annotation with GATK Funcotator, alignment bias removal using WASP [21], and ASE read counting using GATK ASEReadCounter. DAESC-P consolidates seven SALSA scripts into a single command to simplify the process and incorporate multiple technical optimizations. After genotype calling, DAESC-P filters variants using bcftools (v1.9) to retain only SNPs with genotyping quality (GQ) ≥ 30. This step ensures that the final ASE counts do not include low-quality SNPs. For haplotype phasing, we replaced the SALSA default software of SHAPEIT 4.2 to SHAPEIT 5, significantly improving computational speed (Supplementary Fig. 2). For

user convenience, DAESC-P is distributed with reference datasets, including 1000 Genomes phased SNV reference (for haplotype phasing), GNOMAD annotation file (VCF annotation), and hg38 chromosome information (for WASP mapping bias correction) [24]. Since the default memory setting of SALSA is insufficient, we increased memory allocation to 96 GB for each GATK function, enabling the processing of large datasets.

DAESC-GPU: differential analysis module of DAESC +

DAESC-GPU is based on our previously developed statistical method for single-cell differential ASE analysis (DAESC) [12]. For a gene or heterozygous transcribed SNP (tSNP), we assume y_{ij} to be the alternative allele read count of individual i and and cell j, and n_{ij} to be the total allele-specific read count. DAESC is based on beta-binomial regression

$$y_{ij}|n_{ij} \sim BB\left(n_{ij}, \mu_{ij}, \phi\right),$$

where the average allelic proportion μ_{ij} is linked to independent variable x_{ij}. DAESC is comprised of a baseline model (DAESC-BB) and a full model (DAESC-Mix). DAESC-BB incorporates individual-specific random effects a_i in beta-binomial model to account for the sample repeat structure arising from multiple cells measured per individual:

$$\log\left(\frac{\mu_{ij}}{1-\mu_{ij}}\right) = \beta_0 + \beta_1 x_{ij} + a_i, a_i \sim N\left(0, \sigma_a^2\right)$$

DAESC-Mix further incorporates implicit haplotype phasing using latent variable z_i that is either 1 or – 1:

$$\log\left(\frac{\mu_{ij}}{1-\mu_{ij}}\right) = z_i\left(\beta_0 + \beta_1 x_{ij}\right) + a_i,$$

$$z_i = 2\delta_i - 1, \delta_i \sim Bernoulli\left(\pi_0\right), a_i \sim N\left(0, \sigma_a^2\right).$$

The value of z_i depends on whether the alternative allele of the eQTL SNP is on the same haplotype on the alternative or reference allele of the tSNP, which leads to approximately opposite allelic fractions [12]. Variational EM algorithm is used to estimate the unknown parameters and conduct hypothesis testing of differential ASE (H_0: $\beta_1 = 0$).

DAESC-GPU is an accelerated, scalable software for single-cell differential ASE analysis, which overcomes computational limitations of the existing DAESC R package and facilitates the analysis of large-scale data. DAESC-GPU enhances computational efficiency by utilizing Graphics Processing Units (GPUs), which is an electronic circuit that can perform massively parallel computation at high speed. GPUs are integrated in our software through CuPy, an open-source Python library for GPU-accelerated computing using CUDA Toolkit. CuPy provides kernel functions which allow users to write highly specialized and optimized codes running on GPUs.

A number of computational techniques are employed to transform the original R code of DAESC into Python and CuPy. The original R package analyzes one gene at a time. To best leverage the power of GPU computing, our software reframes the algorithm into large matrix operations that can analyze thousands of genes concurrently. An essential technique is a fully vectorized L-BFGS framework that enables simultaneous

optimization of many objective functions (one objective function for each gene). This vectorization is one of the most critical components of DAESC-GPU (Supplementary Fig. 9). Each step, including gradient updates, inverse Hessian approximations, and search directions, is structured to operate in a parallelized manner across multiple objective functions. Since built-in functions of CuPy are often not available or efficient for our model, DAESC-GPU uses element-wise kernels to integrate multiple mathematical operations into one single specialized function written in C++. For instance, computing the log-likelihood of the model requires the sigmoid function and the difference between digamma functions, for which there are no built-in CuPy functions. By employing custom kernels, these computations are combined into one single function in C++, which maximizes low-level control over GPUs. These features significantly improve the speed of our software. Finally, the efficiency of DAESC-GPU is further enhanced by combining kernel functions and the memory pool feature in CuPy. Computations are performed directly on the original data without requiring additional memory for latent variables.

Evaluation of DAESC-GPU on endoderm differentiation data

We compared the performance of DAESC-GPU, run on T4 and A100 on Google Colab, and the original DAESC package, run on 8 CPUs from Google Colab (80 GB RAM) and UW Biostatistics Computing Cluster (referred to as "UW Server", 96 GB RAM), respectively. In silico experiments were conducted on single-cell ASE data from an endoderm differentiation experiment by Cuomo et al. [9]. This study profiled gene expression at 4 differentiation times points using Smart-seq2. We obtained SNP-level allele-specific read counts for 105 donors and followed the same preprocessing steps as in our previous paper [12]. In short, we removed SNPs with low mappability and monoallelic expression to reduce potential genotyping error. For each gene, we aggregated SNP-level ASE counts across all SNPs in the exons of the gene and obtained two haplotype-specific counts (hap1 count and hap2 count). We further removed the genes which had non-zero ASE counts in $\leq 20\%$ of the cells. See Qi et al. [12] for more details. After preprocessing, we obtained allele-specific read counts for 4102 genes and 30,474 cells. We conducted differential ASE analysis along pseudotime to detect dynamically regulated genes during endoderm differentiation. Pseudotime was inferred and provided by the original study [9].

Next, we evaluated the scalability of DAESC-GPU to larger datasets by simulating additional cells based on the endoderm differentiation data. First, we randomly selected 2000 genes. To mimic realistic read depth, we used the total read counts (TRC) from real data and kept them fixed in the simulations. To simulate datasets with more cells, we duplicated the real TRC as TRC for the additional cells. For example, we duplicated the $2000 \times 30{,}474$ TRC matrix once and created a $2000 \times 60{,}948$ matrix of TRC for 60,948 cells. Larger datasets were simulated by duplicating the TRC matrix more times. Alternative allele counts were simulated using the same procedure as described in the Qi et al. [12] , which adopted beta-binomial models incorporating donor-level random effects and haplotype combinations. Model coefficients $(\beta_0, \beta_1, \sigma^2, \phi)$ for the simulations were derived by fitting DAESC-BB on the real data and selecting 500 genes with largest $|\beta_1|$, as described in Qi et al. [12]. The number of cell observations ranged from 30,474 to 1.2 million. Memory consumption scaled with the number of genes and cells. Analyzing the

original endoderm differentiation data required 7.8 GB of GPU memory. In this simulation study, DAESC-GPU was able to simultaneously analyze 2000 genes with 0.35 million cells on an NVIDIA A100 GPU. To accommodate the increased memory demand associated with larger number of cells, we partitioned the genes into groups to accommodate the limited RAM. For cell counts below 0.35 million, DAESC-GPU 2000 genes were analyzed simultaneously. For counts between 0.35 and 0.7 million, the genes were divided into two groups and analyzed sequentially. For cell counts between 0.7 and 1.05 million, the genes were split into three groups. For counts exceeding 1.05 million, the genes were divided into four groups. The reported runtime is the sum of runtime of the groups. Due to the rapidly increasing runtime, we restricted the use of DAESC (R package) to datasets with no more than 0.27 million cells. For comparison, we ran the R package on the UW Biostatistics Server with 8 CPUs and 96 GB RAM, and on the Google Colab with 8CPUs and 80 GB RAM.

OneK1K data analysis

OneK1K is comprised of 982 donors of Northern European ancestry, with scRNA-seq data for ~ 1.27 million peripheral blood mononuclear cells (PBMCs). The scRNA-seq data are multiplexed and contain 77 pools, each of which is comprised of 9–18 individuals in multiple runs. We analyzed a subset of the OneK1K cohort (pools 1–10, 104 individuals) to detect differential allelic imbalance between naïve and memory immune cells: naïve versus central memory CD4 + T cells, naïve versus central memory CD8 + T cells, and naïve versus memory B cells. Cell type labels were provided by the original dataset. We used DAESC-P to process the raw sequencing data in SRA format. After completing the pipeline, we obtained a total of 5.7 million nonzero read counts for 28,533 variants across 104 individuals. For each comparison, we first filtered out variants with nonzero read counts in fewer than five individuals. We further filtered out variants with nonzero total expression in less than 1% of all cells. For genes that contained multiple exonic variants, we used the variant with the highest cell-type-specific read depth to represent the gene, ignoring any other variants. Following the preprocessing of the OneK1K dataset, the input data for DAESC-GPU consisted of two large sparse count matrices: one for total read counts and the other for alternative read counts. For benchmarking, we analyzed the data using both the R package and DAESC-GPU. The R package analysis was conducted on the UW Biostatistics servers and Google Colab with 8 CPUs, while DAESC-GPU was run on a single NVIDIA A100 GPU or NVIDIA T4 GPU using Google Colab.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-026-06426-y.

Supplementary Material 1.

Author contributions

G.Q. conceptualized the problem. T.C. conducted all experiments, wrote the Python package, and drafted the manuscript. G.Q. and T.C. reviewed and edited the manuscript. Both authors approved the final manuscript.

Funding

National Human Genome Research Institute award K01HG013983

Data availability
The endoderm differentiation data by Cuomo et al. are available on Zenodo: [https://zenodo.org/records/3625024#.YnJ-ivPMKi4]. Raw single-cell RNA-seq data of Cuomo et al. are available under the accession numbers ERP016000 (ENA project). The OneK1K data is available on Gene Expression Omnibus (GEO) via accession number GSE196830.

Code availability
Project name: DAESC+: High-performance, integrated software for single-cell allele-specific expression data. Project home page: https://github.com/ITCUI-XJTLU/DAESC-GPU . Scripts for bioinformatics processing of the OneK1K data are also available on the GitHub site. Operating system: Linux (for DAESC-P). Programming Language: Bash (for DAESC-P), Python (for DAESC-GPU). Other requirements: Singularity; SLURM (for large-scale parallel execution); GPU (for DAESC-GPU). License: GPL-3.0. Any restriction to use by non-academics: None.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 8 January 2026 / Accepted: 11 March 2026
Published online: 18 March 2026

References

1. Lappalainen T, et al. Transcriptome and genome sequencing uncovers functional variation in humans. Nature. 2013;501:506–11.
2. Castel SE, Levy-Moonshine A, Mohammadi P, Banks E, Lappalainen T. Tools and best practices for data processing in allelic expression analysis. Genome Biol. 2015;16:195.
3. Castel SE, et al. A vast resource of allelic expression data spanning human tissues. Genome Biol. 2020;21:234.
4. Qi G, Battle A. Computational methods for allele-specific expression in single cells. Trends Genet. 2024;40:939–49.
5. Jiang Y, Zhang NR, Li M. SCALE: modeling allele-specific gene expression by single-cell RNA sequencing. Genome Biol. 2017;18:74.
6. Larsson AJM, et al. Genomic encoding of transcriptional burst kinetics. Nature. 2019;565:251–4.
7. Deng Q, Ramsköld D, Reinius B, Sandberg R. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. Science. 2014;343:193–6.
8. Santoni FA, et al. Detection of Imprinted genes by single-cell allele-specific gene expression. The Am J Hum Genetics. 2017;100:444–53.
9. Cuomo ASE, et al. Single-cell RNA-sequencing of differentiating iPS cells reveals dynamic genetic effects on gene expression. Nat Commun. 2020;11:810.
10. Heinen T, et al. scDALI: modeling allelic heterogeneity in single cells reveals context-specific genetic regulation. Genome Biol. 2022;23:8.
11. Mu W, et al. Airpart: interpretable statistical models for analyzing allelic imbalance in single-cell datasets. Bioinformatics. 2022. https://doi.org/10.1093/bioinformatics/btac212.
12. Qi G, et al. Single-cell allele-specific expression analysis reveals dynamic and cell-type-specific regulatory effects. Nat Commun. 2023;14:6317.
13. Jun S-H, et al. Reconstructing clonal tree for phylo-phenotypic characterization of cancer using single-cell transcriptomics. Nat Commun. 2023;14:982.
14. Park D, Cenik C. Long-read RNA sequencing reveals allele-specific N 6-methyladenosine modifications. Genome Res. 2025;35:999–1011.
15. Ramasamy R, et al. Genome-wide allele-specific expression in multi-tissue samples from healthy male baboons reveals the transcriptional complexity of mammals. Cell Genom. 2025;5:100823.
16. Dobin A, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics. 2013;29:15–21.
17. Kaminow, B., Yunusov, D. & Dobin, A. STARsolo: accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data. 2021.05.05.442755 Preprint at 2021. https://doi.org/10.1101/2021.05.05.442755
18. Brüning RS, Tombor L, Schulz MH, Dimmeler S, John D. Comparative analysis of common alignment tools for single-cell RNA sequencing. Gigascience. 2022;11:giac001.
19. RNAseq short variant discovery (SNPs + Indels). *GATK* https://gatk.broadinstitute.org/hc/en-us/articles/360035531192-RNAseq-short-variant-discovery-SNPs-Indels (2023).
20. Delaneau O, Coulonges C, Zagury J-F. Shape-IT: new rapid and accurate algorithm for haplotype inference. BMC Bioinform. 2008;9:540.
21. van de Geijn B, McVicker G, Gilad Y, Pritchard JK. WASP: allele-specific software for robust molecular quantitative trait locus discovery. Nat Methods. 2015;12:1061–3.
22. Darby CA, Stubbington MJT, Marks PJ, Martínez Barrio Á, Fiddes IT. scHLAcount: allele-specific HLA expression from single-cell gene expression data. Bioinformatics. 2020;36:3905–6.
23. Prashant NM, et al. SCReadCounts: estimation of cell-level SNVs expression from scRNA-seq data. BMC Genomics. 2021;22:689.

24. Wilson PC, et al. Multimodal single cell sequencing implicates chromatin accessibility and genetic background in diabetic kidney disease progression. Nat Commun. 2022;13:5253.
25. Leinonen R, Sugawara H, Shumway M, International Nucleotide Sequence Database Collaboration. The sequence read archive. Nucleic Acids Res. 2011;39:D19-21.
26. Katz K, et al. The Sequence Read Archive: a decade more of explosive growth. Nucleic Acids Res. 2022;50:D387–90.
27. Choi K, Raghupathy N, Churchill GA. A Bayesian mixture model for the analysis of allelic expression in single cells. Nat Commun. 2019;10:5188.
28. Fan J, Wang X, Xiao R, Li M. Detecting cell-type-specific allelic expression imbalance by integrative analysis of bulk and single-cell RNA sequencing data. PLoS Genet. 2021;17:e1009080.
29. Yazar S, et al. Single-cell eQTL mapping identifies cell type-specific genetic control of autoimmune disease. Science. 2022;376:eabf3041.
30. Sayers EW, et al. Database resources of the national center for biotechnology information in 2025. Nucleic Acids Res. 2025;53:D20–9.
31. Bandala-Sanchez E, et al. CD52 glycan binds the proinflammatory B box of HMGB1 to engage the Siglec-10 receptor and suppress human T cell function. Proc Natl Acad Sci U S A. 2018;115:7783–8.
32. Tani-ichi S, et al. Interleukin-7 receptor controls development and maturation of late stages of thymocyte subpopulations. Proc Natl Acad Sci U S A. 2013;110:612–7.
33. Ramsköld D, et al. Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. Nat Biotechnol. 2012;30:777–82.
34. Picelli S, et al. Full-length RNA-seq from single cells using Smart-seq2. Nat Protoc. 2014;9:171–81.
35. Bushnell, B. BBMap: a fast, accurate, Splice-Aware Aligner. https://escholarship.org/uc/item/1h3515gn (2014).
36. Hofmeister RJ, Ribeiro DM, Rubinacci S, Delaneau O. Accurate rare variant phasing of whole-genome and whole-exome sequencing data in the UK Biobank. Nat Genet. 2023;55:1243–9.

Publisher's Note