Supplementary text and figures for

"SCAN-ATAC Sim: a scalable and efficient method to simulate single-cell ATAC-seq from bulk-tissue experiments"

Table of Contents

1 Cell Group Selection

Bulk-tissue ATAC-seq for Natural Killer (NK), Common Lymphoid Precursor (CLP), Erythrocytes (ERY), and Monocytes (Mono) were used as a cell group. The selected cell group is meant to resemble the different cell types in real scATAC-seq experiments. The scATAC-seq experiments simulated by our software would inherit these bulk-tissue cell line labels. For benchmarking, only scATAC-seq experiment of the CLP cells were generated, so the other cell lines only contribute to the background of the simulation.

2 Input and Output

The input to our software is a collection of aligned bam files of the cell group. The data preprocessing step also provides intermediate outputs. The intermediate outputs consist of the read coverage of the foreground and background region and the reads intersecting with the foreground and background region for one cell line. The single-cell simulation step then uses the intermediate output to perform the simulation for each cell. The final output is foreground and background reads of every cell in the scATAC-seq experiment for one cell type.

3 Data Preprocessing

3.1 ***Foreground***

Peaks for each cell were called using MACS2 from aligned bam files using the ENCODE ATAC-seq pipeline (Zhang, et al., 2008). For exploratory analysis, the ATAC-seq peak length distribution of several cell types from ENCODE were found to range from 150bp to 5000bp as shown in Figure S1. These foreground peaks will be sampled with probability proportional to their width (and depth). The narrow peaks were merged to define the foreground region.

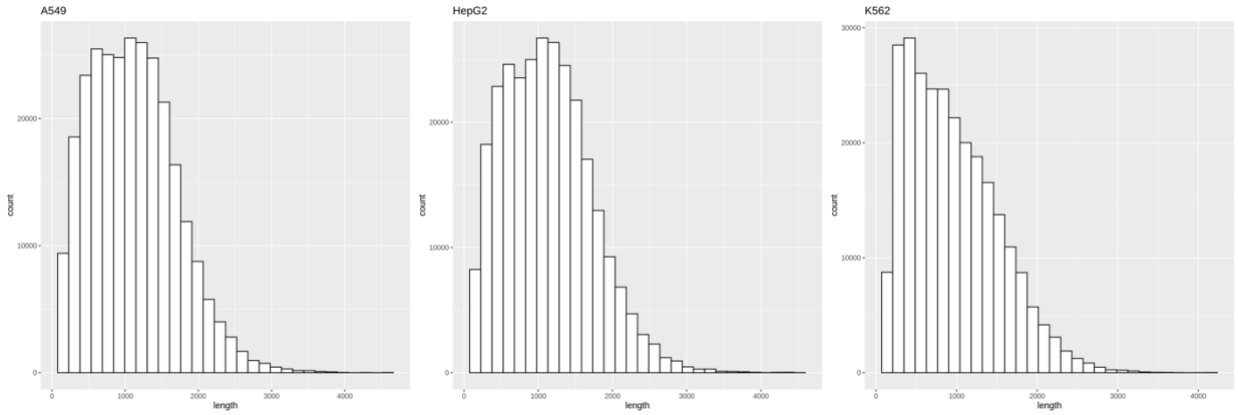

Figure S1. The ATAC-seq peak length distribution of A549, HepG2, K562 (ENCFF734LFV, ENCFF876IFK, ENCFF242PXW) from the ENCODE Project.

3.2 ***Background***

The complement of the 1kb-extended foreground divided in bins of fixed sizes (i.e. 500bp or 1kb) were defined as the background. Hence, to make the background regions, windows of fixed sizes (i.e. 500bp or 1kb) were generated based on the hg38 genome from bedtools makewindows, and the windows that does not overlap with the 1kb-extended foreground regions were considered background regions.

3.3 ***Read Filtering and Processing***

Paired and deduplicated reads for each bulk-tissue ATAC-seq were filtered using the SAMtools (Li, et al., 2009) with Pysam. The filtered reads for each cell-type were then intersected with the foreground regions to obtain the foreground reads using bedtools intersect. The background reads are obtained by intersecting the midpoint of all reads with background regions to prevent overlap between multiple background regions. The intersection was also performed with bedtools intersect using pybedtools. Lastly, this step would output the background read coverage and region-to-read intersection summed across all cell types, but the foreground read coverage and region-to-read intersection is unique for each cell line. The read coverage was obtained using bedtools coverage, and the region-to-read intersection was obtained using bedtools intersect.

4 Single-Cell Simulation

4.1 ***Sampling Regions According to Read Coverage***

A total of N_c regions are sampled for each cell. The number of regions/reads for each cell could vary across cells in single-cell sequencing, and the user can provide the mean and variance for a log-normal distribution for the region number in each cell.

The regions are sampled with read coverage as weights. The foreground and background regions are sampled separately, hence the total allocation of N_c regions is divided into N_c^F and N_c^B by the signal-to-noise ratio ρ. Furthermore, the number of regions for each cell is evenly distributed across the two (maternal and paternal) rounds for the foreground and background ($N_{c,M}^F$ and $N_{c,P}^F$; $N_{c,M}^B$ and $N_{c,P}^B$). Therefore, each allele in the diploid genome could only be sampled twice or less.

The following series of steps are performed to sample regions represented by reads in each cell as shown in Figure 1. First, the application samples $N_{c,M}^F$ foreground regions without replacement. After that, the application samples $N_{c,P}^F$ regions again from a fresh set of foregrounds without replacement. Then, it samples $N_{c,M}^B$ background regions without replacement. Finally, the application samples $N_{c,B}^F$ regions again from a fresh set of backgrounds without replacement.

Here, we make the assumption that the sampling of two alleles (maternal and paternal) on the same locus is independent of one another because a flow cell could sequence hundreds of millions of reads at the same time after amplification, effectively sampling the alleles instantaneously and independently. Previous research on single-cell allelic coverage also supports our assumption (Zhang, et al., 2015).

Weighted reservoir sampling is a novel, efficient algorithm for weighted sampling without replacement. The normalized read coverage of the regions is used as weights for weighted reservoir

sampling without replacement. The main computational cost is iterating through all the regions when simulating every cell. However, exponential jumps accelerate the computation by reducing the reservoir update to only $O(n \cdot log(m))$ where n is the total population of regions and m is the region number. The algorithm also uses a heap to efficiently updates to the reservoir. Since sampling without replacement is a sequential and memory-intensive process, parallelism within each cell is very limited. However, each cell is independent of one another, so light-weight shared memory parallelism across multiple central processing units (CPU) cores with OpenMP achieves a scalable speed up (Dagum and Menon, 1998).

4.2 ***Sampling Reads from Each Selected Region***

Once a set of regions is obtained for each cell, a read is sampled from all reads that intersect with a region under a uniform distribution using the region-to-read intersection, which is an intermediate output from the data preprocessing step. To sample a read for each region, the application uses the region-to-read intersection to construct a dictionary that maps from one region to its intersecting reads. It counts the number of intersecting reads, then it generates a random number between zero and that read count. The number generated is the index of the selected read, and the read is accessed from the dictionary in $O(1)$ time.

The majority of the computational overhead of this step consists of creating the mapping between regions and reads, so the mapping is shared among multiple cores when parallelized with OpenMP.

5 Comparing SCAN-ATAC Sim with Previous Methods

In terms of analysis, our sampling methods provides a more cell-type-specific representation of a scATAC-seq dataset compared to direct down-sampling (Table 1, Fig. 1f). We also allow users to adjust the signal-to-noise ratio specific to the unified background in scATAC-seq for the benchmarking task, whereas Fang et al. 2019 and Xiong et al. 2019 do not. Most importantly, we also provide a read-level simulation suitable for a wide variety of analysis software, whereas Xiong et al. 2019 only selects the representative peak regions.

In terms of runtime, the direct sampling method mentioned in Fang et al. 2019 took 26.6 hours to simulate a total of 1 million cells (with 2000 reads for each cell) while our method took around an hour to simulate the same number of cells under the same computational configuration.

	Direct Down-sampling (Fang et al.)	Peak region sampling (Xiong et al.)	SCAN-ATAC Sim
Read-level simulation	Yes	No	Yes
Flexible signal-to-noise from unified background	No	No	Yes
Diploid genomic constraint	No	No	Yes
Short runtime (<1hr for 1 million cells)	No	No	Yes

Table S1. Comparison between SCAN-ATAC Sim and previous methods.

6 Cell-Type Specificity of Simulated scATAC-seq

In Figure 1f, 100k CLP cells were simulated with only CLP in the group (1000 reads per cell, 0.4 signal-to-noise ratio). Peaks were called for the simulated datasets and the bulk ATAC-seq using MACS2 (-g hs -n CLP.s70 --nomodel --shift -100 --extsize 200 -B --SPMR --keep-dup all --call-summits). As shown in the figure, the foreground and background pileup signals correlate with the peak and non-peak signals from the original bulk ATAC-seq, which demonstrates that simulated data is able to maintain cell-type specificity within the cell type.

Furthermore, 10k CLP and 10k Mono cells were simulated from a group of four cell types (mentioned above) with 3000 reads per cell and a 0.5 signal-to-noise ratio. The pileup signal correlates only with bulk ATAC-seq peaks of its own cell type but not other cell types, further confirming the cell-type specificity of the simulation.

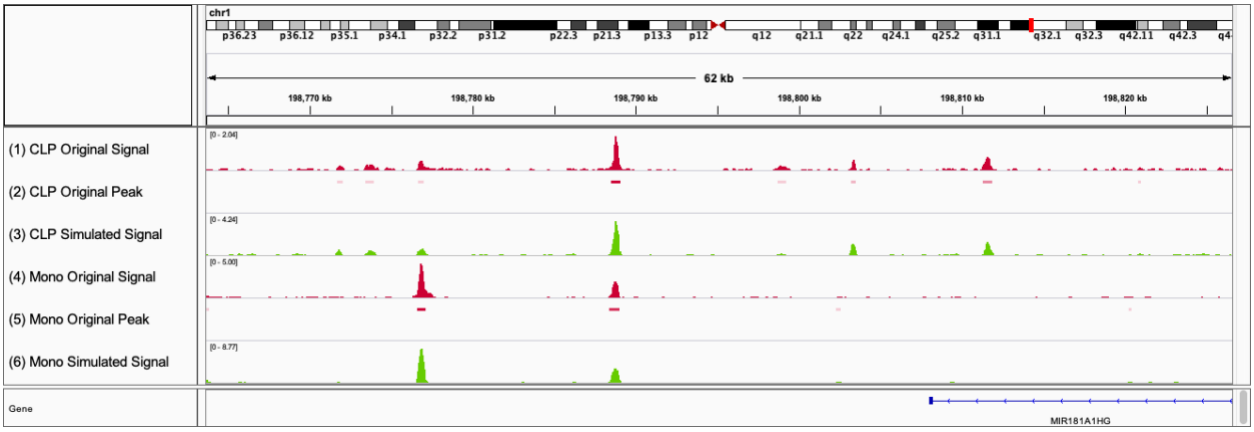

Figure S2. chr1 visualization of bulk and simulated ($\boldsymbol{\rho = 0.5, f = 3k, c = 10k}$) data of two different cell types (CLP and Mono) in Integrative Genome Browser. (1) CLP Bulk ATAC-seq pileup signal. (2) CLP Bulk ATAC-seq peaks. (3) Simulated scATAC-seq pileup signal of 10k simulated CLP cells. (4) Monocyte Bulk ATAC-seq pileup signal. (5) Monocyte Bulk ATAC-seq peaks. (6) Simulated scATAC-seq pileup signal of 10k simulated Monocyte cells.

7 Analysis of Simulated scATAC-seq with SnapATAC

We analyzed various simulated scATAC-seq datasets with SnapATAC (Fang, et al., 2019). The reads are aggregated by cell type. The genome is divided into 5kb bins, and transformed into a binarized matrix by read coverage. Bins with coverage of zero and top 5% of invariant bins were filtered out from the analysis. Diffusion maps were generated, and the top eigen dimensions were selected for further analysis. KNN graphs were constructed using k=15 for Louvain community detection (with a resolution of 0.1).

7.1 ***Signal-to-noise ratio***

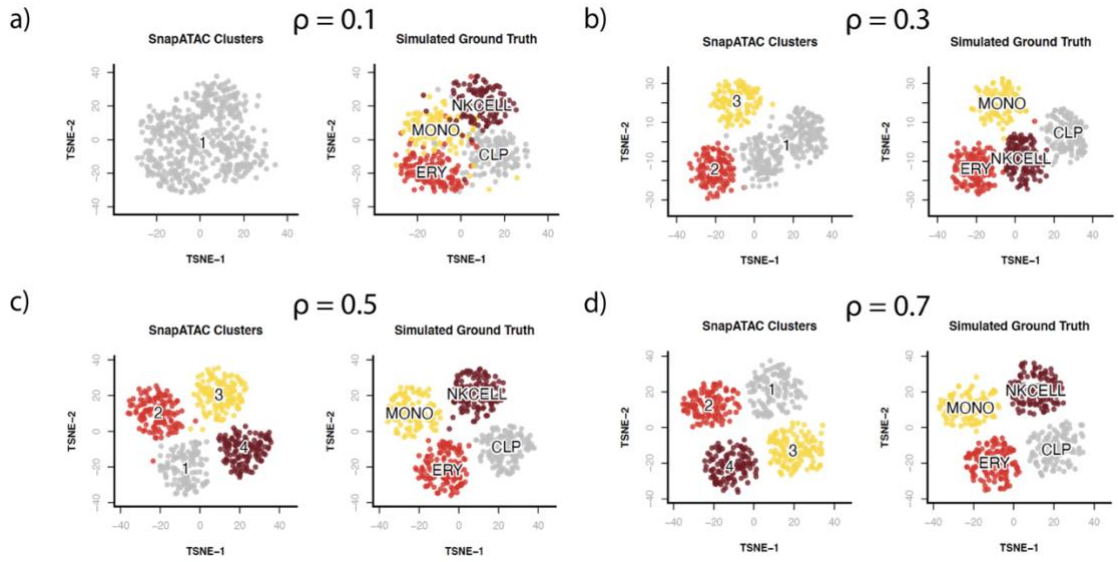

Figure S3. The separability of scATAC-seq simulated from bulk cell-line ATAC-seq under different signal-to-noise ratios. The left graph contains SnapATAC cell cluster labels, the right graph contains ground truth labels. a) ρ = 0.1 b) ρ = 0.3 c) ρ = 0.5 d) ρ = 0.7

7.2 ***Read depth***

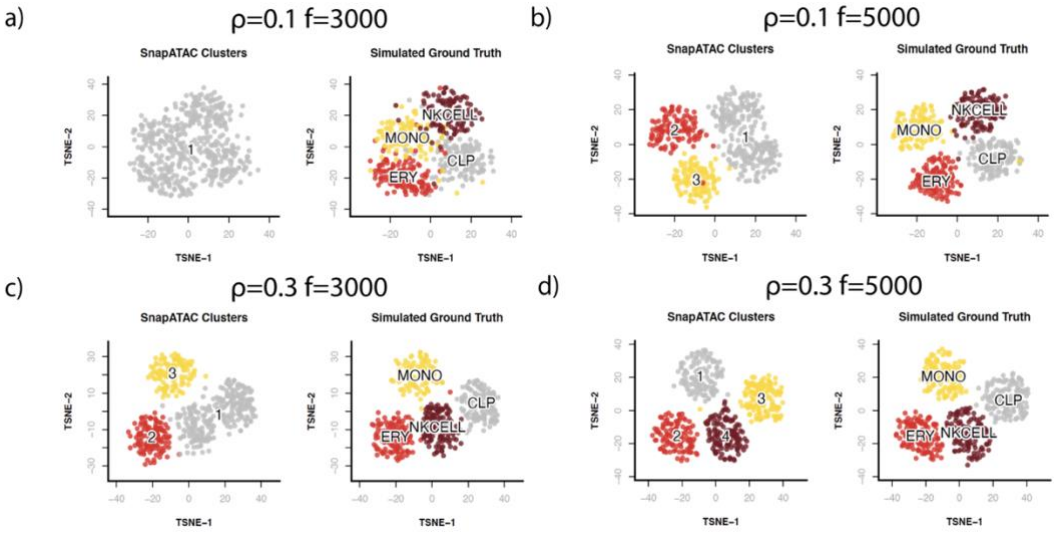

Figure S4. The separability of simulated scATAC-seq under different read depths. The left graph contains SnapATAC cell cluster labels, the right graph contains ground truth labels. a) ρ = 0.1, 3000 reads per cell b) ρ = 0.1, 5000 reads per cell c) ρ = 0.3, 3000 reads per cell d) ρ = 0.3, 5000 reads per cell

7.3 ***Heterogenous Cell Types***

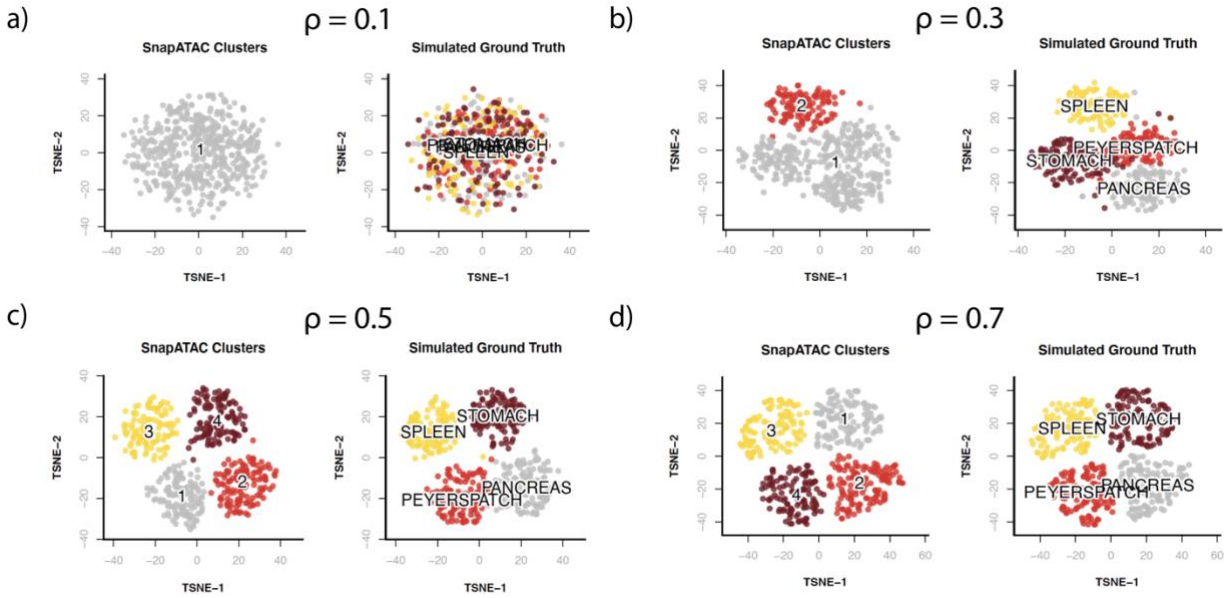

Figure S5. The separability of scATAC-seq simulated from bulk-tissue (Stomach: ENCFF464KVN, Spleen: ENCFF688AAU, Peyer's Patch: ENCFF334IBE, Pancreas: ENCFF414FVH) ATAC-seq under different signal-to-noise ratios. The left graph contains SnapATAC cell cluster labels, the right graph contains ground truth labels. a) ρ = 0.1 b) ρ = 0.3 c) ρ = 0.5 d) ρ = 0.7

8 Downloading and Using the SCAN-ATAC Sim software

The source code and the binary executables are available for download at scan-atac-sim.gersteinlab.org.

8.1 Quick Start

Simulation can be quick started by running the wrapper script:

```
./run.sh --cell_types CLP,Ery,Mono,NKcell --peak_dir ./peak/ --bam_dir ./bam/ -t ./temp/ -o ./output/
```

In the example above, the script will look for CLP.narrowPeak, Ery.narrowPeak, Mono.narrowPeak, NKcell.narrowPeak files in the ./peak/ folder and CLP.bam, Ery.bam, Mono.bam, NKcell.bam files in the ./bam/ folder.

The default parameters for the simulation are:

Short Flag	Long Flag	Name	Default Value
-v	--variance	Variance of read coverage across cells	0.5
-c	--cell_number	Number of cells simulated from each cell type	10,000
-s	--signal_to_noise	Signal-to-noise ratio	0.7
-f	--frag_num	Average Number of reads per cell	3000
-i	--min_frag	Minimum number of reads per cell	1000
-a	--max_frag	Maximum number of reads per cell	20,000
-l	--cell_types	Cell types	N/A
-e	--extend_peak_size	Length of regions between foreground and background	1000
-b	--bin_size	Background bin size	1000

Table S2. Full list of parameters that affect the simulation

8.2 Custom Preprocessing

The data preprocessing step is performed with our python script:

```
python preprocessing.py -c <comma separated file prefixes for each cell type> -i <directory containing the peak files with prefix and .narrowPeak suffix> -j <directory containing the bam files with prefix and .bam suffix> -e <custom configuration for bp length separating between foreground and background fragments> -b <bp length for background fragment size> -o <output directory>
```

The dependencies for the data processing python script include python version 3.6, Pysam version 0.15.4, numpy version 1.16.4, pandas version 1.0.3, pyBigWig version 0.3.17, wget version 3.2, and pybedtools version 0.8.0. The source code could be found at http://www.scan-atac-sim.gersteinlab.org/#/Preprocess.

8.3 Custom Single-Cell Simulations

The single-cell simulation step is divided into two parts. The sampling of regions using weighted reservoir sampling is performed with the script `weighted_sampling`, and the sampling of reads with a uniform distribution is performed with the script `uniform_sampling`. Both scripts and their source code could be found at http://www.scan-atac-sim.gersteinlab.org/#/Simulation. It requires the intel c++ compiler preinstalled.

```
./weighted_sampling -f NKcell.peak_counts.bed -b bg_counts.bed -of NKcell.foreground.sampled.bed -ob
```

```
NKcell.background.sampled.bed -n 2000 -nv 0.5 -c 10000 -s 0.6 -
min 1000 -max 20000
```

The previous example for weighted reservoir sampling would sample the foreground and background regions of the NK cells using read coverage from data preprocessing. The number of regions for each cell follows a log-normal distribution with a mean of 2000 as specified by the -n flag and a variance of 0.5 as specified by the -nv flag, with a minimum (-min) of 1,000 and a maximum (-max) of 20,000. The number of cells simulated is 10,000 with the -c parameter, and the signal to noise is 0.6 with the -s flag. The output of the foreground and background regions are specified by the -of and -ob flag, respectively. The script automatically detects for the number of cores available for parallelization, but could also be manually set using environment variables, i.e. `OMP_NUM_THREADS=4` would limit to 4 cores.

```
   ./uniform_sampling                    NKcell.peak_intersect.bed
NKcell.foreground.sampled.bed NKcells.foreground.bed

   ./uniform_sampling                    bg_intersect.expanded.bed
NKcell.background.sampled.bed NKcells.background.bed

   cat     NKcells.foreground.bed     NKcells.background.bed     >
NKcells.bed
```

The foreground and background reads are sampled for NK cells with the selected regions and the region-to-read intersection. Lastly, the foreground and background reads are combined into a single file. The output file would contain reads of all cells from that cell line.

The process is then repeated for other cell lines in the cell group to obtain scATAC-seq experiments with different cell types.

References

Dagum, L. and Menon, R. OpenMP: An industry standard API for shared-memory programming. *Ieee Comput Sci Eng* 1998;5(1):46-55.

Fang, R.*, et al.* Fast and Accurate Clustering of Single Cell Epigenomes Reveals Cis-Regulatory Elements in Rare Cell Types. *bioRxiv 615179 [**Preprint**]* 2019.

Li, H.*, et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 2009;25(16):2078-2079.

Zhang, C.Z.*, et al.* Calibrating genomic and allelic coverage bias in single-cell sequencing. *Nat Commun* 2015;6:6822.

Zhang, Y.*, et al.* Model-based analysis of ChIP-Seq (MACS). *Genome Biol* 2008;9(9):R137.