Working with (big) single-cell (ATAC) datasets

scATAC-seq: attacking open chromatin in single cells (2024)

Christian Arnold

10.04.2024

EMBL

General ways of analyzing single-cell (ATAC) data

- Using all-in-one approaches
  - As provided by the technology / producer (here: **Cell Ranger ATAC from 10x**)
  - General preprocessing software
    - R: *ArchR*, *Signac*, *ChrAccR*, *destin*, *ATACseqQC*, *esATAC*, …
    - Python: *PUMATAC*, *scATACpipe*, *MAESTRO*, *scATAC-pro*, *snapATAC2*, *EpiScanpy*, *muon* …
- Using a custom set of programs that are executed in a particular order (workflow)

Single-cell ATAC processing using Cell Ranger ATAC count

Overview with all details:
https://support.10xgenomics.com/single-cell-atac/software/pipelines/latest/algorithms/overview

Main steps:

- Data demultiplexing (Illumina *bcl2fastq*, *BCL Convert* or *Cell Ranger mkfastq or* Element Biosciences *Bases2Fastq*)
- Barcode Processing (barcode detection, filtering and correction)
- **Alignment**
- Duplicate marking
- **Peak calling**
- **Cell calling**
- Peak-barcode matrix
- Additional (optional) downstream analyses:
  - Dimensionality reduction, clustering, and visualization
  - Peak annotation
  - Transcription factor motif enrichment analysis
  - Differential accessibility analysis
  - Aggregation
  - Chemistry batch correction

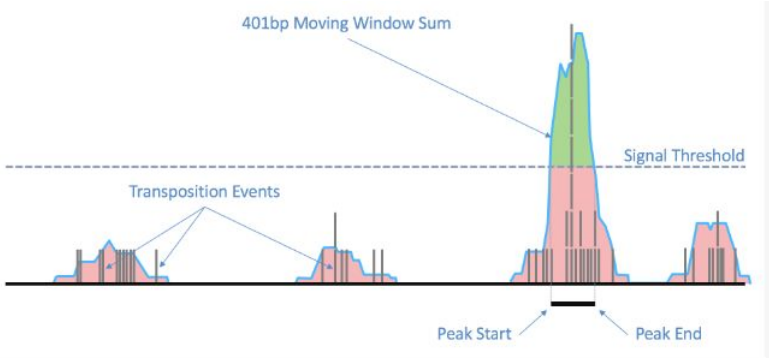
What is Cell Ranger ATAC?

Cell Ranger ATAC is a set of analysis pipelines that process Chromium Single Cell ATAC data. Cell Ranger ATAC includes four pipelines relevant to Single Cell ATAC experiments:

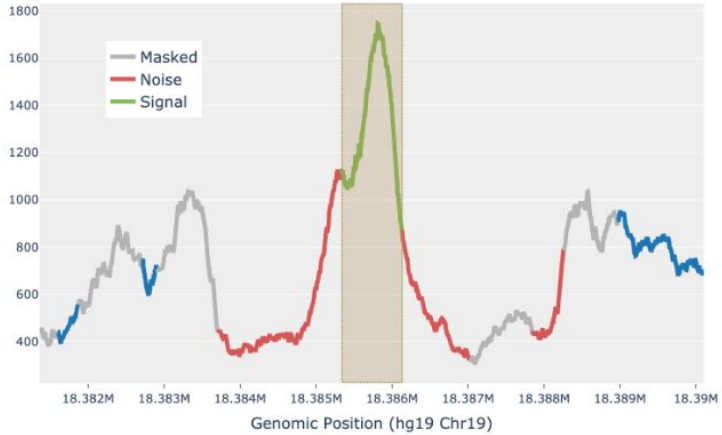
- **cellranger-atac mkfastq** demultiplexes raw base call (BCL) files generated by Illumina® sequencers into FASTQ files. It is a wrapper around `bcl2fastq` from Illumina®, with additional useful features that are specific to 10x Genomics libraries and a simplified sample sheet format.
- **cellranger-atac count** takes FASTQ files from `cellranger-atac mkfastq` and performs ATAC analysis, including:
  - Read filtering and alignment
  - Barcode counting
  - Identification of transposase cut sites
  - Detection of accessible chromatin peaks
  - Cell calling
  - Count matrix generation for peaks and transcription factors
  - Dimensionality reduction
  - Cell clustering
  - Cluster differential accessibility
- **cellranger-atac aggr** aggregates and analyzes the outputs from multiple runs of `cellranger-atac count` (such as from multiple samples from one experiment) by performing the following steps:
  - Normalization of input runs to same median fragments per cell (sensitivity)
  - Detection of accessible chromatin peaks
  - Count matrix generation for peaks and transcription factors for the aggregate data
  - Dimensionality reduction
  - Cell clustering
  - Cluster differential accessibility
  - Chemistry batch correction
- **cellranger-atac reanalyze** takes the analysis files produced by `cellranger-atac count` or `cellranger-atac aggr` and reruns secondary analysis with tunable parameter settings:
  - Cell calling
  - Dimensionality reduction
  - Cell clustering
  - Cluster differential accessibility

Output is delivered in standard BAM, MEX, CSV, TSV, HDF5 and HTML formats that are augmented with cellular information.

Peak calling: The Cell Ranger ATAC way



Above: Raw transposition events are used to produce a local smoothed signal track with a 401bp moving window sum. After fitting and selecting a global peak threshold, contiguous regions with signal above the threshold (shown in green) are produced as candidate peak calls.



Above: a diagram of how the local signal-to-noise estimate is performed for a single putative peak in a candidate region. The green section of the signal shows the putative peak under examination, with the peak signal measured as the median value across the green section. The grey sections are masked out, as they are other putative peaks and so are not used to estimate the local background. The red sections are used for local background estimates, with the peak background as the median value across all red sections.

Many more details how this is done exactly on the 10x website (https://support.10xgenomics.com/single-cell-atac/software/pipelines/latest/algorithms/overview#peak-bc)

EMBL

CellRanger output files

- For a full list, see https://support.10xgenomics.com/single-cell-atac/software/pipelines/latest/output/singlecell

- BAM file
  - *possorted_bam.bam*: Barcode-corrected reads aligned to the user-specified reference, sorted by reference position + index
- Fragment files
  - *fragments.tsv.gz*: BED-like tabular file, where each line represents a unique ATAC-seq fragment captured by the assay. Each fragment is created by two separate transposition events, which create the two ends of the observed fragment (+index)
- Peaks
  - *peaks.bed*: BED file with the output of the peak calling algorithm. Each peak is an genomic interval that has a local enrichment of transposase cut-sites, and has a corresponding row in the feature-barcode matrix output files. Peaks are the primary 'features' that are measured for each cell. Numerically sorted and non-overlapping
- Feature-Barcode Matrices
  - 3 types of feature-barcode matrices in Market Exchange Format (MEX) format: (un)filtered peak-barcode matrix + filtered TF-barcode matrix. The unfiltered matrix contains every observed barcode including background and non-cellular barcodes, while the filtered contains only detected cellular barcodes

CellRanger output files (2)

- Additional output files:
  - Web summary and analyses
  - per Barcode QC, ATAC Signal, and Cell Calling
  - Summary Metrics
  - Peak Annotations

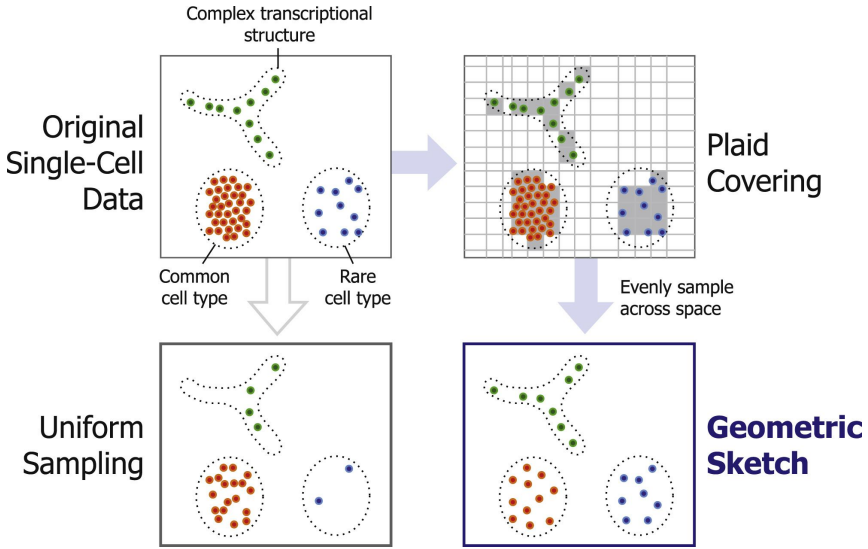
- Most big data are block-gzipped (.gz) to allow indexing and to save disk space. For example, the *fragments.tsv.gz.tbi* file is a tabix index of the fragment intervals facilitating random access to records from an arbitrary genomic interval

Computational strategies to deal with datasets with many cells

- Find a big enough computer / server and time to analyze all cells at once
- Use newest available tools, versions, formats
- Reduce the size of your dataset
  - Main challenge: preserve the heterogeneity of the original dataset
  - 1. Data subsampling: Select among existing cells
  - 2. Metacell / pseudobulking approaches: Create new cells

1. Data subsampling (sketching) methods

- sketch-based techniques: select a subset of **representative** cells for analysis instead of all cells
- cells are not selected simply randomly



Taken from Hie et al. (2019)

EMBL

1. Data subsampling (sketching) methods: Example frameworks

- *Seurat v5 data sketching*
  - diversity-preserving: oversample rare populations, retaining the biological complexity of the sample while drastically compressing the dataset (Hao et al., 2022)
  - a subset of cells are stored in memory to enable rapid and iterative exploration, while the remaining cells are stored on-disk. Users can flexibly switch between both data representations
- *sketchR*
  - provides a simple interface to the *geosketch* and *scSampler* python packages, which implement subsampling algorithms described in Hie et al. (2019) and Song et al. (2022), respectively

2. Metacell approaches

- combines cells to “meta” cells to decrease scarcity and reduce dataset dimensions
- general pseudobulk approaches
- Single-modality
  - seaCells
  - MetaCells-2
  - SuperCell
- Multi-modality
  - utilize an embedded space leveraging the information of multiple modalities to perform aggregation
  - (theoretically) COEM (https://arxiv.org/abs/2207.07734)

EMBL