A guide to the single-cell epigenomics analysis

Kai Zhang

Table of contents

Preface

This book is used to complement the documentation of the SnapATAC2 Python/Rust package.

1 Introduction

This is a book created from markdown and executable code.

2 Input data format

SnapATAC2 accepts BAM or BED-like tabular file as input. The BED-like tabular file can be used to represent fragments (paired-end sequencing) or insertions (single-end sequencing). BAM files can be converted to BED-like files using `snapatac2.pp.make_fragment_file`.

2.1 Fragment interval format

Fragments are created by two separate transposition events, which create the two ends of the observed fragment. Each unique fragment may generate multiple duplicate reads. These duplicate reads are collapsed into a single fragment record. **A fragment record must contain exactly five fields**:

1. Reference genome chromosome of fragment.
2. Adjusted start position of fragment on chromosome.
3. Adjusted end position of fragment on chromosome. The end position is exclusive, so represents the position immediately following the fragment interval.
4. The cell barcode of this fragment.
5. The total number of read pairs associated with this fragment. This includes the read pair marked unique and all duplicate read pairs.

During data import, a fragment record is converted to two insertions corresponding to the start and end position of the fragment interval.

2.2 Insertion format

Insertion records are used to represent single-end reads in experiments that sequence only one end of the fragments, e.g., Paired-Tag experiments. While fragment records are created by two transposition events, insertion records correspond to a single transposition event.

Each insertion record must contain six fields:

1. Reference genome chromosome.
2. Adjusted start position on chromosome.
3. Adjusted end position on chromosome. The end position is exclusive.

4. The cell barcode of this fragment.
5. The total number of reads associated with this insertion.
6. The strandness of the read.

During data import, the 5' end of an insertion record is converted to one insertion count.

Note: in both cases, the fifth column (duplication count) is not used during reads counting. In other words, we count duplicated reads only once. If you want to count the same record multiple times, you need to duplicate them in the input file.

3 Dimension reduction

Single-cell ATAC-seq (scATAC-seq) produces large and highly sparse cell by feature count matrix. Working directly with such a large matrix is very inconvinent and computational intensive. Therefore typically, we need to reduce the dimensionality of the count matrix before any downstream analysis. Most of the counts in this matrix are very small. For example, ~50% of the counts are 1 in deeply sequenced scATAC-seq data. As a result, many methods treat the count matrix as a binary matrix.

Different from most existing approaches, the dimension reduction method used in SnapATAC2 is a pairwise-similarity based method, which requires defining and computing similarity between each pair of cells in the data. This method was first proposed in (Fang et al. 2021), the version 1 of SnapATAC, and was called "diffusion map". In SnapATAC2, we reformulate this approach as spectral embedding, *a.k.a.*, Laplacian eigenmaps.

3.1 Spectral embedding

Start with $n \times p$ cell by feature count matrix M, we first compute the $n \times n$ pairwise similarity matrix S such that $S_{ij} = \delta(M_{i*}, M_{j*})$, where $\delta : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is the function defines the similarity between any two cells. Typical choices of δ include the jaccard index and the cosine similarity.

We then compute the normalized graph Laplacian $L = I - D^{-1/2} S D^{-1/2}$, where I is the identity matrix and D is a diagonal matrix such that $D_{ii} = \sum_k S_{ik}$.

The eigenvectors correspond to the k+1-smallest eigenvalues of L are selected as the lower dimensional embedding.

3.2 Nyström method

For samples with large numbers of cells, computing the full similarity matrix is slow and requires a large amount of memory. To address this limitation and increase the scalability of spectral embedding, we used the Nystrom method to perform a low-rank approximation of the full similarity matrix.

We will be focusing on generating an approximation $\tilde{S}$ of S based on a sample of $l \ll n$ of its columns.

Suppose $S = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$ and columns $\begin{bmatrix} A \\ B^T \end{bmatrix}$ are our samples. We first perform eigendecomposition on $A = U\Lambda U^T$. The nystrom method approximates the eigenvectors of matrix S by $\tilde{U} = \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix}$.

We can then compute $\tilde{S}$:

$$\begin{aligned} \tilde{S} &= \tilde{U}\Lambda\tilde{U}^T \\ &= \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix} \Lambda \begin{bmatrix} U^T & \Lambda^{-1} U^T B \end{bmatrix} \\ &= \begin{bmatrix} U\Lambda U^T & U\Lambda\Lambda^{-1}U^T B \\ B^T U \Lambda^{-1} \Lambda U^T & B^T U \Lambda^{-1} \Lambda \Lambda^{-1} U^T B \end{bmatrix} \\ &= \begin{bmatrix} A & B \\ B^T & B^T U \Lambda^{-1} U^T B \end{bmatrix} \end{aligned}$$

In practice, $\tilde{S}$ does not need to be computed. Instead, it is used implicitly to estimate the degree normalization vector:

$$\tilde{d} = \tilde{S}\mathbf{1} = \begin{bmatrix} A\mathbf{1} + B\mathbf{1} \\ B^T\mathbf{1} + B^T A^{-1} B\mathbf{1} \end{bmatrix}$$

References

Fang, Rongxin, Sebastian Preissl, Yang Li, Xiaomeng Hou, Jacinta Lucero, Xinxin Wang, Amir Motamedi, et al. 2021. "Comprehensive analysis of single cell ATAC-seq data with SnapATAC." *Nature Communications* 12 (1): 1337. https://doi.org/10.1038/s41467-021-21583-9.