Supplementary Methods

Detailed description of our pipeline for APA analysis from 3' tag scRNA-seq data

Scripts implementing the pipeline and input sample files are available from https://github.com/ElkonLab/scAPA

Our pipeline consists of the following 5 steps:

1. Defining 3'UTR peaks
2. Quantifying the usage of each peak in each cell cluster
3. Filtering peaks
4. Detecting dynamical APA events
5. Inferring global trends of APA modulation

Input Files:

- Alignment BAM files generated by cellranger count, for each of the experiment's *n* samples: Aligned$_1$.BAM Aligned$_2$.BAM … Aligned$_n$.BAM
- Cell cluster annotation files: each file contains the cell barcodes that belongs to each cell cluster *j*, from sample *i* (Cluster$_{ji}$.txt).

We provide an R shell script that automatically runs all the analysis steps. The following text describes in detail the commands and tools used for the first two steps of the analysis and more general description of the other three steps ran by the pipeline.

1. Defining 3'UTR peaks

The input for this step is the alignment BAM files, generated by cellranger count, for each of the experiment's *n* samples: Aligned$_1$.BAM Aligned$_2$.BAM … Aligned$_n$.BAM

a. PCR duplicates removal

i. PCR duplicates are removed using UMI-tools *dedup*. As UMI tools dedup requires that each line in the BAM file has a molecular barcode tag, Drop-seq tools is first used to filter the BAMs, leaving only reads for which cell ranger counts produced corrected molecular barcode tag.

```
FilterBAM TAG_RETAIN=UB  I=Aligned_i.BAM O= UB.Aligned_i.BAM
```

ii. Then UMI tools is ran with “method=unique” so that cellranger corrected molecular barcodes are used:

```
umi_tools dedup -I UB.Aligned_i.BAM -S dedup.Aligned_i.BAM --method=unique --extract-umi-method=tag --umi-tag=UB --cell-tag=CB
```

b. Peak detection

i. Homer is used to create a tag directory (Tagdirectory) from the PCR duplicate removed BAMs:

```
makeTagDirectory Tagdirectory dedup.Aligned_1.BAM dedup.Aligned_2.BAM …
```

ii. Homer findPeaks is used to identify peaks. By default, findPeaks adjusts reads to the center of their fragment. To avoid this, fragLength is set to the average read length. In order to find peaks of variable width, Homer is set to find peaks of width 50nt and a minimum distance of 1 nt between peaks.

```
findPeaks Tagdirectory -size 50 -fragLength 100 -minDist 1 -strand separate -o
Peakfile
```

iii. Bedtools is used to merge peaks less than 100 nt apart

```
mergeBed -d 100 -s –i  peakfile > merge.peakfile
```

iv. Intersect peaks file with a 3' UTR bed file to create a GTF of the 3’ UTR peaks:

```
bedtools intersect -wa -wb -s -a merge.peakfile -b 3UTR.BED
```

The output file is edited to produce a valid bed file (peaks.BED) where the peaks are annotated according to their 3’ UTR and their position within it.

c. Separating Peaks with bimodal UMI counts distribution

Adjacent pA sites may result in a single peak. To detect and separate such peaks the R package *mclust* is used to fit a Gaussian finite mixture model with two components to the UMI counts distribution in the interval of each peak. The input to mclust, the UMI counts distribution, is prepared as follows:

i. To detect reads from the union of all reads from all samples, the duplicate-removed BAM files are merged.

```
samtools merge merged. Aligned.BAM  dedup.Aligned_1.BAM
dedup.Aligned_2.BAM …
```

ii. Two BEDGRAPHS files are produced from this BAM, one for the plus strand and one for the minus strand, using bedtools genomcove

```
bedtools genomcov -strand +(-) -bg -ibam merged. Aligned.BAM  >
covrage.plus(minus).wig
```

iii. The BEDGRAPHS files are converted into a BED format and bedtools intersect is used to intersect them with the peaks' BED file.

iv. The intersected file is read in R and converted to a list such that each element of the list, corresponding to a specific peak, is a numeric vector whose values represent read coverage observed across the peak.

v. mclust is used to fit an equal variance Gaussian finite mixture model with two components to (G=2, modelNames="E") to each list element.

vi. If the predicted means of the two fitted Gaussian components are separated by more than three standard deviations and at least 75 nt, the peak is split into two, according to mclust classification.

vii. The peak's bed is edited accordingly (Correct peak index).

2. Quantifying the usage of each peak in each cell cluster

This step uses *featureCounts* to count the reads that overlap each peak in each cluster ("cell type").

i. A separate BAM file for each cell cluster is generated. First, Drop-seq tools is used to split the reads in each sample BAM into separate BAMs that correspond to the different clusters. This is done using *FilterBAMByTag* together with text files (Cluster$_{ji}$.txt), where each file contains the cell barcodes that belongs to each cell cluster *j*, from sample *i*.

FilterBAMByTag TAG=CB TAG_VALUES_FILE= Cluster$_{ji}$.txt I= dedup.Aligned$_{i}$.BAM O= Cluster$_{ji}$.BAM

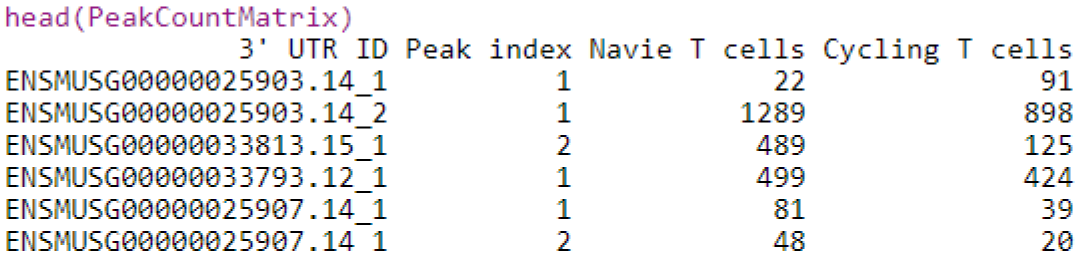
ii. Next, for each cluster *j* all *n* files corresponding to this cluster are merged to produce one BAM file for that cluster (Cluster$_{j}$ .BAM):

samtools merge Cluster$_{j}$ .BAM Cluster$_{j1}$.BAM Cluster$_{j2}$.BAM Cluster$_{jn}$.BAM …

iii. Rsubread package *featureCounts* function is used, where the annotation file is a SAF data.frame edited from the peaks bed file (peaks.SAF). largestOverlap = True is specified so that reads spanning two peaks are counted according to their largest overlap.

featureCounts(files = Cluster$_{1}$ .BAM Cluster$_{2}$ .BAM …, annot.ext = peaks.SAF, largestOverlap = T)

The result is a count matrix, were the rows are peaks, and columns are cell clusters.



```
head(PeakCountMatrix)
                3' UTR ID Peak index Navie T cells Cycling T cells
ENSMUSG00000025903.14_1            1            22              91
ENSMUSG00000025903.14_2            1          1289             898
ENSMUSG00000033813.15_1            2           489             125
ENSMUSG00000033793.12_1            1           499             424
ENSMUSG00000025907.14_1            1            81              39
ENSMUSG00000025907.14_1            2            48              20
```

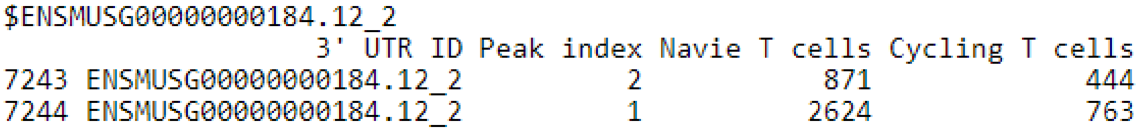
3. Peak filtering

a. First, in each dataset, after conversion of peak counts to counts-per-million (CPM) units, only peaks whose total sum over all cell clusters is above 10 CPMs are considered.

b. To exclude internal priming suspected peaks, peaks having a stretch of at least 8 consecutives As in the region between 10 nt to 140 nt. to the peak's end are filtered.

4. Statistical analysis – detection of dynamic APA events between cell clusters

a. Each 3’UTR with more than one peak is represented by a table where rows are peak indices and columns cell clusters, e.g.:



```
$ENSMUSG00000000184.12_2
                   3' UTR ID Peak index Navie T cells Cycling T cells
7243 ENSMUSG00000000184.12_2          2           871             444
7244 ENSMUSG00000000184.12_2          1          2624             763
```

For each such table we perform a Chi-squared test

b. p-values are corrected for multiple testing using BH FDR.

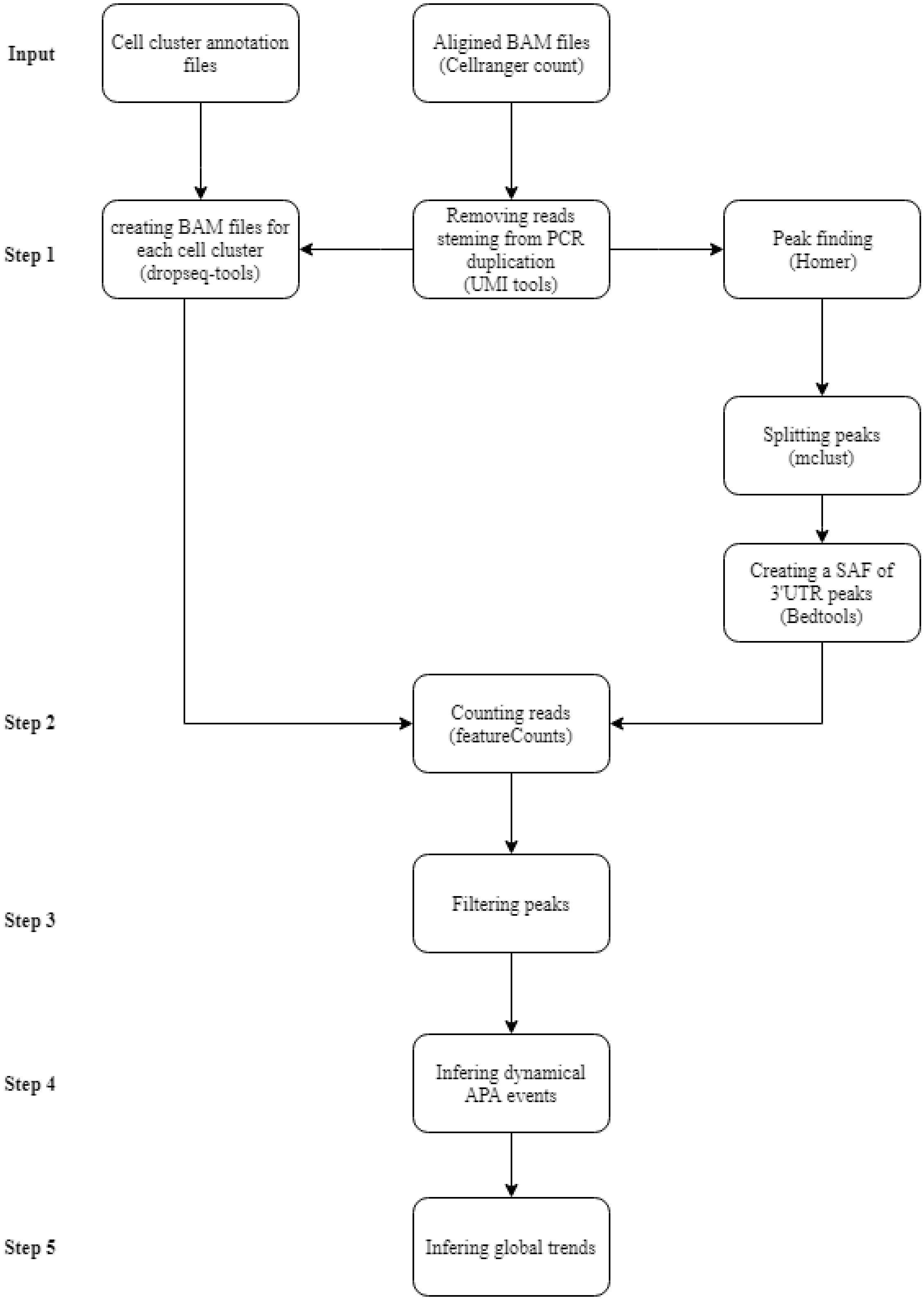
5. Inferring global trends of APA modulation

a. The proximal pA site usage index (*proximal PUI*) is used to quantify the relative usage of the most proximal pA site within a 3' UTR (with two or more peaks). For a given 3' UTR, the proximal PUI is defined by:

$$proximal\ PUI = \log_2\left(\frac{C_1+1}{<C+1>}\right),$$

where C_1 is the read count of the proximal peak, and $< C >$ is the geometric mean of the counts of all the peaks associated with the 3' UTR. To avoid zeros in the denominator and in the log function, a pseudo count of 1 is add to all before calculating the PUI.

b. For 3' UTR with more than two peaks that show significant usage change, for each peak *i*, chi-square test for goodness of fit is performed.


Input
Cell cluster annotation files
Aligined BAM files (Cellranger count)
Step 1
creating BAM files for each cell cluster (dropseq-tools)
Removing reads steming from PCR duplication (UMI tools)
Peak finding (Homer)
Splitting peaks (mclust)
Creating a SAF of 3'UTR peaks (Bedtools)
Step 2
Counting reads (featureCounts)
Step 3
Filtering peaks
Step 4
Infering dynamical APA events
Step 5
Infering global trends