Page 1 Outline of the data analysis and data flow in the manuscript

A high throughput sequence
Nova-seq 6000

↓

Output reads (Fastq files)

↓

Shell script: 001_Data_Prep.sh

See Page2.

What this script Do?

Data preparation for all subsequent analysis in this manuscript.

1. Adaptor trimming
2. Demultiplexing antibody barcodes and cell barcodes
3. Mapping to the genome
4. Removing amplification duplicates

Shell script: 002_Fig1a-1b.sh

See Page3.

What this script Do?

This shell script calculates average of read distributions of 1st , 2nd and 3rd experiments in regions where signals were detected in 1st experiment. The generated data were used in Figure 1a-1b.

Shell script: 003_Fig1c-1f_Bootstrap.sh

See PageX.

What this script Do?

This shell script identifies signal-enriched regions and putative signals of H3K27ac and H3K27me3 in each single cell without data aggregation using bootstrap test. The generated data were used in Figure 1c-1f.

Shell script: 004_Fig3a-3d.sh

See PageX.

What this script Do?

This script identifies statistically significantly signal-enriched promoters and enhancers by statistical analysis, bootstrap test compare to random control of the putative signals. The generated data were used in Figure 3a-3d.

Shell script: 005_Supplementary_Fig4_Bootstrap.sh

See PageX.

What this script Do?

This shell script identifies signal-enriched regions and putative signals of Med1 and 5hmC in each single cell with data aggregation using bootstrap test. The generated data were used in Supplementary Figure 4.

Shell script: 007_Fig4a-4b.sh

Shell script: 006_Fig3e-3h.sh

See PageX.

What this script Do?

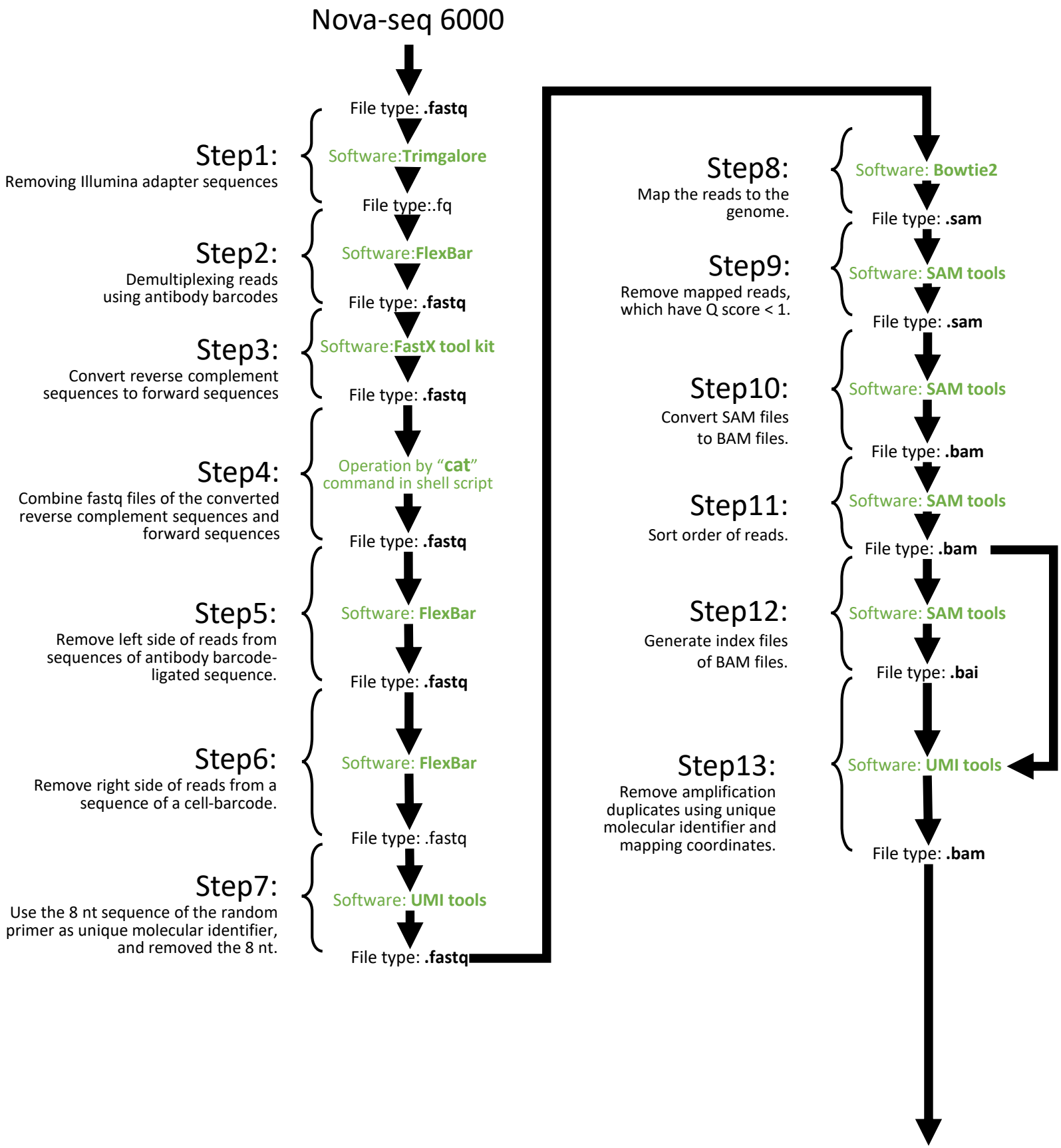
Classifying enhancers based on relative ratio Log2(H3K27ac/H3K27me3), and calculate average of putative signals of H3K27ac, H3K27me3, Med1 and 5hmC in the classified enhancers.

Shell script: 008_Fig4c-4g.sh

Shell script: 009_Fig5a.sh

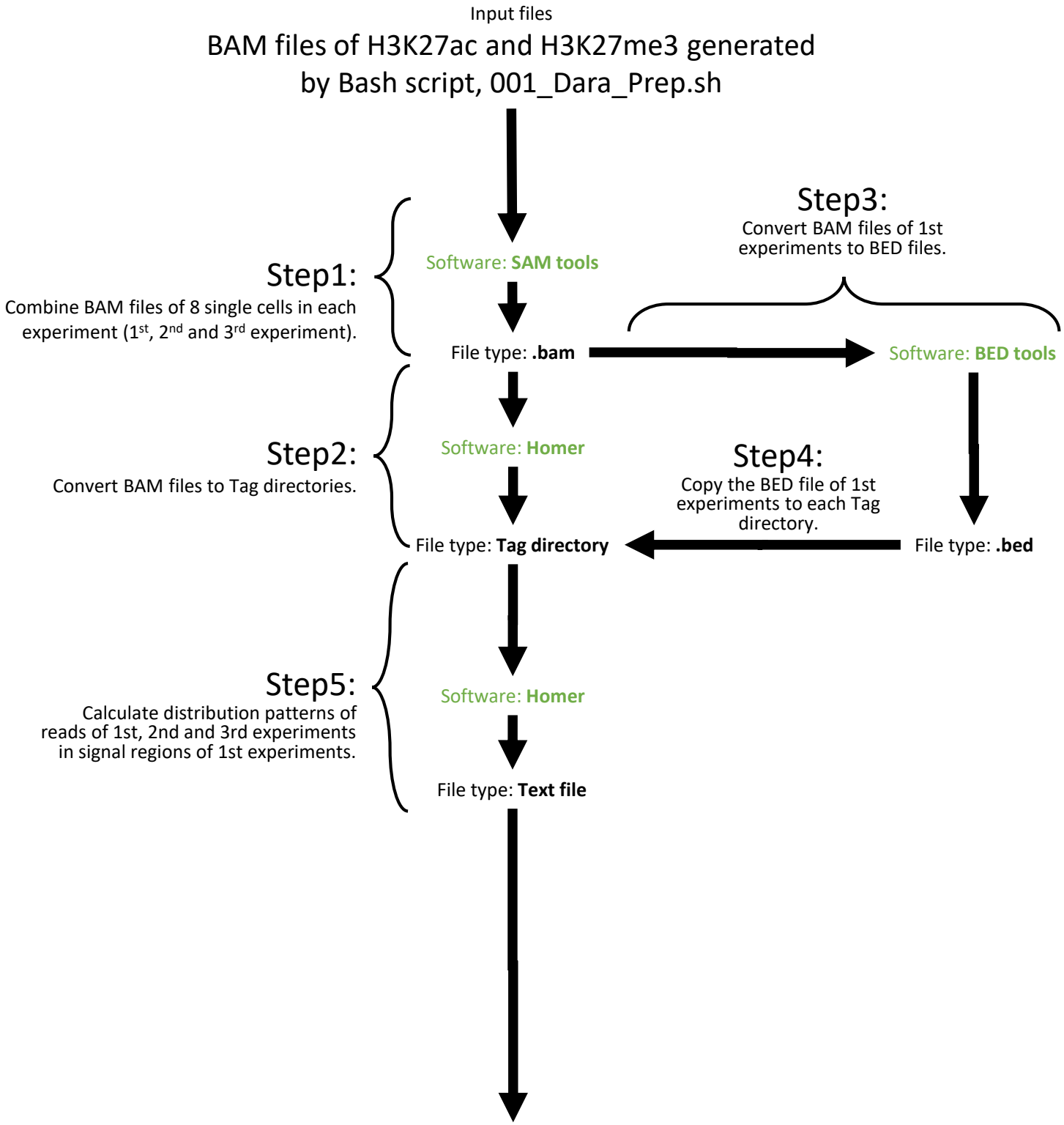
Shell script: 010_Fig5b-5e+Supplementary_Fig5a-5c.sh

Bash script: **001_Dara_Prep.sh**



The generated BAM files were used in the subsequent data analysis of all figures.

Bash script: **002_Figure2a-2b.sh**



Bash script: **003_Fig2c-2f_Bootstrap.sh**

Input files

BAM files of H3K27ac, H3K27me3 and control IgG generated by Bash script, 001_Dara_Prep.sh

Step1:
Combine BAM files of 1st, 2nd and 3rd experiments derived from the same single cell

Software: **SAM tools**

File type: **.bam**

Step2:
Convert the combined BAM files to BED files.

Software: **BED tools**

File type: **.bed**

Step3:
Generate randomized controls from H3K27ac, H3K27me3 and control IgG of each cell.

Software: **BED tools**

File type: **.bed**

Step4:
Generate genome bins, 500bp bin sliding 250 bp.

Input file: **GRCh38.genome**

Software: **BED tools**

Output file: Genome_bin_500bp_250 bp_sliding_GRCh38.bed

Step5:
Count reads in the bins.

Software: **BED tools**

File type: **.bed**

Step6:
Remove genomic bins, which do not contain single-cell reads.

Linux command "**cat**" & "**awk**"

File type: **.bed**

Step7:
Calculate the following in each bin.
- **Ab- IgG**
- **Random Ab - Random IgG**

Ab: antibody
Random Ab: Randomized antibody reads
Random IgG: Randomized IgG reads

Linux command "**cat**" & "**awk**"

File type: **.bed**

Step8:
Add a header to the BED files for the subsequent bootstrap test.

Linux command "**cat**" & "**sed**"

File type: **.bed**

Step10:
Copy the BED files into the directories with unified file name.

Step9:
Make directories for bootstrap test.

Linux command "**mkdir**"

Directories

Step11:
Copy a R script file of bootstrap test into the directories.

See R script in "c200_Rscript_boot.r"

File type: **.bed**

Linux command "**cat**" & "**head**"

File type: **.bed**

Step12:
Adjust bin number to 200,000 bins for bootstrap test.

R package: "**Boot**"

Output file: **Text file**

Step13:
Bootstrap test using R package "boot".

Confidence interval 0.99

Step14:
Extract putative signal regions using upper side of 99.9% confidence interval determined by bootstrap test.

Linux command "**cat**" & "**awk**"

Putative signal regions
File type: .bed

Step15:
Count region number of putative signals extracted in previous step.

Linux command "**find**" & "**wc -l**"

Number of putative signal regions/cell
File type: **.csv**

Step16:
Count read number in the putative signal regions.

Linux command "**cat**" & "**awk**"

File type: **.bed**

Step17:
Count number of putative signals per cell.

Linux command "**find**" & "**wc -l**"

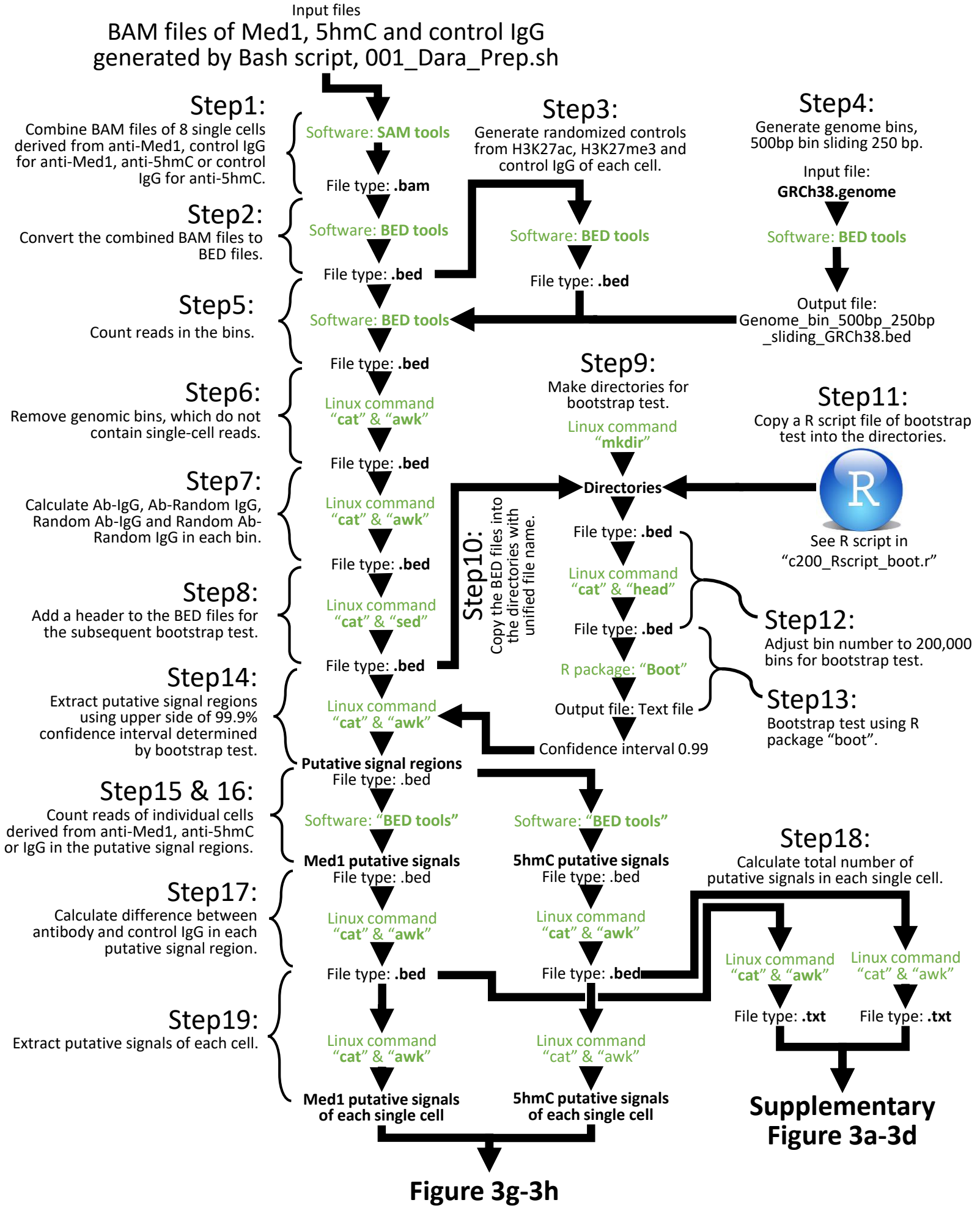
Number of putative signals/cell
File type: **.csv**

The generated data were used in
- Figure 3a-3h
- Figure
- Figure

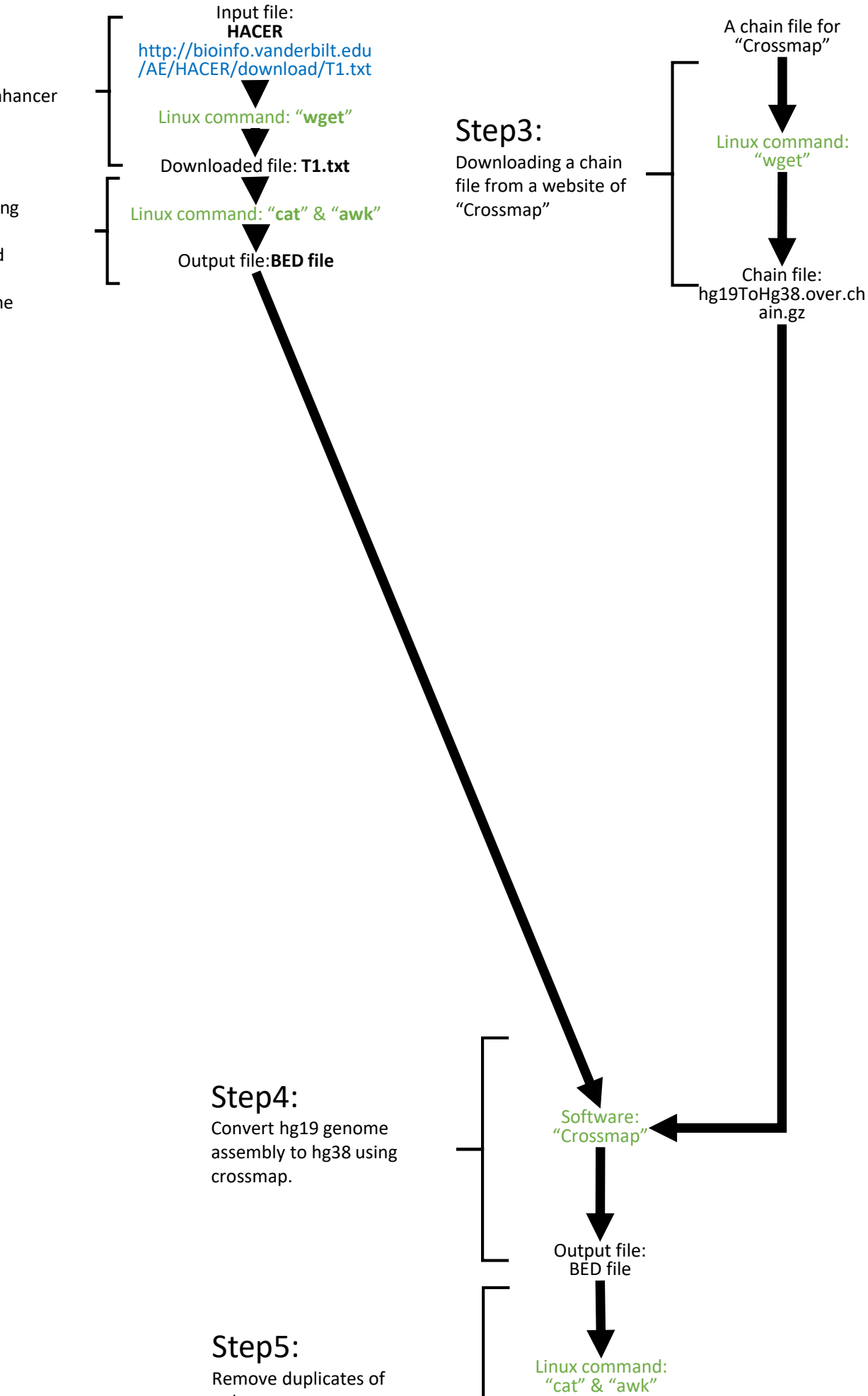
Figure 2c
Figure 2e

Figure 2d
Figure 2f

Bash script: **004_Supplementary_Fig4_bootstrap.sh**



Bash script: **005_Fig3a-3d.sh**



Bash script: **005_Fig3a-3d.sh**

Step1: Download enhancer datasets.
Input file: **HACER** http://bioinfo.vanderbilt.edu/AE/HACER/download/T1.txt
Linux command: "**wget**"
Downloaded file: **T1.txt**

Step2: Generate a BED file.
Linux command: "**cat**" & "**awk**"
Output file: **BED file**

Step3: Downloading a chain file.
A chain file for "Crossmap"
Linux command: "**wget**"
Chain file: hg19ToHg38.over.chain.gz

Step4: Convert genome assembly
Software: "**Crossmap**"
Output file: **BED file**

Step5: Remove duplicates.
Linux command: "**cat**" & "**awk**"
Output file: **BED file**

Step6: Download enhancer datasets.
Input file: **Human promoters** ftp://ccg.epfl.ch/epdnew/human/current/Hs_EPDnew.bed
Linux command: "wget"
Downloaded file: Hs_EPDnew.bed

Step7: Extract required info.
Linux command: "cat" & "awk"
Output file: BED file

Step8: Extract target gene symbols.
Linux command: "**cat**", "**awk**" & "**sed**"
Output file: **BED file**

Step9: Add target gene symbols.
Linux command: "cat" & "awk"

Step11: Add length info.
Output file: BED file
Linux command: "cat" & "awk"
Output file: BED file

Step12: Download bulk ChIP-seq datasets of H3K27ac and H3K27me3 (K562) from ENCODE.
ENCODE Bulk ChIP-seq data of K562 cells https://www.encodeproject.org/files/ENCFF301TVL/@@download/ENCFF301TVL.bam
Linux command: "wget"
Downloaded file: BAM files of H3K27ac bulk ChIP-seq

Step13: Convert BAM files of bulk ChIP to BED files.
Software: "BED tools"
Output file:BED files

Step14: Generate randomized controls of bulk ChIP (H3K27ac and H3K27me3).
Software: "BED tools"
Output file: BED files of randomized control

Step18: Sort order of rows.
Software: "**BED tools**"
Output file: **BED file**

Step19: Count number of the putative signals in an enhancer.
Software: "BED tools"
Option: "intersect"
Output file: BED file
Software: "BED tools"
Option: "intersect"
Output file: BED file

Software: "BED tools"
Option: "intersect"
Output file: BED file
Software: "BED tools"
Option: "intersect"
Output file: BED file

Software: "BED tools"
Option: "map"

Software: "BED tools"
Option: "map"

Step25: Bootstrap test to calculate confidence interval 0.999 in bulk ChIP-seq and single cells compare to randomized control

```
1  library(boot)
2  x=read.delim("e211_H3K27ac_RPKM_Bulk_SC_in_Enhancers_200000.bed", header=TRUE)
3  SC <- x$RandBulkH3K27ac
4  as.vector(SC)
5  sample <- function(SC, d) {
6  return(print(SC[d]))
7  }
8  Bout=boot(SC, sample, R=25000)
9  ci=boot.ci(Bout, conf=c(0.90, 0.95, 0.99, 0.999), type="basic")
10 sink("e311_Enhancer_Rand_bulk_H3K27ac.txt")
11 print (ci)
12 sink()
13 q()
14
```

e311_Enhancer_Rand_bulk_H3K27ac.r

Output: Confidence interval 0.99 of bulk ChIP-seq

```
1  library(boot)
2  x=read.delim("e211_H3K27ac_RPKM_B
3  SC <- x$RandscH3K27ac
4  as.vector(SC)
5  sample <- function(SC, d) {
6  return(print(SC[d]))
7  }
8  Bout=boot(SC, sample, R=25000)
9  ci=boot.ci(Bout, conf=c(0.90, 0.9
10 sink("e311_Enhancer_Rand_SC_H3K27
11 print (ci)
12 sink()
13 q()
14
```

e311_Enhancer_Rand_SC_H3K27ac.r

Output: Confidence interval 0.99 of single-cell data

Outpu Enhancer