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 Page**2**.

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_Fig2a-2b.sh

See Page**3**.

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_Fig2c-2f_Bootstrap.sh

See Page**4**.

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_Sup_Fig4_Bootstrap.sh

See Page**5**.

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: 005_Fig3a-3d.sh

See Page**6**.

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: 006_Fig3e-3h.sh

See Page**7**.

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: 007_Fig4a-4b.sh

See Page**8**.

What this script Do?

This script counts genes interacting K562-cell typical and atypical active enhancers, which were detected by RscEpi-seq and bulk ChIP-seq.

Shell script: 008_Fig4d.sh

See Page**9**.

What this script Do?

This script generates input data for pathway enrichment analysis shown in Figure 4d from data sets of RscEpi-seq and bulk ChIP-seq.

Shell script: 009_Fig5a.sh

See Page**10**.

What this script Do?

This script generates input data for t-stochactic neighbor embedding plots from RscEpi-seq and bulk ChIP-seq of various cell types.

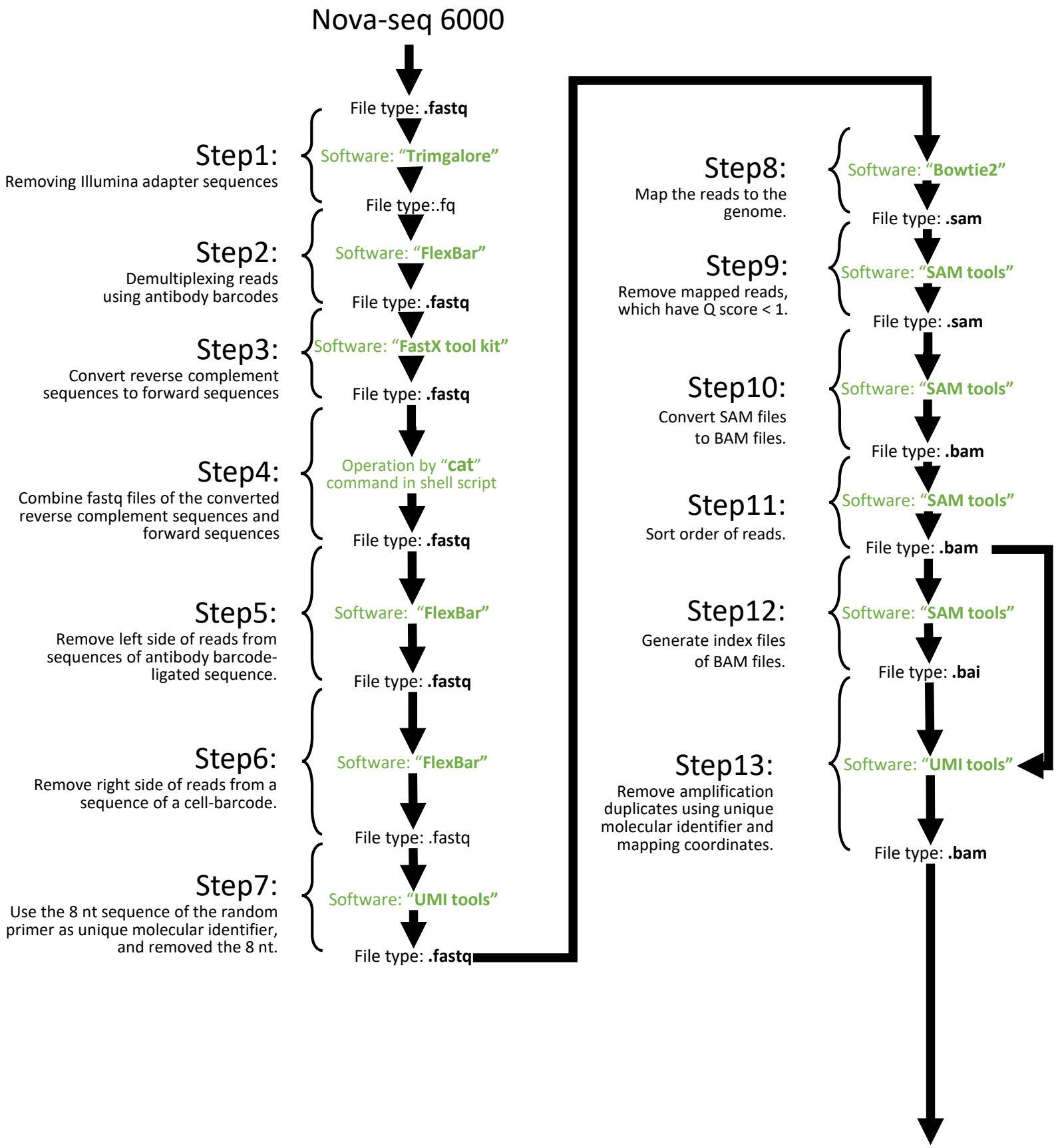
Shell script: 010_Fig5b-5e+Sup_Fig5a-5c.sh

See Page**11**.

What this script Do?

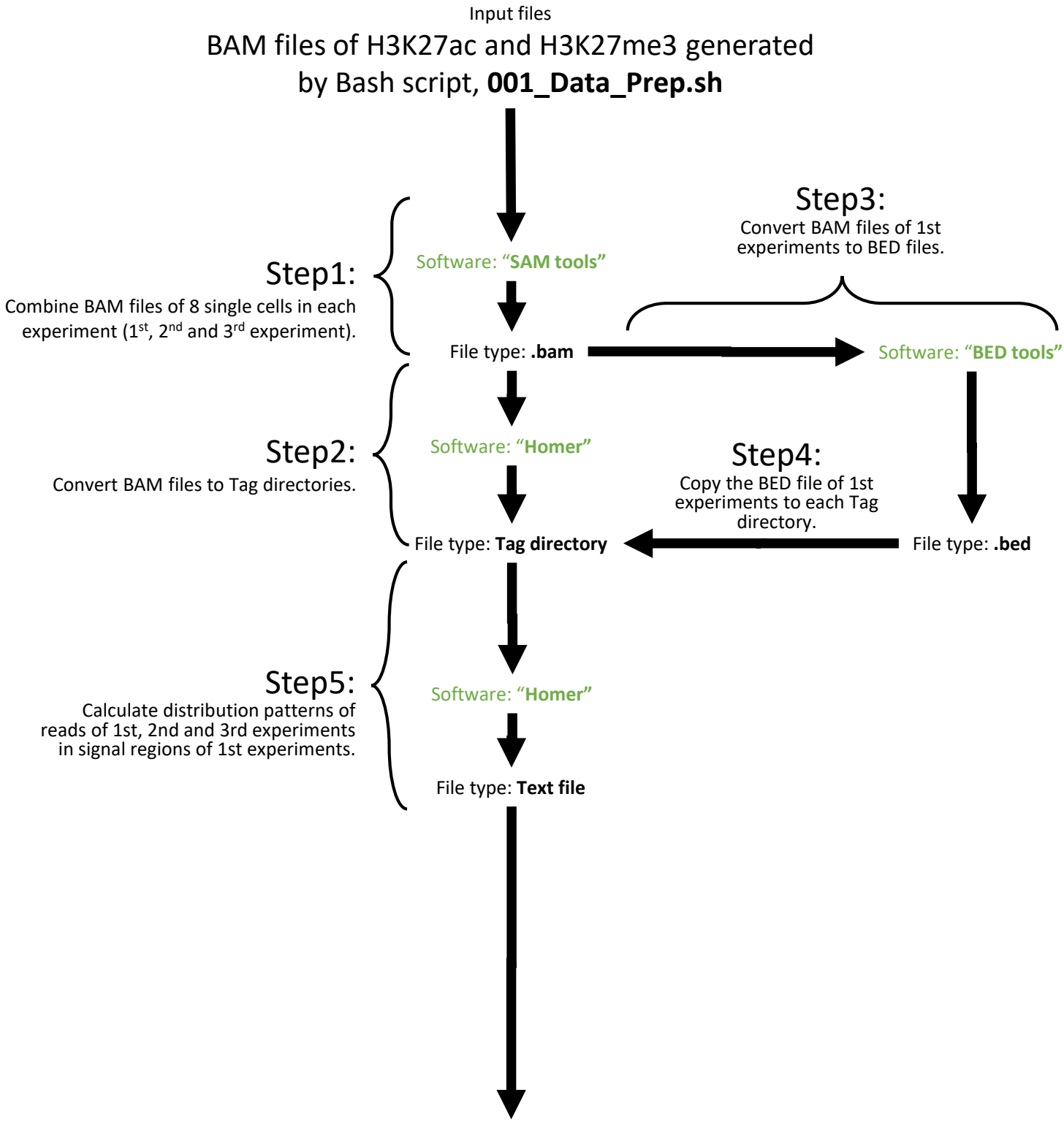
This script generates TDF and BED files used in Figure 5b-5e and Supplementary Figurre 5a-5c from data of RscEpi-seq.

Bash script: **001_Data_Prep.sh**



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

Bash script: **002_Fig2a-2b.sh**



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

Input files

BAM files of H3K27ac, H3K27me3 and control IgG generated by Bash script, **001_Data_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**

Step9:
Make directories for bootstrap test.

Linux command "**mkdir**"

Directories

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

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

See R script in "c200_Rscript_boot.r"

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

File type: **.bed**

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

File type: **.bed**

Step13:
Bootstrap test using R package "boot".

R package: "**Boot**"

Output file: **Text file**

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**"

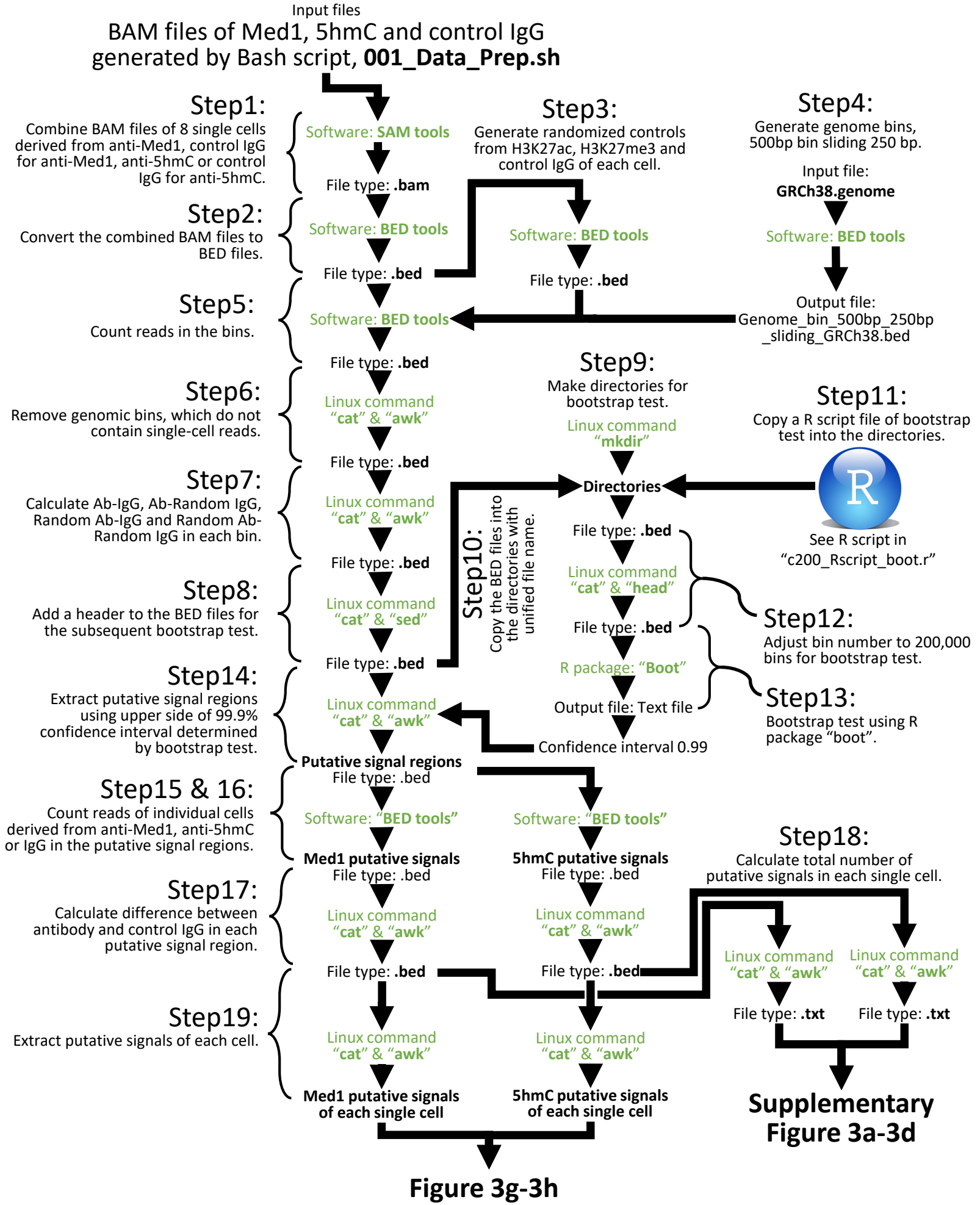
Number of putative signals/cell
File type: **.csv**

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

Figure 2c
Figure 2e

Figure 2d
Figure 2f

Bash script: **004_Sup_Fig4_bootstrap.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"

Step10-11: 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 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

Step15: Combine BED files of 8 single cells into a BED file.
Data from Bash script, **003_Fig2c-2f_Bootstrap.sh**
H3K27ac and control IgG: 8 BED files of 8 single cells
Linux command: "**cat**"
Output file: BED files

Step16: Remove a header in the BED files.
Linux command: "grep"
Output file: BED files

Step17: Sort order of rows in the BED files.
Software: "BED tools"
Output file: BED files

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

Step19 & 20: Count number of the putative signals in an enhancer and a promoter.
Software: "BED tools" Option: "intersect"
Output file: BED file
Software: "BED tools" Option: "intersect"
Output file: BED file
Software: "BED tools" Option: "map"
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"
Output file: BED file

Step21: Calculate RPKM.
Linux command: "cat" & "awk"
Output file: Enhancer BED file

Linux command: "cat" & "awk"
Output file: Promoter BED file

Step22: Remove blank rows.
Linux command: "cat" & "awk"
Output file: Enhancer BED file

Linux command: "cat" & "awk"
Output file: Promoter BED file

Step23: Add a header to the BED files.
Linux command: "cat" & "sed"
Output file: Enhancer BED file

Linux command: "cat" & "sed"
Output file: Promoter BED file

Step24: Extract first 200,000 rows
Linux command: "head" & "nl"
Output file: Enhancer BED file

Linux command: "grep –v K562"
Output file: K562-cell atypical active enhancers

Linux command: "grep K562"
Output file: K562-cell typical active enhancers

Linux command: "cat" & "awk"
Output file: K562-cell active promoters

Linux command: "head" & "nl"
Output file: Promoter BED file

Step25: Bootstrap test
See R script, "e311_Enhancer_Rand_bulk_H3K27ac.r
Output: Confidence interval 0.99 of bulk ChIP-seq

See R script, "e311_Enhancer_Rand_SC_H3K27ac.r
Output: Confidence interval 0.99 of single-cell data

A list of proximal genes of K562-cell typical active enhancers
Linux command: "cat" & "awk"
Select enhancers and promoters, which have p-value less than 0.05

Step25: Bootstrap test
e321_Promoter_Rand_bulk_H3K27ac.r
Output: Confidence interval 0.99 of bulk ChIP-seq

e321_Promoter_Rand_SC_H3K27ac.r
Output: Confidence interval 0.99 of single-cell data

Software: Microsoft EXCEL
Software: Microsoft EXCEL
Software: Microsoft EXCEL
Software: Microsoft EXCEL

Figure 3c **Figure 3d** **Figure 3b** **Figure 3a**

Bash script: **006_Fig3e-3h.sh**

Step1: Download enhancer datasets.
Input file: **HACER** http://bioinfo.vanderbilt.edu/AE/HACER/download/T1.txt
→ Linux command: "wget"
→ Downloaded file: D200_HANCER_hg19.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 hg19 to hg38.
→ Software: "Crossmap"
→ Output file: BED file

Step5: Remove duplicates of enhancers.
→ Linux command: "cat" & "awk"
→ Output file: BED file

Step8: Sort order of rows in enhancer BED files.
→ Software: "BED tools"
→ Output file: BED file

Data from **003_Fig2c-2f_Bootstrap.sh**

Data from **004_Sup_Fig4_Bootstrap.sh**

Step6: Combine BED files of 8 single cells into 1 BED file.
H3K27ac and control IgG: 8 BED files of 8 single cells
H3K27me3 and control IgG: 8 BED files of 8 single cells
Med1 and control IgG: 8 BED files of 8 single cells
5hmC and control IgG: 8 BED files of 8 single cells
→ Linux command: "cat"
→ Output file: BED files

Step7: Remove a header in the BED files.
→ Linux command: "grep"
→ Output file: BED files

Step8: Sort order of rows in BED files.
→ Software: "BED tools"
→ Output file: BED files

Step9: Count putative signals in each enhancer.
Software: "BED tools" Option: "map"
→ Output file: BED file

Step10: Count number of putative signals as an input.
→ Linux command: "cat" & "awk"
→ Number of putative signals as inputs Output file: Text

Step11: Calculate RPKM, reads per kilobase per million inputs.
→ Linux command: "cat" & "awk"
→ Output file: BED file

Step12: Remove enhancers, which were not detected by H3K27ac, H3K27me3, Med1 and 5hmC.
→ Linux command: "cat" & "awk"
→ Output file: BED file

Step13: To avoid error in calculation of ratio H3K27ac/H3K27me3, 0 value were converted to 0.01 in H3K27ac and H3K27me3.
→ Linux command: "sed"
→ Output file: BED file

Step14: Calculate log2(H3K27ac/H3K27me3).
→ Linux command: "cat" & "awk"
→ Output file: BED file

Step15: Separate enhancers based on ranges of Log2(H3K27ac/H3K27me3).
→ Linux command: "awk"
→ Output file: BED files

Step16, 18, 20, 22, 24, 26, 28 and 30: Calculate average of antibody signals and random controls (RPKM) in enhancers at each Log2(H3K27ac/H3K27me3) range.

Step16: H3K27ac (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step17: Summarizing results in step16.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step18: Randomized control of H3K27ac (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step19: Summarizing results in step18.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step20: H3K27me3 (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step21: Summarizing results in step20.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step22: Randomized control of H3K27me3 (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step23: Summarizing results in step22.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step24: Med1 (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step25: Summarizing results in step24.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step26: Randomized control of Med1 (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step27: Summarizing results in step26.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step28: 5hmC (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step29: Summarizing results in step28.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

Step30: Randomized control of 5hmC (RPKM)
→ Linux command: "cat" & "awk"
→ Output file: BED files

Step31: Summarizing results in step30.
→ Linux command: "find", "cat" & "paste"
→ Output file: BED file

→ **Figure 3e**
→ **Figure 3f**
→ **Figure 3g**
→ **Figure 3h**

Bash script: **007_Fig4a-4b.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 hg19 to hg38.
Software: "Crossmap"
Output file: BED file

Step5: Extract K562-cell specific active enhancers.
Linux command: "grep K562"
Output file: BED file

Step7: Add length info of an enhancer.
Linux command: "cat" & "awk"
Output file: BED file

Step11: Add signal counts in an enhancer.
Software: "bedtools" Option: "sort"
Output file: BED file
Software: "bedtools" Option: "intersect"
Output file: BED file
Software: "bedtools" Option: "intersect"
Output file: BED file
Software: "bedtools" Option: "map"
Output file: BED file
Software: "bedtools" Option: "map"
Output file: BED file

Step12: Calculate RPKM.
Linux command: "cat" & "awk"
Output file: BED file

Step13: Convert 0 value to 0.01
Linux command: "sed"
Output file: BED file

Step14: Calculate log2(H3K27ac/H3K27me3)
Linux command: "cat" & "awk"
Output file: BED file

Step15: Extract putative active enhancers
Linux command: "cat" & "awk"
Output file: BED file

Step16: Extract interacting genes of the putative active enhancers.
Linux command: "cat", "awk" and "sed"
Output file: BED file

Figure 4a

Step6: Extract K562-cell atypical enhancers.
Linux command: "grep –v K562"
Output file: BED file
Linux command: "cat" & "awk"
Output file: BED file
Software: "bedtools" Option: "sort"
Output file: BED file
Software: "bedtools" Option: "intersect"
Output file: BED file
Software: "bedtools" Option: "intersect"
Output file: BED file
Software: "bedtools" Option: "map"
Output file: BED file
Software: "bedtools" Option: "map"
Output file: BED file
Linux command: "cat" & "awk"
Output file: BED file
Linux command: "sed"
Output file: BED file
Linux command: "cat" & "awk"
Output file: BED file
Linux command: "cat" & "awk"
Output file: BED file
Linux command: "cat", "awk" and "sed"
Output file: BED file

Figure 4b

ENCODE Bulk ChIP-seq data (H3K27ac) of K562 cells https://www.encodeproject.org/files/ENCFF301TVL/@@download/ENCFF301TVL.bam
Linux command: "wget"
Downloaded file: BAM files of H3K27ac bulk ChIP-seq
Software: "BED tools"
Output file:BED files

ENCODE Bulk ChIP-seq data (H3K27me3) of K562 cells https://www.encodeproject.org/files/ENCFF330YFF/@@download/ENCFF330YFF.bam
Linux command: "wget"
Downloaded file: BAM files of H3K27me3 bulk ChIP-seq
Software: "BED tools"
Output file:BED files

Step8: Download bulk ChIP-seq datasets of H3K27ac and H3K27me3 and H3K27me3 (K562) from ENCODE.

Step9: Convert BAM files of bulk ChIP to BED files.

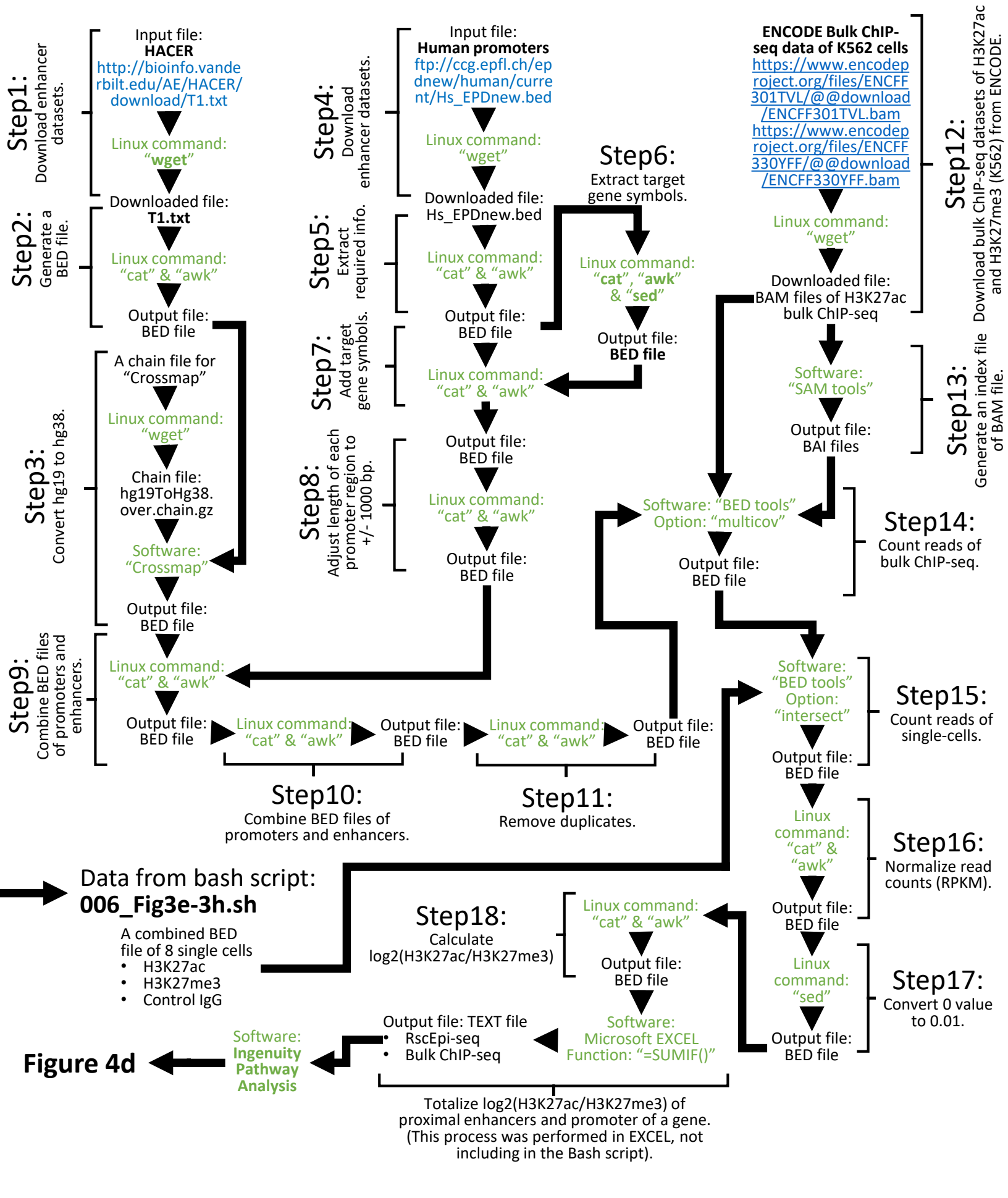
006_Fig3e-3h.sh

H3K27ac and control IgG: 8 BED files of 8 single cells
Linux command: "cat"
Output file: BED files
Linux command: "grep"
Output file: BED files
Software: "BED tools"
Output file: BED files

H3K27me3 and control IgG: 8 BED files of 8 single cells
Linux command: "cat"
Output file: BED files
Linux command: "grep"
Output file: BED files
Software: "BED tools"
Output file: BED files

Step10: Preparation of single-cell data.
Combine BED files of 8 single cells into 1 BED file.
Remove a header in the BED files.
Sort order of rows in BED files.

Bash script: **008_Fig4d.sh**



Bash script: **009_Fig5a.sh**

Step1: Download BAM files of bulk ChIP-seq (H3K27ac and H3K27me3).
- **ENCODE Bulk ChIP-seq data of 258 BAM files (see Table S4)**
- Linux command: "wget"
- Downloaded file: BAM files of H3K27ac bulk ChIP-seq

Step2: Convert BAM files to BED files.
- Software: "BED tools"
- Output file: BED files

Data generated by the shell script, **003_Fig2c-2f_Bootstrap.sh**

Data of H3K27ac and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Data of H3K27me3 and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Step14: Remove a header in a BED file.
- Linux command: "cat" & "awk"
- Output file: BED file

Data of H3K27ac and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Data of H3K27me3 and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Step17: Count number of reads in input BED files.
- Linux command: "cat" & "awk"
- Output file: Text files
- Linux command: "find", "wc –l", "echo" and "paste"
- Output file: CSV files

Step3: Download enhancer datasets.
- Input file: **HACER** http://bioinfo.vanderbilt.edu/AE/HACER/download/T1.txt
- Linux command: "**wget**"
- Downloaded file: **T1.txt**

Step4: Generate a BED file.
- Linux command: "cat" & "awk"
- Output file: BED file

Step5: Downloading a chain file.
- A chain file for "Crossmap"
- Linux command: "wget"
- Chain file: hg19ToHg38.over.chain.gz

Step6: Convert hg19 to hg38.
- Software: "Crossmap"
- Output file: BED file

Step7: Remove duplicates and add length info of an enhancer.
- Linux command: "cat" & "awk"
- Output file: BED file

Software" "BED tools" Option: "map"
- Output file: BED file

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

Step8: 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

Step9: Extract required info.
- Linux command: "cat" & "awk"
- Output file: BED file

Step10: Extract target gene symbols.
- Linux command: "**cat**", "**awk**" & "**sed**"
- Output file: **BED file**

Step11: Add target gene symbols.
- Linux command: "cat" & "awk"
- Output file: BED file

Step12: Adjust length of promoter region to +/- 1000 bp.
- Linux command: "cat" & "awk"
- Output file: BED file

Step13: Add length info of promoter regions.
- Linux command: "cat" & "awk"
- Output file: BED file

Software" "BED tools" Option: "map"
- Output file: BED file

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

Step15&16: Count single-cell putative signals and bulk ChIP signals in enhancers (step15) and promoters (step16).

Linux command: "cat" & "awk"
- Output file: BED files
- Linux command: "paste"
- Output file: BED files

Step18&19: Calculate RPKM (reads per kilobase per million input).

Linux command: "sed"
- Output file: BED files
- Linux command: "sed"
- Output file: BED files

Step20: Convert 0 value to 0.01 to avoid error in the subsequent process.

Step21: Calculate log2(H3K27ac/H3K27me3).

Linux command: "cat"
- Output file: BEDfiles

Step22: Combine promoters and enhancers BED files.

Linux command: "cat" & "awk"
- Output file: BEDfiles

Step23: Extract promoter ID, enhancer ID and Log2(H3K27ac/H3K27me3) values of cells.

Step24: Add cell codes for software, SeqGeq.
- Linux command: "cat" & "sed"
- Output file: BEDfiles

Step25: Pivot columns and rows.
- Linux command: "cat" & Python command
- Output file: BEDfiles

This part is not external software, SeqGeq. SeqGeq generated t-Stochastic Neighbor Embedding plot.
- Software: SeqGeq (FlowJo)

Figure 5a

Bash script: **010_Fig5b-5e+Sup_Fig5a-5c.sh**

Unique reads generated by the shell script, **001_Data_Prep.sh**

Data of Med1 and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Data of 5hmC and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Signal regions identified by bootstrap test in the shell script, **004_Sup_Fig4_Bootstrap.sh**

Data of Med1 and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Data of 5hmC and control IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Step1: Extract signals.

```
$ intersectBed –abam unique_reads.bam –b putative_signal_regions.bed –wa >output.bam
```

Step2: Convert BAM files to BED files
Software: "BED tools"
Option: "bamtobed"
Output file: BED files

Step3: Binarization
Software: "BED tools"
Option: "merge"
Output file: BED files

Step23: Convert BED files to TDF files.
Software: "IGV tools"
Option: "count"
Output file: BED files

Software: Integrative Genomics Viewer

Step4: Download enhancer datasets.
Input file: **HACER** http://bioinfo.vanderbilt.edu/AE/HACER/download/T1.txt
Linux command: "**wget**"
Downloaded file: **T1.txt**

Step5: Generate a BED file.
Linux command: "cat" & "awk"
Output file: BED file

Step6: Downloading a chain file.
A chain file for "Crossmap"
Linux command: "wget"
Chain file: hg19ToHg38.over.chain.gz

Step7: Convert hg19 to hg38.
Software: "Crossmap"
Output file: BED file

Step8: Add length info of enhancer.
Linux command: "cat" & "awk"
Output file: BED file

Signal regions identified by bootstrap test in the shell script, **003_Fig2c-2f_Bootstrap.sh**

Data of H3K27acΔcontrol IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Data of H3K27me3Δcontrol IgG:
- Single cell #1
- Single cell #2
- Single cell #3
- Single cell #4
- Single cell #5
- Single cell #6
- Single cell #7
- Single cell #8

Step9: Remove the header.
Linux command: "grep"
Output file: BED file

Step10: Sort order of rows.
Software: "BED tools"
Option: "sort"
Output file: BED file

Step11: Count signals.
Software: "BED tools"
Option: "map"
Output file: BED file

Step12: Count input number of reads.
Linux command: "cat" & "awk"
Output file: BED file

Step13: Calculate RPKM.
Linux command: "cat" & "awk"
Output file: BED file

Step14: Remove enhancers, which do not contain signals.
Linux command: "cat" & "awk"
Output file: BED file

Step15: Convert 0 value to 0.01 to avoid error in next step.
Linux command: "sed"
Output file: BED file

Step16: Calculate log2(H3K27ac/H3K27me3).
Linux command: "cat" & "awk"
Output file: BED file

Step17: Convert a calculation error to 0 value.
Linux command: "sed"
Output file: BED file

Step18: Extract enhancers, which have positive values.
Linux command: "sed"
Output file: BED file

Step19: Add header.
Linux command: "cat" & "sed"
Output file: BED file

Step20: Add line number to the first column.
Linux command: "cat" & "sed"
Output file: BED file

Step21: Calculate confidence intervals of the random controls.
R script, "boot"

Step22: Extract active enhancers (P<0.05).
Linux command: "cat" & "sed"
Output file: BED file

Step24: Extract K562-cell typical active enhancers detected in a single cell.
Linux command: "cat", "grep" and "awk"
Output file: BED files

Figure 5b-5e and Supplementary Figure 5a-5c