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 does this script Do??

Data preparation for all subsequent analysis in this manuscript.

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

Shell script: 003_Fig2c-2f_Bootstrap.sh

See Page4.

What does 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: 005_Fig3a-3d.sh

See Page6.

What does this script Do??

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

Shell script: 007_Fig4a-4b.sh

See Page8.

What does this script Do??

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

Shell script: 009_Fig5a.sh

See Page 10.

What does this script Do??

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

Shell script: 002_Fig2a-2b.sh

See Page3.

What does this script Do??

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

Shell script: 004_FigS4_Bootstrap.sh

See Page5.

What does 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 Figure S4.

Shell script: 006_Fig3e-3h.sh

See Page7.

What does 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_Fig4d.sh

See Page**9**.

What does this script Do??

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

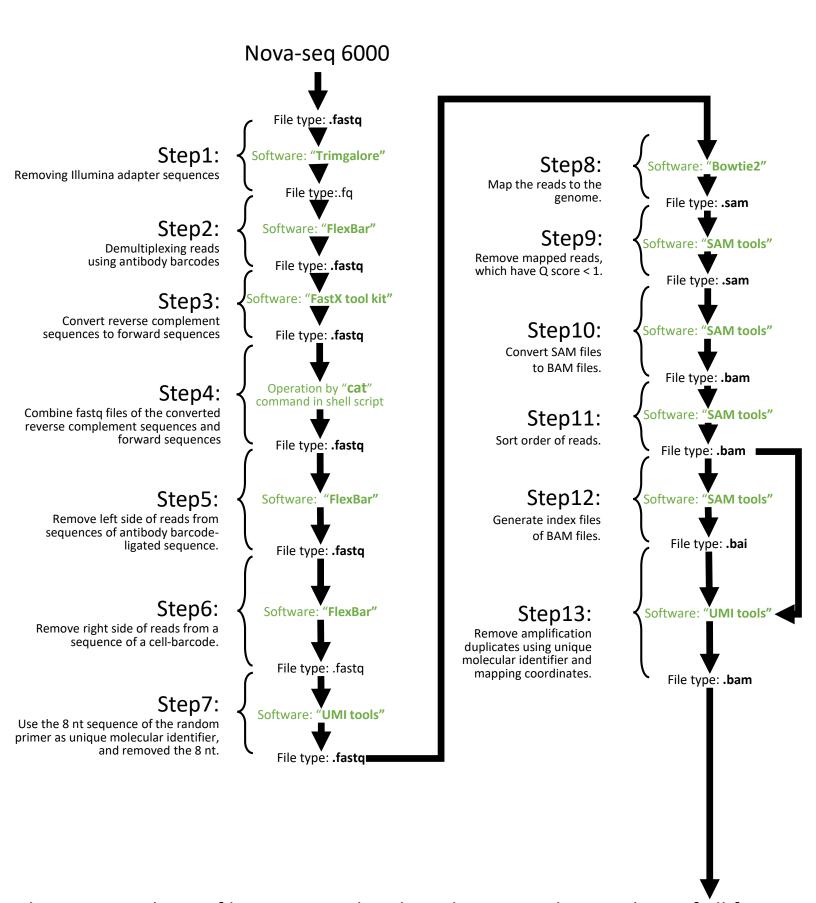
Shell script: 010 Fig5b-5e+FigS5a-S5c.sh

See Page11.

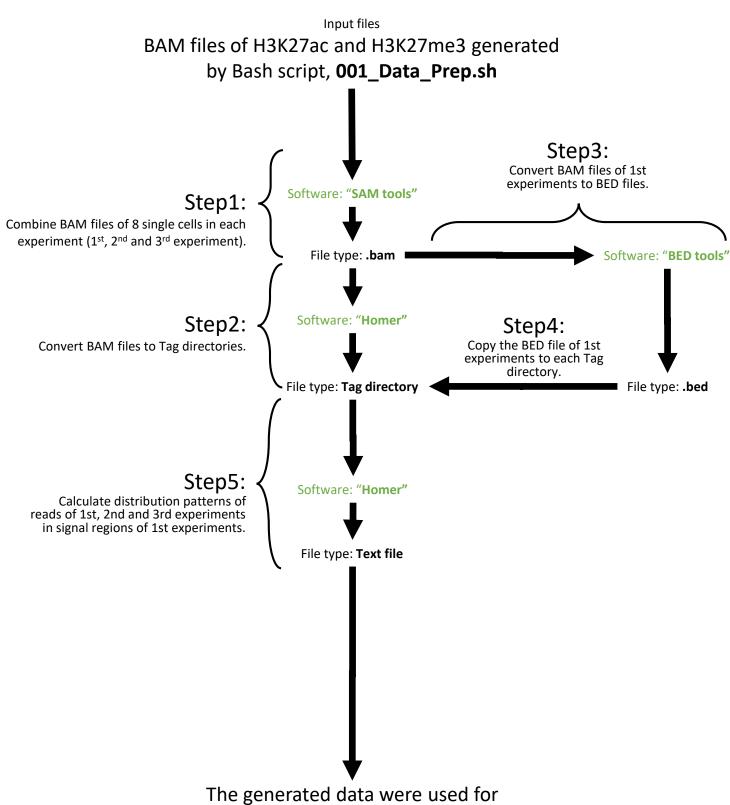
What does this script Do??

This script generates TDF and BED files used in Figures 5b-5e and Figures S5a-S5c from data of REpi-seq.

Bash script: 001_Data_Prep.sh



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



The generated data were used to

Figure 2a & 2b

Page 4 Bash script: 003_Fig2c-2f_Bootstrap.sh

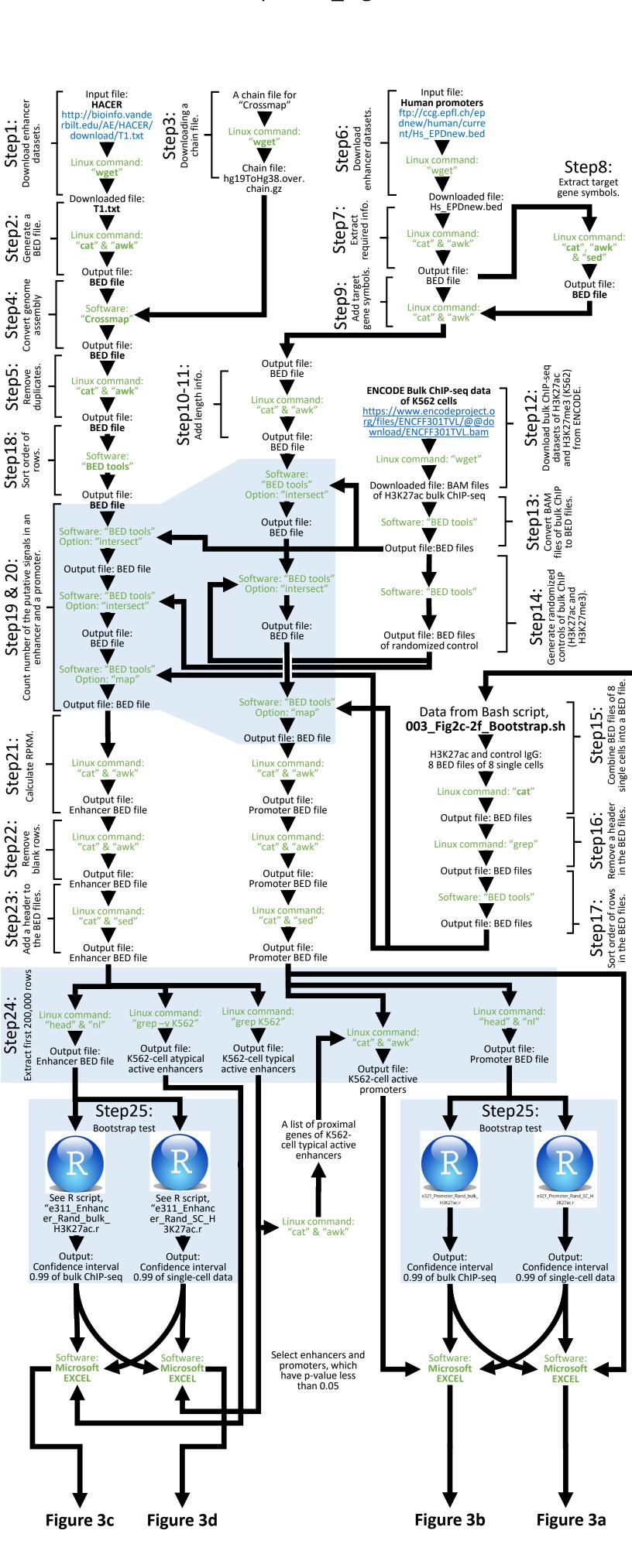
Input files

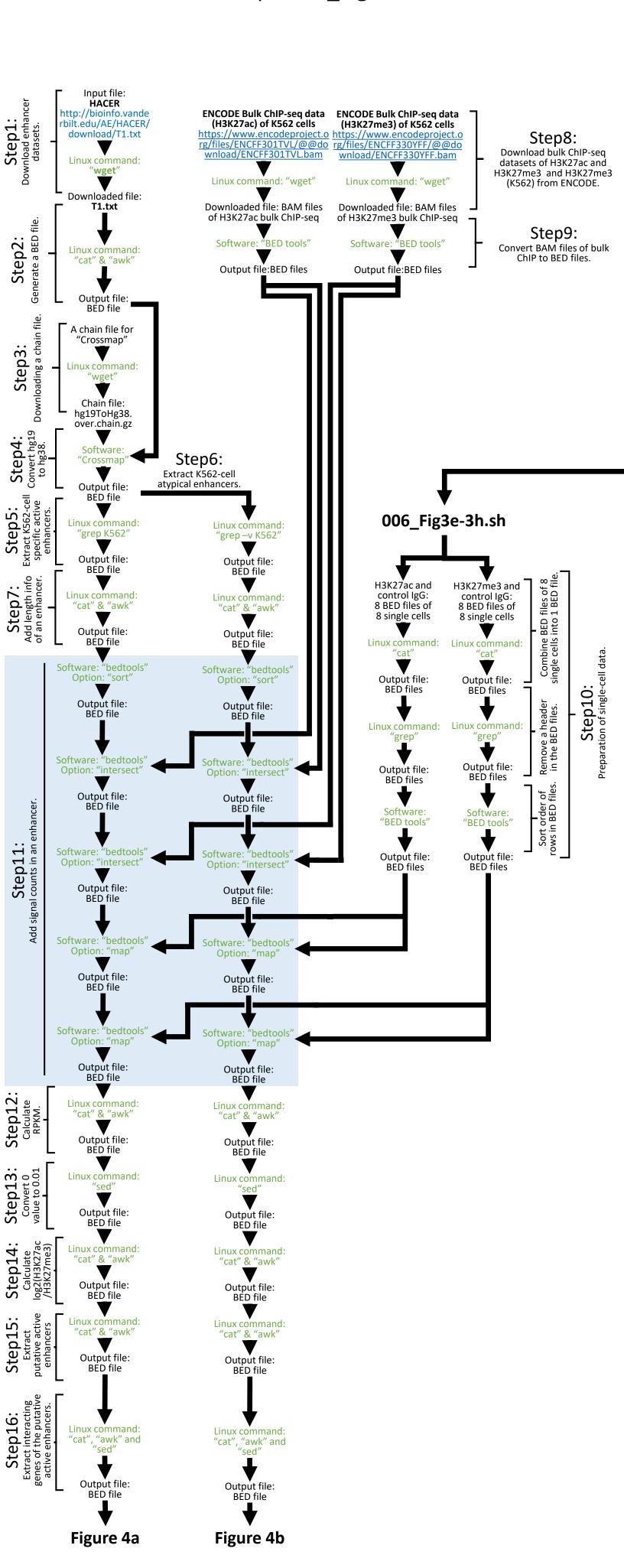
BAM files of H3K27ac, H3K27me3 and control IgG generated by Bash script, 001 Data Prep.sh Step3: Step4: Generate randomized controls Generate genome bins, Step1: from H3K27ac, H3K27me3 and 500bp bin sliding 250 bp. Software: SAM tools Combine BAM files of 1st, 2nd control IgG of each cell. and 3rd experiments derived Input file: from the same single cell File type: .bam GRCh38.genome Software: BED tools Step2: Software: **BED tools** Software: BED tools Convert the combined BAM files to BED files. File type: .bed File type: .bed Output file: Step5: Software: BED tools Genome_bin_500bp_250 bp_sliding_GRCh38.bed Count reads in the bins. File type: .bed Step9: Step6: Linux command Make directories Remove genomic bins, 'cat" & "awk" Step11: for bootstrap test. which do not contain single-cell reads. Copy a R script file of bootstrap Linux command test into the directories. "mkdir" File type: .bed Step7: **Directories** Calculate the following in each bin. Ab- IgG Copy the BED files into the directories with unified file name. Linux command Random Ab - Random IgG File type: **.bed** "cat" & "awk" Ab: antibody See R script in Random Ab: Randomized antibody reads "c200 Rscript boot.r" Linux command Random IgG: Randomized IgG reads "cat" & "head" File type: .bed Step12: Step8: File type: **.bed** Linux command Adjust bin number to 200,000 Add a header to the BED 'cat" & "sed" bins for bootstrap test. files for the subsequent R package: "Boot" bootstrap test. Step13: File type: .bed Output file: Text file Bootstrap test using R package "boot". Step14: Extract putative signal regions Linux command Confidence interval 0.99 'cat" & "awk" using upper side of 99.9% confidence interval determined by bootstrap test. Step16: **Putative signal regions** File type: .bed Count read number Linux command "cat" & "awk" in the putative Step15: Linux command "find" & "wc -l" signal regions. Count region number of File type: .bed putative signals extracted Step1/: Number of putative signal regions/cell in previous step. Linux command "find" & "wc -l" File type: .csv Count number of putative signals Number of putative signals/cell per cell. File type: .csv The generated data were used in Figure 2c Figure 2d Figure 3a-3h **Figure** Figure 2e Figure 2f **Figure**

Bash script: **004_FigS4_bootstrap.sh**

Input files BAM files of Med1, 5hmC and control IgG generated by Bash script, 001 Data Prep.sh Step4: Step1: Step3: Generate genome bins, Combine BAM files of 8 single cells Software: SAM tools Generate randomized controls 500bp bin sliding 250 bp. derived from anti-Med1, control IgG from H3K27ac, H3K27me3 and for anti-Med1, anti-5hmC or control control IgG of each cell. Input file: IgG for anti-5hmC. File type: .bam GRCh38.genome Step2: Software: BED tools Software: BED tools Software: BED tools Convert the combined BAM files to BED files. File type: **.bed** File type: .bed Step5: Output file: Software: **BED tools**◀ Genome bin 500bp 250bp Count reads in the bins. sliding GRCh38.bed Step9: File type: .bed Make directories for Step6: Step11: bootstrap test. Linux command Remove genomic bins, which do not "cat" & "awk" Copy a R script file of bootstrap Linux command contain single-cell reads. test into the directories. "mkdir" File type: .bed Step7: Directories ◄ Copy the BED files into the directories with unified file name. Calculate Ab-IgG, Ab-Random IgG, Linux command "cat" & "awk" Random Ab-IgG and Random Ab-File type: **.bed** Random IgG in each bin. See R script in "c200 Rscript boot.r" File type: .bed Linux command "cat" & "head" Step8: Linux command Add a header to the BED files for Step12: "cat" & "sed" File type: .bed the subsequent bootstrap test. Adjust bin number to 200,000 bins for bootstrap test. File type: .bed R package: "Boot" Step14: Step13: Extract putative signal regions Linux command Output file: Text file using upper side of 99.9% "cat" & "awk" Bootstrap test using R confidence interval determined package "boot". Confidence interval 0.99 by bootstrap test. **Putative signal regions** File type: .bed Step15 & 16: Count reads of individual cells Software: "BED tools" Software: "BED tools" derived from anti-Med1, anti-5hmC Step18: or IgG in the putative signal regions. Med1 putative signals 5hmC putative signals Calculate total number of File type: .bed File type: .bed putative signals in each single cell. Step17: Calculate difference between Linux command Linux command "cat" <u>& "</u>awk" antibody and control IgG in each 'cat" & "awk" putative signal region. Linux command Linux command File type: **.bed** File type: .bed■ 'cat" & "awk" "cat" & "awk" Step19: File type: .txt File type: .txt Linux command Linux command Extract putative signals of each cell. "cat" & "awk" 'cat" & "awk" 5hmC putative signals Med1 putative signals Figure S3a-S3d of each single cell of each single cell

Figure 3g-3h





Bash script: 008_Fig4d.sh

