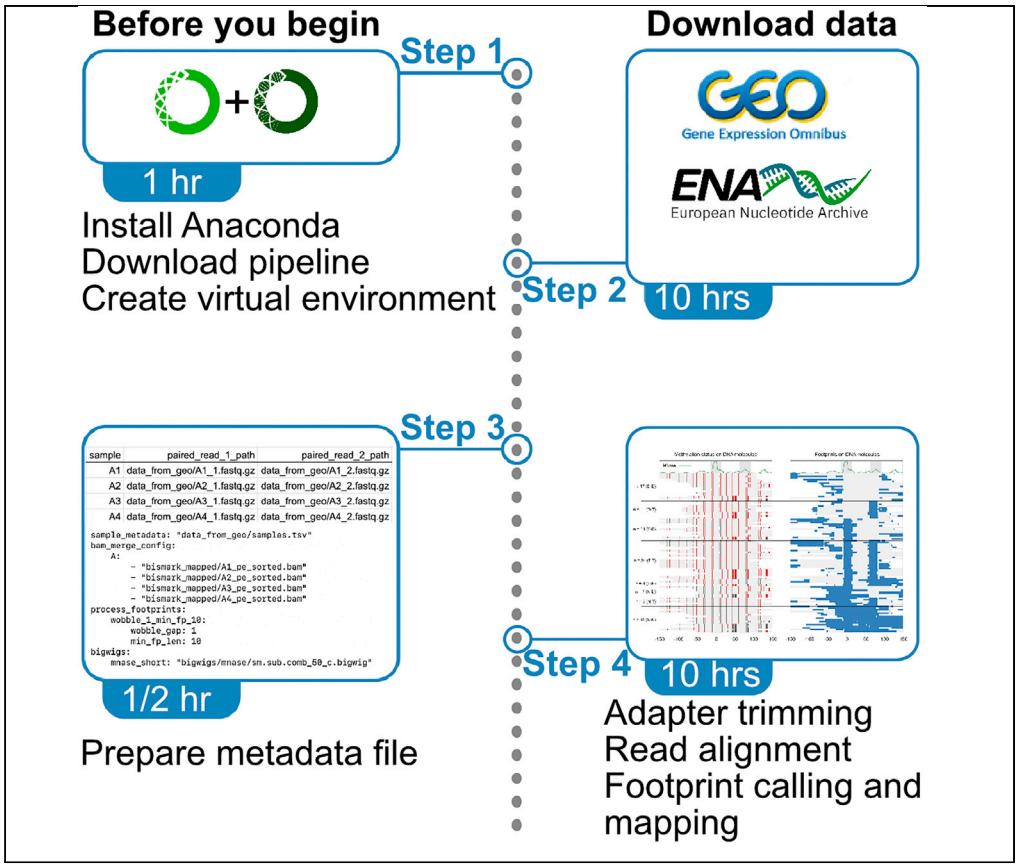
STAR Protocols

CellPress
OPEN ACCESS

Protocol

A computational pipeline to visualize DNA-protein binding states using dSMF data



Here, we present a pipeline to map states of protein-binding DNA *in vivo*. Our pipeline infers as well as quantifies cooperative binding. Using dual-enzyme single-molecule footprinting (dSMF) data, we show how our workflow identifies binding states at an enhancer in *Drosophila* S2 cells. Data from cells lacking endogenous DNA methylation are a prerequisite for this pipeline.

Satyanarayan Rao, Srinivas Ramachandran

satyanarayan.rao@cuanschutz.edu (S.R.)
srinivas.ramachandran@cuanschutz.edu (S.R.)

Highlights

Pipeline that uses dSMF data to quantify DNA-protein binding states

This pipeline can reliably map cobinding events on chomatinized DNA

Unlike MNase- and DNase-seq, dSMF maps the unbound state of genomic DNA

Rao & Ramachandran, STAR Protocols *3*, 101299
June 17, 2022
https://doi.org/10.1016/j.xpro.2022.101299

STAR Protocols

CellPress
OPEN ACCESS

Protocol

A computational pipeline to visualize DNA-protein binding states using dSMF data

Satyanarayan Rao[1,2,3,*] and Srinivas Ramachandran[1,2,4,*]

[1]Department of Biochemistry and Molecular Genetics, University of Colorado School of Medicine, Aurora, CO 80045, USA

[2]RNA Bioscience Initiative, University of Colorado School of Medicine, Aurora, CO 80045, USA

[3]Technical contact

[4]Lead contact

*Correspondence: satyanarayan.rao@cuanschutz.edu (S.R.), srinivas.ramachandran@cuanschutz.edu (S.R.)
https://doi.org/10.1016/j.xpro.2022.101299

SUMMARY

Here, we present a pipeline to map states of protein-binding DNA *in vivo*. Our pipeline infers as well as quantifies cooperative binding. Using dual-enzyme single-molecule footprinting (dSMF) data, we show how our workflow identifies binding states at an enhancer in *Drosophila* S2 cells. Data from cells lacking endogenous DNA methylation are a prerequisite for this pipeline.
For complete details on the use and execution of this protocol, please refer to Rao et al. (2021) and Krebs et al. (2017).

BEFORE YOU BEGIN

Overview

This pipeline facilitates end-to-end analysis of dSMF reads for analyzing protein-DNA binding. Apart from standard processing of raw sequencing reads, the pipeline implements a footprint calling algorithm and maps footprints to the genomic region of interest. The pipeline is built using Snakemake (Köster and Rahmann, 2012). Snakemake is a scalable bioinformatics workflow engine that enables execution of reproducible pipelines easily in a high-performance cluster.

Downloading the pipeline

Timing: ~5 min

1. If git command is available on user's machine, then please use this command "git clone https://github.com/satyanarayan-rao/star_protocol_enhancer_cooperativity.git", otherwise, please visit https://github.com/satyanarayan-rao/star_protocol_enhancer_cooperativity and click on the "code" button and select "Download Zip" to download the pipeline.

Installing required software and tools

Timing: ~1 h

We recommend using Anaconda software to build a virtual environment on user's machine to enable smooth operation of the pipeline without interference with other python packages already present in the system.

2. Download and install Anaconda (individual edition): https://www.anaconda.com/products/individual).

3. Once Anaconda is installed, create a virtual environment named "dsmf_analysis" or "dsmf_viz" using the command "conda create -n dsmf_analysis python=3.6".
4. After successful creation of the virtual environment, activate this environment by using the command "conda activate dsmf_analysis".
5. Run "install_required_packages.sh" to install required packages to run the pipeline.

Data collection

Timing: days; factors that affect timing include internet speed and available computing resources

This pipeline comprises the specific steps required for visualizing protein-binding on DNA molecules using dSMF data. To perform the analysis, the following data are needed: A reference genome in FASTA format; genomic regions of interest in BED format and reads from high-throughput sequencing of dSMF libraries:

6. Since the data we analyzed was generated in *Drosophila* S2 cells, we are using the Apr. 2006 assembly of *D. melanogaster* genome (dm3, BDGP Release 5). The reference genome FASTA file can be downloaded from: http://hgdownload.cse.ucsc.edu/goldenpath/dm3/bigZips/dm3.fa.gz. For convenience, we have provided a script ' download_reference_genome.sh' to download and place genome FASTA file in appropriate directory.
7. Any region of interest can be chosen, although regulatory regions of the genome are most meaningful. Here, we used cis-regulatory enhancers in Drosophila determined by STARR-seq. The data can be downloaded from: https://data.starklab.org/publications/yanez-cuna_genomeRes_2014/S2_peakSummits.txt.
8. We are using published paired-end bisulfite sequencing data (Krebs et al., 2017) as dSMF libraries here, but the user can easily perform similar analyses using their own data. Publicly available dSMF libraries used here can be downloaded using the URLs provided in the README.md file (see "Download reference genome and dSMF data" section in the README.md file). We have included demo data (subset of the bisulfite sequencing data from (Krebs et al., 2017)) to illustrate a quick run through the whole pipeline, starting from sequencing data to publication-quality figures.

⚠ CRITICAL: The bisulfite sequencing data should be paired-end.

Preparing metadata

Timing: ~10 min

9. Prepare metadata files: create a tab separated values (.tsv) file consisting of three columns with headers: "sample", "paired_read_1_path", "paired_read_2_path". For each sequenced sample, write a convenient name (please refrain from using special characters and spaces in the name to avoid the pipeline from failing), path for read1 and path for read2 in corresponding columns (e.g.,: "S2_tandem_R1_1", "data_from_geo/SRR3133326_1.fastq.gz", "data_from_geo/SRR3133326_2.fastq.gz". TSV file referring to the test dataset can be found here: "data_from_geo/samples.tsv".
10. For single binding sites, prepare a file containing regions of interest (ROI) using bed format (see format here: https://genome.ucsc.edu/FAQ/FAQformat.html#format1). Example can be seen in the GitHub repository "input_bed/example.bed".
11. For a pair of binding sites, prepare a file containing ROI using bedpe format (see .bedpe format description here: https://bedtools.readthedocs.io/en/latest/content/general-usage.html#bedpe-format). This format will allow us to provide the centers of the two binding sites. For an example, see "input_bed/example_cobinding.bedpe".

12. Prepare configuration file as a key-value pair (yaml) file. Please see "configs/config.yaml" for an example.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
dSMF bisulfite sequencing data (*Drosophila* S2 cells)	(Krebs et al., 2017)	GSE77369
STARR-seq summits from *Drosophila* S2 cells	(Yáñez-Cuna et al., 2014)	https://data.starklab.org/publications/yanez-cuna_genomeRes_2014/S2_peakSummits.txt
Software and algorithms		
Snakemake	(Köster and Rahmann, 2012)	https://snakemake.readthedocs.io/en/stable/
Gnuplot	N/A	http://www.gnuplot.info/
Trim Galore	N/A	https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
Bowtie2	(Langmead and Salzberg, 2012)	http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
Bismark	(Krueger and Andrews, 2011)	https://www.bioinformatics.babraham.ac.uk/projects/bismark/
Bamtools	(Barnett et al., 2011)	https://github.com/pezmaster31/bamtools
Samtools	(Danecek et al., 2021)	http://www.htslib.org/
Deeptools	(Ramírez et al., 2016)	https://deeptools.readthedocs.io/en/develop/
Anaconda	N/A	https://www.anaconda.com/
Bioconda	(Grüning et al., 2018)	https://bioconda.github.io/
Other		
Operating System	Linux/Unix	N/A

MATERIALS AND EQUIPMENT

- Hardware.
  - Local – Memory 8 GB required, 16 GB recommended; Processors: 1 required, 4 recommended,
  - Computational Cluster – Memory: >16 GB recommended, Processors: >8 recommended.
- Software and packages: A detailed list of software and packages can be found in the README.md file.

STEP-BY-STEP METHOD DETAILS

In a single command, you can visualize protein-DNA binding state of a region of your interest from raw sequencing data:

```
snakemake --snakefile cooperative_binding_analysis.smk

plots/single_binding/suppressed_merged_demo_S2_to_example_spanning_lf_15_r

f_15_extended_left_150_right_150_roi_peak_229.fp.pdf

plots/single_binding/suppressed_merged_demo_S2_to_example_spanning_lf_15_r

f_15_extended_left_150_right_150_roi_peak_229.methylation.pdf --configfile

configs/config.yaml
```

Below we will discuss the steps executed by the pipeline to generate the final figures.

CAUTION: This command will run relatively quickly because demo dataset has been used. If using full raw sequencing data, it will take around ~10–15 h for the first run to prepare close to final datasets required for visualization. After that, for any regions of your interest it will take less than 5 min for visualization.

Adapter trimming and alignment

Timing: ~5–10 h

In this step, we perform standard preprocessing of NGS data and alignment to the reference genome of interest. We use "Trim Galore" (see rule "trim_galore_pe" in snakemakes/trim_galore_pe.smk) for adapter trimming and "Bismark" (see rule "Bismark_align_pe" in snakemakes/Bismark_align_pe.smk) for bisulfite sequence alignment.

Suppressing cytosine methylation in contexts other than CpG or GpC dinucleotides

Timing: ~1 h

In this step, we refine the methylation calls by Bismark based on the methyl-transferases used in the dSMF experiment.

Note: We should only be considering methylation in CpG and GpC context based on the enzymes used in dSMF. To ensure this, we suppress methylation calls that are in other contexts (HCH; H: A,C,T) and (DGD; D: A,T,G).

Extracting adjacent or overlapping bisulfite reads

Timing: ~1 h

In this step, we extract properly aligned paired-end reads based on their SAM flags (<read, mate>: <99, 147> or <83, 163>) from the merged alignment file (see step 1.3), and generate longer reads by concatenating mate to read in case of pair <99, 147> and vice-versa in the case of <83, 163> (for further details about the meaning of SAM flags, see https://broadinstitute.github.io/picard/explain-flags.html).

Note: In overlapping regions, information on the leftmost read is kept. We then extract all longer reads that are overlapping or adjacent in nature. These reads form "DNA molecules" on which we will map binding states. Standard Illumina paired-end sequencing with read lengths 150 bp leads to typically long enough (median length of 269 bp) "DNA molecules" (Rao et al., 2021) to map transcription factor (TF) and nucleosomal binding.

Defining footprints

Timing: ~1 h

In this step, we define footprints on individual DNA molecules.

Note: It can be assumed that exogenous CpG and GpC methyltransferases reach saturation by methylating all accessible CpG and GpC dinucleotides. A footprint is called when one or more unmethylated cytosines are found between two methylated cytosines. We consider footprints of at least 10 bp long. If two footprints are separated by just one bp, we merge them to define a longer footprint. A DNA molecule with all unmethylated cytosines in CpG or GpC contexts, which results in no footprint is given special consideration because it could

arise due to complete occlusion by nucleosomes, thus a footprint size of the whole DNA molecule length is assigned in this case.

⚠ CRITICAL: With Drosophila S2 cells dSMF data, in open enhancers, one should expect about 40%–50% of naked-DNA, about 15%–20% of TF-bound and about 30%–40% of nucleosomal DNA (Rao et al., 2021). We imagine a similar distribution in other organisms based on the conservation of chromatin structure across eukaryotes.

Assigning states on DNA molecules at single binding sites

Timing: ~1 h

In this step, we assign binding states on individual DNA molecules mapped to ROI listed in a bed file.

Note: An example file is available at the GitHub repository "input_bed/example.bed". The ROI coordinates are midpoints of the broader regions one is interested in. For TFs, we consider 15 bp upstream and downstream of the midpoint to be sufficient. We then apply conditions to assign a footprint as naked-DNA, TF, or nucleosomal: No footprint in ROI+-15 or footprint of length less than 10 bp is assigned as naked-DNA, footprints of length between 10 and 50 bp in ROI+-15 are assigned as TFs, and others are assigned as nucleosomal. We discard < 50 bp sized footprints on the edge of DNA molecules because we only know the starting point of the footprint in these cases and hence the real length of these footprints is unknown.

Assigning binding states at a pair of binding sites

Timing: ~5 min

In this step, we assign co-binding states on individual DNA molecules that span a pair of TF binding sites. Prepare a bedpe formatted file (see .bedpe format description here: https://bedtools.readthedocs.io/en/latest/content/general-usage.html#bedpe-format) to provide the centers of the two binding sites. For an example, see "input_bed/example_cobinding.bedpe". This step can be executed by running the following command:

```
snakemake -np --snakefile cooperative_binding_analysis.smk
plots/cobinding_bedpe/suppressed_merged_S2_to_example_cobinding_lf_15_rf_
15_extended_left_300_right_300_roi_peak_110_4_and_peak_110_6.fp.pdf --
configfile configs/config.yaml
```

Note: Each TF binding site can have three states: naked or unbound, TF-bound, and nucleosome-bound. Thus, a pair of TF binding sites will have a total of nine states. States at individual sites are assigned as described in step 6. Additionally, a footprint spanning both sites that is <100 bp in length is labeled as co-bound.

⚠ CRITICAL: The bedpe file can only have the filename extension ".bedpe".

EXPECTED OUTCOMES

When the pipeline runs successfully, it automatically creates plots using Gnuplot. These plots feature footprints that are sorted based on the chromatin state they represent at the ROI (Figures 1 and 2).

Figure 1. An example of three binding states observed at Peak 229 (position 0 is at chr2L:480305)

(Left) Each line in the heatmap is a DNA molecule (the grey fill). A red dot represents a methylated cytosine and a dark gray dot represents an unmethylated cytosine. (Right) Blue lines are footprint calls on DNA molecules. These two heatmaps are directly taken from the output of the pipeline and custom labeled. Number of DNA molecules representing protein-DNA binding states is denoted on Y-axis. D: Naked DNA; T: TF-bound; N: Nucleosome-bound. For optimal visualization, in each state, DNA molecules are sorted by their length and coordinates relative to position 0.

LIMITATIONS

This pipeline can only be used for experiments conducted in cells predominantly lacking endogenous DNA methylation (for example *Drosophila* S2 cells). When regions of significant endogenous methylation (>10% of a cytosine is methylated endogenously) are known, they should not be included in the analysis (Raddatz et al., 2013; Takayama et al., 2014). The pipeline requires paired-end datasets. This pipeline could be modified and extended for approaches that use nanopore and pacbio sequencing technologies that directly sequence methylated nucleotides and for methods that involve adenine methylation.

TROUBLESHOOTING

Problem 1

Where else can I locate all binding state assignments for my region of interest after "assigning states on DNA molecules at single binding sites", and "assigning binding states at a pair of binding sites" steps?

Potential solution

Files in 'binding_states_fragments_extended/' and 'cobinding_states_bedpe' has binding state information for single sites ('input_bed/example.bed') and pair of sites ('input_bed/example_cobinding.bedpe') respectively. Please locate the corresponding file and search for your ROI. The first column has binding state information (followed by #). 0: Naked-DNA; 1: TF 2: Nucleosome; 3: To be discarded.

Problem 2

I have fewer rows (DNA molecules) in methylation/footprint heatmap than the input matrix after the steps: "assigning states on DNA molecules at single binding sites" and "assigning binding states at a pair of binding sites".

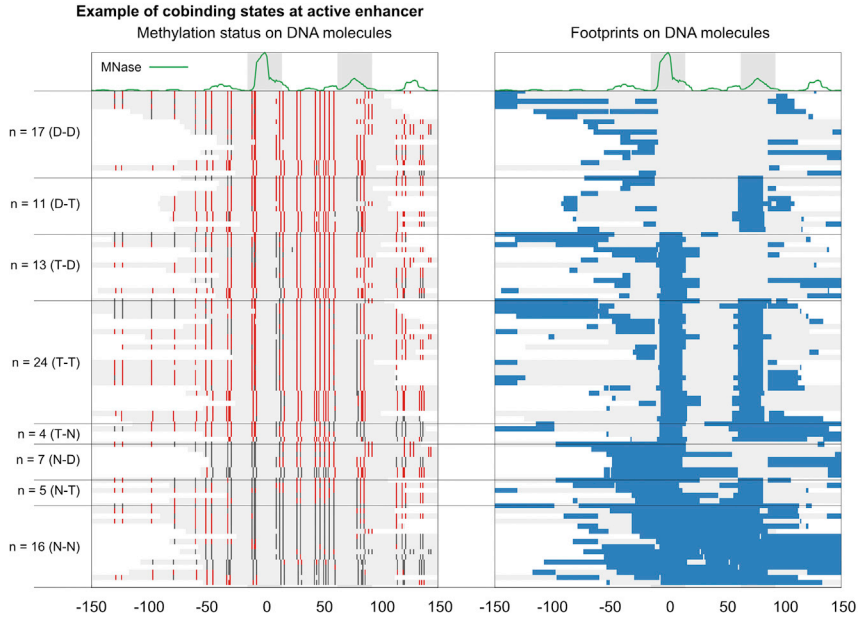


Figure 2. An example of cobinding states observed at enhancer Peak 110 (position 0 is at chr2L: 19155173)
(Left) Each line in the heatmap is a DNA molecule (the grey fill). A red dot represents a methylated cytosine and a dark gray dot represents an unmethylated cytosine. (Right) Blue lines are footprint calls on DNA molecules. These two heatmaps are directly taken from the output of the pipeline and custom labeled. DNA molecules spanning the two MNase peaks (shown in grey box at the top) are included in the plot. The MNase peaks are separated by 78 bp. Eight of nine possible states are found at this locus. The number of DNA molecules mapped to each protein-DNA binding state is denoted on Y-axis. D: Naked DNA; T: TF-bound; N: Nucleosome-bound. T-T, for example represents both sites bound by TFs simultaneously. For optimal visualization, in each state, DNA molecules are sorted by their length and coordinates relative to position 0.

Potential solution

We are only plotting valid protein-DNA and naked DNA states. Mapped DNA molecules with no valid footprints (footprints less than 50 bp on the edge of DNA molecule) are discarded in heatmap plotting. For instance, 25 DNA molecules are discarded in Figure 1.

Problem 3

The pipeline failed for my ROI named as "roi_peak_x" in my single site bed file when I ran the following snakemake command (step: "assigning binding states at a pair of binding sites"):

```
snakemake --snakefile cooperative_binding_analysis.smk
plots/single_binding/suppressed_merged_demo_S2_to_example_spanning_lf_15_
rf_15_extended_left_150_right_150_roi_roi_peak_229.fp.pdf --configfile
configs/config.yaml
```

How do I resolve this?

Potential solution

It is noteworthy that Snakemake builds workflow based on parsing the output filename. The example peak_id written here as "roi_peak_x" overlaps with one of the string literals ("roi") used by Snakemake to retrieve wildcards. Thus, in this case rextend wildcard assigned a value "150_roi" in spite of "150". Please refrain from using names and other parameters that overlaps with Snakemake

wildcards. Please see the description of parameters here: https://github.com/satyanarayan-rao/star_protocol_enhancer_cooperativity#interpreting-file-names.

Problem 4

I want to customize the pipeline, but I want to know which components of the pipeline will be affected by my changes?

Potential solution

Snakemake facilitates visualization of pipeline up to your desired output file. To do this, you should check if 'dot' command is available. If not, then please install graphviz (https://graphviz.org/). On Linux machines, you can install this tool using:

```
sudo apt-get install graphviz
```

On macOS, you can install using:

```
brew install graphviz
```

Once graphviz is installed, you should be able to run the following:

```
snakemake -np --dag --snakefile cooperative_binding_analysis.smk
plots/single_binding/suppressed_merged_demo_S2_to_example_spanning_1f_15_
rf_15_extended_left_150_right_150_roi_peak_229.fp.pdf --configfile
configs/config.yaml | dot -Tpng > raw_sequencing_to_single_binding.png
```

This will visualize the pipeline after you make edits.

Problem 5

I ran jobs on HPC cluster, and one or more jobs aborted due to error, where should I look to trace-back errors?

Potential solution

Please look at files in the directory 'logs/cluster'. Look for the ID of aborted jobs in error files using the command 'grep -w <job_id> *.err' and then look for error(s) in these error files.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Srinivas Ramachandran (srinivas.ramachandran@cuanschutz.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

Data and code are available at the GitHub repository: https://github.com/satyanarayan-rao/star_protocol_enhancer_cooperativity and at Zenodo: https://doi.org/10.5281/zenodo.5914775.

ACKNOWLEDGMENTS

This work was supported by the RNA Bioscience Initiative, University of Colorado School of Medicine, and NIH grant R35GM133434 (S. Ramachandran).

AUTHOR CONTRIBUTIONS

S. Rao designed the project, generated all the code, performed the analysis, and wrote the manuscript. S. Ramachandran designed and supervised the project and edited the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Barnett, D.W., Garrison, E.K., Quinlan, A.R., Strömberg, M.P., and Marth, G.T. (2011). BamTools: a C++ API and toolkit for analyzing and managing BAM files. Bioinformatics *27*, 1691–1692.

Danecek, P., Bonfield, J.K., Liddle, J., Marshall, J., Ohan, V., Pollard, M.O., Whitwham, A., Keane, T., McCarthy, S.A., Davies, R.M., et al. (2021). Twelve years of SAMtools and BCFtools. GigaScience *10*, giab008.

Grüning, B., Dale, R., Sjödin, A., Chapman, B.A., Rowe, J., Tomkins-Tinch, C.H., Valieris, R., and Köster, J. (2018). Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat. Methods *15*, 475–476.

Köster, J., and Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. Bioinformatics *28*, 2520–2522.

Krebs, A.R., Imanci, D., Hoerner, L., Gaidatzis, D., Burger, L., and Schübeler, D. (2017). Genome-wide single-molecule footprinting reveals high RNA polymerase II turnover at paused promoters. Mol. Cell *67*, 411–422.e4.

Krueger, F., and Andrews, S.R. (2011). Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. Bioinformatics *27*, 1571–1572.

Langmead, B., and Salzberg, S.L. (2012). Fast gapped-read alignment with Bowtie 2. Nat. Methods *9*, 357–359.

Raddatz, G., Guzzardo, P.M., Olova, N., Fantappié, M.R., Rampp, M., Schaefer, M., Reik, W., Hannon, G.J., and Lyko, F. (2013). Dnmt2-dependent methylomes lack defined DNA methylation patterns. PNAS *110*, 8627–8631.

Ramírez, F., Ryan, D.P., Grüning, B., Bhardwaj, V., Kilpert, F., Richter, A.S., Heyne, S., Dündar, F., and Manke, T. (2016). deepTools2: a next generation web server for deep-sequencing data analysis. Nucleic Acids Res. *44*, W160–W165.

Rao, S., Ahmad, K., and Ramachandran, S. (2021). Cooperative binding between distant transcription factors is a hallmark of active enhancers. Mol. Cell *81*, 1651–1665.e4.

Takayama, S., Dhahbi, J., Roberts, A., Mao, G., Heo, S.-J., Pachter, L., Martin, D.I.K., and Boffelli, D. (2014). Genome methylation in D. melanogaster is found at specific short motifs and is independent of DNMT2 activity. Genome Res. *24*, 821–830.

Yáñez-Cuna, J.O., Arnold, C.D., Stampfel, G., Boryń, Ł.M., Gerlach, D., Rath, M., and Stark, A. (2014). Dissection of thousands of cell type-specific enhancers identifies dinucleotide repeat motifs as general enhancer features. Genome Res. *24*, 1147–1156.