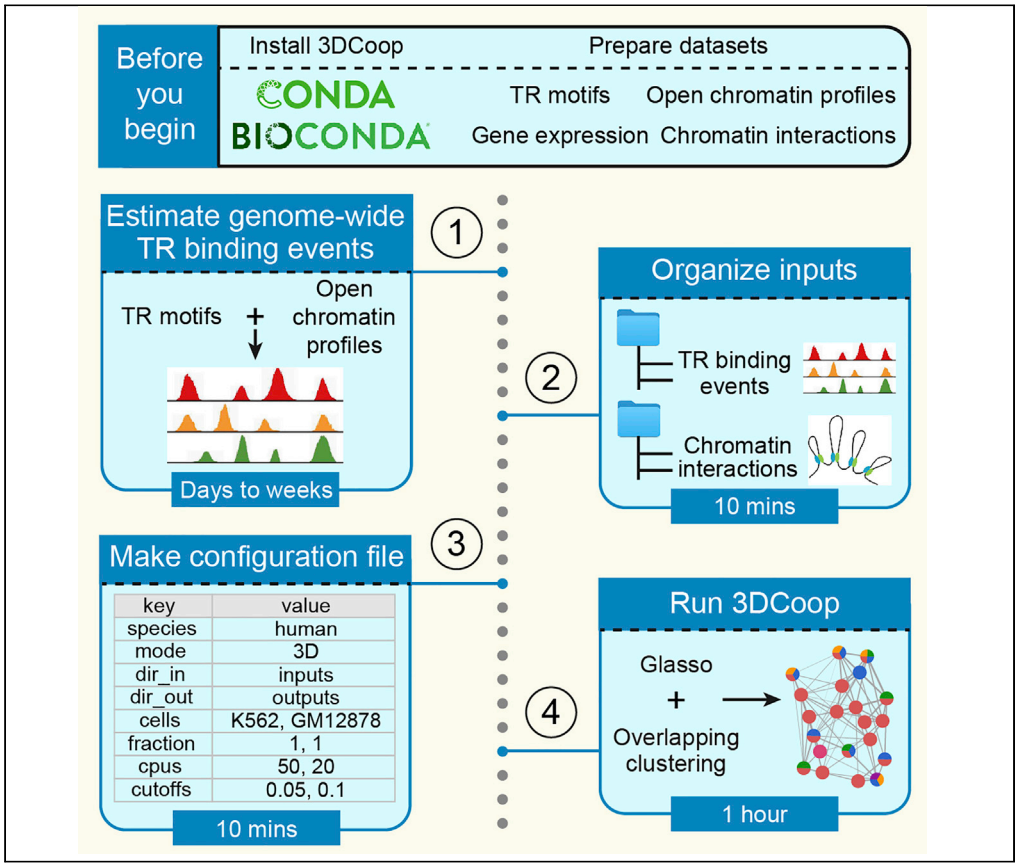
STAR Protocols

CellPress
OPEN ACCESS

Protocol

3DCoop: An approach for computational inference of cell-type-specific transcriptional regulators cooperation in 3D chromatin



Precise identification of context-specific transcriptional regulators (TRs) cooperation facilitates the understanding of complex gene regulation. However, previous methods are highly reliant on the availability of ChIPped TRs. Here, we provide a protocol for running 3DCoop, a pipeline for computational inference of cell type-specific TR cooperation in 3D chromatin by integrating TR motifs, open chromatin profiles, gene expression, and chromatin loops. 3DCoop provides a feasible solution to study the potential interplay among TRs across multiple human or mouse tissue/cell types.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Xianfu Yi, Menghan Luo, Xiangling Feng, Yao Zhou, Jianhua Wang, Mulin Jun Li

yixfbio@gmail.com (X.Y.)
mulinli@connect.hku.hk (M.J.L.)

Highlights

Inference of transcriptional regulator (TR) cooperation in 3D chromatin

Integration of TR motifs, open chromatin, gene expression, and chromatin loops

Identification of context-specific TR cooperation not relying on ChIPped factors

3DCoop facilitates TR cooperation study across multiple human/mouse cell types

Yi et al., STAR Protocols 3, 101382
June 17, 2022
https://doi.org/10.1016/j.xpro.2022.101382

STAR Protocols

CellPress
OPEN ACCESS

Protocol

3DCoop: An approach for computational inference of cell-type-specific transcriptional regulators cooperation in 3D chromatin

Xianfu Yi,[1,2,4,*] Menghan Luo,[1] Xiangling Feng,[1] Yao Zhou,[1] Jianhua Wang,[1] and Mulin Jun Li[1,3,5,*]

[1]Department of Bioinformatics, School of Basic Medical Sciences, Tianjin Medical University, Tianjin 300070, China

[2]School of Biomedical Engineering and Technology, Tianjin Medical University, Tianjin 300070, China

[3]Department of Epidemiology and Biostatistics, Tianjin Key Laboratory of Molecular Cancer Epidemiology, The Province and Ministry Co-sponsored Collaborative Innovation Center for Medical Epigenetics, National Clinical Research Center for Cancer, Tianjin Medical University Cancer Institute and Hospital, Tianjin Medical University, Tianjin 300070, China

[4]Technical contact

[5]Lead contact

*Correspondence: yixfbio@gmail.com (X.Y.), mulinli@connect.hku.hk (M.J.L.)
https://doi.org/10.1016/j.xpro.2022.101382

SUMMARY

Precise identification of context-specific transcriptional regulators (TRs) cooperation facilitates the understanding of complex gene regulation. However, previous methods are highly reliant on the availability of ChIPped TRs. Here, we provide a protocol for running 3DCoop, a pipeline for computational inference of cell type-specific TR cooperation in 3D chromatin by integrating TR motifs, open chromatin profiles, gene expression, and chromatin loops. 3DCoop provides a feasible solution to study the potential interplay among TRs across multiple human or mouse tissue/cell types.

For complete details on the use and execution of this protocol, please refer to Yi et al. (2021).

BEFORE YOU BEGIN

The identification of spatiotemporal patterns of gene regulation in different biological conditions has long been a critical problem in functional genomics. Transcription regulators (TRs) modulate the tissue/cell type-specific transcription events in the nucleus. Precise identification of context-specific TR cooperation can facilitate the understanding of complex gene regulation. Many approaches have been developed to infer TR cooperation in a specific cellular context. However, they are highly reliant on the availability of ChIPped factors, which limits their broader application to various biological conditions.

By integrating TR motifs, open chromatin profiles, gene expression, and chromatin loops, we developed a pipeline for computational inference of cell type-specific TR cooperation in 3D chromatin, called 3DCoop. Without using the endogenous binding sites of interested TRs, 3DCoop provides a feasible solution to estimate potential interplay among TRs across a broad context. By applying 3DCoop to multiple human or mouse tissue/cell types, the inferred TR cooperation information can facilitate the understanding of the complex gene regulation during cell differentiation and disease development. It can also promote the interpretation of the disease-causal variants identified by genome-wide association studies and the recapitulation of cell states during neural development (Yi et al., 2021).

The following section includes the installation procedure, as well as the preparation of required datasets to be used in 3DCoop (Yi et al., 2021). The protocol is illustrated based on the **human K562 cell line**. However, the 3DCoop pipeline can also be used for any human/mouse tissue or cell type when the required datasets are available.

Install 3DCoop pipeline

Timing: <30 min

1. 3DCoop pipeline is built via conda and bioconda (Gruning et al., 2018). Please check the bioconda installation page for conda installation and bioconda channel set up.

2. Install the 3DCoop pipeline and its dependencies in the terminal (on Unix-like systems or Windows Subsystem for Linux (WSL) on Windows 10 or above) with the bash command. The installation is automated via conda.
   a. Clone the repository from GitHub using the git command:

```
$git clone https://github.com/mulinlab/3DCoop
```

   b. Install the dependent packages. A bash script has been provided for convenience.

```
$cd 3DCoop

$bash conda.sh
```

   If the installation errors occur during the environment setting, individual packages can be installed one by one until the issue reproduces. The step-by-step commands are:

```
$conda create -n 3DCoop # Create the environment

$conda activate 3DCoop # Activate the environment

$conda install -c bioconda bedtools # Install bedtools

$conda install -c bioconda samtools # Install samtools

$conda install -c bioconda perl-list-moreutils # Install Perl package,

List::MoreUtils

$conda install -c bioconda perl-parallel-forkmanager # Install Perl

package, Parallel::ForkManager

$conda install -c r r-tidyverse # Install R package, tidyverse

$conda install -c r r-reshape # Install R package, reshape

$conda install -c conda-forge r-huge # Install R package, huge

$conda install -c conda-forge r-igraph # Install R package, igraph

$conda install -c conda-forge r-desctools # Install R package, DescTools

$conda install -c conda-forge r-ggnetwork # Install R package, ggnetwork

$conda install -c conda-forge r-intergraph # Install R package,

Intergraph

$conda deactivate 3DCoop # Deactivate the environment
```

c. Activate the conda environment "3DCoop" before running any 3DCoop scripts as follows: Troubleshooting 1.

```
$conda activate 3DCoop
```

Prepare datasets

Timing: hours to days; depending on the number of tissues or cell types

Note: More than 100 GB of disk space is needed to store all the required datasets. Please note the free hard disk space.

⚠ CRITICAL: The genome builds or assembly releases must be consistent for all required datasets. GRCh37/hg19 is used in this protocol. Besides, the chromosome identifiers or FASTA headers in different files (BAM, BEDPE, etc.) should be matched, such as chr1 vs. chr1, otherwise chr1 vs. 1.

3. Collect the motifs information of human TRs. The TR motifs can be collected from the existing databases, such as CIS-BP (Weirauch et al., 2014), JASPAR (Fornes et al., 2020), and HOCOMOCO (Kulakovskiy et al., 2018). 3DCoop systematically incorporates 3,105 motifs of 1,480 human TRs by collecting and uniformly processing 16 existing transcription factor (TF) motif resources, including the aforementioned databases and others. The final non-redundant motifs are stored in JASPAR PPM (Position Probability Matrix) format, and can be downloaded from the 3DCoop pipeline repository:

```
$wget -c

https://github.com/mulinlab/3DCoop/blob/master/resource/human_T

R_motif.txt
```

Note: The motifs collected from different resources usually contain redundancy. To reduce such redundancy, we select the best up to three distinct motifs per TR by measuring their similarity using MACRO-APE (Vorontsov et al., 2013). It results in 2.1 motifs per TR on average.

Note: The JASPAR PPM format of each motif includes a header line that begins with the ">" symbol, which is followed by a unique identifier indicating the motif ID and TR name. The lines for each base start with a label for the nucleotides (A, C, G, or T), and then the columns indicating the probability for each position enclosed in square brackets. Here is one example:

```
>HM05564 ABCF2

A  [ 0.000000 0.833333 1.000000 0.833333 0.166667 0.000000 ]

C  [ 0.166667 0.166667 0.000000 0.000000 0.000000 0.666667 ]

G  [ 0.833333 0.000000 0.000000 0.000000 0.500000 0.000000 ]

T  [ 0.000000 0.000000 0.000000 0.166667 0.333333 0.333333 ]
```

Note: For the human TRs, we classify them into seven categories, including TF, transcription cofactor, RNA-binding protein (RBP), chromatin remodeler, nuclear enzyme, polycomb group (PcG) protein, and other factors. The classifications have been provided in "resource/human_TR_category.txt".

Note: For most conditions, the TR motifs from a single database (CIS-BP, JASPAR, or HOCOMOCO) are sufficient. We have provided scripts in the GitHub repository to convert the formats of files downloaded from these databases. It will take about 10 minutes to get the final motif file.

4. Prepare the DNase-seq profile in BAM format. Troubleshooting 2.
   a. Download the DNase-seq profile in tagAlign format named "E123-DNase.tagAlign.gz" from the Roadmap Epigenomics Project (Roadmap Epigenomics et al., 2015).
   b. Convert tagAlign format to BAM format using bedtobam sub-command from bedtools (version 2.30.0) (Quinlan and Hall, 2010) and samtools (version 1.12) (Danecek et al., 2021):

```
$zcat E123-DNase.tagAlign.gz | bedtools bedtobam -i stdin
-g hg19.genome | samtools sort - > K562.bam
```

Note: Except for the Roadmap Epigenomics Project web portal, the DNase-seq signal profile for K562 can also be obtained from the ENCODE or GEO databases. Please keep in mind that the DNase-seq profile should be based on the GRCh37/hg19 genome build. It will take no more than 30 minutes to generate the final BAM file on a personal computer.

Note: A BAM file usually uses ".bam" as the suffix and is a compressed binary file to represent the aligned sequencing reads. Please don't try to open it with text editors or use the cat/head command on Unix-like systems. The tool named samtoolsis recommended to process/view/convert BAM files.

5. Filter the TRs by gene expression.
   a. Download gene expression in RPKM (Reads Per Kilobase per Million mapped reads) from the Roadmap Epigenomics Project (Roadmap Epigenomics et al., 2015).
   b. Extract the expression data for K562, which is stored in the column named "E123", and convert the expression value from RPKM to TPM (Transcripts Per Million).
   c. Keep the TRs with a sufficient expression level. The cutoff TPM $\geqq$ 10, which results in 739 TRs remaining is used here.

Note: RPKM, FPKM (Fragments Per Kilobase per Million mapped reads), and TPM are all used to represent gene expression levels from RNA-seq. RPKM is made for single-end RNA-seq, while FPKM is made for paired-end RNA-seq. Compared to RPKM or FPKM, TPM is now becoming popular because the sum of all TPMs is same across samples. This property makes it easier to compare the expression levels of genes in each sample or across samples. For convenience, we have provided scripts in the GitHub repository to convert RPKM/FPKM to TPM and to filter genes by specified TPM cutoff. It will take about 10 minutes to get the final TRs based on the gene expression.

6. Prepare the chromatin loops in BEDPE format. The 10-kb chromatin loops predicted by Peakachu (Salameh et al., 2020) can be downloaded from the "Download" page in the 3D Genome Browser database (Wang et al., 2018). The file "loops-hg19.zip" should be downloaded and saved on the disk. Then the file "Rao_2014.K562.hg19.peakachu-merged.loops" should be extracted from the compressed file and renamed to "K562.bedpe". There are 16,629 chromatin loops in K562. It will take no more than 10 min to get the BEDPE file. Troubleshooting 3.

Note: The chromatin loops for other tissue/cell types from human or other species can be retrieved from ENCODE, GEO, and other databases, accordingly.

Note: The BEDPE format is used to describe pairs of genomic regions. For the "K562.bedpe" file, it contains one chromatin loop per line with the tab-delimited columns. There are six mandatory columns and additional optional columns. "chrom1", "start1", "end1", "chrom2", "start2", and "end2" are the first six mandatory columns and represent the chromosome, start position, and end position of the first and second genomic regions, respectively. If there are optional columns, the first two columns should be "name" and "score", representing the name and score of the chromatin loop, respectively. The detailed definition of the BEDPE format can be acquired from the homepage of bedtools. A snippet of chromatin loops in BEDPE format is shown here:

```
chr10  4850000   4860000   chr10  5660000   5670000
     chr10:4850000-5660000   0.99373
chr10  4860000   4870000   chr10  5420000   5430000
     chr10:4860000-5420000   0.96148
chr10  4860000   4870000   chr10  5140000   5150000
     chr10:4860000-5140000   0.97014
```

Note: 3DCoop does not restrict the resolution of chromatin loops strictly, but resolutions between 5 to 20-kb are recommended. A too low resolution, such as 100-kb, may result in false-positive TR cooperation because it exaggerates the co-binding between TRs.

7. Gather all inputs into one folder. For clarification, we recommend organizing all input datasets into one folder, such as:

```
$mkdir datasets
$cp <PATH>/human_TR_motif.txt datasets/ # Prepare datasets, Step 3
$cp <PATH>/K562.bam datasets/ # Prepare datasets, Step 4
$cp <PATH>/K562.bedpe datasets/ # Prepare datasets, step 6
```

Note: For running 3DCoop on the mouse genome (GRCm38/mm10), we have systematically integrated 1,636 motifs of 836 mouse TRs by collecting and uniformly processing 10 existing TF motif resources. It has been incorporated into the 3DCoop repository as "resource/mouse_TR_motif.txt". As aforementioned for the human motifs, for most conditions, TR motifs from a single database (CIS-BP, JASPAR, or HOCOMOCO) are sufficient. The scripts for converting formats have also been provided in the GitHub repository.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Human TR motifs	Yi et al. (2021)	https://github.com/mulinlab/3DCoop/blob/master/resource/human_TR_motif.txt
K562 Hi-C	3D Genome Browser	http://3dgenome.fsm.northwestern.edu/downloads/loops-hg19.zip
K562 DNase-seq	Roadmap	https://egg2.wustl.edu/roadmap/data/byFileType/alignments/consolidated/E123-DNase.tagAlign.gz

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
K562 RNA-seq	Roadmap	https://egg2.wustl.edu/roadmap/data/byDataType/rna/expression/57epigenomes.RPKM.pc.gz
Software and algorithms		
3DCoop	Yi et al. (2021)	https://github.com/mulinlab/3DCoop
bedtools (version 2.30.0)	Quinlan and Hall (2010)	https://bedtools.readthedocs.io/en/latest/
samtools (version 1.12)	Danecek et al. (2021)	http://www.htslib.org/
PIQ (version 1.3)	Sherwood et al. (2014)	https://bitbucket.org/thashim/piq-single/src/master/
ClusterONE (version 1.0)	Nepusz et al. (2012)	https://paccanarolab.org/cluster-one/
R (version 4.0.5)	(R Core Team, 2021)	https://www.r-project.org/
tidyverse (version 1.3.1)	R package	https://www.tidyverse.org/
huge (version 1.3.5)	R package	https://cran.r-project.org/web/packages/huge/index.html
igraph (version 1.2.6)	R package	https://igraph.org/r/
ggnetwork (version 0.5.10)	R package	https://cran.rstudio.com/web/packages/ggnetwork/index.html
Other		
Equipment	an x86_64 GNU/Linux (CentOS 7.9) platform (Intel Xeon CPU processor E7-4850 V4 and 512 GB of memory)	N/A

MATERIALS AND EQUIPMENT

Computer hardware

3DCoop in this protocol was run and tested on an x86_64 GNU/Linux (CentOS 7.9) platform (Intel Xeon CPU processor E7-4850 V4 and 512 GB of memory). One core with at least 8 GB of memory is minimally required, while four or more cores are recommended. Besides, the 3DCoop pipeline depends on the Perl and R languages, which should be installed on the system. This should already be done if the installation setup was completed successfully.

3DCoop is designed based on the GNU/Linux system, so it can be used on all Unix-like systems, including GNU/Linux and Apple macOS. To run the 3DCoop pipeline on Windows, there is more than one approach. One way is to install a GNU/Linux virtual machine using Oracle VM VirtualBox or VMware Workstation Player. But using a virtual machine is not recommended for the limited resources. An alternative way is to install the Windows Subsystem for Linux (WSL) on Windows 10 or Windows 11. For Windows 10 version 2004 and higher (Build, 19041 and higher) or Windows 11, the following command in an administrator PowerShell or Windows Command Prompt will enable the required optional components, including downloading the latest Linux kernel and installing the default Ubuntu Linux distribution:

```
$wsl -install
```

For the older builds, a step-by-step manual installation is mandatory. The Manual installation steps for older versions of WSL can be referred to.

STEP-BY-STEP METHOD DETAILS

Estimate the genome-wide TR binding events

Timing: days to weeks; computational time scales with sample number and resources

When the TR motifs and the open chromatin profile of a specified cell type are prepared, the genome-wide TR binding events can be estimated by TF footprint analysis. Here, TF footprint sites are identified by using PIQ (version 1.3) (Sherwood et al., 2014) based on the K562 DNase-seq and uniformly integrated TR motifs. PIQ has been incorporated into the 3DCoop repository within the

"PIQ" folder. It can also be obtained from https://bitbucket.org/thashim/piq-single/src/master/. The detailed usage of PIQ is described in the "README.md" file from the Bitbucket repository or the "PIQ/README.md" file from the 3DCoop GitHub repository. We only describe the main key steps here. According to the hardware on which we run the 3DCoop pipeline, nearly one week is needed to get the final genome-wide binding events for all TRs using one thread. On average, it takes 5–15 min per TR with one thread. Troubleshooting 4.

1. Generate the putative binding site via PWM (Position Weight Matrix) across the whole genome. There are 3,105 motifs in our uniformly integrated dataset. This step should be executed for each motif, which can be achieved by using the following script:

```
$cd 3DCoop # Change the work directory to 3DCoop folder if not
$mkdir -p PIQ_results/PWM
$for idx in $(seq 1 3105)
$do
$Rscript PIQ/pwmmatch.exact.r PIQ/common.r datasets/ human_TR_motif.txt
$idx PIQ_results/PWM/
$done
```

   Note: This step does not depend on the cell type and the choice of DNase-seq BAM file. So, the running result can be used many times once executed. Besides, for convenience, we have provided a script in the GitHub repository to run this step in batch mode.

   Note: Before using PIQ, please change the path to the "3DCoop" folder to make sure that the R scripts can be found. Another way is to add the PIQ to the system "PATH" by adding a line such as "export PATH=/path_to_3DCoop/PIQ:$PATH" in the "$HOME/.bashrc" file.

2. Convert the DNase-seq BAM file to the internal binary RData format which can be read by the R language. It will be used in the next step. Troubleshooting 4.

```
$cd 3DCoop # Change the work directory to 3DCoop folder if not
$mkdir -p PIQ_results/BAM
$Rscript PIQ/bam2rdata.r PIQ/common.r >PIQ_results/BAM/K562.RData
datasets/K562.bam
```

   Note: This step does not depend on the choice of motifs. But it should be executed for each cell type.

3. Identify the potential genome-wide TF footprint by combining PWM and BAM. Use the following script to automate such a step:

```
$cd 3DCoop # Change the work directory to 3DCoop folder if not
$mkdir -p PIQ_results/tmp
$mkdir -p PIQ_results/call
```

```
$for idx in $(seq 1 3105)
$do
$Rscript PIQ/pertf.r PIQ/common.r PIQ_results/PWM PIQ_results/tmp
PIQ_results/call PIQ_results/BAM/K562.RData $idx
$done
```

Note: The temporary folder "PIQ_tmp" is used to store some large temporary matrices. It can be deleted after the successful completion of this step. Besides, for convenience, we have provided a script in the GitHub repository to run this step in batch mode.

4. Filter the binding events. As per the recommendation by PIQ, the purity value of 0.7, which means that 70% of instances of motif matches could be true binding sites, can be used to filter the PIQ results.

Note: For each binding site, PIQ will give it a purity value. The purity, which is estimated using the background binding sites, is a proxy for positive predictive value (PPV).

5. Merge the binding events by TRs. This can be achieved by the merge sub-command from bedtools.

Note: The genome-wide binding events are estimated by motifs. The match instances of all motifs should be merged when there is more than one motif for a certain TR.

⚠ CRITICAL: For compatibility reasons, our embedded version of PIQ is recommended for use since there is a small bug in the original codes. We have fixed it in our incorporated version.

Organize inputs

Timing: <10 min

3DCoop needs two main inputs, the TR binding events estimated by computational approaches (or detected by experiments) and the high-resolution chromatin loops. These files should be named and organized correctly (Figure 1).

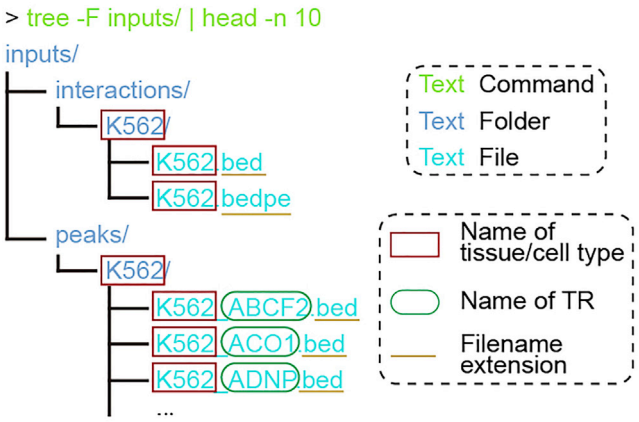


Figure 1. The folder structure of the inputs

The input files must be renamed to the required format. The green, blue, and cyan text indicate the shell command, folder name, and file name, respectively. The rectangle, rounded rectangle, and underscore indicate the name of tissue/cell type, name of TR, and filename extension, respectively.

6. Organize the binding events in BED format. The binding events should be placed in one folder with the cell type information.
   a. Each BED file must be renamed as "<CELL>_<TR>.bed", such as "K562_CTCF.bed".
   b. Organize all BED files into a designed folder. There are 738 BED files for K562 in total.

```
$mkdir -p inputs/peaks/K562

$cp <PATH>/*.bed inputs/peaks/K562/
```

⚠ CRITICAL: The binding events must be stored in BED format with at least three columns: chromosome, start coordinate, and end coordinate.

7. Organize the chromatin loops. Similar to the BED files, the chromatin loops should also be placed in one folder with the cell type information. Troubleshooting 5.
   a. Make sure the chromatin loops are stored in BEDPE format and renamed as "<CELL>.bedpe", such as "K562.bedpe".

```
$mkdir -p inputs/interactions/K562

$cp datasets/K562.bedpe inputs/interactions/K562/
```

⚠ CRITICAL: The chromatin loops must be stored in BEDPE format with eight columns, chromosome 1, start coordinate 1, end coordinate 1, chromosome 2, start coordinate 2, end coordinate 2, loop ID or name, and interaction score. The last column for the interaction score can be set to 1 when this information is unavailable.

   b. Break the chromatin loops into genomic intervals. We provide the "bin/bedpe2bed.pl" script to do this. It will generate a file named "K562.bed" beside "K562.bedpe". There are 23,498 unique genomic intervals in this BED file.

```
$# Usage: perl bin/bedpe2bed.pl <INPUT_DIR (containing one or more BEDPE

files)>

$perl bin/bedpe2bed.pl inputs/interactions/K562
```

Note: The BED (Browser Extensible Data) format provides a flexible way to represent genomic features. It contains one feature per line with tab-delimited columns. Three columns are required and nine additional columns are optional. The first three required columns are "chromosome", "start", and "end", representing the chromosome name, start position, and end position of the genomic feature, respectively. More details about the BED format can be found on the page "Data File Formats" from UCSC. Here is a snippet of the genomic intervals used for this protocol in BED format:

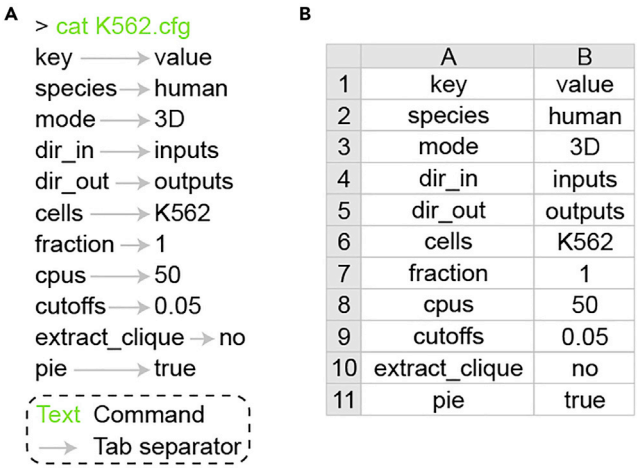
```
chr1    850000    860000

chr1    900000    910000

chr1    910000    920000

chr1    960000    970000

chr1    1000000    1010000
```



	A	B
1	key	value
2	species	human
3	mode	3D
4	dir_in	inputs
5	dir_out	outputs
6	cells	K562
7	fraction	1
8	cpus	50
9	cutoffs	0.05
10	extract_clique	no
11	pie	true

Figure 2. The configuration file

(A and B) The configuration file in the text view (A) and Microsoft Excel view (B). The green text indicates the shell command. The gray arrow indicates the tab separator.

Make configuration file

Timing: <5 min

A configuration file for tuning the 3DCoop pipeline is needed. The given name of the configuration file will be passed as a parameter to the scripts. For clarification, the name "K562.cfg" is used here.

8. Write the configuration file. The configuration file is a text file with two columns (tab-delimited) and several rows. A real configuration file like this (Figure 2):

```
key  value
species  human
mode  3D
dir_in  inputs
dir_out  outputs
cells  K562
fraction  1
cpus  50
cutoffs  0.05
extract_clique  no
pie  true
```

Note: Here, we briefly explain the keys and corresponding values in the configuration file. "species" can be set to "human" or "mouse". "mode" can be set to "3D" or "1D" to indicate the running mode. "dir_in" and "dir_out" indicate the folders for inputs and outputs. "cells" indicates which cell type will be processed. "fraction" indicates a minimal overlap fraction of a peak to assign it to a certain interaction loop. "cpus" specifies the number of threads. "cutoffs" specifies the cutoff for cluster extraction. "extract_clique" indicates whether to find

and extract all cliques (subsets of TRs, all adjacent to each other) from each cluster. "pie" indicates whether to display the pie chart for each TR to show the TR categories.

Note: When the chromatin loops are not available, the user can use 3DCoop in 1D mode by setting "mode" to "1D". Troubleshooting 5.

Note: 3DCoop can be used for several tissue/cell types in batch mode. Please refer to the usage manual for details.

Note: "cutoffs" is recommended to be set to "auto" to define the cutoff automatically for the first time. Then the user can choose a detailed cutoff and change it in the configuration file based on the results. Troubleshooting 6.

Note: "no" is recommended for "extract_clique" because enabling it will use a huge amount of memory and take a long time to extract cliques from large clusters.

Note: "true" for "pie" is only valid for human. Please set "false" for mouse.

Run 3DCoop pipeline

Timing: <1 h; computational time scales with sample number and resources

The 3DCoop pipeline starts with the TR binding events and the chromatin loops to get the TR-specific contact maps. Then the generalized Jaccard similarity is used to construct the TR pair-wise correlation matrix, and the graphical Lasso algorithm (Glasso) is adopted to estimate the precision matrix. Finally, the overlapping clustering method is incorporated to compute communities, and then the TR clusters, TR maximum cliques, and TR pairs are extracted. These steps have been split into sequential scripts. The only mandatory one is the configuration file.

⚠ CRITICAL: Make sure that the "3DCoop" conda environment has been activated before running the pipeline. If not, it can be activated by "conda activate 3DCoop". Troubleshooting 1.

9. Prepare data for Jaccard calculation. This step is to connect the binding events and the chromatin loops for each TR and build the TR-specific contact maps for all TRs that reflect the TR co-binding in 3D chromatin. The chromatin interaction is assigned as the TR-specific contact map when a peak can be mapped to either end of this interaction.

```
$# Usage: perl bin/01_02_prepare4jaccard.pl <CONFIGURATION_FILE>

$perl bin/01_02_prepare4jaccard.pl K562.cfg
```

10. Calculate the Jaccard for TR pairs. The generalized Jaccard similarity is calculated by considering the interaction intensity of each TR-associated contact. A TR pair-wise correlation matrix will be generated.

```
$# Usage: perl bin/03_jaccard.pl <CONFIGURATION_FILE>

$perl bin/03_jaccard.pl K562.cfg
```

⚠ CRITICAL: For hundreds of TRs, please set "cpus" in the configuration file to use multiple threads (20 or more) for saving running time.

11. Estimate the precision matrix. Based on the TR pairwise correlation matrix, the Glasso is adopted to estimate the precision matrix to reduce the false positive rate of potential TRs dependency. The copula nonparanormal graphical model is used with the huge package (version 1.3.5) in R.

```
$# Usage: Rscript bin/04_glasso.R <CONFIGURATION_FILE>

$Rscript bin/04_glasso.R K562.cfg
```

12. Identify the TR clusters. The overlapping clustering method, which allows a single TR to be involved in multiple cooperation communities, is incorporated to estimate the network modules using ClusterONE (version 1.0) (Nepusz et al., 2012). Then, the igraph R package (version 1.2.6) is used to analyze and extract the TR clusters, TR maximum cliques, and TR pairs based on the results from ClusterONE.

```
$# Usage: perl bin/05_clusterone.pl <CONFIGURATION_FILE>

$perl bin/05_clusterone.pl K562.cfg
```

13. Visualize the TR cooperation network. The TR cooperation network is visualized using the igraph and ggnetwork (version 0.5.10) R packages from different angles, including with or without the TR names, with or without the TR categories.

```
$# Usage: perl bin/06_network.pl <CONFIGURATION_FILE>

$perl bin/06_network.pl K562.cfg
```

Applications

Timing: hours to days; computational time scales with sample number and resources

The TR cooperation detected by the 3DCoop pipeline can be used in many biological scenarios, such as interpreting the disease-causal variants identified by a genome-wide association study, understanding the dynamics during cell development or differentiation, and so on. For convenience, we have provided a script to identify the TRs and TR pairs associated with given disease-causal variants using TR cooperation information:

```
$perl bin/map_variant2TRpair.pl K562.cfg K562 variants.bed
```

The program can be used to get the variant-associated TRs and TR pairs based on the coordinates of variants. The variants are stored in BED format and should be provided as an additional input. The script will produce one folder named "07_variants2TRs". "variants2TRpairs.txt", which stores the relationship between variants, TRs, and TR pairs, is the key output file. Besides, "snp2peak2bin2loop.txt" reports the mapping relationship between variants, TR peaks, genome bins, and chromatin loops. A detailed description of inputs and outputs can be found in the usage manual.

EXPECTED OUTCOMES

When complete, the 3DCoop pipeline will generate six folders, corresponding to each step (steps 9–13), in the output folder specified in the configuration file (Figures 3A and 3B). The first four folders contain all intermediate files, including the results in BED and BEDPE formats by intersecting the TR binding events and the chromatin loops ("01_intersection_bed" and "02_intersection_bedpe"), the

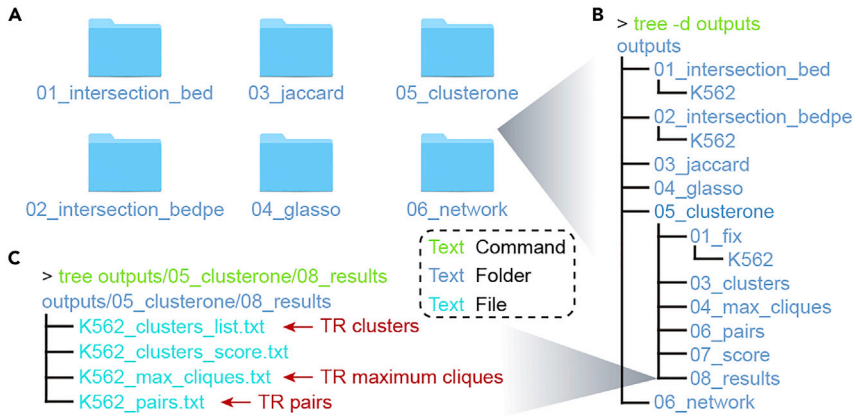


Figure 3. The folder structure of the outputs

(A–C) The folder structure of the outputs (A and B) and filenames for the key results (C). The green, blue, and cyan text indicate the shell command, folder name, and file name, respectively. The filenames for TR clusters, TR maximum cliques, and TR pairs are pointed.

TR pair-wise correlation matrix based on the generalized Jaccard similarity ("03_jaccard"), and the precision matrix based on the Glasso ("04_glasso").

The folder named "05_clusterone" stores the results from ClusterONE and the modularity analysis of TR cooperation. The files in "05_clusterone/08_results" are the key results, including the detected TR clusters, TR maximum cliques, and TR pairs (Figure 3C). The TR clusters are stored in "05_clusterone/08_results/<CELL>_clusters_list.txt" with each line per cluster. The TRs in one cluster are sorted in alphabetical order and separated by tabs (Figure 4A). The TR maximum cliques and TR pairs are stored in "05_clusterone/08_results/<CELL>_max_cliques.txt" and "05_clusterone/08_results/<CELL>_pairs.txt", respectively. They are all stored in tabular format with the first line as the header and one TR maximum clique or TR pair per line in the main content (Figures 4B and 4C). Besides, the TR cooperation network is plotted in the PNG and PDF formats ("06_network", Figure 5).

LIMITATIONS

The 3DCoop pipeline relies heavily on publicly available data, which can be both a benefit and an obstacle. It is likely that only a subset of data is available for certain tissue/cell types. In such a case, it is the user's choice to run this protocol in 3D mode or alternatively 1D mode, using ChIP-seq peaks or computational TF footprint.

This protocol aims to detect TR cooperation based on a large number of TRs. Using fewer TRs may not guarantee a successful run, but it is worth trying. It is recommended to include hundreds of TRs (~100–800).

As aforementioned, the TR cooperation identified by the 3DCoop pipeline can be used to better understand biological processes, such as interpreting disease-causal variants and detecting dynamic cooperation. However, the true TR cooperation should be validated by the experiments with the matched conditions.

TROUBLESHOOTING

Problem 1

The conda environment cannot be activated using "conda". The error might be as follows:

```
CommandNotFoundError: Your shell has not been properly configured to use 'conda deactivate'
```

A TR clusters

	A	B	C	D	E	F	G
1	ARRB1	ASXL1	ASH2L	CTNNB1	DIDO1	ELF1	...
2	BRCA1	CCNT2	ESRRA	ESRRB	HP1BP3	ING3	...
3	BRCA1	ESRRA	ESRRB	HP1BP3	ING3	MDM2	...
4	AEBP2	ARID1B	BACH1	BCLAF1	CBX1	CDK6	...
5	CBX3	CDC5L	CHD4	DTL	FOXA3	FOXJ3	...
6	BRCA1	CCNT2	ESRRA	ESRRB	GLYR1	ING3	...
7	DIABLO	ICE1	JMJD6	MITF	MLX	SIN3A	...
8	FOXA3	FOXJ3	FOXK2	FOXO3	FOXO4		
9	ADNP	BCOR	BMI1	BMPR1A	BRD4	CDK7	...
10	DGCR8	GIT2	LARP1	MIEF1	NMI	SNAPC4	...
11	ATF1	ATF4	ATF6	ATF6B	ATF7	CREB1	...
12	SOAT1	STAT1	STAT3	STAT5A	STAT5B		
13	...	...	...	...	...	...	...

C TR pairs

	A	B	C	D
1	TF1	TF2	jaccard	glasso
2	ABCF2	ANXA1	0.14813	0.07936
3	ABCF2	ATF5	0.15347	0.07702
4	ABCF2	CBFA2T2	0.23346	0.09753
5	ABCF2	CEBPB	0.18966	0.10221
6	ABCF2	FOXM1	0.16895	0.06386
7	ABCF2	GTF3A	0.16226	0.08544
8	ABCF2	GZF1	0.13211	0.02884
9	ABCF2	NF1	0.15238	0.04501
10	ABCF2	NFAT5	0.16917	0.06989
11	ABCF2	NFKBIA	0.16582	0.09868
12	ABCF2	NONO	0.21025	0.13695
13	...	...	...	...

B TR maximum cliques

	A	B	C	D	E	F
1	clique	number	jaccardSum	jaccardMean	glassoSum	glassoMean
2	ABCF2-ANXA1-CBFA2T2-FOXM1-...	13	11.39335	0.14606	6.43755	0.08253
3	ABCF2-ANXA1-CBFA2T2-NF1-...	12	8.98203	0.13609	5.36702	0.08131
4	ABCF2-ANXA1-FOXM1-GZF1-NF1-...	10	5.73134	0.12736	3.52911	0.07842
5	ABCF2-ANXA1-FOXM1-GZF1-NF1-...	13	10.41171	0.13348	5.88071	0.07539
6	ABCF2-ANXA1-FOXM1-GZF1-NF1-...	13	10.63159	0.13631	5.98065	0.07667
7	ABCF2-ANXA1-FOXM1-GZF1-NFAT5-...	9	4.3564	0.12101	2.72392	0.07566
8	ABCF2-ANXA1-FOXM1-GZF1-NFKBIA-...	11	7.22302	0.13132	4.01625	0.07302
9	ABCF2-ANXA1-GZF1-NF1-NFAT5-...	10	5.31178	0.11803	3.30541	0.07345
10	ABCF2-ANXA1-GZF1-NF1-NFKBIA-...	13	9.80481	0.12571	5.71253	0.07323
11	ABCF2-ANXA1-GZF1-NFAT5-NFKBIA-...	9	4.00291	0.11119	2.61022	0.072506
12	ABCF2-ANXA1-GZF1-NFKBIA-NONO-...	11	6.72318	0.12223	3.90241	0.07095
13	ABCF2-ATF5-CEBPB-GTF3A-NF1-...	11	7.7587	0.14106	4.47315	0.08133
14	...	...	...	...	...	...

Figure 4. The key results

(A–C) The Microsoft Excel view for TR clusters (A), TR maximum cliques (B), and TR pairs (C). The dotted lines on the right and lower bounds indicate that the columns and rows continue, respectively.

Potential solution

Firstly, please restart the shell by opening a new terminal. If the error still exists, try to enable the automatic base environment activation by entering "conda config –set auto_activate_base true" on the terminal. If all these actions fail, we suggest using "source activate" instead of "conda activate" to activate the environment, and "source deactivate" instead of "conda deactivate" to exit the environment. For convenience, we have provided the Frequently Asked Questions (FAQs) in the GitHub repository to discuss the problems with conda and their corresponding solutions.

Problem 2

The DNase-seq dataset is not available.

Potential solution

The open chromatin data is required for estimating the TR binding events. Not only the DNase-seq but also the ATAC-seq data can be used. Except for the Roadmap Epigenomics Project, the ENCODE and GEO databases also provide the DNase-seq/ATAC-seq open chromatin profiles for many tissue/cell types.

Problem 3

The chromatin loops predicted from Hi-C are not available.

Potential solution

The chromatin loops can be reached from several 3C-based technologies, such as ChIP-PET and HiCHIP, not only Hi-C. These 3D genome datasets can be used for trying. Except for the 3D Genome

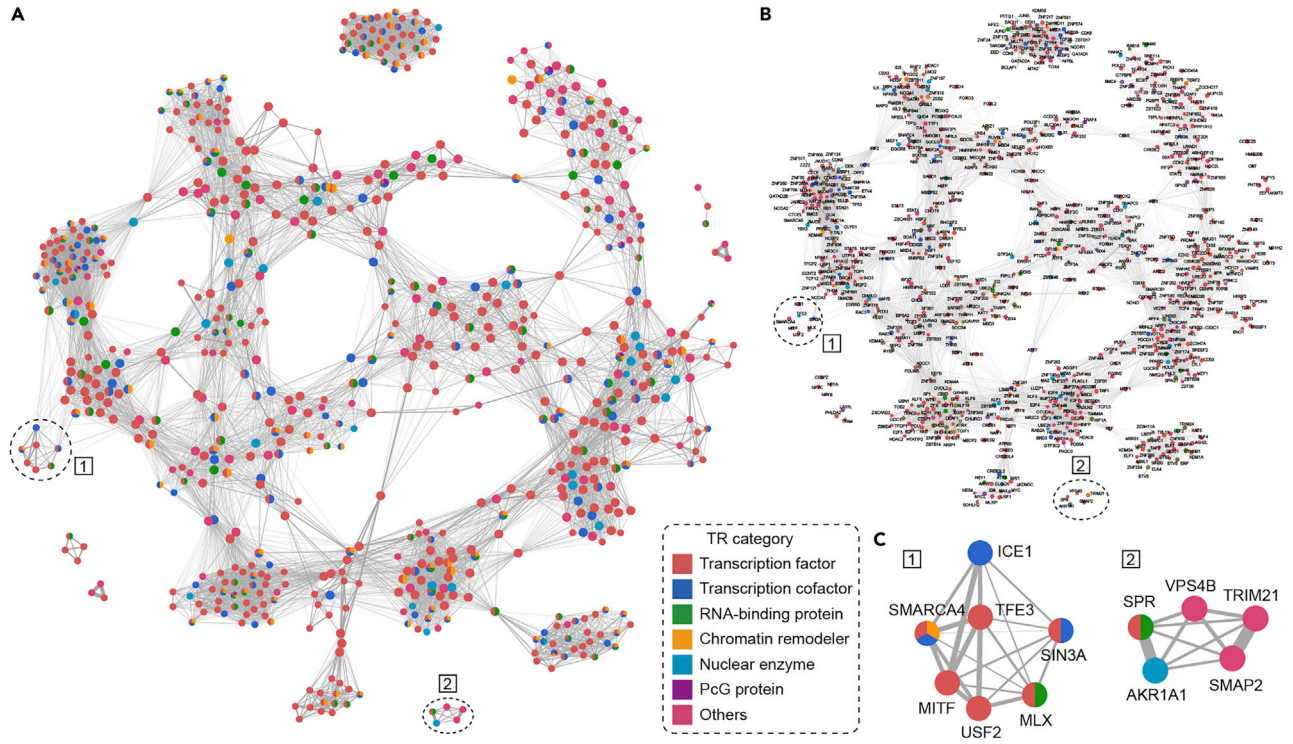


Figure 5. The TR cooperation network

(A–C) The TR cooperation network using the same layout with TR categories (A) or TR names (B) and two zoomed examples of TR cooperation (C). The categories for each TR are shown using the pie plot in (A and C). TRs are colored just for distinction in (B).

Browser database, the ENCODE and GEO databases can also be used to search the required 3D genome data for desired tissue/cell types. But please keep in mind the identity of chromatin interactions represented by these different datasets. For example, ChIP-PET only captures the query protein-directed chromatin conformation mediated by the specific protein of interest, while Hi-C captures all possible interactions between fragments from the whole genome.

Problem 4

ATAC-seq but not DNase-seq dataset is available.

Potential solution

According to the FAQs of PIQ, ATAC-seq data with the same coverage seems to perform similarly to DNase-seq. Users are encouraged to try different tools or methods to deal with ATAC-seq and DNase-seq data, such as HINT (Li et al., 2019), DNase2TF (Sung et al., 2014), TRACE (Ouyang and Boyle, 2020), and TOBIAS (Bentsen et al., 2020).

Problem 5

There is no 3D genome data available for certain tissue/cell types.

Potential solution

The 3DCoop pipeline can be used in 1D mode when the chromatin loops are not available. The genome is binned at a certain length specified by the user, and then the TR binding sites are mapped to these bins. The details for using 3DCoop in 1D mode have been described in the usage manual.

Problem 6

Do not know how to choose the cutoff for ClusterONE.

Potential solution

The cutoff sets the minimum density of predicted complexes. For hundreds of TRs, it should be set to a value between 0.05 and 0.3. We have provided an "auto" mode to choose the best cutoff automatically. When the user doesn't know how to set the cutoff, the "auto" mode can be used. The user can choose the cutoff based on the results from the "auto" mode and then set this cutoff in the configuration file to run this step again.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Mulin Jun Li (mulinli@connect.hku.hk).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The codes used during this study are available at https://github.com/mulinlab/3DCoop. A version containing all scripts, tools, resources, and testing data sets has been released and can be found at https://doi.org/10.5281/zenodo.6464106.

ACKNOWLEDGMENTS

This work was supported by the following grants: the National Natural Science Foundation of China (32070675 to M.J.L. and 31871327 to M.J.L.) and the Natural Science Foundation of Tianjin (19JCJQJC63600 to M.J.L. and 19JCQNJC09000 to X.Y.). Part of the data used in the analyses described in this article was obtained from the Roadmap Epigenomics Project, the 3D Genome Browser, and other resources. We appreciate all tool and resource providers.

AUTHOR CONTRIBUTIONS

X.Y. wrote the code, analyzed the data, and wrote the manuscript. M.J.L. supervised the project and wrote the manuscript. M.L., X.F., Y.Z., and J.W. evaluated the computational tool and reviewed the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Bentsen, M., Goymann, P., Schultheis, H., Klee, K., Petrova, A., Wiegandt, R., Fust, A., Preussner, J., Kuenne, C., Braun, T., et al. (2020). ATAC-seq footprinting unravels kinetics of transcription factor binding during zygotic genome activation. Nat. Commun. *11*, 4267. https://doi.org/10.1038/s41467-020-18035-1.

Danecek, P., Bonfield, J.K., Liddle, J., Marshall, J., Ohan, V., Pollard, M.O., Whitwham, A., Keane, T., McCarthy, S.A., Davies, R.M., and Li, H. (2021). Twelve years of SAMtools and BCFtools. Gigascience *10*, giab008. https://doi.org/10.1093/gigascience/giab008.

Fornes, O., Castro-Mondragon, J.A., Khan, A., van der Lee, R., Zhang, X., Richmond, P.A., Modi, B.P., Correard, S., Gheorghe, M., Baranasic, D., et al. (2020). Jaspar 2020: update of the open-access database of transcription factor binding profiles. Nucleic Acids Res. *48*, D87–D92. https://doi.org/10.1093/nar/gkz1001.

Gruning, B., Dale, R., Sjodin, A., Chapman, B.A., Rowe, J., Tomkins-Tinch, C.H., Valieris, R., Koster, J., and Bioconda, T. (2018). Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat. Methods *15*, 475–476. https://doi.org/10.1038/s41592-018-0046-7.

Kulakovskiy, I.V., Vorontsov, I.E., Yevshin, I.S., Sharipov, R.N., Fedorova, A.D., Rumynskiy, E.I., Medvedeva, Y.A., Magana-Mora, A., Bajic, V.B., Papatsenko, D.A., et al. (2018). HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. Nucleic Acids Res. *46*, D252–D259. https://doi.org/10.1093/nar/gkx1106.

Li, Z., Schulz, M.H., Look, T., Begemann, M., Zenke, M., and Costa, I.G. (2019). Identification of transcription factor binding sites using ATAC-seq. Genome Biol. *20*, 45. https://doi.org/10.1186/s13059-019-1642-2.

Nepusz, T., Yu, H., and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods *9*, 471–472. https://doi.org/10.1038/nmeth.1938.

Ouyang, N., and Boyle, A.P. (2020). TRACE: transcription factor footprinting using chromatin accessibility data and DNA sequence. Genome Res. *30*, 1040–1046. https://doi.org/10.1101/gr.258228.119.

Quinlan, A.R., and Hall, I.M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics *26*, 841–842. https://doi.org/10.1093/bioinformatics/btq033.

R Core Team, 2021. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/.

Roadmap Epigenomics, C., Kundaje, A., Meuleman, W., Ernst, J., Bilenky, M., Yen, A., Heravi-Moussavi, A., Kheradpour, P., Zhang, Z., Wang, J., et al. (2015). Integrative analysis of 111 reference human epigenomes. Nature *518*, 317–330. https://doi.org/10.1038/nature14248.

Salameh, T.J., Wang, X., Song, F., Zhang, B., Wright, S.M., Khunsriraksakul, C., Ruan, Y., and Yue, F. (2020). A supervised learning framework for chromatin loop detection in genome-wide contact maps. Nat. Commun. *11*, 3428. https://doi.org/10.1038/s41467-020-17239-9.

Sherwood, R.I., Hashimoto, T., O'Donnell, C.W., Lewis, S., Barkal, A.A., van Hoff, J.P., Karun, V., Jaakkola, T., and Gifford, D.K. (2014). Discovery of directional and nondirectional pioneer transcription factors by modeling DNase profile magnitude and shape. Nat. Biotechnol. *32*, 171–178. https://doi.org/10.1038/nbt.2798.

Sung, M.H., Guertin, M.J., Baek, S., and Hager, G.L. (2014). DNase footprint signatures are dictated by factor dynamics and DNA sequence. Mol. Cell *56*, 275–285. https://doi.org/10.1016/j.molcel.2014.08.016.

Vorontsov, I.E., Kulakovskiy, I.V., and Makeev, V.J. (2013). Jaccard index based similarity measure to compare transcription factor binding site models. Algorithms Mol. Biol. *8*, 23. https://doi.org/10.1186/1748-7188-8-23.

Wang, Y., Song, F., Zhang, B., Zhang, L., Xu, J., Kuang, D., Li, D., Choudhary, M.N.K., Li, Y., Hu, M., et al. (2018). The 3D Genome Browser: a web-based browser for visualizing 3D genome organization and long-range chromatin interactions. Genome Biol. *19*, 151. https://doi.org/10.1186/s13059-018-1519-9.

Weirauch, M.T., Yang, A., Albu, M., Cote, A.G., Montenegro-Montero, A., Drewe, P., Najafabadi, H.S., Lambert, S.A., Mann, I., Cook, K., et al. (2014). Determination and inference of eukaryotic transcription factor sequence specificity. Cell *158*, 1431–1443. https://doi.org/10.1016/j.cell.2014.08.009.

Yi, X., Zheng, Z., Xu, H., Zhou, Y., Huang, D., Wang, J., Feng, X., Zhao, K., Fan, X., Zhang, S., et al. (2021). Interrogating cell type-specific cooperation of transcriptional regulators in 3D chromatin. iScience *24*, 103468. https://doi.org/10.1016/j.isci.2021.103468.