SUPPLEMENTAL METHODS

Crosslinking

Cells were cultured in T-175 flasks containing 10 × 10^6 cells for Hi-C and 5 × 10^6 cells for CUT&RUN at 1 × 10^5 cells/mL. The collection protocol differs for where the cells are at during the differentiation. For suspended cells (0h and 6h): Media was centrifuged at 300 × g for 5 min. Cells are in suspension for 0h and 6h. Suspended cells were collected and spun down 300 × g for 5 min. Pellets were resuspended in 10 mL of 1% formaldehyde (Thermo Fisher Scientific, cat # 28908) in RPMI for 10 min with rotation. For semi-adherent cells (72h): The suspended cells were harvested and centrifuged for 300 × g for 5 min. Pelleted cells were resuspended in 10 mL of 1% formaldehyde, and added to the adherent cells in the T-175 flask to crosslink all cells at once, where they were put on the shaker for 10 min (from here, the protocol is the same for all time points). Cells were quenched with cold glycine for 5 min (Invitrogen, cat # 15527013) to a final concentration of 2.0 M for 5 min. Cells were then centrifuged at 562 × g, resuspended in cold PBS (Corning, cat # 21040CV), and split into 3 tubes of approximately 3 × 10^6 cells each (HiC) or 10 tubes of approximately 5 × 10^5each (CUT&RUN). Cells were spun again at 562 × g for 5 min and washed again with cold PBS, then aspirated and flash frozen in liquid nitrogen, and stored at -80ºC.

In situ Hi-C library preparation

Four treatments (biological replicates) were performed. For each treatment, either two or three frozen pellets (3 × 10^6 each) were used to generate technical replicates (3 technical replicates for the first two biological replicates and 2 technical replicates for the last two biological replicates). Libraries were generated according to the protocol as described in Rao et al(Rao et al. 2014). Briefly, crosslinked we lysed the cells, isolated nuclei, and used MboI (New England Biolabs, cat

R0147L) to digest chromatin overnight. The fragment ends were biotinylated, proximity ligated, and reverse crosslinked. Quantification of shearing DNA was achieved with Qubit (dsDNA Broad Range (BR) assay) (Thermo Fisher Scientific, cat # Q32850). We then sheared the samples on a Covaris LE 220 (duty factor 25, PIP 500, 200 cycles/burst, 90 seconds). 2% of each sample was run on a 2% agarose gel to confirm fragmentation. Size selection with AMPure XP beads (Beckman Coulter, cat # A63881) was then performed to select for DNA fragments between 300 and 500 bp. Biotinylated chromatin was pulled down with streptavidin beads. Biotin was then removed from unligated ends and the libraries were end repaired. We added the Illumina TruSeq Nano (Set A) (Illumina, cat # 20015960) indices to each sample in a combination appropriate for pooling and amplified using 9 cycles of PCR. Final quantification was achieved using Qubit (dsDNA High Sensitivity (HS) assay) (Thermo Fisher Scientific, cat # Q32851) and TapeStation (D1000 screentape) (Agilent, cat # 5067-5584). Libraries were pooled to 10 nM and sequenced across 7 Illumina NovaSeq S4 lanes (Novogene, 150-bp paired-end).

RT-qPCR

We extracted RNA from 5 × 10^5 cells using the QIAGEN RNeasy Mini kit (Qiagen, cat # 74014) with DNase I treatment (Qiagen, cat # 79254) and quantified with a Qubit Broad Range assay (Thermo Fisher Scientific, cat # Q32850). Reverse transcription into cDNA was performed with the iScript cDNA synthesis kit (Bio-Rad, cat # 1708891). qPCR was performed with the TaqMan reagents using probes for ITGB3, KLF1, and GAPDH (Thermo Fisher Scientific, cat # Hs01001469, Hs00610592, Hs02786624).

RNA-seq library preparation

We extracted RNA from 5 × 10^5 cells using the QIAGEN RNeasy Mini kit with DNase I treatment. To confirm quality of libraries, we checked RNA integrity numbers with a TapeStation RNA screentape (Agilent, cat # 5067-5577) and confirmed them all to be above 9.7. We determined the concentration of all RNA samples with the Qubit Broad Range assay (Thermo Fisher Scientific, cat # Q10211).

The KAPA RNA HyperPrep kit with RiboErase (HMR) (Kapa Biosciences, cat # KK8560) was used for library preparation. Illumina TruSeq adapters (Illumina, cat # 20015960) were diluted and 0.0075 nmol was added to each sample. We determined library concentration and fragments size with Qubit (dsDNA HS assay) and TapeStation (D1000 screentape). Libraries from each timepoint were pooled at 10 nM and each biological replicate was sequenced on an Illumina NextSeq 500 (75-bp paired-end, high output kit)

ATAC-seq library preparation

We used the Omni ATAC-seq protocol as described in Corces et al(Corces et al. 2017) with some adjustments to perform ATAC-seq. Two treatments (biological replicates) were performed. For untreated and 6h, cells were harvested and centrifuged at 500 × g for 5 min. For semi-adherent 72h cells, all of the floating cells were harvested. Adherent cells in each well were washed with 2 mL PBS, lifted with 500 µL 0.5 M EDTA for 5 min, and quenched with 3 mL of RPMI before combining with the floating cells. 5×10^5 cells were used for library preparation. Illumina Nextera XT indices (Illumina, cat # FC-131-1001) (3.75 µL/sample) were used for PCR.

After 5 PCR cycles, 5% of each sample was used in qPCR to determine how many more cycles were necessary. We found that 4-7 cycles were sufficient for final amplification. AMPure XP beads were used to perform a final cleanup (0.5X followed immediately by 1.3X) and quantified with the Qubit (ds DNA HS assay). The concentration in molarity of samples was determined by the KAPA Library Quantification Kit (Kapa Biosystems, cat # 4854). Each replicate was pooled to 4 nM and sequenced separately on an Illumina NextSeq 500 (75-bp paired-end, high output kit).

CUT&RUN library preparation

We generated CUT&RUN libraries following existing protocols(Skene and Henikoff 2017), but modified for the use of crosslinked cells. Cells were centrifuged at 500 × g at 4ºC for 10 min. For H3 K27ac, 0.5 µL of 1:10 diluted antibody (Abcam, cat # ab4729) was added to each sample. For CTCF, 0.5µL of 1:10 diluted antibody (Thermo Fisher Scientific, cat # MA5-31344) was added to each sample. For JUN, 2.08 µL of stock antibody (Thermo Fisher Scientific, cat # MA5-15172) was added to each sample. For RAD21, 0.625 µL of stock antibody (Abcam, cat # ab992) was added to each sample. We then added 5 µL of KAPA Unique Dual-Indexed Adapters (Roche, cat # 08861919702) diluted to 750 nM. Libraries from each timepoint for H3 K27ac, CTCF, and JUN were pooled to 6 nM, and sequenced on an Illumina NextSeq 500 (75-bp paired-end, high output kit). RAD21 libraries from each timepoint were pooled to 9 nM, and were sequenced on an Illumina NextSeq 500 (75-bp paired-end, high output kit)

Hi-C data processing and calling compartments, domains, and loops

We processed our Hi-C data using a modified version of the Juicer pipeline (version 1.9.8) (Rao et al. 2014). Hi-C contact maps were generated at 5, 10, 25, 50, 100, 200, 250, 500, 1000, and 2500-kb resolution for each individual technical replicate that was sequenced. This was for 4 biological replicates, 3 timepoints, and 2-3 technical replicates each, totaling 30 unique samples. Additionally, all of the samples for each timepoint were merged to create merged Hi-C maps. All samples and replicates across all timepoints were also merged to create a “Mega” map.

All downstream computational analysis was performed in R (R Core Team)

Compartments were identified using the EigenVector R package at a 10-kb resolution (Olshansky et al. 2021).

TADs were identified using the arrowhead command within the Juicer pipeline at 25-kb resolution. Cell type specific TADs were identified by merging with the mariner R package and using the denovo function.

Loops were called from the merged timepoint Hi-C files and Mega map with SIP (Rowley et al. 2020) (version 1.6.1). The settings “-g 2 -5 2000 -fdr 0.05” were used both on the timepoint and the Mega map. Loops were merged in R with mariner using the mergeBedpe function, providing a list of 33,914 loops.

A count matrix was prepared using mariner (https://github.com/EricSDavis/mariner), where unnormalized counts at each loop pixel from each technical replicate were extracted.

The compartment, TAD, and loop-level Hi-C maps were SCALE normalized and visualized with plotgardener (Kramer et al. 2022) at 100-, 10-, and 5-kb resolutions respectively.

K562 hic files from Belaghazl et al 2021 were downloaded and we called loops with SIP with the same parameters used in this study. To compare our data, we subsampled the K562 data to a sequencing depth of 500 million reads and called loops with the same SIP parameters. The data from Belaghazl was lifted over with the liftOver function within UCSC tools (Kent et al. 2002) to hg38 before overlapping the loop calls.

Perturb-seq from (Gasperini et al 2019) and CRISPRi valid pairs (Fulco et al 2019) were downloaded and lifted over to hg38 and were filtered for pairs longer than 30-kb. Overlaps were performed with the GRanges function subsetByOverlaps. These findings are shown in supplemental table S3.

Differential loop and aggregate peak analysis

DESeq2 was used to identify differential loops using the count matrix prepared as described above. Loops with a median count of 5 counts or less were filtered out. Counts from the technical replicates from each biological replicate were summed together and the design “~rep + time” was used, with a reduced design of “~rep” used to form a likelihood ratio test (LRT). Apeglm was used to calculate $\log_2$ fold changes for each loop, comparing both 6 and 72h to 0h. Loops were deemed significant if they had an adjusted p-value < 0.05 and a $\log_2$(fold-change) > 1.5.

Aggregate peak analysis (APA) was performed with mariner. For all, gained, and lost loops, the loop pixel and 10 pixels around the loop were extracted with SCALE normalization at 100-kb resolution.

Additionally, we used the hicdcdiff function of the HiC-DC+ package (Sahin et al. 2021) to identify differential loops. The input of hicdcdiff was files with summed counts across technical replicates for each condition with genomic distance D (start ranges2 - start ranges 1). Hicdcdiff estimates distance-dependent DESeq2 normalization factors and identifies differential loops on each chromosome. We identified 2145 differential loops using hicdcdiff. More than 77% of the differential loops identified by DESeq in our analysis were also identified with hicdcdiff.

Hi-C Power Analysis

Power analysis was performed with the RNAPower package (Hart et al. 2013). Dispersion was calculated from the differential loop analysis in DESeq2 as described above, where the minimum dispersion value was used. Power was modeled across various theoretical sequencing depths and replicates for identifying a $\log_2$(fold-change) of 2 with a p-value of 0.05. The rnapower function was used with an alpha of 0.05/33914 to account for multiple hypothesis testing and a cv of the square root of the dispersion value.

We subsampled our Hi-C data from the merged_nodups files to approximate sequencing depths of 100, 300, 500, and 700M per biological replicate. We then repeated our differential loop analysis using the subsampled data for either 2, 3, or 4 replicates using all of the same parameters and loop calls.

RNA-seq processing

FASTQ quality was assessed using the FastQC (version 0.11.5, https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) and MultiQC tools (version 1.5) (Ewels et al. 2016). FASTQ files were trimmed with Trim Galore! (version 0.4.3, https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/) and quantified with Salmon (version 1.4.0) to the hg38 genome (Patro et al. 2017). Alignment was performed using HISAT2 (version 2.1.0), which generated BAM files that were indexed using SAMtools (version 1.9) (D. Kim et al. 2019; Danecek et al. 2021). The BAM files for each timepoint's two biological replicates were merged using SAMtools and converted to bigWigs using deepTools (version 3.0.1) (Ramírez et al. 2016) for easy visualization of signal tracks. Reads were summarized into a format compatible with DESeq2 using txImport (r version 3.3.1, tximport version 1.2.0) (Soneson, Love, and Robinson 2015; Love, Huber, and Anders 2014).

Differential gene analysis

DESeq2 was again used to identify differential genes. The txi file was used as input, and the DESeqDataSetFromTximport was used with "~rep + time" as the design. A reduced design of "~rep" was used to form an LRT, as previously for peak analysis. Shrunken $\log_2$(fold-change) values were calculated for each gene by comparing the counts at each time point to 0h with apeglm (Zhu, Ibrahim, and Love 2019). Significant genes had an adjusted p-value < 0.05 and a $\log_2$(fold-change) > 2.

The DESeq2 dataset was normalized with variance stabilized transformation. We then filtered for differential genes and calculated *Z*-scores based on standard deviation and mean. Replicates were then averaged, and *k*-means clustering was used to identify 6 temporal clustered based on the vectors of *Z*-scores.

To compare biological replicates to each other, the counts for each gene from the DESeq2 design were correlated with the base R cor function.

ATAC-seq processing and peak calling

Adapters were trimmed and low quality reads were filtered out using Trim Galore! (version 0.4.3, https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/). BWA-MEM (version 0.7.17) was used to align reads to the hg38 genome and sorted using SAMtools (version 1.9) (Danecek et al. 2021). PicardTools (version 2.10.3, https://broadinstitute.github.io/picard/) was used to remove duplicate reads. Mitochondrial reads were filtered out with SAMtools idxstats (Danecek et al. 2021). For each timepoint, biological replicates were merged and indexed with SAMtools. We called peaks were called on the merged files using MACS2 with the following parameters: -f BAM -q 0.01 -g hs --nomodel --shift 100 --extsize 200 --keep-dup all -B --SPMR (version 2.1.1.20160309) (Zhang et al. 2008). A comprehensive peak list was generated by merging peaks across all time points (181,136 peaks). For each peak across all biological replicates independently, counts were extracted with BEDTools multicov, which was the input for differential peak analysis (Quinlan and Hall 2010). Signal tracks were generated from merged time points with deepTools (version 3.0.1) for visualization (Ramírez et al. 2016).

CUT&RUN processing peak calling

Adapters were trimmed off and low quality reads were filtered out with Trim Galore! (version 0.4.3, https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/)(“Babraham Bioinformatics - Trim Galore!” n.d.). BWA-MEM (version 0.7.17) was used to align reads to the hg38 genome, sorted with SAMtools (version 1.9), and filtered for duplicates with PicardTools (version 2.10.3, https://broadinstitute.github.io/picard/) (Danecek et al. 2021). SAMtools was again used to index all BAM files. For each timepoint, biological replicates were merged and indexed with SAMtools. We called peaks on the merged files using MACS2 with the following parameters: -f BAM -q 0.01 -g hs --nomodel --shift 0 --extsize 200 --keep-dup all -B --SPMR (version 2.1.1.20160309) (Zhang et al. 2008). A comprehensive peak list was generated for each antibody by merging peaks across all time points (X, Y, Z peaks for X, Y, Z datasets). For each biological replicate independently, counts were extracted with BEDTools multicov, which was the input for differential peak analysis(Quinlan and Hall 2010). Signal tracks were generated from merged time points with deepTools (version 3.0.1) for visualization (Ramírez et al. 2016).

Differential ATAC-seq and CUT&RUN Peak Analysis

DESeq2 was again used to identify differential peaks from ATAC-seq and all CUT&RUN data. The count matrix generated in the ATAC-seq and CUT&RUN processing was used as the input with the function DESeqDataSetFromMatrix. We used “~rep + time” as the design and a reduced design of “rep” to form an LRT. Apeglm was used to calculate shrunken $\log_2$(fold-changes) for each peak for each dataset (Zhu, Ibrahim, and Love 2019). Significant peaks had an adjusted p-value of < 0.05 and an absolute $\log_2$(fold-change) > 2. Peaks were clustered into up early, up mid, up late, down early, down mid, and down late by determining whether their max/min value was at 0, 6, or 72h.

Gene Ontology and KEGG Pathway Enrichment Analysis

We used the HOMER function findMotifs.pl on each of our 6 gene clusters to identify enriched Gene Ontology terms and KEGG pathways with default settings (Heinz et al. 2010). For GO Terms, the biological_process.txt file was used and for KEGG pathways, the kegg.txt file was used.

Motif Enrichment Analysis

We used the HOMER function findMotifsGenome.pl for all motif enrichment (Heinz et al. 2010). For motifs at the anchors of all loops, we intersected all ATAC peaks with all loop anchors. For motifs at the anchors of gained loops, we intersected all differential gained ATAC peaks with gained loop anchors. For motifs at the anchors of lost loops, we intersected all differential lost

ATAC peaks with lost loop anchors. For each motif enrichment, all ATAC peaks were used as the background. The default parameters were used with the following adjustments: -size given.

For the motif enrichment of proximal elements located at the promoters of our clustered genes, we used subsetByOverlaps to intersect promoters for each gene cluster with ATAC peaks. For the background, we used the nullranges Bioconductor package function matchRanges to obtain a set of ATAC peaks at the promoters of static genes matched for gene expression. For distal element enrichment, we used the linkOverlaps function within InteractionSet to find ATAC peaks that are connected to the promoters of the genes in each cluster via any chromatin loop (either static or differential). For the background, we again used matchRanges to obtain a set of ATAC peaks looped to the promoters of static genes that are matched for expression. For each of these enrichment analyses, we added the additional argument of -size 500.

For the enrichment of specific motifs, we used the HOMER function findMotifsGenome.pl with the additional -find parameter for each of the motifs shown in Figure 2 on the ATAC peaks distal to the promoters of the genes in the cluster where that motif was the most enriched with -size 500.

Genomic Intersections

The GenomicRanges and InteractionSet R packages were used to perform all genomic intersections (Lawrence et al. 2013). Loop bedpe files were converted into GInteractions objects and were intersected with the coordinates for ATAC, H3K27ac, JUN, CTCF, RAD21 peaks and genes with subsetByOverlaps. The unshrunken fold-changes as calculated by DESeq2 analysis were extracted from each of the peaks that overlapped a loop anchor.

Chromatin Looping Linear Model

The count matrices from previous analysis were again used to generate the data for the linear model. We used loop counts, peak counts from ATAC-seq and all CUT&RUN data, and transcripts per million (TPM) per 10-kb bin from RNA-seq. For all loops, each of these counts was extracted from both the anchors and each anchor individually. In the case of multiple peaks intersecting with loops, the sum of all peaks was recorded. We also recorded the maximum values from extracting counts from bigWig files instead. A final anchor measure was calculated by taking the product of signal (either sum or max) at both of the anchors for each loop. All interior measures were normalized to the length of the loop.

We added a pseudocount of 1000 to the entire dataframe (which is roughly 0.04 times the average count value) and then calculated the log(fold-change) between 72h and 0h. This "delta" matrix was then scaled, and DESeq2 $\log_2$(fold-changes) were used for looping. Our model consisted of all differential loops and twice as many static loops matched for distance and contact. Matched static loops were generated from the matchRanges (E. S. Davis et al. 2022) function within the nullranges Bioconductor package. 75% of the dataset was used for training and the remaining 25% was reserved for testing.

Each feature was tested against loop LFC with the base R function lm to determine R^2 values. The sign of correlation was determined with the cor function. LASSO regression was used to find a sparse model combining features, calling glmnet (Friedman, Hastie, and Tibshirani 2010) within the caret R package (Kuhn 2008). We trained the LASSO model on the training set, using anchor features only. We evaluated selected LASSO models on the test set using R^2. This was repeated again for all interior features, and with all anchor and interior features combined. The R^2 was calculated with the cor function. This was repeated for all interior features, and again repeated with all anchor and interior features combined.

We also used a random forest (Breiman 2001) (RF) regressor model to predict the chromatin loop fold change. 75% of the dataset was used for training and the remaining 25% was reserved for testing. The “randomForest” and “caret” R packages were used, and the “randomForest” and “train” functions were used respectively to build the models. Based on the training, the model predicts changes in chromatin looping with an R^2 value of 0.44 and a Root Mean Square Error (RMSE) of 0.38. Since RF works well for both classification and regression problems, a multi-class classifier was also built to predict three classes of loops, namely “gained”, “lost”, and “static” with an accuracy of 73.06% for the test data set. The percent included mean squared error was used as a metric to compare the relative importance of each of the features included.

Gene Expression Linear Model

We used the lm function within the stats R package to model how gene expression changes correlate with changes in changes in proximal and distal acetylation, and looping. We used the GenomicRanges function subsetByOverlaps and linkOverlaps to determine which differential genes had promoter H3K27ac and were looped to a distal H3K27ac peak (Lawrence et al. 2013), identifying 332 genes, and included 332 static genes matched for expression. We also identified the nearest enhancer with the nearest function of GenomicRanges. For proximal and distal H3K27ac, we extracted the counts and calculated $\log_2$(fold-change). For genes that had multiple enhancers, we took the sum of the counts at 0h and 72h and then calculated $\log_2$(fold-change). For the ABC score, we scaled all enhancer and loop counts to be between 1 and 100 to ensure that both factors were contributing equally to the interaction despite differences in sequencing depth. For each enhancer-promoter pair, we multiplied the normalized distal enhancer counts by the normalized loop strength counts, then calculated $\log_2$(fold-change). For genes with multiple enhancer-promoter pairs, we summed the multiplied score at 0h and 72h and then calculated $\log_2$(fold-change).

The first model only used changes in promoter acetylation to predict changes in gene expression. The following three models used promoter acetylation in addition to either nearest enhancer, distal enhancer, or the interaction between distal enhancers and loops. We trained on 60% of the data and tested on the remaining 40%. The predict function was used to predict on the testing dataset from the trained model and cor was used to calculate R^2 values. For R^2 and coefficient estimate calculations, we performed 1000 permutations of splitting the data into testing and training datasets.