SUPPLEMENTAL DATA

Data S1: Additional Hi-C Contact Matrices Relevant to Main Findings, Related to Figures 1, 2, 3, and 5

Data S1, I. Comparison of Hi-C Protocol Variants

(A) Contact matrices for the entirety of chromosome 17 and chromosome 10 at 500 kb resolution (top two rows) and a 10 megabase region (60-70 Mb) on chromosome 10 at 50 kb resolution (bottom row), generated via *in situ* Hi-C, tethered *in situ* Hi-C, agar *in situ* Hi-C, pellet Hi-C and dilution Hi-C. The chromatin interaction data is highly reproducible across all variants of the Hi-C protocol, despite the order of magnitude variation in sequencing depth in the libraries shown here.
(B) The results of a typical *in situ* Hi-C experiment (first column), side by side with two agar *in situ* contact maps (2nd and 3rd columns) and a pair of pellet/supernatant Hi-C experiments (4th and 5th columns). We show the intrachromosomal contact matrices for chromosome 9 at 500 kb resolution (1st row), for a 10 megabase region on chromosome 9 (104-114 Mb) at 50 kb resolution (2nd row), for chromosome 18 at 500 kb resolution (3rd row), and for a 9 Mb region on chromosome 18 (64-73 Mb) at 50 kb resolution (4th row). The *in situ* Hi-C, agar *in situ* Hi-C and pellet Hi-C maps strongly resemble one another; however, compartment interactions and domain structures are largely absent from the supernatant Hi-C contact map, confirming that *in situ* Hi-C ligation happens inside the nucleus. The uniform distribution of long-range contacts in the supernatant Hi-C contact map and the relative absence of reads close to the diagonal of the matrix suggest a higher rate of random ligation in the supernatant Hi-C map.

Data S1, II. Comparison of Different Hi-C Normalizations

(A) Contact matrices without normalization (left-most column), with vanilla coverage (VC) normalization (2nd column), with square root VC normalization, (3rd column), and with KR normalization (last column). First row: The entirety of chromosome 14 at 500 kb resolution; the long-range plaid pattern corresponding to A/B compartmentalization is clearly visible in all maps. Second row: an example domain in a 1.6 megabase region of chromosome 14 (35-36.6 Mb) at 5 kb resolution is visible in all maps. Third row: an example peak in a 1 megabase region of chromosome 14 (37.5-38.5 Mb) is visible in all maps.
(B) Interchromosomal contact maps for interactions between chromosome 20 and chromosome 14 (top row), and between chromosome 17 and chromosome 1 (bottom row) are shown at 100 kb resolution without any normalization (1st column), with KR normalization (2nd column), with genome-wide KR normalization (3rd column), and with interchromosomal KR normalization (4th column). In all contact matrices, fine-grained plaid patterning indicative of compartmentalization can be seen.

Data S1, III. Comparison of Contact Domains to TADs

(A-B) Two examples of large TAD calls (blue) from Dixon et al. annotated in IMR90 in (A) chromosome 3 and (B) chromosome 20, superimposed on our *in situ* IMR90 map (left column, 10 kb resolution), the Dixon et al. IMR90 map (middle, 25 kb resolution), and the Jin et al. IMR90 map (right, 25 kb resolution). Our domain annotations are shown in black. The TADs have numerous smaller domains in them, many of which can be seen in the previously published lower resolution maps. The last row of each panel shows long-range intrachromosomal patterns of the regions above in the same maps (25 kb, 50 kb, and 50 kb resolutions respectively). Numerous changes in long-range contact patterns occur within intervals annotated as individual TADs. At the top of each panel we show H3K36me3 marks and H3K27me3 marks in the regions, which tend to switch at the contact domain boundaries we annotate, but vary greatly inside TADs.
(C) Two examples of TAD calls from Dixon et al. on IMR90 in chromosome 17 and chromosome 5, superimposed on our *in situ* IMR90 map (left column, 10 kb resolution), the Dixon et al. IMR90 map (middle column, 25 kb resolution), and the Jin et al. IMR90 map (right column, 25 kb resolution). The Dixon et al. IMR90 TAD annotations are shown in blue, and our IMR90 contact domain annotations are shown in black. Numerous domains can be seen in the larger TADs; the TADs themselves are not visible.
(D) TAD calls from Dixon et al. on IMR90 in a region of chromosome 10, superimposed on our *in situ* IMR90 map (left column, 25 kb resolution), the Dixon et al. IMR90 map (middle column, 25 kb resolution), and the Jin et al. IMR90 map (right column, 25 kb resolution). The long-range interactions with off diagonal regions display extensive pattern switching inside TADs.

Data S1, IV. Subcompartment Interchromosomal Contact Patterns

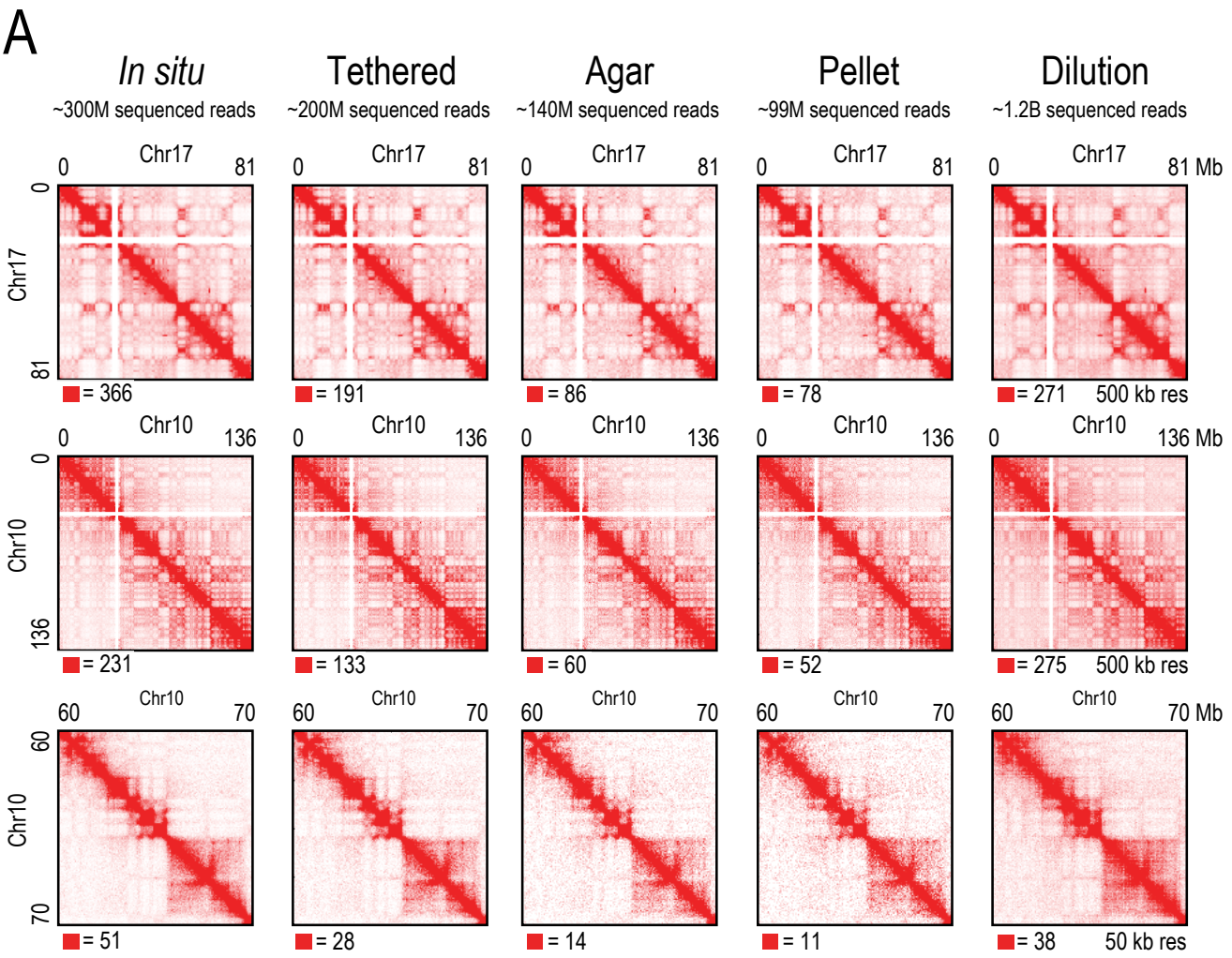
(A-D) Example subsections of interchromosomal contact matrices, where each panel shows the interactions of a region on one chromosome with regions on two other chromosomes: (A) Chr20 with Chr15 and Chr16; (B) Chr10 with Chr5 and Chr7; (C) Chr22 with Chr17 and Chr21; and (D) Chr9 with Chr2 and Chr3. Loci within the same subcompartment tend to have the same interchromosomal contact pattern. This is true even of loci residing on different chromosomes, but still in the same subcompartment.

Data S1, V. Peak Reproducibility

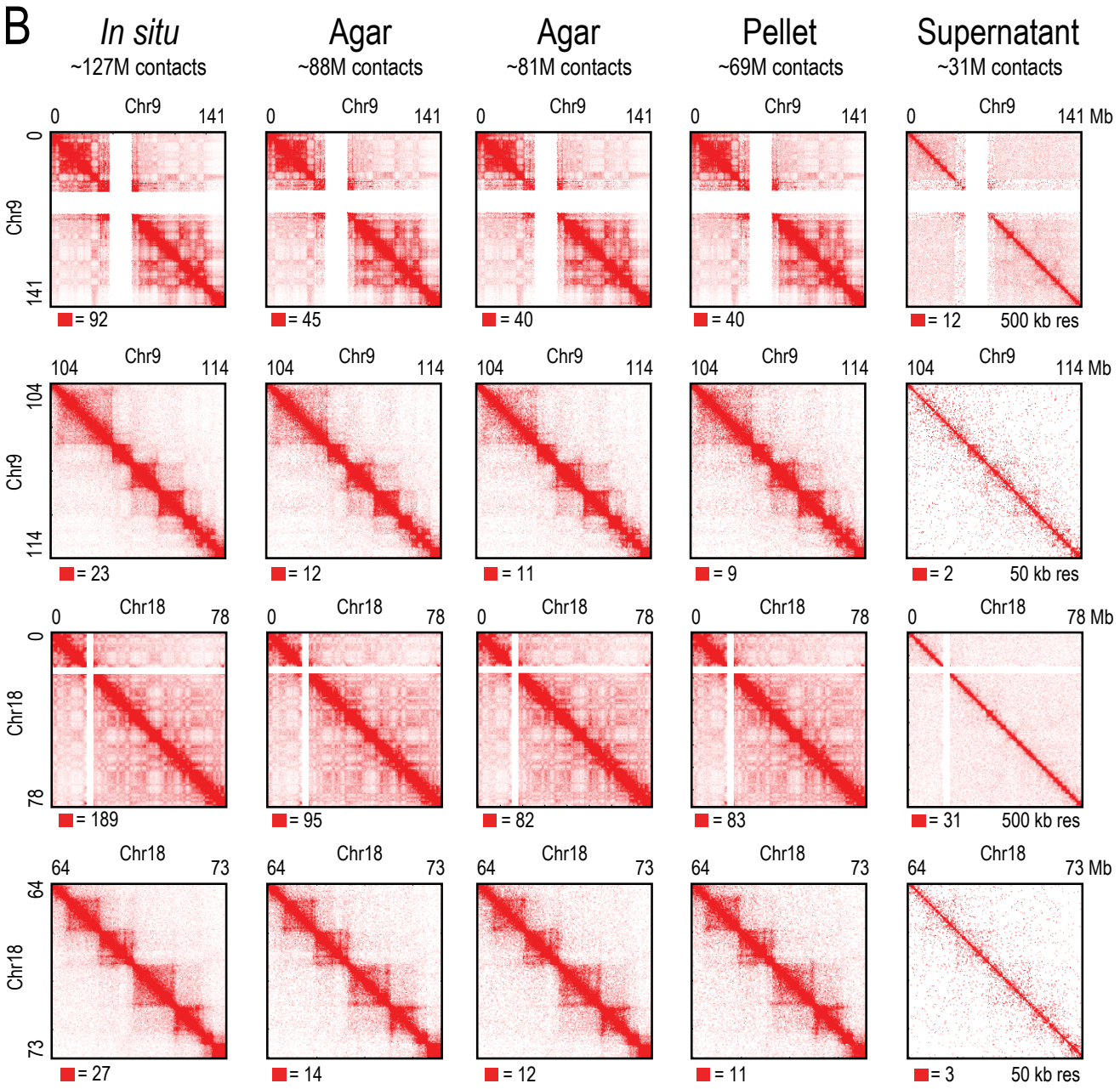
(A) We show 18 example regions with peaks annotated by HiCCUPS in both our primary and replicate GM12878 *in situ* Hi-C maps at 10 kb resolution. Peaks are marked with blue circles (centered at the peak pixel) in the lower-left half of each heatmap. The radius of all blue circles shown is 10 kb. The number of raw contacts at the peak is indicated, illustrating our deep coverage of most peaks in GM12878. Peaks are highly reproducible: for every pair, data and annotations based on our primary GM12878 map are shown on the left; completely independent data and completely independent annotations based on our replicate GM12878 experiment are shown on the right.

Data S1, VI. Cell-Type Specific Loops and Gene Activation

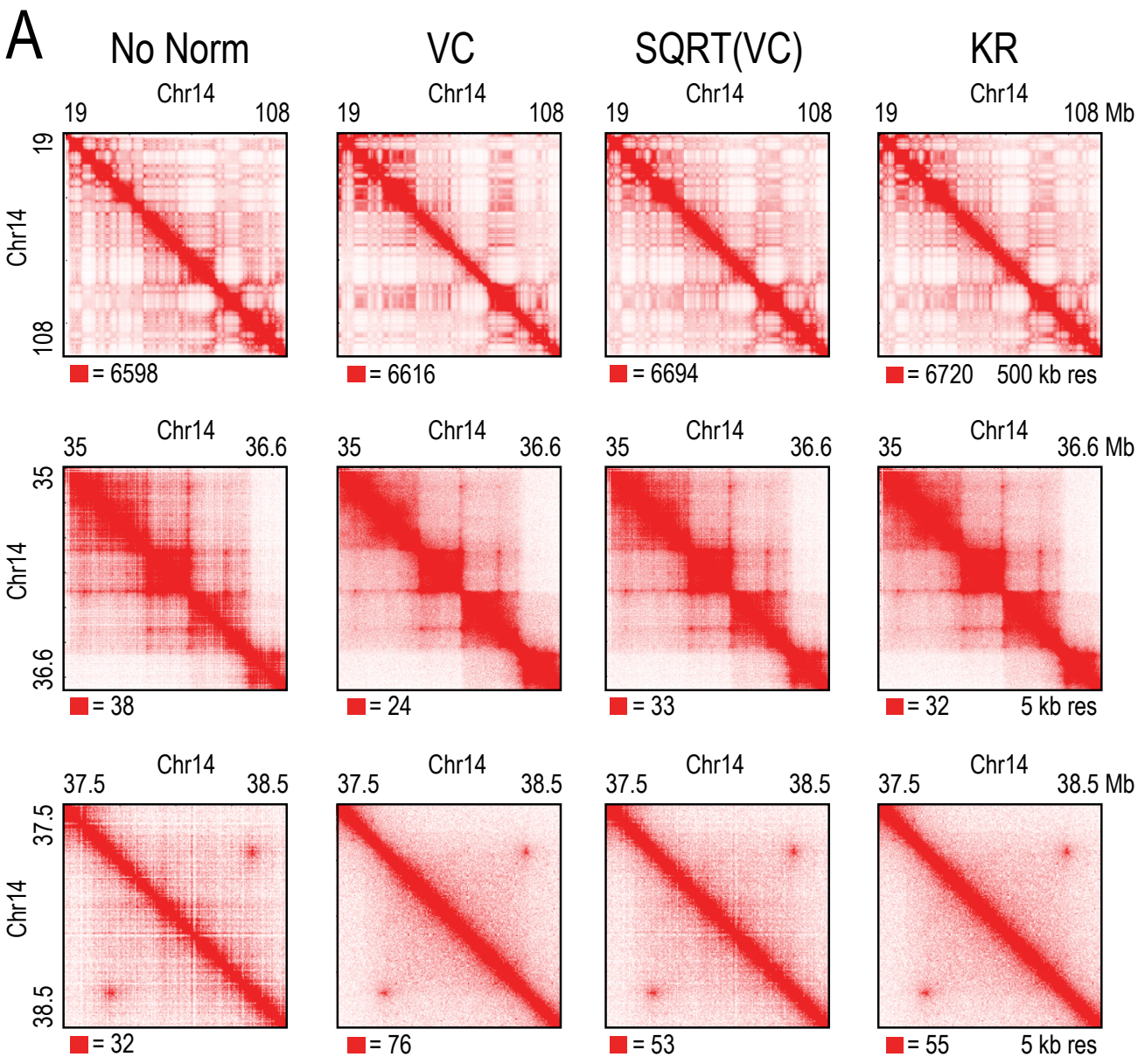
(A-H) Examples of cell-type specific loop domains that differ between GM12878 and other cell types. Appearance of a cell-type specific loop is often accompanied by the appearance of a cell-type specific CTCF peak, as well as the activation of a gene whose promoter lies at one of the peak loci. In every example, the activated gene is labeled; other genes are shown but not named. For every example, the contact matrix corresponding to the GM12878 data is shown on the left and the contact matrix corresponding to the second cell type is shown on the right.



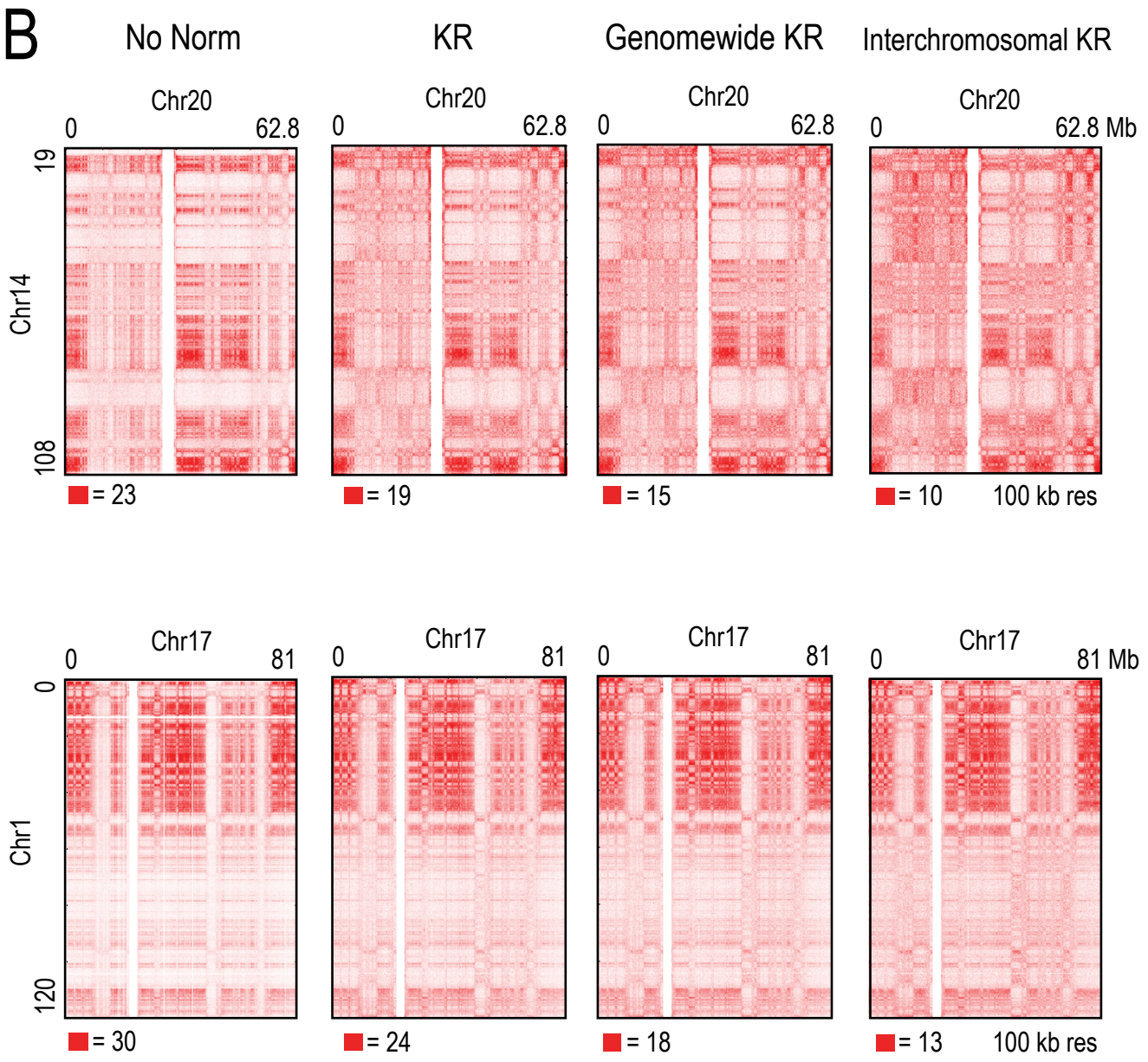
Data S1, I. Comparison of Hi-C Protocol Variants



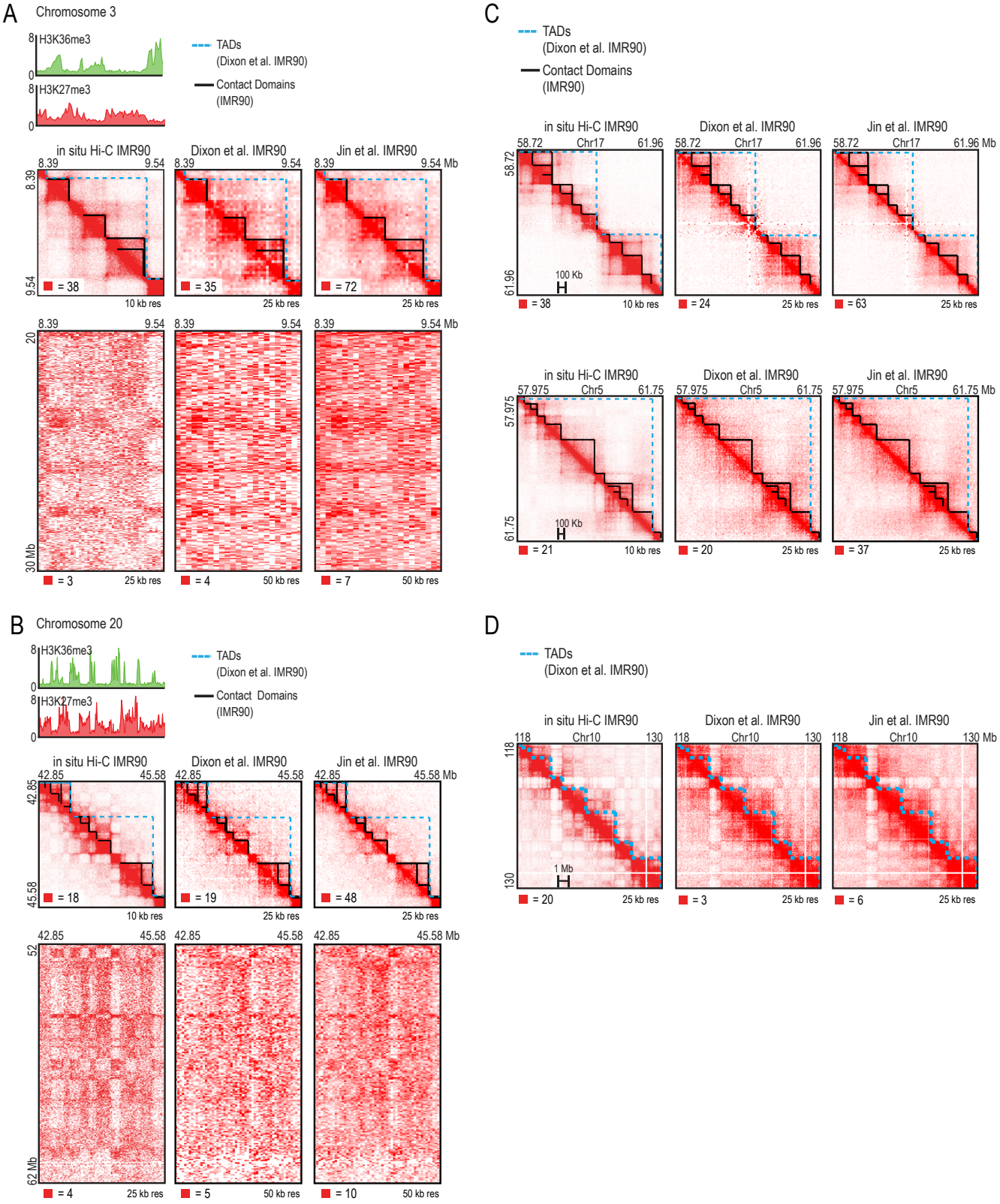
Data S1, I. Comparison of Hi-C Protocol Variants



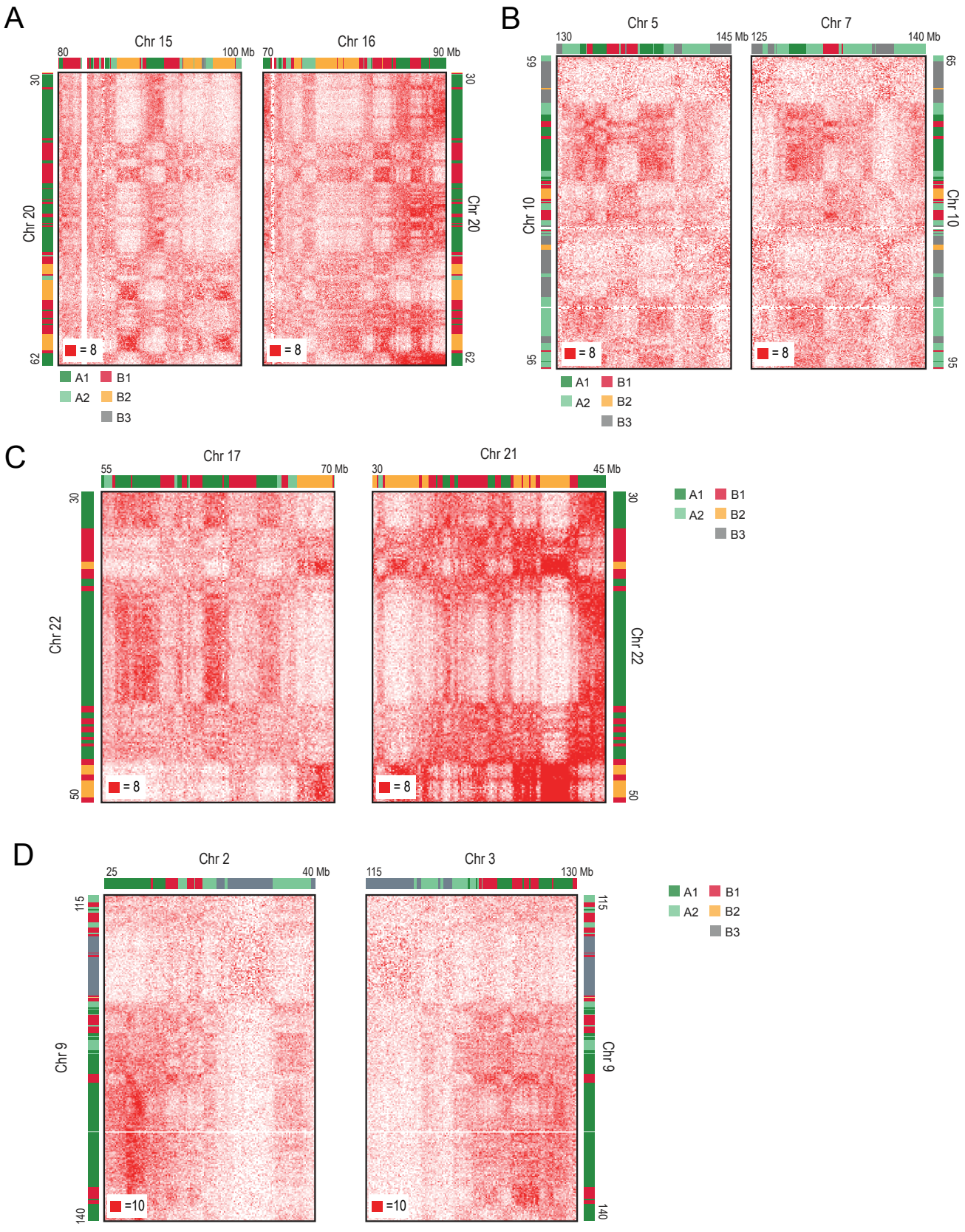
Data S1, II. Comparison of Different Hi-C Normalizations



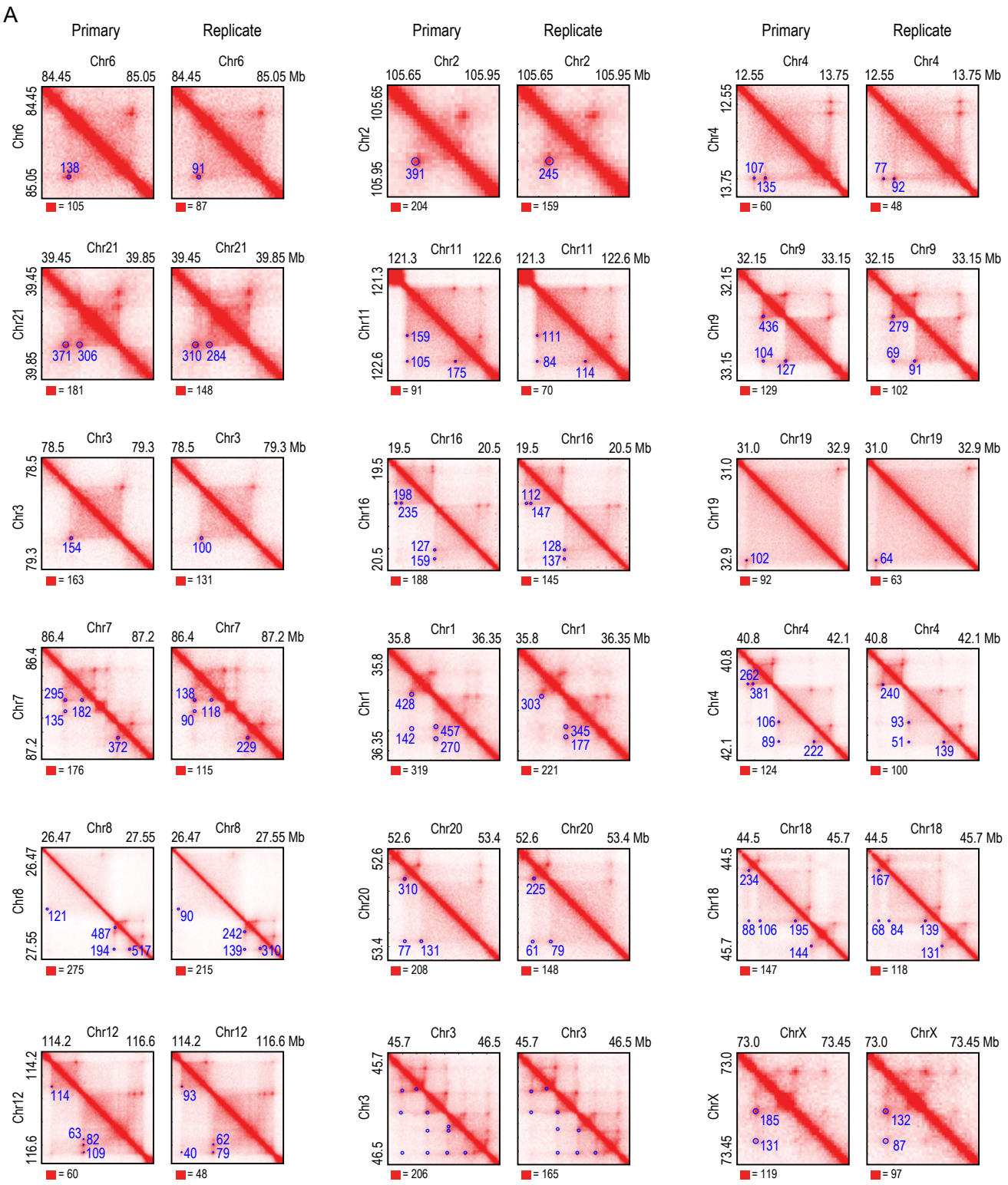
Data S1, II. Comparison of Different Hi-C Normalizations



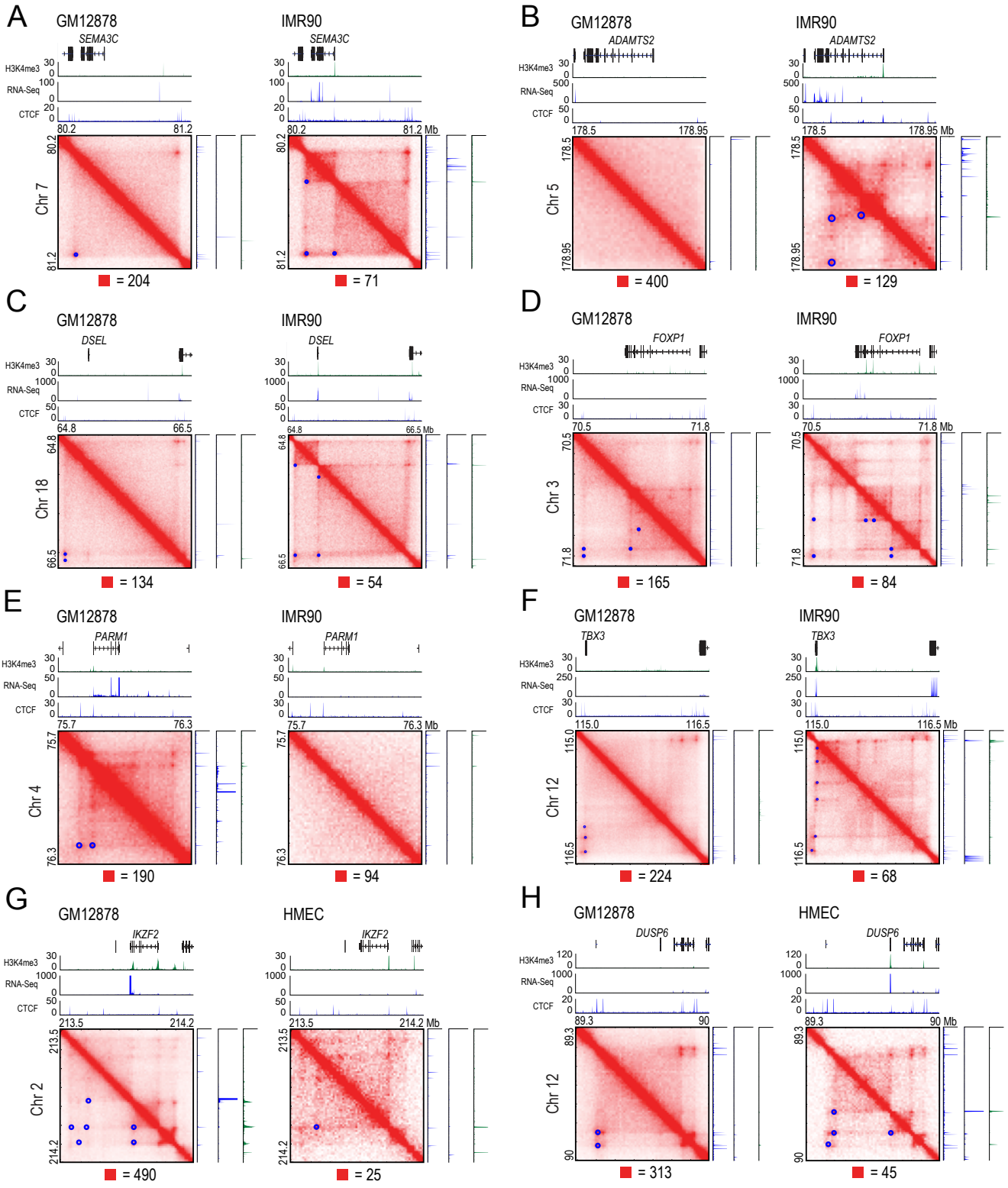
Data S1, III. Comparison of Contact Domains to TADs



Data S1, IV. Subcompartment Interchromosomal Contact Patterns



Data S1, V. Peak reproducibility



Data S1, VI. Cell-type Specific Loops and Gene Activation

Data S2: Validation of Peak Lists, Related to Figure 3

Data S2, I. APA of HiCCUPS Peak Lists

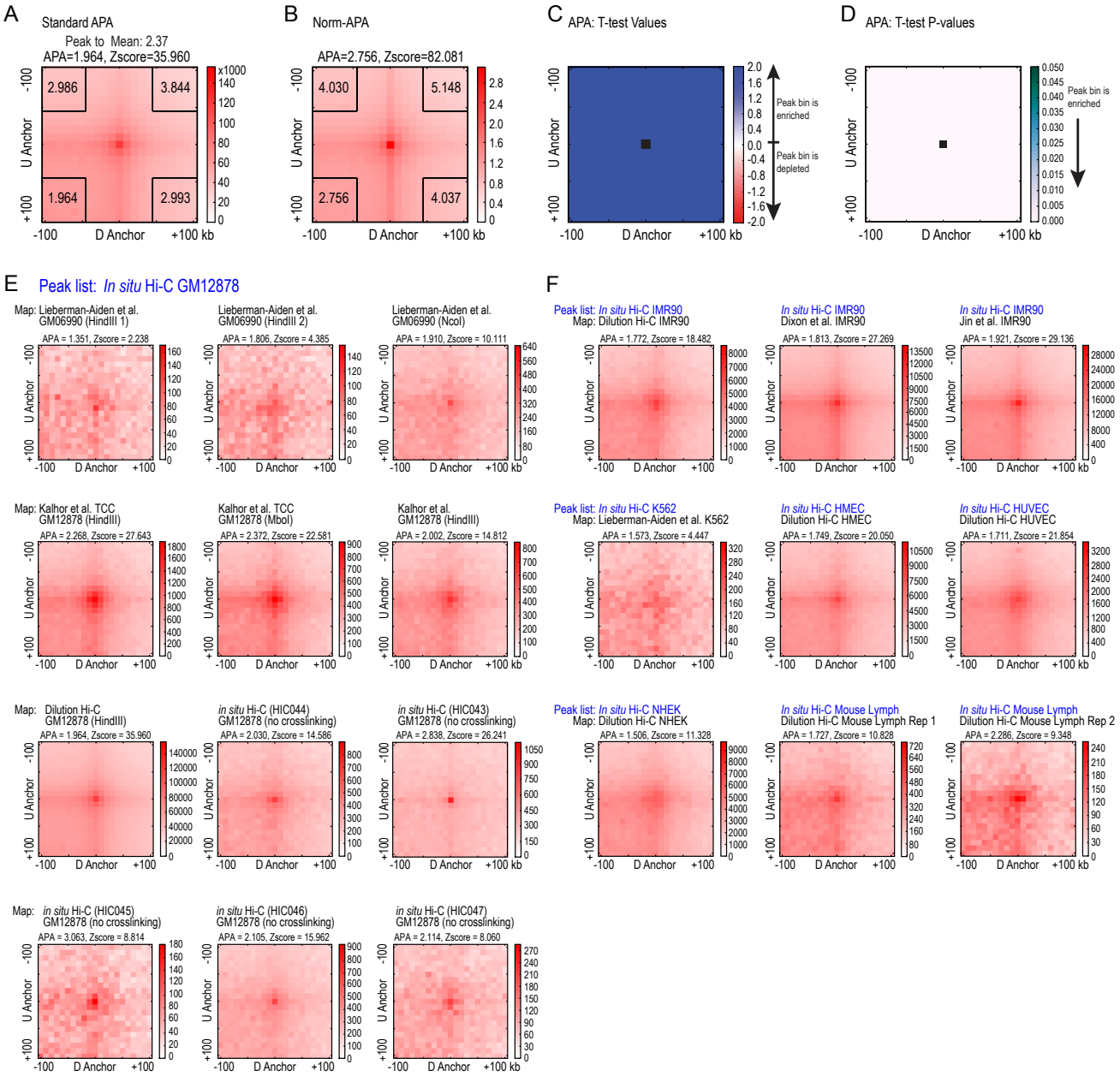
(A-D) Additional APA measures are shown. (A) Standard APA of our *in situ* GM12878 peak list examined on our GM12878 dilution map. The APA score is the ratio of the number of contacts in the central bin to the mean number of contacts in the lower-left corner (outlined by the black box). The ratio of the central bin to the other three corners is also shown inside each corner, though we do not use these scores in this paper. APA scores above 1 signify that the peak bin is enriched relative to the bins inside the corner. The ratio of the central bin to the mean of the remaining matrix, along with the z-score of the central bin, using the mean and standard deviation of the lower-left corner, are also shown. Z-scores above 1.64 indicate enrichment ($p<.05$). (B) Normalized APA of our *in situ* GM12878 peak list examined on our GM12878 dilution map. In Normalized APA, each submatrix is first normalized before being added to the aggregate matrix; normalization is performed by dividing each entry by the mean of the submatrix. The final Normalized APA matrix is the average of all of these submatrices. The maximum color scale in all APA plots is set to five times the mean value in the upper right corner of the matrix. (C-D) We show a more formal measure of aggregate peak enrichment of our *in situ* GM12878 peak list in our GM12878 dilution map. We use the one-sided paired Student t-test, measuring enrichment between the set of all values in the central peak bin (across all peaks) and the set of values in every other bin in the matrix. T-statistics are plotted in (C), with the corresponding p-values shown in (D). We find that our peak set is statistically enriched ($p<0.05$) relative to every other bin in the matrix. (No multiple hypothesis correction was performed on the p-values).
(E) We show APA plots for our *in situ* GM12878 peak list on every published human lymphoblastoid Hi-C contact map as well as additional lymphoblastoid maps generated in this study. In all cases the aggregate focal enrichment of our peaks can be seen.
(F) We show APA plots for our *in situ* IMR90, K562, HMEC, HUVEC, NHEK, and mouse lymphoblastoid peak lists against all available Hi-C maps in each cell line. In all cases focal enrichment is observed.

Data S2, II. APA of External Peak Lists

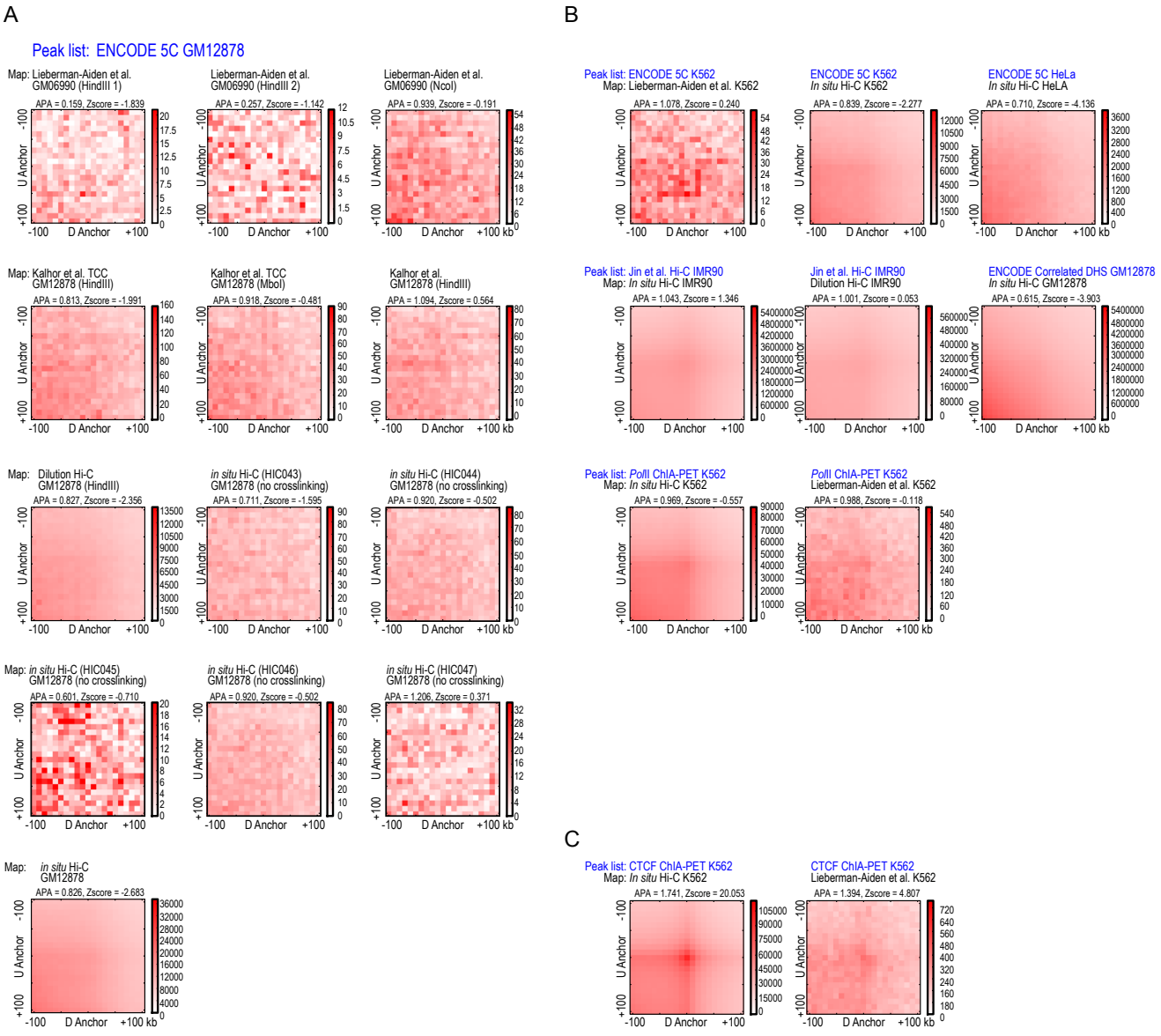
(A) We show APA plots for the ENCODE 5C GM12878 peak list on every published human lymphoblastoid Hi-C contact map as additional lymphoblastoid maps generated in this study. No aggregate focal enrichment can be seen.
(B) We show APA plots for the ENCODE 5C K562 peak list, the ENCODE 5C HeLa peak list, the Jin et al. Hi-C IMR90 peak list, the ENCODE correlated GM12878 DHS pairs list, and the *Pol*II ChIA-PET K562 peak list against all available Hi-C maps in each cell line. No aggregate focal enrichment can be seen.
(C) APA plots for the CTCF ChIA-PET K562 peak list in the two K562 Hi-C maps indicate that these peaks do exhibit focal enrichment.

Data S2, III. Comparison of Peak Lists with Hand Annotation

(A-G) We show annotations from seven peak lists, all in the same region (Chr 21:33746367-35442351). Putative peaks from each list are drawn in blue on our corresponding *in situ* contact map. Our hand annotation is drawn in black circles. Peak lists shown are (A) our *in situ* GM12878 peak list, (B) ENCODE's 5C GM12878 peak list, (C) ENCODE's 5C K562 peak list, (D) Jin et al.'s Hi-C IMR90 peak list, (E) Li et al.'s *Pol*II ChIA-PET K562 peak list, (F) Li et al.'s CTCF ChIA-PET K562 peak list, and (G) our Pseudo peak list, generated by using the genome-wide background (instead of local background) for peak calling. Many false positives are seen in all lists except our *in situ* GM12878 peak list and the CTCF ChIA-PET list.



Data S2, I. APA of HiCCUPS Peak Lists

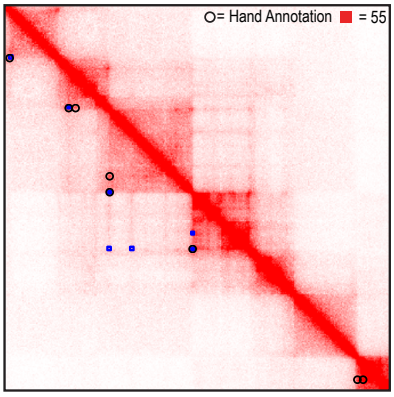


Data S2, II. APA of External Peak Lists

Region: Chr21:33746367-35442351

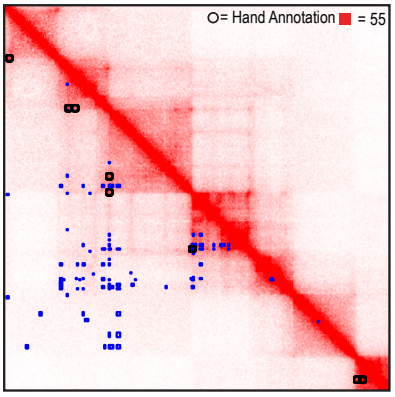
A

Peak list: *In situ* Hi-C GM12878
Map: *In situ* Hi-C GM12878



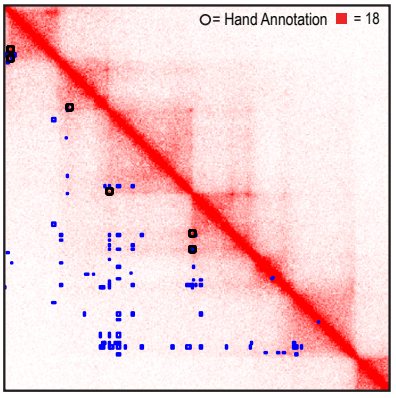
B

Peak list: 5C ENCODE GM12878
Map: *In situ* Hi-C GM12878



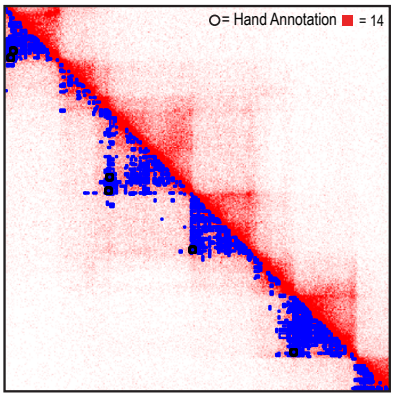
C

Peak list: 5C ENCODE K562
Map: *In situ* Hi-C K562



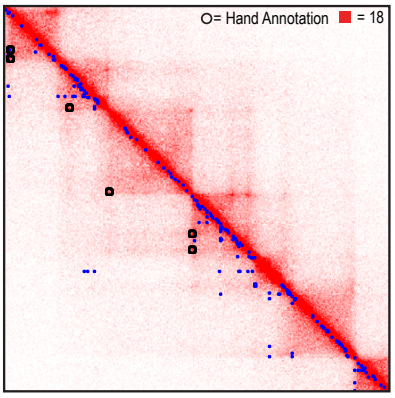
D

Peak list: Jin et al. IMR90
Map: *In situ* Hi-C IMR90



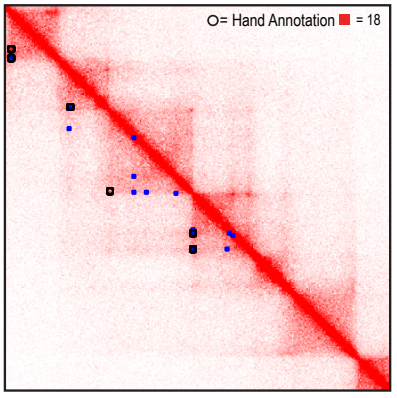
E

Peak list: *Pol*II ChIA-PET K562
Map: *In situ* Hi-C K562



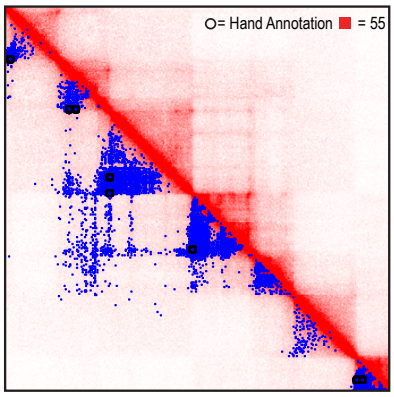
F

Peak list: CTCF ChIA-PET K562
Map: *In situ* Hi-C K562



G

Peak list: Pseudo Peaks GM12878 (GW background)
Map: *In situ* Hi-C GM12878



Data S2, III. Comparison of Peak Lists with Hand Annotation

EXTENDED EXPERIMENTAL PROCEDURES

Contents

3. *Arrowhead scoring*
4. *Dynamic programming for fast calculation*

b. Random Shuffling Control Algorithms

c. Properties of Domains

1. *Interaction probability drops at the boundaries of domains*
2. *Domains exhibit consistent patterns of histone modifications*
3. *Changes in patterns of long-range contact tend to occur at the boundaries of domains*
4. *Domains are conserved across many cell types*
5. *Changes in the histone modifications of a domain correspond to changes in long-range contact pattern*
6. *Relation of Topological Domains to Domains*

V. Subcompartments in Hi-C Data

a. Clustering Methodology

1. *Previous approaches towards clustering*
2. *Clustering algorithm*
3. *Creating an A/B pattern annotation*
4. *An additional cluster on Chromosome 19*

b. Properties of Subcompartment Clusters

1. *Clusters display distinct chromosomal and size distributions*
2. *Clusters display unique patterns of epigenetic modifications*
3. *Self-organization of a chromosome fragment in K562*

VI. Peaks in Hi-C Data

a. HiCCUPS: Identification of DNA Loops through Annotation of Focal Peaks

1. *Background and motivation*
2. *What does evidence for chromatin looping look like in a proximity ligation experiment?*
    i. *The definition of a chromatin loop implies that loops will manifest as local peaks in DNA-DNA proximity ligation data*
    ii. *Appearance of peaks in Hi-C data: local enrichment vs. background*
    iii. *Section outline*
3. *Summary of peak calling algorithms*
4. *Simple local peak calling on Hi-C data*
    i. *Local expected value calculation*
    ii. *Statistical significance and multiple hypothesis testing*
    iii. *Filtering of pixels landing in repetitive regions*
    iv. *Clustering of nearby enriched pixels*
5. *HiCCUPS (Hi-C Computational Unbiased Peak Search)*
    i. *Additional local neighborhoods*
    ii. *Multiple hypothesis testing*
    iii. *Additional filtering of pixels landing in repetitive regions*
    iv. *Clustering of nearby enriched pixels*
    v. *Additional filtering of peak pixels based on local enrichment thresholds*
    vi. *Additional filtering of "singlet clusters."*
    vii. *Combining peak annotations at different resolutions*
6. *Peak calling using a global expectation*
7. *Computational considerations*

b. HiCCUPS Validation

1. *Peaks are biologically reproducible and reliably annotated between replicates*
2. *Sensitivity and specificity of HiCCUPS*
3. *Simplified local peak caller recapitulates results of HiCCUPS*
4. *HiCCUPS can also be performed at fragment resolutions*
5. *Saturation analysis of HiCCUPS peak calls*
6. *Comparison with external loop annotations*

c. 3D DNA FISH

1. *3D DNA FISH confirms that peak loci are more likely to be spatially proximate*

d. Aggregate Peak Analysis (APA)

1. *Standard APA analysis*

I. Experimental Methods

I.a. Hi-C protocols

In this paper, we report the results of 201 Hi-C experiments. To produce most of the libraries reported in this paper, we employed *in situ* Hi-C. Several additional libraries were generated using variants of *in situ* Hi-C. Still other libraries were generated using the original "dilution" Hi-C protocol (Lieberman-Aiden et al., 2009). The contact maps produced by Hi-C are extremely robust to changes in the protocol, except as described below and in the main text (see Data S1.I.A).

I.a.1. In situ Hi-C protocol

Crosslinking

1) Grow two to five million mammalian cells under recommended culture conditions to about 80% confluence. Pellet suspension cells or detached adherent cells by centrifugation at 300xG for 5 minutes.
2) Resuspend cells in fresh medium at a concentration of $1x10^6$ cells per 1ml media. In a fume hood, add freshly made formaldehyde solution to a final concentration of 1%, v/v. Incubate at room temperature for 10 minutes with mixing.
3) Add 2.5M glycine solution to a final concentration of 0.2M to quench the reaction. Incubate at room temperature for 5 minutes on rocker.
4) Centrifuge for 5 minutes at 300xG at 4°C. Discard supernatant into an appropriate collection container.
5) Resuspend cells in 1ml of cold 1X PBS and spin for 5 minutes at 300xG at 4°C. Discard supernatant and flash-freeze cell pellets in liquid nitrogen or dry ice/ ethanol.
6) Either proceed to the rest of the protocol or store cell pellets at -80°C.

Lysis and Restriction Digest

7) Combine 250µl of ice-cold Hi-C lysis buffer (10mM Tris-HCl pH8.0, 10mM NaCl, 0.2% Igepal CA630) with 50µl of protease inhibitors (Sigma, P8340). Add to one crosslinked pellet of cells.
8) Incubate cell suspension on ice for >15 minutes. Centrifuge at 2500xG for 5 minutes. Discard the supernatant.
9) Wash pelleted nuclei once with 500µl of ice-cold Hi-C lysis buffer.
10) Gently resuspend pellet in 50µl of 0.5% sodium dodecyl sulfate (SDS) and incubate at 62°C for 5-10 minutes.
11) After heating is over, add 145µl of water and 25µl of 10% Triton X-100 (Sigma, 93443) to quench the SDS. Mix well, avoiding excessive foaming. Incubate at 37°C for 15 minutes.
12) Add 25µl of 10X NEBuffer2 and 100U of MboI restriction enzyme (NEB, R0147) and digest chromatin overnight or for at least 2 hours at 37°C with rotation.

Marking of DNA Ends, Proximity Ligation, and Crosslink Reversal

13) Incubate at 62°C for 20 minutes to inactivate MboI, then cool to room temperature.
14) To fill in the restriction fragment overhangs and mark the DNA ends with biotin, add 50µl of fill-in master mix:
    37.5µl of 0.4mM biotin-14-dATP (Life Technologies, 19524-016)
    1.5µl of 10mM dCTP
    1.5µl of 10mM dGTP
    1.5µl of 10mM dTTP
    8µl of 5U/µl DNA Polymerase I, Large (Klenow) Fragment (NEB, M0210)
15) Mix by pipetting and incubate at 37°C for 45 minutes-1.5 hours with rotation.
16) Add 900µl of ligation master mix:
    663µl of water
    120µl of 10X NEB T4 DNA ligase buffer (NEB, B0202)
    100µl of 10% Triton X-100
    12µl of 10mg/ml Bovine Serum Albumin (100X BSA)
    5µl of 400 U/ µl T4 DNA Ligase (NEB, M0202)
17) Mix by inverting and incubate at room temperature for 4 hours with slow rotation.
18) Degrade protein by adding 50µl of 20mg/ml proteinase K (NEB, P8102) and 120µl of 10% SDS and incubate at 55°C for 30 minutes. (Note that nuclei can be pelleted after ligation and then resuspended, both to remove random ligation products that may have occurred in solution and to reduce the overall volume for ease of handling.)
19) Add 130µl of 5M sodium chloride and incubate at 68°C overnight or for at least 1.5 hours.

DNA Shearing and Size Selection

20) Cool tubes at room temperature.
21) Split into two 750µl aliquots in 2ml tubes and add 1.6X volumes of pure ethanol and 0.1X volumes of 3M sodium acetate, pH 5.2, to each tube. Mix by inverting and incubate at -80°C for 15 minutes.
22) Centrifuge at max speed, 2°C for 15 minutes. Keep the tubes on ice after spinning and carefully remove the supernatant by pipetting.
23) Resuspend, combining the two aliquots, in 800µl of 70% ethanol. Centrifuge at max speed for 5 minutes.
24) Remove all supernatant and wash the pellet once more with 800µl of 70% ethanol.
25) Dissolve the pellet in 130µl of 1X Tris buffer (10 mM Tris-HCl, pH 8) and incubate at 37°C for 15 minutes to fully dissolve the DNA.
26) To make the biotinylated DNA suitable for high-throughput sequencing using Illumina sequencers, shear to a size of 300-500bp using the following parameters:
    Instrument: Covaris LE220 (Covaris, Woburn, MA)
    Volume of Library: 130µl in a Covaris microTUBE
    Fill Level: 10
    Duty Cycle: 15
    PIP: 500
    Cycles/Burst: 200
    Time: 58 seconds
27) Transfer sheared DNA to a fresh 1.5ml tube. Wash the Covaris vial with 70µl of water and add to the sample, bringing the total reaction volume to 200µl. Run a 1:5 dilution of DNA on a 2% agarose gel to verify successful shearing. For libraries containing fewer than $2x10^6$ cells, the size selection using AMPure XP beads described in the next steps could be performed on final amplicons rather than before biotin pull-down.
28) Warm a bottle of AMPure XP beads (Beckman Coulter, A63881) to room temperature. To increase yield, AMPure XP beads can be concentrated by removing some of the clear solution before the beads are mixed for use in the next steps.
29) Add exactly 110µl (0.55X volumes) of beads to the reaction. Mix well by pipetting and incubate at room temperature for 5 minutes.
30) Separate on a magnet. Transfer the clear solution to a fresh tube, avoiding any beads. The supernatant will contain fragments shorter than 500bp.
31) Add exactly 30µl of fresh AMPure XP beads to the solution. Mix by pipetting and incubate at room temperature for 5 minutes.
32) Separate on a magnet and keep the beads. Fragments in the range of 300-500bp will be retained on the beads. Discard the supernatant containing degraded RNA and short DNA fragments.
33) Keeping the beads on the magnet, wash twice with 700µl of 70% ethanol without mixing.
34) Leave the beads on the magnet for 5 minutes to allow remaining ethanol to evaporate.
35) To elute DNA, add 300µl of 1X Tris buffer, gently mix by pipetting, incubate at room temperature for 5 minutes, separate on a magnet, and transfer the solution to a fresh 1.5ml tube.
36) Quantify DNA by Qubit dsDNA High Sensitivity Assay (Life Technologies, Q32854) and run undiluted DNA on a 2% agarose gel to verify successful size selection.

Biotin Pull-Down and Preparation for Illumina Sequencing

Perform all the following steps in low-bind tubes.

37) Prepare for biotin pull-down by washing 150µl of 10mg/ml Dynabeads MyOne Streptavidin T1 beads (Life technologies, 65602) with 400µl of 1X Tween Washing Buffer (1X TWB: 5mM Tris-HCl (pH 7.5); 0.5mM EDTA; 1M NaCl; 0.05% Tween 20). Separate on a magnet and discard the solution.
38) Resuspend the beads in 300µl of 2X Binding Buffer (2X BB: 10mM Tris-HCl (pH 7.5); 1mM EDTA; 2M NaCl) and add to the reaction. Incubate at room temperature for 15 minutes with rotation to bind biotinylated DNA to the streptavidin beads.
39) Separate on a magnet and discard the solution.
40) Wash the beads by adding 600µl of 1X TWB and transferring the mixture to a new tube. Heat the tubes on a Thermomixer at 55°C for 2 min with mixing. Reclaim the beads using a magnet. Discard supernatant.
41) Repeat wash.
42) Resuspend beads in 100ul 1X NEB T4 DNA ligase buffer (NEB, B0202) and transfer to a new tube. Reclaim beads and discard the buffer.
43) To repair ends of sheared DNA and remove biotin from unligated ends, resuspend beads in 100µl of master mix:
    88µl of 1X NEB T4 DNA ligase buffer with 10mM ATP

2µl of 25mM dNTP mix
5µl of 10U/µl NEB T4 PNK (NEB, M0201)
4µl of 3U/µl NEB T4 DNA polymerase I (NEB, M0203)
1µl of 5U/µl NEB DNA polymerase I, Large (Klenow) Fragment (NEB, M0210)

44) Incubate at room temperature for 30 minutes. Separate on a magnet and discard the solution.
45) Wash the beads by adding 600µl of 1X TWB and transferring the mixture to a new tube. Heat the tubes on a Thermomixer at 55°C for 2 min with mixing. Reclaim the beads using a magnet. Discard supernatant.
46) Repeat wash.
47) Resuspend beads in 100µl 1X NEBuffer 2 and transfer to a new tube. Reclaim beads and discard the buffer.
48) Resuspend beads in 100µl of dATP attachment master mix:
    90µl of 1X NEBuffer 2
    5µl of 10mM dATP
    5µl of 5U/µl NEB Klenow exo minus (NEB, M0212)
49) Incubate at 37°C for 30 minutes. Separate on a magnet and discard the solution.
50) Wash the beads by adding 600µl of 1X TWB and transferring the mixture to a new tube. Heat the tubes on a Thermomixer at 55°C for 2 min with mixing. Reclaim the beads using a magnet. Discard supernatant.
51) Repeat wash.
52) Resuspend beads in 100µl 1X Quick ligation reaction buffer (NEB, B6058) and transfer to a new tube. Reclaim beads and discard the buffer.
53) Resuspend in 50µl of 1X NEB Quick ligation reaction buffer.
54) Add 2µl of NEB DNA Quick ligase (NEB, M2200). Add 3µl of an Illumina indexed adapter. Record the sample-index combination. Mix thoroughly.
55) Incubate at room temperature for 15 minutes. Separate on a magnet and discard the solution.
56) Wash the beads by adding 600µl of 1X TWB and transferring the mixture to a new tube. Heat the tubes on a Thermomixer at 55°C for 2 min with mixing. Reclaim the beads using a magnet. Remove supernatant.
57) Repeat wash.
58) Resuspend beads in 100µl 1X Tris buffer and transfer to a new tube. Reclaim beads and discard the buffer.
59) Resuspend in 50µl of 1X Tris buffer.

Final Amplification and Purification

60) Amplify the Hi-C library directly off of the T1 beads with 4-12 cycles of PCR, using Illumina primers and protocol (Illumina, 2007). (Note that recent lots of some streptavidin beads may interfere with PCR; to avoid this, one can remove the DNA from the streptavidin beads by heating at 98C for 10 minutes after step 59 and then removing the beads with a magnet.)
61) After amplification is complete, bring the total library volume to 250µl.
62) Separate on a magnet. Transfer the solution to a fresh tube and discard the beads.
63) Warm a bottle of AMPure XP beads to room temperature. Gently shake to resuspend the magnetic beads. Add 175µl of beads to the PCR reaction (0.7X volumes). Mix by pipetting and incubate at room temperature for 5 minutes.
64) Separate on a magnet and remove the clear solution.
65) Keeping the beads on the magnet, wash once with 700µl of 70% ethanol without mixing.
66) Remove ethanol completely. To remove traces of short products, resuspend in 100µl of 1X Tris buffer and add another 70µl of AMPure XP beads. Mix by pipetting and incubate at room temperature for 5 minutes.
67) Separate on a magnet and remove the clear solution.
68) Keeping the beads on the magnet, wash twice with 700µl of 70% ethanol without mixing.
69) Leave the beads on the magnet for 5 minutes to allow the remaining ethanol to evaporate.
70) Add 25-50µl of 1X Tris buffer to elute DNA. Mix by pipetting, incubate at room temperature for 5 minutes, separate on a magnet, and transfer the solution to a fresh labeled tube. The result is a final *in situ* Hi-C library ready to be quantified and sequenced using an Illumina sequencing platform.

I.a.2. In situ Hi-C libraries can be constructed in three days: The above protocol takes either 3 or 4 days, depending on whether shorter incubation times are used for the restriction step (2 hours; see step 12), the fill-in step (45 minutes; see step 15), and the crosslink reversal step (1.5 hours; see step 19). Most of the *in situ* Hi-C libraries reported in this paper were performed using the 4 day protocol, but in our experience, use of the 3 day protocol has no effect on library quality.

I.a.3. "Tethered" in situ Hi-C: Kalhor et al. (2012) introduced a variant of Hi-C that they called "Tethered Conformation Capture" (TCC), in which proteins are biotinylated prior to restriction so that crosslinked chromatin can be tethered to

streptavidin beads. Fill-in of restricted fragment ends and blunt-end ligation is then performed on beads. They reasoned that this would limit interactions between non-crosslinked fragments.

While tethering might have a significant impact on chromatin in dilution, we reasoned that tethering proteins to beads prior to ligation should have no effect on our *in situ* protocol, as chromatin is already constrained by the intact nucleus. We adapted the TCC protocol in order to develop a tethered variant of our *in situ* protocol and confirm that it does not have an impact on library quality.

After step 10 of the *Lysis and Restriction Digest* section above, we mixed the suspension with 20µl of 25 mM EZlink Iodoacetyl-PEG2-Biotin (IPB) (Pierce Protein Biology Products, 21334) and rocked at room temperature for 60 minutes. We then mixed the sample with 260µl of NEBuffer2 and 45µl of 10% Triton X-100, incubated on ice for 10 minutes and then at 37°C for 10 minutes. Next, we added 20µl of NEBuffer2, 1µl of 1M DTT, 86µl of water, and 100U of MboI and incubated at 37°C overnight to digest the chromatin.

The next day, we passed the sample through a 2mL Zeba spin desalting column (Thermo Scientific, 89889) in order to remove any unreacted IPB.

The steps between attachment to MyOne Streptavidin T1 beads (Invitrogen) and detachment from the beads were performed as in TCC (Kalhor et al., 2012), with the exception that the dNTPs used in the fill-in step were the same as the ones that we use in our *in situ* Hi-C protocol, and the ligation was either performed in 5mL (as in TCC) or in 1mL (with all volumes scaled down). In both cases, 5µl of 400 U/ µl T4 DNA Ligase (NEB, M0202) was added during ligation. After detachment of the library from the T1 beads, the library was completed using the standard *in situ* Hi-C protocol beginning with step 19.

I.a.4. In situ Hi-C in agar plugs: After lysis (as in the usual *in situ* Hi-C protocol, step 11), nuclei were resuspended in 100*µ*l 2X NEBuffer2 and mixed with 100*µ*l molten 2% NuSieve agarose (Lonza, 5009) and allowed to solidify into an agarose plug. The nuclei embedded in agar were restricted overnight in 500*µ*l 1X NEBuffer2 with 100U of MboI at 37°C.

After restriction, the buffer was discarded and the agar plug was washed twice with 1ml of 1X NEB T4 DNA ligase buffer for 30min at 37°C. The buffer was discarded and the agar plug was submerged in 0.5ml fill-in reaction mix:

398µl of water
50µl of 10X NEB T4 DNA ligase buffer
37.5µl of 0.4mM biotin-14-dATP
1.5µl of 10mM dCTP
1.5µl of 10mM dGTP
1.5µl of 10mM dTTP
10µl of 5U/µl DNA Polymerase I, Large (Klenow) Fragment

The library was incubated for 1.5 hours at room temperature. After incubation, 2000U of T4 DNA Ligase were added to the reaction and the library was ligated at room temperature for 4 hours.

After ligation, the buffer was discarded and the agar plug was washed twice with 1ml of 1X NEB β-agarase I buffer (NEB, B0392) for 30min at 37°C. The buffer was removed and the agarose was melted by incubation at 68°C for 10 minutes. Liquid agarose was equilibrated at 42°C for 15 minutes. The agarose was digested with 4U of β-Agarase I (NEB, M0392) at 42°C for 1 hour. Next, we reversed the crosslinks. All subsequent steps were performed following the standard *in situ* Hi-C protocol beginning at step 18.

I.a.5. Pellet and Supernatant Hi-C: In situ Hi-C was performed in the usual fashion, but with an additional centrifugation step added after restriction (step 12) and prior to fill-in. We pelleted nuclei after restriction, transferred the supernatant to another tube, resuspended the nuclei in fresh buffer and proceeded with the rest of the protocol simultaneously on both the nuclei and the supernatant.

I.a.6 In situ Hi-C without crosslinking: In situ Hi-C can be performed without the use of crosslinking. In this study, we constructed five Hi-C libraries without the use of formaldehyde, or any, crosslinker. One library was constructed using the standard *in situ* Hi-C protocol (without crosslinking) with extremely gentle handling to avoid disrupting the nucleus or genome structure within the nucleus. The other four no-crosslinking libraries were constructed after embedding the uncrosslinked nuclei in agar plugs (section *I.a.4*) in order to maintain nuclear shape and structure. While the data generated by the no-crosslinking protocols is noisier, the main features we report in this study are all visible in our no-crosslinking maps.

I.a.7. Dilution Hi-C: Dilution Hi-C was performed as in Lieberman-Aiden et al. (2009)

I.a.8. Other experimental variations: In addition to the nine main variants of Hi-C outlined above (*in situ*, 3-day *in situ*, tethered *in situ*, agar *in situ*, pellet, supernatant, *in situ* without crosslinking, agar *in situ* without crosslinking, and dilution Hi-C), we performed over one hundred additional experiments in which we modified cell crosslinking time, choice of restriction enzyme, choice of biotinylated nucleotide, and ligation volume. The experimental parameters for each library are listed in Table S1.

I.a.9. Replicate Experiments: In this paper, we refer to both "technical replicates" and "biological replicates." Two Hi-C libraries are "technical replicates" if the cells were crosslinked together and identical Hi-C protocols were applied to two aliquots. Two samples are "biological replicates" if the cells were not crosslinked together; more specifically, the underlying cell populations were different due to additional passaging.

In Table S1, we label each of the 201 Hi-C libraries we report here with a unique identifier of the form HIC* where the * represents numerical values. The number after 'HIC' in the identifier is unique for every Hi-C library. We also provide second number (in the biological replicate column in Table S1) that indicates the biological replicate number; for a given cell type, libraries with the same biological replicate number were constructed from the same batch of crosslinked cells. Two libraries that are constructed from the same cell type and using the same protocol variant are technical replicates if they have the same biological replicate number or biological replicates if they have different biological replicate numbers.

I.a.10. Sequencing: Paired-end sequencing was performed using the Illumina MiSeq, HiSeq 2000, and HiSeq 2500 platforms.

I.a.11. No-ligation control: In order to assess the biases introduced during the digestion step of our protocol, we routinely perform "no-ligation" controls. In a no-ligation control, crosslinked pellets are lysed and digested as in the standard *in situ* Hi-C protocol (steps 1-12), but the post-digestion fill-in and ligation steps are omitted (steps 13-19) and the library is prepped for sequencing. For this study, we sequenced two such no-ligation controls, one digested with HindIII (113M paired-end reads) and one digested with MboI (114M paired-end reads).

I.b. ChIP-Seq

ChIP-Seq was performed following the protocol outlined by the ENCODE consortium (Landt et al., 2012). We performed ChIP-Seq for H3K9me3 using three different antibodies: one replicate using the same antibody as ENCODE (Abcam ab8898), three replicates using an antibody from Millipore (Millipore 17-625) and three replicates using an antibody from Diagenode (Diagenode C15410056 pAb-056-50). We also performed three replicate ChIP-Seq experiments for H3K36me3 using an antibody from Abcam (ab9050). The ten libraries were each sequenced to a depth of 8-10M reads along with two input controls and an IgG control. Data was processed using MACS (Zhang et al., 2007).

I.c. 3D DNA FISH

3D DNA fluorescence *in situ* hybridization (FISH) was performed on GM12878 human lymphoblastoid cells essentially as described in (Beliveau et al., 2012), with the following minor changes.

For the experiments testing the HiCCUPS annotated GM12878 loops we called in our *in situ* map, a pool of 32bp sequences tiling nine 30 kb target loci at a density of 9-15 probes per kilobase was designed with OligoArray (Table S5, (Beliveau et al., 2012)). For each set of 32bp sequences corresponding to one locus, a pair of 21bp random primers was generated to flank the 32bp of genomic sequence. The forward and reverse primers were generated to include Nb.BsmI and Nb.BsrDI (New England Biolabs R0706, R0648) nicking sites, respectively, to allow for probe generation using the OligoPaints protocol. The resulting pool of 74bp oligomers was ordered as dsDNA from CustomArray, Inc. (Bothell, WA). Forward primers were synthesized with a 5' conjugated fluorophore (Alexa Fluor 488, Alexa Fluor 546/ATTO 565, or Alexa Fluor 647) and purified by HPLC (Integrated DNA Technologies); reverse primers contained no dyes and were purified by standard desalting. Oligopaints were amplified via PCR without emulsion, and fluorophores were kept in darkness as much as possible to minimize photobleaching. Cells in serum-free growth media were placed on poly-L-lysine coated slides (Electron Microscopy Sciences) at a concentration of 1-2x10^6 cells/ml and allowed to adhere for 0.5-1.5 hours at 37°C, 5% CO_2 prior to fixation for 7-10 minutes in 4% (v/v) paraformaldehyde in 1X PBS. The hybridization cocktail (15-20

μl per slide) was composed of 2X SSCT, 30-50% formamide, 10% (w/v) dextran sulfate, 10 μg RNase A, and 10-20 pmol of each probe. Probes were hybridized in a humidified chamber at 42°C for ~19 hours.

To confirm that the results were robust to imaging modality, we examined Peaks 1 and 4 using widefield fluorescence microscopy with iterative deconvolution, and examined Peaks 2 and 3 using confocal microscopy. Image stacks were acquired with a z-distance of 0.2 μm.

For Peak 1 and Peak 4, slides were imaged using an Olympus IX71 microscope equipped with a CoolSNAP_HQ2/HQ2-ICX285 camera and Olympus PlanApo N 60X/1.42 oil immersion objective. Images were deconvolved (constrained iterated deconvolution algorithm with conservative ratio method) using DeltaVision softWoRx 5.5 software. For Peaks 2 and 3, a PerkinElmer Ultraview Spinning Disk Confocal microscope with Volocity acquisition software and a Hamamatsu ORCA-ER CCD camera was used.

Coordinates for the centroid of each FISH signal and distances between the three centroids at each cluster were obtained using Volocity v.6 software. Intensity thresholds for each signal were manually optimized. To confirm that the centroid coordinates were robust to a wide range of intensity thresholds, we wrote a custom image processing pipeline in ImageJ (Schneider et al., 2012). To rule out bias due to chromatic aberration, we measured the centroid coordinates and mean pairwise distances for all three fluorophores in 108 images of 0.1 μm TetraSpeck microspheres (Life Technologies, T7279). Since we observed a consistent chromatic aberration bias on the z-axis using the TetraSpeck beads, we applied a mean offset to the z-coordinates of the centroids for all experiments we performed in this study before calculating distances. Similar results were observed without correction or using alternative correction strategies (such as correcting x,y, and z axes).

I.d. Karyotyping Methodology

For the G-banding analysis, cells were exposed to Colcemid (0.04 μg/ml) for 25 minutes at 37°C and to hypotonic treatment (0.075 M KCl) for 20 minutes at room temperature. Cells were fixed in a methanol and acetic acid (3:1 by volume) mixture for 15 minutes, and washed three times in the fixative. The slides prepared by air drying technique were aged and treated for the induction of G-banding following the routine procedure (Pathak, 1976).

Spectral karyotyping was performed using the human chromosome SKY probe Applied Spectral Imaging (ASI, Vista, CA, USA) according to the manufacturer's instructions to determine chromosomal rearrangements.

The slides were analyzed using Nikon Eclipse 80i microscope. G-banding as well as SKY images were captured and karyotyped using ASI system. At least 15-20 metaphases were analyzed in detail from each karyotyping experiment.

I.e. Cell Culture

See Table S1 for a list of cell lines used and vendors. All cell lines were cultured according to the supplier's instructions.

Note that, to make our data maximally comparable to that of the ENCODE project, all of our GM12878 cells were obtained from Coriell's "Expansion A", from a lot set aside for ENCODE. The cells derive from two different batches. One of the biological replicates in our dilution experiment (labeled HIC034, br10) and one other biological replicate (HIC048, br13) derive from the first batch ("Batch 1", received at the Broad Institute on 3/11/2008); the rest of our biological replicates derive from the second batch ("Batch 2", received at the Broad Institute on 5/13/2008).

II. Hi-C Data Processing

II.a. Hi-C Data Processing Pipeline

II.a.1. Sequence alignment: All Hi-C data reported in this paper was generated using Illumina paired-end sequencing. Most reads were 101bp paired ends, and the numbers reported below correspond to 101bp PE data unless otherwise noted. The methodology below applies to all read lengths, *mutatis mutandis*.

The Illumina sequencer produces two fastq files, one for each read end. Each file is sorted by "read name." One lane of HiSeq data comprises approximately 150 million raw reads. Uncompressed, the data occupies roughly 80 GB of disk space.

We processed Hi-C sequence data using a custom pipeline, optimized for parallelized computation on a cluster. The pipeline begins by splitting each of the two fastq files into chunks containing 1.5 million single end reads, with roughly 200 chunks for one lane of data. This takes about 25 minutes of wall clock time. Each chunk is mapped to b37 (for human) or mm9 (for mouse) using the Burrows-Wheeler single end aligner, *bwa-sw* (Li and Durbin, 2009), with default parameters. The alignment is performed in parallel on a computer cluster. Each cluster node consists of twelve six-core AMD Opteron 2431 800 MHz processors, and each alignment job requires 6 Gb of RAM. For one lane of HiSeq data, there are ~200 alignment jobs, and BWA takes on average 17 minutes to align the 1.5 million single end reads. We do not perform any trimming or read quality filtering.

(We have found that the newest version of BWA, *bwa mem*, works equally well; the short end aligner *bwa aln* does not return chimeric reads and is therefore suboptimal for aligning long-read paired end Hi-C data where the chimera rate is likely to be high. Paired end aligners should be avoided, as they make assumptions about the insert size that are false for Hi-C data. Since we expect a ligation product, the read ends may be quite far from one another.)

After alignment, each fastq file chunk has a corresponding SAM file. Since the alignment doesn't change the order of the reads, the SAM files for individual chunks are sorted by read name.

Next, the two sorted SAM files for each chunk (corresponding to both the first and second read) are merged into a single, paired-end SAM file. The latter is also sorted by read name. This procedure takes linear time with respect to the length of the chunk (in this case, 1.5 million reads). The procedure is equivalent to the final stage of the classic Merge Sort algorithm. This reduces the number of files from ~200 to ~100 "chunks."

II.a.2. Filtering of abnormal alignments: About 75% of the time, each read in a read pair will align to a single site in the genome. We call such read pairs "normal."

Another 20% of read pairs are "chimeric". This means that at least one of the two reads comprises multiple subsequences, each of which align to different parts of the genome. For instance, the first 50 base pairs might map perfectly to one position, whereas the next 50 map perfectly to a second position several megabases away.

Chimeric read pairs are classified as "unambiguous" or "ambiguous." In an "unambiguous" chimeric read pair, one read maps chimerically to both locus A and locus B, and the other read maps to either locus A or locus B, but not to both. Such reads commonly arise from ligation products between locus A and locus B, in a case where one of the two reads crosses the ligation junction. These "unambiguous" chimeric read pairs comprise roughly 15% of all read pairs and are included in our maps as ligation junctions between locus A and locus B. All other chimeric read pairs are "ambiguous" and are not included in our Hi-C maps.

Finally, about 5% of read pairs are "unalignable": they have at least one end that cannot be successfully aligned. We do not use data from unalignable read pairs in our Hi-C maps.

Note that low-quality "normal" and "unambiguous chimeric" alignments are filtered, but not at this stage; see below.

II.a.3. Filtering of duplicates: After eliminating both ambiguous chimeric reads and unalignable reads, we retain roughly 90% of the original read pairs in the form of paired-end SAM files for each "chunk".

Next, we use binary search to append a restriction fragment number (index of the fragment demarcated by restriction sites in the genome) to each record based on the genome and the restriction enzyme used in the experiment. To do this, we search the genome for all instances of the restriction enzyme's motif, producing a text file that lists all motif locations. Using this text file as a reference, we determine the fragment in which each read lies on the basis of the aligned position of the read, and append this fragment information to each record. At this point, each file chunk consists of one line per read, containing the following fields: read name, strand1, chromosome1, position1, fragment1, strand2, chromosome2, position2, fragment2. These fields describe the alignment of both reads in a read pair.

We then rearrange the record for each read pair so that the chromosome of the first read precedes the chromosome of the second read. If both reads in a read pair are on the same chromosome, we rearrange the read pair so that the fragment of the first read precedes the fragment of the second read. If reads share both chromosome and fragment, we sort by strand; and finally, by position in base pairs. We then use Unix sort to sort all records in the file chunk, with precedence for chromosome, then for fragment, then for strand, and finally for position. Unix sort on relatively small files, such as these, is quite efficient. For each file chunk, assigning the fragment and sorting takes on average 3.5 minutes.

Once they are organized in this way, we merge all 100 chunks into a single master file sorted by chromosome, then fragment, then strand, then position. That is, all chromosome 1 reads will be grouped together; within that set, all fragment 1 reads will come before fragment 2 reads; within the set of chromosome 1/fragment 1 reads, all forward strand reads will come before all reverse strand reads; and finally all position 1 reads will come before all position 2 reads. This can be accomplished in O(n) time by taking the 100 sorted chunks and employing the same merge sort methodology pointed out above. Essentially, this merge takes only the time required to write the file, which is approximately 25 minutes in practice.

Using this sorted master file, it is possible to identify and remove duplicate read pairs in linear time. We consider two read pairs to be duplicates of one another if their reads lie at closely corresponding positions (i.e., within 4bp of one another). More precisely, given two sorted read pairs (A1, A2) and (B1, B2), we compare read A1 to read B1 and read A2 to read B2. The read pairs are considered duplicates if (i) A1 and B1 align to the same chromosome and strand; (ii) the 5' base of A1 aligns to a position within 4bp of the position of the 5' base of B1; (iii) A2 and B2 align to the same chromosome and strand; and (iv) the 5' base of A2 aligns to a position within 4bp of the position of the 5' base of B2.

The duplication removal step is accomplished by a simple awk script. To further speed up the duplicate removal step, we parallelize it. We first split the sorted master file into chunks containing roughly 1 million read pairs each. The files are split at points known not to be duplicates: we start at the beginning of the file, and after 1 million reads, we look for the instance where there is a sufficiently large difference in position between the read pair in one record and the read pair in the next record. The split is performed in "real time" in the following sense: as soon as a break-point is found, the chunk is written out, and the chunk is immediately filtered for duplicates. The filtering takes 30 seconds and occurs at the same time as the sorted master file continues to read and write chunks; total time until the final chunk is written is usually 16 minutes. Because the chunks are numbered, putting them back together merely requires us to concatenate the file. This is dominated, as usual, by the time it takes to write the file, roughly 25 minutes.

The rate of duplication varies from experiment to experiment, as a function of both the Hi-C library's molecular complexity (i.e., the number of unique ligation products contained in the library) and the number of read pairs sequenced. Indeed, the complexity of a given Hi-C library can be estimated based on the number of read pairs sequenced, and the number of duplicates observed (see below).

At this point, the text file contains a duplicate-free list of read pairs. The record for each read pair contains two alignments. To create the final master list of read pairs, we throw out all read pairs where both reads align to the same fragment.

II.a.4. Filtering of low-quality alignments: When creating our Hi-C maps, we require a minimum alignment quality for each read that is included in the map. This requirement is enforced in our very final step, in which we throw out read pairs where the alignment of one or both reads fails to meet this threshold. One of two thresholds is applied: MAPQ > 0 (which means that a unique "best" alignment exists) or MAPQ ≥ 30 (which means that the chances that an alignment is erroneous is at most 1 in 1000). The results of this procedure are two lists of Hi-C "contacts," one for MAPQ>0, and one for MAPQ ≥ 30. Thus, in all of our maps, reads mapping to repeats where a unique mapping cannot be determined are thrown out. While our longer read length allows us to map more often around smaller repeats, there are still regions of low sequence complexity where our contact maps are sparse due to high repeat density. To determine whether, and how well, the Hi-C experiment worked, we calculate a variety of library statistics using the final contact lists (see below). Note that, prior to

this step, a read pair is not called a "contact," and read pairs that are filtered out in this step or previously are not referred to as contacts anywhere in the paper or the Extended Experimental Procedures.

II.a.5. Construction of contact matrices: As discussed in the main text, for each filtered contact list, we generate contact matrices using varying locus sizes. For example, to calculate the contact matrix with a 1 Mb locus size (also called "1 Mb matrix resolution"), we divide the linear genome into 1 Mb bins and count the number of contacts we observe between each pair of bins. The number of contacts observed between locus i and locus j is denoted M_{ij}. For each of the two filtered files (MAPQ > 0 and MAPQ ≥ 30), we calculate contact matrices at resolutions of 2.5 Mb, 1 Mb, 500 kb, 250 kb, 100 kb, 50 kb, 25 kb, 10 kb, and 5 kb; for our largest maps, we also create a 1 kb contact matrix. For our 1 kb matrices in our *in situ* maps constructed using MboI/DpnII and our 5 kb and 10 kb matrices in our dilution Hi-C maps constructed using HindIII/NcoI, it is likely that there are restriction fragments larger than the pixel size. In these cases the contacts for those restriction fragments are split between the two rows/columns containing the ends of the restriction fragments. If the restriction fragment stretches for 3 or more bins, then the middle bins containing no restriction sites will appear as empty columns in the contact matrix, since we know that no contacts can be formed from those intervals. This is especially seen in high-resolution dilution Hi-C maps created by us and all previously published experiments, as the use of HindIII and NcoI restriction enzymes leaves significant restriction site density biases. This is why the panels shown in Figure 1C illustrating what previously published Hi-C maps look like at high resolutions (5 kb and 1 kb) contain many entirely sparse rows and columns.

We also create fragment-delimited contact matrices, using the fragment number assigned to each read in the procedure above. It is thus straightforward to bin using a fixed number of fragments. For each of the two filtered files, we calculate fragment-delimited contact matrices at resolutions of 500f, 200f, 100f, 50f, 20f, 5f, 2f, and 1f.

Throughout the main text and in these supplemental materials, we often say that we examined a given Hi-C map at "higher resolution" and "lower resolution." For clarity, we note that "higher resolution" means decreasing the locus size of the contact matrix (i.e., going from a 100 kb locus size/100 kb matrix resolution to a 25 kb locus size/25 kb matrix resolution). "Lower resolution" means increasing the locus size of the contact matrix (i.e., going from a 5 kb locus size/5 kb matrix resolution to a 250 kb locus size/250 kb matrix resolution).

The pipeline described above was run on every one of the 201 Hi-C experiments performed in this paper as well as on all external Hi-C datasets analyzed in this study.

II.b. Contact Matrix Normalization

Ideally, the entries of the matrix of raw contact counts, M_{ij}, would be proportional to the true contact frequency of locus i and locus j. However, due to biases in the Hi-C experiment, this is not the case. Chromatin accessibility, nucleosome occupancy, alignability and restriction site density at a locus can all affect the contact count. We refer to such effects as "one-dimensional biases": biases that are a function of the locus itself and influence contact frequency between that locus and any other locus. If there is a strong bias towards observing contacts containing locus i, then entry M_{ij} will tend to have more reads, regardless of whether or not locus i and locus j actually interact very frequently. One-dimensional biases affect not only Hi-C, but also all ligation assays (NLA, 3C, 4C, and 5C).

II.b.1. Vanilla coverage normalization (Lieberman-Aiden et al., 2009): There have been various approaches in the literature towards normalizing Hi-C matrices in order to eliminate bias. Our original Hi-C paper (Lieberman-Aiden et al., 2009) addressed this issue via a coverage normalization step. (As noted in the supplemental materials to that paper, this coverage normalization step did not significantly affect the intrachromosomal results reported in Lieberman-Aiden et al. (2009), but it was employed for all interchromosomal analyses presented there.) A row-specific normalization term R_i was calculated by summing the counts in a row (the L_1 norm) and taking the reciprocal. A column-specific normalization term C_j was calculated similarly, by summing the counts in a column and taking the reciprocal. For symmetric (typically intrachromosomal) matrices, $C_i=R_i$. For every entry in the matrix $M_{ij,}$ the normalized matrix entry M^*_{ij} is therefore $R_iM_{ij}C_j$. Here, we refer to this coverage normalization as "vanilla coverage normalization," or "VC normalization". VC normalization is very simple to implement, can be calculated quickly, and is highly robust, even in the setting of extremely sparse data.

There are several ways to motivate VC normalization. One is to compare with the notion of "whole cell extract" (WCE) controls that are frequently used for ChIP-Seq experiments. In ChIP-Seq, differences in signal from locus to locus should ideally reflect differences in binding of the target protein, but in practice can also reflect biases in the protocol due to

factors such as alignability and chromatin accessibility. In WCE controls, the ChIP-Seq experiment is performed without the immunoprecipitation (IP) step. Locus-to-locus variation in the WCE signal reflects the protocol's underlying biases. To incorporate WCE data into a ChIP-Seq experiment, the number of reads seen at a given locus in the ChIP-Seq protocol is divided by the number of reads seen in the WCE-Seq experiment. A second way to think of VC normalization is as a single iteration of the classic matrix balancing algorithm of Sinkhorn and Knopp (Sinkhorn and Knopp, 1967), see below.

One problem with VC normalization is that it tends to overcorrect. A simple approach toward reducing this effect is to use the square root of the VC vector. The square root can be motivated very briefly by observing that such a correction makes the entries of M^*_{ij} dimensionless by changing units of [reads] to units of [reads]/[reads$^{0.5}$][reads$^{0.5}$]. We have found that square root normalization provides values that are surprisingly close to those of much more sophisticated and computationally intensive algorithms (see Figure S1A-B, Data S1.II).

II.b.2. Explicit-factor methods for bias correction: Other groups have highlighted the importance of the Hi-C normalization problem and developed new methods to address it. Yaffe and Tanay (2011) constructed a multiplicative model that explicitly takes into account three factors thought to strongly influence Hi-C coverage: fragment mappability, fragment GC content, and fragment length. Hu et al. (2013) developed a similar approach called HiCNorm.

II.b.3. A role for matrix balancing in Hi-C: It is also possible to normalize matrices without making any presuppositions about which factors are responsible for the observed biases, using a method that is nearly a century old known as "matrix balancing." In matrix balancing we make the assumption – similar to that of VC normalization – that the only biases present are scalar, multiplicative, one-dimensional biases; i.e. that the true matrix of contact probabilities M^*_{ij} is of the form $C_i M_{ij} C_j$, where the C_i are unknown, locus-specific bias factors. By enforcing M^*_{ij} to be doubly stochastic (each of its rows and columns must sum to 1), it is possible to solve for the bias factors C_i.

The problem of matrix balancing is extremely well-studied in data analysis, with the oldest algorithms dating back to the 1930s (Kruithof, 1937). A famous 1967 paper by Sinkhorn and Knopp provided an algorithm that converts any square nonnegative matrix to a doubly stochastic matrix via multiplication by diagonal matrices (Sinkhorn and Knopp, 1967). The algorithm works by repeatedly performing VC normalization on M_{ij} until convergence is achieved, that is, until all of the rows and columns sum to the same value. Sinkhorn and Knopp proved that this algorithm would converge as long as the matrix is nonnegative and has total support. Though Sinkhorn and Knopp were the first to formalize the convergence results, this algorithm has had a very long history; see (Brown et al., 1993; Knight, 2008) for a historical review. The literature on "iterative proportional fitting," dating back to the 1940s, also includes many related methods (Deming and Stephan, 1940; Csiszár, 1975).

Note that the Sinkhorn-Knopp algorithm has been independently rediscovered many times. Recently, two groups working on Hi-C data used matrix balancing for normalization, Cournac et al. (2012) and Imakaev et al. (2012). Cournac et al. calculated the row-specific normalization term R_i using the L_2 norm of the row vector instead of the traditional L_1 norm used in VC normalization and by Sinkhorn and Knopp. Imakaev et al. use the standard Sinkhorn-Knopp-style approach in which VC normalization is repeatedly performed until convergence is achieved.

II.b.4. New methods for matrix balancing: Within the matrix balancing literature, dramatic algorithmic improvements have been made since the 1960s. Recently, Knight and Ruiz introduced a new matrix balancing algorithm whose convergence properties closely resemble those of the Sinkhorn-Knopp algorithm, but which converges much faster (Knight and Ruiz, 2012). Their approach applies a combination of the inexact Newton's method and inner-outer iteration with conjugate gradients to a system of linear equations to quickly find the next matrix in the iteration process. Empirically, their method converges two orders of magnitude faster than the Sinkhorn-Knopp algorithm (Knight and Ruiz, 2012). They provide MATLAB code for their method (hereafter referred to as "KR normalization"), which we reimplemented in Java and incorporated into our pipeline.

KR normalization always results in a balanced matrix as long as the original matrix is not too sparse. We only observed sparsity issues at very high resolutions, which we handled by throwing out the sparsest rows (up to 5% of the total number of rows) and rerunning the algorithm. Because the KR algorithm is so fast and numerically stable, it makes it possible to reliably balance our Hi-C contact matrices at extremely high matrix resolutions.

We calculate the VC and KR normalization vectors for all intrachromosomal contact matrices at all computed resolutions. The intrachromosomal analyses in the paper are all done on contact matrices that have been normalized using the KR intrachromosomal vector. For our largest datasets, we also calculate normalizations interchromosomally and genome-wide. For the interchromosomal normalization, we take the genome-wide contact matrix at each resolution and

remove the intrachromosomal matrices, and then run the VC and KR algorithms to obtain normalization vectors. For the genome-wide normalization, we include both intra- and interchromosomal matrices. These non-intrachromosomal normalizations require a large amount of memory, and so we are only able to calculate them down to 25 kb or 50 kb resolution, depending on the particular dataset.

The features we report in this paper (such as domains, compartmentalization and focal peaks) are robust to any of the above choices of normalization (Data S1.II). We saw extremely high correlations between VC, VC-SQRT, KR, ICE and Cournac, et al. normalization vectors. We observed very high overlaps between peak calls generated with HiCCUPS (see below, Section *VI.a*) using all of the above normalization techniques. In addition, the peaks called by HiCCUPS did not arise as a result of improper normalization; nearly all of the peaks were unaffected when peak calling was performed on the raw data. All figures and analyses employ the KR normalization except when otherwise noted.

II.c. Additional contact matrix analyses.

There are two forms of Hi-C contact matrix analysis that we introduced in Lieberman-Aiden et al. (2009). Although we do not use either method in the main text of the present paper, we refer to them in the supplemental materials and include a description here for the sake of completeness.

II.c.1. The "observed over expected" (O/E) matrix: This uses a genome-wide 1d model to account for the increased number of contacts seen at short distances due to random polymer interactions driven by one-dimensional genome proximity. We calculate this as in Lieberman-Aiden et al. (2009). For a given matrix resolution, we iterate through the possible distances d between locus pairs and count the "observed" number of contacts in all intrachromosomal contact matrices such that $|i-j| = d$. To estimate the "expected," we calculate the total number of locus pairs separated by distance d. The latter is accomplished as follows. Let L_c be the length of chromosome c at the given resolution. Each chromosome contributes $(L_c - d)$ to the number of pixels at distance d. Once we have calculated both of these values, the expected number of contacts per pixel at distance d is the total number of contacts we observe at distance d divided by the number of pixels at distance d. As we move away from the diagonal, this vector becomes noisy. To compensate, we smooth it so that the signal-to-noise ratio is less than 5%; i.e. so that $\sqrt{N} < 0.05N$, which implies that $N > 400$ (where N is the number of reads). This is accomplished by increasing the window size around 1D bins with low counts until the count is greater than or equal to 400.

It is important to note that more sophisticated expected models must be used in order to reliably identify loops; genome-wide 1D distance models are not adequate, and local background must be employed.

II.c.2. Pearson's correlation matrix of the O/E: This identifies spatial relationships between loci by looking for correlations in their contact patterns. The Pearson's correlation matrix can be computed from the O/E matrix, or from the log of the O/E matrix. Pearson correlation matrices can also be produced from interchromosomal contact matrices.

II.d. Hi-C Library Statistics and Quality Control

A Hi-C experiment can fail in a number of ways. Failures are sometimes obvious when viewing Hi-C heatmaps, but their underlying cause can be difficult to diagnose. Here, we describe quality metrics that we calculate on all of our Hi-C libraries. Together, these quality metrics can detect a variety of failure mechanisms. Note that, prior to performing a high-resolution Hi-C experiment, we often sequenced 200K – 2M reads from a "test aliquot" before deciding whether the library quality was sufficiently high to justify deep sequencing. As an example, Table S2 shows library statistics for our *in situ* GM12878 libraries used in our primary and replicate experiments. Note that the "replicate experiment" discussed in the main text is in fact the aggregate of the eight biological replicates shown in Table S2. We discuss these statistics in detail below.

II.d.1. Standard sequencing and alignment statistics: At the top of the table we record a series of statistics that characterize the sequencing and alignment. In a high-quality sequencing run, few reads are unmapped. If more than 10% of reads fail to align, it typically indicates either a problem with the sequencing run, or sample contamination. The frequency of chimeras is an indicator of the frequency of long-range ligation junctions in the data, although the specific value seen depends on numerous experimental parameters, such as aligner and read length. A sudden anomaly in this value as compared to experiments with similar parameters can suggest a failure of the ligation step.

II.d.2. Duplicate frequency: High duplication rate indicates low molecular complexity (Lander and Waterman, 1988). A Hi-C library was considered a good candidate for deeper sequencing if our complexity estimates suggested it consisted of hundreds of millions or billions of unique contacts. In general, in constructing high resolution Hi-C maps, we found that it was more efficient to sequence many replicate libraries at lower depth (e.g. one lane on an Illumina HiSeq, or less than 20% of the total library complexity) rather than sequencing many lanes of a single library, since the latter strategy typically leads to extensive duplication. In addition, the duplicate removal step requires vastly more time and memory if a single library is sequenced very deeply. For instance, if we estimated that an otherwise high-quality library contains 1.5 billion contacts, we usually planned to sequence at most two lanes from that library on an Illumina HiSeq (~300M reads total).

The exact molecular complexity was calculated using the Picard tools formulation of the Lander-Waterman equation (Lander and Waterman, 1988), which entails solving the following equation for the molecular complexity *m* (measured in molecules):

$$R/m = 1 - e^{N/m}$$

Here *R* is the number of distinct reads observed, *N* is the total number of reads sequenced. Note that "optical duplicates" created by the Illumina sequencing process rather than PCR were not included in either *R* or *N*. In our experience, the resulting value is typically an underestimate of the true library complexity.

II.d.3. Fraction of "Hi-C contacts": After duplication removal, we filter out read pairs where both ends align to the same fragment. If this step filters out over 20% of read pairs, it indicates that the library failed in the restriction, fill-in or ligation steps of the protocol and thus is not a good candidate for deeper sequencing.

We also filter out read pairs where the mapping quality of either read falls short of the desired threshold. (In the example of Table S2, the threshold is MAPQ > 0.) The rest of the quality metrics are calculated using the list of read pairs that remain once all filtering was completed. We refer to these read pairs as "contacts."

II.d.4. Ligations: This statistic measures how often a ligation junction is found inside a read. (A ligation junction is the sequence created when the ends of two filled-in restriction fragments ligate to one another. For MboI, the ligation junction sequence is GATCGATC. For HindIII, the sequence is AAGCTAGCTT.) A paucity of ligation junctions in a Hi-C library suggests that the ligation failed. Note that there can be many causes of such a failure, ranging from a bad batch of DNA ligase to rupture of the cell nuclei. This statistic is also dependent on sequence read length and insert size. We typically sequenced a 300-500bp insert using 101bp PE reads, and observed that ligation rates tended to fall into the 30-40% range. Of course, with shorter reads and longer insert sizes, this value tends to be smaller. With longer reads and shorter inserts, it is larger.

II.d.5. Proximity to 5' and 3' restriction fragment ends: For long-range contacts (defined as intrachromosomal and over 20 kb apart, or interchromosomal), we looked at both read ends to see if the end is closer to the 5' or 3' end of the restriction fragment and to which strand the read maps. When Hi-C libraries are generated using a six-cutter restriction enzyme and, after the shearing step, are size selected for 300-500bp molecules, we find that the large fragment size to insert size ratio causes most contacts (>85%) to come from the 3' ends of fragments. A much lower value indicates that the restriction enzyme had not cut effectively. Note that when the fragment size to insert size ratio declines (i.e., when a four-cutter restriction enzyme is used) we find that the 5' to 3' bias is markedly attenuated.

II.d.6. Percentage of contacts at various distances: We broke down contacts into intrachromosomal and interchromosomal contacts. We then further subdivided the intrachromosomal contacts to short range (<20 kb) and long range (>20 kb) contacts.

A crucial metric is the percentage of long-range intrachromosomal contacts. In successful Hi-C libraries, we found that at least 15% of unique reads were long-range intrachromosomal contacts. Lower values usually indicated that the experiment had failed. If more than 40% of unique reads are long-range intrachromosomal contacts, a library was considered a good candidate for sequencing. If the fraction was above half, a library was considered an excellent candidate for sequencing. In general, this value was one of the statistics we found most important to scrutinize in performing cost-effective high-depth Hi-C.

A library with many interchromosomal contacts and a paucity of contacts at shorter distances (i.e., absence of both a strong diagonal and robust distance decay effects in the intrachromosomal contact matrices) suggests that the library comprises mostly random ligation products, likely due to the rupture of a large fraction of nuclei.

II.d.7. Percentage of contacts by read pair type: We broke down intrachromosomal contacts by type: in a "left" pair, both ends map to the reverse strand. In a "right" pair, both reads map to the forward strand. In an "inner" pair, the ends map to different strands and point (5' to 3') towards each other. In an "outer" pair, reads land on opposite strands but point away from one another. If the chimeras observed are due to proximity ligation, this statistic should be random, i.e., each pair type should account for roughly 25% of contacts. Thus, the distance at which the percentage of each pair type converges to 25% is a good indication of the minimum distance at which it is meaningful to examine Hi-C contact patterns. For six-cutter restriction enzymes, such as HindIII and NcoI, this distance is approximately 30 kb. For four-cutter restriction enzymes (MboI, DpnII), this distance is approximately 3 kb (Figure S1D). Note that the existence of read pairs in the "right" and "left" configuration is rarely seen outside of Hi-C experiments. DNA-Seq reads, for instance, are by design all "inner" pairs; "jumping libraries" tend to produce outer pairs.

III. Evaluation of *in situ* Hi-C

III.a. Ligation Takes Place Inside the Nucleus in *in situ* Hi-C

In order to verify that ligation occurs *in situ* in intact nuclei rather than in dilute solution, we first confirmed that nuclei remain intact by inspecting the cells under a microscope at various steps throughout the protocol: (A) while cells are in the lysis buffer (after step 7), (B) before restriction (after step 12), (C) after restriction with MboI and subsequent heat inactivation of MboI (after step 13), and (D) after ligation (after step 17). The presence of intact nuclei was visually apparent in all cases.

We also conducted a series of experiments to ensure that the ligation junctions were forming *in situ.* First, we performed a pellet/supernatant variant of our *in situ* protocol (see Section *I.a.5*). If most or all of the ligation between crosslinked chromatin was occurring in solution, we would expect that the Hi-C library obtained from the supernatant would be of at least comparable quality to the Hi-C library obtained from the nuclei, and possibly better. We sequenced half a lane of each library on a HiSeq 2500 (pellet: ~69M contacts, supernatant: ~31M contacts) and found that while the library generated from the nuclei was comparable to our other libraries generated using *in situ* Hi-C, the library generated from the supernatant consisted primarily of random ligations (Data S1.I.B).

In addition, we generated two libraries using agar *in situ* Hi-C. By embedding nuclei in agar plugs during the protocol, we ensure that any ligations seen in these libraries would be occurring inside embedded nuclei. We sequenced half a lane of each library on a HiSeq 2500 (~88M contacts and ~81M contacts) and verified that both libraries were of high-quality and closely resembled the results of *in situ* Hi-C (Data S1.I.B).

Finally, we constructed 5 libraries without the use of crosslinking (four in agar, one without). In all five cases, we observed meaningful compartment, domain and loop signals. These signals could not be obtained if meaningful ligations were occurring in solution, because without crosslinking there is nothing which keeps DNA fragments in proximity in solution.

III.b. *In situ* Hi-C Minimizes Random Convection of Chromatin Present in Dilution Hi-C

To confirm that the nuclear membrane reduces chromatin diffusion during *in situ* Hi-C, we counted the number of contacts between the mitochondrial genome (MT) and the nuclear genome (following an approach suggested in Dixon et al. (2012)), since we know that these contacts can only occur if DNA migrates across both the nuclear and mitochondrial membranes. Upon examining our two GM12878 maps, we found that there were 4M contacts between the mitochondrion and the nuclear genome in dilution Hi-C, but only 1.6M such reads in *in situ* Hi-C. In contrast, the number of intra-mitochondrial contacts was the same in both maps ($2.5x10^5$ and $2.7x10^5$ reads in dilution and *in situ*, respectively). This implies there is a 2.8-fold depletion of junctions between mitochondrial and nuclear DNA in the *in situ* Hi-C protocol as compared to the dilution Hi-C protocol. This in turn implies that membranes are at least partially intact in *in situ* Hi-C, and lead to a reduction in cross-membrane chromatin migration.

Notably, Kalhor et al. (2012) introduced a "TCC" variant of Hi-C in which ligation is performed on beads. TCC takes more time than *in situ* Hi-C (7 days vs. 3 days). It also produces fewer high quality contacts. For instance, when we compared the fraction of sequenced reads that corresponded to intra-large contacts in Kalhor's MboI TCC experiment as compared to one of our MboI *in situ* Hi-C experiments, we found that the latter had markedly higher yield: 23.9% for TCC vs. 41.0% for *in situ* Hi-C.

III.c. Our GM12878 *in situ* Hi-C Map Achieves a Resolution of ~1 kb

The map resolution of our combined (primary and replicate) GM12878 map was calculated by counting the number of contacts for every 50bp bin in the genome (defined as any contact where one read mapped within that 50bp bin). This vector was then binned into larger and larger increments (i.e. 100bp, 150bp, etc.) until the number of bins with >1000 contacts was at least 80% of the total number of bins. This value, 950bp, is the map resolution we reported for our GM12878 map.

Of course, in reality, the effective resolution of our GM12878 map varies along the genome as it depends on the density of restriction sites. In fact, there is a large dynamic range of contacts formed by 1 kb loci across the genome: most loci are

covered by far more than 1000 contacts. 75% of loci have at least 2015 contacts; 50% have at least 3241; 25% have over 4729, and 1% have over 8585 contacts. As such, one will have greater power to detect fine scale structure at some loci over others. However, we calculate this map resolution in order to give a general sense of size scales that can be examined in our data set.

It is also important to mention that our definition of map resolution is bounded below by the average fragment size. That is, the highest resolution that one can attain in a Hi-C experiment is single fragment resolution. Since 950bp is greater than the average MboI fragment size (~400bp), we thought it would be appropriate to describe the resolution of the map in an easily interpretable way.

III.d. Data Generated Using *in situ* Hi-C is Extremely Reproducible

The maps we generated using both *in situ* Hi-C and dilution Hi-C are highly reproducible at a variety of matrix resolutions. We compared the data generated by our primary GM12878 *in situ* experiment, our replicate GM12878 *in situ* experiment, and our dilution GM12878 experiment at 500 kb, 50 kb, and 25 kb resolution (evaluated by flattening the contact matrices and correlating the resulting vectors); we saw extremely high correlations at all resolutions (*is*Hi-C primary vs. *is*Hi-C replicate: Pearson's $r > .998$ (500 kb), $r > .996$ (50 kb), $r > .993$ (25 kb); *is*Hi-C primary vs. dilution Hi-C: $r > .96$ (500 kb), $r > .90$ (50 kb), $r > .87$ (25 kb); *is*Hi-C replicate vs. dilution Hi-C: $r > .97$ (500 kb), $r > .92$ (50 kb), $r > .90$ (25 kb); p-values $< 10^{-324}$ in all cases, bivariate normal distribution). All correlations and p-values were calculated using the *scipy.stats.pearsonr* function.

Even at extremely high matrix resolutions (5 kb, 1 kb), the contact matrices from our primary and replicate experiments were highly correlated (we did not correlate our *in situ* data with our dilution data at extremely high matrix resolutions; this would be confounded by fragment density biases in the dilution data). At 5 kb resolution, we flattened all intrachromosomal contact matrices (except for chromosomes >120 Mb long, which we split in half in order to examine the submatrices separately) and correlated the resulting vectors from our primary and replicate *in situ* experiments. At 1 kb resolution, we flattened the submatrices corresponding to disjoint 10 Mb tiles along every intrachromosomal contact matrix and correlated the resulting vectors from our primary and replicate experiments. (Pearson's $r > .96$ (5 kb), Pearson's $r > .85$ (1 kb); p-values $< 10^{-324}$, bivariate normal distribution).

While we calculated correlations on entire flattened contact matrices in order to generate metrics comparable to those calculated previously in the literature (Lieberman-Aiden et al., 2009; Yaffe and Tanay, 2011; Dixon et al., 2012), correlation of full intrachromosomal observed contact matrices is confounded by the distance dependence of interactions. In order to better assess the reproducibility of our data independent of the genomic distance-dependence of Hi-C interactions, we measured intrachromosomal correlations in two other ways: (1) we calculated the Pearson's *r* of flattened observed/expected matrices (see construction of O/E matrix, Section *II.c.1*) at various resolutions (500 kb, 50 kb, 25 kb) between various maps (primary, replicate, dilution), and (2) we calculated the Pearson's r between Hi-C maps as a function of a distance (i.e. correlated each of the N upper diagonals between two NxN contact matrices). Once again, even at high resolutions, our data remained highly correlated (O/E Pearson's *r* (500,50,25 kb): *is*Hi-C primary vs. *is*Hi-C replicate, $r > .985, >.65, >.40$; *is*Hi-C primary vs. dilution Hi-C, $r > .94, >.51, >.36$; *is*Hi-C replicate vs. dilution Hi-C, $r > .94, >.54, >.38$; p-values $< 10^{-324}$ in all case, bivariate normal distribution; see Figure S1E-I for correlation as a function of distance).

Additionally, we observed strong reproducibility between our interchromosomal contact matrices from our primary and replicate experiments; unlike the intrachromosomal matrices, interchromosomal matrices are not affected by distance dependence. At 500 kb matrix resolution, the average Pearson's *r* was >.78, p-value$< 10^{-324}$ (Figure S1J).

III.e. Measurement of bias in Hi-C experiments via a No-ligation control

III.e.1. Relationship between Hi-C coverage and chromatin accessibility: We compared the vanilla coverage normalization factors for loci at various fragment resolutions to the chromatin accessibility of the fragment as assessed by DNase-Seq. The vanilla coverage normalization factor (see Section *II.b.1*) provides information about how overrepresented a certain locus is: a locus with a VC factor of 3 is 3-fold more represented than the average locus, a locus with a VC factor of .33 is 3-fold less represented than the average locus. Many experimental biases can affect the coverage of a locus in a proximity ligation experiment including the chromatin accessibility of a locus. Loci that are digested more frequently than the average locus are more available to form contacts and thus will have higher Hi-C coverage. The efficiency of digestion is

closely tied to the accessibility of the locus; in fact, it is this principle that underlies DNase digestion experiments to assay open chromatin.

We conducted all analyses in this section at fragment resolution to avoid any contribution of restriction site density to Hi-C coverage. We found that Hi-C coverage and DNase-Seq signal were highly correlated: the VC factors of 20 fragment MboI loci in our primary and replicate combined GM12878 map and the DNase-Seq signal over the 20 fragment MboI loci exhibited a Spearman's *r* of .69 (p-value negligible). Similar results were seen for HindIII (VC factors for 5 fragment HindIII loci in our dilution GM12878 map were highly correlated with DNase-Seq signal; Spearman's *r*=.45, p-value negligible) and for DpnII (VC factors for 20 fragment DpnII loci in our DpnII *in situ* map in GM12878 were highly correlated with DNase-Seq signal; Spearman's *r*=.69, p-value negligible). 10 fragment MboI loci (5 fragment HindIII loci) with VC factors >2 had a median DNase-Seq signal 3.5-fold (1.6-fold) greater than the median for all loci; similarly 10 fragment MboI loci (5 fragment HindIII loci) with VC factors <0.5 had a median DNase-Seq signal 1.6-fold (1.3-fold) lower than the median for all loci.

While our data becomes noisier when analyzed at single fragment resolution, the same effect was seen. MboI (DpnII) fragments with VC factors >3 had a median DNase-Seq signal 1.9-fold (2-fold) greater than the median for all loci, while MboI (DpnII) fragments with VC factors <0.33 had a median DNase-Seq signal 1.3-fold (1.7-fold) lower than the median for all loci.

This effect cannot be explained by differences in MboI restriction efficiency due to its sensitivity to CpG methylation. First, the MboI motif (5'-GATC-3') does not contain a CpG, and can only be CpG methylated if the 5' upstream base is C or the 3' downstream base is G. This is true for only 11% of MboI motifs in hg19, hence only a small percentage of all MboI motifs can be CpG methylated. Furthermore, when we examined only those MboI fragments that were flanked by restriction sites that could not be methylated, we once again saw that fragments with VC factors >3 had a median DNase-Seq signal 1.9-fold greater than median for all loci. Finally, similar results were seen with two methylation insensitive restriction enzymes, DpnII and HindIII. (Note that DpnII is a methylation-insensitive isoschizomer of MboI.)

III.e.2. No-ligation control strongly correlates with Hi-C coverage and chromatin accessibility: In order to confirm whether the correlation we observed between Hi-C coverage and chromatin accessibility was influenced by differential digestion at restriction sites with varying chromatin accessibility, we utilized our no-ligation controls, where cross-linked genomic DNA was prepped for sequencing directly after digestion. As stated in Section *I.a.11*, we performed two no-ligation controls, one using HindIII (113M paired-end reads) and one using MboI (114M paired-end reads). As expected, we observed a close correspondence between the frequency of cutting at a restriction site, the chromatin accessibility at that site (as assayed by Dnase-Seq) and the coverage of that restriction site. In addition, we observed an extremely close correspondence between the experimentally determined biases using the no-ligation control and the computationally determined biases using KR normalization (At 10 kb resolution, Spearman's $r > .77$, p-value negligible). This suggests that our matrix balancing normalization approach effectively corrects biases due to digestion efficiency, in addition to other biases. Furthermore, when we corrected the Hi-C data based on the empirical cutting frequencies, the loops we report were essentially unaffected (>90% are identical).

IV. Domains in Hi-C Maps

IV.a. Arrowhead Algorithm for Domain Annotation

IV.a.1. Motivation and related work: The formation of square megadomains along the diagonal of a contact map is a striking feature that was apparent in our 2009 maps, and which we explained in terms of compartmentalization (Lieberman-Aiden et al., 2009). Subsequent work has highlighted the computational problem of identifying domains (which manifest as squares along the diagonal of a contact map) as distinct from the problem of identifying compartments (Dixon et al., 2012; Sexton et al., 2012).

The fact that domains manifest as squares along the diagonal of a contact map suggests that they should be straightforward to identify. In practice, however, the identification of domains is tricky. This is due to experimental factors such as noise and inadequate coverage. It is also because of the intrinsic difficulty of the problem: the decline in contact frequency at domain edges can be subtle, and the very rapid decline in contact probability observed as one moves away from the diagonal of a contact map is a major confound for most approaches.

Nevertheless, several methods exist for identifying domains. Notably, Dixon et al. (2012) defined a directionality index (DI), which measures the tendency of a locus to interact with upstream vs. downstream sites. This is useful for identifying domains because the upstream boundary of a domain should prefer to interact with downstream loci, and vice-versa.

IV.a.2. Description of Arrowhead transformation: The arrowhead transformation (see Figure S2A-F) is a matrix transformation defined as $A_{i,i+d} = (M^*_{i,i-d} - M^*_{i,i+d})/(M^*_{i,i-d} + M^*_{i,i+d})$. This transformation can be thought of as equivalent to calculating a matrix equal to -1*((observed/expected)-1), where the expected model controls for local background and distance from the diagonal in the simplest possible way: the "expected" value at *i,i+d* is simply the mean observed value at *i,i-d* and *i,i+d*. By choosing variants on this expected model, one can create a family of related transformations with similar properties. Alternatively, one can think of $A_{i,i+d}$ as a measurement of the directionality preference of locus *i*, restricted to contacts at a linear distance of *d*.

Consider the behavior of this transformation when a domain is present in M^* between locus *a* and locus *b* (i.e., there is a square of enriched contact frequency whose vertices lie at <*a,a*>, <*a,b*>, <*b,b*>, and <*b,a*>). $A_{i,i+d}$ will be strongly positive if and only if locus *i-d* is inside the domain (i.e., in the range *[a,b]*) and locus *i+d* is not. $A_{i,i+d}$ will be strongly negative when locus *i+d* is inside the domain and locus *i-d* is not. If both loci are inside the domain, or both loci are outside the domain, $A_{i,i+d}$ will be close to zero. (Note that this behavior also exploits the fact that one typically observes squares of depleted contact frequency adjacent to domains.)

Thus, the general behavior of the arrowhead matrix A can be seen by solving a series of simple inequalities that follow from the above statement. If we think of the solution geometrically, we see that *A* takes on very negative values inside an "upper" triangle $U_{a,b}$, whose vertices lie at *[a,a]*, *[a,b]*, and *[(a+b)/2,b]*. We also see that *A* takes on very positive values inside a "lower" triangle $L_{a,b}$, whose vertices lie at *[(a+b)/2,b]*, *[b,b]*, and *[b,2b-a]*. Everywhere else, the entries of *A* are close to zero.

One can think of the "upper" and "lower" triangles as a smear that exaggerates the original edges of the domain, making these features easier to detect. The negligible values seen everywhere else also have an important consequence: they replace the steep decline seen inside a domain in M^* – which tends to confound feature detection algorithms – with a relatively constant region in *A*. Because of the mirror symmetry of the matrix *A*, the effect of the transformation, when examined as a whole, is to transform an (abnormally-hard-to-annotate) square feature into a (relatively-easy-to-annotate) arrowhead shaped feature. See Figure S2A-F.

IV.a.3. Arrowhead scoring: The goal of our algorithm is to identify the pairs of loci *a* and *b*, where there is a domain between *a* and *b* (equivalently, where the pixel $M^*_{a,b}$ is the corner of a domain). As noted above, it is useful to apply the arrowhead transform to M^*, yielding the arrowhead matrix *A*. Every domain will produce the two triangles $U_{a,b}$ and $L_{a,b}$ described above. By empirically studying the results of *A* on a series of domains, we noted the following facts about $U_{a,b}$ and $L_{a,b}$:

(i) almost all entries in $U_{a,b}$ are negative, and almost all entries of $L_{a,b}$ are positive.
(ii) when the sum of the entries in $U_{a,b}$ is subtracted from the sum of the values in $L_{a,b}$, the resulting value is large (relative to a random model)

(iii) the variance of the entries in $U_{a,b}$ and $L_{a,b}$ were both small (relative to a random model).

These properties were not satisfied when $M^*_{a,b}$ was not a domain corner. We therefore used these three observations as a heuristic to find domain corners. To calculate the corner score for a pixel $M^*_{a,b}$, we first calculate a set of subscores for the corresponding $U_{a,b}$ and $L_{a,b}$: $S_{sign,}$ the sum of the signs of entries in $L_{a,b}$ minus the sum of the signs of the entries in $U_{a,b}$; $S_{sum,}$ the sum of the values of entries in $L_{a,b}$ minus the sum of the values of entries in $U_{a,b}$; and $S_{variance,}$ the total variances of both $U_{a,b}$ and $L_{a,b}$. We normalize each of these three subscores by calculating each score for every possible *a,b*, and then dividing by the maximal value observed. The "raw corner score" matrix *S'* comprises the sum of the three normalized scores for all pixels $M^*_{a,b}$. If $M^*_{a,b}$ is a true domain corner, the value of $S'_{a,b}$ will typically be large.

To identify domain corners using the corner score, we create a filtered version of the matrix *S'*, labeled *S*, in which we set all pixels whose individual subscores do not pass certain thresholds to zero. These thresholds were determined empirically; we believe most of the genome to be partitioned into domains, but erred on the side of fewer false positives when choosing thresholds. We apply thresholding twice, and in each round choose two thresholds, t_1 and t_2. In the first pass, we look for small, very distinct blocks with low variance ($S_{variance}$ <0.2 = t_1; Mean(sgn($U_{a,b}$))<-0.5 = -t_2.; Mean(sgn($L_{a,b}$))>0.5 = t_2.). In the second pass, we identify larger blocks (Mean(sgn($U_{a,b}$))<-0.4 = -t_2.; Mean(sgn($L_{a,b}$))>0.4 = t_2.). These larger blocks are not permitted to contain any of the previously annotated smaller blocks.

When we examine the matrix *S,* we find that corners of domains appear as blobs of high scoring pixels. To precisely annotate domain corners, we first use MATLAB's connected component algorithm to identify groups of adjacent pixels. The pixel within the connected component whose corner score *S* is largest is marked as the domain corner. See Figure S2C,F.

IV.a.4. Dynamic programming for fast calculation: Naively, the above algorithm would require us to calculate all the above noted scores for $U_{a,b}$ and $L_{a,b}$ for all *a,b*. Thus, the naive running time of the above algorithm is $O(n^4)$, where *n* is the number of loci in the genome. This makes the algorithm infeasible on large-scale datasets.

However, we developed a dynamic programming implementation of this scheme which requires only $O(n^2)$ operations, which makes the algorithm much more useful in practice.

To create a more practical implementation, we realized that summing entries of a matrix contained in $U_{a,b}$ and $L_{a,b}$ can be thought of as summing the calculations for smaller triangles, plus a sum for the additional row or column. In particular, given the sum for $U_{a,b-1}$, we add the column *b* sum from rows (*a+b*)/2 to *a*, and similarly for $L_{a,b-1}$. The additional column and row sums are themselves calculated ahead of time via dynamic programming and then accessed when needed to calculate sums for all possible $U_{a,b}$ and $L_{a,b}$

This approach can be applied very broadly. For instance, the variance of the entries in $U_{a,b}$ and $L_{a,b}$, the score matrix $S_{variance}$, can be calculated using dynamic programming by transforming the problem into a sum, relying on the fact that $Var(X) = E[(X-\mu)^2] = E[X^2] - (E[X])^2$.

By exploiting this method, all the above scores can be calculated using only $O(n^2)$ operations.

IV.b. Random Shuffling Control Algorithms

When performing quantitative analyses on our feature annotations, it was frequently desirable to have a "random control" for the feature annotation in question. We generated such annotations through a random permutation procedure. For one-dimensional features, such as peak loci, we randomly placed the one-dimensional features throughout the genome such that (1) the number of features on any one chromosome stayed the same; (2) the random features did not overlap any gaps in the assembly (i.e. centromeres, telomeres, etc.). Similarly, for two-dimensional features (domains, peaks), we randomly placed the two ends of the features across the genome such that (1) the size distribution of the two-dimensional features stayed the same; (2) the number of features on any one chromosome stayed the same; (3) the interval between the ends of the randomized two-dimensional features did not overlap any gaps in the assembly. The random shuffling procedure was used to compare the correlation of chromatin marks within domains and within random domains (Section *IV.c.2*), to assess the changes in long-range contact pattern at the boundaries of domains and random domains (section *IV.c.3),* and to create random domain lists for assessment of conservation of GM12878 domains in other cell types (Section *IV.c.4*). The random shuffling procedure was also used to create the random peak lists and peak locus lists for assessment of the likelihood that peaks lie at the corner of domains (Section *VI.e.5*), assessment of depletion of "loop

crossings" (Section *VI.e.6*), assessment of protein enrichment on peak loci (Section *VI.e.7*), and assessment of the enrichment of convergent motif orientations (Section *VI.e.8*).

IV.c. Properties of Domains

The Arrowhead algorithm on GM12878 was performed at 5 kb resolution. When we performed the Arrowhead algorithm separately on our primary and our replicate GM12878 maps, we called 7105 and 6082 domains respectively, 5041 of which overlapped. This suggested high reproducibility of our domain annotation algorithm. We then applied the Arrowhead algorithm to our combined GM12878 map; the resulting annotation of 9,274 domains was used for subsequent analyses.

IV.c.1. Interaction probability drops at the boundaries of domains: Loci within a domain preferentially form contacts with other loci inside the domain relative to neighboring loci outside the domain. This creates a drop in contacts at the borders of domains that is visually apparent in Hi-C maps. To quantify this drop in contacts, we assessed the ratio of inter-domain contacts to intra-domain contacts at various distances, *d*, in our *in situ* map at 25 kb resolution. To do this, we took all pairs of 25 kb loci (that were separated up to a maximum distance of 475 kb) and split these pairs into two lists: those for which both loci in the pair were in the same domain, and those for which the two loci were not in the same domain (at least one locus had to be annotated inside a domain). We then calculated the mean contact frequency at a given distance for each of the two lists. Figure S2G shows the ratio of the two mean contact frequencies as function of distance.

IV.c.2. Domains exhibit consistent patterns of histone modifications: To determine how domain structure affected chromatin marks, we first took each of our domains and divided it into 10 bins, where the bin size was a tenth of the size of the domain. For each domain, we then recorded the mean value of the chromatin mark of interest within each of these bins. We also recorded the mean chromatin mark value in the ten bins to the left and to the right of the domain boundaries, where the bin size was set to a fixed size (10 kb). This was the procedure used for the matrices shown in Fig 2B. We repeated this procedure with one additional variation, setting the size of the loci flanking a domain to the size of the loci within the domain. This was the procedure used for the matrices shown in Fig S2I-J. Results from both methods were similar. To control for outliers, bins whose mark values were above the 99.9th percentile of all bins over all domains were reduced to the value of the 99.9th percentile.

For a chromatin mark of interest, the above procedures yielded a matrix whose length was the number of domains, and whose width was 30. By calculating the correlation of the columns of this matrix, we obtain a 30x30 correlation matrix that can be computed for any specific chromatin mark. This correlation represents how correlated the chromatin marks are at any two loci, and makes it possible to explore the effects of domain boundaries. The correlation matrices show that chromatin marks exhibit strong positive correlation within domains, and a sharp drop in correlation at the domain boundaries. This is in marked contrast to the result we obtain when we randomly shuffle our domain list (see Section *IV.b*). For random domains, we expect loci near each other in the genome to have correlated chromatin marks; however we do not expect anything special to occur at the random domain boundaries. Indeed, the correlation matrices for chromatin marks near random domains are smooth.

We can also compute the correlation between different chromatin marks at different loci relative to a chromatin domain. We expect repressive marks like H3K27me3 to anticorrelate with active marks such as H3K36me3. This is what we see. The correlation matrices between two chromatin marks also display strong drops at the domain boundaries, in contrast to the random domain correlation matrices. All correlations matrices (for true and random domains), with the bin size kept constant both inside and outside of domains, are shown in Figure S2K-L.

Recently, Naughton et al. (2013) devised a high-throughput experiment to measure the amount of supercoiling in cells by using biotinylated 4,5,8-trimethylpsoralen (bTMP) as a probe. They measured the amount of bTMP binding on chromosome 11, where enriched binding (relative to input) indicates negative supercoiling and depleted binding indicates positive supercoiling. In Naughton et al. (2013) the authors found that 30% of the boundaries of their annotated supercoiling domains fell within 20 kb of topological domain boundaries. Here we examined the bTMP signal/input track across whole domains in chromosome 11, and found that, like many of the epigenetic marks discussed here, there was higher correlation of supercoiling signal between loci located in the same domain than between loci located in different domains (see Figure S2J). While the correlation drop at the domain boundaries does not seem as sharp as the drops that appear in the chromatin modification correlation matrices, it is certainly enriched above random domains, indicating that supercoiling status is related to domain structure. This is consistent with early studies (Cook and Brazell, 1975; Vogelstein et al., 1980; Zehnbauer and Vogelstein, 1985; Goldman, 1988) regarding chromatin organization which posited that the genome was folded into distinct 'topologically constrained domains' or 'chromatin domains', each of which was

thought to be “a unit of supercoiling, in that its torsional state is independent of the torsional state of the surrounding loops” (Goldman, 1988).

We also computed the correlation of chromatin signals at a fixed distance for loci in the same and in different domains, and found for instance, the correlation between H3K36me3 (resp. H3K27me3) signals for two loci 50 kb apart was 0.52 (resp. 0.59) if the loci were in the same domain, but only 0.23 (resp. 0.19) if they were not. For this calculation we took all pairs of 25 kb loci separated by a 50 kb interval and split them into two lists: pairs for which both loci were in the same domain, and pairs for which the two loci were in different domains (at least one locus had to be annotated inside a domain). We then calculated the Spearman correlation of chromatin marks at these pairs of loci, and found all marks are more correlated if they are in the same domain than if they are in different domains (see Figure S2I).

We note that, while we see that all marks are more correlated between loci in the same domain than between loci in different domains, the strength of the correlation will depend on many factors. The various chromatin marks differ in terms of how easily they spread, how specifically they bind to punctate versus broad features in the genome, how frequently they appear in a given cell-line, etc. All of these factors will influence the strength of the correlation when averaged over the length of the domain; thus, in order to compare correlation values between any two chromatin marks, care must be taken to account for these differences.

IV.c.3. Changes in patterns of long-range contact tend to occur at the boundaries of domains: When examining Hi-C maps, we noticed that loci within a domain seemed to have the same long-range interaction pattern, while changes in long-range interaction patterns occurred on the boundary between domains. To quantify this, we devised a gradient score, which measured the difference in long-range interaction pattern between all neighboring loci.

For each 25 kb locus, *i,* along the genome, we calculated a score $G_{i,j}$ at every pixel $M^*_{i,j}$ for all $|i-j| > 10$ Mb and < 40 Mb, where:

$$G_{i,j} = \frac{(A_{i,j} - E_{i,j})^2}{E_{i,j}} + \frac{(B_{i,j} - E_{i,j})^2}{E_{i,j}}$$

and:

$$A_{i,j} = \sum_{a=i-4}^{i-1} \sum_{b=j-2}^{j+2} M^*_{a,b}$$

$$B_{i,j} = \sum_{a=i+1}^{i+4} \sum_{b=j-2}^{j+2} M^*_{a,b}$$

$$E_{i,j} = \frac{A_{i,j} + B_{i,j}}{2}$$

Our final gradient score for every locus *i*, G_i, was the sum of all $G_{i,j}$ for all $|i-j| > 10$ Mb and < 40 Mb. We then examined the distribution of *G* at bins inside domains and at domain boundaries, and then repeated this procedure with domains defined by a random shuffle of our domain list. Values of *G* were higher at the boundaries of true domains and were depleted within domains as compared to our randomly shuffled domain list (see Figure S2H), indicating that changes in long range interaction patterns tend to occur at domain boundaries.

IV.c.4. Domains are conserved across many cell types: Previous analyses of proximity ligation data have suggested that topological domains (TADs) are strongly conserved between cell types (Dixon et al., 2012). More specifically, however, this analysis by Dixon, et al. did not explore the preservation of entire TADs. Instead, they created lists of domain boundaries - which, as we note below (section *IV.c.6*), they are very good at calling - and showed that these boundaries were strongly preserved. Of course, the preservation of domain boundaries does not imply the preservation of whole domains. For instance, if a domain in one cell type splits into two domains in another cell type, both of the original boundaries are preserved, although the domain itself is now gone, having been split along a new boundary. The rate of preservation of entire domains, be they TADs or physical domains, has not, to our knowledge, been quantified by any group and is of obvious interest.

In order to assess the conservation of domains across human cell types, we employed two strategies. First, we identified domains using the Arrowhead algorithm (see Section *IV.a*) at 10 kb resolution in all seven of the other human

cell types that we produced high resolution maps for (IMR90, HMEC, NHEK, K562, KBM7, HUVEC and HeLa). We identified 7680 domains in IMR90, 4096 domains in HMEC, 6014 domains in NHEK, 5975 domains in K562, 4949 domains in KBM7, 4173 domains in HUVEC, and 4475 domains in HeLa. Because the number of domains identified in GM12878 was greater than the number of domains identified in other cell types (largely due to sequencing depth), we calculated the percentage of domains identified in non-GM12878 cell types that were also preserved in GM12878. We defined a domain call as "preserved" if there was a called domain in a cell type with top right corner at $M_{i,j}$ and a called domain in GM12878 with top right corner at $M_{i',j'}$ such that $[(i-i')^2+(j-j')^2]^{1/2} \leq min(0.2*|i-j|, 50\text{ kb})$. We found that, on average, 54% of domains called in a given cell type were also called in GM12878 (Figure S3A).

Our estimate of domain preservation by directly comparing annotations was limited by our rate of false negatives in our sparser maps. Consequently, we also assessed domain preservation by examining the distribution of corner scores for the GM12878 domain annotation in all the other cell types. We compared the result to the distribution of corner scores for a randomly permuted domain annotation (taking care to maintain the size distribution; see Section *IV.b*). We also compared the result to the distribution of corner scores for called domains in the other cell types (Figure S3B). We found that the median corner score for the GM12878 domains corresponded to the 95th percentile of corner scores at random domains in IMR90, the 96th percentile in HMEC, the 95th percentile in K562, the 95th percentile in NHEK, the 95th percentile in KBM7, the 95th percentile in HUVEC, and the 95th percentile in HeLa. Additionally, we found that, for the GM12878 domains, 58% had a corner score above the minimum corner score (thresholded at the 1st percentile) of called blocks in IMR90 (compared to 8% of random domains), 71% in HMEC (random: 12%), 56% in K562 (random: 7%), 56% in NHEK (random: 7%), 59% in KBM7 (random: 7%), 66% in HUVEC (random: 10%), and 62% in HeLa (random: 9%). Taken together, this suggests that the repertoire of domains is well conserved across cell lineages.

IV.c.5. Changes in the histone modifications of a domain correspond to changes in long-range contact pattern: In order to determine whether domains were preserved when the histone modifications marking the interval changed between cell types, we identified intervals associated with annotated domains in GM12878 where the median amount of H3K36me3 over that interval (binning the H3K36me3 signal track in 5 kb bins) changed by more than a factor of 1.5. We also required that in one cell type the interval be depleted for H3K36me3 (i.e., the median amount in the interval should be less than the genome-wide median) and that in the other cell type the interval must be enriched for H3K36me3 (i.e., the median amount in the interval should be greater than the genome-wide median). When we examined a cumulative distribution of the corner scores of these redecorated GM12878 domains in other cell types, we found that the distribution was shifted rightward from random: the median corner score corresponded to the 95th percentile of random domains in IMR90, the 95th percentile in HMEC, the 89th percentile in K562, and the 96th percentile in NHEK, indicating that even as the histone modifications decorating a domain change, the domain is often preserved.

Previous studies have suggested that there is a relationship between histone modification patterns and nuclear localization (Lieberman-Aiden et al., 2009). We sought to determine whether conserved domains that had different histone modification patterns in different cell types also behaved differently in terms of their long-range contact patterns. In order to test this, we identified 257 domains called in both GM12878 and IMR90 where the median amount of H3K36me3 over the interval changed by more than a factor of 1.5 between the two cell types (same as above). From this list of domains, we identified 153 boundaries where the amount of H3K36me3 in a 100 kb window just outside the domain was comparable in both cell types (either the median amount of H3K36me3 in the 100 kb window was at least 1.25-fold enriched over the genome-wide median in both cell types, or it was 1.25-fold depleted in both cell types). This constraint, coupled with the changing histone modifications over the domains between cell types, gave us a set of boundaries where in one of the cell types, there were comparable amounts of H3K36me3 on both sides of the boundary (a "concordant boundary") and in the other cell type, there was an enrichment of H3K36me3 on one side of the boundary and a depletion on the other (a "discordant boundary"). We hypothesized that if the decoration of histone modifications on a domain was associated with its pattern of long-range contacts, then contact patterns on either side of a discordant boundary would be decoupled, and the correlation of long-range contact patterns would be higher across a concordant boundary than across a discordant boundary. After identifying this set of conserved domain boundaries whose concordance/discordance status flipped between GM12878 and IMR90, we calculated the Spearman's correlation of the contact patterns for two 25 kb loci located either 100 kb upstream or downstream of the boundaries. (In order not to be confounded by the matrix sparsity at long distances, we analyzed only their contacts with other 25 kb loci located up to 5 Mb upstream or 5 Mb downstream of the boundary.) We observed that the correlations of contact patterns across discordant boundaries were significantly lower than the correlations of contact patterns across concordant boundaries (mean Spearman's r for concordant boundaries: .59; mean Spearman's r for discordant boundaries: .32; KS-statistic=.39, $p<10^{-10}$; see Figure S3C). Additionally, we visually examined many of the domains in question and confirmed that the change in histone modifications across the boundary often corresponded to a change in the long-range patterns of contacts (i.e., we saw a transition from the A-type long-range pattern to the B-type long range pattern, or vice versa, across the boundary) (see

Figure 2C, Figure S3D-E). Taken together, these data suggest that even while domains are conserved between cell types, the histone modifications marking them, and their corresponding spatial location within the nucleus, may change.

IV.c.6. Relation of Topological Domains to Domains: Dixon et al. (2012) reported that large, megabase-sized regions in mammalian genomes formed squares of enhanced contact frequency that tile the diagonal of Hi-C contact matrices. They called these large structures "topological domains" (also called "topologically associated domains" or "TADs"), and used the Directionality Index described above to create an algorithm for annotating such domains.

One important difference between TADs and the domains observed here is the size of the structures. The TADs annotated by Dixon et al. in IMR90 using their maps are much larger than the domains we annotate in IMR90 using our higher-resolution maps (median length = 280 kb) and roughly 5-fold larger than the domains we annotate in GM12878 using our highest-resolutions maps (median length = 185 kb), see Data S1.III. With our larger dataset, we also observe switching between subcompartments (roughly every 300 kb) and between the A and the B compartment (roughly every 400 kb) at a scale finer than TADs (See Data S1.III for examples showing differences between epigenetic marks and both short and long-range patterns in TADs versus domains.)

To confirm that the qualitative differences in our findings about domain formation were due to the fact that our maps contain higher resolution data, rather than being due to fundamental differences in our feature annotation algorithms, we re-ran the Dixon et al. domain annotation algorithm (DI HMM, the "Directionality Index Hidden Markov Model", available on the Ren lab website, http://bioinformatics-renlab.ucsd.edu/collaborations/sid/domaincall_software.zip) on our higher-resolution GM12878 *in situ* Hi-C map. Initially, we used all of the default parameter settings in their code. (Note that the algorithm on their website would often fail at the Cholesky decomposition step for large *M*, so we modified the script so that, if it did not reach the original default of *M*=20, it would run up until the maximum *M* at which the Cholesky decomposition failed.) The algorithm produced usable results for all chromosomes except 7, 8, 9, and 11, where the algorithm failed to complete successfully. The resulting domains had a median length of 473 kb, roughly half the size of the domains in the Dixon et al. annotation. Notably, the domains were now much smaller than the default domain length parameter used by the DI HMM (the default is 2 Mb). To eliminate this inconsistency, we changed the length parameter to 200 kb and re-ran the algorithm. After removing the chromosomes for which the algorithm failed to complete successfully (7,8,9,11,16,21,X), the resulting "DI/*in situ* Hi-C domains" had a median length of 265 kb, closely matching the results of the Arrowhead algorithm, and representing more than a 3-fold decrease in size from the domains reported in Dixon et al.

Crucially, although we find that domains annotated with the Arrowhead algorithm are more reliable than DI/*in situ* Hi-C domains, all of the major analyses in the paper gave similar results when we replaced the domains annotated using the Arrowhead algorithm with DI/*in situ* Hi-C domains. Thus, all the main results of our paper are robust to the choice of domain-annotation algorithm.

In comparing our findings, we noted that the major issue that arises when running the DI HMM on the lower-resolution Dixon et al. data is that the algorithm misses a significant number of domain boundaries, and consequently joins boundaries of distinct domains. This results in a large domain that is in fact the concatenation of multiple separate domains. As such, although our domain annotation and the domain annotation of Dixon et al. are very different, our findings and those of Dixon et al. agree regarding many of the statistical properties of domain boundaries, such as the fact that domain boundaries are enriched for CTCF. We note that several studies have reported domains smaller than TADs. These sub-TADs (median length: 500 kb) (Phillips-Cremins et al., 2013; Zuin et al., 2014) are larger than the domains we observe. As with TADs, this is likely due to our improved ability to resolve domain boundaries with respect to earlier maps.

We also note that our findings are consistent with the "physical domains" in *Drosophila* reported by Sexton et al. (Sexton et al., 2012). Like physical domains (whose median length is ~62 kb), the domains reported in GM12878 are short (185 kb); physical domains exhibit a consistent repertoire of epigenetic marks; and the specific combination of epigenetic marks is closely associated with the compartment to which the domain belongs.

V. Subcompartments in Hi-C Data

V.a. Clustering Methodology

V.a.1. Previous approaches towards clustering: The most common method used for classifying Hi-C patterns is the principal component (PC) approach, which we introduced in Lieberman-Aiden et al. (2009). In this approach, each intrachromosomal contact matrix is converted to an observed/expected matrix, and the first principal component of this matrix is used to bifurcate the data into two clusters. When this was performed at 1 Mb resolution on the original Hi-C maps, it was found that one cluster (A) was enriched for open chromatin marks, while the other cluster (B) was enriched for closed chromatin.

When we tried this approach on our higher resolution data, we found that it did not capture all of the patterns we saw, and in fact misclassified some B-type patterns. We therefore wondered if other methods might do a better job of matching the multiple distinct patterns that we observed.

Previous work in the field has studied alternatives and modifications to the 2-compartment model. Yaffe and Tanay (2011) performed a 3-pattern clustering of Hi-C data, applying k-means on the interchromosomal contact matrix, and found a third, gene-poor cluster. Imakaev et al. (2012) also examined possibilities outside the 2-compartment model; they found that the principal components of contact matrices vary continuously, and caution that classification into two clusters may be incomplete. With our larger maps, we sought to update the classification of contact patterns using the clustering method described below.

V.a.2. Clustering Algorithm: To cluster loci based on long-range contact patterns, we constructed a 100 kb resolution contact matrix C comprising a subset of the interchromosomal contact data. 100 kb loci on odd chromosomes appeared on the rows, and 100 kb loci from the even chromosomes appeared on the columns. The total length in base pairs of these two groups is roughly equal. (Chromosome X was excluded due to the differences in interaction pattern seen for the active and inactive homologs.) Thus $C_{i,j}$ represents the number of normalized contacts between the i-th locus on the odd chromosomes and the j-th locus on the even chromosomes. Genome-wide KR was used for normalization. Rows and columns for which more than 30% of the entries were either undefined or zeros were removed from the matrix. These bins were excluded from all further analyses involving the cluster tracks unless otherwise noted. We then took the logarithm of each entry in the C matrix.

To cluster loci on the odd chromosomes, we applied the z-score function from Python's *scipy* library to each row of C. We used the resulting matrix as input to the *scikit-learn* library's unsupervised Gaussian hidden Markov model clustering algorithm (GaussianHMM) (Pedregosa et al., 2011). We set the covariance type to diagonal and allowed 1000 iterations.

To perform clustering on loci located on the even chromosomes, we began by transposing the matrix C and then performed all steps exactly as they were performed for the clustering of odd chromosomes.

We found that each of the clusters on the odd chromosomes preferentially interacted with one of the clusters on the even chromosomes (Figure S4D). This defined a one-to-one mapping between the odd and even cluster annotations.

The result of a clustering algorithm typically depends on the choice of a parameter, k, which determines the number of clusters to be identified. We report the results of clustering using k=5 clusters; however, we also performed clustering using all values between k=2 and k=14 clusters as input. The Akaike Information Criterion and Bayes Information Criterion for the different cluster results clearly ruled out a value of k=2, and suggested a value of k between 4 and 8. Our final use of k=5 was based on this finding as well as careful examination of the data to determine how many clusters were necessary to explain the patterns that could be visually discerned. We found that, for k=5, the clusters corresponded to visually distinct patterns; this was no longer true if we increased k beyond 5. Nonetheless, it is possible that there are additional clusters that our algorithm could not identify; our results should be considered a lower bound on the number of subcompartments rather than an exact determination. Clustering via k-means and hierarchical clustering yielded similar results (Figure S4B).

V.a.3. Creating an A/B Pattern Annotation: To define the pattern of a cluster, we use the first derivative of interactions along the linear genome. More precisely, for each locus i on an odd chromosome, we obtain its 1-dimensional

interchromosomal interaction vector with all of the even chromosomes, C_i, and then calculate $d_i(j)=[C_i(j)-C_i(j-1)]$, where j and $j-1$ are adjacent loci on an even chromosome. The intuition for using such a measure is based in how we expect the interaction vector C_i for a given locus to change (or switch) when it exits one cluster and enters another cluster. When locus i is interacting with a stretch of loci (on an even chromosome) that are all in the same cluster, the derivative is close to zero, as the amount of interaction, whether high or low, does not change. However, at the border between two different clusters, when j and $j-1$ are in different clusters on the other chromosome, we expect $|d_i(j)|$ to be large. It is these switches that we use to determine cluster similarity. Using the derivative as a measure of pattern similarity is a simple way to account for the one-dimensional nature of the polymer. (This is akin to measures in finance that correlate returns of prices to identify similarities between stocks.)

To use this measure, we first create the difference matrix D by taking the difference between every adjacent pair of columns in the odd/even interchromosomal matrix C mentioned above. We then calculate a mean vector for each of the clusters (on the odd chromosomes) by averaging the rows of D for loci within the same cluster. Next, we examine the Spearman correlation of these mean derivative vectors for different pairs of clusters.

When examining the correlation matrix of the patterns, we found that the 5 patterns separate into two groups, with A1 and A2 in one group, and B1, B2 and B3 in the other. Patterns in a group correlate with each other and anticorrelate with patterns of the other group (Figure S4E). These correlations were confirmed by visually examining the five patterns. The first group of patterns correlated with the A compartment and the second group correlated with the B compartment.

V.a.4. An additional cluster on Chromosome 19: Careful visual inspection of the contact map for chromosome 19 revealed an additional, distinctive pattern which was visually apparent on a small set of loci but which was not found by the clustering algorithm. We suspected that this was because these loci occupy only 11 Mb, or 0.3% of the genome. We therefore created a matrix containing all interchromosomal contact data for chromosome 19, excluding the X chromosome. We processed the data as above, calculating the log of each entry and a z-score. (Rows and columns for which more than 50% of the entries were either undefined or zeros were removed from the matrix prior to normalization.) Finally, we used the same Gaussian HMM algorithm as before, using k=5.

One of the clusters the algorithm returned (labeled 19*) corresponded precisely to loci exhibiting the sixth pattern that we had noticed on visually inspecting the matrix. The other four chromosome 19-specific clusters (labeled 19-1 through 19-4) needed to be matched up to clusters found genome-wide in order to create a single classification. We did this by examining the frequency of interaction between the 19-specific clusters and the clusters previously labeled in the interchromosomal contact matrix. Cluster 19-1 interacted most frequently with loci in B2; Cluster 19-2 interacted most frequently with cluster B1; and Cluster 19-3 interacted most frequently with cluster A1. Cluster 19-4 interacted most frequently with A1, and next-most-frequently with B1. Since A1 had already been assigned to 19-3, and 19-4 exhibited H3K27me3 enrichment and H3K36me3 depletion, we labeled the 3.3 Mb that fell in 19-4 as B1.

To determine whether the loci in cluster 19* were in the A compartment or the B compartment, we compared the derivative of its intrachromosomal contact pattern to the derivatives of the subcompartments in the A and B patterns, which revealed a stronger correlation to compartment B patterns (Figure S4F). This was confirmed by visual inspection (see Figure 2F), and led us to label this cluster B4. It is worth noting that, interchromosomally, the B4 cluster pattern's derivative is more correlated with compartment A-type derivatives, and that loci exhibiting the B4 pattern also tend to possess both activating and repressive chromatin marks. These observations suggest that subcompartment B4 may be difficult to classify as either A or B, as it appears to have some degree of affinity for both.

V.b. Properties of Subcompartment Clusters

V.b.1. Clusters display distinct chromosomal and size distributions: There were differences in the distribution of clusters among individual chromosomes. For example, the larger chromosomes had much more B3 than B2, while the opposite was true for the smaller chromosomes (Figure S4G). The total number of megabases of the genome covered by each of the clusters is given in Figure S4C. Many adjacent 100 kb loci belong to the same cluster. We call a stretch of contiguous loci belonging to the same cluster a "cluster interval". The median size of cluster intervals is 300 kb. The mean and median size of cluster intervals for the different clusters is given in Figure S4C.

V.b.2 Clusters display unique patterns of epigenetic modifications: After clustering the data, we sought to determine enrichment of epigenetic signal tracks in the different clusters. To do so, we first binned the signals into 100 kb bins (taking the mean in each bin). For the enrichment analysis (Figure 2D), we calculated the median value of the signal track

in bins within the cluster of interest and divided that by the median value of the signal track across all bins. To determine how the signal tracks correlated with the clusters, we calculated the Spearman correlation coefficient between the binned signal track and a pseudo cluster track, where the pseudo cluster track had 1s at each 100 kb locus that belonged to the cluster of interest and -1s at all other loci (Figure S4I). We also performed the Wilcoxon rank-sum test, comparing the values of the mark in loci within the cluster of interest to the values of the mark in all loci of the remaining clusters, and the results were similar.

For the B4 cluster, we observed a simultaneous enrichment of H3K36me3 and H3K9me3, a seemingly paradoxical combination of activating and repressive marks. Additionally, we observed an enrichment of H3K9me3 over the A2 compartment when compared to the A1 compartment (both A1 and A2 contain active, open chromatin). In order to confirm that these results did not occur as a consequence of inappropriate crosstalk because of a faulty H3K9me3 antibody used in the ENCODE experiment we utilized, we repeated the H3K9me3 ChIP-Seq experiment seven times with three different antibodies from three different companies (see Section *I.b*). In all cases, the simultaneous enrichment of H3K9me3 and H3K36me3 over the B4 compartment and the enrichment of H3K9me3 on the A2 compartment compared to the A1 compartment was reproduced. In fact the simultaneous enrichment of H3K36me3 and H3K9me3 over KRAB-ZNF genes (which make up most of the B4 compartment) has been noted several times previously and reproduced by many groups (Vogel et al., 2006; Barski et al., 2007).

In Fig2D, we calculated enrichment of Nucleolar Associated Domains (NADs), which is encoded in a binary track, by taking the ratio of the length (in base pairs) of NADs found in the cluster and the length (in base pairs) of NADs found in the cluster when it was randomly permuted in the genome (Németh et al., 2010). For the supplemental figure, we examined the correlation with a continuous Nucleolus association track from van Koningsbruggen et al. (2010). Enrichment for nuclear periphery was evaluated by association of the clusters with the results of a Chip-Seq experiment from McCord et al. (2013) using lamin A/C, a protein known to localize at the nuclear periphery.

Pericentromeric chromatin was defined as the 2 Mb before and after each centromere, where the locations of the centromeres were given by the hg19 consensus annotation (http://genome.ucsc.edu/cgi-bin/hgTables). Of the 54.2 Mb of pericentromeric chromatin that was annotated by our clustering algorithm, 33.8 Mb of it (62.3%) is located within B2, a 3.8-fold increase from what would be expected at random. B1 contains 5.9 Mb, a 4% increase over expected. The other clusters are depleted for pericentromeric chromatin.

We also examined GM12878 Repli-Seq data, and found A1 and A2 were enriched for early-replicating chromatin, and B2 and B3 enriched for late-replicating chromatin, in agreement with Ryba et al. (2010) and Hansen et al. (2010). More specifically, we found that A1 and A2 both begin to replicate in the G1 phase; however, A1 tended to finish replicating by the S1 phase while A2 continued to replicate through the S2 phase. B2 and B3 do not begin replicating until the S3 phase and replicate primarily in the S4 and G2 phases. B1 begins in the G1 phase, but primarily replicates in the S1, S2, and S3 phases (Figure 2D).

V.b.3. Self-organization of a chromosome fragment in K562: We noticed that a 20 Mb stretch of chromatin on chromosome 9, from 0-21 Mb, has broken off of the rest of chromosome 9 in K562 cells. Notably, it is disconnected from the centromere. Nevertheless, the compartmental pattern seen at this locus in K562 cells closely resembles the pattern seen in GM12878, where the karyotype is intact. This suggests that compartmental structure is robust to chromosomal rearrangements, including breakage, and that compartments may self-assemble in the presence of appropriate chromatin marks.

VI. Peaks in Hi-C Data

VI.a. HiCCUPS: Identification of DNA Loops through Annotation of Focal Peaks

VI.a.1 Background and motivation: Much of the work on genome architecture so far has centered on the study of chromatin looping. In fact, the key question in the study of the three dimensional folding of the genome to date has been whether regulatory elements, and in particular promoters and enhancers, form DNA loops.

The existence of DNA loops was first demonstrated in the 1980s via studies of operons in prokaryotes and in phage (Schleif, 1992). These early studies used a variety of methods to demonstrate that DNA looping plays a role in transcription, replication, and recombination, (Dunn et al., 1984; Griffith et al., 1986; Eismann et al., 1987; Krämer et al., 1987; Mukherjee et al., 1988). These loops observed in prokaryotes were relatively small in size, on the order of hundreds of base pairs long.

The discovery of "enhancers" (Banerji et al., 1981), distal sequences that could affect transcription of their target genes at a distance (often separated by tens or hundreds of kilobases), prompted many to hypothesize about the existence of large chromatin loops in higher eukaryotes as a mechanism by which enhancers interacted with their target promoters (Ptashne, 1986). These hypothesized loops were conceptually similar to those observed in prokaryotes, but the loops themselves would have to be two to three orders of magnitude larger. The proposed size of these loops rendered many of the techniques utilized to demonstrate looping in prokaryotes inapplicable.

"Cyclization Enhancement" (Mukherjee et al., 1988) was an important exception: today, a family of proximity ligation methods borne out of the "cyclization enhancement" assay have become the most commonly-used approaches for testing hypotheses about DNA looping in eukaryotes. The frequency of contact between DNA loci *in situ* was first interrogated using the nuclear ligation assay (NLA) (Cullen et al., 1993), which couples proximity ligation in intact nuclei with locus-specific polymerase chain reaction in order to estimate the rate of contact between pairs of nearby loci. Chromosome conformation capture (3C) replaced this *in situ* ligation step with ligation in a large volume, and introduced the use of two primers instead of one in order to interrogate proximity relationships between arbitrary locus pairs (Dekker et al., 2002). Subsequent adaptations of 3C (4C, 5C) have increased the throughput of the 3C protocol (Dostie et al., 2006; Zhao et al., 2006). Recently, we introduced Hi-C (Lieberman-Aiden et al., 2009), which assays chromatin contacts genome-wide in an unbiased manner. In theory, one could use these technologies to systematically annotate all DNA loops across the genome, and as such, the advent of these technologies has spurred an enormous interest by many groups in decoding the cis-regulatory network of the genome via high-throughput identification of DNA loops.

Here, we bring our ultra-high resolution, genome-wide maps of chromatin contacts to bear on the problem of identifying chromatin loops. Systematic discovery of chromatin loops in high throughput proximity ligation data requires careful analysis of the data in light of several important potential pitfalls. Below, we discuss these issues, describing various algorithmic approaches to identifying chromatin loops in proximity ligation data.

VI.a.2. What does evidence for chromatin looping look like in a proximity ligation experiment?

Before detailing our peak-calling algorithm, we provide background on how chromatin loops in the nucleus manifest in proximity ligation.

VI.a.2.i. The definition of a chromatin loop implies that loops will manifest as local peaks in DNA-DNA proximity ligation data. In order to consider how to best identify chromatin loops systematically from a large Hi-C data set, it is worth considering what a chromatin loop ought to look like in a proximity ligation data set. To do this, it is helpful to consider how chromatin loops are defined. Here is a typical definition:

> "A chromatin loop occurs when stretches of genomic sequence that lie on the same chromosome (configured in cis) are in closer physical proximity to each other than to intervening sequences." (Kadauke and Blobel, 2009).

Indeed, the predominant definition of "chromatin loop" in the literature is as a structure in which two loci located in *cis* are closer, in 3D space, than intervening loci or other neighboring loci. This definition implies that chromatin loops manifest as local peaks in a proximity ligation dataset, which occur between two points whenever they interact with each other significantly more than with random points in their neighborhood. Note that if two points don't form a peak in the

contact map, then they show no greater interaction than expected for points in their local regions. Thus, they cannot be considered to form a loop.

Promoters, enhancers, and protein binding sites are all short – at most a few kilobases – suggesting that the hypothesized loops between them must be small, centered on a pair of loci that each span a few kilobases. (If the peaks were uniform over a width of, say, 100 kb, the corresponding loop cannot be said to be anchored at the enhancer any more than it can be said to be anchored at any other random site within the 100 kb locus.)

The idea that chromatin loops manifest as small local peaks in proximity ligation data is widely accepted in the literature.

VI.a.2.ii. Appearance of Peaks in Hi-C data: local enrichment vs. background. As we began to explore our *in situ* Hi-C data at increasingly high resolution (≤10 kb), we observed a large number of focal peaks. The typical peak reflects a two-to-five-fold increase in contact frequency at a peak pixel, and tends to decay in a roughly circular fashion across an "interaction region" in the heatmap. These focal peaks imply that two loci are in close proximity, but that this proximity relationship is not shared by intervening loci. As such, they indicate the presence of chromatin loops.

Note that, if individual rows of the *in situ* Hi-C data are viewed as a one-dimensional "quasi-3C" plot, each of the loops we observe exhibits a local peak at the expected position as well as perfect "reciprocity" between the two peak loci: if one peak locus is used as the index fragment, the other peak locus will be the site of the local, quasi-3C, peak.

Crucially, these focal peaks do not resemble, and cannot be explained by, any other local features visible in our heatmaps, such as compartment interaction patterns, or the interiors, edges, and corners of domain blocks.

We perform all of our computations on contact matrices, which we usually represent as 2D maps. However, one useful method for visualizing Hi-C maps, and peaks in particular, is to represent them as 3D objects. The X and Y coordinates of a 3D contact map give the location of any two loci, while the height (in the Z direction) gives the number of contacts between two loci. When contact data is examined in this manner, chromatin loops appear as sharp local apexes off of the diagonal (see Figure S5A; images were created using a custom pipeline that imported a 2D contact map, performed color inversion and Gaussian filtering, converted the image to an .stl file, and then rendered in Blender).

VI.a.2.iii. Section outline: We design and implement two completely new peak callers, designed for extremely high-resolution Hi-C maps, which systematically identify chromatin loops genome-wide by identifying pixels that are enriched above local background in a 2-dimensional contact map. One of these is meant to be a proof-of-principle, producing reliable results using an extremely simplified approach; the other is more complex, and further reduces the rate of false positives. We compare these new peak callers to a strategy in which peaks are identified by searching for enrichments over a global genome-wide average.

VI.a.3. Summary of peak calling algorithms: We implemented three different peak calling algorithms in this study: two are based on a local peak search in Hi-C data and one is based on the global enrichment methods used in other high-throughput studies.

Of our two local peak callers for Hi-C data, one is a simple method that calculates a single local expectation value for contacts at every pixel, compares the observed value to the expected value, and corrects for multiple hypotheses using the Benjamini-Hochberg FDR control procedure. This "BH-FDR" loop caller is sufficient for the purposes of generating all the main results of the paper, but it can produce false positives around contact matrix features such as the edges of domains.

The other, dubbed Hi-C Computational Unbiased Peak Search (HiCCUPS) is designed to reduce this rate of false positives and produce maximally reliable peak annotations. HiCCUPS calculates multiple local expectation values for every pixel in order to rule out the possibility that another local feature, such as the edge of a domain, could lead to a spurious, peak-like enrichment. HiCCUPS uses a modified Benjamini-Hochberg FDR control procedure, dubbed "λ-chunking," which is specifically designed to work with the unique statistical structure of Hi-C data and rigorously enforces thresholds of local enrichment.

While both methods give fundamentally similar results, HiCCUPS displays a significantly lower false positive rate. All peaks reported in the main text were annotated using HiCCUPS.

Finally, we implement a "Global-expected" peak calling method that searches for pixels enriched compared to a global expected that only takes into account one-dimensional genomic distance effects, in order to assess the methodology used in recent high-throughput peak annotation studies. The results of this algorithm are not used for any of the main analyses reported.

Detailed descriptions of each of the algorithms are provided below. Note that, as we point out above, all the loops we report in the main paper were generated using HiCCUPS. The point of BH-FDR is to demonstrate that these results do not depend on the use of complex statistical models, but can instead result (albeit with more noise) from an extremely simple loop-calling approach. The Global-expected peak caller is implemented only in order to better assess the confounds in peak calling on proximity ligation data.

VI.a.4. Simple local peak calling on Hi-C data: BH-FDR: Briefly, the BH-FDR peak caller calculates an expected value for each pixel and then tests the hypothesis – for each pixel – that it is significantly enriched over what one would predict based on the expected value. Since we are testing many pixels simultaneously, we employ the Benjamini-Hochberg false discovery rate (FDR) control procedure in order to correct for multiple tests, hence the term BH-FDR. Thus BH-FDR is an extremely simple approach to identifying pixels that are enriched above local background.

Note that, when running BH-FDR, we limit ourselves to examining pixels between loci ≤2 Mb apart, reasoning that we see almost no peaks larger than this size.

The use of this restriction relates to the motivation for the "λ-chunking" procedure used in HiCCUPS and described in Section *VI.a.5*.

VI.a.4.i. Local expected value calculation. In order to assess the local background and thus the level of contact frequency one would expect to see in a given pixel if a peak was not present, we examine contacts in a donut-shape around the pixel. This establishes a local baseline level of interaction that is seen between neighboring loci.

We begin with a normalized contact matrix M^* whose corresponding one dimensional expected matrix E^* has been calculated. (See Section *II.c.1* for details on the construction of E^*; E^* is used to account for the fact the not all pixels are the same distance from the diagonal, and is comparable to the "global background" correction used in 5C experiments. Here, we supplement it with a detailed local background model as well.)

For every pixel $M^*_{i,j}$ where $|i-j| \leq 2$ Mb, the local expected is calculated by sampling pixels in a donut surrounding $M^*_{i,j}$ as follows:

$$donut\ filter: \quad E^{d*}_{i,j} = \frac{\sum_{a=i-w}^{i+w}\sum_{b=j-w}^{j+w} M^*_{a,b} - \sum_{a=i-p}^{i+p}\sum_{b=j-p}^{j+p} M^*_{a,b} - \sum_{a=i-w}^{i-p-1} M^*_{a,j} - \sum_{a=i+p+1}^{i+w} M^*_{a,j} - \sum_{b=j-w}^{j-p-1} M^*_{i,b} - \sum_{b=j+p+1}^{j+w} M^*_{i,b}}{\sum_{a=i-w}^{i+w}\sum_{b=j-w}^{j+w} E^*_{a,b} - \sum_{a=i-p}^{i+p}\sum_{b=j-p}^{j+p} E^*_{a,b} - \sum_{a=i-w}^{i-p-1} E^*_{a,j} - \sum_{a=i+p+1}^{i+w} E^*_{a,j} - \sum_{b=j-w}^{j-p-1} E^*_{i,b} - \sum_{b=j+p+1}^{j+w} E^*_{i,b}} \times E^*_{i,j}$$

For a simple visual diagram of the pixels sampled by this local neighborhood, see Figure 3A. The $M^*_{a,b}$ terms in the numerator are counts from the normalized contact matrix. The remaining terms use the one-dimensional model E^* to correct for the fact that the sampled pixels and the central pixel $M^*_{i,j}$ may be at different distances from the diagonal.

The parameters p and w in the above equation specify the width of the interaction region surrounding the peak and the size of the donut sampled, respectively. Both p and an initial minimum w are given as inputs to the algorithm. Both parameters must be multiples of the matrix resolution. We always set p to correspond to a 20-25 kb distance (at 25 kb resolution, we set $p=1$; at 10 kb resolution, we set $p=2$; at 5 kb resolution, we set $p=4$) based on empirical observation of the size of true interaction regions. While an initial minimum w is provided as input to the algorithm, w is set individually for each pixel by starting at the minimum w, checking if the sum of pixels in the lower left region of M^* is ≥16 reads, and if not (suggesting an overly noisy estimate) incrementing w by 1. This is repeated until either the sum of the pixels is ≥16 reads or w=20. Our initial minimum w is set to 3 at 25kb resolution, to 5 at 10 kb resolution, and to 7 at 5 kb resolution. We only include pixels in the upper triangle of the contact matrix in our calculations, i.e. pixels where $j>i$. Thus any pixels inside the neighborhood window where $j \leq i$ are automatically excluded from the calculations. Furthermore, in order to ensure that sufficient numbers of pixels are included inside our local neighborhood, we only examine pixels that are between loci $>p+2$ apart (measured in units of resolution; i.e., at 25 kb resolution, p=1, and (1+2)*25 kb=3*25 kb implies that we only examine

pixels between loci ≥75 kb apart; at 10 kb resolution, we similarly examine pixels between loci ≥50 kb apart; at 5 kb resolution, we examine pixels between loci ≥35 kb apart).

The resulting expected value at i,j is called $E^{*d}_{i,j}$. Note that this is a KR normalized expected value, and will therefore not obey Poisson statistics. (A single contact can count for more or less than one contact after normalization.) To create an expected raw contact count – which will obey Poisson statistics – we must multiply by the appropriate KR coverage correction factors (C'_i and C'_j) for the pixel (Note that $C'_i = 1/C_i$ from section *II.b*). Thus the expected value becomes $E^{*d}_{i,j} \times C'_i \times C'_j$.

VI.a.4.ii. Statistical significance and multiple hypothesis testing. For each pixel, we test the hypothesis that the number of raw contacts seen at $M_{i,j}$ is significantly enriched relative to our model, which is that the number of contacts seen at $M_{i,j}$ is distributed according to a Poisson process with $\lambda = E^{*d}_{i,j} \times C'_i \times C'_j$.

In order to correct for multiple hypotheses, we use the standard Benjamini-Hochberg FDR control procedure (Benjamini and Hochberg, 1995). That is, we first calculate a p-value for each pixel based on the probability of seeing $M_{i,j}$ from a Poisson process with $\lambda = E^{*d}_{i,j} \times C'_i \times C'_j$. We then rank the p-values, $P_1, P_2, \ldots P_n$ and accept all hypotheses with rank less than i where i is the largest rank where $P_i \leq i^*\alpha/n$ (α is the desired FDR). After this multiple hypothesis correction, we are left with a list of pixels that are significantly enriched over the interactions in a local donut in their vicinity.

VI.a.4.iii. Filtering of Pixels Landing in Repetitive Regions. Repetitive regions either present in the reference assembly or unique to the specific karyotype in question can result in spurious peak calls because contacts between repetitive loci that are close in 1D can erroneously map to regions far away in genomic distance and result in massive enrichments far off the diagonal of the Hi-C contact matrix.

Repetitive elements that are incorrectly annotated as being present at only one locus in the reference assembly are difficult to correct for. However, repetitive regions that are annotated at multiple sites in the reference assembly can be handled in a relatively straightforward fashion: as noted above, we remove all reads aligning to the genome with MAPQ=0 in the computational processing of a Hi-C library.

However, this procedure leaves gaps in the contact matrix at pixels where one of the loci is highly repetitive. Similar gaps are seen when there is an underlying gap in the assembly. When calculating a local background, the highly sparse areas at or near such regions can lead to anomalous expected calculations and thus anomalous loop calls. We handled this by removing pixels from our enriched pixels list where one of the pixel loci landed within 5*(the matrix resolution) of a gap in the assembly or of an extremely sparse row (specifically, any row that was discarded by the KR algorithm before converging).

VI.a.4.iv. Clustering of nearby enriched pixels. Often, the BH-FDR algorithm called multiple nearby pixels "enriched." We collapsed each such cluster into a single peak call using a "greedy" algorithm.

We began by identifying the pixel with the highest number of observed reads in our list of enriched pixels after multiple hypothesis testing. We then iterated through the list, searching for any enriched pixels that were within a 20 kb Euclidean distance radius of our first pixel (the Euclidean distance between $M_{i,j}$ and $M_{i',j'}$ is defined as the locus size multiplied by $\sqrt{((i-i')^2+(j-j')^2)}$). If we found an enriched pixel within that radius, the pixel was removed from the list of enriched pixels, a centroid was calculated for our new cluster of two pixels, and a radius for the cluster, r, was also calculated. (The radius is defined as the Euclidean distance from the centroid to the furthest pixel in the cluster.) The process was then repeated, this time looking for any pixels in the enriched pixel list that were within 20 kb+r of the cluster centroid. Any time a peak was found within the radius, the pixel was removed from the list, and a new centroid and radius were calculated for the peak cluster. The process was repeated until no further pixels could be added. At that point, we annotated the initial pixel with the highest value as the "peak pixel", and recorded the centroid and radius resulting from the algorithm as the centroid and radius of the interaction region associated with the pixel. We then went back to the remaining list of enriched pixels, once again chose the pixel with the highest value, and repeated the process. Eventually no pixels were left, all enriched pixels having been assigned to a cluster. The output of the BH-FDR peak caller was a list of "peak pixels" and the associated cluster centroids and radii.

Note that the initial 20 kb radius for the clustering was empirically chosen, but initial radii up to 50 kb were also tested with minimal impact on the final numbers of peak pixels.

The results of the BH-FDR algorithm do not appear in the main text of the paper, but are detailed in section *VI.b.3* of the Extended Experimental Procedures.

VI.a.5. HiCCUPS (Hi-C Computational Unbiased Peak Search): While BH-FDR did a reasonable job of identifying the focal peaks we observed in our data and its output was of sufficiently high quality to obtain all the main results of the paper, we noticed that the algorithm made several types of errors which could be resolved using a more nuanced approach.

BH-FDR often annotated peaks erroneously when in fact the enrichment present was due to certain types of larger-scale but locally asymmetric features that could not be effectively filtered out by "donut" sampling. A simple example is the presence of false positives along the edges of domains. False positives of this type could easily be resolved by examining neighborhoods to the left and right of a target pixel, as well as neighborhoods above and below a target pixel. Such neighborhoods would be equally enriched if the effect at a pixel was due to its presence on a domain edge.

We also noted that we could not effectively probe pixels at arbitrary distances, and instead had to limit ourselves to pixels near the diagonal (≤2 Mb away). This was not a significant problem in practice, because it was visually apparent that most peaks lie close to the diagonal. Still, this issue reflected a basic problem in the applicability of the Benjamini-Hochberg FDR control procedure in a setting like ours. With no maximum distance restriction, the trillions of nearly empty (Poisson mean and variance ~ 0) long-range pixels confounded the BH procedure. Occasionally, a read pair would land in such a bin; in aggregate, a vast number of read pairs landed in such bins. Because such bins had such a tiny mean and variance, a very stringent FDR threshold had to be used in order to avoid erroneously annotating them as peaks. However, such a stringent FDR threshold led to no peaks being called near the diagonal, where the Poisson means could be in the 100s and the variance was large. As a result, a given choice of FDR threshold called either nonsensical peaks far from the diagonal, or no peaks at all. This problem could be resolved by binning pixels into "hypothesis classes" based on their expected value, a procedure we call "λ-chunking." The Benjamini-Hochberg FDR procedure is separately applied to each λ-chunk, so the millions of high-mean bins are treated differently than the trillions of low-mean bins.

(Note that the underlying issue is the fact that we have a vast number of hypotheses with dramatically different means; that these means are spread out along the positive real number line, with no large gaps; and the number of hypotheses blows up as the hypothesis mean declines. To our knowledge, adequately addressing such a setting requires non-standard methods for multiple hypothesis correction. "λ-chunking," described below, is a simple approach we have developed for addressing this problem, but better approaches are needed.)

On the basis of these observations, we decided to implement a more sophisticated local peak caller, dubbed HiCCUPS (Hi-C Computational Unbiased Peak Search), that would integrate information from multiple local neighborhoods in its peak calling procedure, and that would handle multiple hypothesis testing in a more nuanced fashion. HiCCUPS applies a modified Benjamini-Hochberg FDR procedure separately to each λ-chunk, and also samples multiple local neighborhoods instead of just one.

VI.a.5.i. Additional local neighborhoods: We realized that the donut neighborhood described in Section *VI.a.4.i* led to a poor expected model when the target pixel was near the perimeter of a large-scale feature, as explained above.

To remedy this, we added three additional neighborhoods of various shapes to our procedure. These neighborhoods also serve as "filters": by requiring certain enrichment thresholds with respect to a particular neighborhood, we can filter out certain types of enrichment artifacts. We note that we only sample pixels in the upper half of the symmetric contact matrix, so for example an area to the "lower left" of the pixel represent contacts between loci closer together.

The lower left neighborhood, which is just the lower left quadrant of the donut neighborhood, was designed to ensure that pixels inside domains were not erroneously identified as peaks. This neighborhood takes advantage of the fact that if pixel $M_{i,j}$ is located in the interior of a domain, then all the pixels located to its lower left (i.e. pixels $M_{i',j'}$ such that $i'>i$ and $j'<j$) will also be located inside of the domain. As such this neighborhood gives a more accurate assessment of the local background for pixels contained within domains.

Quantitatively, the lower-left neighborhood is defined as follows:

$$lower\ left\ filter: \quad E^{ll*}_{i,j} = \frac{\sum_{a=i+1}^{i+w}\sum_{b=j-w}^{j-1} M^*_{a,b} - \sum_{a=i+1}^{i+p}\sum_{b=j-p}^{j-1} M^*_{a,b}}{\sum_{a=i+1}^{i+w}\sum_{b=j-w}^{j-1} E^*_{a,b} - \sum_{a=i+1}^{i+p}\sum_{b=j-p}^{j-1} E^*_{a,b}} \times E^*_{i,j}$$

See Figure 3A for a visual diagram. Again, this equation reflects a sum of all values at pixels in the lower-left neighborhood, corrected for the difference in distance from the diagonal between the neighborhood pixel and the target pixel *i,j*. (See Section *VI.a.4.i.*)

The vertical and horizontal neighborhoods are intended to ensure that peaks are not erroneously called on the edges of domains. These are defined as follows:

$$vertical\ filter: \quad E^{v*}_{i,j} = \frac{\sum_{a=i-w}^{i-p-1}\sum_{b=j-1}^{j+1} M^*_{a,b} - \sum_{a=i+p+1}^{i+w}\sum_{b=j-1}^{j+1} M^*_{a,b}}{\sum_{a=i-w}^{i-p-1}\sum_{b=j-1}^{j+1} E^*_{a,b} - \sum_{a=i+p+1}^{i+w}\sum_{b=j-1}^{j+1} E^*_{a,b}} \times E^*_{i,j}$$

$$horizontal\ filter: \quad E^{h*}_{i,j} = \frac{\sum_{b=j-w}^{j-p-1}\sum_{a=i-1}^{i+1} M^*_{a,b} - \sum_{b=j+p+1}^{j+w}\sum_{a=i-1}^{i+1} M^*_{a,b}}{\sum_{b=j-w}^{j-p-1}\sum_{a=i-1}^{i+1} E^*_{a,b} - \sum_{b=j+p+1}^{j+w}\sum_{a=i-1}^{i+1} E^*_{a,b}} \times E^*_{i,j}$$

See Figure 3A for a visual diagram.

The parameters *p* and *w* in the above three neighborhoods are defined and treated in exactly the same way as we described for the donut neighborhood above (see Section *VI.a.4.i.*).

As with the donut neighborhood procedure described above (Section *VI.a.4.i.)*, we rescaled our calculated local expectations by the KR coverage factors for the target pixel in order to obtain a value for the number of raw contacts we expect to see; this is, as noted above, essential for the applicability of Poisson statistics. We next test the hypothesis that the number of contacts seen in $M_{i,j}$ is significantly enriched with respect to the expected value for a given neighborhood, i.e., with respect to a Poisson process whose parameter $\lambda = E^{*local}_{i,j} \times C'_i \times C'_j$. For HiCCUPS to call a pixel "enriched," we require that it be enriched with respect to all four neighborhoods.

VI.a.5.ii. Multiple hypothesis testing: In order to perform multiple hypothesis correction while probing all intra-chromosomal pixels (rather than restricting to pixels between loci ≤ 2 Mb apart), we had to employ a different FDR control strategy. As noted above, while the Benjamini-Hochberg FDR control procedure was adequate for controlling the FDR when restricting to pixels between loci ≤ 2 Mb apart, it did not perform well after including all intra-chromosomal pixels. A basic issue is that the Benjamini-Hochberg procedure and other commonly used FDR control procedures are designed to control the FDR given that the hypotheses that are being tested are independent and identically distributed. However, in a Hi-C data set, this requirement is very dramatically violated. Due to the strong genomic distance dependence of contact frequency, testing whether every pixel is a peak involves testing hypotheses where the expectations range by nearly 6 orders of magnitude (from thousands of contacts expected in some pixels, to thousandths of contacts expected in others, all at 10 kb resolution in a single contact matrix). Additionally, Hi-C data has the property that there are orders of magnitude more low expectation hypotheses than high expectation hypotheses, because there are far more possible long-range interactions (i.e. far off the diagonal of the contact matrix) than potential interactions between closely spaced loci (i.e. close to the diagonal). We found that, because there are millions of low-expectation pixels for every high-expectation pixel, at the FDR thresholds at which we observe many easily apparent peaks (2-5 fold enrichment over an expectation of tens of reads), we see many low-expectation pixels with similar p-values arising by chance (2 or more reads over an expectation of thousandths of reads). To get rid of the spurious low-expectation pixels, the FDR must be set so high that no pixels are called at all. As such, FDR control procedures like Benjamini-Hochberg, where all hypotheses are lumped together, are not suitable for our data. (Although some widely used FDR control procedures exist that allow the independent-and-identically-distributed assumptions to be relaxed, these cannot address the dramatic range of hypotheses seen in Hi-C loop calling procedures.)

To overcome these challenges, we developed a method in which pixels are assigned to hypothesis families together with other pixels whose expected contact frequency – based on local background – is similar. We call this procedure "λ-chunking." Each pixel is assigned to a λ-chunk based on its expected value, *E*. All pixels with $E<1$ were placed in one bin. Subsequent bins were logarithmically spaced every $2^{1/3}$. Thus, for each type of expected value *E*, bin 1 contained all pixels with $E < 1$, bin 2 contained all pixels where $1 < E <2^{1/3}$, bin 3 contained all pixels where $2^{1/3}< E <2^{2/3}$, and so on. We then perform an FDR procedure separately for each λ-chunk. Within each λ-chunk, the distribution of observed counts over all pixels was compared to a null Poisson distribution with λ equal to the maximum expectation assigned to the bin (i.e. for bin 1, we compared to a null Poisson distribution with $\lambda=1$; for bin 2, we compared to a null Poisson distribution with $\lambda=2^{1/3}$, and so on). An FDR threshold corresponding to 10% FDR was identified by finding the minimum value *t* such that the integral of the null Poisson distribution from *t* to ∞ was less than 0.1 times the integral of the distribution of observed counts from *t* to ∞ This procedure is essentially equivalent to applying the Benjamini-Hochberg FDR control procedure at an FDR rate of $\alpha=.1$ on each λ-chunk independently and then combining the results (albeit with a slightly modified calculation of the p-values in each λ-chunk).

After identifying the FDR thresholds for each bin separately and repeating this process separately for each local expected filter, pixels were identified as locally enriched if the number of contacts in the pixel was greater than the 10% FDR threshold on each of their four local expected values (i.e., the values obtained using the four neighborhoods). Because this enrichment was assessed for each neighborhood separately, the resulting procedure is stringent and markedly improves peak-call reliability.

Empirical validations of this FDR control procedure are provided in Section *VI.b.3* and in Table S4.

VI.a.5.iii. Additional filtering of pixels landing in repetitive regions: Pixels landing in or near repetitive regions and assembly gaps are filtered using the same method described in Section *VI.a.4.iii*.

VI.a.5.iv. Clustering of nearby enriched pixels: Often, the HiCCUPS algorithm called multiple nearby pixels "enriched." We collapsed each such cluster into a single peak call using the same "greedy" algorithm described in Section *VI.a.4.iv*.

VI.a.5.v. Additional filtering of peak pixels based on local enrichment thresholds: In order to be extremely stringent in our peak calls, we took pixels that were enriched with respect to all four local neighborhoods using the above FDR procedure and filtered them further. In this additional filtering step, we removed all pixels that did not show a minimum fold-enrichment with respect to the expected values for each of the four neighborhoods. Thus, even pixels that showed statistically significant enrichment after multiple hypothesis testing with respect to all four neighborhoods were excluded if they did not further show sufficient fold enrichment with respect to all four neighborhoods.

More specifically, we required that every pixel in our final annotation of peak pixels be enriched by at least 50% over the horizontal and vertical expected values, and at least 75% over the lower-left and donut expected values. Finally, we required that each pixel was at least 2-fold enriched above either the donut expected value or the lower-left expected value.

For our replicate GM12878 experiment peak annotation at 10 kb resolution, we required that peak pixels were enriched by at least 50% over the horizontal and vertical expected values, by at least 50% over both the lower-left and donut expected values, and by at least 75% for either the lower-left or donut expected values. The change was made in this single case in order to better compare the results of both our primary and replicate experiments by calling a similar number of peaks in both experiments.

We determined these enrichment thresholds based on visual assessment of false positive and false negative rates; removing these thresholds entirely does not substantially increase the number of peaks called.

Details about the robustness of the peak annotations to the choice of local enrichment thresholds are provided in Section *VI.b.2*.

VI.a.5.vi. Additional filtering of "singlet clusters." We noticed that when the HiCCUPS algorithm called a pixel that was not part of a cluster of nearby pixels, that pixel was usually a false positive, i.e., true focal peaks typically lead to enrichment of multiple nearby pixels.

This was especially true if they had a higher q-value (>.01, q-value is defined as the minimum FDR threshold that the peak would be called at, i.e. the ratio of the integral of the expected distribution from the peak value to infinity to the

integral of the observed distribution from the peak value to infinity) on at least one of the local filters. As such, we filtered out peak pixels if there were no other enriched pixels that were collapsed into it and the sum of its four q-values (for the four local filters) was >.02 (for the primary 10 kb peak annotation and replicate 10 kb peak annotation, >.04 and >.08 thresholds were used respectively; these lists were only used to assess biological reproducibility of our peak annotations).

VI.a.5.vii. Combining peak annotations at different resolutions. In order to localize peaks as effectively as possible, but at the same time to annotate as many peaks as possible, we applied HiCCUPS at several resolutions in each cell type. Depending on the strength of a peak and the sequencing depth of the map, some peaks might reach statistical significance at 5 kb resolution, while others might only reach statistical significance at 10 kb resolution or 25 kb resolution.

As such, we developed a method to combine peak lists generated at various resolutions, in a way that does not double count the same peak called at different resolutions. When combining peak lists at different resolutions, we always accepted the highest resolution version of a peak. Thus, if two peaks were within 20 kb of each other (or within 50 kb if one was a 25 kb peak), we accepted the higher resolution peak and discarded the lower resolution peak.

We found that the failure to call a peak at coarse resolution that was seen at fine resolution was in certain cases a warning sign of a false positive. Specifically, we threw out peaks between loci >100 kb apart that were called at 5 kb resolution but not at 10 or 25 kb resolution, because we found that such peaks were enriched for false positives. (Note that these long-distance “orphan” 5 kb peaks were usually no more than 5% of peaks called at 5 kb).

This procedure produced the final combined peak annotations that were used for all the peak analyses reported in the main text of the paper.

Notably, while the peak annotations we used for all analyses reported in the paper included only peaks called at up to 25 kb resolution, the local peak calling approaches outlined in this section and in Section *VI.a.4* can easily be applied at arbitrary resolutions. However, at lower resolutions, it becomes more difficult to identify meaningful biological features among the output; typically, peaks annotated at 50 kb and 100 kb and not at finer resolutions correspond to domains or compartment structures rather than an independent focal feature such as a loop. Specifically, we called peaks at 50 kb resolution in GM12878 and in the vast majority of cases did not observe any structures that did not correspond to either a compartmental effect, a domain, or a loop that could also be discerned at higher resolution. As such, we did not include the results of peak calling at lower than 25 kb resolution in any of our analyses and discussions. The sole exception to this was the network of superloops on chrX; we used HiCCUPS with slightly modified parameters at 50 kb resolution to identify this family of superloops in our haploid GM12878 and IMR90 maps.

HiCCUPS can easily be applied to fragment-delimited contact matrices rather than base pair-delimited contact matrices. We called peaks using HiCCUPS on our primary and replicate combined GM12878 map at 20 fragment resolution; the results presented in the paper are unchanged. This analysis is discussed in greater detail in Section *VI.b.4*.

VI.a.6. Peak calling using a global expectation: In order to assess the results of performing peak calling against a global genome-wide expectation rather than to a local background, we implemented our own peak calling algorithm that compared each pixel to its genome-wide expected, E^*, based purely on one-dimensional distance (see Section *II.c.1* for details on the calculation of the one-dimensional expected). We performed multiple hypothesis testing as in Section *VI.a.5.ii*.

The results of this analysis do not appear in the main text of the paper, but are detailed in section *VI.b.6*. of the Extended Experimental Procedures.

VI.a.7 Computational considerations: Searching all intrachromosomal pixels for local peaks at 10 kb resolution requires surveying roughly 10 billion pixels (at 5 kb resolution, the number is 40 billion). This is a computationally intensive but highly parallelizable process. We therefore recoded HiCCUPS (and the other peak calling algorithms detailed above) using CUDA in order to perform these local expected calculations on a workstation containing 4 NVIDIA GPUs. Using an NVIDIA Tesla C2075 GPU (which contains 448 cores and is capable of highly parallelized computation), we obtain a 200-fold speedup over the CPU implementation. HiCCUPS is also compatible with NVIDIA’s faster Kepler architecture; we tested HiCCUPS on a Tesla K40c GPU. In general we have found that extensive genome-wide peak-calling on CPU architectures is extremely difficult and time-consuming.

VI.b. HiCCUPS Validation

HiCCUPS was used to annotate 8054 peaks at 10 kb matrix resolution in our primary GM12878 map, 7484 peaks at 10 kb matrix resolution in our replicate map, 2677 peaks at 10 kb resolution in a replicate *in situ* Hi-C map constructed using the DpnII restriction enzyme, and 3073 peaks at 25 kb matrix resolution in our dilution Hi-C GM12878 map. These peak annotations were used to assess the biological reproducibility of our peaks as well as the reliability of HiCCUPS.

Additionally, HiCCUPS was used to annotate 9448 peaks in our primary and replicate combined GM12878 map, 8040 peaks in IMR90, 5152 peaks in HMEC, 4929 peaks in NHEK, 6057 peaks in K562, 2634 peaks in KBM7, 3865 peaks in HUVEC, 3094 peaks in HeLa and 3331 peaks in CH-12 mouse lymphoblasts. Peaks were called at 5 kb resolution and 10 kb resolution in all cell types and additionally at 25 kb resolution in NHEK, KBM7, K562, HUVEC, and HeLa. The peak lists returned by HiCCUPS at various resolutions for a single cell type were combined into a single list following the procedure outlined above (see Section *VI.a.5.vii*). In CH-12 mouse lymphoblasts, the final peak list used only included peaks called at 10 kb resolution and greater than 100 kb between peak loci, as short-range peak calling was not effective.

For all peak annotations, the input data fed to HiCCUPS was the MAPQ ≥ 30 filtered data, in order to be stringent about avoiding false positives due to bad alignments. In Figure 3A and Data S1.V, the contact matrices shown contain the MAPQ ≥ 30 filtered data, i.e. the data used to annotate the calls is shown.

VI.b.1. Peaks are biologically reproducible and reliably annotated between replicates: To check the biological reproducibility of our 10 kb resolution peak calls between our primary GM12878 *in situ* Hi-C experiment and our replicate GM12878 *in situ* Hi-C experiment, we compared the two annotations and called a peak pixel in one annotation, $M_{i,j}$, reproducible if there existed a peak pixel in the second annotation, $M_{i',j'}$, such that the Euclidean distance between the two, $\sqrt{((i-i')^2+(j-j')^2)}$, was less than min(0.2*|*i*-*j*|, 50 kb). We found that 5403 peaks appeared in both the primary and replicate peak annotations (out of 8054 total primary peaks and 7484 total replicate peaks). We varied the maximum absolute Euclidean distance allowed between two peaks for reproducibility between 0 kb and 100 kb, and varied the maximum relative Euclidean distance as f*|*i*-*j*|, with $0 \leq f \leq 1$, and confirmed that there was only modest effect on the percentage of peaks called as reproducible. Example peaks are shown in Data S1.V; data and annotations from both our primary and our replicate experiments are shown side by side.

Similarly, we compared the overlap between our dilution Hi-C peak list and our primary *in situ* Hi-C peak list, as well as the overlap between our DpnII *in situ* Hi-C peak list and our primary *in situ* Hi-C peak list, once again using the min(0.2*|*i*-*j*|, 50 kb) Euclidean distance radius. Similar results were observed (1990 out of 3073 dilution Hi-C peaks overlapped a peak annotated in the primary *in situ* Hi-C annotation, 2097 out of 2677 DpnII *in situ* Hi-C peaks overlapped a peak annotated in the primary *in situ* Hi-C annotation).

Additionally, we examined the local enrichments of all peak pixels from our primary experiment annotation in our replicate map and vice versa. We found that 92% of the primary peak pixels were at least 1.3-fold enriched relative to all local neighborhoods in the replicate map and 92% of the replicate peak pixels were at least 1.3-fold enriched relative to all local neighborhoods in the primary map.

Since there were high overlaps between our different replicates, we used the peak calls derived from our summed primary and replicate *in situ* Hi-C MboI maps for all subsequent analyses on GM12878 loops, in order to increase the resolution of our loop calls and ensure that we called as many peaks as possible.

VI.b.2. Sensitivity and specificity of HiCCUPS: We judged the false positive and false negative rates of HiCCUPS by comparing to a hand annotation of peaks in the 14 manually chosen ENCODE regions used in Sanyal et al. (2012). We first visually inspected each of the 14 ENCODE regions in our GM12878 primary and replicate combined *in situ* map, labeling all peaks that could be seen visually. We identified a total of 53 peaks throughout the 14 regions. We then counted how many of the hand annotated peaks overlapped peaks called by HiCCUPS. We also varied the local enrichment thresholds used to filter peaks (see Section *VI.a.5.v.*) in order to assess the robustness of HiCCUPS to various parameter choices.

Using the local enrichment thresholds described in Section *VI.a.5.v* (>1.5-fold enrichment over all four local expectations, >1.75-fold enrichment over both the donut and lower left expectations, and >2-fold enrichment over either the donut or the lower left expectations), we annotate 9,448 peaks in GM12878. 51 of these peaks fall within the 14 manual ENCODE regions. Of those 51, 37 overlap a hand annotated peak, suggesting an empirical false positive rate of 27% and a false negative rate of 30%.

If we relax the local enrichment thresholds to only require >1.5-fold enrichment over all four local expectations, we annotate 12,178 peaks in GM12878. 67 of these peaks fall within the 14 manual ENCODE regions. Of those 67, 41 overlap a hand annotated peak, suggesting an empirical false positive rate of 39% and a false negative rate of 23%.

If we enforce no additional local enrichment thresholds, we annotate 15,142 peaks in GM12878. 96 of these peaks fall within the ENCODE regions. Of those 96, 46 overlap a hand annotated peak, suggesting an empirical false positive rate of 52% and a false negative rate of 13%.

See Table S4 for a region-by-region breakdown of overlaps for each of the above lists. While HiCCUPS is robust to parameter choice, we believe that the parameter choice described in Section *VI.a.5.v* best balances the false positive and false negative rates.

VI.b.3. Simplified local peak caller recapitulates results of HiCCUPS: We further assessed the validity of HiCCUPS by comparing it to a simplified local peak caller (see Section *VI.a.4*). This simple local peak caller utilizes only one local filter (the donut filter) and corrects for multiple hypotheses using the classic Benjamini-Hochberg FDR control procedure (only pixels ≤ 2 Mb are examined). We find extremely strong correspondence between the annotation returned by this completely different algorithm and HiCCUPS. The simplified local peak caller annotates 25,776 peaks at 10 kb resolution in GM12878; of those 25,776 peaks, 6,984 (27%) overlap with a peak in our primary and replicate combined GM12878 annotation returned by HiCCUPS. Additionally, we observe that 58% of peak loci contain a CTCF binding site, that CTCF is 1.6-fold enriched at peak loci relative to other proteins, and that the peak annotation is 5.2-fold enriched for peaks occurring at the corners of domains. The strong correspondence of the results obtained using the simplified local peak caller and the results obtained using HiCCUPS are obvious despite the high false positive rate for the simplified local peak caller (see Table S4).

If we enforce a local enrichment threshold on the peaks annotated by the simplified local peak caller, requiring them to be >2-fold enriched over the donut expectation (similar to what we require for HiCCUPS), we see that the correspondence of results is even stronger. 5096 out of 8446 peaks (60%) overlap with a peak in our primary and replicate combined GM12878 annotation returned by HiCCUPS. Additionally, we observe that 75% of peak loci contain a CTCF binding site, that CTCF is 2-fold enriched at peak loci relative to other proteins, and that the peak annotation is 8.3-fold enriched for peaks occurring at the corners of domains.

However, it is still obvious that the simplified local peak caller suffers more from both a higher false positive and false negative rate than does HiCCUPS (see Table S4). In particular, we observe more pixels located in the interior and edges of domains annotated as peaks due to the lack of appropriate filters to correct for these structures. As such, we believe that HiCCUPS provides the best framework for annotating peaks genome-wide.

VI.b.4. HiCCUPS can also be performed at fragment resolutions: All of the annotations we present in the main text of this study were performed on fixed-width delimited contact matrices. However, peaks can easily be annotated on fragment-delimited contact matrices. We annotated peaks at 20 MboI fragment resolution (~10 kb resolution) in our primary and replicate combined GM12878 map. The peak annotation obtained via HiCCUPS was consistent with all results presented in this study. In brief, we annotated 7585 peaks, of which 6175 (81%) overlapped a peak called in our fixed width annotation. The 7585 peaks collapsed to 10706 peak loci, 88% of which contained a CTCF binding site. 42% of the peaks overlapped the corner of a domain (35% of domains contained a peak in the corner). We conducted all analyses in this paper on fixed-width contact matrices simply for ease of interpretation.

VI.b.5 Saturation analysis of HiCCUPS peak calls: In order to assess what percentage of the total loops present in a given cell type we were able to detect with our highest resolution maps, we performed a saturation analysis of peak calls returned by HiCCUPS on chr20 in GM12878. We subsampled our main primary+replicate GM12878 map at various percentage thresholds and called loops on chr20 using same parameters as used above (section *VI.a.5*). We found that the number of peaks called on chr20 converged very strongly: ~50% of peaks were called with 20% of the data, ~75% of peaks with 40% of the data, ~87% of peaks with 60% of the data, and ~93% of peaks with 80% of the data (see Figure S5C). As such, further higher-coverage maps than the ones we report in GM12878 are unlikely to reveal substantially more loops than we observe here.

VI.b.6. Comparison with External Loop Annotations: A 5C-based study in GM12878, reported by ENCODE, annotated 1187 putative peaks across 1% of the human genome (Sanyal et al., 2012); this corresponds to an average of 40 putative peaks per megabase. A CHIA-PET experiment in K562 cells, also reported by ENCODE, observed 126,886 putative

peaks anchored at PolII sites, suggesting 40 peaks/Mb mediated by PolII (Li et al., 2012). Jin et al. (2013) used Hi-C in IMR90 and reported 1,116,312 putative peaks, corresponding to 356 peaks/Mb.

In this study, we find 9448 loops genome-wide, corresponding to 3/Mb in GM12878. Similarly, our K562 map shows 1/Mb and our IMR90 map shows 2/Mb. These loops were identified by comparing the contact frequency at a pixel to the local background.

As a comparison, we also implemented a peak calling algorithm that compared each pixel to its genome-wide expected (based purely on one-dimensional distance; see Section *II.c.1*). We required that every peak call be at least 2-fold enriched and applied a 1% FDR threshold. (Multiple hypothesis testing was done as in HiCCUPS.) The algorithm identified 2,029,252 "pseudo" peaks. A comparison of all of the peak lists mentioned above can be found in Figure S8, Data S2, and Table S6.

Our HiCCUPS loop lists show strong concordance with recent ChIA-PET experiments using CTCF or RAD21 as the bait protein. Li et al. (2012) called 21306 peaks in K562 using CTCF ChIA-PET, while we called 6057 peaks in K562 using in situ Hi-C. 8742 (41%) of their peaks overlapped 4241 (70%) of our peaks (p-value < 10^{-13311}, hypergeometric distribution). Heidari et al. (2014), called 11134 peaks in K562 using RAD21 ChIA-PET, while we called 6057 peaks in K562 using in situ Hi-C. 5000 (45%) of their peaks overlapped 2884 (48%) of our peaks (p-value < 10^{-8914}, hypergeometric distribution). Heidari et al. (2014) called 11363 peaks in GM12878 using RAD21 ChIA-PET, while we called 9448 peaks in GM12878 using in situ Hi-C. 5394 (62%) of their peaks overlapped 3924 (42%) of our peaks (p-value < 10^{-11860}, hypergeometric distribution).

VI.c. 3D DNA FISH

VI.c.1. 3D DNA FISH confirms that peak loci are more likely to be spatially proximate: In Peak 1, locus L2 is situated on chr 17 between 67.22 and 67.25 Mb, and it forms a peak with L1, 460 kb upstream. L3, an equal distance downstream, does not form a peak with L2 and served as a control. In 63 of 101 loci (62%), the distance between the peak loci, L1 and L2, was significantly shorter (by at least 0.2 μm) than the distance between L3 and L2. The distance between L3 and L2 was significantly shorter than the distance between L1 and L2 in 11 of 101 cases (11%).

In Peak 2, locus L2 is situated on chr14 between 72.20 and 72.23 Mb, and it forms a peak with L1, 600 kb upstream. L3, an equal distance downstream, does not form a peak with L2 and served as a control. In 35 of 83 loci (42%), the distance between the peak loci, L1 and L2, was significantly shorter (by at least 0.2 μm) than the distance between L3 and L2. The distance between L3 and L2 was significantly shorter than the distance between L1 and L2 in 10 of 83 cases (12%).

In Peak 3, locus L2 is situated on chr11 between 130.29 and 130.32 Mb, and it forms a peak with L1, 430 kb downstream. L3, an equal distance upstream, does not form a peak with L2 and served as a control. In 58 of 103 loci (56%), the distance between the peak loci, L1 and L2, was significantly shorter (by at least 0.2 μm) than the distance between L3 and L2. The distance between L3 and L2 was significantly shorter than the distance between L1 and L2 in 9 of 103 cases (9%).

In Peak 4, locus L2 is situated on chr13 between 85.46 and 85.49 Mb, and it forms a peak with L1, 910 kb downstream. L3, and equal distance upstream, does not form a peak with L2 and served as a control. In 26 of 50 loci (52%), the distance between peak loci, L1 and L2, was significantly shorter (by at least 0.2 μm) than the distance between L3 and L2. The distance between L3 and L2 was significantly shorter than the distance between L1 and L2 in 13 of 50 cases (26%). Note that the slides used for Peak 4 had somewhat elevated background levels, which may explain the more muted statistical effect.

The above calculations require a difference of 0.2 μm in the measured distances for the difference to be included in our counts. This threshold can be set anywhere between 0.0 and 0.6 μm without affecting the results. (For larger values, the number of "significant differences" is negligible.)

In Figure S5B, we plot cumulative distributions of the L1-L2 and L2-L3 distances for each of Peaks 1-4. In all four cases, L2 tends to be closer to its looping pair L1 than to the control locus, L3 (Peak 1: p-value<10^{-9}; Peak 2: p-value<0.003; Peak 3: p-value=10^{-9}; Peak 4: p-value<.027).

VI.d. Aggregate Peak Analysis (APA)

Aggregate Peak Analysis (APA) is a method that allows us to test the aggregate enrichment of an entire set of putative two-dimensional peaks, as opposed to verifying individual peaks one-by-one. This method is especially useful for checking a set of peak calls on a low-resolution Hi-C map, where individual peaks may be impossible to discern but where the aggregate signal from the full peak set should be detectable if the peak set is reliable and the Hi-C map was the result of a successful experiment in the same cell type. Notably, we have not found an instance in which one of the *in situ* Hi-C peak lists reported here failed to be validated, in aggregate, when examined with respect to a successful Hi-C experiment in the same cell type.

VI.d.1. Standard APA analysis: APA quantifies the enrichment of a peak set in aggregate by plotting the sum of a series of submatrices derived from a contact matrix. These submatrices are chosen so that each one surrounds a single putative peak pixel (note that for intrachromosomal pixels, we always choose the pixel in the upper-right half of the matrix). In the resulting APA plot, the total number of contacts that lie within the peak pixel set is shown at the center; the entry immediately to the right of center corresponds to the total number of contacts in the pixel set obtained by shifting the peak set 10 kb to the right; the entry two positions above center corresponds to an upward shift of 20 kb; and so on. Focal enrichment across the peak set in aggregate manifests as larger values at the center of the APA plot.

To perform APA, a resolution and window size is chosen. In Figure 3, we have chosen 10 kb resolution and a window of +/- 100 kb (10 pixels) about each putative peak. The APA resolution determines the resolution at which the contact matrix of a Hi-C map is generated. Note that, in APA analyses, contact matrices will often be generated for a given Hi-C map at resolutions vastly higher than the resolution at which the map would usually be examined. For instance, we generated contact matrices for our 2009 Hi-C maps (Lieberman-Aiden et al., 2009) at 10 kb resolution, despite the fact that the map resolution of these maps is orders of magnitude larger. (The aggregation process makes it possible to resolve features at much higher resolutions than would ordinarily be possible with a map, producing a “super-resolution” image.)

We center a submatrix at each peak in the target peak set at the chosen resolution. The width of the matrix is the window size above. For peak calls that span an area larger or smaller than one pixel in the chosen resolution, we choose the center of the peak call as the center of the matrix. Only one submatrix is created per peak call, even if the peak call extends to multiple pixels. If the center pixel of multiple peak calls falls into the same pixel, we use that submatrix only once. To avoid strong distance effects, we only examine peak calls where the peak loci are separated by more than a minimum threshold t.

For APA performed at 10 kb resolution with a window of +/- 100 kb, t = 300 kb. For APA performed at 5 kb resolution with a window of +/-25 kb, t = 100 kb.

Submatrices for the peaks are taken from the normalized (intrachromosomal KR corrected) Hi-C maps. These submatrices are then summed (entry-wise), obtaining an APA matrix in which the center pixel represents the sum of the number of reads in the entire target loop set. For the MiSeq datasets, a KR correction is not available at 10 kb due to the sparsity of the data, so the KR correction factors for all the maps are approximated by first calculating them for a coarser resolution.

To determine if the center pixel of the APA plot is focally enriched, we calculate the APA score, which is the ratio of the number of reads in the center bin to the average number of reads in the lower-left corner of the APA matrix. We define the lower left bins for 10 kb resolution and a +/- 100 kb window size as the bins lying in the bottom-left 6 x 6 square of the matrix. For 5 kb resolution and a +/- 25 kb window we define the lower left bins as those lying in the lower left 3 x 3 square of the matrix. We chose to use this particular score because it is very simple to understand and calculate, and because it corresponds to the widely-accepted notion of a loop: in order for the APA score to be above 1, the number of contacts between a typical pair of loop anchors in the peak set must be higher than the number of contacts between intervening pairs of loci.

To calculate a p-value for this score, we calculate the z-score which compares the central bin to the set of bins in the lower left window defined above. The z-score is then converted into a p-value (1-sided).

The color scale in all APA plots is set as follows. The minimum of the color range is 0. The maximum is 5 x UR, where UR is the mean value of the bins in the upper-right corner of the matrix. The upper-right corner of the 10 kb resolution APA plots is a 6 x 6 window (or 3 x 3 for 5 kb resolution APA plots).

VI.d.2. Control peak sets. We created control peak sets to investigate the effect of domains on APA. To generate these control peak sets, we randomly choose pixels inside of the GM12878 domains that we annotated using the Arrowhead algorithm. The control peak sets did not display APA enrichment.

VI.d.3. Advanced APA analyses. Although the paper focuses on the most straightforward form of APA analysis, which is described above, we also explored a number of more sophisticated methods.

A simple variant on standard APA is "normalized" APA. In normalized APA, each submatrix is "normalized" by dividing all entries of the submatrix by the mean value of the submatrix, such that the mean value of the entries of the resulting submatrix is 1. This ensures that short-range loops do not exert a disproportionate influence on the results of the analysis.

There are also more sophisticated APA scores. For instance, the standard APA score is the ratio of the central bin to average of the lower-left corner. One can check this ratio for all four corners of the matrix. The upper-right is a less useful control: even a random set of pixels should be enriched relative to the upper-right corner of the matrix, due purely to distance effects. The upper-left and lower-right corners comprise pairs of loci that are close to the same distance as the peak loci, so the ratio of the central bin to these corners should be close to 1 for control peak sets. However, enrichment relative to these corners may result from the fact that pixels in the peak set tend to be in domains, rather than at true peaks.

One can also examine the peak to mean value (the ratio of the center bin to the mean of the rest of the matrix). Because the number of contacts varies as a function of distance, this value is less informative. In Data S2.I.A-B we show examples of these other measures.

We can also perform a more formal measure of enrichment by using the paired Student's T-test. For this, we calculate the one-sided t-statistic, measuring enrichment between the central peak bin and every other bin in the matrix. Values above 2, with low p-values, indicate that the central bin is enriched relative to the bin of interest; values below 2 indicate depletion. Data S2.I.C-D show examples of these matrices.

VI.d.4 Our peak annotations exhibit focal enrichment in all published Hi-C datasets via APA: APA was performed on all published lymphoblast Hi-C datasets (Table S6) using our GM12878 peak annotations. The resulting APA plots are shown in Data S2.I.E. Focal enrichment was seen in all cases, even on sparse maps with <10M reads. We also performed APA on maps we constructed using the dilution Hi-C protocol and maps we constructed using the *in situ* Hi-C protocol without crosslinking (Data S2.I.E). Notably, even without the use of crosslinking, we observed a robust enrichment (HIC043, APA score: 2.838, Z-score: 26.241; HIC044, APA score: 2.030, Z-score: 14.586; HIC045, APA score: 3.063, Z-score: 8.814; HIC046, APA score: 2.105, Z-score: 15.962; HIC047, APA score: 2.114, Z-score: 8.060), see Table S6. As such, our loops cannot be a result of crosslinking biases. Finally, we performed APA on 107 additional coarse resolution experiments covering a variety of experimental conditions (see Table S1); enrichment was seen in every case (Figure S8E).

Additionally, APA was performed on the relevant published Hi-C maps or our own dilution maps for our other *in situ* peak lists: IMR90 peaks (APA performed on 3 maps), K562 peaks (1 map), HMEC peaks (1 map), HUVEC peaks (1 map), NHEK peaks (1 map), and mouse lymphoblastoid peaks (2 maps). Again, focal enrichment was seen in all cases (see Data S2.I.F).

VI.e. Properties of Peaks

VI.e.1. Peaks are conserved through different cell lineages: To assess the rate of conservation of the peaks we identified in our six human cell types, we first compared our GM12878 peak annotation to each of the other annotations. Using our standard metric for overlap, we found that approximately 60% of peaks called in any cell type were conserved in GM12878 (see Figure S5D). Additionally, we observed slightly higher rates of conservation for other mesodermal cell lines in GM12878 (68.2% vs. 58.9%), which is consistent with the hypothesis that three dimensional genome structure varies in a developmentally dependent manner; further experiments are needed to confirm this. However, these estimates of conservation are likely underestimates due to the higher false negative rate of our peak annotation algorithm in our non-GM12878 maps, where the sequencing depth is lower.

VI.e.2. Three-dimensional structure is strongly conserved through mammalian evolution: In order to compare our annotations across organisms, we first used liftOver (Hinrichs et al., 2006) to convert our CH12 mouse lymphoblast peak and domain annotations from mm9 mouse genome coordinates to hg19 human genome coordinates. For both peaks and

domains, we used liftOver to lift over the entire interval defined by the peak or the domain, requiring at least a .1 sequence match rate. To check if a feature was conserved after lifting over, we checked if the pixel $M_{i,j}$ defined by the endpoints of the interval was within min(0.5**|i-j|*, 50 kb) of a relevant feature in GM12878 (the more permissive 0.5**|i-j|* was used instead of 0.2**|i-j|* to allow for errors in liftOver across organisms). We lifted over 3245 of 3331 mouse peaks to $(10\text{ kb})^2$ pixels in human genome coordinates, and found that 1649 (49.5%) overlapped a corresponding peak in GM12878 human lymphoblastoid cells (Figure 4B-E). Similarly, we lifted over 2884 of 2927 intervals associated with domains to a corresponding interval in human genomic coordinates. We found that 1309 out of 2927 domains (44.7%) annotated in CH12 mouse lymphoblasts had a corresponding annotated domain in GM12878 human lymphoblastoids.

VI.e.3. Peaks are enriched for promoters and enhancers: To identify promoter-enhancer peaks in our peak annotation, we utilized the ENCODE combined HMM segmentation. We looked for peaks that had a 'TSS' state on one peak locus and either an 'E' or 'WE' state on the other. Of course many 10-15 kb loci in the genome contain both promoters and enhancers, so it is possible that our labeled promoter-enhancer loops are in addition promoter-promoter or enhancer-enhancer loops. We observed 2854 peaks with promoters and enhancers in our GM12878 annotation of 9448 peaks (30%), as opposed to only 653 (7%) in a comparable random annotation. We found that the transcription start sites for 3361 protein coding genes lay in peak loci in GM12878 while the TSS's of 17,281 protein coding genes did not. The median RNA-Seq expression of the genes that lay in peak loci was 6-fold higher (1.91 RPKM vs. 0.3 RPKM).

VI.e.4. Formation of differential peaks is associated with gene activation: To identify peaks that were different between cell types, we intersected our peak annotation in GM12878 with our peak annotations in our other cell types. In order to ensure that we did not call pixels that were enriched in both cell types but happened not to be annotated in one cell type as a differential peak, we restricted ourselves to only looking at pixels that were annotated as a peak in one of the cell types and that displayed less than 1.3-fold enrichment over all of the local neighborhoods in the other cell type. In this manner, we identified 557 peaks called in GM12878 but not enriched in IMR90, 640 peaks called in GM12878 but not enriched in HMEC, 737 peaks called in GM12878 but not enriched in K562, 917 peaks called in GM12878 but not enriched in HUVEC, 1050 peaks called in GM12878 but not enriched in HeLa, and 910 peaks called in GM12878 but not enriched in NHEK. Similarly, we identified 510 peaks called in IMR90 but not enriched in GM12878, 439 peaks called in HMEC but not enriched in GM12878, 323 peaks called in K562 but not enriched in GM12878, 192 peaks called in HUVEC but not enriched in GM12878, 242 peaks called in HeLa but not enriched in GM12878, and 446 peaks called in NHEK but not enriched in GM12878.

We identified genes whose transcription start site (according to the GENCODE V7 annotation) lay within 2.5 kb of the 10 kb peak loci of our putative differential loops (455 genes at loops in GM12878 but not IMR90; 593 genes at loops in GM12878 but not HMEC; 660 genes at loops in GM12878 but not K562; 881 genes at loops in GM12878 but not HUVEC; 1033 genes at loops in GM12878 but not K562; 952 genes at loops in GM12878 but not NHEK; 450 genes at loops in IMR90 but not GM12878; 414 genes at loops in HMEC but not GM12878; 323 genes at loops in K562 but not GM12878; 232 genes at loops in HUVEC but not GM12878; 240 genes at loops in HeLa but not GM12878; 355 genes at loops in NHEK but not GM12878). We examined the gene expression of these genes (using RNA-Seq data produced by ENCODE) and found that genes whose promoter lay at the peak loci of a differential loop domain tended to be upregulated in the cell type where the loop domain was present (Figure 5C, Table S7, Data S1.VI).

Notably, the appearance of a loop is usually accompanied by the appearance of a domain. We observed similar results when we further restricted the set of peaks we examined by requiring that any differential peak pixel exhibit a corner score that was 0.2 higher in the cell type that contained the annotated loop. (See below in section *VI.e.5* for more discussion on the relationship between loops and domains.)

VI.e.5. Peaks tend to demarcate the corners of domains: To identify peaks that demarcated domains (and vice versa), we integrated our peak annotations and domain annotations in a manner similar to the method used to check peak and domain reproducibility. For every domain (resp. peak) defined by top right corner (resp. peak pixel) $M_{i,j}$, we asked if there was a peak (resp. domain) at $M_{i',j'}$, such that the two pixels were within Euclidean distance r of each other. The parameter r was defined to be 0.2*|*i-j*|. The 0.2 coefficient was chosen by repeating the same procedure between our domain annotation and a randomly permuted peak annotation (and vice versa), and visually determining the threshold at which the marginal increase in threshold identifies similar numbers of additional peak/domain coincidences in both the real annotations and the random control. Once again, this overlap parameter is robust to reasonable variations. We observed that 38% of GM12878 peaks (3628 out of 9448) overlapped the corner of a called domain (40% of domains [3669 out of 9274] were demarcated by a peak). In IMR90, 38% of peaks and 39% of domains overlapped; in HMEC, 25% of peaks and 34% of domains overlapped; in NHEK, 41% of peaks and 37% of domains overlapped; in K562, 39% of peaks and 37% of domains overlapped; in KBM7, 42% of peaks and 25% of domains overlapped; in HUVEC, 16% of peaks and 39%

of domains overlapped (half as many domains were called as peaks); in HeLa, 38% of peaks and 29% of domains overlapped; and in CH12 mouse lymphoblasts, 31% of peaks and 39% of domains overlapped.

Additionally, we used the Arrowhead algorithm to identify the corner scores at peak pixels and at a randomly permuted control peak list (keeping size distribution the same; see random shuffling procedure). The corner score distributions for GM12878 peaks are shown in the main text (Figure 6A); we repeated the same procedure in all the cell types (Figure S6A-F). The median corner score for an IMR90 peak corresponded to the 96th percentile of random pixels; in HMEC, the median corner corresponded to the 97th percentile; in NHEK, the 99th percentile; in K562, the 97th percentile; in KBM7, the 98th percentile; in HUVEC, the 98th percentile; and in HeLa, the 98th percentile.

Notably, this relationship between loops and domains was observed for the subset of our loops that are promoter-enhancer loops as well. 36% of our 2854 GM12878 promoter-enhancer loops were at the corner of domains and the median corner score for these loops corresponded to the 96th percentile of random pixels. We also found that when promoter-enhancer loops are not associated with the boundary of a domain, they tend to lie inside the domains; only 15% of promoter-enhancer loops cross domain boundaries, a 3-fold depletion.

Similarly, we used HiCCUPS to quantify the local enrichments at pixels at the corners of annotated domains. We classified a domain as having no enrichment at the corner if all the pixels at the corner were less than 1.3-fold enriched over at least one of the four local neighborhoods. In GM12878, we found that 15% of domains showed no local enrichment at the corner. While some of these annotated domains without local enrichment at the corner may be false negatives, there appear to be many domains throughout the genome independent of peaks. (Notably, we analyzed the domains without any enrichment in the corner separately and found that they showed similar correlations of histone modification across the domain as the full list of domains.)

The appearance of a loop is strongly associated with the appearance of domains. 648 out of 1088 loops (60%) that appear in IMR90 but not GM12878 (defined as not being called in GM12878 and <1.5-fold enriched in GM12878) show an associated increase of 0.2 in corner score, indicative of the appearance of a domain. Similarly, 645 out of 890 loops (72%) that appear in GM12878 but not in IMR90 show an associated increase of 0.2 in corner score. In order to determine how domains that do not show enrichment at the corners may form, we analyzed those 15% of domains separately. We found that 55% of the boundaries of non-loop domains were shared with loop domains or peak loci, indicating that an interval of chromatin flanked on both sides by loop domains may form a domain itself despite the fact that it is not contained within a loop. The other 45% of boundaries were not enriched in CTCF (see below, section *VI.e.7*), but they showed similar properties to the full list of domain boundaries in terms of the long-range contact gradients seen at the boundaries (see section *IV.c.3*) and transitions in histone modification pattern at the boundaries. Thus, we hypothesize that transitions in compartment status can exist independent of looping and often establish boundaries of domains independent of looping.

Sometimes, peaks demarcating nearby domains exhibit transitivity; if L_1 and L_2 form a peak, and L_2 and L_3 form a peak, the transitive peak would be between L_1 and L_3. One possible explanation for transitive peaks is that all loci are simultaneously co-located. However, this need not always be the case. For instance, in the haploid Hi-C maps, we observed focal interactions between the HIDAD locus upstream of both H19 and Igf2 and both the H19 promoter and the Igf2 promoter. However, these interactions are not simultaneous; the interaction between H19 and HIDAD takes place on the maternal allele and the interaction between Igf2 and HIDAD takes place on the paternal allele.

VI.e.6. The formation of a peak is accompanied by a depletion of additional peaks from its interior to its exterior: In order to test the hypothesis that our peak annotation showed a depletion of "loop crossings" (two peak pixels $M_{a,c}$ and $M_{b,d}$ such that $a<b<c<d$), for every peak in our annotations, we asked whether there were any additional peaks in our annotation that crossed it. In order to avoid categorizing two pixels $M_{a,b}$ and $M_{a',c}$ where a and a' were actually equivalent but slightly offset from each other due to peak localization error, we imposed the restriction that for peak $M_{b,d}$ to be defined as a crossing peak $M_{a,c}$, both loci L_b and L_d had to be at least 20 kb or 0.1*(c-a) (whichever one was smaller) away from locus L_a and L_c. Additionally, since a network of transitive loops between loci L_a, L_b, L_c, and L_d will result in two peaks $M_{a,c}$ and $M_{b,d}$ such that $a<b<c<d$, but which is not really a true loop crossing since all involved loci could be simultaneously colocated, we did not count a set of two peaks $M_{a,c}$ and $M_{b,d}$ such that $a<b<c<d$ as a loop crossing if there existed an additional peak $M_{b',c'}$ such that locus $L_{b'}$ was within 20 kb of locus L_b and locus $L_{c'}$ was within 20 kb of locus L_c. Finally, we excluded peaks between loci separated by more than 2 Mb for this analysis, since those extremely long range peak calls contain many false positives.

After employing this method, we found that 1386 out of 9448 peaks smaller than 2 Mb (15%) were crossed in GM12878. When we examined a randomly permuted peak annotation controlled for size and chromosome distribution (see Section

IV.b), we observed crossings for 5381 of 9447 peaks (57%), i.e. our peak annotation is 4-fold depleted for "loop crossings". The number of crossings we do observe in our annotation is likely inflated due to false positive peak calls. Similar depletions were seen in all other cell types (IMR90, 8-fold depleted; HMEC, 18-fold depleted; NHEK, 8-fold depleted; KBM7, 5-fold depleted; K562, 5-fold depleted; HUVEC, 5-fold depleted; HeLa, 8-fold depleted).

Additional evidence supporting the notion that loops cannot cross was observed at loci that loop to both upstream and downstream sites; see Section *VI.e.8* on strand orientation below.

VI.e.7. CTCF and cohesin are enriched at peak loci: For each peak annotation, we identified a list of loci involved in the peaks by separating each peak into its two component peak loci, removing non-unique loci from the resulting list, and merging any adjacent intervals into one larger interval. For GM12878, this resulted in our list of 9448 peaks being reduced to a list of 12903 peak loci (the reason that the number of loci isn't closer to 2*(# of peaks) is due to transitive looping). For IMR90, 8040 peaks were formed between 11,459 peak loci; for HMEC, 5,152 peaks between 8202 loci; for NHEK, 4929 peaks between 7,435 loci; for K562, 6,057 peaks between 8436 loci; for KBM7, 2634 peaks between 4,279 loci; for CH12 mouse lymphoblasts, 3331 peaks between 5312 loci.

We then downloaded the ENCODE uniform ChIP-Seq peak calls for 76 transcription factors (TFs), 10 histone modifications, and DNase hypersensitivity site (DHS) peak calls in GM12878; 5 TFs, 8 histone modifications, and DHS in IMR90 (for IMR90 fetal lung fibroblasts, we used histone modification data produced in normal human lung fibroblasts (NHLF)); 2 TFs, 8 histone modifications, and DHS in HMEC; 2 TFs, 9 histone modifications and DHS in NHEK; 98 TFs, 11 histone modifications and DHS in K562; 8 TFs, 8 histone modifications and DHS in HUVEC; 55 TFs, 8 histone modifications and DHS in HeLa; and 30 TFs, 7 histone modifications, and DHS in CH12 mouse lymphoblasts (Table S3). ENCODE has not generated any ChIP-Seq data in KBM7.

For each TF, histone modification, or DHS, we then iterated through our list of peak loci in that cell type and counted how many peak loci had at least one called ChIP-Seq peak for that TF, histone modification or DHS inside the locus. In cases where multiple replicate experiments for the same TF were performed, we handled this in one of two ways: we either only allowed peaks called in all replicates (i.e. overlapping intervals in all replicates) for all TFs or we allowed all peaks called in any replicate for all TFs. For peak loci smaller than 15 kb, we extended the locus on either side until the interval was 15 kb. We also repeated the same procedure for two lists of randomly permuted loci (allowing loci to be randomly placed genome-wide or randomly placed within a chromosome; see Section *IV.b.*) as well as a list of loci shifted by 100 kb in either direction as a control. Enrichments for each TF/histone modification/DHS were then calculated by dividing the number of peak loci with a ChIP-Seq peak by the number of control loci with a ChIP-Seq peak. The fraction of peak loci containing a ChIP-Seq peak was calculated by dividing the number of peak loci with a ChIP-Seq peak by the total number of peak loci.

We first noticed that almost all proteins were enriched in peak loci. In GM12878, the average TF/histone modification was 3.4-fold enriched over random expectation, and DHSs were 2.8-fold enriched; similar enrichments were seen in other cell types. However, we wanted to know which proteins were enriched relative to other proteins; thus, we also calculated normalized protein enrichments by dividing all enrichments by the average enrichment (or in cases where not many proteins were tested, by dividing by the DHS enrichment). Figure 6C and Figure S6G-N shows the normalized protein enrichments.

Figure 6C also only includes TF enrichments; histone modification and DHS enrichments are included in Fig S6G-N. No histone modification peaks were present on >50% of peak loci or enriched more than 2-fold (normalized enrichment). DHSs were present on 91% of peak loci in GM12878. Additionally, the data shown in Figure 6C was generated by only including peaks that appeared in all replicates done for that TF. Notably, ENCODE performed four separate replicates for CTCF in GM12878. If all peaks that appear in any replicate are included, >91% of peak loci contain a CTCF binding site (but the enrichment drops due to the large number of CTCF ChIP-Seq peaks called). In general, similar results were seen when performing the above calculations while including any peak from any replicate. In IMR90, we observed that 87.7% of peak loci contained CTCF and 87.2% of peak loci contained RAD21; in HMEC, 84.9% of peak loci contained CTCF; in NHEK, 73.2% of peak loci contained CTCF (90.2% if any peak from any replicate is included); in K562, 88.1% of peak loci contained CTCF (93.8% if any peak from any replicate is included), 78.8% of peak loci contained RAD21 (89.6% with any peak from any replicate), and 84.2% of peak loci contained SMC3; in HUVEC, 86.9% of peak loci contained CTCF (91.2% with any peak from any replicate); in HeLa, 87.7% of peak loci contained CTCF (91.3% with any peak from any replicate), 88.3% of peak loci contained RAD21, and 88.1% of peak loci contained SMC3; and in CH12 mouse lymphoblasts, 80.9% of peak loci contained CTCF (94% with any peak from any replicate), 94.2% of peak loci contained RAD21, and 92.7% of peak loci contained SMC3 (See Figure S6G-N for enrichments).

Also of note, the proteins YY1 and ZNF143 were present on a large fraction (>60%) of our peak loci in GM12878. ZNF143 was also present on a large fraction (72.7%) of the peak loci in K562. YY1 was present on a smaller but still substantial fraction (34.3%). These proteins are known to colocalize with CTCF, although their exact functional role is uncertain (Donohoe et al., 2007; Gerstein et al., 2012).

To localize peak loci down to a small number of nucleotides, in every peak locus we checked if there was one and only one CTCF ChIP-Seq peak (in the 15 kb+ window, only allowing ChIP-Seq peaks that were present in all replicates). If there was, we additionally required that the CTCF ChIP-Seq peak overlapped a RAD21 ChIP-Seq peak, an SMC3 ChIP-Seq peak and a good CTCF sequence motif (in cell types with no SMC3 ChIP-Seq data, we only required that the CTCF ChIP-Seq peak overlap a RAD21 peak and contain a good motif; in cell types with no RAD21 or SMC3 data; we only required a good motif). The presence of a good motif was ascertained using STORM (Schones et al., 2007). For every candidate CTCF ChIP-Seq peak, we identified the underlying 20bp sequence with the highest match to the consensus CTCF position weight matrix (CTCF PWM from Kim et al. (2007)). Positive scores returned by STORM indicate good alignments to the PWM used, and negative scores indicate bad alignments. We required that the best motif within candidate CTCF ChIP-Seq peaks have a positive PWM match score. In this manner, we were able to localize 6991 of 12903 (54%) peak loci down to 20bp CTCF motifs that bind CTCF/RAD21/SMC3 in GM12878, 5334 of 11459 (46.5%) peak loci in IMR90 (no SMC3 data); 3995 of 8202 (48.7%) peak loci in HMEC (no RAD21/SMC3 data); 3037 of 7435 (40.8%) peak loci in NHEK (no RAD21/SMC3 data); 3303 of 8436 (39.1%) peak loci in K562; 2966 of 6161 (48.1%) peak loci in HUVEC; 1837 of 5020 (36.6%) peak loci in HeLa; and 2633 of 5312 (50%) peak loci in CH12 mouse lymphoblasts.

To calculate what percentage of CTCF and cohesin bound loci fall within our peak loci in GM12878, we compared the number of ChIP-Seq peaks that fall within our peak loci to the total number of ChIP-Seq peaks (14639/35392 (41.4%) for CTCF, 13917/30349 (45.9%) for RAD21, 14379/30517 (47.1%) for SMC3, 12678/23972 (52.9%) for overlapping CTCF/RAD21/SMC3), again only allowing peaks that were called in all replicates. If this procedure is repeated, allowing any ChIP-Seq peak from any replicate, we find that 18943 of 62580 (30.3%) CTCF ChIP-Seq peaks are contained within our peak loci, 16156 of 42727 (37.8%) RAD21 ChIP-Seq peaks and 13496 of 27438 (49.2%) overlapping CTCF/RAD21/SMC3 peaks. Similar numbers are seen in other cell types: in IMR90, 32.8% of CTCF ChIP-Seq peaks, 37.7% of RAD21 peaks, and 39.6% of overlapping CTCF/RAD21 peaks are found in peak loci; in HMEC, 24.8% of CTCF ChIP-Seq peaks are found in peak loci; in NHEK, 25.2% of CTCF ChIP-Seq peaks are found in peak loci (19.8% including any peak in any replicate); in K562, 32.2% of CTCF ChIP-Seq peaks (23.4% including any peak in any replicate), 48.9% of RAD21 peaks (33.1% including any peak in any replicate), 41.8% of SMC3 peaks, and 51.7% of overlapping CTCF/RAD21/SMC3 peaks (43.1% including any peak from any replicate) are found in peak loci; in HUVEC, 26% of CTCF ChIP-Seq peaks (20.3% including any peak in any replicate) are found in peak loci; and in CH12 mouse lymphoblasts, 23% of CTCF ChIP-Seq peaks (13% including any peak in any replicate), 13% of RAD21 peaks, 17% of SMC3 peaks, and 24% of overlapping CTCF/RAD21/SMC3 peaks (18% including any peak in any replicate) are found in peak loci.

We also calculated the percentage of intra-peak locus CTCF ChIP-Seq peaks near promoters by identifying all CTCF ChIP-Seq peaks within our peak loci that were within 2.5 kb of a transcription start site according to the GENCODE V7 annotation. Also note that ZNF143 (present on ~60% of our peak loci in GM12878) is a human homolog of the STAF protein in *Xenopus*, and is a strong transcriptional activator in vertebrates (Myslinski et al., 2006); it is possible that its colocalization with CTCF at promoters mediates some of CTCF's regulatory influence on gene expression.

Notably, the promoter-enhancer peaks we identified (see Section *VI.e.3*) were overwhelmingly CTCF-CTCF loops. 2404 out of 2854 promoter-enhancer peaks (84%) had CTCF at both ends (seen in any CTCF ChIP-Seq replicate), compared to 8419 out of 9448 total peaks (89%). We observed no more promoter-enhancer peaks without CTCF altogether than would be expected by chance; we observed only 62 promoter-enhancer peaks that had no evidence of CTCF binding at either peak locus.

VI.e.8. Loops bind CTCF in a convergent orientation at their anchors: As noted in the main text, CTCF binds a motif (5'-CCACNAGGTGGCAG-3') that is asymmetric; as such it has an orientation, and can be present on either the forward or reverse strand at a given genomic position. We refer to a motif as a forward motif if it is present on the forward strand of hg19 and a reverse motif if it is present on the reverse strand of hg19. For a given pair of motifs, the motifs can be oriented in a convergent orientation (forward-reverse, i.e. the motif closer to the p-terminus is on the forward strand, and the motif closer to the q terminus is on the reverse strand), a divergent orientation (reverse-forward), the same direction on the forward strand (forward-forward) or the same direction on the reverse strand (reverse-reverse).

To assess the effect of motif orientation on looping, we examined the peak loci that we were able to localize down to a single CTCF motif (see Section *VI.e.7* for how unique motifs were identified) as well as the peaks where we were able to localize both anchors to a single motif. We observed 6991 unique motifs in GM12878 of which 3401 motifs appeared only in an upstream anchor (i.e. the anchor closer to the p-terminus of the chromosome) of a loop, and 3393 motifs appeared only in a downstream anchor (i.e. the anchor closer to the q-terminus of the chromosome) of a loop. Out of 3401 motifs in upstream anchors, 3267 (96%) were forward motifs. Out of 3393 motifs in downstream anchors, 3271 (96%) were reverse motifs. Out of 4322 peaks where we could localize both anchors down to unique motifs, we observed that 3971 (92%) of the peaks contained a pair of motifs in a convergent orientation. We also observed that our loops were strongly enriched for convergent orientation motif pairs over random expectation. 6608 of our loops contained at least one forward bound motif in the upstream peak locus and at least one reverse bound motif in the downstream peak locus; by chance, one would expect only 65 loops to satisfy this property (102-fold enrichment).

This surprising result was also observed in other cell types as well as in mouse lymphoblasts. In IMR90, 1456 out of 1639 peaks (89%) that could be localized to two unique motifs exhibited a convergent pair of motifs. In HMEC, 968 out of 1160 peaks (83%) exhibited convergent orientations. In K562, 723 out of 822 peaks (88%) exhibited convergent motifs. In HUVEC, 671 out of 771 peaks (87%) exhibited convergent motifs. In HeLa, 301 out of 356 peaks (85%) exhibited convergent motifs. In NHEK, 556 out of 731 peaks (76%) exhibited convergent motifs. In CH-12 mouse lymphoblasts, 625 out of 772 peaks (81%) exhibited convergent motifs.

This striking correspondence also allowed us to assign more loop anchors to unique CTCF motifs. For example, if an upstream peak locus contained three CTCF binding sites, two with reverse motifs and one with a forward motif, our initial scheme for localizing a peak locus to a single motif would not have succeeded at this example locus. However, by requiring that an upstream locus must correspond to a forward motif, we are now able to ignore the two reverse motifs in the locus and localize the peak locus down to the single forward motif in the locus. In this manner, we localized an additional 1184 peak loci to unique CTCF motifs in GM12878. In total, we were able to localize 8175 of 12903 GM12878 peak loci to unique motifs.

The constraint that every loop must form between a forward and reverse motif in a convergent orientation has strong implications for the possible loop networks that can form in the genome. Consider a simple example, the transitive triple: three loci, L1, L2 and L3 that lie consecutively in the genome and form all pairwise loops (L1-L2, L2-L3, and L1-L3). Since L2 participates in loops in both directions (towards the p-terminus (L1) and towards the q-terminus (L3) of the chromosome), L2 must contain at minimum two motifs (a forward motif and a reverse motif). Thus, our convergent orientation constraint implies that a transitive triple cannot form between three CTCF sites, rather there must be at minimum four CTCF sites involved. To test this, we identified all transitive triples in our GM12878 loop annotation (a transitive triple was defined as three loops (L1-L2, L2'-L3, L1'-L3') where L1 and L1', L2 and L2', and L3 and L3' were within 20 kb of each other. Specifically from these transitive triples, we examined the L2 loci in order to determine whether they contained multiple CTCF binding sites. We found that 376 of 474 L2 loci (79%) contained more than one binding site. When we visually inspected the 21% that didn't contain multiple binding sites, we found that they were usually not part of a true transitive triple and were instead spurious calls due to the presence of one false positive loop among the triple.

We wondered whether the order of motifs within the L2 locus had an effect on loop formation; we will describe the motif in L2 that is closer to L1 as L2-A, and the motif that is closer to L3 as L2-B. Since we had previously observed a depletion for loop crossings in our data (see Section *VI.e.6*), we reasoned that if an L2 locus contained first a forward motif (as the L2-A motif) and then a reverse motif (as the L2-B motif), this might be disadvantageous to loop formation as then the L1 – L2-B loops and L2-A – L3 loops would cross. Strikingly, when we examined all L2 loci containing exactly two binding sites, we observed 134 instances of a reverse motif at the L2-A site and a forward motif at the L2-B site, but only 19 instances of a forward motif at the L2-A site and a reverse motif at the L2-B site. When we examined all L2 loci containing exactly three binding sites, the effect was even stronger. We observed 75 instances of either a reverse-forward-forward triplet of motifs or a reverse-reverse-forward triplet of motifs (both contain the reverse-forward dyad); conversely, we only observed 5 instances of either a forward-reverse-reverse triplet of motifs or a forward-forward-reverse triplet of motifs (both contain the forward-reverse dyad). This further supports the notion that the loops we observe are topologically insulating in that they cannot cross one another. It also suggests that the loop residence times are long enough such that at some point in time the L1 – L2-B loop and the L2-A – L3 loop are happening simultaneously; otherwise there would be no pressure for a reverse-forward dyad over a forward-reverse dyad. More work is needed to explore this possibility.

VI.e.9. Exapted SINE/B2 repeats can form loops: Frequent observations of CTCF binding sites contained within rodent-specific SINE/B2 repeat elements have suggested that retrotransposon expansions are a mechanism by which novel CTCF binding sites can spread across the mouse genome. Using our loop lists in mouse lymphoblasts, we sought to

probe whether these exapted repeats could modulate the three dimensional structure of the genome. We identified 1804 CTCF motifs where we had very high confidence that the motifs were the few base pairs responsible for looping (by intersecting CTCF, RAD21, and SMC3 ChIP-Seq binding data and requiring that only a single high quality match to the consensus CTCF motif be present inside the binding site). We then overlapped those CTCF motifs with SINE/B2 repeat tracks downloaded from UCSC (as determined by Repeat Masker). We found that 136 CTCF motifs (7.3%) overlapped a SINE/B2 repeat element.

When we compared the orientations of these SINE-overlapping CTCF motifs to the orientations of the loop loci they fell in, we found that 118 out of 133 times (89%), the orientation of the CTCF and the orientation of the loop locus agreed, i.e. the CTCF motif was a forward motif if contained in an upstream loop locus (towards the p-end of the chromosome) and a reverse motif if contained in a downstream loop locus (towards the q-end of the chromosome). (3 sites were left out of the analysis because they participated in loops as both an upstream and downstream locus.) This was consistent with the 94% agreement in orientation we observed genomewide between unique CTCF motifs and their loop loci. Since the orientation of the SINE/B2 sequence (as determined by Repeat Masker) was aligned with the orientation of the CTCF motif 97% of the time (130/133 cases), this implies that the insertion orientation of SINE/B2 strongly influences the possible novel three dimensional structures that can be formed by its exaptation.

VII. Diploid Hi-C

VII.a. Construction of Diploid Hi-C maps

The cell line GM12878 is derived from the daughter in the CEU trio; GM12891 and GM12892 are derived from her parents. Maternal and paternal SNPs are available as part of the 1000 genomes project (Gil et al., 2012). Given a SNP list, we examined all Hi-C contacts filtered by MAPQ ≥10 in our GM12878 libraries (including both the *in situ* and dilution protocol) that overlapped a SNP within the first 70 nucleotides of the read. For each overlapping read, we located the nucleotide at the appropriate position and matched it to the paternal or maternal SNP. If it didn't match either paternal or maternal SNP, we excluded it from further processing. We created the Hi-C diploid contact maps seen in the paper from the unambiguous paternal and maternal contacts. Table S8 contains the breakdown of allele assignments.

One way of judging the quality of our diploid assignments is to look at the allele mismatch rate for intra-chromosomal reads. When read ends are close together and both overlap SNPs, we expect that the read is almost always intramolecular and thus we should see the same allelic assignment. Our initial SNP list from the 1000 Genomes Project was generated from GATK 2.5 (McKenna et al., 2010). The allelic mismatch rate was 7.5%, even for read ends fewer than 20 kb apart. This seemed unrealistically high. Investigating further, we ran the same algorithm on a paired-end DNA-Seq experiment and still saw a mismatch rate of 5.6%. We suspected that errors were due to inaccuracies in the SNP phasing.

Next, we intersected our original phased SNP list with newer phased SNP lists for the same trio generated by the 1000 Genomes Project using additional data and GATK 2.8, as well as phased SNP lists generated by Illumina as part of their Platinum Genomes project (http://www.illumina.com/platinumgenomes/). We included only single nucleotide modifications, and not additions, deletions, or insertions. The intersected list contained 1,787,252 SNPs. When we used this intersected SNP list, our allelic mismatch rate fell to < 0.7% for read pairs fewer than 20 kb apart; this is comparable to the mismatch rate seen when performing an identical analysis on a paired-end DNA-Seq library. We therefore used the intersected SNP list for all analyses and figures in the paper. Our observations suggest that the quality of diploid Hi-C maps depends significantly on the quality of the SNP annotation used.

VII.b. Properties of Diploid Hi-C maps

VII.b.1. Analysis of phased interchromosomal interactions reveals chromosomal organization as well as a unique unbalanced translocation: To create the map of phased interchromosomal interactions seen in Figure 7B, we looked at contacts that overlap SNPs on both read ends. We then created a 46x46 matrix consisting of the maternal/paternal homolog of each chromosome. We excluded intrachromosomal reads, and normalized the matrix using the KR algorithm. We then divided by the mean of the matrix to obtain the observed/expected matrix seen in the figure.

When we examined this matrix, we found that homologs of chromosomes 16, 17, 19, 20, 21, and 22 formed a strongly interacting group in which each homolog tends to form contacts with its partner homolog and with both homologs of the other chromosomes in the group. This implies that homologs are enriched for interactions with each other and cannot be factored into two clusters. This finding is consistent with the interchromosomal contact pattern we previously reported based on our low-resolution Hi-C experiments (Lieberman-Aiden et al., 2009).

We observed that there was a high rate of interchromosomal contacts between the paternal chromosome 6 and paternal chromosome 11, but not between any of the other three 6/11 homolog pairs. We suspected that this was due to a novel translocation in some fraction of the cells used to create the Hi-C libraries. Examination of the interchromosomal chr6-chr11 Hi-C contact matrix suggested an unbalanced translocation between chromosome 11 and chromosome 6 (Figure S7B). We estimated the incidence of the translocation by comparing the intensity in the pixels between the fused loci at 1 Mb matrix resolution (chr6, 0-1 Mb; chr11, 72-74 Mb) to the intensity of pixels on the diagonal or 1 Mb off-diagonal in the chromosome 6 intra-chromosomal contact matrix. We observed an average of 4740 contacts in the pixels corresponding to the translocation versus an average of 720,599 contacts on the diagonal of chromosome 6 and 166,879 contacts 1 Mb off of the diagonal. Assuming that the translocation occurs in at most 1 allele per cell (the paternal allele) and that most of the 4740 contacts come from the translocation, this suggests that the translocation is present in between 0.6% and 2.8% of alleles, or between 1.2% and 5.6% of cells. Karyotyping via Giemsa staining (Figure S7C-D) identified 3 out of 100 cells with an unbalanced translocation between chromosome 6 and chromosome 11 on one of the chromosome 6 homologs (notably, Giemsa staining could not identify which homolog, maternal or paternal, had the translocation). No other translocations were seen in any of the 100 cells examined. Additionally, we confirmed the unbalanced translocation

between chromosome 6 and chromosome 11 via spectral karyotyping (Figure S7E-F). Thus, diploid Hi-C maps allow for the identification of low-frequency translocations and can determine the specific homologs involved.

Notably, there was no evidence of the translocation in the Hi-C maps generated using GM12878 "Batch 1" (see Section *I.e.*). In contrast, nearly all Hi-C libraries derived from "Batch 2" showed clear evidence of the translocation. (This observation is therefore of relevance to users of ENCODE GM12878 data, who – given the modest karyotypic variability between batches – may be interested in determining which batch a given ENCODE experiment used.)

VII.b.2. Diploid maps of individual homologs reveal the existence of massive "superdomains" partitioning the inactive X chromosome: Our diploid Hi-C maps allow us to examine the three-dimensional structure of the maternal active X and paternal inactive X separately in GM12878. Notably, we find that the paternal inactive X is partitioned into two massive domains (Figure 7D). Such differences between two homologs were not seen in the autosomes (Figure S7A). We hypothesized that any cell line with an inactive X (i.e. female cell lines) would retain the trace signature of the superdomains on the haploid map of chromosome X. We found that the superdomains were visible in the haploid interaction matrix for chromosome X in all of our karyotypically normal female cell lines (GM12878, IMR90, HMEC and NHEK) but not visible in KBM7, which is a haploid cell line and lacks an inactive X chromosome or in HUVEC, which is a male cell line (Figure S7G-H).

VII.b.3. Diploid maps reveal the existence of long-range "superloops" on the inactive X chromosome: When we examined the unphased GM12878 X-chromosome Hi-C map, we observed the presence of visually obvious, extremely long-range (tens of Mb) loops anchored by large loci (100 kb). These loops were seen on the paternal diploid map, but not on the maternal diploid map; similarly, they were seen in all karyotypically normal XX cell lines, but not in X0 or XY cell lines (See Fig S7I). No such phenomenon was observed on any other chromosome.

To analyze these long-range loops systematically, we used HiCCUPS with slightly different parameters than we used for identifying normal peaks. We called peaks that spanned 5 Mb or more at 50 kb resolution using a window size of 4 pixels by 4 pixels, and a peak width of 2 pixels (see Section *VI.a.5.i*). In GM12878, we used an FDR of .01%; in IMR90, KBM7 and HUVEC we used an FDR of 1% (as our data was sparser).

In GM12878, the algorithm identified 27 long-range loops between 24 anchor regions. Six anchor regions participated in 3 or more long-range loops. We examined all 27 peaks in both the maternal and paternal GM12878 maps; since these maps are much sparser than the unphased map, we performed the comparisons at 100 kb resolution. The 27 peaks were enriched in the paternal map relative to the maternal map (paired one-sided t-test, $p<10^{-4}$). When we ran the algorithm on IMR90 (XX), 8 superloops were found; 6 of them were also identified in GM12878. The algorithm identified no superloops in either HUVEC (XY) or KBM7 (X0).

VIII. Supplemental Tables

Table S1. Hi-C Experiments, Related to Fig. 1

See attached Excel file.

Table S2. Quality Metrics, Related to Fig. 1

	Primary (HIC*_br1)	Bio Rep 1 (HIC*_br2)	Bio Rep 2 (HIC*_br3)	Bio Rep 3 (HIC*_br4)	Bio Rep4 (HIC*_br5)	Bio Rep5 (HIC*_br6)	Bio Rep6 (HIC*_br7)	Bio Rep7 (HIC*_br8)	Bio Rep8 (HIC*_br9)
Sequenced Reads	3.6B	314M	389M	178M	669M	112M	705M	328M	240M
Normal Paired	2.7B (75%)	244M (78%)	305M (78%)	124M (70%)	477M (71%)	82M (74%)	531M (75%)	244M (74%)	180M (75%)
Chimeric Paired	563M (16%)	48M (15%)	59M (15%)	41M (23%)	124M (18%)	20M (18%)	121M (17%)	58M (18%)	36M (15%)
Chimeric Ambiguous	153M (4%)	11M (4%)	12M (3%)	7M (4%)	26M (4%)	4M (4%)	26M (4%)	15M (5%)	9M (4%)
Unmapped	187M (5%)	11M (4%)	13M (3%)	6M (3%)	43M (6%)	5M (4%)	27M (4%)	11M (3%)	15M (6%)
Alignable Reads	3.2B (90%)	291M (93%)	364M (93%)	165M (93%)	600M (90%)	103M (92%)	652M (93%)	302M (92%)	217M (90%)
Duplicates	300M (8%)	42M (13%)	15M (4%)	3M (2%)	18M (3%)	1M (1%)	13M (2%)	7M (2%)	5M (2%)
Unique Reads	2.9B (81% / 100%)	250M (80% / 100%)	348M (89% / 100%)	163M (91% / 100%)	582M (87% / 100%)	101M (91% / 100%)	639M (91% / 100%)	295M (90% / 100%)	212M (88% / 100%)
Intra-fragment	43M (1% / 1%)	2M (1% / 1%)	20M (5% / 6%)	6M (3% / 4%)	26M (4% / 5%)	4M (3% / 4%)	11M (2% / 2%)	5M (2% / 2%)	2M (1% / 1%)
Low Mapping Quality	268M (7% / 9%)	22M (7% / 9%)	31M (8% / 9%)	15M (8% / 9%)	55M (8% / 9%)	10M (9% / 10%)	59M (8% / 9%)	27M (8% / 9%)	21M (9% / 10%)
HiC Contacts	2.6B (73% / 89%)	226M (72% / 91%)	297M (76% / 85%)	142M (79% / 87%)	501M (75% / 86%)	88M (78% / 86%)	569M (81% / 89%)	262M (80% / 89%)	188M (78% / 89%)
Inter chromosomal	644M (18% / 22%)	51M (16% / 20%)	70M (18% / 20%)	31M (18% / 19%)	105M (16% / 18%)	22M (20% / 22%)	178M (25% / 28%)	65M (20% / 22%)	58M (24% / 27%)
Intra chromosomal	2B (55% / 68%)	176M (56% / 70%)	227M (58% / 65%)	110M (62% / 68%)	395M (59% / 68%)	66M (59% / 65%)	391M (56% / 61%)	198M (60% / 67%)	131M (54% / 62%)
Intra Short Range (<20 kb)	602M (17% / 20%)	47M (15% / 19%)	82M (21% / 23%)	38M (21% / 24%)	147M (22% / 25%)	22M (20% / 22%)	106M (15% / 17%)	62M (19% / 21%)	37M (15% / 17%)
Intra Long Range (≥20 kb)	1.4B (39% / 47%)	129M (41% / 51%)	145M (37% / 42%)	72M (40% / 44%)	248M (37% / 43%)	44M (39% / 43%)	285M (40% / 45%)	136M (41% / 46%)	94M (39% / 44%)
Ligations	950M (27% / 32%)	80M (26% / 32%)	97M (25% / 28%)	76M (42% / 47%)	222M (33% / 38%)	38M (34% / 37%)	225M (32% / 35)	110M (33% / 37%)	74M (31% / 35%)
3' bias (long range)	68% - 32%	69% - 31%	69% - 31%	75% - 25%	72% - 28%	72% - 28%	72% - 28%	70% - 30%	73% - 27%
Read Pair type (L-I-O-R)	25-25-25-25	25-25-25-25	25-25-25-25	25-25-25-25	25-25-25-25	25-25-25-25	25-25-25-25	25-25-25-25	25-25-25-25

Table S3. External Datasets Used, Related to Fig. 2, Fig. 5, Fig. 6

See attached Excel file.

Table S4. Sensitivity/Specificity of HICCUPS GM12878 Peaks, Related to Fig. 3

ENCODE REGION	Number of hand annotated peaks	HiCCUPS (annotations reported in text, overlap with hand annotation in parentheses)	HiCCUPS (only 1.5 enrichment required over all expectations, overlap with hand annotation in parentheses)	HiCCUPS (no local enrichment filtering, overlap with hand annotation in parentheses)	BH-FDR (2-fold enrichment required)	BH-FDR (1.75 fold enrichment required)	BH-FDR (1.5 fold enrichment required)	BH-FDR (no local enrichment filtering)
ENm001	14	13 (11)	15 (12)	16 (13)	7 (6)	9 (6)	11 (6)	15 (6)
ENm002	5	5 (2)	8 (4)	10 (4)	2 (1)	3 (1)	7 (1)	13 (1)
ENm003	0	0 (0)	0 (0)	2 (0)	0 (0)	0 (0)	0 (0)	0 (0)
ENm004	5	6 (4)	8 (5)	13 (5)	10 (4)	13 (5)	22 (5)	24 (5)
ENm005	8	7 (4)	8 (5)	18 (7)	10 (4)	18 (4)	28 (5)	33 (5)
ENm006	3	4 (3)	6 (3)	9 (3)	2 (1)	2 (1)	4 (2)	4 (2)
ENm007	2	2 (0)	2 (0)	4 (2)	1 (0)	4 (0)	5 (0)	8 (1)
ENm008	3	2 (2)	3 (2)	3 (2)	0 (0)	1 (0)	2 (0)	2 (0)
ENm009	2	2 (2)	3 (2)	3 (2)	2 (2)	3 (2)	4 (2)	4 (2)
ENm010	0	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
ENm011	3	5 (3)	7 (3)	7 (3)	0 (0)	0 (0)	3 (1)	4 (1)
ENm012	2	1 (1)	1 (1)	2 (1)	1 (1)	2 (2)	5 (2)	5 (2)
ENm013	5	4 (4)	5 (4)	7 (4)	4 (4)	6 (4)	12 (4)	13 (4)
ENm014	1	0 (0)	1 (0)	2 (0)	2 (0)	2 (0)	4 (0)	7 (0)
Total	53	51 (37)	67 (41)	96 (46)	41 (23)	63 (25)	107 (28)	132 (29)

Table S5. FISH, Related to Fig. 3

Peak	Chromosome	L1 (position, Mb)	# of oligos tiling L1	L2 (position, Mb)	# of oligos tiling L2	L3 (position, Mb)	# of oligos tiling L3
1	17	66.76-66.79	337	67.22-67.25	338	67.68-67.71	328
2	14	71.60-71.63	328	72.20-72.23	404	72.80-72.83	390
3	11	130.72-130.75	424	130.29-130.32	420	129.86-129.89	279
4	13	86.37-86.40	292	85.46-85.49	267	84.55-84.58	307

Table S6. APA Scores, Related to Fig. 3

Hi-C Map	*In situ* Peak List	APA	Z-score	P-value
Dilution Hi-C GM12878 (HindIII)	*In situ* GM12878	1.964	35.960	1.74E-283
In situ Hi-C (HIC043) GM12878 (no crosslinking)	*In situ* GM12878	2.838	26.241	4.58E-152
In situ Hi-C (HIC044) GM12878 (no crosslinking)	*In situ* GM12878	2.030	14.586	1.72E-48
In situ Hi-C (HIC045) GM12878 (no crosslinking)	*In situ* GM12878	3.063	8.814	6.02E-19
In situ Hi-C (HIC046) GM12878 (no crosslinking)	*In situ* GM12878	2.105	15.962	1.17E-57
In situ Hi-C (HIC047) GM12878 (no crosslinking)	*In situ* GM12878	2.114	8.060	3.80E-16
Kalhor et al. TCC GM12878 (HindIII)	*In situ* GM12878	2.268	27.643	1.67E-168
Kalhor et al. TCC GM12878 (MboI)	*In situ* GM12878	2.372	22.581	3.37E-113
Kalhor et al. GM12878 (HindIII)	*In situ* GM12878	2.002	14.812	6.08E-50
Lieberman-Aiden et al. GM06990 (HindIII 1)	*In situ* GM12878	1.351	2.238	1.26E-02
Lieberman-Aiden et al. GM06990 (HindIII 2)	*In situ* GM12878	1.806	4.385	5.80E-06
Lieberman-Aiden et al. GM06990 (NcoI)	*In situ* GM12878	1.910	10.111	2.47E-24
Dixon et al. IMR90	*In situ* IMR90	1.813	27.269	5.01E-164
Jin et al. IMR90	*In situ* IMR90	1.921	29.136	6.21E-187
Dilution Hi-C IMR90 (HindIII)	*In situ* IMR90	1.772	18.482	1.43E-76
Dilution Hi-C HMEC (HindIII)	*In situ* HMEC	1.749	20.050	1.02E-89
Dilution Hi-C HUVEC (HindIII)	*In situ* HUVEC	1.711	21.854	3.52E-106
Dilution Hi-C NHEK (HindIII)	*In situ* NHEK	1.506	11.328	4.77E-30
Lieberman-Aiden et al. K562	*In situ* K562	1.573	4.447	4.36E-06
Dilution Hi-C Mouse Lymph (HindIII 1)	*In situ* Mouse Lymph	1.727	10.828	1.26E-27
Dilution Hi-C Mouse Lymph (HindIII 2)	*In situ* Mouse Lymph	2.286	9.348	4.46E-21
Hi-C Map	**External Peak List**	**APA**	**Z-score**	**P-value**
In situ Hi-C (HIC043) GM12878 (no crosslinking)	ENCODE 5C GM12878	0.711	-1.595	9.45E-01
In situ Hi-C (HIC044) GM12878 (no crosslinking)	ENCODE 5C GM12878	0.832	-1.075	8.59E-01
In situ Hi-C (HIC045) GM12878 (no crosslinking)	ENCODE 5C GM12878	0.601	-0.710	7.61E-01
In situ Hi-C (HIC046) GM12878 (no crosslinking)	ENCODE 5C GM12878	0.920	-0.502	6.92E-01
In situ Hi-C (HIC047) GM12878 (no crosslinking)	ENCODE 5C GM12878	1.206	0.371	3.55E-01
Kalhor et al. TCC GM12878 (HindIII)	ENCODE 5C GM12878	0.813	-1.991	9.77E-01
Kalhor et al. TCC GM12878 (MboI)	ENCODE 5C GM12878	0.918	-0.481	6.85E-01
Kalhor et al. GM12878 (HindIII)	ENCODE 5C GM12878	1.094	0.564	2.86E-01
Lieberman-Aiden et al. GM06990 (HindIII 1)	ENCODE 5C GM12878	0.159	-1.839	9.67E-01
Lieberman-Aiden et al. GM06990 (HindIII 2)	ENCODE 5C GM12878	0.257	-1.142	8.73E-01
Lieberman-Aiden et al. GM06990 (NcoI)	ENCODE 5C GM12878	0.939	-0.191	5.76E-01
In situ GM12878	ENCODE 5C GM12878	0.826	-2.683	9.96E-01
In situ HeLa	ENCODE 5C HeLa	0.710	-4.136	1.00E+00
In situ K562	ENCODE 5C K562	0.839	-2.277	9.89E-01
Lieberman-Aiden et al. K562	ENCODE 5C K562	1.078	0.240	4.05E-01
Dilution Hi-C IMR90 (HindIII)	Jin et al. Hi-C IMR90	1.001	0.053	4.79E-01
In situ IMR90	Jin et al. Hi-C IMR90	1.043	1.346	8.91E-02
In situ K562	Li et al. *Pol*II ChIA-PET K562	0.969	-0.557	7.11E-01
Lieberman-Aiden et al. K562	Li et al. *Pol*II ChIA-PET K562	0.988	-0.118	5.47E-01
In situ GM12878	Thurman et al. DHS pairs GM12878	0.615	-3.903	1.00E+00
In situ K562	Li et al. CTCF ChIA-PET K562	1.741	20.053	9.43E-90
Lieberman-Aiden et al. K562	Li et al. CTCF ChIA-PET K562	1.394	4.807	7.65E-07

Table S7. Differential Gene Expression, Related to Fig. 5

Cell type with differential loop (CT1)	Cell type without differential loop (CT2)	# of differential loops	# of genes with promoter at peak loci	# of genes with 2-fold higher expression in CT1	# of genes with 2-fold higher expression in CT2	# of genes with 5-fold higher expression in CT1	# of genes with 5-fold higher expression in CT2	# of genes with 10-fold higher expression in CT1	# of genes with 10-fold higher expression in CT2	# of genes with 50-fold higher expression in CT1	# of genes with 50-fold higher expression in CT2
IMR90	GM12878	510	450	153	17	133	7	117	6	94	3
HMEC	GM12878	439	414	96	61	86	22	80	11	63	6
K562	GM12878	323	323	44	18	22	6	17	5	9	5
HUVEC	GM12878	192	232	50	22	43	13	41	7	34	3
HeLa	GM12878	242	240	38	25	27	10	23	7	15	1
NHEK	GM12878	446	355	42	16	36	7	29	4	19	3
GM12878	IMR90	557	455	96	27	68	9	56	6	43	1
GM12878	HMEC	640	593	170	19	95	7	66	4	45	3
GM12878	K562	737	660	145	32	109	15	88	10	71	3
GM12878	HUVEC	917	881	213	88	136	43	82	33	52	16
GM12878	HeLa	1050	1033	161	71	104	27	91	19	74	9
GM12878	NHEK	910	952	182	77	112	32	80	24	51	10

Table S8. Allele Assignments, Related to Fig. 7

Total reads overlapping SNPs	488,076,083
Overlap SNP on one read end	476,383,010
Maternal	233,469,859
Paternal	234,957,021
Neither	7,956,130
Overlap SNP on both read ends	11,693,073
Maternal agreement	4,671,619
Paternal agreement	4,705,734
Maternal/Paternal disagreement	1,797,407
Neither	518,313
Total maternal contacts	238,141,478
Total paternal contacts	239,662,755

XI. References

Banerji, J., Rusconi, S., and Schaffner, W. (1981). Expression of a beta-globin gene is enhanced by remote SV40 DNA sequences. Cell *27*, 299-308.

Barski, A., Cuddapah, S., Cui, K., Roh, T.-Y., Schones, D., Wang, Z., Wei, G., Chepelev, I., and Zhao, K. (2007). High-resolution profiling of histone methylations in the human genome. Cell *129*, 823-837.

Beliveau, B., Joyce, E., Apostolopoulos, N., Yilmaz, F., Fonseka, C., McCole, R., Chang, Y., Li, J., Senaratne, T., Williams, B.*, et al.* (2012). Versatile design and synthesis platform for visualizing genomes with Oligopaint FISH probes. Proceedings of the National Academy of Sciences of the United States of America *109*, 21301-21306.

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B (Methodological), 289-300.

Brown, J.B., Chase, P.J., and Pittenger, A.O. (1993). Order independence and factor convergence in iterative scaling. Linear Algebra Applications *190*, 7-11.

Cook, P.R., and Brazell, I.A. (1975). Supercoils in human DNA. Journal of cell science *19*, 261-279.

Cournac, A., Marie-Nelly, H., Marbouty, M., Koszul, R., and Mozziconacci, J. (2012). Normalization of a chromosomal contact map. BMC genomics *13*, 436.

Csiszár, I. (1975). *I*-Divergence Geometry of Probability Distributions and Minimization Problems. The Annals of Probability *3*, 146-158.

Cullen, K., Kladde, M., and Seyfred, M. (1993). Interaction between transcription regulatory regions of prolactin chromatin. Science *261*, 203-206.

Dekker, J., Rippe, K., Dekker, M., and Kleckner, N. (2002). Capturing chromosome conformation. Science *295*, 1306-1311.

Deming, W.E., and Stephan, F.F. (1940). On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known. The Annals of Mathematical Statistics *11*, 427-444.

Dixon, J., Selvaraj, S., Yue, F., Kim, A., Li, Y., Shen, Y., Hu, M., Liu, J., and Ren, B. (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature *485*, 376-380.

Donohoe, M.E., Zhang, L.F., Xu, N., Shi, Y., and Lee, J.T. (2007). Identification of a Ctcf cofactor, Yy1, for the X chromosome binary switch. Mol Cell *25*, 43-56.

Dostie, J., Richmond, T., Arnaout, R., Selzer, R., Lee, W., Honan, T., Rubio, E., Krumm, A., Lamb, J., Nusbaum, C.*, et al.* (2006). Chromosome Conformation Capture Carbon Copy (5C): a massively parallel solution for mapping interactions between genomic elements. Genome research *16*, 1299-1309.

Dunn, T., Hahn, S., Ogden, S., and Schleif, R. (1984). An operator at -280 base pairs that is required for repression of araBAD operon promoter: addition of DNA helical turns between the operator and promoter cyclically hinders repression. Proceedings of the National Academy of Sciences of the United States of America *81*, 5017-5020.

Eismann, E., von Wilcken-Bergmann, B., and Müller-Hill, B. (1987). Specific destruction of the second lac operator decreases repression of the lac operon in Escherichia coli fivefold. Journal of molecular biology *195*, 949-952.

Gerstein, M.B., Kundaje, A., Hariharan, M., Landt, S.G., Yan, K.K., Cheng, C., Mu, X.J., Khurana, E., Rozowsky, J., Alexander, R.*, et al.* (2012). Architecture of the human regulatory network derived from ENCODE data. Nature *489*, 91-100.

Gil, A.M., David, M.A., Richard, M.D., Gonçalo, R.A., David, R.B., Aravinda, C., Andrew, G.C., Peter, D., Evan, E.E., Paul, F.*, et al.* (2012). An integrated map of genetic variation from 1,092 human genomes. Nature *491*.

Goldman, M. (1988). The chromatin domain as a unit of gene regulation. BioEssays : news and reviews in molecular, cellular and developmental biology *9*, 50-55.

Griffith, J., Hochschild, A., and Ptashne, M. (1986). DNA loops induced by cooperative binding of lambda repressor. Nature *322*, 750-752.

Hansen, R.S., Thomas, S., Sandstrom, R., Canfield, T.K., Thurman, R.E., Weaver, M., Dorschner, M.O., Gartler, S.M., and Stamatoyannopoulos, J.A. (2010). Sequencing newly replicated DNA reveals widespread plasticity in human replication timing. Proceedings of the National Academy of Sciences of the United States of America *107*, 139-144.

Heidari, N., Phanstiel, D.H., He, C., Grubert, F., Jahanbanian, F., Kasowski, M., Zhang, M.Q., and Snyder, M.P. (2014). Genome-wide map of regulatory interactions in the human genome. Genome research.

Hinrichs, A.S., Karolchik, D., Baertsch, R., Barber, G.P., Bejerano, G., Clawson, H., Diekhans, M., Furey, T.S., Harte, R.A., Hsu, F., *et al.* (2006). The UCSC Genome Browser Database: update 2006. Nucleic Acids Res *34*, D590-598.

Hu, M., Deng, K., Qin, Z., Dixon, J., Selvaraj, S., Fang, J., Ren, B., and Liu, J. (2013). Bayesian inference of spatial organizations of chromosomes. PLoS computational biology *9*.

Illumina (2007). Preparing Samples for Sequencing Genomic DNA Using the Genomic DNA Sample Prep Oligo Only Kit (Part # 1003492 Rev. A).

Imakaev, M., Fudenberg, G., McCord, R., Naumova, N., Goloborodko, A., Lajoie, B., Dekker, J., and Mirny, L. (2012). Iterative correction of Hi-C data reveals hallmarks of chromosome organization. Nature methods *9*, 999-1003.

Jin, F., Li, Y., Dixon, J., Selvaraj, S., Ye, Z., Lee, A., Yen, C.-A., Schmitt, A., Espinoza, C., and Ren, B. (2013). A high-resolution map of the three-dimensional chromatin interactome in human cells. Nature *503*, 290-294.

Kadauke, S., and Blobel, G. (2009). Chromatin loops in gene regulation. Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms *1789*, 17-25.

Kalhor, R., Tjong, H., Jayathilaka, N., Alber, F., and Chen, L. (2012). Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. Nature biotechnology *30*, 90-98.

Kim, T., Abdullaev, Z., Smith, A., Ching, K., Loukinov, D., Green, R., Zhang, M., Lobanenkov, V., and Ren, B. (2007). Analysis of the vertebrate insulator protein CTCF-binding sites in the human genome. Cell *128*, 1231-1245.

Knight, P. (2008). The Sinkhorn-Knopp algorithm: Convergence and applications. SIAM Journal on Matrix Analysis and Applications *30*, 261-275.

Knight, P., and Ruiz, D. (2012). A fast algorithm for matrix balancing. IMA Journal of Numerical Analysis.

Krämer, H., Niemöller, M., Amouyal, M., Revet, B., von Wilcken-Bergmann, B., and Müller-Hill, B. (1987). lac repressor forms loops with linear DNA carrying two suitably spaced lac operators. EMBO Journal *6*, 1481-1491.

Kruithof, J. (1937). Telefoonverkeersrekening. De Ingenieur *52*, E15-E25.

Lander, E.S., and Waterman, M.S. (1988). Genomic mapping by fingerprinting random clones: a mathematical analysis. Genomics *2*, 231-239.

Landt, S.G., Marinov, G.K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., Bernstein, B.E., Bickel, P., Brown, J.B., Cayting, P., *et al.* (2012). ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. Genome Res *22*, 1813-1831.

Li, G., Ruan, X., Auerbach, R., Sandhu, K., Zheng, M., Wang, P., Poh, H., Goh, Y., Lim, J., Zhang, J., *et al.* (2012). Extensive promoter-centered chromatin interactions provide a topological basis for transcription regulation. Cell *148*, 84-98.

Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics *25*, 1754-1760.

Lieberman-Aiden, E., van Berkum, N., Williams, L., Imakaev, M., Ragoczy, T., Telling, A., Amit, I., Lajoie, B., Sabo, P., Dorschner, M., *et al.* (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science *326*, 289-293.

McCord, R., Nazario-Toole, A., Zhang, H., Chines, P., Zhan, Y., Erdos, M., Collins, F., Dekker, J., and Cao, K. (2013). Correlated alterations in genome organization, histone methylation, and DNA-lamin A/C interactions in Hutchinson-Gilford progeria syndrome. Genome research *23*, 260-269.

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., *et al.* (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res *20*, 1297-1303.

Mukherjee, S., Erickson, H., and Bastia, D. (1988). Enhancer-origin interaction in plasmid R6K involves a DNA loop mediated by initiator protein. Cell *52*, 375-383.

Myslinski, E., Gerard, M.A., Krol, A., and Carbon, P. (2006). A genome scale location analysis of human Staf/ZNF143-binding sites suggests a widespread role for human Staf/ZNF143 in mammalian promoters. The Journal of biological chemistry *281*, 39953-39962.

Naughton, C., Avlonitis, N., Corless, S., Prendergast, J., Mati, I., Eijk, P., Cockroft, S., Bradley, M., Ylstra, B., and Gilbert, N. (2013). Transcription forms and remodels supercoiling domains unfolding large-scale chromatin structures. Nature structural & molecular biology *20*, 387-395.

Németh, A., Conesa, A., Santoyo-Lopez, J., Medina, I., Montaner, D., Péterfia, B., Solovei, I., Cremer, T., Dopazo, J., and Längst, G. (2010). Initial genomics of the human nucleolus. PLoS Genetics *6*.

Pathak, S. (1976). Chromosome banding techniques. The Journal of reproductive medicine *17*, 25-28.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V.*, et al.* (2011). Scikit-lern: Machine Learning in Python. Journal of Machine Learning Research *12*, 2825-2830.

Phillips-Cremins, J.E., Sauria, M.E.G., Sanyal, A., Gerasimove, T.I., Lajoie, B.R., Bell, J.S.K., Ong, C.-T., Hookway, T.A., Guo, C., Sun, Y.*, et al.* (2013). Architectural Protein Subclasses Shape 3D Organization of Genomes during Lineage Commitment. Cell *153*, 1281-1295.

Ptashne, M. (1986). Gene regulation by proteins acting nearby and at a distance. Nature *322*, 697-701.

Ryba, T., Hiratani, I., Lu, J., Itoh, M., Kulik, M., Zhang, J., Schulz, T., Robins, A., Dalton, S., and Gilbert, D. (2010). Evolutionarily conserved replication timing profiles predict long-range chromatin interactions and distinguish closely related cell types. Genome research *20*, 761-770.

Sanyal, A., Lajoie, B., Jain, G., and Dekker, J. (2012). The long-range interaction landscape of gene promoters. Nature *489*, 109-113.

Schleif, R. (1992). DNA looping. Annual review of biochemistry *61*, 199-223.

Schneider, C.A., Rasband, W.S., and Eliceiri, K.W. (2012). NIH Image to ImageJ: 25 years of image analysis. Nature methods *9*, 671-675.

Schones, D.E., Smith, A.D., and Zhang, M.Q. (2007). Statistical significance of cis-regulatory modules. BMC bioinformatics *8*, 19.

Sexton, T., Yaffe, E., Kenigsberg, E., Bantignies, F., Leblanc, B., Hoichman, M., Parrinello, H., Tanay, A., and Cavalli, G. (2012). Three-dimensional folding and functional organization principles of the Drosophila genome. Cell *148*, 458-472.

Sinkhorn, R., and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics *21*, 343-348.

van Koningsbruggen, S., Gierlinski, M., Schofield, P., Martin, D., Barton, G.J., Ariyurek, Y., den Dunnen, J.T., and Lamond, A.I. (2010). High-resolution whole-genome sequencing reveals that specific chromatin domains from most human chromosomes associate with nucleoli. Molecular biology of the cell *21*, 3735-3748.

Vogel, M.J., Guelen, L., de Wit, E., and Hupkes, D.P. (2006). Human heterochromatin proteins form large domains containing KRAB-ZNF genes. Genome ….

Vogelstein, B., Pardoll, D.M., and Coffey, D.S. (1980). Supercoiled loops and eucaryotic DNA replicaton. Cell *22*, 79-85.

Yaffe, E., and Tanay, A. (2011). Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. Nature genetics *43*, 1059-1065.

Zehnbauer, B.A., and Vogelstein, B. (1985). Supercoiled loops and the organization of replication and transcription in eukaryotes. BioEssays.

Zhang, Y., Liu, T., Meyer, C.A., Eeckhoute, J., Johnson, D.S., Bernstein, B.E., Nusbaum, C., Myers, R.M., Brown, M., Li, W.*, et al.* (2007). Model-based analysis of ChIP-Seq (MACS). Genome biology *9*.

Zhao, Z., Tavoosidana, G., Sjölinder, M., Göndör, A., Mariano, P., Wang, S., Kanduri, C., Lezcano, M., Sandhu, K., Singh, U.*, et al.* (2006). Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions. Nature genetics *38*, 1341-1347.

Zuin, J., Dixon, J., van der Reijden, M., Ye, Z., Kolovos, P., Brouwer, R.W., van de Corput, M., van de Werken, H., Knoch, T., van Ijcken, W.*, et al.* (2014). Cohesin and CTCF differentially affect chromatin architecture and gene expression in human cells. Proceedings of the National Academy of Sciences of the United States of America *111*, 996-1001.