Zhan *et al.,* 'Reciprocal insulation analysis of Hi-C data show that TADs represent a functionally but not structurally privileged scale in the hierarchical folding of chromosomes'

Supplemental Methods

Hi-C datasets

ESCs and NPCs Hi-C datasets were obtained in Ref. (Giorgetti et al. 2016). Reads from 129Sv and Cast/EiJ alleles were combined to increase read depth, and data were binned at 20 kb resolution. CH12 data are from Rao *et al.* (Rao et al. 2014), binned at 10 kb. Mouse fetal liver Hi-C data are from Nagano *et al.* (Nagano et al. 2015), binned at 25 kb. ESC, NPC and liver Hi-C were normalized with iterative correction (Imakaev et al. 2012). CH12 were normalized with the VC-SQRT method (Rao et al. 2014).

Domain-calling algorithm

The CaTCH algorithm takes as an input a normalized Hi-C matrix, binned at an arbitrary resolution r. The genome is first partitioned into seeds of domains of size 2*r, which are then progressively merged into large domains. Merging of two consecutive domains A and B is determined by the reciprocal insulation (RI) measure:

$$RI(A,B)=[\ P_{in}(A)+P_{in}(B)-P_{out}(A,B)\ ]/\ [P_{in}(A)+P_{in}(B)]*100 \quad (1)$$

Where P_{in} and P_{out} are the average Hi-C counts within domains A and B, and across their boundary respectively (see **Figure 1a** in the main text).

A threshold on RI is then defined, and any two consecutive domains whose RI is below the threshold are merged in a single domain. The threshold is progressively increased from 0% to 100% in steps of 0.1%, resulting in increasingly larger domains. The fact that only consecutive domains can be merged ensures that the overall organization of the domains is tree-like, excluding the possibility of interactions between distant domains. This could be observed otherwise by imposing a different distance based on the Hi-C map, which is not strictly ultrametric. In order to lose dependency on the initial partitioning of the genome in the final determination of domain boundaries, we allowed small shifts in the boundaries of domains (2 genomic bins) at each step. Note that the domains identified by CaTCH do not depend on bin size, provided the domain is larger than the genomic bin.

Since the increase of the threshold is discrete, the above procedure undergoes the risk of being dependent on the order of mergings, which would result in a non-unique tree. To overcome this problem, we set a specific rule on the matching order. Namely, if a domain can be merged with either the preceding or the following along, the pair that has the lowest RI is merged first. This is in fact equivalent to merging domains according to their order along the chromosomes, and increasing smoothly (rather than in discrete steps) the threshold on the reciprocal insulation value. Indeed, smoothly increasing the threshold corresponds to cutting the hierarchical tree densely enough that one is able to always merge the domains with the lowest RI.

Computationally generated contact maps with preferential folding levels

To generate contact maps characterized by one or two preferential folding levels, we generated a contact map for each individual level (where contact probabilities decrease as a power law with increasing genomic distance), to which a weak background Gaussian noise was added. For example, to generate a pseudo-genome with two folding levels (see right panel in main **Figure 1f**), we first generated a uniform (power-law decaying) contact map with Gaussian noise. Then, we partitioned the matrix into a set of small domains **d1**=$\{d1_i\}$ (smallest

squares along the diagonal in **Figure 1f**). The first folding level was generated within this set of domains by adding a new power-law decreasing interaction pattern. We then merged pairs of adjacent domains (e.g. $d1_1$ with $d1_2$; $d1_3$ with $d1_4$ and so on) leading to a second set of domains **d2**=$\{d2_i\}$ to which the same power-law decreasing interaction pattern was added. The contact map with no folding layer was generated by replacing the actual Hi-C counts in the contact map for chr19 in ESCs with the average genome-wide counts for oci with the same genomic distance, and adding Gaussian noise.

CTCF motif analysis

We called CTCF peaks using macs2 (Zhang et al. 2008) using default parameters. We used the top 1000 high-significance peaks to define a CTCF position-weight matrix, resulting in a PWM that is indistinguishable from previous reports (Jaspar accession number MA0139.1; see Ref. (Mathelier et al. 2013)). We then used MEME tool (Bailey and Elkan 1994) with a custom background, which includes non-overlapping mappable sequences with the same distribution size of the top 1000 peaks, to perform *de novo* motif discovery. Finally, we used the motif identified within the top 1000 CTCF peaks called by macs2 to extract the position and directionality of CTCF-bound sites among all the peaks using the MAST tool (Bailey and Gribskov 1998).

Boundary conservation

In order to identify the fraction of boundaries that are conserved either between cell types or domain sets, we allowed a 40-kb tolerance in boundary conservation between contact domains and sets of domains in the hierarchy of CH12 cells; for comparison with compartments we allowed a tolerance of 750kb; for all the other comparisons, we allowed a tolerance of 100-kb.

Cell culture

The female mouse ES cell line F121.6 (129Sv-Cast/EiJ) was grown on mitomycin C-inactivated MEFs in ES cell media containing 15% FBS (Gibco), 10^{-4}M ß-mercaptoethanol (Sigma), and 1000U/ml of leukaemia inhibitory factor (LIF, Chemicon). Culture of the same NPC clone that was analyzed in wa(Giorgetti et al. 2016)s performed as previously described (Gendrel et al. 2014; Giorgetti et al. 2016). All cells used in this study were characterized for absence of mycoplasma contamination.

RNA-seq data, analysis and transcript annotation

After Trizol extraction, strand-specific total RNA-seq libraries from two biological replicates for both ESCs and NPCs were prepared with the ScriptSeq v2 kit (Illumina) and sequenced on an Illumina HiSeq 2000 for a total of ~30 million uniquely aligned reads per sample on average. Libraries were prepared in two technical replicates per biological replicate (technical replicates were pooled for subsequent analyses). All samples were aligned to mouse mm9 using QuasR (Gaidatzis et al. 2015) keeping uniquely mappable reads only. A complete list of all non-overlapping known genes from UCSC (Carlson M and Maintainer BP. *TxDb.Mmusculus.UCSC.mm9.knownGene: Annotation package for TxDb object(s)*. R package version 3.2.2) was used to quantify both exonic and intronic transcription. Levels were estimated by separately aligning the reads to exonic and intronic regions and quantifying RPKMs as

$$RPKM=M/(N*L)*1'000*1'000'000$$

Where M is the mapped reads to the genomic region, L is the length of the region (sum of all exons or introns for each gene) and N is the total number of mapped reads.

We used the DESeq2 package (Love et al. 2014) to perform differential gene expression analysis between ESCs and NPCs. Cutoff on q-value<=0.05 and on a fold-change larger than 3 were used to define differentially expressed genes.

ChIP-seq analysis

We analysed the available ChIP-seq datasets listed in **Supplemental Table S1**. Reads were aligned to mouse mm9 using (Gaidatzis et al. 2015) and only the uniquely mapped reads were kept for further analysis. Quantification of ChIP-seq signal was made using the csaw package (Lun and Smyth 2016), in particular using the function windowCounts with options dedup=T and minq=28. A window of 10 kb was used for quantification. If more than one replicate were available, all replicates were combined using the geometric mean of the mapped reads. Normalisation over input was performed as in (Perner et al. 2014) using a pseudo-count of 8. Peaks were called with macs2 (Zhang et al. 2008) using default parameters. A peak is assigned to a specific boundary if it belongs to the 40kb window centered on the boundary coordinate.

Transcriptional coregulation

To determine whether a domain is transcriptionally co-regulated during differentiation, a cyclic permutation of gene locations is performed. We defined a domain at any scale in the hierarchy to be co-regulated, if the number of co-regulated genes in the domain is larger than in 95% of the cyclic permutated genomes (empirical p<=0.05). For each insulation value, we calculated the number of domains (N_{obs}) that are up or down-regulated. In order to measure the statistical enrichment of N_{obs}, we calculated a Z-score as follows. We randomly reshuffled gene positions in the genome N=2000 times, and calculated the mean value (N_{exp}) and the standard deviation (σ) of the number of up- or down-domains (defined as described above) in the randomized genomes. The Z-score was defined as:

$$\text{Z-score} = (N_{obs} - N_{exp}) / \sigma$$

Enhancer calling

Enhancer regions were identified taking advantage of H3K27ac, H3K4me1, H3K4me3 and CTCF ChIP-Seq data (**Supplemental Table S1**) as follows. We used H3K27ac peaks (called with macs2 (Zhang et al. 2008) with qvalue <= 10E-8) as landmark regions. We then expanded the peak regions to +/-1kb and evaluated the ratio between H3K4me1 and H3K4me3 signal in these regions. Since the distribution of ratios is bimodal, we could define a list of regions with high H3K4me1 and low H3K4me3 (Heintzman et al. 2007). This allows us to distinguish enhancer regions (characterized by high H3K4me1 and low H3K4me3) from promoter regions (characterized by low H3K4me1 and high H3K4me3). From this list of regions, we finally defined enhancers by discarding those regions that overlap with conserved CTCF peaks conserved across ESCs and NPCs (putative insulators), and those that localize within +/-2.5kb from the gene promoters (putative core promoters and TSS-proximal, *cis*-acting regulatory elements).

Analysis of enhancer-promoter interactions

For each pair of genomic loci used in the analysis, we calculated the ratio between the observed Hi-C counts and the genome-wide average Hi-C count (including zeroes) for loci that are separated by the same genomic distance. The median ratios for interactions occurring within a domain, or across two adjacent domains, were used for plotting the curves in **Figure 4**. Similar results were found using mean values (data not shown). To avoid including under sampled interactions due to limited Hi-C coverage at large genomic distances, we only considered pairs of loci separated by less than 2Mb in ESCs and NPCs, and 1 Mb in CH12 cells. Cutoffs were chosen to exclude genomic distances where average Hi-C counts are dominated by experimental noise (**Supplemental Figure S4d**). Genomic 20-kb (ESCs and

NPCs) and 10-kb (CH12) bins were assigned to 'enhancer', 'promoter' or 'CTCF' categories if they contain at least one of these elements, identified as described before. If a bin shows multiple classifications, we assigned it to all the categories.

Correction to account for the presence of an inactive X chromosome in NPCs

The presence of an inactive X chromosome in the NPC sample we analyzed implies that only one copy of the genes on chromosome X is active (except the set of escape genes identified in the same NPC clone in (Giorgetti et al. 2016)). As a consequence, the expression level of a gene (excluding escapees) that increases by a factor 2 specifically on the active X during differentiation will be detected as unchanged in non-allelic RNA-seq data. To correct for this issue in the definition of down- and up- regulated chrX genes (except escapees), we introduced a modified criterion compared to autosomal genes:

$$FC < -log2(3) - log2(2) \qquad \text{for down} - \text{regulation}$$

$$FC > log2(3) - log2(2) \qquad \text{for up} - \text{regulation}$$

where the factor -log2(2) accounts for the twofold reduction in the detected expression level of genes on the active X in NPCs.

Correlation of histone marks within and across domains

To look at correlation of histone modification we proceeded as in (Rao et al. 2014). To briefly summarize the method, we divided each domain into 10 bins, where the bin size was a tenth of the size of the domain. For each domain and its corresponding adjacent domains we then recorded the mean value of the chromatin mark of interest for each bin.

This procedure 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 (**Supplemental Figure S1k**). This correlation matrix represents how correlated the chromatin marks are at any two loci within and across domains.

Source Code

```
#include <R.h>
#include <Rinternals.h>
#include <Rmath.h>
#define ND          1000
#define MINSIZE 15
#define MAXMOVE          3
#define MINDIST 1

float max(float a, float b){
        if(a>b) return a;
        else return b;
}

float min(float a, float b){
    if(a<b) return a;
    else return b;
}
//calculate total counts
float sum(int i, int j, unsigned short **mat)
{
        float x=0,h;
        int k,l;
```

```
	for (k=i;k<=j;k++)
		for (l=i;l<=j;l++)
			x += mat[k][l];
	return x;
}

float dist(int i1, int j1, int i2, int j2, unsigned short **mat)
{
	float x=0,v=0,di=0,d1=0,d2=0;
	int k,l;
	for(k=i1;k<=j1;k++)
	  for(l=i2;l<=j2;l++) if(k!=l && abs(l-k)>=MINDIST) x+=mat[k][l];
	v=(j1-i1+1)*(j2-i2+1)-1;

	for(k=i1;k<=j1;k++)
		for(l=i1;l<=j1;l++)	if(k!=l && abs(l-k)>=MINDIST) d1+=mat[k][l];
	for(k=i2;k<=j2;k++)
		for(l=i2;l<=j2;l++)	if(k!=l && abs(l-k)>=MINDIST) d2+=mat[k][l];
	di=(x/v)/((d1+d2)/((j1-i1+1)*(j1-i1)+(j2-i2+1)*(j2-i2)));
	return di;
}

SEXP catch(SEXP input)
{
	int i,j,k,id,joined,imin=99999,size=0,tot=0,appo=0;
	//float **insulation;
	float dt,p[ND+1],prevdist=0,newdist=0;
	int nrow,ncol;

	unsigned short **cfrom,**cto,*ncl;
	unsigned short **mat;
	SEXP out,attrib,prof,ncluster;
	FILE *fp;

	nrow = INTEGER(getAttrib(input, R_DimSymbol))[0];
	ncol = INTEGER(getAttrib(input, R_DimSymbol))[1];

      for(i=0;i<nrow;i++)
	for(j=0;j<2;j++) {
			if((j==0 || j==1) && REAL(input)[i+2*nrow]!=-1){
				if(REAL(input)[i+j*nrow]>size)  size=REAL(input)[i+j*nrow];
				if(REAL(input)[i+j*nrow]<imin) imin=REAL(input)[i+j*nrow];
			}
	}

	size++;
	mat = (unsigned short **) calloc(size,sizeof(unsigned short *));
      for (i=0;i<size;i++) mat[i] = (unsigned short *) calloc(size,sizeof(unsigned short));
      for (i=0;i<size;i++)
	  for (j=0;j<size;j++) mat[i][j]=0;
	for(i=0;i<nrow;i++){
		if(REAL(input)[i+2*nrow]!=-1)
	mat[(int)REAL(input)[i+0*nrow]][(int)REAL(input)[i+1*nrow]]=(unsigned short) REAL(input)[i+2*nrow];
	}

	cfrom = (unsigned short **) calloc(ND+1,sizeof(unsigned short *));
	for (i=0;i<ND+1;i++) cfrom[i] = (unsigned short *) calloc(size,sizeof(unsigned short));
	cto = (unsigned short **) calloc(ND+1,sizeof(unsigned short *));
	for (i=0;i<ND+1;i++) cto[i] = (unsigned short *) calloc(size,sizeof(unsigned short));
	ncl = (unsigned short *) calloc(ND+1,sizeof(unsigned short));

	for (i=0;i<ND+1;i++) ncl[i]=0;
	for (i=0;i<(int)(size-imin)/(MINDIST+1);i++)
	{
		cfrom[0][i]=i*(1+MINDIST)+imin;
		cto[0][i]=(i+1)*(1+MINDIST)+imin-1;
```

```
		ncl[0]++;
	}

	Rprintf("Clustering on different thresholds: \n");
	for (id=1;id<=ND;id++)	// increasing threshold
	{
		dt = (float) (ND-id)/ND;
		if(id%100==0)	Rprintf("Relative Insulation: %f\n",1-dt);
		for (i=0;i<ncl[id-1];i++)	// run on clusters
		{
			joined=-1;

			cfrom[id][ncl[id]] = cfrom[id-1][i];
			cto[id][ncl[id]] = cto[id-1][i]
			for (k=i+1;k<ncl[id-1];k++)  // clusters to join previous
			{


				if ( dist(cfrom[id][ncl[id]],cto[id][ncl[id]],cfrom[id-1][k],cto[id-1][k],mat) >= dt )
				{
					cfrom[id][ncl[id]] = cfrom[id-1][i];
					cto[id][ncl[id]] = cto[id-1][k];

					joined = k;
				}
				else break;
			}

			if (joined==-1)
			{
				cfrom[id][ncl[id]] = cfrom[id-1][i];
				cto[id][ncl[id]] = cto[id-1][i];
				ncl[id] ++;
			}
			else
			{
				i=joined+1;

				ncl[id] ++;
				i = joined;
			}
		}

		//movement
	for(i=0;i<ncl[id]-1;i++){
		//except last
			if((cto[id][i]-cfrom[id][i]>(2*MAXMOVE) && cto[id][i+1]-cfrom[id][i+1]>(2*MAXMOVE))
&& (cto[id][i]-cfrom[id][i]>MINSIZE || cto[id][i+1]-cfrom[id][i+1]>MINSIZE)){
				prevdist=dist(cfrom[id][i],cto[id][i],cfrom[id][i+1],cto[id][i+1],mat);
					for(j=1;j<MAXMOVE;j++){
					newdist=dist(cfrom[id][i],cto[id][i]+j,cfrom[id][i+1]+j,cto[id][i+1],mat);
						if(newdist<prevdist){
							prevdist=newdist;
							cto[id][i]=cto[id][i]+j;
							cfrom[id][i+1]=cfrom[id][i+1]+j;
						}
				newdist=dist(cfrom[id][i],cto[id][i]-j,cfrom[id][i+1]-j,cto[id][i+1],mat);

			    if(newdist<prevdist){
				prevdist=newdist;
				cto[id][i]=cto[id][i]-j;
				cfrom[id][i+1]=cfrom[id][i+1]-j;
			    }



					}
```

```
                        }

        }

        }
        Rprintf("\n");

        PROTECT(ncluster=allocMatrix(REALSXP,ND+1,2));

        for (i=0;i<ND+1;i++) {
                REAL(ncluster)[i+0*(ND+1)]=(float)i/ND;
                REAL(ncluster)[i+1*(ND+1)]=ncl[i];
        }
    tot=0;
    for (i=1;i<=ND;i++)
       for (j=0;j<ncl[i];j++)  tot++;
        appo=0;
        PROTECT(out = allocMatrix(REALSXP, tot,3));

    for (i=1;i<=ND;i++)
       for (j=0;j<ncl[i];j++){
              REAL(out)[appo+tot*0] =(float)i/ND;
              REAL(out)[appo+tot*1] =cfrom[i][j];
              REAL(out)[appo+tot*2] =cto[i][j];
              appo++;
       }

        PROTECT(prof=allocVector(VECSXP,2));
        PROTECT(attrib=allocVector(STRSXP,2));
        SET_STRING_ELT(attrib,0,mkChar("clusters"));
        SET_STRING_ELT(attrib,1,mkChar("ncluster"));

        SET_VECTOR_ELT(prof,0,out);
        SET_VECTOR_ELT(prof,1,ncluster);
        setAttrib(prof, R_NamesSymbol,attrib);

        UNPROTECT(4);
        return prof;

}
```

References

Bailey TL, Elkan C. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. pp. 28–36, AAAI Press, Menlo Park, California.

Bailey TL, Gribskov M. 1998. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics* **14**: 48–54.

Gaidatzis D, Lerch A, Hahne F, Stadler MB. 2015. QuasR: quantification and annotation of short reads in R. *Bioinformatics* **31**: 1130–1132.

Gendrel A-V, Attia M, Chen C-J, Diabangouaya P, Servant N, Barillot E, Heard E. 2014. Developmental Dynamics and Disease Potential of Random Monoallelic Gene Expression. *Dev Cell* **28**: 366–380.

Giorgetti L, Lajoie BR, Carter AC, Attia M, Zhan Y, Xu J, Chen CJ, Kaplan N, Chang HY, Heard E, et al. 2016. Structural organization of the inactive X chromosome in the mouse. *Nature* **535**: 575–579.

Heintzman ND, Stuart RK, Hon G, Fu Y, Ching CW, Hawkins RD, Barrera LO, Van Calcar S, Qu C, Ching KA, et al. 2007. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat Genet* **39**: 311–318.

Imakaev M, Fudenberg G, McCord RP, Naumova N, Goloborodko A, Lajoie BR, Dekker J, Mirny LA. 2012. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat Methods* **9**: 999–1003.

Love MI, Huber W, Anders S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* **15**: 550.

Lun ATL, Smyth GK. 2016. csaw: a Bioconductor package for differential binding analysis of ChIP-seq data using sliding windows. *Nucleic Acids Res* **44**: e45–e45.

Mathelier A, Zhao X, Zhang AW, Parcy F, Worsley-Hunt R, Arenillas DJ, Buchman S, Chen C, Chou A, Ienasescu H, et al. 2013. JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids* **42**: D142-7.

Nagano T, Várnai C, Schoenfelder S, Javierre B-M, Wingett SW, Fraser P. 2015. Comparison of Hi-C results using in-solution versus in-nucleus ligation. *Genome Biol* **16**: 175.

Perner J, Lasserre J, Kinkley S, Vingron M, Chung H-R. 2014. Inference of interactions between chromatin modifiers and histone modifications: from ChIP-Seq data to chromatin-signaling. *Nucleic Acids Res* **42**: 13689–13695.

Rao SSP, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, Sanborn AL, Machol I, Omer AD, Lander ES, et al. 2014. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell* **162**: 687–688.

Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, et al. 2008. Model-based Analysis of ChIP-Seq (MACS). *Genome Biol* **9**: R137.