1 METHODS

1.1 HI-C DATA

1.1.1 Mapping

Raw Hi-C reads were downloaded from published datasets (Table 1) through the Gene Expression Omnibus (GEO)[1] or the Short Read Archive (SRA)[2] with identifiers: GSE35156 (H1 hESC), GSE18199 (K562) and SRX030113 (GM12878). These paired reads were mapped independently to a reference genome: hg19/GRCh37 for human data, and mm10/GRCm38 for mouse.

Mapping was performed using the `hiclib` software package[6] and `bowtie2`[7] with the `--very-sensitive` flag. An iterative mapping approach was used to maximise the number of aligning fragments.[6] Each fragment end was aligned first using short terminal sub-sequences. Those unmapped or with ambiguous mapping were then taken forward into the next iteration and extended until the entire fragment end had been aligned. Those remaining pairs with one or more unmapped ends were discarded.

This approach is designed to maximise uniquely-alignable fragment ends, while avoiding mismappings caused by crossing the fragment junction.[8]

1.1.2 Filtering

After mapping, interactions are first aggregated into restriction fragments then by regular binning of various resolutions (particularly 40 kb, 100 kb and 1 Mb). Several filters were applied at this stage, with the following cases removed:[6,8]

- Reads directly adjacent to a restriction enzyme site (within 5 bp)
- Identical read pairs (presumed PCR duplicates)

Table 1: Public Hi-C data used in this work.

Cell line	Total reads	Accession	Citation
Gm12878	31×10^6	SRX030113	3
H1 hESC	331×10^6	GSE35156	4
K562	36×10^6	GSE18199	5
Cortex	373×10^6	GSE35156	4
mESC	476×10^6	GSE35156	4
IMR90	355×10^6	GSE35156	4

- Very large restriction fragments (> 100 kb) which are likely from a repetitive or poorly-assembled region
- Extremely over-represented fragments (top .05%) which may throw-off eigenvector derivation

1.1.3 Correction

Iterative correction and eigenvector expansion (ICE) is an approach to normalisation and processing Hi-C data, implemented as software library written in `python`.[6] The iterative correction algorithm performs matrix balancing with the aim of generating a doubly stochastic matrix from raw interaction counts.[8] That is, such that symmetric matrix **A** has both row and columns of equal sum. In practice, this effectively enforces "equal visibility" of each fragment, correcting for previously-described biases in interaction recovery such as GC-content and fragment length[9] but without explicitly modelling these latent variables. This procedure is thus converting actual interaction counts into normalised interaction frequencies (IF), and to relative rather than absolute quantities. Scaling of IFs permits comparison of Hi-C experiments with very different sequencing depths (as is the case in this work, see Table 1). Despite differences in the levels of sequencing, otherwise the experiment methods underlying the produced Hi-C data were similar: the HindIII restriction enzyme was used in each case and the Hi-C protocol was largely unchanged (that is, we did not consider data from Hi-C variants such as TCC[3] and *in-situ* Hi-C[10]).

1.1.4 Eigenvector calculation

Additional functionality provided by ICE is the eigenvector expansion of normalised contact maps. Eigenvectors from observed/expected matrices were chosen for consistency with Lieberman Aiden *et al.*,[5] as opposed to the related eigenvectors calculated in Imakaev *et al.*[6] from the corrected maps alone. The details of this procedure are described in section 1.5.3. Briefly, observed contacts (O) are divided by an expected matrix (E) which is generated by averaging the super- and sub-diagonals of the O matrix. That is, the E matrix gives the expected value of interactions at a given distance.

Importantly, the first two principle components (PCs) were calculated, and that with the highest absolute Spearman correlation with GC content is taken to reflect A/B compartmentalisation. PC eigenvectors were then orientated to positively correlate with GC, ensuring positive values reflected A compartments and negative values B compartments. Another subtlety is the calculation of eigenvectors per chromosome arm as opposed to per chromosome, this prevents issues with some meta- and submetacentric chromosomes where the first principle component indicated chro-

mosome arms.[5,6] Eigenvector expansion was performed on both 1 Mb and 100 kb matrices, below these resolutions results became less stable, and besides it has been shown that eigenvectors at higher resolution — when they do indeed capture A/B compartmentalisation — add little, if any, additional information.

1.1.5 TAD calling

TADs were called using the software provided in Dixon *et al.*[4] and their recommended parameters. This method is introduced in Section **??** (see also Fig. **??**) but will be described here in greater detail.

The TAD calling algorithm is a multi-stage process. Firstly, a statistic called the "directionality index" (DI) is calculated for each bin.[4] The equation for calculating the DI of a given bin is shown (Eqn. 1), where U represents the sum of reads mapped up to 2 Mb upstream of a given 40 kb bin, and D likewise for downstream contacts. Here E is the expected number of downstream or upstream contacts (equal under the null hypothesis), hence is $E = \frac{U+D}{2}$.

$$\mathrm{DI} = \left(\frac{U-D}{|U-D|}\right)\left(\frac{(D-E)^2}{E} + \frac{(U-E)^2}{E}\right) \tag{1}$$

Equation 1 can be intuitively understood as first determining the direction of the bias (the sign is given by $\frac{U-D}{|U-D|}$) and then calculating the extent of the bias (with $\frac{(D-E)^2}{E} + \frac{(U-E)^2}{E}$ being akin to a χ^2-type statistic).[4]

This DI metric could be used as-is to call domains. Peaks of downstream contacts culminating in a peak of upstream contacts would delineate a self-interacting domains. However, Dixon *et al.*[4] instead use a hidden Markvoc model (HMM) in a manner similar of that we later employed to call compartments (Section 1.5.3).

Here, the DI metric is considered a noisy observation emitted by an unobserved underlying three-state sequence of upstream, downstream or no- directional contact bias.[4] The HMM was fit to each chromosome with between 1 and 20 Gaussian mixtures allowed per state, however in some cases the expectation-maximisation (EM) algorithm used to parameterise these hidden states failed to converge; such cases were ignored. The Akaike information criterion (AIC) was used to selection the optimal number of mixtures (in practice, we found 5–10 were selected).

Finally, given a fully-specified HMM we can calculate the posterior probability of a given state in a specific bin, using the forward-backward algorithm and given its observed data and preceding state sequence. Dixon *et al.*[4] enforce the heuristic that regions are only classified as downstream- or upstream-biased if the state is called for two consecutive bins, or if a single bin has an especially high posterior probability ($\geq .99$). Domains are called from this state sequence and run from an

initial downstream-biased bin through to the last in a run of ≥ 2 of upstream biased states. This procedure was implemented by Dixon *et al.*[4] in `Matlab`.

1.2 ENCODE FEATURES

Genome-wide ChIP-seq datasets for: 22 DNA binding proteins and 10 histone marks were made available by the ENCODE consortium[11,12] along with DNase I hypersensitivity and H2A.z occupancy, for each of the Tier 1 ENCODE cell lines used in this work: H1 hESC, K562 and GM12878. These data were pre-processed using MACSv2[13] to produce signal fold-change relative to input chromatin. In most cases a paired input control was generated per cell type and by the same laboratory as that which performed a set of ChIP-seq experiments.[12] GC content was also calculated and used in the featureset to give 35 total inputs (Table 2).

Table 2: ChIP-seq and other public datasets used in this work.

Histone modifications	DNA binding proteins	Other
H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me2, H3K4me3, H3K79me2, H3K9ac, H3K9me3, H4K20me1	ATF3, CEBPB, CHD1, CHD2, CMYC, CTCF, EGR1, EZH2, GABP, JUND, MAX, MXI1, NRSF, POL2, P300, RAD21, SIX5, SP1, TAF1, TBP, YY1, ZNF143	DNase, GC content, H2A.Z

1.2.1 Clustering input features

To quantify collinearity of input features, correlation matrices built from genome-wide vectors of input feature measures were build and hierarchicaly clustered. The "significance" of observed clustering was assessed using sub- and super-sampled bootstrapping, with stable clusters deemed significant, as implemented in the `pvclust` R package.[14]

1.3 MODELLING COMPARTMENT EIGENVECTORS

1.3.1 Random Forest

Random Forest (RF) regression,[15] was used as implemented in the R package `randomForest`.[16] The RF algorithm (Fig. 1) makes use of a collective of regres-

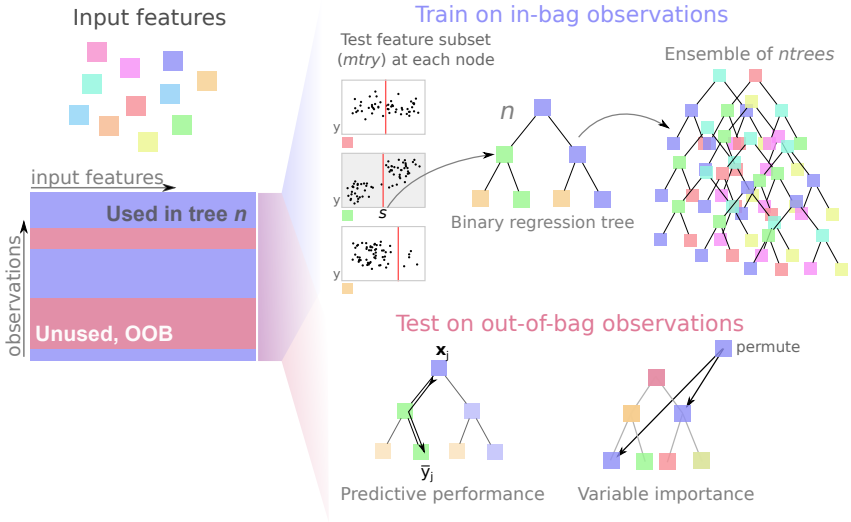

Figure 1: Random Forests overview. Random Forests are an ensemble of bagged, decorrelated classification or regression trees first described by Breiman.[15]

sion trees (size *ntrees*), each built from a bootstrapped sample of the training set. In growing each tree, a small number of variables (*mtry*) is tested at each bifurcation node, and that which minimises the variance in child node subsets is selected at a specific threshold. Having trained a group of trees, these can then be used as predictive tools by inputting a vector of features to each tree and averaging the output leaf node value across the forest. RF regression was used as it is known to be one of the most powerful regression methods developed to date,[17,18] typically providing low bias and low variance predictions without the need for variable selection.[19,20]

Additionally the RF method represents an example of "algorithmic modelling"[21] in that it makes no assumptions about the underlying data model. Parameters of $mtry = \frac{n}{3}$ (where n is the number of input features) and $ntrees = 200$ were assumed as they are known to be largely insensitive;[20,22] this was verified with the dataset used in this work (Fig. 2).

Variable importance within Random Forest regression models was measured using mean decrease in accuracy in the out-of-bag (OOB) sample. This represents the average difference (over the forest) between the accuracy of a tree with permuted and unpermuted versions of a given variable (Fig. 1), in units of mean squared error (MSE).[18,20]

1.3.2 Model performance

The effectiveness of the modelling approach was measured by four different metrics. Prediction accuracy was assessed by the Pearson correlation coefficient between the OOB predictions and observed eigenvectors, and the root mean-squared error

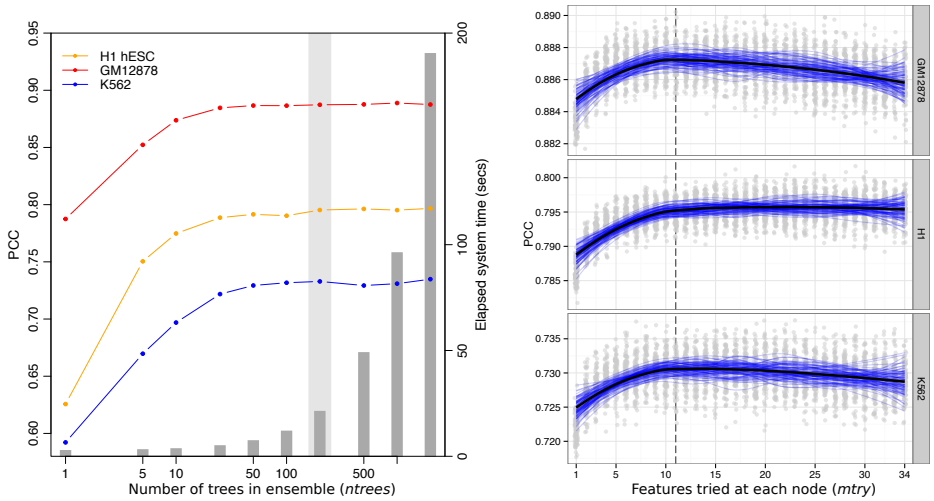

Figure 2: Random Forest parameters are largely insensitive. Two user-facing Random Forest paramters are known to be insensitive over a broad range.[22] Optimisations for *ntrees* (the number of trees in the forests) and *mtry* (the number of features tested at each node) are shown for three different models, with typical values of 200 trees and $\frac{1}{3}$ of input variables highlighted.

(RMSE) of the same data. Classification error, when predictions where thresholded into $A \geq 0; B < 0$, was also calculated using accuracy (% correct classifications or True Positives) and area under the receiver operating characteristic (AUROC) curve. Together these give a comprehensive overview of the model performance, both in terms of regression accuracy of the continuous eigenvector, and in how that same model could be used to label discrete chromatin compartments.

For cross-application of cell type specific models, a single Random Forest regression model was learned from all 1 Mb bins for a given cell type. This was then used to predict all bins from each of the other two cell types.

To test the sensitivity of the models to resolution, we also applied cell-type specific models learnt at 1 Mb resolution to input features binned at 100 kb. This was done by training a Random Forest regression model on all available 1 Mb bins in a given cell type, then applying that model to the prediction of all compartment eigenvectors derived at 100 kb. Model performance was then assessed as above, with the caveat that here the test set represents a higher-resolution window onto the original training set, therefore we might expect this to inflate the measures of generalisation error.

1.3.3 Stepwise regression

Stepwise regression is a form of model selection used with multiple regression. This simple approach starts with a complete model and serially remove and/or add variables, then calculate a metric (here we use the Bayesian information criterion, BIC)

which weighs the the model likelihood against model complexity. This process is iterated until the metric reaches a (local) minimum, thus creating a more parsimonious model which retains predictive accuracy and should be less prone to overfitting. Stepwise regression also aids interpretation by selecting representative features from collinear clusters.[23]

It should be noted that despite its continued widespread usage, several statistical issues have been identified with the stepwise procedure for model selection.[24,25]

1.3.4 LASSO

The least absolute shrinkage and selection operator (LASSO) is a form of ℓ_1 regularisation that penalises the sum of absolute values of standardised regression coefficients. By penalising absolute values and sums, rather than squared values as in ℓ_2 regularisation (Ridge regression, for example), coefficients can be shrunk to 0 thereby removing terms from the model. Thus LASSO combines coefficient shrinkage of techniques like Ridge regression with a type of feature selection by promoting model sparsity.[22,26]

Simply put, the LASSO minimises the sum of squared errors subject to a tuneable constraint on the sum total of absolute model coefficients. In equation form, we are fitting a simple linear model:

$$\begin{aligned} \hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \\ \text{or} \quad \hat{y} = \boldsymbol{X}\boldsymbol{\beta} \end{aligned} \tag{2}$$

We then wish to find that $\boldsymbol{\beta}$ which minimises $\sum_{j=1}^{n}(\hat{y}_j - y_j)^2$ while at the same time satisfying the inequality:

$$\sum_{i=1}^{p} |\beta_i| \leq c \tag{3}$$

Where here c represents a tuneable parameter inversely proportional to the level of regularisation imposed on the model. It can be seen, for example, that of c is set to the sum of the coefficients fit by ordinary least squares, the LASSO solution will be equivalent. Eqn. 3 can be contrasted with Ridge regression, where the same in inequality instead constrains $\sum_{i=1}^{p} \beta_j^2$.

Formally, the LASSO problem has been expressed as:[22]

$$\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=i}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{4}$$

This formulation introduces the λ parameter as used in Section **??**. This parameter translates β coefficients, such that larger values of λ place stronger constraints on the coefficient total, thus encourage greater shrinkage and model sparsity (Eqn. 4).

In this thesis, we used the `glmnet` R package to fit LASSO models.[27,28] In order to select λ, we use a 10-fold cross-validation approach on a separate held-out training set. We chose that λ which produced a mean cross-validated error within 1 standard error of the minimum, thus favouring a sparser model than the global minimima.

1.3.5 Other modelling approaches

Linear regression was used as a baseline for comparison with more complicated approaches such as Random Forest. If the same modelling accuracy could be achieved with simple multiple linear regression, this would be a faster and more interpretable modelling framework.

Partial least squares (PLS) regression was also used to model compartment profiles. PLS regression is well-suited to highly correlated inputs, employing a dimensionality reduction step to help address this redundancy, yet lacks the interpretability of a multiple linear regression. Similar to RF, PLS regression is aimed at building highly-predictive models rather than understanding singular relationships between a predictor and independent variable.[29] The `plsdepot` R implementation of PLS regression was used in this work.

1.4 VARIABLE REGIONS

1.4.1 Stratification by variability

Median absolute deviation (MAD) was chosen as a robust measure of the variability in a given 1 Mb block between the three primary cell types used in this work: H1, K562 and GM12878. Blocks were ranked by this measure and split into thirds that represented "low" variability (the third of blocks with the lowest MAD), "mid" and "high" variability. Each subgroup was then independently modelled using the previously-described Random Forest approach.

"Flipped" regions are those whose compartment state differs in one cell type relative to the other two. For example, if a 1 Mb bin was classified as "open" in H1 hESC and "closed" in both K562 and GM12878, this is said to be a "flipped" compartment (to open).

1.4.2 Chromatin state enrichment

Chromatin state annotations used in this work were retrieved from the ChromHMM[30] and SegWay[31] combined annotations.[32] These represent the consensus from two independent chromatin state prediction algorithms, and ignore regions of apparent disagreement; hence in theory making more robust and conservative predictions than either algorithm independently. Nevertheless, Hoffman *et al.* caution that in areas of disagreement, each algorithm may highlight differing biological phenomena so should also be considered separately.[32]

The set of state predictions from the combined algorithms are:

1. Predicted transcription start sites (TSS)
2. Promoter flanking regions
3. Transcribed reigons
4. Repressed regions
5. Predicted enhancers
6. Predicted weak enhancer or *cis* regulatory element
7. CTCF-enriched elements

Short, discrete state predictions such as enhancers were considered "shared" if there was an overlapping enhancer annotation in either of the two other cell types, and labelled as "tissue-specific" otherwise. This was repeated for each of the called chromatin states.

1.4.3 Gene ontology analysis

Variable regions (section 1.4.1) were tested for functional enrichments using Gene Ontology (GO) annotations.[33] The DAVID tool[34] was used to compare GO terms for genes located in variable compartments with a background set of genes within all annotated compartments.

1.5 BOUNDARIES

1.5.1 TAD boundaries

Having called TADs (Section 1.1.5), we than have a set of boundaries at the start and end of each domain. We generated average boundary enrichment or depletion profiles by averaging input features into 50 kb bins spanning $\pm$450 kb from the boundary bin.

To test for the enrichment or depletion of a chromatin feature over a given boundary, a two tailed Mann-Whitney test was used to compare the boundary bin with the ten outermost bins of the window (5 from either side). The significance level at $\alpha = 0.01$ was then Bonferonni-adjusted for multiple testing correction, and results with p-values exceeding this threshold were deemed significantly enriched or depleted at a given boundary.

To compare boundaries between cells, each TAD boundary called in K562 and GM12878 were compared with those called in H1 hESC. For each boundary, the minimum absolute difference to the nearest matching boundary in H1 hESC was recorded, and this was then compared with a null model of an equal number of boundaries randomly-placed along available bins. A Kolmogorov-Smirnov test was then used to compare the empirical cumulative distributions of these distances.

1.5.2 Predicting TAD boundaries

To predict TAD boundaries we used a classification Random Forest model, built with the `AUC-RF` algorithm.[35]

The input feature set for this model was made up of the same 35 matched features used in models of compartment eigenvector (Section **??**), with the addition of counts of Alu repeat elements (Section **??**). Classifications are those produced by the Dixon *et al.*[4] TAD calling algorithm (Methods 1.1.5), therefore TAD boundaries were resolved to 40 kb bins. We class such bins as boundary true positives (TP), and select matched bins 500 kb upstream as boundary true negatives (TN) for our training set.

To build parsimonious and accurate models (as discussed in Section **??**), we used the This is a form of stepwise model selection which optimises feature subset selection relative to the area under the receiver operating characteristic (AUROC), a metric which captures both the speciality and sensitivity of a classifier. This method was used a training set of 80% of boundaries per cell type, with predictions assessed on out-of-bag (OOB) data as each forest was constructed. Selected models were then applied to the remaining held-out test set of 20% of TAD boundaries, with their matched non-boundary bins (full details are in Methods 1.5.2).

1.5.3 Compartments

Eigenvectors were calculated as described in section 1.1.4. A/B compartmentalisation has previously been called simply from the properly-orientated principle component eigenvector, with positive values representing a bin in an A compartment state, and negative values representing a bin in a B, more repressive state.[5]

Compartment boundaries were called by first training a two-state hidden Markov model (HMM) on the compartment eigenvector and then using the Viterbi algorithm to

predict the most likely state sequence that produced the observed values. Justification for this approach is discussed in Section **??** and we also note the similar use of an HMM in TAD calling (Section 1.1.5).

The point at which transitions occurred between states was taken as a boundary which was then extended ± 1.5 Mb to give a 3 Mb window in which a boundary was though to occur. Boundary enrichments and alignments were tested in the same manner as TADs (Section 1.5.1).

1.6 METATAD ANALYSIS

MetaTADs are a concept discovered by collaborators. Their method for calling such features involve the constrained hierarchical cluttering of those neighbouring TADs with the greatest inter-TAD contacts. This pairing was recursed up to the level of whole chromosomes, thus resulting in a tree of increasing metaTAD aggregation.

1.6.1 Size selection

For boundary analysis of metaTADs, again a similar approach was used to that of TADs (Section 1.5.1) but thresholded to within a given range of sizes. MetaTADs below 10 Mb were excluded, as to have no lower bound results in $\frac{2}{3}$ of all TAD boundaries likewise considered MetaTAD boundaries, reducing the power to analyse any differences. 10 Mb was chosen in an attempt to compromise minimising the overlap between TAD and metaTAD boundaries, while also retaining a large enough sample size. An upper bound of 40 Mb was also chosen, as beyond this threshold inter-TAD contacts were found to be no higher than expected by chance. In practice, the tree-like structure means any upper-bound has little impact as a filter: in almost all cases, any boundary in a metaTAD of size > 40 Mb will also fortpalam metaTADs below this value. Additionally, the hierarchical nature of metaTADs means that some boundaries are present at multiple levels of the tree. Only one case of each boundary position was tested for feature enrichments, and this was performed as with TAD boundaries (Section 1.5.1).

1.6.2 Collaborator datasets

Our collaborators in the metaTAD project performed ChIP-seq experiments for PolIII (three variants), H3K27me3, CTCF and DNase-I hypersensitivity. Mapped reads from these experiments were processed using MACSv2[13] to give relative signal over background (from an estimated local model), which was then averaged over all boundaries genome wide.

Cap analysis of gene expression (CAGE) data was produced by the FANTOM consortium.[36,37] This method produces sequencing data from the 5' end of cDNAs, and can be used to quantify expression activity at precise promoter locations.[38] Here, CAGE was performed at multiple points along a neural-differentiation timecourse and tags were clustered to form CAGE TSS (CTSS) in a manner developed for the FANTOM5 project.[36] To count these CTSS over boundary bins, we simply intersect the annotations and count CTSS per bin using `bedtools`.[39]

Gene density over metaTAD boundaries was calculated using UCSC `mm9` gene models.[40] Again simple intersections were taken to count genes over boundaries using `bedtools`[39] and requiring a minimal overlap fraction of 0.5% of a bin (250 bp).

1.6.3 LAD coincidence

To compare metaTAD boundaries with those of LADs, we made use of previously-published Lamin-B1 DamID microarray probe intensities.Peric-Hupkes *et al.*[41] For analysis over boundaries, these values were averaged into the same boundary windows as used previously (50 kb bins ±450 kb around boundary, as in Section 1.5.1).

Transitions between high and low lamina association were detected by fitting a linear regression model across each series of consecutive boundary bins (i.e. Lamina association $= \beta \cdot \text{bin} + c$). Linear models which had an absolute coefficient $|\beta| > .05$ were taken as crossing a LAD transition. This threshold is a heuristic which appears to perform well at conservatively selecting clear transitions. As a method of seriation for the y-axis of heatmap figures (e.g. Fig. **??**), boundaries were divided into those that coincided with a lamin transition and those that did not, and members within each group were then sorted by average intensity.

To test the significance of the association between boundaries and lamin transitions, we circularly permuted both TAD and metaTAD boundaries on each chromosome 1000 times, and calculated the proportion of boundaries that crossed LAD boundaries using the same linear regression procedure described above. Empirical p-values were then calculated as the number of permuted results greater than or equal to the observed value.

1.7 GIEMSA BAND COMPARISON

Cytogenic band data and Giemsa stain results were downloaded from the UCSC genome browser (table `cytoBandIdeo`). The genomic co-ordinates are an approximation of cytogenic band data inferred from a large number of FISH experiments.[42]

To compare G-band boundaries with our compartment data, we allowed for a ±500 kb inaccuracy in G-band boundary. For each G-band boundary, the minimum

absolute distance to any compartment or TAD boundary was calculated for each cell type. To generate a null model, ...

1.8 NUCLEAR POSITIONING

Previously published data on chromosome positioning preference within the nucleus was used to label each chromosome as "inner", "middle" or "outer".[43] Chromosomes whose DAPI hybridisation signals were significantly enriched ($p \leq 2 \times 10^{-2}$) in the inner nuclear shell, as defined by Boyle *et al.*[43], made up the "inner" group and included chromosomes 1 and 16. Similarly the "outer" group had enriched signals ($p \leq 5 \times 10^{-3}$) in the outer shell relative to the inner nuclear shell and included chromosomes 2, 3, 11-13 and 18. The remaining chromosomes in our filtered dataset, 6, 14 and 15, were assigned to the "middle" group and showed no significant to either inner or outer nuclear shells ($p \geq 0.1$).[43] The significance of the difference in distribution of eigenvectors in the inner versus outer shell was determined by a two-sided Kolmogorov-Smirnov (K-S) test, with the alternative hypothesis that the empirical cumulative density function of the inner chromosome eigenvectors F_{inner} was not equal to F_{outer}. This chromosomal positioning data was measured in lymphoblastoid cells though nuclear architecture is though to be largely conserved between cell types[44,45] and even higher primates,[46] so should be comparable in this instance.

1.9 MODELLING TRANSCRIPTIONAL OUTPUT

1.9.1 Reproducing a published study

In Section **??** we reproduce and extend a previously published study by Dong *et al.*[47]. In doing so, we reuse much of the code and materials made available by the authors and more widely the ENCODE consortium[11], of which this paper was a part. Some scripts were extracted from the ENCODE virtual machine, designed to provide an environment in which to reproduce their main findings.[48]

Input features for models of transcription were derived from the January 2011 ENCODE data freeze.[11] Normalised ChIP-seq signals were generated by ENCODE using `wiggler` and retrieved for this study as `bigWig` files. These were averaged into 40×100 bp bins across each GENCODE v7 TSS, to give ± 2 kb windows around each start site. These bins were then used to find the 'bestbin', that which correlates best with transcriptional output on a training subset of TSS. A bin representing the average intensity over the whole gene (TSS to TES) was also considered. That which best correlated on a training set was then used as the representative region for that feature

in subsequent modelling steps.[47] The justification for this approach is discussed in Section **??**.

1.9.2 Predicting FANTOM5 data

We transferred this transcriptional modelling approach to what was at the time novel, unpublished CAGE data produced by the FANTOM consortium. This data has since been released in the FANTOM5 series of publications.[36]

We used H1 hESC t_0 CAGE data from a differentiation timecourse study, as generated by FANTOM5 members. The consortium pre-processed raw CAGE tags into clusters using decomposition-based peak identification.[36] To filter for gene-associated CAGE clusters, we discarded those tag clusters centered on a point > 50 bp from an Ensembl (v69) annotated TSS, thereby removing enhancers and other non-genic transcribed regions. When multiple clusters were linked to the same TSS, that with the highest peak was kept. Expression was matched with ENCODE ChIP-seq data for the H1 hESC cell type (processed as described in Section 1.9.1) and an additional measure of replication timing.

Input data for models of FANTOM5 CAGE are shown in Table 3.

Table 3: ENCODE datasets generated in the H1 hESC cell line and used in models of transcriptional output.

Histone modifications	Other
H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me2, H3K4me3, H3K79me2, H3K9ac, H3K9me3, H4K20me1	HDAC6, DNase I, H2A.Z, Input

1.10 4C ANALYSIS

The experimental protocol used by our collaborators to generate 3C-seq data (also known as 4C) recommends the `r3Cseq` R package,[49,50] part of the bioconductor repository[51,52] for the R programming environment.[53]

This package functions both to produces normalised interaction frequencies which are comparable between experiments, and to assign statistical significance to any identified contacts, thereby reporting regions that co-localise to a greater degree than expected by their genomic proximity alone.

1.10.1 Normalisation

The normalisation procedure is adapted from a previous method for normalising deepCAGE data between samples.[54] In short, the reverse-cumulative distribution of read counts per restriction fragment is fit to a power-law model; this effectively encodes the *a priori* expectation of exponential decay of the number of contacts as distance increases from the viewpoint. Transformed read counts per million (RPM) can then be retrieved from a standardised reverse cumulative distribution, parametrised with the empirical coefficient, $\alpha = -1.35$.[50]

This normalisation procedure has the effect of making the output RPM value independent of the original experiment's sequencing depth and, more importantly, acts to reduces the impact of artefacts and errors by enforcing the expected power-law relationship of restriction fragment read counts.

1.10.2 Significance estimation

The `r3Cseq` package[50] also attempts to assign a measure of significance to observed contact frequencies. This is done through a simple method of background estimation based on observed values. The justification for this non-independent estimate of background signal is that a relatively small proportion of observed contacts are expected to be significantly enriched, thus won't unduly perturb an average signal.[50] An improved method that avoids this assumption has since been developed where a background model was iteratively fitted, with outlier removal at each revision.[55]

Here a non-parametric cubic smooth spline is fit to normalised read count data using a heuristic smoothing parameter. This model then provides an expected level of interaction at a given distance from the viewpoint in *cis*. From this, it is simple to calculate a Z–score as:

$$Z = \frac{(O - E)}{\sigma} \tag{5}$$

Where σ is the standard deviation of residuals from the observed (O), expected (E) difference. This Z-score can then be converted to a p-value which in turn is corrected for multiple testing using bootstrapped estimates of false-discovery rate (FDR) q-values[56] (as implemented in the `qvalue` R package[57]). This Z–test approach assumes a normally-distributed test statistic, an assumption that typically does not hold on 4C data where interactions distal to the viewpoint are increasingly sparse, however this approach and variants thereof have been applied in a variety 4C and 5C analyses (e.g. 58–63). Some publications (e.g. 64) use a more appropriate distribution to assign p-values to the Z statistic, such as the Weibull (extreme value) distribution.

While we are mostly concerned with these *cis* interactions, `r3Cseq` also offers significance testing for *trans* interactions between the viewpoint and restriction frag-

ments on different chromosomes. Here instead of distance scaling, the expected (E) terms in eqn. 5 are genome-wide averages excluding regions $\pm$ 100 kb around the viewpoint.[50] This means the absolute values of normalised RPMs reported for *trans* interactions are in practice upscaled, being equivalent to experimental RPMs less the most deeply-sequenced regions, i.e. the viewpoint and immediately adjacent regions.

1.10.3 3–D modelling

1.11 5C ANALYSIS

1.12 SCRIPTS AND OTHER ANALYSES

Much of this work has been performed by writing custom scripts in the R programming language.[53] Code for the majority of analyses described in this thesis are available through a public git repository hosted on `github` at github.com/blmoore/3dgenome (instructions on how to reproduce analyses and figures are included therein). A special mention goes to the packages of Hadley Wickham which are used throughout, especially `ggplot2`[65] and `dplyr`[66].

The programming language python[67] was also employed to a lesser-extent, as were command–line tools such as `bedtools`[39] and `SAMtools`[68]. Additionally command-line `BigWig*` tools[69] were used, as well as the UCSC genome browser associated data tracks.[70–72].

REFERENCES

[1] Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall Ka, Phillippy KH, Sherman PM, *et al.* (2013) NCBI GEO: archive for functional genomics data sets–update. *Nucleic acids research*, **41**(Database issue): D991–5.

[2] Leinonen R, Sugawara H, Shumway M (2011) The sequence read archive. *Nucleic acids research*, **39**(Database issue): D19–21.

[3] Kalhor R, Tjong H, Jayathilaka N, Alber F, Chen L (2012) Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. *Nature biotechnology*, **30**(1): 90–8.

[4] Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, Hu M, Liu JS, Ren B (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, **485**(7398): 376–80.

[5] Lieberman-Aiden E, van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, *et al.* (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, **326**(5950): 289–93.

[6] 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. *Nature methods*, **9**(10): 999–1003.

[7] Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. *Nature methods*, **9**(4): 357–9.

[8] Lajoie BR, Dekker J, Kaplan N (2014) The Hitchhikers Guide to Hi-C Analysis: Practical guidelines. *Methods*, (November).

[9] Yaffe E, Tanay A (2011) Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nature genetics*, **43**(11): 1059–65.

[10] Rao S, Huntley M, Durand N, Stamenova E, Bochkov I, Robinson J, Sanborn A, Machol I, Omer A, *et al.* (2014) A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell*, **159**(7): 1665–1680.

[11] ENCODE (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**(7414): 57–74.

[12] Boyle AP, Araya CL, Brdlik C, Cayting P, Cheng C, Cheng Y, Gardner K, Hillier LW, Janette J, *et al.* (2014) Comparative analysis of regulatory information and circuits across distant species. *Nature*, **512**(7515): 453–456.

[13] Zhang Y, Liu T, Meyer Ca, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, *et al.* (2008) Model-based analysis of ChIP-Seq (MACS). *Genome biology*, **9**(9): R137.

[14] Suzuki R, Shimodaira H (2006) Pvclust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics (Oxford, England)*, **22**(12): 1540–2.

[15] Breiman L (2001) Random Forests. *Machine learning*, **45**(1): 5–32.

[16] Liaw A, Wiener M (2002) Classification and Regression by randomForest. *R News*, **2**(December): 18–22.

[17] Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP (2003) Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of chemical information and computer sciences*, **43**(6): 1947–58.

[18] Cutler DR, Edwards TC, Beard KH, Cutler A, Hess KT, Gibson J, Lawler JJ (2007) Random forests for classification in ecology. *Ecology*, **88**(11): 2783–92.

[19] Díaz-Uriarte R, Alvarez de Andrés S (2006) Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, **7**: 3.

[20] Dasgupta A, Sun YV, König IR, Bailey-Wilson JE, Malley JD (2011) Brief review of regression-based and machine learning methods in genetic epidemiology: the Genetic Analysis Workshop 17 experience. *Genetic epidemiology*, **35 Suppl 1**(Suppl 1): S5–11.

[21] Breiman L (2001) Statistical Modeling: The Two Cultures. *Statistical Science*, **16**(3): 199–231.

[22] Hastie T, Tibshirani R, Friedman J (2009) *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition. ISBN 978-0-387-84858-7.

[23] Mantel N (1970) Why Stepdown Procedures in Variable Selection. *Technometrics*, **12**(3): 621–625.

[24] Hurvich CM, Tsai Cl (1990) The Impact of Model Selection on Inference in Linear Regression. *The American Statistician*, **44**(3): 214.

[25] Whittingham MJ, Stephens Pa, Bradbury RB, Freckleton RP (2006) Why do we still use stepwise modelling in ecology and behaviour? *Journal of Animal Ecology*, **75**(5): 1182–1189.

[26] Tibshirani R (1994) Regression Selection and Shrinkage via the Lasso.

[27] Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**(1): 1–22.

[28] Simon N, Friedman J, Hastie T, Tibshirani R (2011) Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, **39**(5): 1–13.

[29] Tobias RD (1995) An Introduction to Partial Least Squares Regression. *Proc. Ann. SAS Users Group Int. Conf. 20th*, pp. 1250–1257.

[30] Ernst J, Kheradpour P, Mikkelsen TS, Shoresh N, Ward LD, Epstein CB, Zhang X, Wang L, Issner R, *et al.* (2011) Mapping and analysis of chromatin state dynamics in nine human cell types. *Nature*, **473**(7345): 43–9.

[31] Hoffman MM, Buske OJ, Wang J, Weng Z, Bilmes Ja, Noble WS (2012) Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nature Methods*, **9**(5): 473–476.

[32] Hoffman MM, Ernst J, Wilder SP, Kundaje A, Harris RS, Libbrecht M, Giardine B, Ellenbogen PM, Bilmes Ja, *et al.* (2013) Integrative annotation of chromatin elements from ENCODE data. *Nucleic acids research*, **41**(2): 827–41.

[33] Ashburner M, Ball Ca, Blake Ja, Botstein D, Butler H, Cherry JM, Davis aP, Dolinski K, Dwight SS, *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, **25**(1): 25–29.

[34] Huang BDW, Lempicki R (2008) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols*, (301): 1–43.

[35] Calle ML, Urrea V, Boulesteix AL, Malats N (2011) AUC-RF: A new strategy for genomic profiling with random forest. *Human Heredity*, **72**(2): 121–132.

[36] Consortium TF, Pmi R, Dgt C (2014) A promoter-level mammalian expression atlas. *Nature*, **507**(7493): 462–70.

[37] Itoh M, Kojima M, Nagao-Sato S, Saijo E, Lassmann T, Kanamori-Katayama M, Kaiho A, Lizio M, Kawaji H, *et al.* (2012) Automated workflow for preparation of cDNA for cap analysis of gene expression on a single molecule sequencer. *PLoS ONE*, **7**(1).

[38] Kodzius R, Kojima M, Nishiyori H, Nakamura M, Fukuda S, Tagami M, Sasaki D, Imamura K, Kai C, *et al.* (2006) CAGE: cap analysis of gene expression. *Nature methods*, **3**(3): 211–22.

[39] Quinlan AR, Hall IM (2010) BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**(6): 841–842.

[40] Karolchik D, Barber GP, Casper J, Clawson H, Cline MS, Diekhans M, Dreszer TR, Fujita Pa, Guruvadoo L, *et al.* (2014) The UCSC Genome Browser database: 2014 update. *Nucleic Acids Research*, **42**(D1): 764–770.

[41] Peric-Hupkes D, Meuleman W, Pagie L, Bruggeman SWM, Solovei I, Brugman W, Gräf S, Flicek P, Kerkhoven RM, *et al.* (2010) Molecular Maps of the Reorganization of Genome-Nuclear Lamina Interactions during Differentiation. *Molecular Cell*, **38**(4): 603–613.

[42] Furey TS (2003) Integration of the cytogenetic map with the draft human genome sequence. *Human Molecular Genetics*, **12**(9): 1037–1044.

[43] Boyle S, Gilchrist S, Bridger JM, Mahy NL, Ellis Ja, Bickmore Wa (2001) The spatial organization of human chromosomes within the nuclei of normal and emerin-mutant cells. *Human molecular genetics*, **10**(3): 211–9.

[44] Chambers EV, Bickmore Wa, Semple CA (2013) Divergence of mammalian higher order chromatin structure is associated with developmental loci. *PLoS computational biology*, **9**(4): e1003017.

[45] de Wit E, Bouwman BaM, Zhu Y, Klous P, Splinter E, Verstegen MJaM, Krijger PHL, Festuccia N, Nora EP, *et al.* (2013) The pluripotent genome in three dimensions is shaped around pluripotency factors. *Nature*, pp. 1–7.

[46] Tanabe H, Müller S, Neusser M, von Hase J, Calcagno E, Cremer M, Solovei I, Cremer C, Cremer T (2002) Evolutionary conservation of chromosome territory arrangements in cell nuclei from higher primates. *Proceedings of the National Academy of Sciences of the United States of America*, **99**(7): 4424–9.

[47] Dong X, Greven MC, Kundaje A, Djebali S, Brown JB, Cheng C, Gingeras TR, Gerstein M, Guigó R, *et al.* (2012) Modeling gene expression using chromatin features in various cellular contexts. *Genome biology*, **13**(9): R53.

[48] Myers RM, Stamatoyannopoulos J, Snyder M, Dunham I, Hardison RC, Bernstein BE, Gingeras TR, Kent WJ, Birney E, *et al.* (2011) A user's guide to the Encyclopedia of DNA elements (ENCODE). *PLoS Biology*, **9**(4).

[49] Stadhouders R, Kolovos P, Brouwer R, Zuin J, van den Heuvel A, Kockx C, Palstra RJ, Wendt KS, Grosveld F, *et al.* (2013) Multiplexed chromosome conformation capture sequencing for rapid genome-scale high-resolution detection of long-range chromatin interactions. *Nature protocols*, **8**(3): 509–24.

[50] Thongjuea S, Stadhouders R, Grosveld FG, Soler E, Lenhard B (2013) R3Cseq: An R/Bioconductor package for the discovery of long-range genomic interactions from chromosome conformation capture and next-generation sequencing data. *Nucleic Acids Research*, **41**(13): 1–12.

[51] Gentleman RC, Gentleman RC, Carey VJ, Carey VJ, Bates DM, Bates DM, Bolstad B, Bolstad B, Dettling M, *et al.* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, **5**(10): R80.

[52] Huber W, Carey VJ, Gentleman R, Anders S, Carlson M, Carvalho BS, Bravo HC, Davis S, Gatto L, *et al.* (2015) Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Publishing Group*, **12**(2): 115–121.

[53] Ihaka R, Gentleman R (1996) R: A Language for Data Analysis and Graphics.

[54] Balwierz PJ, Carninci P, Daub CO, Kawai J, Hayashizaki Y, Van Belle W, Beisel C, van Nimwegen E (2009) Methods for analyzing deep sequencing expression data: constructing the human and mouse promoterome with deepCAGE data. *Genome biology*, **10**(7): R79.

[55] Ay F, Bailey TL, Noble WS (2014) Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. *Genome Research*.

[56] Storey JD, Taylor JE, Siegmund D (2004) Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, **66**(1): 187–205.

[57] Storey J (2015) *qvalue: Q-value estimation for false discovery rate control*. R package version 2.0.0.

[58] Simonis M, Klous P, Splinter E, Moshkin Y, Willemsen R, de Wit E, van Steensel B, de Laat W (2006) Nuclear organization of active and inactive chromatin domains uncovered by chromosome conformation capture-on-chip (4C). *Nature genetics*, **38**(11): 1348–1354.

[59] Sanyal A, Lajoie BR, Jain G, Dekker J (2012) The long-range interaction landscape of gene promoters. *Nature*, **489**(7414): 109–13.

[60] Splinter E, de Wit E, van de Werken HJG, Klous P, de Laat W (2012) Determining long-range chromatin interactions for selected genomic sites using 4C-seq technology: From fixation to computation. *Methods*, **58**(3): 221–230.

[61] Gao F, Wei Z, Lu W, Wang K (2013) Comparative analysis of 4C-Seq data generated from enzyme-based and sonication-based methods. *BMC genomics*, **14**(1): 345.

[62] Dixon JR, Jung I, Selvaraj S, Shen Y, Antosiewicz-Bourget JE, Lee AY, Ye Z, Kim A, Rajagopal N, *et al.* (2015) Chromatin architecture reorganization during stem cell differentiation. *Nature*, **518**(7539): 331–336.

[63] Crane E, Bian Q, McCord RP, Lajoie BR, Wheeler BS, Ralston EJ, Uzawa S, Dekker J, Meyer BJ (2015) Condensin-driven remodelling of X chromosome topology during dosage compensation. *Nature*.

[64] Nora EP, Lajoie BR, Schulz EG, Giorgetti L, Okamoto I, Servant N, Piolot T, van Berkum NL, Meisig J, *et al.* (2012) Spatial partitioning of the regulatory landscape of the X-inactivation centre. *Nature*, **485**(7398): 381–5.

[65] Wickham H (2009) *ggplot2: elegant graphics for data analysis*. Springer New York. ISBN 978-0-387-98140-6.

[66] Wickham H, Francois R (2015) *dplyr: A Grammar of Data Manipulation*. R package version 0.4.2.

[67] Van Rossum G (1995) *Python reference manual*.

[68] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**(16): 2078–2079.

[69] Kent WJ, Zweig aS, Barber G, Hinrichs aS, Karolchik D (2010) BigWig and BigBed: Enabling browsing of large distributed datasets. *Bioinformatics*, **26**(17): 2204–2207.

[70] Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D (2002) The Human Genome Browser at UCSC The Human Genome Browser at UCSC. *Genome Research*, pp. 996–1006.

[71] Raney BJ, Dreszer TR, Barber GP, Clawson H, Fujita Pa, Wang T, Nguyen N, Paten B, Zweig AS, *et al.* (2014) Track data hubs enable visualization of user-defined genome-wide annotations on the UCSC Genome Browser. *Bioinformatics*, **30**(7): 1003–1005.

[72] Kuhn RM, Haussler D, James Kent W (2013) The UCSC genome browser and associated tools. *Briefings in Bioinformatics*, **14**(2): 144–161.