

# Unravelling higher order genome organisation [working title]

## Methods section

Benjamin L. Moore

June 25, 2015

# 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)<sup>[2]</sup> or the Short Read Archive (SRA)<sup>[2]</sup> 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<sup>[2]</sup> and bowtie2<sup>[2]</sup> with the `--very-sensitive` flag. An iterative mapping approach was used to maximise the number of aligning fragments.<sup>[2]</sup> 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.

### 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:<sup>[2]</sup>

- Reads directly adjacent to a restriction enzyme site (within 5 bp)
- Identical read pairs (presumed PCR duplicates)
- 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.<sup>[2]</sup> The iterative correction algorithm performs matrix balancing with the aim of generating a doubly stochastic matrix from raw interaction counts. 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<sup>[2]</sup> but without explicitly modelling

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

Cell line	Total reads	Accession	Citation
Gm12878	$31 \times 10^6$	SRX030113	?
H1 hESC	$331 \times 10^6$	GSE35156	?
K562	$36 \times 10^6$	GSE18199	?
Cortex	$373 \times 10^6$	GSE35156	?
mESC	$476 \times 10^6$	GSE35156	?
IMR90	$355 \times 10^6$	GSE35156	?

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).

#### 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.*,<sup>[2]</sup> as opposed to the related eigenvectors calculated in Imakaev *et al.*<sup>[2]</sup> from the corrected maps alone. The details of this procedure are described in section 1.5.2. 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 chromosome arms.<sup>[2]</sup> Eigenvector expansion was performed on both 1 Mb and 100 kb matrices, below these resolutions results became less stable, and it has been shown that eigenvectors at

## 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<sup>[2]</sup> 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<sup>[2]</sup> to produce fold-change relative to input chromatin. GC content was also calculated and used in the featureset.

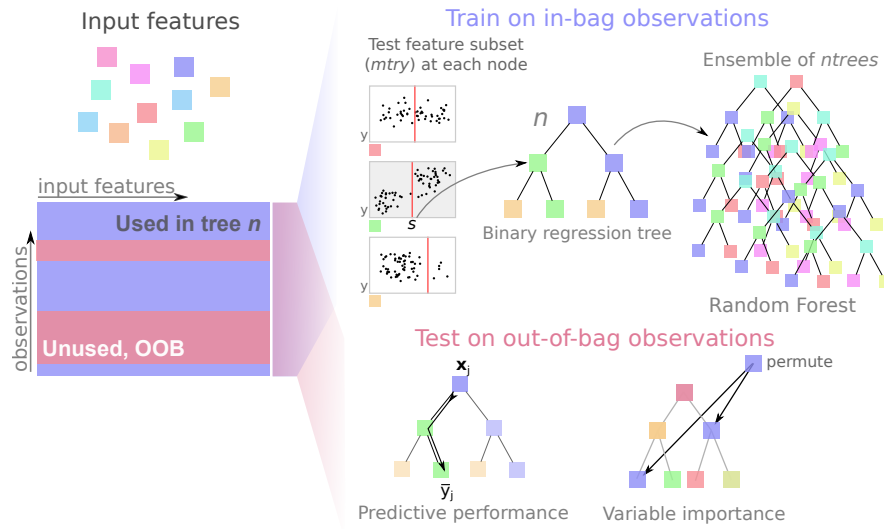
#### 1.2.1 Clustering input features

To quantify collinearity of input features, correlation matrices built from genome-wide vectors of input feature measures were built and hierarchically clustered. The "significance" of observed clustering was assessed using sub- and super-sampled bootstrapping, with stable clusters deemed significant. The pvc1ust R package

## 1.3 MODELLING

#### 1.3.1 Random Forest

Random Forest (RF) regression,<sup>[2]</sup> was used as implemented in the R package `randomForest`.<sup>[2]</sup> The RF algorithm makes use of a collective of regression 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



**Figure 1: Steps in the Hi-C assay.** Schematic of the Hi-C experimental procedure as described in Lieberman Aiden *et al.* [?]

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, [?] typically providing low bias and low variance predictions without the need for variable selection. [?] Additionally the RF method represents an example of “algorithmic modelling” [?] 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; [?] this was verified with the dataset used in this work (Fig. ??).

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, in units of mean squared error (MSE). [?]

### 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 predicted and observed eigenvectors (determined by 10-fold cross-validation), and the root mean-squared error (RMSE) of the same data. Classification error, when predictions were thresholded into  $A \geq 0; B < 0$ , was also calculated using accuracy (% correct classifications or True Positives) and area under the receiver operating characteristic (AU-ROC) 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.

### 1.3.3 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 regression was also used to model compartment profiles. This method is well-suited to highly correlated inputs.

#### 1.3.4 Graphical lasso

Regularised models made use of the Graphical LASSO<sup>[2]</sup> (least absolute shrinkage and selection operator) as a method of  $L_1$ -norm based regularisation, implemented via the `glasso` R package. The graphical lasso provides tuneable regularisation which is capable of feature selection via minimising regression parameters to 0. It was chosen in this case due to the multicollinearity of the featureset, the algorithm's fast speed of execution and the intuitiveness a graphical model presents.<sup>[2]</sup>

More specifically, the graphical lasso regulates the number of 0s in the inverse covariance matrix,  $\Theta = \Sigma^{-1}$ , also known as the precision matrix. Then if element  $\theta_{ij} = 0$ , the variables  $X_i$  and  $X_j$  can be said to be conditionally independent, given the remaining variables.<sup>[2]</sup> The algorithm minimises a negative log-likelihood (Eqn. 1<sup>[2]</sup>) given the tuning parameter  $\lambda$ , which was tuned in this case to leave a small number of variables ( $< 10$ ) directly dependent on the eigenvector data.

$$\underset{\Theta \succ 0}{\text{minimise}} f(\Theta) := -\log \det(\Theta) + \text{tr}(\mathbf{S}\Theta) + \lambda \|\Theta\|_1 \quad (1)$$

## 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 Enhancer enrichment

Enhancer annotations were collected from the ChromHMM / SegWay combined annotations in each cell type.<sup>[2]</sup> 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 other chromatin states.

## 1.5 BOUNDARIES

### 1.5.1 TADs

TAD boundaries were called using the software provided in <sup>[2]</sup> using their recommended parameters. For the generation of boundary profiles, the same

parameters were used: input features were averaged into 40 kb bins spanning  $\pm 500$  kb from the boundary centre.

To align boundaries between cells ...

### 1.5.2 Compartments

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. The point at which transitions occurred between states was taken as a boundary which was then extended  $\pm 1.5$  Mb to give a 3 Mb window in which a boundary was thought to occur.

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.

### 1.5.3 MetaTADs

## 1.6 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.<sup>[2]</sup>

To compare G-band boundaries with our compartment data, we allowed for a  $\pm 500$  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, ...

## 1.7 NUCLEAR POSITIONING

Previously published data on chromosome positioning preference within the nucleus was used to label each chromosome as “inner”, “middle” or “outer”.<sup>[2]</sup> 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.*<sup>[2]</sup>, 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 ( $p \geq 0.1$ ).<sup>[2]</sup> The significance of the difference in distribution of eigenvectors in the inner versus outer shell was determined by a one-sided Kolmogorov-Smirnov (K-S) test, with the alternative hypothesis that the empirical cumulative density function of the inner chromosome eigenvectors  $F_{inner}$  is greater-than or equal-to  $F_{outer}$ . This chromosomal positioning data was measured in lymphoblastoid cells though nuclear architecture is thought to be largely conserved between cell types<sup>[2 ?]</sup> and even higher primates.<sup>[2]</sup>