Supplemental Methods

Contents

1 Tree-Guided Integrated Factorization (TGIF)

TGIF is based on multi-task non-negative matrix factorization (NMF) that can be used to identify low dimensional structure across multiple Hi-C datasets. Below we describe the TGIF framework in detail, which consists of NMF, hierarchical multi-task learning with tree-based regularization, and optimization with block coordinate descent.

NMF is a powerful dimensionality reduction method that can recover the underlying low-dimensional structure from high-dimensional data (Lee and Seung, 2000). It aims to decompose a non-negative matrix, $\mathrm{X} \in \mathbb{R}_{\geq 0}^{(n \times m)}$, into two lower dimensional non-negative matrices, $\mathrm{U} \in \mathbb{R}_{\geq 0}^{(n \times k)}$ and $\mathrm{V} \in \mathbb{R}_{\geq 0}^{(k \times m)}$, to minimize the following objective: $\|\mathrm{X} - \mathrm{UV}\|_{\mathrm{F}}^2$, s.t. $\mathrm{U} \geq 0, \mathrm{V} \geq 0$, where $||\cdot||_{\mathrm{F}}$ indicates the Frobenius norm. We refer to the U and V matrices as factors. Here $k \ll n, m$ is the rank of the factors and is an input parameter. As described previously (Lee and Roy, 2021), to apply NMF to Hi-C data, we represent the Hi-C data for each chromosome as a symmetric matrix $\mathrm{X} = [x_{ij}] \in \mathbb{R}^{(n \times n)}$ where x_{ij} represents the contact count between region i and region j.

TGIF implements multi-task NMF, where tasks correspond to Hi-C datasets that in turn are from hierarchically related contexts, such as cellular stages, species, timepoints. We note that a hierarchy is a general form for capturing relationships among a set of conditions and can capture both branching and linear relationships. Multi-task NMF has been previously implemented in the multi-view NMF approach (Liu et al., 2013; Baur et al., 2022), where a view and task can be used interchangeably. However, this existing framework assumes that all the tasks are equally related. Formally, in multi-view NMF, given T different datasets $\{\mathrm{X}^{(1)} \cdots, \mathrm{X}^{(T)}\}$ where each dataset $\mathrm{X}^{(t)} \in \mathbb{R}_{\geq 0}^{(n_t \times m)}$, the goal is to find view-specific factors $\{\mathrm{U}^{(1)}, \cdots, \mathrm{U}^{(T)}\}$ and $\{\mathrm{V}^{(1)} \cdots, \mathrm{V}^{(T)}\}$, and a consensus factor V^* that minimize the following objective:

$$\sum_{t=1}^{T} \left\| \mathrm{X}^{(t)} - \mathrm{U}^{(t)} \mathrm{V}^{(t)} \right\|_{\mathrm{F}}^2 + \lambda \left\| \mathrm{V}^{(t)} - \mathrm{V}^* \right\|_{\mathrm{F}}^2 \tag{1}$$

where $\mathrm{U}^{(t)} \in \mathbb{R}_{\geq 0}^{(n_t \times k)}$ and $\mathrm{V}^{(t)}, \mathrm{V}^* \in \mathbb{R}_{\geq 0}^{(k \times m)}$. This constrains each of the task-specific factors $\mathrm{V}^{(t)}$ to be similar to the consensus factor V^*. The hyper-parameter λ controls the strength of this constraint. The key benefit of such a framework is that the latent representation or structure within each task can borrow from other complementary data, as guided by the consensus factor V^*.

TGIF generalizes multi-view NMF to allow for integration of datasets that can come from different biological contexts such as time or developmental stage, and therefore may not all be equally related to each other. Accordingly, instead of requiring all the $\mathrm{V}^{(t)}$ to be similar to a single V^*, in TGIF we account for the heterogeneity of the datasets by modeling the tasks to be related by a tree or a hierarchy. This makes TGIF applicable to a wide variety of task collections representing different biological contexts with arbitrary and complex relationships (e.g. Hi-C datasets from different cancer subtypes, cell lineage). In TGIF, the leaves of the tree correspond to the observed dataset while the internal node describe which tasks are most related. The child tasks are then regularized to its immediate parent task.

1.1 Notation and objective

In TGIF, given $t \in \{1, \ldots, T\}$ tasks, each with input matrix $X^{(t)} \in \mathbb{R}^{n_t \times m}$, related to each other in a task hierarchy/tree with a set of nodes $c \in \{r\} \cup \mathcal{B} \cup \mathcal{T}$ where r is the root node, $\mathcal{B}$ a set of internal (or branch)

nodes $b \in \mathcal{B}$, and $\mathcal{T}$ a set of the task-specific leaf nodes, the objective is:

$$O = \sum_{t=1}^{T} \left\| X^{(t)} - U^{(t)} V^{(t)\top} \right\|_{\mathrm{F}}^2 + \alpha \sum_{c} \left\| V^{(c)} - V^{Pa(c)} \right\|_{\mathrm{F}}^2 \tag{2}$$

where $U^{(t)} \in \mathbb{R}^{n_t \times k}$, $V^{(\cdot)} \in \mathbb{R}^{m \times k}$, $k \ll n, m$. The regularization term will:

1. constrain a task-specific latent feature factor $V^{(t)}$ in a leaf node of the task hierarchy to be similar to $V^{\mathrm{Pa}(t)}$ in its parent node;
2. constrain an internal node's latent feature factor $V^{(b)}$ to be similar to its direct child nodes' $V^{(c)}$ and and its parent node's $V^{Pa(b)}$; and
3. constrain the root node's latent feature factor $V^{(r)}$ to be similar to all of its direct child nodes' $V^{(c)}$s.

1.2 Updates rules based on Block Coordinate Descent (BCD)

We chose a block coordinate descent (BCD) optimization scheme to learn these factors because BCD guarantees convergence to a local optimum (Kim et al., 2014). Intuitively, block coordinate descent updates a given block while keeping all other blocks fixed; in TGIF the block is each column of $\mathrm{U}^{(\cdot)}$s or $\mathrm{V}^{(\cdot)}$s.

The objective of TGIF can be re-written as:

$$O = \sum_{t=1}^{T} \left\| X^{(t)} - \sum_{k} u_k^{(t)} v_k^{(t)\top} \right\|_{\mathrm{F}}^2 + \alpha \sum_{c} \sum_{k} \left\| v_k^{(c)} - v_k^{Pa(c)} \right\|_2^2 \tag{3}$$

Where $u_k^{(t)} \in \mathbb{R}^{n_t}$ is the kth column vector of $U^{(t)}$ and $v_k^{(t)} \in \mathbb{R}^m$ is the kth column vector of $V^{(t)}$. Now we 'pull out' terms involving the kth column in all factors:

$$O = \sum_{t=1}^{T} \left\| X^{(t)} - u_k^{(t)} v_k^{(t)\top} - \sum_{j \neq k} u_j^{(t)} v_j^{(t)\top} \right\|_{\mathrm{F}}^2 + \alpha \sum_{c} \left(\left\| v_k^{(c)} - u_k^{Pa(c)} \right\|_2^2 + \sum_{j \neq k} \left\| v_j^{(c)} - v_j^{Pa(c)} \right\|_2^2 \right) \tag{4}$$

Now we will substitute with $R_k^{(t)} = X^{(t)} - \sum_{j \neq k} u_j^{(t)} v_j^{(t)\top}$:

$$O = \sum_{t=1}^{T} \left\| R_k^{(t)} - u_k^{(t)} v_k^{(t)\top} \right\|_{\mathrm{F}}^2 + \alpha \sum_{c} \left\| v_k^{(c)} - v_k^{Pa(c)} \right\|_2^2 + \alpha \sum_{c} \sum_{j \neq k} \left\| v_j^{(c)} - v_j^{Pa(c)} \right\|_2^2 \tag{5}$$

We can now attempt to optimize $u_k^{(t)}$ and $v_k^{(\cdot)}$, fixing all other parameters to be constant.

Optimize $v_k^{(t)}$ To find $v_k^{(t)}$ for each leaf node task t that minimizes the objective, we find the derivative of the objective with respect to $v_k^{(t)}$ and set it to 0, then solve. First we expand the objective into matrix multiplications:

$$O = \left\| R_k^{(t)} - u_k^{(t)} v_k^{(t)\top} \right\|_{\mathrm{F}}^2 + \alpha \left\| v_k^{(t)} - v_k^{Pa(t)} \right\|_2^2 + C \tag{6}$$

$$= \mathrm{Tr} \left[\left(R_k^{(t)} - u_k^{(t)} v_k^{(t)\top} \right)^\top \left(R_k^{(t)} - u_k^{(t)} v_k^{(t)\top} \right) \right] + \alpha \left(v_k^{(t)} - v_k^{Pa(t)} \right)^\top \left(v_k^{(t)} - v_k^{Pa(t)} \right) + C \tag{7}$$

Here C subsumes all elements of the objective that does not involve $v_k^{(t)}$ (including terms involving tasks other than t), since they will be zeroed out when the derivative is taken with respect to $v_k^{(t)}$. Now we keep expanding:

$$O = \mathrm{Tr}\left[R_k^{(t)^\top} R_k^{(t)} - 2R_k^{(t)\top} u_k^{(t)} v_k^{(t)\top} + \left(u_k^{(t)} v_k^{(t)\top}\right)^\top \left(u_k^{(t)} v_k^{(t)\top}\right)\right] \tag{8}$$

$$+ \alpha\left(v_k^{(t)\top} v_k^{(t)} - 2v_k^{(t)\top} v_k^{Pa(t)} + v_k^{Pa(t)\top} v_k^{Pa(t)}\right) + C \tag{9}$$

$$= \mathrm{Tr}\left(R_k^{(t)^\top} R_k^{(t)}\right) - 2\,\mathrm{Tr}\left(R_k^{(t)\top} u_k^{(t)} v_k^{(t)\top}\right) + \mathrm{Tr}\left(v_k^{(t)} u_k^{(t)\top} u_k^{(t)} v_k^{(t)\top}\right) \tag{10}$$

$$+ \alpha v_k^{(t)\top} v_k^{(t)} - 2\alpha v_k^{(t)\top} v_k^{Pa(t)} + \alpha v_k^{Pa(t)\top} v_k^{Pa(t)} + C \tag{11}$$

$$= \mathrm{Tr}\left(R_k^{(t)^\top} R_k^{(t)}\right) - 2\left(R_k^{(t)\top} u_k^{(t)}\right)^\top v_k^{(t)} + \left(u_k^{(t)\top} u_k^{(t)}\right)\left(v_k^{(t)\top} v_k^{(t)}\right) \tag{12}$$

$$+ \alpha v_k^{(t)\top} v_k^{(t)} - 2\alpha v_k^{(t)\top} v_k^{Pa(t)} + \alpha v_k^{Pa(t)\top} v_k^{Pa(t)} + C \tag{13}$$

Now we take the derivative of O w.r.t. $v_k^{(t)}$:

$$\frac{\partial O}{\partial v_k^{(t)}} = 0 - 2R_k^{(t)\top} u_k^{(t)} + 2v_k^{(t)} u_k^{(t)\top} u_k^{(t)} + 2\alpha v_k^{(t)} - 2\alpha v_k^{Pa(t)} + 0 + 0 \tag{14}$$

$$0 = -R_k^{(t)\top} u_k^{(t)} + \left(u_k^{(t)\top} u_k^{(t)} + \alpha\right) v_k^{(t)} - \alpha v_k^{Pa(t)} \tag{15}$$

$$v_k^{(t)} = \frac{R_k^{(t)\top} u_k^{(t)} + \alpha v_k^{Pa(t)}}{\left\|u_k^{(t)}\right\|_2^2 + \alpha} \tag{16}$$

With the non-negativity constraint $v_k^{(t)} \geq 0$, we want $R_k^{(t)\top} u_k^{(t)} + \alpha v_k^{Pa(t)} \geq 0$, because if $R_k^{(t)\top} u_k^{(t)} + \alpha v_k^{Pa(t)} < 0$, O will increase in (12) and (13). So the finalized update rule is:

$$v_k^{(t)} = \frac{\left[R_k^{(t)\top} u_k^{(t)} + \alpha v_k^{Pa(t)}\right]_+}{\left\|u_k^{(t)}\right\|_2^2 + \alpha} \tag{17}$$

Optimize $u_k^{(t)}$ We can derive the update rule for $u_k^{(t)}$ in leaf node task t similarly but much more simply. From (13), we take the derivative of O_t with respect to $u_k^{(t)}$; all regularization terms will zero out since they do not involve $u_k^{(t)}$. Hence the final update rule for $u_k^{(t)}$ is:

$$u_k^{(t)} = \frac{\left[R_k^{(t)} v_k^{(t)}\right]_+}{\left\|v_k^{(t)}\right\|_2^2} \tag{18}$$

Optimize $v_k^{(r)}$ For the overall consensus factor in the root of the task hierarchy, $v_k^{(r)}$, we can again ignore terms that do not involve $v_k^{(r)}$ in the objective (5). Note that we're going to collect the terms involving nodes

c whose parent is the root node, i.e. $\text{Pa}(c) = r$:

$$O = \alpha \sum_{c \in \text{Child}(r)} \left\| v_k^{(c)} - v_k^{(r)} \right\|_2^2 + C \tag{19}$$

$$= \alpha \sum_{c \in \text{Child}(r)} \left(v_k^{(c)} - v_k^{(r)} \right)^\top \left(v_k^{(c)} - v_k^{(r)} \right) + C \tag{20}$$

$$= \alpha \sum_{c \in \text{Child}(r)} \left[v_k^{(c)\top} v_k^{(c)} - 2 v_k^{(c)\top} v_k^{(r)} + v_k^{(r)\top} v_k^{(r)} \right] + C \tag{21}$$

$$= C - \sum_{c \in \text{Child}(r)} 2\alpha v_k^{(c)\top} v_k^{(r)} + \sum_{c \in \text{Child}(r)} \alpha v_k^{(r)\top} v_k^{(r)} \tag{22}$$

Now we take the derivative, set to 0, and solve:

$$\frac{\partial O}{\partial v_k^{(r)}} = 0 - \sum_{c \in \text{Child}(r)} 2\alpha v_k^{(c)} + \sum_{c \in \text{Child}(r)} 2\alpha v_k^{(r)} \tag{23}$$

$$0 = - \sum_{c \in \text{Child}(r)} v_k^{(c)} + |\text{Child}(r)| \cdot v_k^{(r)} \tag{24}$$

$$v_k^{(r)} = \frac{\sum_{c \in \text{Child}(r)} v_k^{(c)}}{|\text{Child}(r)|} \tag{25}$$

where $|\text{Child}(r)|$ is the number of direct child nodes of the root node r.

Optimize $v_k^{(b)}$ For the latent feature factor in an internal/branch node of the task hierarchy, $v_k^{(b)}$, same drill as before: we ignore terms that do not involve $v_k^{(b)}$ for the particular node b of interest in the objective (5). This time we collect terms involving the parent node of b, i.e. $\text{Pa}(b)$, and nodes c whose parent is b, i.e. $\text{Pa}(c) = b$:

$$O = \alpha \left(\left\| v_k^{(b)} - v_k^{Pa(b)} \right\|_2^2 + \sum_{c \in \text{Child}(b)} \left\| v_k^{(c)} - v_k^{(b)} \right\|_2^2 \right) + C \tag{26}$$

$$= \alpha \left(v_k^{(b)} - v_k^{Pa(b)} \right)^\top \left(v_k^{(B)} - v_k^{Pa(b)} \right) + \alpha \sum_{c \in \text{Child}(b)} \left(v_k^{(c)} - v_k^{(b)} \right)^\top \left(v_k^{(c)} - v_k^{(b)} \right) + C \tag{27}$$

$$= \alpha \left[v_k^{(b)\top} v_k^{(b)} - 2 v_k^{(b)\top} v_k^{\text{Pa}(b)} + v_k^{\text{Pa}(b)\top} v_k^{\text{Pa}(b)} \right] + \alpha \sum_{c \in \text{Child}(b)} \left[v_k^{(c)\top} v_k^{(c)} - 2 v_k^{(c)\top} v_k^{(b)} + v_k^{(b)\top} v_k^{(b)} \right] + C \tag{28}$$

$$= \alpha v_k^{(b)\top} v_k^{(b)} - 2\alpha v_k^{(b)\top} v_k^{\text{Pa}(b)} - \sum_{c \in \text{Child}(b)} 2\alpha v_k^{(c)\top} v_k^{(b)} + \sum_{c \in \text{Child}(b)} \alpha v_k^{(b)\top} v_k^{(b)} + C \tag{29}$$

Now we take the derivative, set to 0, and solve:

$$\frac{\partial O}{\partial v_k^{(b)}} = 2\alpha v_k^{(b)} - 2\alpha v_k^{\text{Pa}(b)} - \sum_{c \in \text{Child}(b)} 2\alpha v_k^{(c)} + \sum_{c \in \text{Child}(b)} 2\alpha v_k^{(b)} \tag{30}$$

$$0 = v_k^{(b)} - v_k^{\text{Pa}(b)} - \sum_{c \in \text{Child}(b)} v_k^{(c)} - |\text{Child}(b)| \cdot v_k^{(b)} \tag{31}$$

$$= (1 + |\text{Child}(b)|) v_k^{(b)} - v_k^{\text{Pa}(b)} - \sum_{c \in \text{Child}(b)} v_k^{(c)} \tag{32}$$

$$v_k^{(b)} = \frac{v_k^{\text{Pa}(b)} + \sum_{c \in \text{Child}(b)} v_k^{(c)}}{1 + |\text{Child}(b)|} \tag{33}$$

where $|\text{Child}(b)|$ is the number of direct child nodes of b.

2 Generating randomized background matrix and null distribution of boundary scores.

We first calculate the element-wise mean across T input submatrices to yield M, then create randomized background matrix by shuffling the entries of M. The shuffling is done in a distance-stratified manner; that is, we obtain all pairs of genomic regions that are at a distance d and permute them independent of the region pairs that are at different distance than d. The distance d ranges from 0 to the size of each window (e.g. 2Mb), incremented by the size of each bin (e.g. 10kb). We next performed single-task NMF on this shuffled M matrix for the same range of k factors and derive boundary scores for all regions in the same way described in **Methods**. We treat this set of boundary scores as the samples from the null distribution.

3 Hyper-parameter selection for TGIF-DB.

To determine the setting of α, we examined the agreement between boundary assignments from a pair of input matrices for a given α value and the similarity of the input matrices themselves (SCC, Yang et al., 2017, **Figure S18A**). We used the cardiomyocyte differentiation data and scanned multiple hyper-parameter values in $\alpha \in \{10^2, 10^4, 10^6, 10^8\}$ across all chromosomes and compared the resulting boundary sets between every pair of timepoints using Jaccard index. In parallel, we measured the similarity of interaction count matrices between every pair of timepoints across all chromosomes using SCC, which is a weighted mean of correlation between interaction counts, stratified by genomic distance. SCC enables unbiased measurement of similarity between Hi-C datasets which are heavily distance-dependent (i.e., closer genomic regions tend to have higher interactions). Finally, we measured the correlation between the Jaccard index and the SCC within each chromosome, across all pairs of timepoints (**Figure S18B,C**). We find a slight, though not significant, improvement with $\alpha = 10^6$, which we set as default for TGIF-DB. As BCD is a stochastic algorithm that can reach different local optima depending on the initialization point, we also experimented with multiple random initialization seeds. We used Jaccard index to measure the agreement between pairs of boundary sets from two different seeds, with α fixed at the default value 10^6. We found that the resulting boundary sets from different initialization are relatively consistent with pairwise Jaccard index 0.76-0.77 (**Figure S18D**). We also estimated the memory usage and run time trend of TGIF-DB on 10kb input matrices from the three different timecourse datasets (**Figure S19A,B**). TGIF-DB's submatrix factorization framework with fixed set of k makes it scale linearly with the size of the input matrices.

4 Annotating significantly differential boundary type.

Between a pair of conditions A and B compared, TGIF-DB outputs a list of significantly differential boundary regions (sigDB). Each sigDB is annotated as a boundary deleted in B, created in B, or shifted (by 5 genomic bins in B). A boundary deletion is defined by a sigDB that exists in A but no sigDB is found within a radius of 5 genomic bins in B. A boundary creation is similarly defined by a sigDB that exists in B but no sigDB is found within a radius of 5 genomic bins in A. Finally, a boundary shift is when a sigDB in A is found in B within a radius of 5 genomic bins.

5 Hyper-parameter selection for TGIF-DC.

To determine the best setting of α, we used a similar approach as for TGIF-DB, measuring the agreement (correlation) between the input matrix similarity and the agreement of TGIF-DC compartment assignments for the same inputs. Specifically, we measured the similarity of observed-over-expected (O/E) count matrices between every pair of timepoints across all chromosomes by flattening them into a vector and measuring correlation. Note, O/E counts are already normalized by distance. In parallel, we measured the similarity of the output clusters (compartments) with Rand index (**Figure S20A**). We used the mouse neural differentiation data to study this parameter, scanning $\alpha \in \{10^2, 10^4, 10^6, 10^8\}$ across chromosomes (**Figure S20B**). TGIF-DC consistently yields cluster assignments that are well correlated ($\sim$0.9) with the input matrix similarity across wide a range of α values (**Figure S20C**). Our results are from $\alpha = 10^4$, which is the default for TGIF-DC.

Similar to TGIF-DB, we also examined the stability of compartment assignments with multiple random initialization seeds with $\alpha = 10^4$ using Rand Index between pairs of cluster assignments from two different seeds. At $k = 2$, the compartment setting of TGIF-DC, the Rand Index ranged from 0.99-1 for all pairs of random initializations (**Figure S20D**). At $k = 5$, the subcompartment setting, Rand Index of resulting subcompartments was 0.7-0.8 (**Figure S20E**), showing that TGIF-DC yields a stable set of cluster assignments regardless of random initialization. Finally, we report the memory usage and run time trend of TGIF-DC on 100kb input matrices from the three different timecourse datasets (**Figure S21A,B**). TGIF-DC can analyze 6 datasets needing no more 0.5GB of memory and 400 seconds of run time.

6 Datasets used in analysis

The Hi-C interaction count matrices for pluripotent H1 hESC cell line and for endoderm differentiated from H1 was downloaded from 4D Nucleome consortium (Reiff et al., 2022; Dekker et al., 2017, **Table S1**). 100kb intra-chromosomal count matrices were used for comparison of compartment-calling methods. Additionally, ATAC-seq data for H1 and endoderm was also downloaded and used to measure the accessibility of each 100kb region (i.e. mean ATAC-seq reads per base in the region) in the comparison of compartment-calling methods. 10kb VCSQRT-normalized intra-chromosomal count matrices was used in the analysis of differential boundary and expression analysis.

The mouse neural differentiation data (Bonev et al., 2017) included 3 timepoints during mouse neural differentiation: embryonic stem cell (ES) stage, neural progenitor stage (NPC), and the differentiated cortical neuron (CN) stage. For TGIF-DB, we used intra-chromosomal count matrices at a resolution of 10kb resolution with vanilla-coverage square-root (VCSQRT) normalization as input. When benchmarking boundary-calling methods, we also used 25kb and 50kb VC-SQRT-normalized data, as well as 25kb ICE-normalized data. For TGIF-DC, intra-chromosomal interaction count matrices at 100kb resolution without

normalization was used as input since TGIF-DC computes the O/E correlation matrices. In addition to the Hi-C measurements, this dataset also included ChIP-seq data for 6 histone modification marks (H3K27ac, H3K28me3, H3K36me3, H3K4me1, H3K4me3,H3K9me3), for both CN and NPC (**Table S2**). This data was used to characterize and validate the chromatin structure inferred by TGIF-DC. For each 100kb bin, the ChIP-seq signal in reads-per-million was summed up within each bin, and the signal in each bin was divided by the total number of reads to first normalize by read depth. Subsequently, signals in NPC and CN were quantile normalized to each other to enable log fold change comparison across the two timepoints. The log fold change for each of the 6 marks in each bin was calculated by log transforming (log(x+1)) the normalized signal in NPC divided by the signal in CN. The set of log fold change signals were used as input to a k-means clustering, with $k = 5$ clusters applied to sigDC regions.

The cardiomyocyte differentiation dataset from Zhang et al., 2019 (**Table S3**) measured Hi-C counts at 6 different timepoints (day 0, 2, 5, 7, 15, 80) from the human embryonic stem cell stage (hESC, day 0) to the ventricular cardiomyocyte stage (day 80). Two replicates from each timepoint were first merged and intra-chromosomal interaction count matrices were generated at 10kb resolution with ICE normalization using the Juicer tool (Durand et al., 2016). The 10kb resolution matrix was provided as input to TGIF-DB. The merged dataset was also used to generate 100kb resolution intra-chromosomal count matrices for input to TGIF-DC.

The method section **Benchmarking with downsampled data to assess robustness to depth** contains details on how GM12878 cell line data was processed and used.

7 Description of methods used in benchmarking TGIF-DB.

GRiNCH (Lee and Roy, 2021) is a TAD identification method that also utilizes a variation of NMF. It applies graph-regularized NMF to an input intra-chromosomal matrix as a whole and uses the output factor to cluster the genomic regions; each cluster represents a TAD. The boundary of such clusters were used as TAD boundaries in our analysis. For pairwise differential analysis, boundaries found in both input matrices were considered shared boundaries, while those found in one and not the other were considered differential boundaries. Version 1.0.0 was used with default values for all optional hyperparameters.

We note that based on our previous work in GRiNCH, there are several design criteria we took into account while developing TGIF: (a) scalability and (b) fewer parameters. GRiNCH was developed for an entire chromosome where imposing a graph-based contiguity was important, but it is slower than TGIF applied to a single task. GRiNCH required the user to specify the rank of the low-dimensional space, while in TGIF the user need not do this. TGIF uses a 2MB tiling window approach and considers multiple ranks in the same framework enabling it to be much more scalable and applicable across large sets of conditions.

SpectralTAD (Cresswell et al., 2020) treats an input Hi-C matrix as a graph of interacting genomic regions, and applies eigen decomposition to its graph Laplacian. The eigenvectors are used as latent features of each genomic region to cluster them, with each cluster representing a TAD. Similar to GRiNCH, for pairwise differential analysis, boundaries found in both input matrices were considered shared boundaries; those found in one and not the other differential boundaries. Version 1.16.1 was used in our analysis.

TopDom (Shin et al., 2016) is a TAD identification method which was shown to be robust to noise and yielded TADs enriched in structural proteins such as CTCF (Dali and Blanchette, 2017; Lee and Roy, 2021). It generates a score for each bin along the chromosome, where the score is the mean interaction count between the given bin and a set of upstream and downstream neighbors (neighborhood size is a user-specified parameter). Putative TAD boundaries are picked from a set of bins whose score forms a local minimum; false positive boundaries are filtered out with a significance test. Differential boundaries are identified in a

pairwise manner similar to GRiNCH and SpectralTAD. Version 0.10.1 was used in our analysis, with the window size hyper-parameter set to the recommended value of 5 (Shin et al., 2016).

We report the distribution of TAD sizes in basepairs for these TAD-calling methods, along with TGIF-DB, in **Figure S22**. We also found that in 10kb mESC Hi-C data, TGIF-DB identified zero "singleton" TADs, i.e., no TAD with only a single 10kb bin as its member, out of 6215 TADs. SpectralTAD also identified no singleton TADs (18102 TADs total); GRiNCH identified 5 singleton TADs out of 2393; TopDom 127 singleton TADs out of 16327 TADs total.

TADCompare (Cresswell and Dozmorov, 2020) is a differential TAD identification method that can take as input a pair of Hi-C matrices as well as a time series of Hi-C matrices. It treats each Hi-C matrix as a graph where each genomic region is a node and the pairwise interaction is a weighted edge with the weight corresponding to the count. It performs eigen decomposition on the Graph Laplacian of each Hi-C matrix. Boundary scores are calculated between a pair of regions by computing the Euclidean distance between their corresponding rows in the eigenvectors. Differential boundary scores are calculated by taking the difference in the boundary scores for a pair of conditions and converting it into a z-score and using a threshold of 2 to define a differential boundary. Although TADCompare accepts time-series data differential boundaries are computed for only pairs of matrices at a time. Version 1.2.0 was used in our benchmarking analysis.

We note that TADCompare provides annotation of boundary changes, e.g., boundary creation, deletion, or shifting. For instance, a shifted boundary is a non-overlapping boundary within five bins (or another user-defined threshold) of a boundary in the target condition being compared to. Such definitions should change depending on context, resolution, depth of the data. As such, while the output files of TGIF-DB lists the set of significantly differential boundaries, TGIF-DB does *not* further classify or annotate them.

In addition to the above mentioned tools, we initially considered the TADsplimer (Wang et al., 2020), and HiCExplorer (Wolff et al., 2020) methods. TADsplimer was specifically developed to detect differential TAD identification; however, its implementation has an unresolved issue that fails to return any output or differential boundaries and was excluded from further analysis. HiCExplorer allows TAD finding followed by differential TAD analysis using its hicDifferentialTAD tool. However, hicDifferentialTAD expects the same TAD for different conditions and outputs TADs with significantly different interaction counts rather than detecting boundary changes. Due to these reasons, they were excluded from subsequent benchmarking.

8 Generating simulated Hi-C matrices with known ground-truth boundaries for benchmarking.

Forcato et al., 2017 originally simulated 171 TADs, which at 40kb resolution resulted in a target size of the simulated matrix similar to the size of the human chromosome 5 (180.92Mb). To generate the TADs for each of the four tasks or conditions, we start with these 171 TADs (referred to as A) at the root. Of these, 40 TADs are kept unchanged during the TAD change simulation process. We proceed down the tree branch keeping A unchanged for the left most branch, and performing different TAD change operations on the other branches and at each level detailed in **Figure S16B**. This results in a tree structure where TAD sets A and A_2 are most similar, followed by B_1 and B_2. Pairwise differences between the TAD sets are quantified in **Figure S16C**. The resulting TAD sets are considered our "gold-standard".

Given a TAD structure, we follow the simulation procedure from Forcato et al., 2017 to generate the counts. Each interaction count for a pair of regions (i, j) is sampled from a negative binomial distribution defined by two parameters: (1) the dispersion parameter is fixed at 0.01 and (2) the mean parameter $\mu_{i,j} = K(i - j + 1)^{-0.69}$, which is dependent on the distance $(i - j)$ between the two interacting regions i and j. The mean parameter $\mu_{i,j}$ decays as the distance between i and j grows to reflect how Hi-C interaction

counts decay as a function of pairwise distance. A prior value of 1 is added to this distance when calculating $\mu_{i,j}$. Here -0.69 is the rate of decay parameter estimated from a real contact matrix (chr5 from IMR90 cell line). Additionally, $K = 28$ if a pair of regions falls within the same TAD and $K = 4$ otherwise. Finally, we add noise to a randomly selected portion of counts by adding 2 to the mean parameter $\mu_{i,j}$ to the randomly sampled counts. We added different levels of noise, 0.1, 0.2, 0.3, and 0.4 to each of the matrices, corresponding to randomly sampled 10, 20, 30, 40% of the interactions with noise added.

9 Differential boundaries between conditions overlapping with differential boundaries between replicates.

As a measure of false-positive differences identified between two conditions (day 0 and day 2 of cardiomyocyte differentiation), we calculated, in each chromosome, the percentage of differential boundaries between day 0 and day 2 that overlap with differential boundaries identified between two biological replicates from day 0 or from day 2. The set of differential boundaries between replicates is considered false-positive differences. Lower percentage in **Figure S3B** means inter-condition differential boundary sets are more distinct from inter-replicated (false-positive) differential boundaries. For GRiNCH, SpectralTAD, and TopDom, differential boundaries between A and B are union of boundaries found in A but not in B, and those found in B but not in A. For TADCompare and TGIF-DB, differential boundaries are the significantly differential boundaries outputted by each method for a pair of conditions (day 0 replicate 1 vs day 2 replicate 1) or a pair of biological replicates within the same condition (day 0 replicate 1 vs day 0 replicate 2).

10 Description of methods used in benchmarking TGIF-DC.

The PCA-based method (Rao et al., 2014) applies PCA to the observed over expected count (O/E) correlation matrices. The first principal component (PC1) is used to assign genomic regions to two compartments; values equal to or above zero in PC1 are assigned to one compartment and those below zero are assigned to the other compartment. The actual annotation of each compartment as the active "A" or repressed "B" compartments is done by correlating the compartment structure to one-dimensional regulatory signals such as ATAC-seq assays, histone modifications, gene density, or GC content. We annotated PCA-based A and B compartments in mouse neural progenitors and cortical neurons with GC content and report the distribution of various histone modification signals in **Figure S23**.

Cscore (Zheng and Zheng, 2018) outputs a score that specifies the compartment of a region by modeling the interaction counts and genomic distance with a probabilistic model. Regions with Cscore values above or equal to zero were clustered into one compartment and those below another compartment.

Finally, dcHiC (Chakraborty et al., 2022) is a framework that can identify differential compartment regions. dcHiC performs fast PCA on distance-normalized correlation matrices, quantile-normalizes the PC values so they can be compared across multiple conditions, and identifies a set of genomic regions whose compartment scores (normalized PC values) are significantly different in any of the conditions using a chi-square test (Chakraborty et al., 2022). As the current version of dcHiC requires at least 2 replicates per condition or timepoint, we provide dcHiC with interaction counts for two replicates per condition. For the other methods we provide a replicate-merged count matrix per condition available from the 4D Nucleome consortium.

11 Post-hoc annotation of TGIF-DC clusters into A and B compartments in various datasets

For the H1 to endoderm differentiation dataset we used the available compendium ATAC-seq data from 4D Nucleome (Reiff et al., 2022; Dekker et al., 2023, **Table S1**) as a measurement of chromatin accessibility. For each chromosome, we measure the mean ATAC-seq reads per 100kb bin from H1 within each of the 2 clusters. The cluster with higher mean ATAC-seq signal is assigned to compartment A and the other to compartment B for both H1 and endoderm (this is possible due to the cluster correspondence across timepoints in TGIF-DC; **Figure S24A,B,C**). To validate the compartment annotation, we also measured the GC percentage within each 100kb bin for each compartment, and found that regions assigned to A compartment have higher GC content than those in B compartment, as expected (**Figure S24D**).

We proceeded similarly with cardiomyocyte differentiation data, except we used DNase-seq data for chromatin accessibility from day 0 (H9 cell line, hESC state, **Table S3**). We measure the mean DNase-seq reads per 100kb bin for both clusters in each chromosome. The cluster with higher mean DNase-seq signal is labeled compartment A and the other one is labeled compartment B across all timepoints (**Figure S25A,B**). To validate the compartment annotation, we also measured GC percentage and found that regions assigned to the A compartment has higher GC content (**Figure S25C**). Since H3K27ac was available for this dataset, we also compared the mean H3K27ac reads within each 100kb bin of each compartment and found higher H3K27ac signal in the A compartment compared to B (**Figure S25D**). This is consistent with the definitions of A and B compartments (Nichols and Corces, 2021; Bouwman et al., 2023) and provides further support for TGIF-DC's compartmentalization.

For mouse neural differentiation data, we only used GC content for compartment annotation. We performed this annotation for the ES timepoint and transferred it to the other timepoints. Briefly, for each chromosome, we measure the mean GC% per 100kb bin in each compartment. The compartment with higher mean GC content is called A and the other one B across all timepoints (**Figure S8**).

References

Baur B, Lee DI, Haag J, Chasman D, Gould M, and Roy S. 2022. Deciphering the Role of 3D Genome Organization in Breast Cancer Susceptibility. *Frontiers in Genetics* **12**.

Bonev B, Mendelson Cohen N, Szabo Q, Fritsch L, Papadopoulos GL, Lubling Y, Xu X, Lv X, Hugnot JP, Tanay A, et al.. 2017. Multiscale 3D Genome Rewiring during Mouse Neural Development. *Cell* **171**: 557–572.e24.

Bouwman BA, Crosetto N, and Bienko M. 2023. A GC-centered view of 3D genome organization. *Current Opinion in Genetics & Development* **78**: 102020.

Chakraborty A, Wang JG, and Ay F. 2022. dcHiC detects differential compartments across multiple Hi-C datasets. *Nature Communications* **13**: 6827. Number: 1 Publisher: Nature Publishing Group.

Cresswell KG and Dozmorov MG. 2020. TADCompare: An R Package for Differential and Temporal Analysis of Topologically Associated Domains. *Frontiers in Genetics* **11**. Publisher: Frontiers.

Cresswell KG, Stansfield JC, and Dozmorov MG. 2020. SpectralTAD: an R package for defining a hierarchy of topologically associated domains using spectral clustering. *BMC Bioinformatics* **21**: 319.

Dali R and Blanchette M. 2017. A critical assessment of topologically associating domain prediction tools. *Nucleic Acids Research* **45**: 2994–3005.

Dekker J, Alber F, Aufmkolk S, Beliveau BJ, Bruneau BG, Belmont AS, Bintu L, Boettiger A, Calandrelli R, Disteche CM, et al.. 2023. Spatial and temporal organization of the genome: Current state and future aims of the 4D nucleome project. *Molecular Cell* **83**: 2624–2640. Publisher: Elsevier.

Dekker J, Belmont AS, Guttman M, Leshyk VO, Lis JT, Lomvardas S, Mirny LA, O'Shea CC, Park PJ, Ren B, et al.. 2017. The 4D nucleome project. *Nature* **549**: 219–226. Number: 7671 Publisher: Nature Publishing Group.

Durand NC, Shamim MS, Machol I, Rao SSP, Huntley MH, Lander ES, and Aiden EL. 2016. Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments. *Cell Systems* **3**: 95–98. Publisher: Elsevier.

Forcato M, Nicoletti C, Pal K, Livi C, Ferrari F, and Bicciato S. 2017. Comparison of computational methods for Hi-C data analysis. *Nature Methods* **14**: 679–685.

Kim J, He Y, and Park H. 2014. Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. *Journal of Global Optimization* **58**: 285–319.

Lee DD and Seung HS. 2000. Algorithms for Non-negative Matrix Factorization. In *In NIPS*, volume 13, pp. 556–562.

Lee DI and Roy S. 2021. GRiNCH: simultaneous smoothing and detection of topological units of genome organization from sparse chromatin contact count matrices with matrix factorization. *Genome Biology* **22**: 164.

Liu J, Wang C, Gao J, and Han J. 2013. Multi-View Clustering via Joint Nonnegative Matrix Factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining* (eds. J Ghosh, Z Obradovic, J Dy, ZH Zhou, C Kamath, and S Parthasarathy), pp. 252–260. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Nichols MH and Corces VG. 2021. Principles of 3D compartmentalization of the human genome. *Cell Reports* **35**: 109330.

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* **159**: 1665–1680.

Reiff SB, Schroeder AJ, Kırlı K, Cosolo A, Bakker C, Mercado L, Lee S, Veit AD, Balashov AK, Vitzthum C, et al.. 2022. The 4D Nucleome Data Portal as a resource for searching and visualizing curated nucleomics data. *Nature Communications* **13**: 2365. Number: 1 Publisher: Nature Publishing Group.

Shin H, Shi Y, Dai C, Tjong H, Gong K, Alber F, and Zhou XJ. 2016. TopDom: an efficient and deterministic method for identifying topological domains in genomes. *Nucleic Acids Research* **44**: e70.

Wang G, Meng Q, Xia B, Zhang S, Lv J, Zhao D, Li Y, Wang X, Zhang L, Cooke JP, et al.. 2020. TADsplimer reveals splits and mergers of topologically associating domains for epigenetic regulation of transcription. *Genome Biology* **21**: 84.

Wolff J, Rabbani L, Gilsbach R, Richard G, Manke T, Backofen R, and Grüning BA. 2020. Galaxy HiCExplorer 3: a web server for reproducible Hi-C, capture Hi-C and single-cell Hi-C data analysis, quality control and visualization. *Nucleic Acids Research* **48**: W177–W184.

Yang T, Zhang F, Yardimci GG, Song F, Hardison RC, Noble WS, Yue F, and Li Q. 2017. HiCRep: assessing the reproducibility of Hi-C data using a stratum- adjusted correlation coefficient. *Genome Research* p. gr.220640.117.

Zhang Y, Li T, Preissl S, Amaral ML, Grinstein JD, Farah EN, Destici E, Qiu Y, Hu R, Lee AY, et al.. 2019. Transcriptionally active HERV-H retrotransposons demarcate topologically associating domains in human pluripotent stem cells. *Nature Genetics* **51**: 1380–1388. Number: 9 Publisher: Nature Publishing Group.

Zheng X and Zheng Y. 2018. CscoreTool: fast Hi-C compartment analysis at high resolution. *Bioinformatics* **34**: 1568–1570.