4DNvestigator: Time Series Genomic Data Analysis Toolbox

Stephen Lindsly[1], Can Chen[2], Sijia Liu[3,4], Scott Ronquist[1], Samuel Dilworth[5], Michael Perlman[6], and Indika Rajapakse[1,2,*]

[1]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109, USA
[2]Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA
[3]MIT-IBM Watson AI Lab, IBM Research, Cambridge, MA 02142, USA
[4]Department of Computer Science, Michigan State University, MI 48824, USA
[5]iReprogram, Ann Arbor, MI 48105, USA
[6]Department of Statistics, University of Washington, Seattle, WA 98195, USA

[*]To whom correspondence should be addressed (indikar@umich.edu).

March 24, 2021

Abstract

Data on genome organization and output over time, or the 4D Nucleome (4DN), require synthesis for meaningful interpretation. Development of tools for the efficient integration of these data is needed, especially for the time dimension. We present the "4DNvestigator", a user-friendly network based toolbox for the analysis of time series genome-wide genome structure (Hi-C) and gene expression (RNA-seq) data. Additionally, we provide methods to quantify network entropy, tensor entropy, and statistically significant changes in time series Hi-C data at different genomic scales.

Availability: https://github.com/lindsly/4DNvestigator
Keywords— 4DN, Centrality, Entropy, Networks, Time Series

1 Introduction

4D nuclear organization (4D Nucleome, 4DN) is defined by the dynamical interaction between 3D genome structure and function [1, 2, 3]. To analyze the 4DN, genome-wide chromosome conformation capture (Hi-C) and RNA sequencing (RNA-seq) are often used to observe genome structure and function, respectively (Figure 1A). The availability and volume of Hi-C and RNA-seq data is expected to increase as high throughput sequencing costs decline, thus the development of methods to analyze these data is imperative. The relationship of genome structure and function has been studied previously [4, 5, 6, 3, 7], yet comprehensive and accessible tools for 4DN analysis are underdeveloped. The 4DNvestigator is a unified toolbox that loads time series Hi-C and RNA-seq data, extracts important structural and functional features (Figure 1B), and conducts both established and novel 4DN data analysis methods. We show that network centrality can be integrated with gene expression to elucidate structural and functional changes through time, and provide relevant links to the NCBI and GeneCards databases for biological interpretation of these changes [8, 9]. Furthermore, we utilize entropy to quantify the uncertainty of genome structure, and present a simple statistical method for comparing two or more Hi-C matrices.

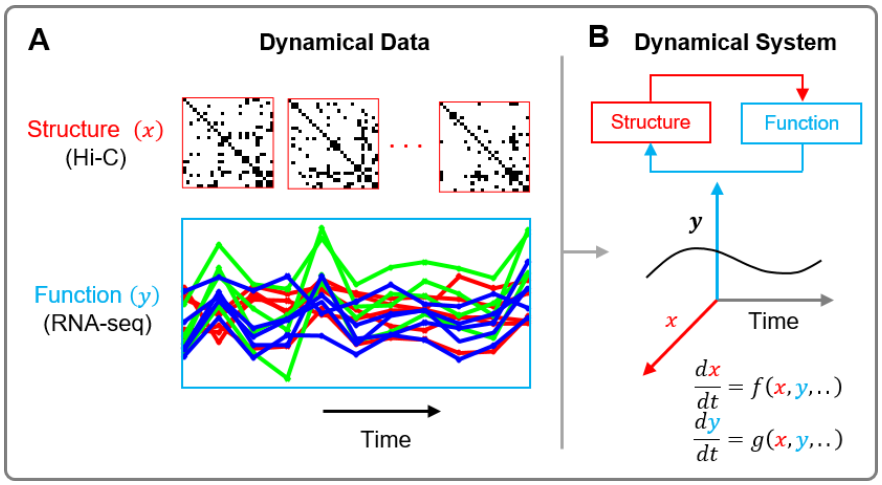


Figure 1: The 4D Nucleome. (A) Representative time series Hi-C and RNA-seq data correspond to genome structure and function, respectively. (B) Genome structure and function are intimately related. The 4DNvestigator integrates and visualizes time series data to study their dynamical relationship.

2 Materials and Methods

An overview of the 4DNvestigator workflow is depicted in Figure 2, and a Getting Started document is provided to guide the user through the main functionalities of the 4DNvestigator. The 4DNvestigator takes processed Hi-C and RNA-seq data as input, along with a metadata file which describes the sample and time

point for each input Hi-C and RNA-seq file (See Supplementary Materials "Data Preparation"). A number of novel methods for analyzing 4DN data are included within the 4DNvestigator and are described below.

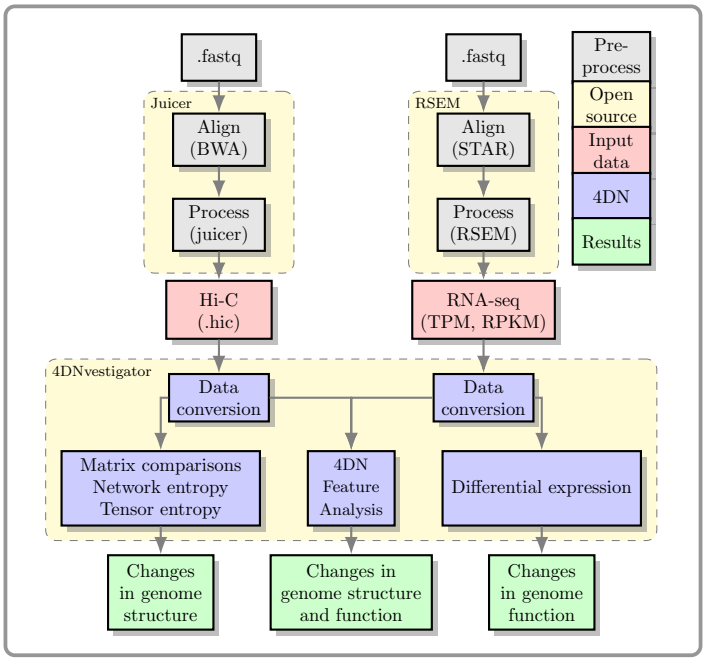


Figure 2: Overview of the 4DNvestigator data processing pipeline. Within this diagram, 4DN refers to the 4DNvestigator.

2.1 4DN Feature Analyzer

The "4DN feature analyzer" quantifies and visualizes how much a genomic region changes in structure and function over time. To analyze both structural and functional data, we consider the genome as a network. Nodes within this network are genomic loci, where a locus can be a gene or a genomic region at a particular resolution (i.e. 100 kb or 1 Mb bins). Edges in the genomic network are the relationships or interactions between genomic loci.

Algorithm 1: 4DN feature analyzer

Input: Hi-C matrices $\mathbf{A}^{(m)} \in \mathbb{R}^{n \times n}$, and RNA-seq vectors $\mathbf{r}^{(m)} \in \mathbb{R}^{n \times 1}$, $m = 1, \ldots, T$

Output: Low dimensional space $\mathbf{Y}^{(m)}$ and genes in loci with the largest structure-function changes

1 Compute degree, eigenvector, betweenness, and closeness centrality of $\mathbf{A}^{(m)}$, and define as $\mathbf{b}^{(m)}_{deg}$, $\mathbf{b}^{(m)}_{eig}$, $\mathbf{b}^{(m)}_{bet}$, $\mathbf{b}^{(m)}_{close}$, respectively, where each $\mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$

2 Compute the first principal component (PC1) of $\mathbf{A}^{(m)}$

3 Form the feature matrices $\mathbf{X}^{(m)} = [\mathbf{b}^{(m)}_{deg}, \mathbf{b}^{(m)}_{eig}, \mathbf{b}^{(m)}_{bet}, \mathbf{b}^{(m)}_{close}, \mathbf{r}^{(m)}]$, where $\mathbf{X}^{(m)} \in \mathbb{R}^{n \times 5}$

4 Normalize the columns of $\mathbf{X}^{(m)}$

5 Compute the common low dimensional space $\mathbf{Y}^{(m)}$

6 Visualize the low dimensional projection $\mathbf{Y}^{(m)}$ or 4DN phase plane

Return: $\mathbf{Y}^{(m)}$ and genes in loci with the largest structure-function changes

Structural Data

Structure in the 4DN feature analyzer is derived from Hi-C data. Hi-C determines the edge weights in our genomic network through the frequency of contacts between genomic loci. To analyze genomic networks, we adopt an important concept from network theory called centrality. Network centrality is motivated by the identification of nodes which are the most "central" or "important" within a network [10]. The 4DN feature analyzer uses *degree*, *eigenvector*, *betweenness*, and *closeness* centrality (step 1 of Algorithm 1), which have been shown to be biologically relevant [7]. For example, eigenvector centrality can identify structurally defined regions of active/inactive gene expression, since it encodes clustering information of a network [7, 11]. Additionally, betweenness centrality measures the importance of nodes in regard to the flow of information between pairs of nodes. Boundaries between euchromatin and heterochromatin, which often change in reprogramming experiments, can be identified in a genomic network through betweenness centrality [7].

Functional Data

Function in the 4DN feature analyzer is derived from gene expression through RNA-seq. Function is defined as the $\log_2$ transformation of Transcripts Per Million (TPM) or Reads Per Kilobase Million (RPKM). For regions containing more than one gene, the mean expression of all genes within the region is used. The 4DN feature analyzer can also use other one-dimensional features (e.g. ChIP-seq, DNase-seq, etc.). The interpretation of the results and visualizations would change accordingly.

Integration of Data

Hi-C data is naturally represented as a matrix of contacts between genomic loci. Network centrality measures are one-dimensional vectors that describe important structural features of the genomic network. We combine network centrality with RNA-seq expression to form a structure-function "feature" matrix that defines the state of each genomic region at each time point (Figure 3A, step 3 of Algorithm 1). Within this matrix, rows represent genomic loci and columns are the centrality measures (structure) and gene expression (function) of each locus. The z-score for each column is computed to normalize the data (step 4 of Algorithm 1).

4DN Analysis

The 4DN feature analyzer reduces the dimension of the structure-function feature matrix for visualization and further analysis (steps 5 and 6 of Algorithm 1). We include the main linear dimension reduction method, Principal Component Analysis (PCA), and multiple nonlinear dimension reduction methods: Laplacian Eigenmaps (LE) [12], t-distributed Stochastic Neighbor Embedding (t-SNE) [13], and Uniform Manifold Approximation and Projection (UMAP) [14] (Figure 3C). These methods are described in more detail in Supplementary Materials "Dimension Reduction". The 4DN feature analyzer can also visualize the dynamics of genome structure and function using the 4DN phase plane (step 6 of Algorithm 1) [3, 15]. We designate one axis of the 4DN phase plane as a measure of genome structure (e.g. eigenvector centrality) and the other as a measure of genome function (gene expression). Each point on the phase plane represents the structure and function of a genomic locus at a specific point in time (Figure 3B). The 4DN feature analyzer identifies genomic regions and genes with large changes in structure and function over time, and provides relevant links to the NCBI and GeneCard databases [8, 9].

Additional 4DNvestigator Tools

General Structure and Function Analysis

The 4DNvestigator also includes a suite of previously developed Hi-C and RNA-seq analysis methods. Euchromatin and heterochromatin compartments can be identified from Hi-C [4, 16], and regions that change compartments between samples are automatically identified. Significant changes in gene expression between RNA-seq samples can be determined through differential expression analysis using established methods [17].

Network Entropy

Entropy measures the amount of uncertainty within a system [18]. We use entropy to quantify the organization of chromatin structure from Hi-C data, where higher entropy corresponds to less structural organization. Since Hi-C is a multivariate analysis measurement (each contact coincidence involves two variables, the two genomic loci), we use multivariate entropy as follows:

$$\textbf{Entropy} = -\sum_j \lambda_j \ln \lambda_j, \tag{1}$$

where λ_i represents the dominant features of the Hi-C contact matrix. In mathematics, these dominant features are called eigenvalues [19]. Biologically, genomic regions with high entropy likely correlate with high proportions of euchromatin, as euchromatin is more structurally permissive than heterochromatin [20, 21]. Furthermore, entropy can be used to quantify stemness, since cells with high pluripotency are less defined in their chromatin structure [22]. We provide the full algorithm for network entropy and calculate the entropy of Hi-C data from multiple cell types in Supplementary Materials "Network Entropy".

Tensor Entropy

The notion of transcription factories supports the existence of simultaneous interactions involving three or more genomic loci [23]. This implies that the configuration of the human genome can be more accurately represented by k-uniform hypergraphs, a generalization of networks in which each edge can join exactly k nodes (e.g. a standard network is a 2-uniform hypergraph). We can construct k-uniform hypergraphs from Hi-C contact matrices by computing the multi-correlations of genomic loci. Tensor entropy, an extension of network entropy, measures the uncertainty or disorganization of uniform hypergraphs [24]. Tensor entropy can be computed from the same entropy formula (1) with generalized singular values λ_j from tensor theory [24, 25]. We provide the definitions for multi-correlation and generalized singular values, the algorithm to compute tensor entropy, and an application of tensor entropy on Hi-C data in Supplementary Materials "Tensor Entropy".

Larntz-Perlman Procedure

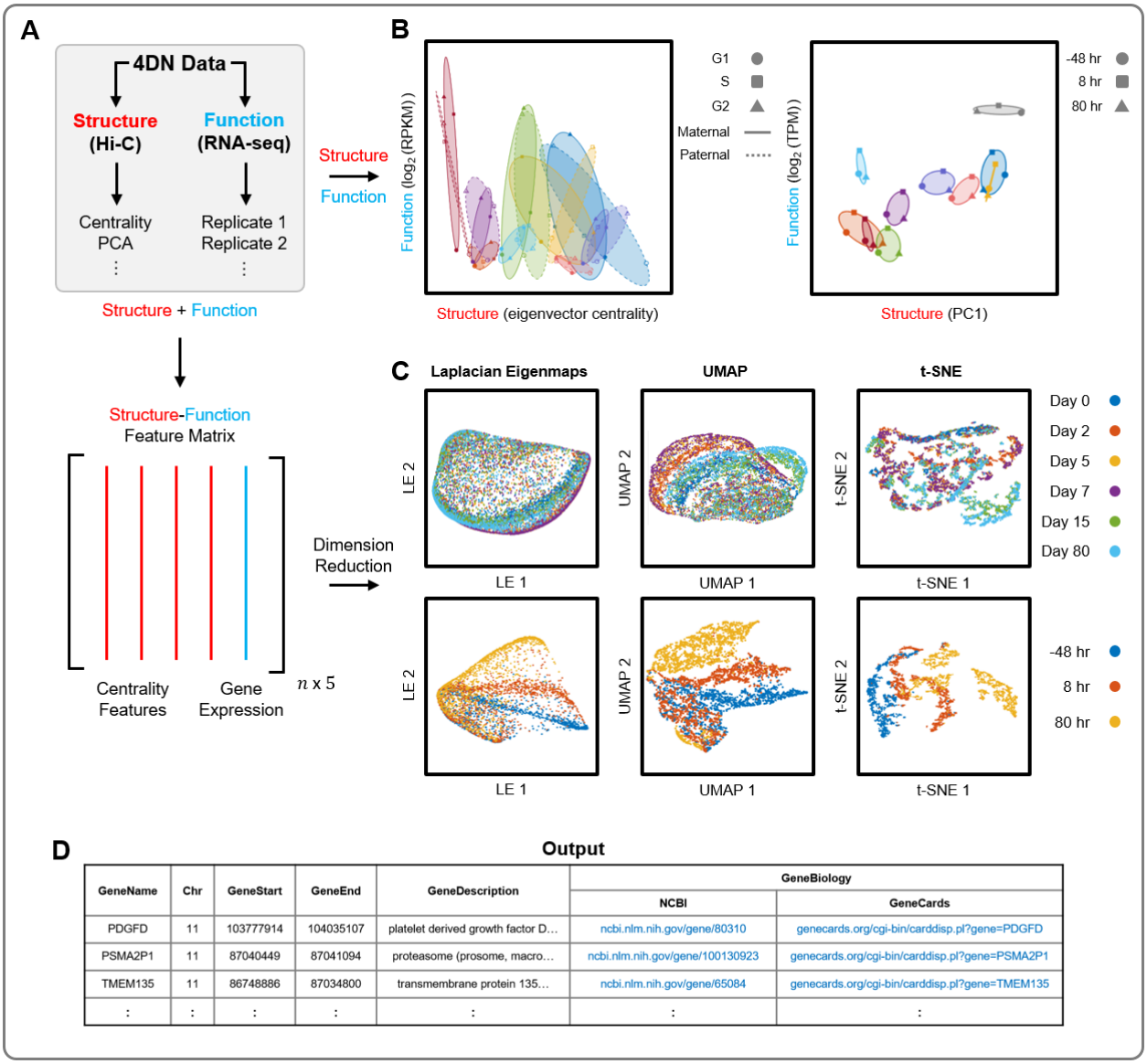
The 4DNvestigator includes a statistical test, proposed by Larntz and Perlman (the LP procedure), that compares correlation matrices [26, 27]. The LP procedure is applied to correlation matrices from Hi-C data, and is able to determine whether multiple Hi-C samples are significantly different from one another. Suppose that $\mathbf{C}^{(m)} \in \mathbb{R}^{n\times n}$ are the sample correlation matrices of Hi-C contacts with corresponding population correlation matrices $\mathbf{P}^{(m)} \in \mathbb{R}^{n\times n}$ for $m = 1, 2, \ldots, k$. The null hypothesis is $H_0 : \mathbf{P}^{(1)} = \cdots = \mathbf{P}^{(k)}$. First, compute the Fisher z-transformation $\mathbf{Z}^{(m)}$ by

$$\mathbf{Z}_{ij}^{(m)} = \frac{1}{2} \ln \frac{1 + \mathbf{C}_{ij}^{(m)}}{1 - \mathbf{C}_{ij}^{(m)}}. \tag{2}$$

Then form the matrices $\mathbf{S}^{(m)}$ such that

$$\mathbf{S}_{ij}^{(m)} = (n-3) \sum_{m=1}^{k} (\mathbf{Z}_{ij}^{(m)} - \bar{\mathbf{Z}}_{ij})^2, \tag{3}$$

where, $\bar{\mathbf{Z}}_{ij} = \frac{1}{k}\sum_{m=1}^{k} \mathbf{Z}_{ij}^{(m)}$. The test statistic is given by $T = \max_{ij} \mathbf{S}_{ij}$, and H_0 is rejected at level α if $T > \chi^2_{k-1,\epsilon(\alpha)}$ where $\chi^2_{k-1,\epsilon(\alpha)}$ is the chi-square distribution with $k-1$ degree of freedom, and $\epsilon(\alpha) = (1-\alpha)^{2/(n(n-1))}$ is the Šidák correction. Finally, calculate the p-value at which $T > \chi^2_{k-1,\epsilon(\alpha)}$. We note that this p-value is conservative, and that the the actual p-value may be smaller depending upon the amount of correlation among the variables. The LP procedure determines the statistical significance of any differences between multiple Hi-C samples for a genomic region of interest. We provide benchmark results of the LP procedure with other Hi-C comparison methods in Supplementary Materials "LP Procedure for Comparing Hi-C Matrices".



GeneName	Chr	GeneStart	GeneEnd	GeneDescription	GeneBiology	
					NCBI	GeneCards
PDGFD	11	103777914	104035107	platelet derived growth factor D…	ncbi.nlm.nih.gov/gene/80310	genecards.org/cgi-bin/carddisp.pl?gene=PDGFD
PSMA2P1	11	87040449	87041094	proteasome (prosome, macro…	ncbi.nlm.nih.gov/gene/100130923	genecards.org/cgi-bin/carddisp.pl?gene=PSMA2P1
TMEM135	11	86748886	87034800	transmembrane protein 135…	ncbi.nlm.nih.gov/gene/65084	genecards.org/cgi-bin/carddisp.pl?gene=TMEM135
:	:	:	:	:	:	:

Figure 3: 4DN feature analyzer. (A) 4DN data is input to the 4DN feature analyzer. Top: Structure data (Hi-C) is described using one dimensional features for compatibility with function data (RNA-seq). Bottom: Multiple structural features and function data are integrated into the structure-function feature matrix. (B) The 4DN feature analyzer can use structure and function data directly to visualize a system's dynamics using the 4DN phase plane [3, 15]. Structure defines the x-axis (left: eigenvector centrality, right: PC1) and function defines the y-axis (left: $\log_2$(RPKM), right: $\log_2$(TPM)), and points show structure-function coordinates through time. Left: Maternal and paternal alleles of nine cell cycle genes through G1, S, and G2/M phases of the cell cycle (adapted from [15]). Right: Top ten genomic regions (100 kb) with the largest changes in structure and function during cellular reprogramming [7]. (C) Multiple dimension reduction techniques can be used to visualize the 4DN feature analyzer's structure-function feature matrix (from left to right: LE, UMAP, and t-SNE). Top: 100 kb regions of Chromosome 4 across six time points during cellular differentiation [28]. Bottom: 100 kb regions of Chromosome 11 across three time points during cellular reprogramming [7]. (D) Example output of the 4DN feature analyzer. The output includes genes contained in loci with the largest changes, and links to their NCBI and GeneCards database entries [8, 9].

3 Results

We demonstrate how the 4DN feature analyzer can process time series structure and function data (Figure 3A) with three examples (Figure 3B-D).

Example 1: Cellular Proliferation. Hi-C and RNA-seq data from B-lymphoblastoid cells (NA12878) capture the G1, S, and G2/M phases of the cell cycle for the maternal and paternal genomes [15]. We visualize the structure-function dynamics of the maternal and paternal alleles for nine cell cycle regulating genes using the 4DN phase plane (Figure 3B, left). We are interested in the importance of these genes within the genomic network through the cell cycle, so we use eigenvector centrality as the structural measure. This analysis highlights the coordination between the maternal and paternal alleles of these genes through the cell cycle.

Example 2: Cellular Differentiation. We constructed a structure-function feature matrix from time series Hi-C and RNA-seq data obtained from differentiating human stem cells [28]. These data consist of six time points which include human embryonic stem cells, mesodermal cells, cardiac mesodermal cells, cardiac progenitors, primitive cardiomyocytes, and ventricular cardiomyocytes [28]. We analyze Chromosome 4 across the six time points in 100 kb resolution by applying three dimension reduction techniques to the structure-function feature matrix: LE, UMAP, and t-SNE (Figure 3C, top). There is a better separation of the cell types during differentiation using UMAP and t-SNE than from LE. The optimal methods for visualization and analysis are data dependent, so the 4DNvestigator offers multiple tools for the user's own exploration of their data.

Example 3: Cellular Reprogramming. Time series Hi-C and RNA-seq data were obtained from an experiment that reprogrammed human dermal fibroblasts to the skeletal muscle lineage [7]. We analyze samples collected 48 hr prior to, 8 hr after, and 80 hr after the addition of the transcription factor MYOD1. The ten 100 kb regions from Chromosome 11 that varied most in structure and function are visualized using the 4DN phase plane in Figure 3B (right). We also construct a structure-function feature matrix of Chromosome 11 in 100 kb resolution. Similar to the differentiation data analysis, we use LE, UMAP, and t-SNE to visualize the structure-function dynamics. These low dimensional projections show the separation of the three time points corresponding to before, during, and after cellular reprogramming (Figure 3C, bottom). We show an example output of the 4DN feature analyzer, which highlights genes contained in the genomic loci that have the largest structure-function changes through time and provides links to the NCBI and GeneCards database entries for these genes (Figure 3D) [8, 9].

4 Discussion

The 4DNvestigator provides rigorous and automated analysis of Hi-C and RNA-seq time series data by drawing on network theory, information theory, and multivariate statistics. It also introduces a simple statistical method for comparing Hi-C matrices, the LP procedure. The LP procedure is distinct from established Hi-C matrix comparison methods, as it takes a statistical approach to test for matrix equality, and allows for the comparison of many matrices simultaneously. Thus, the 4DNvestigator provides a comprehensive toolbox that can be applied to time series Hi-C and RNA-seq data simultaneously or independently. These methods are important for producing rigorous quantitative results in 4DN research.

5 Acknowledgments

We would like to thank Dr. Thomas Ried, Charles Ryan, and Gabrielle Dotson for feedback on the manuscript and helpful discussions.

6 Funding

This work is supported in part by the Air Force Office of Scientific Research (AFOSR) award #FA9550-18-1-0028, the Smale Institute, and a subcontract of the Defense Advanced Research Projects Agency (DARPA) award #140D6319C0020 to iReprogram, LLC.

7 Disclosure Statement

Samuel Dilworth is an employee of iReprogram.

8 Supplemental Material

Please refer to the file '4DNvestigator Supplementary Materials' for additional details on the 4DNvestigator's installation process, methods, and data availability.

References

[1] Job Dekker, Andrew S. Belmont, Mitchell Guttman, Victor O. Leshyk, John T. Lis, Stavros Lomvardas, Leonid A. Mirny, Clodagh C. O'Shea, Peter J. Park, Bing Ren, Joan C. Ritland Politz, Jay Shendure, and Sheng Zhong. The 4D nucleome project. *Nature*, 549(7671):219–226, 2017.

[2] Thomas Ried and Indika Rajapakse. The 4D Nucleome. *Methods*, 123:1–2, 2017.

[3] Haiming Chen, Jie Chen, Lindsey a. Muir, Scott Ronquist, Walter Meixner, Mats Ljungman, Thomas Ried, Stephen Smale, and Indika Rajapakse. Functional organization of the human 4D Nucleome. *Proceedings of the National Academy of Sciences*, 112(26):8002–8007, 2015.

[4] Erez Lieberman-aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, Richard Sandstrom, Bradley Bernstein, M A Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A Mirny, Eric S. Lander, and Job Dekker. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science*, 326:289–294, 2009.

[5] Jesse R. Dixon, Inkyung Jung, Siddarth Selvaraj, Yin Shen, Jessica E. Antosiewicz-Bourget, Ah Young Lee, Zhen Ye, Audrey Kim, Nisha Rajagopal, Wei Xie, Yarui Diao, Jing Liang, Huimin Zhao, Victor V. Lobanenkov, Joseph R. Ecker, James A. Thomson, and Bing Ren. Chromatin architecture reorganization during stem cell differentiation. *Nature*, 518(7539):331–336, 2 2015.

[6] Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, 2012.

[7] Sijia Liu, Haiming Chen, Scott Ronquist, Laura Seaman, Nicholas Ceglia, Walter Meixner, Pin-Yu Chen, Gerald Higgins, Pierre Baldi, Steve Smale, Alfred Hero, Lindsey A. Muir, and Indika Rajapakse. Genome Architecture Mediates Transcriptional Control of Human Myogenic Reprogramming. *iScience*, 6:232–246, 2018.

[8] David L Wheeler, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 36(suppl_1):D13–D21, 2007.

[9] Gil Stelzer, Naomi Rosen, Inbar Plaschkes, Shahar Zimmerman, Michal Twik, Simon Fishilevich, Tsippi Iny Stein, Ron Nudel, Iris Lieder, Yaron Mazor, et al. The genecards suite: from gene data mining to disease genome sequence analyses. *Current protocols in bioinformatics*, 54(1):1–30, 2016.

[10] Mark Newman. *Networks: an introduction*. Oxford university press, New York, 2010.

[11] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001.

[12] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *NIPS*, 14:585–591, 2001.

[13] Laurens Van Der Maaten and George E Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[14] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[15] Stephen Lindsly, Wenlong Jia, Haiming Chen, Sijia Liu, Scott Ronquist, Can Chen, Xingzhao Wen, Gabrielle A Dotson, Charles Ryan, Gilbert S Omenn, et al. Functional organization of the maternal and paternal human 4d nucleome. *bioRxiv*, 2020.

[16] Jie Chen, Alfred Hero, and Indika Rajapakse. Spectral Identification of Topological Domains. *Bioinformatics*, 32(14):2151–2158, 2016.

[17] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome biology*, 11(R106):1–12, 2010.

[18] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[19] Gilbert Strang. *Introduction to Linear Algebra*. Cambridge Press, 2016.

[20] Ben D Macarthur and Ihor R Lemischka. Statistical mechanics of pluripotency. *Cell*, 154(3):484–489, 2013.

[21] I. Rajapakse, M. Groudine, and M. Mesbahi. What can systems theory of networks offer to biology? *PLoS computational biology*, 8(6):e1002543, 2012.

[22] Eran Meshorer and Tom Misteli. Chromatin in pluripotent embryonic stem cells and differentiation. *Nature reviews Molecular cell biology*, 7(7):540, 2006.

[23] Peter R Cook and Davide Marenduzzo. Transcription-driven genome organization: a model for chromosome structure and the regulation of gene expression tested through simulations. *Nucleic acids research*, 46(19):9895–9906, 2018.

[24] C. Chen and I. Rajapakse. Tensor entropy for uniform hypergraphs. *IEEE Transactions on Network Science and Engineering*, 7(4):2889–2900, 2020.

[25] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[26] Kinley Larntz and Michael D Perlman. A simple test for the equality of correlation matrices. *Rapport technique, Department of Statistics, University of Washington*, 141, 1985.

[27] James A Koziol, Joel E Alexander, Lance O Bauer, Samuel Kuperman, Sandra Morzorati, Sean J O'connor, John Rohrbaugh, Bernice Porjesz, Henri Begleiter, and John Polich. A graphical technique for displaying correlation matrices. *The American Statistician*, 51(4):301–304, 1997.

[28] Yanxiao Zhang, Ting Li, Sebastian Preissl, Maria Luisa Amaral, Jonathan D Grinstein, Elie N Farah, Eugin Destici, Yunjiang Qiu, Rong Hu, Ah Young Lee, et al. Transcriptionally active herv-h retrotransposons demarcate topologically associating domains in human pluripotent stem cells. *Nature genetics*, 51(9):1380–1388, 2019.

4DNvestigator: Supplementary Materials

Stephen Lindsly[1], Can Chen[2], Sijia Liu[3,4], Scott Ronquist[1], Samuel Dilworth[5], Michael Perlman[6], and Indika Rajapakse[1,2,*]

[1]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109, USA
[2]Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA
[3]MIT-IBM Watson AI Lab, IBM Research, Cambridge, MA 02142, USA
[4]Department of Computer Science, Michigan State University, MI 48824, USA
[5]iReprogram, Ann Arbor, MI 48105, USA
[6]Department of Statistics, University of Washington, Seattle, WA 98195, USA

[*]To whom correspondence should be addressed.

March 24, 2021

4DNvestigator Overview and Setup

The 4DNvestigator toolbox can be downloaded and installed from the 4DNvestigator GitHub page. Installation instructions are included in the 4DNvestigator README along with all data used in this manuscript. We provide a script, ExampleScript.m, which shows how to load data into MATLAB and runs each of the four major functionalities of the 4DNvestigator: 4DN feature analyzer, network entropy, tensor entropy, and the Larntz-Perlman procedure. We also provide a Getting Started document which provides a concise description of each of the example functions' parameters and outputs.

Data Preparation

To convert input data to a MATLAB compatible format, the 4DNvestigator requires a metadata table, referred to as an "Index File", to describe the data. Rows correspond to sequencing data samples (RNA-seq or Hi-C), and the columns correspond to data descriptors. The Index File must have 5 columns with the following headers:

- "path" defines the computer path to the sequencing data
- "dataType" defines the type of sequencing data, either "hic" or "rnaseq"
- "sample" defines the sample, (e.g. "treatment" or "control")
- "timePoint" defines the time point of sample, (e.g. 0 or 24)
- "refGenome" defines the reference genome for the sample (e.g. hg19)

Index Files can be .csv, .tsv, or .xls. An example of an Index File is available here. The 4DNvestigator can automatically convert common file types for RNA-seq (.results and .rpkm) and Hi-C (.hic) data to be compatible with MATLAB. Users can also manually import gene expression vectors, Hi-C contacts, or other genomic data directly provided that similar data structures and naming conventions are used. For Hi-C, users can import a comma-separated file containing binned Hi-C contacts and use the 4DNvestigator's generic Hi-C data loading function to create the appropriate data structure.

MATLAB Dependencies

The 4DNvestigator requires the following MATLAB Toolboxes:

- Statistics and Machine Learning Toolbox
- Computer Vision Toolbox
- Bioinformatics Toolbox
- Image Processing Toolbox

System Requirements

The 4DNvestigator was written and tested on a Windows 10 machine with an Intel Core i7-8700 CPU and 32 Gb RAM using MATLAB R2019b. During testing, we timed each of the core functionalities' example functions using default settings, including figure generation and saving (Supplementary Table S1). We also measured peak RAM usage over function calls and data loading which was approximately 9.5 Gb (higher RAM usage is caused by higher resolution Hi-C matrices). Therefore, we suggest that users run a recent version of MATLAB (2019 or later) on a Windows 10 machine with at least 12 Gb of RAM for optimal performance. Each of the example functions in Supplementary Table S1 can be run with default parameters using ExampleScript.m, and a step-by-step guide of this script can be found in the Getting Started document.

ExampleScript.m Section	Description	Runtime
Load RNA-seq and Hi-C Data	Loads RNA-seq and Hi-C data using results from RSEM and Juicer, respectively	∼240 sec
featureAnalyzerExample.m	Example function for the 4DN feature analyzer, called using default settings	∼30 sec
entropyExample.m	Example function for entropy calculation on two cell types, called using default settings	∼6 sec
entropyExampleExpanded.m	Example function for entropy calculation on six cell types, called using default settings	∼45 sec
tensorEntropyExample.m	Example function for tensor entropy calculation on reprogramming data, called using default settings	∼6 sec
lpExample.m	Example function for the Larntz-Perman procedure on three samples, called using default settings	∼20 sec

Table S1: Mapping of core 4DNvestigator functionalities and their example functions within the toolbox. Each of these functions are called within ExampleScript.m. Here we list their respective approximate runtimes.

Dimension Reduction

The 4DN feature analyzer provides multiple methods for reducing the dimension of structure-function data. The first is Principal Component Analysis (PCA), a linear dimension reduction technique. PCA is performed by computing the eigendecomposition of the centered covariance matrix. PCA uses the eigenvectors associated with the largest eigenvalues as coordinates for the low dimensional space. Unlike PCA, the other dimension reduction techniques included in the 4DN feature analyzer are non-linear. Non-linear methods are based on the idea that high dimensional data often lie on a curved manifold in lower dimension. In the 4DN feature analyzer, we provide the choice between Laplacian Eigenmaps (LE), t-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximate Projection (UMAP). LE constructs a graph, defined as the similarities between genomic loci in terms of their structural and functional features. It then computes the eigendecomposition of the Laplacian matrix, and uses the eigenvectors associated with

the smallest non-zero eigenvalues as the low dimensional coordinates [1]. The Laplacian matrix is calculated by subtracting the graph's adjacency matrix from its degree matrix. The goal of both t-SNE and UMAP is to reduce the dimension of high dimensional data, while preserving the relationships between data. In other words, t-SNE and UMAP optimize the low dimensional mapping of data to ensure that data which are similar to one another in high dimensional space are also close in low dimensional space [2, 3]. UMAP is conceptually similar to t-SNE, but it is better at preserving global structure of data and more computationally efficient [3].

Network Entropy

Entropy measures the order within a system, where higher entropy corresponds to more disorder [4]. Here, we apply this measure to Hi-C data to quantify the order in chromatin structure. Biologically, genomic regions with high entropy likely correlate with high proportions of euchromatin, as euchromatin is more structurally permissive than heterochromatin [5, 6]. Furthermore, entropy can be used to quantify stemness, since cells with high pluripotency are less defined in their chromatin structure [7]. Since Hi-C is a multivariate analysis measurement (each contact coincidence involves two variables, the two loci), we use multivariate entropy. The algorithm to compute entropy is given in Algorithm 1. We provide two variants of the algorithm to calculate entropy. The first uses the correlation matrix of $\log_2$ transformed Hi-C data, while the second uses the Laplacian matrix from Hi-C data. We find that the first variant performs better at discriminating the stemness of cell types, so we use this variant as the default for the entropy calculation. For both variants of network entropy, we use the convention $0 \ln 0 = 0$.

Algorithm 1: Entropy Computation

Input: Hi-C matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, Entropy variant
Output: Entropy
if *variant == 'corr'* **then**
  Compute the correlation matrix $\mathbf{C} = \text{corr}(\log_2(\mathbf{A}))$
  Compute the eigenvalues λ_i of $\mathbf{C}$ using eigendecomposition
else if *variant == 'laplacian'* **then**
  Compute the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$
  Compute the eigenvalues λ_i of $\mathbf{L}$ using eigendecomposition
Normalize the eigenvalues: $\bar{\lambda}_j = \frac{\lambda_j}{\sum_{i=1}^{n} \lambda_i}$
Compute Entropy: $\textbf{Entropy} = -\sum_j \bar{\lambda}_j \ln \bar{\lambda}_j$
Return: Entropy

An application of entropy is shown in Figure S1. We use entropy to to quantify stemness on the following samples: human Embryonic Stem Cells (hESC), Human Umbilical Vein Endothelial Cells (HUVEC), human lung fibroblasts (IMR90), Human Foreskin Fibroblast cells (HFFc6), human B-lymphocyte cells (GM12878). These data were obtained from the 4DN Portal and other publicly available datasets [8, 9, 10]. A simple comparison between two cell types can be performed using entropyExample and an example of multiple cell types (Figure S1) can be performed using entropyExampleExpanded. Both examples are called by the script ExampleScript.m.

Tensor Entropy

Uniform hypergraphs can be naturally represented by tensors, which are multidimensional arrays generalized from vectors and matrices. Tensor entropy is a spectral measure which can decipher topological attributes of uniform hypergraphs [11]. Here, we try to partially recover the 3D configuration of the genome using uniform hypergraphs via multi-correlation, a generalization of Pearson correlation which can measure the strength of multivariate correlation (defined in Algorithm 2 Step 3) [12]. The reconstruction is able to provide more

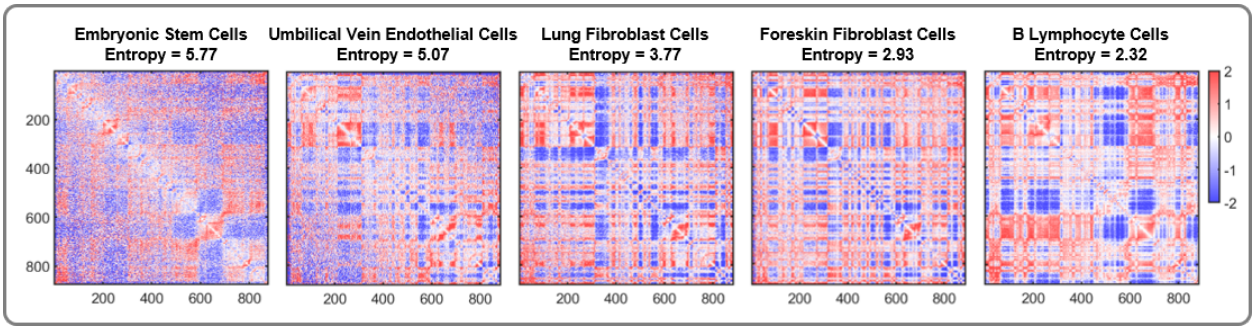


Figure S1: Entropy comparison between cell types. Hi-C matrices are shown in order of decreasing entropy (left to right). The more pluripotent cell lines (hESC, HUVEC) have a much higher entropy than the more differentiated cell lines (IMR90, HFFc6, GM12878). Intra-chromosome $\log_2$ Hi-C correlation matrices are shown at 100 kb resolution for Chromosome 14. The y-axis labels and color scale bar apply to all matrices in this figure.

information about genome structure and patterns, compared to the pairwise Hi-C contacts. Note that the optimal uniformity parameter k depends on the internal data structure, and we leave the choice of k to the users according to their data. Moreover, the constructions of adjacency tensor and degree tensor of a uniform hypergraph can be found in Chen *et al.* [11].

We borrow an example of tensor entropy in the case of cellular reprogramming from Chen *et al.* [11]. The dataset contains normalized Hi-C matrices from fibroblast proliferation and MYOD1-mediated fibroblast reprogramming (MYOD1 is the transcription factor used for control) for Chromosome 14 at 1 Mb resolution with a total of 89 genomic loci. We assume $k = 3$ and $\epsilon = 0.95$ in constructing the uniform hypergraphs since third-order contacts have been shown to occur more often than pairwise contacts in the genome [13]. We can successfully detect a bifurcation in the fibroblast proliferation and reprogramming data, and accurately identify the critical transition point between cell identities during reprogramming using tensor entropy (Figure S2A). To the contrary, the network entropy cannot offer valuable insights on the bifurcation, if one analyzes the Hi-C data as adjacency matrices directly (Figure S2B). This analysis can be performed using the function tensorEntropyExample which is called by the script ExampleScript.m.

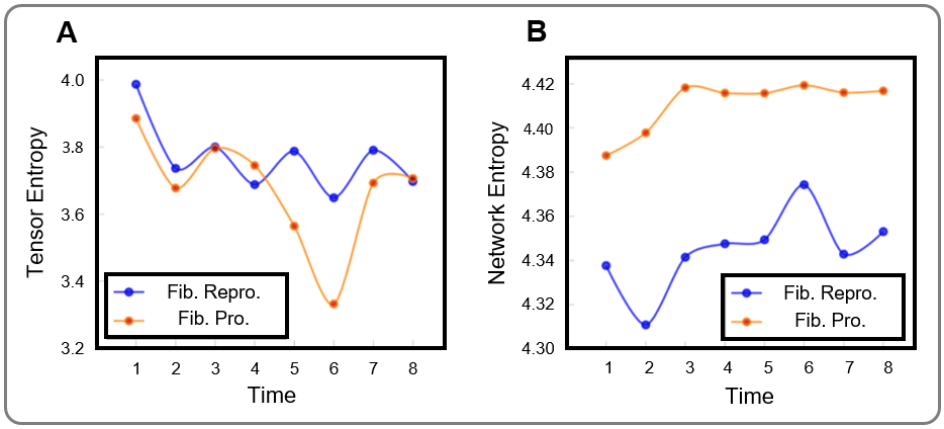


Figure S2: Cellular reprogramming features. (A) Tensor entropies of the uniform hypergraphs recovered from Hi-C measurements with multi-correlation cutoff threshold 0.95. (B) Network entropies of the binarized Hi-C matrices with weight cutoff threshold 0.95.

Algorithm 2: Tensor Entropy Computation

Input: Hi-C matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, uniformity parameter k, edge threshold ϵ

Output: Tensor Entropy

Compute the correlation matrix $\mathbf{C} = \text{corr}(\log_2(\mathbf{A}))$

Extract all $\binom{n}{k}$ submatrices $\mathbf{R} \in \mathbb{R}^{k \times k}$ where $\mathbf{R}$ is the correlation matrix of every k genomic loci

Compute the multi-correlation for every k genomic loci: $\rho = (1 - \det(\mathbf{R}))^{\frac{1}{2}}$

Build a k-uniform hypergraph based on the multi-correlations, i.e. an edge is created if $\rho \geq \epsilon$

Compute the Laplacian tensor $\mathsf{L} = \mathsf{D} - \mathsf{A}$ where $\mathsf{D} \in \mathbb{R}^{n \times n \times \overset{k}{\cdots} \times n}$ is the degree tensor, and A is the adjacency tensor of the k-uniform hypergraph

Reshape L by stacking the first $k-1$ dimensions, i.e. $\mathbf{L} \in \mathbb{R}^{n^{k-1} \times n}$

Compute the singular values λ_i of $\mathbf{L}$ using economy-size singular value decomposition (λ_i are also called the generalized singular values of L)

Normalize the generalized singular values: $\bar{\lambda}_j = \frac{\lambda_j}{\sum_{i=1}^{n} \lambda_i}$

Compute Tensor Entropy: $\textbf{Tensor Entropy} = -\sum_j \bar{\lambda}_j \ln \bar{\lambda}_j$

Return: Tensor Entropy

LP Procedure for Comparing Hi-C Matrices

A number of methods have been developed in the Hi-C research community to compare Hi-C matrices. These methods can be broadly grouped into 2 categories: Hi-C reproducibility methods (e.g. HiCRep, HiC-spector), and Differential Chromatin Interactions (DCI) methods (e.g. SELFISH, HiCCompare, FIND, diffHic) [14, 15, 16, 17, 18, 19]. Hi-C reproducibility methods are useful for assessing the equality of Hi-C matrices genome-wide, and detecting technical bias between samples. DCI methods determine which loci have a significantly different number of Hi-C contacts between samples. All methods listed above are comparisons between only two matrices.

We pose a new method for comparing Hi-C matrices, the Larntz-Perlman procedure (LP procedure). Elements within Hi-C matrices have been shown to be normally distributed when the $\log_2$ transformation is applied to $\mathbf{A}^{(m)}$ (See Supplementary Figure S3B) [20]. For data of this form, a method for testing the equality of correlation matrices was proposed by Larntz and Perlman in 1985 [21]. We recommend that the LP procedure is performed at Hi-C resolutions $\leq$ 100 kb and that regions do not extend >5 Mb, as signal (counts) often become sparse as the genomic distance between loci increases (See Figure S3A). The full algorithm is given in Algorithm 3. The LP procedure outputs a p-value for the equality of the Hi-C matrices and a matrix $\mathbf{S}$ where the largest values in $\mathbf{S}$ correspond to genomic regions that are most different between all samples. This analysis can be performed using the function lpExample which is called by the script ExampleScript.m.

Hi-C Matrix Comparison

In order to assess the LP procedure's ability to detect differences in Hi-C matrices between samples, we have compared the LP procedure against alternate Hi-C comparison methods: HiCRep, HiC-spector, and SELFISH [14, 15, 16]. HiCRep, HiC-spector are Hi-C reproducibility metrics, while SELFISH is a DCI method that was recently shown to outperform all prior methods for DCI detection [16]. We note that there are no methods that are a direct comparison with our method. All of the alternate methods we are using here were designed to address related, but fundamentally different, questions: Hi-C reproducibility metrics for genome-wide comparisons, and DCI for loci interaction differences (not overall structural differences within a region).

There are many ways in which Hi-C matrices can be different between samples. Chromatin loops and

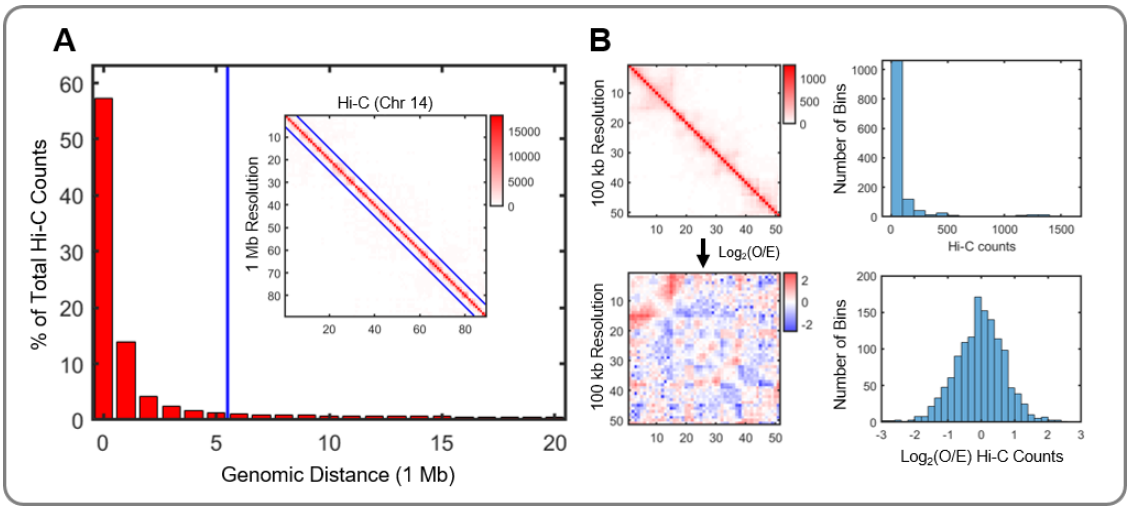


Figure S3: Hi-C normalization for the LP procedure. **(A)** Percentage of total Hi-C contacts by genomic distance. Inset figure shows a typical Hi-C intra-chromosome adjacency matrix at 1 Mb resolution. The blue line denotes where the genomic distance exceeds 5 Mb. **(B)** A 5 Mb Hi-C matrix (100 kb resolution, top left) and a histogram of the counts within the matrix (top right). When the $\log_2$(O/E) transformation is computed on the Hi-C matrix (bottom left), elements within the matrix are normally distributed (bottom right). Here, O/E is the observed Hi-C matrix divided by the expected counts based on genomic distance.

compartments are two features that can be detected through Hi-C, and are often different between cell states [9, 22]. To assess how the LP procedure performs when these features are different between samples, we have created simulated Hi-C data sets with incremental changes in these features (See details in Supplementary Materials "Simulated Hi-C data"). We then determined the point at which the LP procedure, as well as alternate methods, detect differences between the matrices (See Table S1). The LP procedure was tested for its ability to detect changes in loop structure and chromatin domains, but this method is generalizable to any scale provided that the input matrices are the same size and there are sufficient data. For example, the LP procedure could be used to detect differences across multiple samples within Topologically Associated Domains (TADs) that were identified through established TAD calling methods [9, 23].

For Hi-C matrices with changes in loop structure as the only differences between samples, the LP procedure does not detect that the matrices are different until the loop interaction contained six times the mean number of contacts for loci at the given genomic distance. Simulated matrices for this analysis are displayed in Figure S4A. In this situation, SELFISH does the best job of detecting differences between samples. HiC-spector shows a consistent decrease in its reproducibility score, while HiCRep performs similar to the LP procedure. We note that the LP procedure, as well as HiCRep, relies on changes in the correlation between samples, and thus changes to a small number of elements within the Hi-C matrices do not affect the measurement significantly.

For Hi-C matrices with changes in the compartment structure of a single bin (genomic region) as the only differences between samples, the LP procedure performs very well. Simulated matrices for this analysis are displayed in Figure S4B. SELFISH detects differences between all samples, but also detects differences when only a small amount of noise is added to the original matrix. The LP procedure is robust to noise and shows a trend consistent with the amount of change created. The HiC-spector reproducibility measurement shows no clear trend as the matrices diverge, and the LP procedure detects more subtle changes relative to HiCRep. Compartment changes are often observed in time series Hi-C matrices as cells transition between cell states, either through reprogramming or differentiation [24].

Algorithm 3: LP procedure

Input: Hi-C matrices $\mathbf{A}^{(m)} \in \mathbb{R}^{n \times n}$, $m = 1, \ldots, T$

Output: p-value p, and test statistic $\mathbf{S}$

1 Compute the correlation matrix $\mathbf{C}^{(m)} = \text{corr}(\log_2(\mathbf{A}^{(m)}))$. Define the corresponding population correlation matrices as $\mathbf{P}^{(m)}$

2 Define the null hypothesis

$$H_0 : \mathbf{P}^{(1)} = \cdots = \mathbf{P}^{(T)}$$

3 Compute the Fisher z-transformation $\mathbf{Z}^{(m)}$. Elements in $\mathbf{Z}^{(m)}$ and $\mathbf{C}^{(m)}$ are denoted as $z_{ij}^{(m)}$ and $c_{ij}^{(m)}$, respectively, and

$$z_{ij}^{(m)} = \frac{1}{2}\ln\left[\frac{1 + c_{ij}^{(m)}}{1 - c_{ij}^{(m)}}\right]$$

4 Form $\mathbf{S}$, where elements in $\mathbf{S}$ are denoted as s_{ij} and

$$s_{ij} = (n-3)\sum_{m=1}^{T}(z_{ij}^{(m)} - \bar{z}_{ij})^2, \ \bar{z}_{ij} = T^{-1}\sum_{m=1}^{T} z_{ij}^{(m)}$$

5 Calculate the test statistic

$$T = \max_{1 \le i < j \le n} s_{ij}$$

6 Reject H_0 at level α if $T > \chi^2_{T-1,\epsilon(\alpha)}$, where $\chi^2_{T-1,\epsilon(\alpha)}$ is the chi-squared distribution with $T-1$ degrees of freedom and $\epsilon(\alpha) = (1-\alpha)^{2/n(n-1)}$ is the Šidák correction

7 Calculate p, the level α at which $T > \chi^2_{T-1,\epsilon(\alpha)}$

Return: p and $\mathbf{S}$

Simulated Hi-C Data

Simulated Hi-C data was created to compare the LP procedure against alternative Hi-C comparison methods. Two distinct simulated data sets were created to have: (1) changes in chromatin loop structure and (2) changes in chromatin compartment structure. Chromatin loops and chromatin compartments are two features that have been used to characterize Hi-C structure. Both simulated data sets are created by perturbing data from real Hi-C matrices: a 400 kb region (10 kb resolution) is used for the chromatin loop data set and a 2 Mb region (50 kb resolution) is used for the chromatin compartment data set.

One additional simulated matrix was also created by adding a small amount of random noise to the 400 kb region (10 kb resolution) Hi-C matrix. A random number sampled from a normal distribution, $\mathcal{N}(0,\ 0.05)$, is added to each element in $\log_2(\mathbf{A})$. This matrix is used to determine how robust Hi-C matrix comparison methods are to small amounts of noise.

For each simulated data set, 10 matrices were created that are incrementally more divergent from the original Hi-C matrix. For changes in loop structure, counts were added to a specific off-diagonal region, following a 2D Gaussian distribution ($\sigma = 1$), to model a chromatin loop structure. For changes in compartment structure, counts aligned to a specific genomic region (bin) were changed to decrease the correlation coefficient between the specified bin in the simulated $\log_2$ Hi-C matrix and the specified bin in the original $\log_2$ Hi-C matrix. These changes reflect what would be observed if the specified bin was changing its compartment structure. Methods to recreate the simulated Hi-C matrices are provided within the 4DNvestigator.

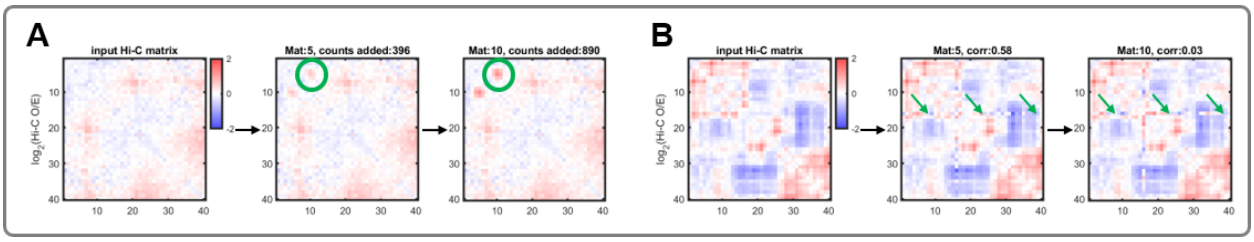


Figure S4: Simulated Hi-C data. **(A)** Simulated Hi-C matrices for loop changes (See Supplementary Table S2). The original Hi-C matrix is shown on the left, with matrices that are increasingly more divergent from left to right. Green circles indicate where the matrix has been perturbed to change the chromatin loop structure. **(B)** Simulated Hi-C matrices for compartment changes (See Supplementary Table S2). The original Hi-C matrix is shown on the left, with matrices that are increasingly more divergent from left to right. Green arrows indicate where the matrix has been perturbed to change the chromatin compartment structure.

	Counts added					Correlation					
Method	53	160	266	372	479	1 (noise)	0.87	0.57	0.2	-0.26	-0.65
LP (p-value)	1.0	1.0	0.99	0.41	0.045	1.0	1.0	2.1E-03	2.5E-07	1.7E-13	0.0
SELFISH (p-value)	0.0	0.0	0.0	0.0	0.0	0.02	2.9E-05	1.4E-08	1.3E-10	3.5E-13	2.1E-07
HiCRep (SCC)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.99	0.98	0.97	0.96
HiC-spector (Q)	0.99	0.98	0.96	0.94	0.92	0.39	0.43	0.33	0.48	0.44	0.50

Table S2: Hi-C matrix comparison methods. "Counts added" refers to the number of counts added to the simulated Hi-C matrix at the specified location. "Correlation" refers to the correlation between the selected column in the original Hi-C matrix, and the same column in the simulated Hi-C matrix. We note here that each method outputs a different unit, as specified within the table. The lowest p-value output from SELFISH is given here.

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *Nips*, 14:585–591, 2001.

[2] Laurens Van Der Maaten and George E Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[3] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[4] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[5] Ben D Macarthur and Ihor R Lemischka. Statistical mechanics of pluripotency. *Cell*, 154(3):484–489, 2013.

[6] I. Rajapakse, M. Groudine, and M. Mesbahi. What can systems theory of networks offer to biology? *PLoS computational biology*, 8(6):e1002543, 2012.

[7] Eran Meshorer and Tom Misteli. Chromatin in pluripotent embryonic stem cells and differentiation. *Nature reviews Molecular cell biology*, 7(7):540, 2006.

[8] Job Dekker, Andrew S. Belmont, Mitchell Guttman, Victor O. Leshyk, John T. Lis, Stavros Lomvardas, Leonid A. Mirny, Clodagh C. O'Shea, Peter J. Park, Bing Ren, Joan C. Ritland Politz, Jay Shendure, and Sheng Zhong. The 4D nucleome project. *Nature*, 549(7671):219–226, 2017.

[9] Suhas SP Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, and Erez Lieberman Aiden. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell*, 159(7):1665–1680, 2014.

[10] Emily M. Darrow, Miriam H. Huntley, Olga Dudchenko, Elena K. Stamenova, Neva C. Durand, Zhuo Sun, Su-Chen Huang, Adrian L. Sanborn, Ido Machol, Muhammad Shamim, Andrew P. Seberg, Eric S. Lander, Brian P. Chadwick, and Erez Lieberman Aiden. Deletion of DXZ4 on the human inactive X chromosome alters higher-order genome architecture. *Proceedings of the National Academy of Sciences*, 113(31):E4504–E4512, 2016.

[11] C. Chen and I. Rajapakse. Tensor entropy for uniform hypergraphs. *IEEE Transactions on Network Science and Engineering*, 7(4):2889–2900, 2020.

[12] Jianji Wang and Nanning Zheng. Measures of correlation for multiple variables. *arXiv preprint arXiv:1401.4827*, 2014.

[13] Netha Ulahannan, Matthew Pendleton, Aditya Deshpande, Stefan Schwenk, Julie M Behr, Xiaoguang Dai, Carly Tyer, Priyesh Rughani, Sarah Kudman, Emily Adney, et al. Nanopore sequencing of dna concatemers reveals higher-order features of chromatin structure. *bioRxiv*, page 833590, 2019.

[14] Tao Yang, Feipeng Zhang, Galip Gurkan Yardimci, Fan Song, Ross C Hardison, William Stafford Noble, Feng Yue, and Qunhua Li. HiCRep: assessing the reproducibility of Hi-C data using a stratum- adjusted correlation coefficient. *Genome Research*, 27(11):1939–1949, 2017.

[15] Koon Kiu Yan, Galip Gürkan Yardlmcl, Chengfei Yan, William S. Noble, and Mark Gerstein. HiC-spector: A matrix library for spectral and reproducibility analysis of Hi-C contact maps. *Bioinformatics*, 33(14):2199–2201, 2017.

[16] Abbas Roayaei Ardakany, Ferhat Ay, and Stefano Lonardi. Selfish: Discovery of Differential Chromatin Interactions via a Self-Similarity Measure. *bioRxiv*, page 540708, 2019.

[17] Mohamed Nadhir Djekidel, Yang Chen, and Michael Q. Zhang. FIND: DifFerential chromatin INteractions Detection using a spatial Poisson process. *Genome Research*, 28(3):412–422, 2018.

[18] John C. Stansfield, Kellen G. Cresswell, Vladimir I. Vladimirov, and Mikhail G. Dozmorov. HiCcompare: An R-package for joint normalization and comparison of HI-C datasets. *BMC Bioinformatics*, 19(1):13–16, 2018.

[19] Aaron TL Lun and Gordon K Smyth. diffHic: A Bioconductor package to detect differential genomic interactions in Hi-C data. *BMC Bioinformatics*, 16(1):1–11, 2015.

[20] Riccardo Calandrelli, Qiuyang Wu, Jihong Guan, and Sheng Zhong. GITAR: An Open Source Tool for Analysis and Visualization of Hi-C Data. *Genomics, Proteomics and Bioinformatics*, 16(5):365–372, 2018.

[21] Kinley Larntz and Michael D Perlman. A simple test for the equality of correlation matrices. *Rapport technique, Department of Statistics, University of Washington*, 141, 1985.

[22] Erez Lieberman-aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, Richard Sandstrom, Bradley Bernstein, M A Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A Mirny, Eric S. Lander, and Job Dekker. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science*, 326:289–294, 2009.

[23] Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380, 2012.

[24] Jesse R. Dixon, Inkyung Jung, Siddarth Selvaraj, Yin Shen, Jessica E. Antosiewicz-Bourget, Ah Young Lee, Zhen Ye, Audrey Kim, Nisha Rajagopal, Wei Xie, Yarui Diao, Jing Liang, Huimin Zhao, Victor V. Lobanenkov, Joseph R. Ecker, James A. Thomson, and Bing Ren. Chromatin architecture reorganization during stem cell differentiation. *Nature*, 518(7539):331–336, 2 2015.