SUPPLEMENTARY INFORMATION

PHi-C2: interpreting Hi-C data as the dynamic 3D genome state

Soya Shinkai¹, Hiroya Itoga¹, Koji Kyoda¹, and Shuichi Onami¹,²

¹Laboratory for Developmental Dynamics, RIKEN Center for Biosystems Dynamics Research, Kobe, 650-0047, Japan.
²Life Science Data Sharing Unit, Infrastructure Research and Development Division, RIKEN Information R&D and Strategy Headquarters, Kobe, 650-0047, Japan.

ABSTRACT

Supplementary information includes 4 Supplementary Notes, 5 Supplementary Figures, and 3 Supplementary Tables.
Supplementary Note S1: Inverse transformation to reconstruct an input Hi-C matrix

PHi-C is a simulation tool to decipher Hi-C data into polymer dynamics based on the mathematical formalism of the polymer network model. The normalized interaction matrix $\tilde{K} = (\tilde{K}_{ij})$ of the model has a one-to-one correspondence to a normalized contact matrix $\tilde{C} = (C_{ij})$ through the following matrix transformations, where the matrix size is $N \times N$: (i) $\tilde{K}$ is converted into the normalized Laplacian matrix $\tilde{L} = (L_{ij})$ by $\tilde{L} = \tilde{D} - \tilde{K}$, with the normalized degree matrix $\tilde{D} = \text{diag} (\tilde{D}_0, \tilde{D}_1, \ldots, \tilde{D}_{N-1})$, where $\tilde{D}_i = \sum_{j=0}^{N-1} \tilde{K}_{ij}$. (ii) As the Laplacian matrix $\tilde{L}$ is real symmetric, $\tilde{L}$ is diagonalizable. In addition, the $N$ eigenvalues satisfy $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{N-1}$, as long as $\tilde{L}$ is positive semidefinite and there is an orthogonal matrix $Q$ such that $Q^T \tilde{L} Q = \text{diag} (0, \lambda_1, \ldots, \lambda_{N-1})$. Then, $\tilde{L}$ is transformed into the normalized covariance matrix $\bar{M} = (\bar{M}_{ij})$ by $\bar{M} = Q \text{diag} (0, \lambda_1^{-1}, \ldots, \lambda_{N-1}^{-1}) Q^T$. (iii) $\bar{M}$ is converted into the normalized variance matrix $\bar{S}^2 = (\bar{S}_{ij}^2) = \left( \frac{M_{ii} + M_{jj} - 2M_{ij}}{3} \right)$. (iv) $\bar{S}^2$ is transformed into the contact matrix $C = (1 + \bar{S}^2)^{-3/2}$. Here, we briefly express this procedure as $C = \varphi (\bar{K})$. If we can find the inverse function $\varphi^{-1}$, we can solve the inverse problem $\bar{K} = \varphi^{-1}(C)$ for any input contact matrix.

The first step of the inverse function is $\bar{S}^2 = (C^{-2/3} - 1)$. As we previously noted, there is an inverse transformation from $\bar{S}^2$ to $\bar{M}$ because the matrix $\bar{M}$ satisfies the condition $\bar{M} 1 = 0$, where $1$ is a vector $(1, 1, \ldots, 1)^T$. Here, we explicitly provide the inverse transformation. Let $m$ be a vector that consists of the diagonal elements of the matrix $\bar{M}$, $(\bar{M}_{00}, \bar{M}_{11}, \ldots, \bar{M}_{N-1,N-1})^T$, and we define a square matrix $B$ as $B = (m, m, \ldots, m)$. Then, the relationship between $\bar{S}^2$ and $\bar{M}$ can be written as $3\bar{S}^2 = B + B^T - 2\bar{M}$. Multiplying $1$ from the right, we obtain $3\bar{S}^2 1 = A m$, where $A$ is a square matrix defined by $N I + 11^T$. Since the matrix $A$ is non-singular, there exists the inverse matrix given by $A^{-1} = \frac{2N I - 11^T}{2N^2}$. Therefore, we can determine the diagonal elements of the matrix $\bar{M}$ by $m = 3A^{-1} \bar{S}^2 1$ and calculate all elements by $\bar{M}_{ij} = \frac{M_{ii} + M_{jj} - 3\bar{S}_{ij}^2}{2}$.

Next, we considered a transformation from $\bar{M}$ to $\tilde{L}$. Theoretically, the matrix $\bar{M}$ should be positive semidefinite according to the definition. However, $\bar{M}$ calculated through the above transformation did not necessarily satisfy the positive semidefiniteness requirement, which is mainly due to the experimental noise of Hi-C data. Therefore, the Laplacian matrix $L$ could not be found in a straightforward manner by solving the eigenvalue problem. Instead, a different strategy was used to construct a pseudo-Laplacian matrix $\tilde{L}$ using the Moore-Penrose (MP) inverse matrix $\tilde{M}^+$ of the matrix $\bar{M}$. Matrices $\tilde{M}$ and $\tilde{L}$ theoretically satisfy the relationship $\tilde{M} \tilde{L} = I - \frac{11^T}{N}$; therefore, the MP inverse matrix minimizes the Frobenius norm $\| \tilde{M} \tilde{L} - I + \frac{11^T}{N} \|_F$ and provides the solution by $\tilde{L} = \tilde{M}^+ \left( I - \frac{11^T}{N} \right)$.

The construction of the MP inverse matrix is based on the singular value decomposition of the matrix $\bar{M}$, and we can easily implement it on numerical scripts. Finally, we can obtain a pseudo-interaction matrix $\bar{K}$ by $\bar{K} = -\tilde{L}$ and zero-diagonals ($\bar{K}_{ii} = 0$).

Based on the above algorithm, we reconstructed a contact matrix $\tilde{C} = \varphi (\bar{K})$ from an input matrix $C$ (Supplementary Figure S1A). Note that although the pseudo-Laplacian matrix $\tilde{L}$ is no longer positive semidefinite (Supplementary Figure S1B), the numerical calculation of the eigenvalues and eigenvectors proceeds normally. We estimated the closeness of all elements between $\tilde{C}$ and $\tilde{C}$. Surprisingly, the allclose function in NumPy returned True, which means that the two matrices are element-wise equal within the default numerical tolerance. Taken together, we were able to find the inverse transformation $\varphi^{-1}$. 

2/12
Supplementary Figure S1. Matrix transformations in PHi-C polymer modeling. (A) Inverse transformation $\varphi^{-1}$ converts an input normalized contact matrix $C$ [chr8: 42,100–44,525 kb (25-kb bins) for mouse embryonic stem cells (mESCs)] into the pseudo-interaction matrix $K$ via matrices $\tilde{\Sigma}^2$, $\tilde{M}$ and $\tilde{L}$. The reconstructed contact matrix $\tilde{C} = \varphi(K)$ and the input contact matrix $C$ are equal within the default numerical tolerance. (B) Eigenvalues of the pseudo-Laplacian matrix $\tilde{L}$. Since there are negative eigenvalues, $\tilde{L}$ is no longer positive semidefinite.
**Supplementary Note S2: Updated optimization operation**

**Mathematical concept of the PHi-C optimization**

The optimization procedure is a central part of PHi-C. We define the optimization process as an iterative finding of the normalized interaction matrix $\tilde{K}$ that decreases a cost function $f(C, \varphi(\tilde{K}))$ for an input contact matrix $C$. To clarify the mathematical concept, we introduce the following sets (Supplementary Figure S2):

- $\mathbb{K}^N = \{\tilde{K} | \text{a symmetric } N \times N \text{ matrix with zero diagonals } \tilde{K}_{ii} = 0\}$,
- $U_\epsilon = \{\tilde{K} \in \mathbb{K}^N | f(C, \varphi(\tilde{K})) < \epsilon\}$,
- $A = \{\tilde{K} \in \mathbb{K}^N | \text{the induced Laplacian matrix } \tilde{L} \text{ is positive-semidefinite and } \tilde{K}_{i,i+1} > 0\}$.

For a small $\epsilon$, a matrix $\tilde{K} \in U_\epsilon$ is an optimal candidate. In Supplementary Note 1, we found the minimal solution $\varphi^{-1}(C) = \tilde{K} \in U_{\epsilon=0}$ by the inverse transformation. However, since the induced Laplacian matrix does not necessarily satisfy the positive semidefiniteness, the polymer system becomes unstable and the normalized interaction matrix $\tilde{K}$ is unrealistic. Thus, the set $A$ ensures the physically acceptable matrix $\tilde{K}$ with connections along the polymer backbone $\tilde{K}_{i,i+1} > 0$ as a stable polymer system.

Let $\tilde{K}_n$ be the normalized interaction matrix at the $n$-th iteration step in the optimization procedure. An initial matrix $\tilde{K}_0$ should be in the set $A$, and an optimization path in $\mathbb{K}^N$ would go pass into set $U_\epsilon$ for a small $\epsilon$ (Supplementary Figure S2). Then, paths that output an optimal $\tilde{K} \in U_\epsilon \cap A$ are acceptable, while paths that pass into $U_\epsilon \cap A^c$ should be forbidden.

**Supplementary Figure S2. Mathematical concept of the PHi-C optimization.** The PHi-C optimization iteratively decreases the cost function in $\mathbb{K}^N$. The set $U_\epsilon$ has optimal candidates of the matrix $\tilde{K}$. As the inverse transformation induces the pseudo-interaction matrix $\hat{K}$, the matrix $\hat{K}$ is just an element of the set $U_{\epsilon=0}$. Optimization paths can pass into the set $U_\epsilon \cap A^c$, where the polymer systems are physically unrealistic. Optimization paths should move in the physically acceptable set $A$.

**Updated algorithm**

We mainly updated the optimization algorithm regarding the definition of the cost function and the iterative operation at each step (Supplementary Table S1). Accordingly, both factors provided faster and more accurate optimization results than the previous version.

First, we re-defined the cost function as

$$f(C, \varphi(\tilde{K})) = \sqrt{\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[C_{ij} - (\varphi(\tilde{K}))_{ij}\right]^2}.$$
PHi-C1 PHi-C2

Cost function: \( f(C, \varphi(\vec{K})) \)

\[
\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[ \log_{10} C_{ij} - \log_{10}(\varphi(\vec{K}))_{ij} \right]^2
\]

\[
\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[ C_{ij} - (\varphi(\vec{K}))_{ij} \right]^2
\]

For a randomly selected pair \((i, j)\),
\( \xi = \eta \times \text{random.uniform}(0, 1) \)

if \( \log_{10}(\varphi(\vec{K}_{n-1}))_{ij} > \log_{10} C_{ij} \) then
\( (\vec{K}_n)_{ij} = (\vec{K}_{n-1})_{ij} - \xi \)
\( (\vec{K}_n)_{ji} = (\vec{K}_{n-1})_{ji} - \xi \)

else
\( (\vec{K}_n)_{ij} = (\vec{K}_{n-1})_{ij} + \xi \)
\( (\vec{K}_n)_{ji} = (\vec{K}_{n-1})_{ji} + \xi \)

Optimization operation at every iteration step

For all the elements,
\( \vec{K}_n = \vec{K}_{n-1} - \eta(\varphi(\vec{K}_{n-1}) - C) \)

Supplementary Table S1. Mainly updated algorithms of PHi-C2 compared to PHi-C1. \( \eta \) is the learning rate parameter in the optimization operation.

This value represents the standard deviation between the contact probabilities \( C \) and \( \varphi(\vec{K}) \) per element. Moreover, we calculated the standard deviation between the logarithmic contact probabilities in the previous version (Supplementary Table S1). Since the logarithmic function is not defined for a zero value, an interpolate operation is needed for the zero-valued elements to ensure that the shape of the contact probability curve remains unaltered\(^1\). Therefore, the newly defined cost function does not require the interpolation process.

Every iterative optimization step updates all the elements of the matrix \( \vec{K}_n \) and obeys the following rule:

\[ \vec{K}_n = \vec{K}_{n-1} - \eta(\varphi(\vec{K}_{n-1}) - C) \]

where \( \eta \) is the learning rate parameter. In the previous version of PHi-C, the random choice of a matrix element \((i, j)\) and its slight change of the element value at every iteration step were a bottleneck (Supplementary Table S1). The new iterative rule overcomes the bottleneck by updating the values of all the elements at one time.

Finally, we set a stop condition due to a monotonical decrease of the cost function,

\[
0 < f(C, \varphi(\vec{K}_{n-1})) - f(C, \varphi(\vec{K}_n)) < \delta \implies \text{output } \vec{K}_n \text{ as an optimal solution } \vec{K}_{\text{opt}},
\]

where \( \delta = \eta \alpha \) and \( \alpha \) is a parameter for the stop condition.

Overall, the above three points are significant updates of the PHi-C2 algorithm. Thus far, we have not been able to mathematically ensure that every optimization path decreases the cost function monotonically in the set \( A \) (Supplementary Figure S2). The update coefficient \( (\varphi(\vec{K}_{n-1}) - C) \) does not exactly correspond to the gradient \( \frac{\partial f(C, \varphi(\vec{K}))}{\partial \vec{K}_{ij}} \bigg|_{\vec{K} = \vec{K}_{n-1}} \), although the gradient-based algorithm has already been formulated\(^4\) and shows a good performance\(^3\). However, we can operationally obtain an optimal solution in the set \( A \) by tuning the initial point \( \vec{K}_0 \in A \), learning rate \( \eta \), and stop condition parameter \( \alpha \).
Supplementary Note S3: Benchmarks

After the optimization process, we can estimate the closeness of the optimal contact matrix \( C_{\text{opt}} = \varphi(\bar{K}_{\text{opt}}) \) to the input contact matrix \( C \) using the Pearson correlation coefficient (PCC), \( r \), and distance-corrected PCC, \( r' \), for the non-diagonal elements of both matrices. Here, we show benchmarks for Hi-C data of mESCs\(^3\) [chr1: 50–60 Mb (25-kb bins)].

For the 400 × 400-sized input matrix of mESCs and a default set of optimization parameters (\((\bar{K}_0)_{i,i+1} = 0.5, \eta = 10^{-4}, \alpha = 10^{-4}\)), PHi-C2 output an optimal solution with \( r = 0.998 \) and \( r' = 0.970 \) in 26,307 iteration steps (Supplementary Figure S3A and Supplementary Table S2). The differences between the PHi-C1 and PHi-C2 optimization algorithms are summarized in Supplementary Table S1, and they indicate that the calculation of the cost function \( f(C, \varphi(\bar{K})_n) \) for all matrix elements in PHi-C1 seems relatively inefficient for a slight change of a randomly selected matrix element at each iterative step. In contrast, all matrix elements are updated together using the feature of NumPy for matrix calculations in PHi-C2. Therefore, the optimization process should be more efficient. As expected, for the same iteration steps and \( \eta = 4 \times 10^{-3} \), an optimal solution of the previous PHi-C (PHi-C1) had a value of \( r = 0.966 \) (Supplementary Figure S3B), although the meaning of the cost function and the learning rate differs between PHi-C1 and PHi-C2. We can confirm that the updated algorithm improves the speed and accuracy of the optimization procedure compared to that of PHi-C1.

Next, we show benchmarks for different optimization parameters: initial values of the polymer backbone \((\bar{K}_0)_{i,i+1}\), learning rate \( \eta \), and stop condition parameter \( \alpha \) (Supplementary Table S2). We carried out all calculations using an Intel® Xeon® Gold 6154 processor (24.75 M Cache, 3.00 GHz) with Intel® distribution for the Python environment and Google Colab environment. The procedure stopped within less than 100,000 iteration steps and 30 min and 90 min for these parameters in our workstation and the Google Colab environment, respectively. Interestingly, although the optimal cost function values differed, the correlation values reached 0.998. As our stop condition relates to the cost function curve saturation, the optimal solutions would be close to local minimums in the set \( U_{\epsilon=0.00223} \). Also, we could confirm that the optimal solutions are located in the set \( A \) due to the positive semidefiniteness (Supplementary Figure S3C).

The speed of the PHi-C2 optimization depends on the matrix calculations at every iterative step. To clarify the speed and accuracy features, we benchmarked the PHi-C2 optimization by changing the different bin sizes from 25 kb to 50, 100, and 250 kb for the same genomic region [chr1: 50–60 Mb] (Supplementary Table S3). The optimized accuracy and number of iterative steps differed due to the different input Hi-C matrices. At the same time, the calculation time dramatically decreased depending on the reduction of the matrix size. A rough estimation of the ratio \((9m54s)/(2m41s) = 3.69 \cdots \approx 4\) for 25- and 50-kb bin sizes implies that the optimization time is in \( O(N^2) \), where \( N \) is the number of the diagonal bins or beads of our polymer model.

As we stated, the update algorithm does not always decrease the cost function monotonically. We observed such counter-examples for Hi-C data of the mouse erythroid cells\(^7\) [chr1: 3–195.5 Mb (500-kb bins)] in the prometaphase. Here, we altered only the initial values of the polymer backbone \((\bar{K}_0)_{i,i+1} = 0.1 \sim 1.0 \) (fixed \( \eta = 10^{-4} \) and \( \alpha = 10^{-4} \)). Some curves showed an increase in the cost function within the initial 100 steps (Supplementary Figure S4A). Nevertheless, every curve eventually converged and output an optimal solution. Moreover, for Hi-C data in the late G1 phase, we confirmed a monotonical decrease of the cost function (Supplementary Figure S4B).
**Supplementary Figure S3.** Comparison between (A) PHi-C2 and (B) PHi-C1. We used Hi-C data of mESCs [chr1: 50–60 Mb (25-kb bins)] as input. (Left) Heatmap of the contact matrix consists of the input (upper-right) and the optimal (lower-left) matrices. (Middle) Scatter plot with the PCC $r$ visualizes the closeness of the input and optimized contact matrices. (Right) Curve of the cost function in the optimization steps. (C) Eigenvalues of the induced Laplacian matrix from the optimal solution are positive except for $\lambda_0 = 0$. We plotted eigenvalue spectra for the optimal solutions in Supplementary Table S2.
| \( (\bar{K}_0)_{i,i+1} \) | \( \eta \) | \( \alpha \) | # iterations | time | time (Colab) | \( f(C, \varphi(\bar{K}_{opt})) \times 10^{-3} \) | \( r \) | \( r' \) |
|---|---|---|---|---|---|---|---|---|
| 0.1 | \( 10^{-4} \) | \( 10^{-4} \) | 68,164 | 24 m 27 s | 88 m 13 s | 2.10086 | 0.998 | 0.982 |
| 0.2 | \( 10^{-4} \) | \( 10^{-4} \) | 57,477 | 20 m 52 s | 72 m 51 s | 2.09129 | 0.998 | 0.980 |
| 0.3 | \( 10^{-4} \) | \( 10^{-4} \) | 36,388 | 13 m 21 s | 47 m 15 s | 2.11411 | 0.998 | 0.976 |
| 0.4 | \( 10^{-4} \) | \( 10^{-4} \) | 17,931 | 7 m 00 s | 20 m 56 s | 1.94762 | 0.997 | 0.966 |
| 0.5 | \( 10^{-4} \) | \( 10^{-4} \) | 26,307 | 9 m 54 s | 32 m 7 s | 1.73502 | 0.998 | 0.970 |
| 0.6 | \( 10^{-4} \) | \( 10^{-4} \) | 35,097 | 13 m 48 s | 43 m 46 s | 1.62454 | 0.998 | 0.972 |
| 0.7 | \( 10^{-4} \) | \( 10^{-4} \) | 43,714 | 16 m 51 s | 54 m 46 s | 1.56127 | 0.998 | 0.973 |
| 0.8 | \( 10^{-4} \) | \( 10^{-4} \) | 52,089 | 19 m 54 s | 64 m 39 s | 1.52338 | 0.998 | 0.974 |
| 0.9 | \( 10^{-4} \) | \( 10^{-4} \) | 60,228 | 20 m 48 s | 77 m 54 s | 1.50057 | 0.998 | 0.975 |
| 1.0 | \( 10^{-4} \) | \( 10^{-4} \) | 68,139 | 24 m 34 s | 75 m 0 s | 1.48751 | 0.998 | 0.976 |

Supplementary Table S2. Benchmarks of PHI-C2 for different optimization parameters. We used Hi-C data of mESCs [chr1: 50–60 Mb (25-kb bins)].

| Bin size | Matrix size | # iterations | time | \( f(C, \varphi(\bar{K}_{opt})) \times 10^{-3} \) | \( r \) | \( r' \) |
|---|---|---|---|---|---|---|
| 25 kb | 400×400 | 26307 | 9 m 54 s | 1.73502 | 0.998 | 0.970 |
| 50 kb | 200×200 | 41978 | 2m 41 s | 1.57381 | 0.999 | 0.988 |
| 100 kb | 100×100 | 63261 | 60 s | 1.93041 | 0.999 | 0.991 |
| 250 kb | 40×40 | 79840 | 18 s | 3.80607 | 0.998 | 0.982 |

Supplementary Table S3. Benchmarks of PHI-C2 for different matrix sizes. We used Hi-C data of mESCs [chr1: 50–60 Mb] and set default initial optimization parameters ((\( \bar{K}_0 \)\( _{i,i+1} = 0.5, \eta = 10^{-4}, \alpha = 10^{-4} \)).
**Supplementary Figure S4.** PHi-C2 results for the (A) prometaphase and (B) late G1 phase. We used Hi-C data of mouse erythroid cells [chr1: 3–195.5 Mb (500-kb bins)] as the input. (Left) Curves of the cost function. (Right) Heatmap of the contact matrix that consists of the input (upper-right) and optimal (lower-left) matrices. (A) (Inset) Magnification of the initial 100 iteration steps. The black points represent the local maximum point for \((K_0)_{i,i+1} = 0.5 \sim 1.0\). The heatmap consists of the input contact matrix (upper-right) and the optimal matrix (lower-left) for \((K_0)_{i,i+1} = 0.2\). (B) The heatmap consists of the input contact matrix (upper-right) and the optimal matrix (lower-left) for \((K_0)_{i,i+1} = 0.7\).
Supplementary Note S4: Rheology analysis as an additional function

We added rheology analysis commands as a new function in PHi-C2. This new feature provides a method of interpreting the spatial hierarchy of the 3D genome organization as dynamic 3D genome information\(^8\). For example, we could conclude that TAD boundaries behave as more rigid nodes than intra-TAD regions. Using the algorithm, we can convert the optimal solution \(\hat{K}_{\text{opt}}\) into the rheology spectra of the normalized complex compliance \(\tilde{J}^\ast\), the normalized complex modulus \(\tilde{G}^\ast\), and the loss tangent \(\tan\delta\).

In soft matter physics, the complex compliance \(J'(\omega)\) for periodic mechanical stress with the angular frequency \(\omega\) is the ratio of strain to stress\(^9\). In addition, the complex compliance possesses real and imaginary components: \(J'(\omega) = \left|J'(\omega)e^{-i\delta(\omega)} = J'(\omega) - iJ''(\omega)\right|\). The real component is the storage compliance related to the elastic response, while the imaginary component is the loss compliance as the viscous response. Therefore, the ratio \(J''(\omega)/J'(\omega)\) corresponds to \(\tan\delta(\omega)\) and is a measure of liquid-like (\(\tan\delta(\omega) > 1\)) or solid-like (\(\tan\delta(\omega) < 1\)) behavior. Furthermore, the complex modulus \(G^*(\omega) = |G^*(\omega)\rangle e^{i\delta(\omega)} = G'(\omega) + iG''(\omega)\) relates to the complex compliance by \(G^*(\omega) = 1/J'(\omega)\).

Here, we outline the calculation of the complex compliance\(^8\). Note that in PHi-C1 and PHi-C2, the spatial and temporal scales of all physical variables are normalized by combining the contact distance \(\sigma\), friction coefficient of the beads \(\gamma\), Boltzmann constant \(k_B\), and temperature \(T\). First, we diagonalize the normalized Laplacian matrix \(\hat{L}\) induced from the normalized optimal solution \(\hat{K}_{\text{opt}}\) to obtain the orthogonal matrix \(Q\) and the normalized eigenvalues \(0 < \tilde{\lambda}_0 < \tilde{\lambda}_1 < \cdots < \tilde{\lambda}_{N-1}\) that satisfy \(Q^T\hat{L}Q = \text{diag}(\tilde{\lambda}_0, \tilde{\lambda}_1, \cdots, \tilde{\lambda}_{N-1})\). Then, we derive the normalized complex compliance by

\[
\tilde{J}^\ast(\tilde{\omega}; n) = \sum_{p=1}^{N-1} \frac{Q_{np}^2}{3\tilde{\lambda}_p + i\tilde{\omega}}.
\]

Therefore, the normalized storage and loss compliances can be given as

\[
\tilde{J}'(\tilde{\omega}; n) = \sum_{p=1}^{N-1} \frac{3Q_{np}^2\tilde{\lambda}_p}{9\tilde{\lambda}_p^2 + \tilde{\omega}^2}
\quad \text{and} \quad
\tilde{J}''(\tilde{\omega}; n) = \sum_{p=1}^{N-1} \frac{Q_{np}^2\tilde{\omega}}{9\tilde{\lambda}_p^2 + \tilde{\omega}^2},
\]

respectively. Since \(n\) represents the index of the beads or bins (\(n = 0, 1, \cdots, N-1\)) along the genomic coordinate, we can plot a two-dimensional heatmap as the rheology spectrum of \(|\tilde{J}^\ast(\tilde{\omega}; n)|\) and arrange it under the input Hi-C pattern map. The normalized complex modulus \(\hat{G}^*(\tilde{\omega}; n)\) and the loss tangent \(\tan\delta(\tilde{\omega}; n)\) can be calculated from the following definitions: \(\hat{G}^*(\tilde{\omega}) = 1/\tilde{J}^\ast(\tilde{\omega})\) and 
\(\tan\delta(\tilde{\omega}) = \tilde{J}'(\tilde{\omega})/\tilde{J}^\ast(\tilde{\omega})\).

Here, we show examples for the Hi-C data of mESCs\(^3\) chr17: 3–95 Mb (250-kb bins), chr17: 41–59.4 Mb (50-kb bins), and chr17: 49–52.68 Mb (10-kb bins) (Supplementary Figure S5). We fixed the matrix size to 368 × 368.
Supplementary Figure S5. Rheology analysis. We used Hi-C data of mESCs [(left) chr17: 3–5 Mb (250-kb bins),
(middle) chr17: 41–59.4 Mb (50-kb bins), (right) chr17: 49–52.68 Mb (10-kb bins)]. (A) Heatmaps of the contact
matrix that consist of the input (upper-right) and the optimal (lower-left) matrices. (B) Rheology spectra of the
normalized complex compliance $|\bar{J}(\bar{\omega})|$ as a measure of the dynamic flexibility of three cases. (C) Rheology spectra
of the normalized complex modulus $|\bar{G}(\bar{\omega})|$ as a measure of the dynamic rigidity of three cases. (D) Rheology
spectra of the loss tangent $\tan\delta(\bar{\omega})$ as a measure of the liquid-like (> 1) and solid-like (< 1) states of three cases.
References

1. Shinkai, S. et al. PHi-C: deciphering Hi-C data into polymer dynamics. *NAR Genom. Bioinform.* **2**, lqaa020, DOI: https://doi.org/10.1093/nargab/lqaa020 (2020).

2. Rao, C. R. & Mitra, S. K. *Generalized Inverse of Matrices and its Applications* (Wiley, New York, 1971).

3. Bonev, B. et al. Multiscale 3D genome rewiring during mouse neural development. *Cell* **171**, 557–572.e24, DOI: https://doi.org/10.1016/j.cell.2017.09.043 (2017).

4. Le Treut, G., Képès, F. & Orland, H. A polymer model for the quantitative reconstruction of chromosome architecture from HiC and GAM data. *Biophys. J.* **115**, 2286–2294, DOI: https://doi.org/10.1016/j.bpj.2018.10.032 (2018).

5. Liu, L., Zhang, B. & Hyeon, C. Extracting multi-way chromatin contacts from Hi-C data. *PLoS Comput. Biol.* **17**, e1009669, DOI: https://doi.org/10.1371/journal.pcbi.1009669 (2021).

6. Bianco, S. et al. Polymer physics predicts the effects of structural variants on chromatin architecture. *Nat. Genet.* **50**, 662–667, DOI: https://doi.org/10.1038/s41588-018-0098-8 (2018).

7. Zhang, H. et al. Chromatin structure dynamics during the mitosis-to-G1 phase transition. *Nature* **576**, 158–162, DOI: https://doi.org/10.1038/s41586-019-1778-y (2019).

8. Shinkai, S., Sugawara, T., Miura, H., Hiratani, I. & Onami, S. Microrheology for Hi-C data reveals the spectrum of the dynamic 3D genome organization. *Biophys. J.* **118**, 2220–2228, DOI: https://doi.org/10.1016/j.bpj.2020.02.020 (2020).

9. Larson, R. G. *The Structure and Rheology of Complex Fluids* (Oxford University Press, 1999).