Computational 3D genome modeling using Chrom3D

Jonas Paulsen[1], Tharvesh Moideen Liyakat Ali[1] & Philippe Collas[1,2]

[1]Department of Molecular Medicine, Institute of Basic Medical Sciences, Faculty of Medicine, University of Oslo, Oslo, Norway. [2]Department of Immunology, Oslo University Hospital, Oslo, Norway. Correspondence should be addressed to J.P. (jonas.paulsen@medisin.uio.no) or P.C. (philc@medisin.uio.no).

Published online 26 April 2018; doi:10.1038/nprot.2018.009

Chrom3D is a computational platform for 3D genome modeling that simulates the spatial positioning of chromosome domains relative to each other and relative to the nuclear periphery. In Chrom3D, chromosomes are modeled as chains of contiguous beads, in which each bead represents a genomic domain. In this protocol, a bead represents a topologically associated domain (TAD) mapped from ensemble Hi-C data. Chrom3D takes as input data significant pairwise TAD–TAD interactions determined from a Hi-C contact matrix, and TAD interactions with the nuclear periphery, determined by ChIP-sequencing of nuclear lamins to define lamina-associated domains (LADs). Chrom3D is based on Monte Carlo simulations initiated from a starting random bead configuration. During the optimization process, TAD–TAD interactions constrain bead positions relative to each other, whereas LAD information constrains the corresponding bead toward the nuclear periphery. Optimization can be repeated many times to generate an ensemble of 3D genome models. Analyses of the models enable estimations of the radial positioning of genomic sites in the nucleus across cells in a population. Chrom3D provides opportunities to reveal spatial relationships between TADs and LADs. More generally, predictions from Chrom3D models can be experimentally tested in the laboratory. We describe the entire Chrom3D protocol for modeling a 3D diploid human genome, from the creation of input files to the final rendering of 3D genome structures. The procedure takes ~18 h. Chrom3D is freely available on GitHub.

INTRODUCTION

Analysis of spatial organization of the mammalian genome

DNA sequencing–based methodologies have unveiled several important principles of 3D folding of the genome in the eukaryotic cell nucleus. Coupling high-throughput sequencing to chromosome conformation capture techniques such as Hi-C makes it possible to identify regions of contacts within chromosomes and between chromosomes in entire genomes[1]. Chromosome conformation capture methods entail (i) a cross-linking of DNA molecules that are in close spatial proximity, (ii) fragmentation of chromatin, (iii) ligation of the cross-linked fragments and (iv) identification of the interacting DNA sequences. In Hi-C, the ligated DNA fragments are purified and sequenced[2]. Outputs from Hi-C experiments are genome-wide matrices of pairwise intra and interchromosomal contact frequencies from which spatial representations of the genome can be estimated. In addition, techniques such as Dam-methylase identification (DamID, a proximity-labeling method used to map binding sites of DNA- and chromatin-binding proteins)[3,4] or chromatin immunoprecipitation (ChIP) of nuclear lamins coupled to high-throughput sequencing (ChIP-seq)[5–9] have been used to identify genomic regions in contact with the nuclear periphery.

From these analyses, a hierarchical view of spatial genome organization emerges[1]. At low resolution, the genome appears to be broadly divided into multimegabase transcriptionally active or ‘open’ A compartments and repressive or ‘closed’ B compartments. Within compartments, smaller TADs of ~0.5–1 Mb define regions with a high frequency of chromosomal contacts, whereas much less frequent contacts occur between adjacent TADs[10,11]. Along the linear genome, TAD positions are conserved between cell types, suggesting that they constitute important genome units. At the nuclear periphery, chromosomes are anchored to the nuclear lamina through 0.1- to 100-kb LADs[3,5]. The structural stability of genomic domains such as TADs remains unclear. Although the borders between TADs are stable during the cell cycle, their intensity and strength have been reported to change[12]. It has also been shown that insulation factors such as CCCTC-binding factor (CTCF) and cohesin bind to TAD boundaries in a dynamic manner, suggesting that TADs arise from more transient processes[13].

Computational methods to infer 3D structural representations of the genome in the nuclear space can reveal spatial relationships between genomic regions that are not directly visible in the underlying data. Physics-based modeling techniques infer genome-folding principles based on physical properties of chromatin[1]. For instance, polymer simulations have provided useful information on the folding state of individual chromosomes based on physical constraints driving polymer compaction[14–19], on the formation of chromosome territories[16] (e.g., from physical forces that prevent polymer mixing[15]) and on chromosomal interactions[20,21]. Other genome-modeling approaches, such as that described here, take into account pairwise interaction probabilities between genomic domains determined from, for example, Hi-C data, to infer a 3D positioning of these domains relative to each other[22–24]. Analysis of the properties of the models can provide insights into higher-order 3D genome conformation. Spatial investigations of the genome can also be carried out to generate working hypotheses.

Overview and development of the method

Chrom3D is a computational platform for 3D genome structural modeling that optimizes the positioning of genomic domains in the nuclear space, relative to each other and relative to the nuclear periphery[24]. In Chrom3D, chromosomes are modeled as chains of contiguous beads, in which each bead represents a genomic domain. In this protocol, a bead represents a TAD, defined by Hi-C data, and bead diameter is determined by the genomic size of the TAD it represents. Chrom3D takes as input data (i) statistically significant pairwise TAD–TAD interactions determined from an ensemble Hi-C contact matrix and (ii) significant interactions between chromosomes and the nuclear lamina, determined from ChIP-seq analysis of nuclear lamins (in this protocol, lamin B1). The nuclear boundary is modeled as a sphere with a radius of 5 μm.

In the modeling process, the starting spatial position of the beads is random, and Monte Carlo optimization is used to generate a modeled genome structure. The process can be repeated tens, hundreds or thousands of times, each starting with a different random bead configuration, to generate an ensemble of 3D structural models of the genome. The models can be analyzed to, for instance, measure distance distributions between loci, and predict the variability in the 3D location of a bead (here, a TAD) between cells in a population. Importantly, structural 3D models provide, beyond mere Hi-C data, a way of incorporating information on the size, scale, and distance relations of the modeled chromosomal regions relative to each other and to the nucleus boundary. This is also important for comparing the 3D models with, for example, fluorescence *in situ* hybridization (FISH) data, as such comparisons can be done more directly, rather than comparing contact counts with observed distances.

With integrated 3D modeling of Hi-C and lamin ChIP-seq data, Chrom3D provides opportunities to reveal the spatial relationship between TADs and LADs. The main benefit of constructing integrated 3D genome models, as compared with more conventional single–data set analyses, is the ability to gain insight into restraint violations and how these relate to the conflicting inputs into the 3D conformation. The optimized structure then provides a resolved conformation, taking such conflicting information into account. Thus, by incorporating multiple data sources, Chrom3D can generate distinct states of genome conformation from potentially antagonistic data, which are present in a multicell population average. More generally, Chrom3D also enables investigations of the mechanisms of gene regulation in diseases susceptible to affect the spatial organization of chromatin[24,25]. Predictions arising from the analysis of Chrom3D models can be experimentally tested in the laboratory.

We describe a protocol for 3D genome modeling, from the creation of the required and optional input files to the final rendering of 3D genome structural models (**Fig. 1**). The protocol assumes that the user has at hand (i) the genomic coordinates of mapped TADs (or other chromatin domains of interest), (ii) a Hi-C contact matrix and (iii) the genomic coordinates of LADs (hereafter, 'LAD information'). In the modeling process, statistically significant pairwise TAD–TAD interactions, i.e., TAD–TAD interactions occurring more frequently than what would be expected by chance, define the positioning of the beads relative to each other. In addition, identification of a LAD associated with a TAD (from the lamin B1 ChIP-seq data) imposes a constraint on the corresponding bead toward the nuclear periphery. Note that although 3D genome modeling from single-cell Hi-C data is technically feasible using Chrom3D, the protocol was not initially designed for single-cell Hi-C data 3D modeling. The protocol described here therefore focuses on 3D genome modeling from ensemble Hi-C data.

The PROCEDURE starts with summing Hi-C contact counts for all TAD pairs: this includes the creation of a BED file containing the genomic coordinates of the beads (Steps 1 and 2) and the summing of intrachromosomal TAD–TAD interaction counts from the Hi-C data (Steps 3–5). Significant interactions between TADs are then identified (Steps 6 and 7). A critical step is the compilation of the genomic positions of the TADs and their significant pairwise interactions into a Model Setup File (in GTtrack format) (Step 8). If no options are selected, Chrom3D can be launched (Step 15). Optionally, LAD information and/or interchromosomal interactions are added to the Model Setup File (Steps 9–13). Another option is to create a diploid Model Setup File (Step 14). If any of these options are chosen, Chrom3D is launched to generate 3D genome models (Step 15) that can be visualized and stored for downstream analyses (Steps 16–24).

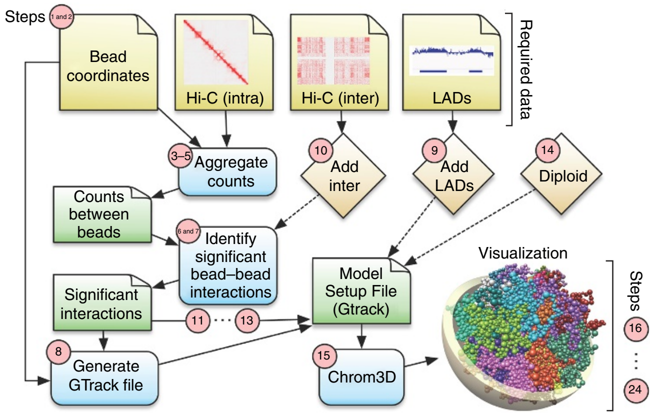

Figure 1 | Flowchart of the protocol.

Chrom3D is based on Monte Carlo optimization initiated from a starting random bead configuration. Mandatory constraints are imposed by bead–bead interactions within chromosomes (intrachromosomal interactions; **Fig. 2a**). Optional constraints can be imposed by interactions between beads on two different chromosomes (interchromosomal interactions) and by LAD information (**Fig. 2a**). For each iteration, one among several preset chromosome moves is randomly selected (**Fig. 2b**); this affects the position of one or more beads yet maintains connectivity between beads in the chain and keeps the rest of the chromosomes intact. The resulting Euclidian distances between interacting beads or between a bead and the nuclear periphery are evaluated through a loss-score function (Eq. 4) using the Metropolis–Hastings algorithm until a minimum of the loss-score is reached (**Fig. 2c**). An optimized structure can then be rendered (**Fig. 2c**).

Chrom3D can be applied in a variety of settings, including as a tool integrated as part of a more general research approach. Chrom3D can be used to visualize whole-genome structures (**Fig. 3a**,**b**), specific genome regions (TADs, LADs or other domains) or epigenetic modifications (**Fig. 3c–e**). Chrom3D is also useful for generating predictions researchers can subsequently test in wet-lab experiments. We have shown that optimized 3D genome models respect the concept of chromosome territories[24] initially elaborated from microscopy studies[26]. After generating hundreds of models from the same Hi-C dataset[27], Chrom3D predictions of TAD positioning were validated by FISH[24], an imaging technique in which specific loci are detected using fluorescently labeled cDNA probes. Similarly, by analyzing the frequency of radial position of TADs across hundreds of models, and by comparing the predicted Chrom3D radial positioning of TADs with LAD information in single cells in a separate study[28], Chrom3D can recapitulate global genome conformation in single cells. As such, Chrom3D is able to predict the stability

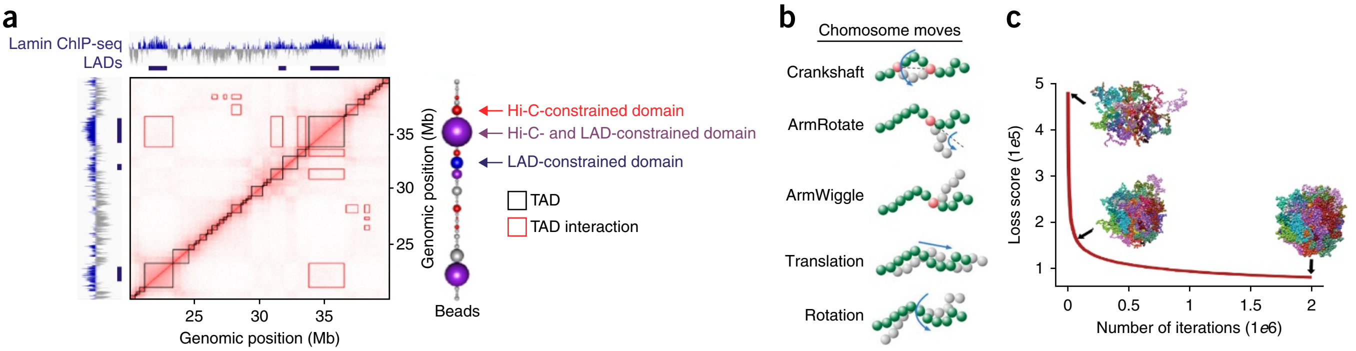

Figure 2 | Principles of Chrom3D. (**a**) Chromosomes are represented as a chain of connected beads, each bead being of the size of a genomic domain. A domain can be a TAD, as in this example. Chrom3D generates 3D models from the combination of Hi-C data (Hi-C matrix) and lamin ChIP-seq data (LADs) for a given cell type. An example of lamin ChIP-seq profile and corresponding LADs is shown in blue on top and on the left of the Hi-C matrix. During the modeling, beads are constrained by the Hi-C data to interact pairwise between them (red beads), and/or are constrained by the LAD data to interact with the nuclear periphery (blue and purple beads). Beads not subjected to constraints are shown in gray. (**b**) For each iteration, beads are repositioned by a chromosome 'move' taking into account the constraints, in order to optimize the structure. (**c**) Loss-score values attained during one round of modeling (2 million iterations). Representative Chrom3D models are shown for IMR90 fibroblasts. **a**,**b** adapted from ref. 24, Creative Commons license (https://creativecommons.org/licenses/by/4.0/legalcode).

of a genomic domain at the nuclear periphery or in the nuclear center in a population of cells.

Nuclear lamins (A-type lamins, in particular) are found not only at the nuclear periphery but also in the nuclear interior, where they can also interact with chromatin[6,7,29,30]. Interestingly, single-cell DamID data show that only ~30% of LADs are placed in the periphery in a given single cell[31]. In the mouse, a subset of LADs also seem to associate with the nucleolus, in the nuclear interior[32]. Chrom3D can recapitulate the radial placement of LADs in single cells, including central, intermediate and peripheral LAD positions, by running multiple optimizations and analyzing the radial distribution of LADs in the resulting ensemble of models. This was made possible only by inclusion of both LAD and Hi-C information in the modeling. This combination of constraints provides complementary, although sometimes conflicting, information on the radial placement of genomic regions in the models.

Chrom3D has been applied to model the radial placement of TADs in fibroblasts from patients with a laminopathy-causing lamin A mutation[24]. The models reveal lamin A LADs specific to patient cells located toward the nuclear center, rather than at the nuclear periphery. This observation could not be anticipated from a mere analysis of Hi-C and lamin ChIP-seq data in a linear genome, for the very reason that LADs are a priori assumed to be at the nuclear periphery. RNA-sequencing data show concordant downregulation of gene expression in the patient-specific LADs, which functionally validates inferences from the Chrom3D models[24]. We have also used Chrom3D to establish and visualize the radial positioning of carcinogen-induced DNA lesions in the nucleus[33]. Chrom3D is also able to model local chromosome conformations in 3D, as shown for the α-globin locus in two different cell lines[24]. This was achieved without the need for LAD information, using in this instance only 5C chromosome conformation capture data at the α-globin locus[34]. Chrom3D can therefore be used in a range of applications to make inferences testable in a wet laboratory. As such, Chrom3D can constitute an integral component of a research investigation, inasmuch as a genome visualization endpoint.

Applications of the method

The protocol described here considers intra and interchromosomal interactions defined by Hi-C data, and anchoring of chromatin at the nuclear periphery defined by lamin ChIP-seq data. The flexible options of the Model Setup File (**Boxes 1–4**) make, in principle, any type of genome 3D modeling possible using the protocol. Moreover, **Box 5** describes an application of local (as opposed to genome-wide) modeling of a locus, here the ENCODE ENm008 region containing the α-globin locus. The protocol can also be used to model whole genomes without LAD information, model whole or parts of chromosomes, or model 3D genome structure in any cell type or species. **Box 6** describes the expected initial output from Chrom3D. Modeling of genomes based on other types of domain information, not necessarily from Hi-C

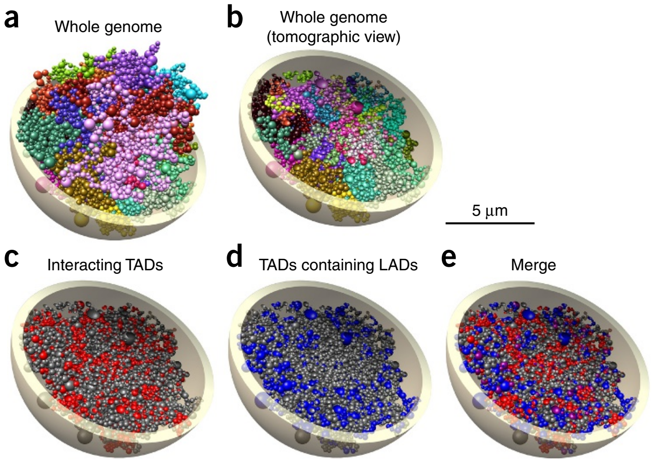

Figure 3 | 3D genome models of IMR90 fibroblasts produced by Chrom3D and visualized with Chimera. (**a**) A whole-genome model in which each chromosome is distinctively colored. (**b**) Tomographic view of whole-genome model shown in **a**. (**c**–**e**) Tomographic representations of the genome model shown in **a**, visualizing pairs of interacting beads (**c**, red), beads constrained to the nuclear periphery (**d**, blue), and both interacting and peripherally constrained beads (**e**, purple).

Box 1 | Model Setup File (GTrack)

The Model Setup File is the only input file in Chrom3D. This file provides information on the genomic coordinates of the beads, bead size, other bead characteristics and all constraints of the model, using the GTrack format[46]. This genomic annotation format allows a flexible definition of constraints and other attributes of the genomic segments (here, the beads).

A simple example is

```
###seqid    start     end      id    edges    periphery
chr1        0         1000     a     e        .
chr1        1000      2000     b     .        .
chr1        2000      3000     c     .        .
chr1        3000      4000     d     .        .
chr1        4000      5000     e     a        1
```

The header (starting with `###`) specifies the information in the columns below. The first column (`seqid`) indicates on which chromosome a given bead is defined and specifies which beads should be considered part of the same bead chain. The second and third columns (`start, end`) provide information on the genomic position of the bead. The first three columns are compatible with the standard BED file definition, and the Model Setup File can therefore be used as an input to, e.g., BedTools[55]. The fourth column (`id`) specifies a unique bead id. The first four columns are mandatory for all Model Setup Files in Chrom3D. The fifth column (`edges`) in the example gives information on contact pairwise (distance) constraints between beads. The term '`edges`' is reserved in the GTrack format and refers to any type of interconnection between genomic segments. Here, bead a is constrained as close as possible to bead e. Such pairwise constraints must be added in a symmetric manner, such that if bead a is constrained toward e, e is constrained toward a. A dot (.) refers to no constraint. Multiple pairwise constraints can be added in a semicolon-separated list:

```
chr1         0        1000      a    e;d
```

To constrain specific beads toward the nuclear periphery, a column named `periphery` can be added (see upper example). In this case, bead e is constrained toward the nuclear periphery, whereas all other beads (`.`) are not constrained. If instead a `0` is specified, beads are constrained toward the nuclear center:

```
chr1         4000     5000       e     a        0
```

data, is also possible because Chrom3D can, in principle, accommodate any type of domain data. Positional information other than LADs reflecting radial positioning of loci can be encoded in the Model Setup File (**Box 1**). For example, when LAD information is lacking, B compartments referred to earlier may be used as a proxy for attribution of beads to the nuclear periphery; however, this must be thoroughly investigated.

Related methods

Chrom3D is a recently developed genome 3D modeling platform able to combine Hi-C data and positional constraints to provide information on the radial positioning of loci in the 3D nucleus. Several computational frameworks able to model 3D genomes exist[1,35–37], but a description of these packages is beyond the scope of this protocol. In addition to Chrom3D, a few are able to incorporate locus positional constraints. Protocols based on the Integrated Modeling Platform (IMP)[38] can integrate any kind of spatial restraints and have been applied in various genomic settings[22,23,38,39]. IMP was, however, originally designed to model protein structure, and subsequently adapted to model genomes. IMP requires advanced programming skills to make use of its functionality. Nevertheless, a software package based on IMP (TADbit) now provides a more user-friendly interface[40]. TADbit also allows defining of spatial constraints in the modeled nucleus[40]. A protocol relying on yeast genomic and nuclear architectural data also enables integration of non-Hi-C data, such as nucleolus constraints and centromere position, into the models[41]. Very recent work further introduces a protocol to integrate specifically Hi-C and LAD data[42].

We have earlier specifically compared IMP with Chrom3D[24]. Unlike Chrom3D, IMP-based tools initialize individual beads in random positions and handle the issue of connectivity between beads in a chain (a chromosome) by specifying distance constraints between linearly consecutive beads. This can cause entangled genome models[24]. By contrast, Chrom3D advantageously maintains bead connectivity within chromosomes throughout the optimization, by initialization of a fully connected bead chain and by iteratively moving only parts of the bead chain without breaking chain connectivity (**Fig. 2b**).

Nevertheless, IMP-based methods[23,42,43] allow a more direct estimation of the cell-to-cell variability in 3D genome structures as compared with Chrom3D. Using deconvolution techniques, these methods estimate an ensemble of structures as part of a single simulation. By contrast, Chrom3D generates a single structure as a result of one simulation, and simulations are repeated multiple times to obtain statistical estimates of the positioning of the beads across a large number of structures. It is currently not clear how much extra information is gained from population-based modeling as compared with using statistical estimates from multiple independent simulations. Specifically, although both modeling strategies can recapitulate average properties of chromatin structures[23,24], further investigation is needed to determine whether the variation in these structures can also be reliably predicted.

Limitations

In Chrom3D, association of TADs with the nuclear periphery is established by constraining selected beads toward a nuclear periphery defined according to a prefixed radius (in this protocol, 5 µm). This assumes a spherical nuclear boundary, which is not the case for all cell types. This is notably an issue when cells are cultured *in vitro*, in which case the nucleus tends to be ellipsoid. Additional development is needed to accommodate other shapes delineating the nuclear boundary.

Experimental design

The PROCEDURE uses published Hi-C data for IMR90 diploid human fibroblasts[27] and lamin A ChIP-seq data for human fibroblasts[44] as example input data to model a single 3D genome structure. We recommend that users not familiar with 3D genome modeling familiarize themselves with Chrom3D by first using the example data provided in this protocol. Input Hi-C data are assumed to be provided in a coordinate (COO) format, which represents raw contact counts as a list of coordinates (row, column, contact count) for each entry in the upper triangle of the Hi-C matrix, excluding zeros[27]. LADs should be provided in the readily-available BED format. Further, we assume that the user has already-available contact domains from the Arrowhead algorithm[45], based on the same input Hi-C data. These contact domains are available for download for a range of cell types through Gene Expression Omnibus (GEO) GSE63525. For users using the protocol with their own Hi-C data, contact domains will expectedly come from their data sets. Alternatively, users can specify any partitioning of the genome in a BED format and start from Step 2.

Our original report on Chrom3D[24] relies on input data sets for HeLa cells[27]; however, reproducing HeLa genome models with the PROCEDURE described here requires raw Hi-C data, which can be obtained only after permission is granted by the HeLa Genome Data Access Working Group at the National Institutes of Health (Bethesda, MD, USA). We also highlight that, unless the objective of the modeling is to generate a single 3D genome model solely for visualization purposes, simulation should be run multiple times to achieve statistical estimates of the positioning of loci in the ensemble of models.

The PROCEDURE begins with the definition of the genomic coordinates of the beads, that is, their start and end positions in the linear genome. These coordinates are determined from the mapped TADs. A bead is the basic unit of the 3D genome structure. Bead–bead contacts are also determined from Hi-C data, and LADs are mapped from lamin ChIP-seq data. To constrain distances between pairs of beads in the model, statistically significant Hi-C contact counts between their respective genomic positions are calculated (Eqs. 2 and 3). LADs are used to define bead constraints toward the nuclear periphery. Throughout the protocol, constraint specifications are added to a Model Setup File in GTrack format[46]. This file is used as a core input in the Chrom3D executable file, which then computes the 3D structural model. The output 3D genome model is provided in XML format, which is compatible with the interactive 3D visualization tool Chimera[47] and can be imported into other software.

Modeling resolution. Modeling resolution refers to the number of beads that make up the 3D genome model. This number depends on the resolution of the input data, such as bin size of the Hi-C data, but may also depend on the sizes and numbers of genomic regions that will constitute the basic unit of the 3D model, which are defined by the user. The protocol described here considers a

Box 2 | Specifying bead characteristics

In addition to specifying bead id and position, other bead characteristics can be added in the Model Setup File. Bead size can be adjusted in a column named `radius`:

```
###seqid      start      end      id      edges      periphery  radius
chr1          0          1000     a       e          .          1.0
chr1          1000       2000     b       .          .          0.5
chr1          2000       3000     c       .          .          1.0
chr1          3000       4000     d       .          .          0.5
chr1          4000       5000     e       a          1          1.0
```

Note that units of bead radii are arbitrary unless information on nucleus size is given. Nucleus size is specified using the `-r/--radius` parameter in Chrom3D. If instead beads need to have a size proportional to their genomic size (`end-start`), then the `-y/--occupancy` parameter can be specified. This option will cause each bead to have a radius depending on bead genomic size, and scaled such that the total volume occupancy of all beads is as specified by the option. A recommended total volume occupancy value is 0.15, indicating that total bead volume is 15% of the nucleus volume. This option requires that the `-r/--radius` be specified. Each bead can also include information that can be color-coded. For example, we have differentially color-coded beads containing LADs, interacting beads and interacting beads containing LADs (**Fig. 3c–e**). Other examples may be a gene-expression value or a given chromatin state for a bead. Color can be included by adding a column with the name `color`:

```
###seqid      start      end      id      edges      periphery color
chr1          0          1000     a       e          .         1.0,0.0,0.0
chr1          1000       2000     b       .          .         0.5,0.5,0.5
chr1          2000       3000     c       .          .         0.5,0.5,0.5
chr1          3000       4000     d       .          .         0.5,0.5,0.5
chr1          4000       5000     e       a          1         0.0,0.0,1.0
```

Colors are specified as RGB values using a comma-separated list of values between 0 and 1. If no bead color is specified, beads will be given a random color per chromosome. Colors do not affect anything in the 3D model other than bead color in the output (CMM) file.

Box 3 | Adding weights to constraints

Each constraint in the Model Setup File can be prioritized with a given weight. This is done using an equal sign (=) for edges and a comma-separated list for nuclear periphery constraints:

```
###seqid     start     end     id     edges     periphery
chr1         0         1000    a      e=0.5     .
chr1         1000      2000    b      .         .
chr1         2000      3000    c      .         .
chr1         3000      4000    d      .         .
chr1         4000      5000    e      a=0.5     1,1.5
```

Here, the constraint between beads a and e is given a relative weight of 0.5. Weight must be specified symmetrically for both beads. The periphery constraint on bead e is given a weight of 1.5. The specified weights correspond to k_{ij} in equation (4) in the Introduction. If no weight is given, a weight of 1 is assumed.

total of 15,282 beads for a diploid whole human genome, based on contact domains[27] (here called TADs for simplicity). The portions of the genome that are not a priori defined by a domain, such as centromeres, gaps or unmappable parts of the genome, are still represented by a bead. In this protocol, total bead volume represents 15% of the volume of a nucleus of 10 μm in diameter, similar to previous studies[22,48]. Total bead volume can also be specified by the user (**Box 2**). When the user specifies the desired total bead volume fraction (V) (e.g., 15%) for a nucleus with a radius of size R (the default is 5 μm), all individual bead radii (r_i) will be scaled according to their genomic size (l_i) relative to the genomic size of the entire genome (L), as described earlier[22] and as shown here:

$$r_i = \sqrt[3]{\frac{Vl_i}{L}}R \qquad (1)$$

Significant bead–bead interactions determined from Hi-C data. This protocol relies on the identification of significant interactions between pairs of beads, together with LAD information, to define distance constraints between two beads and between a bead and the nuclear periphery, respectively. LADs are provided as a BED file and are identified from significant associations of nuclear lamins with chromatin as compared with background profiles. These interactions are provided by analysis of lamin ChIP-seq data (in this protocol), using a domain-calling algorithm we developed earlier (Enriched Domain Detector (EDD))[44]. Generation of the LAD BED file is not part of Chrom3D and should be done before starting with Chrom3D.

The use of significant Hi-C interactions in our protocol contrasts with other modeling techniques that rely on all contact frequencies, for example, by taking the inverse contact count as a proxy for distance[49,50], or that select contacts from a cutoff value[34,51]. Our protocol uses a noncentral hypergeometric

Box 4 | Specifying specific constraint distances between beads

As an advanced option, it is possible to define specific distances between pairs of beads as constraints, in contrast to simply constraining beads as close as possible to one another. This is done using a list of three comma-separated numbers in the `edges` column:

```
###seqid     start     end     id     edges
chr1         0         1000    a      e=0.5,2.0,0
chr1         1000      2000    b      .
chr1         2000      3000    c      .
chr1         3000      4000    d      .
chr1         4000      5000    e      a=0.5,2.0,0
```

The first number (`0.5`) specifies that beads a and e are constrained with a weight of `0.5`, the second number (`2.0`) gives the optimal distance between bead a and e. The third number (`0`) indicates that the given distance is an upper-bound distance, meaning that higher values than this should be penalized (i.e., loss score >0). If instead a `1` is specified at this position, a lower bound is used on the distance (i.e., lower distances are penalized):

```
chr1     0     1000     a     e=0.5,2.0,1
```

If a dot (`.`) is given, the specified distance (`2.0`) is neither bounded below or above. and it indicates the specific optimal distance for the given bead–bead constraint:

```
chr1     0     1000     a     e=0.5,2.0,.
```

All given distances should be specified with the same units as used for bead and nucleus radii. Bead–bead distances are specified between the surfaces of their beads. As for regular pairwise constraints (**Box 1**), multiple constraints can be specified by using a semicolon-separated list.

Box 5 | Local modeling

A powerful option of Chrom3D is to model a selected genomic region to investigate looping and folding properties, for example, around a gene locus. This has previously been explored for the α-globin locus region, based on IMP[34]. We illustrate here a local modeling of the same region in the GM12878 cell line using Chrom3D (**Fig. 4**):

```
./Chrom3D -n 10000 -l 1000 -T 1.0 -c 0.0005 -e translate -e rotate ./test_files/GM12878.
gtrack > GM12878.cmm
```

The 'GM12878.gtrack' file is a Model Setup File adapted and modified from the example in ref. 34. In that example, each bead is an individual restriction fragment. In the command above, `-n 10000` specifies 10,000 iterations, `-l 1000` specifies that logging information should be output every 1,000 iterations, `-T 1.0` sets the starting temperature to 1.0 and `-c 0.0005` sets the cooling rate of the temperature. The parameters `-e translate` and `-e rotate` exclude the 'translate' and 'rotate' moves from the optimization (**Fig. 2b**); these are redundant for local simulations, as they will not alter the loss score.

(NCHG) distribution to calculate the probability of observing a given number of Hi-C contacts (n_{ij}), dependent on the total number of contacts (n), the number of contacts involved for the two beads (n_i and n_j) and the genomic (linear) distance between them:

$$P(n_{ij}\,|\,n,n_i,n_j,\omega_{ij}) = \frac{\binom{n_i}{n_{ij}}\binom{2n-n_i}{n_j-n_{ij}}\omega_{ij}^{n_{ij}}}{\sum_{n'_{ij}}\binom{n_i}{n'_{ij}}\binom{2n-n_i}{n_j-n'_{ij}}\omega_{ij}^{n'_{ij}}} \qquad (2)$$

where

$$\omega_{ij} = \frac{\lambda_{ij}(2\lambda-\lambda_i-\lambda_j+\lambda_{ij})}{(\lambda_i-\lambda_{ij})(\lambda_j-\lambda_{ij})} \qquad (3)$$

defines the hypothesized odds ratio of expectations, based on λ_{ij}, the expected number of interactions for beads i and j. This can be estimated from the contacts between all possible bead pairs in the Hi-C data, for given interbead genomic distances. This probability is then used to calculate a P value in order to identify significant contacts, after correction for multiple testing with a 1% false-discovery rate (FDR).

With 3D structural modeling from selected bead–bead interactions, there must be a balance of the ratio of constraints to the total number of possible pairwise interactions. We have found that a reasonable balance is achieved by applying an additional selection criterion to the effect size by requiring that the difference between the observed and expected $\log_2$-odds be ≥2. This cutoff is based on our experience of using these contacts as constraints in our modeling and can be compared with log-fold cutoff values used in gene-expression analyses[52]. It may be possible to use other statistical tools to identify significant interactions for constraints[52,53], but these must be tested.

Local modeling. Our protocol describes the modeling of a whole diploid human fibroblast genome. Using Chrom3D, it is also possible to model local regions such as individual loci. We have previously illustrated a modeling scenario from 5C data for an ~500-kb region (ENm008 ENCODE) containing the α-globin locus, using individual restriction fragments as bead sizes[24] (**Fig. 4**).

Weighted constraints. In this protocol, all constraints are weighed equally. However, Chrom3D is designed to allow a versatile model definition by specification of weights on each constraint through the k parameter of the loss-score function (L), where the Euclidean distance between pairs of beads in the model (b_i and b_j) is compared with user-specified target distances (d_{ij}).

$$L = \sum_{i,j} k_{ij}(\|b_i - b_j\| - d_{ij})^2 \qquad (4)$$

Box 6 | Chrom3D initial output

```
# Rescaled bead sizes from 122.256 to 0.15
# chromosomes: 46
# beads: 15282
# interactions: 15780
# interactions with given weight: 0
# non-bounded interactions with given distance: 0
# upper-bounded interactions with given distance: 0
# lower-bounded interactions with given distance: 0
# periphery beads: 2528
# periphery beads (with weights): 0
# center beads: 12754
# center beads (with weights): 0
```

Defining *k* for each constraint enables weighting and prioritizing of individual constraints during the simulations. These weights can for instance be based on prior knowledge of the degree of trust one has in the data used to define the constraints. Weights are defined in the Model Setup File (**Boxes 3** and **4**). If a target distance is not specified, it is set to zero, constraining the selected beads as close as possible to each other.

Adapting the procedure. Our protocol has proven successful with whole human genome modeling from Hi-C and lamin ChIP-seq data. Chrom3D can also be used for genome modeling at restriction fragment size resolution, as shown for the α-globin locus[24]. These examples illustrate the versatile nature of Chrom3D. It is therefore reasonable to expect that Chrom3D can be tuned for a range of scales and resolutions. In these cases, however, we recommend that the user explore how the model constraints should be selected before modeling. For instance, for local modeling at the megabase scale, bead–bead interaction constraints should probably be based more directly on contact frequencies, as proposed by Bau *et al.*[34] in their analysis of the alpha-globin locus region using IMP, rather than using the statistical approach used in this protocol. **Box 5** describes modifications to the constraints that can be made in order to achieve local modeling.

We have tested our protocol on Hi-C resolutions ranging from 50 kb to 1 Mb and expect that higher- and lower-resolution input data are also suitable. Model resolution, i.e., the size and number of beads in the model, is defined in Step 1. Here, we have illustrated the specification of beads based on Arrowhead domains. Arrowhead is available on GitHub (https://github.com/theaidenlab/juicer/wiki/Arrowhead). A variety of TAD and domain-calling algorithms exist, which can be used to define genomic coordinates in this protocol. If the beads are to be based on some domain other than TADs, the list of Arrowhead domains in Step 1 should be replaced by a BED file listing the genomic coordinates of each of the beads.

In addition, even though only lamin ChIP-seq data have been used to date to define peripheral constraints, the use of other datasets, such as DamID[3], should not be problematic. In this case, the BED file provided in Step 9 must be replaced by another BED file specifying the genomic coordinates of the regions to be constrained toward the periphery. Any overlap between genomic coordinates in this BED file and the beads specified in Step 1 will result in a constraint of these beads toward the nuclear periphery. To call broad peaks from LAD or similar data, EDD[44] can be used (available on GitHub: https://github.com/CollasLab/edd).

The computational time required for an entire simulation (Step 15) increases depending on the total number of beads and the total number of constraints defined in the Model Setup File. This also means that especially large genomes, or very-high-resolution models will require considerably more computation time. The memory requirement is, however, not substantially affected.

We anticipate no problems using our protocol on data from nonhuman species. If genome assemblies other than hg19 are used, the user must provide the appropriate (and sorted) genome assembly–specific chromosome size file instead of the 'hg19.chrom.sizes.sorted' file used in this protocol. If resolutions other than 1 Mb are used for interchromosomal interaction data, the `resolution` variable in the 'make_interchr_NCHG_input.py' script must be changed accordingly. Filenames of the Hi-C data files must have the same format if the protocol is run on Hi-C data other than those provided with this protocol. If data from a species other than human are to be used, the Hi-C data must be provided in COO format per chromosome, and filenames must be formatted in the same way as for the protocol described here, that is, with chromosome names in the file name. Moreover, a sorted file with chromosome sizes must be provided, as well as a custom 'Unmappable_blacklist.bed' file containing unmappable regions of the genome[54] (if applicable). All other data must be provided in the same format as described in this protocol. When filtering the significant TAD–TAD interactions (Step 4), the fraction of significant interactions relative to the total number of TAD–TAD pairs should be in the range of 0.5–1%. This fraction can be tuned by altering the log-odds cutoff value, if needed.

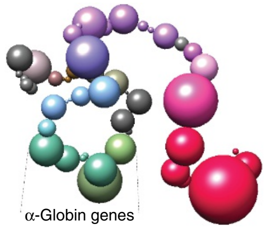

Figure 4 | Local modeling using Chrom3D. Chrom3D model of the ENCODE ENm008 region (~500 kb) containing the human α-globin locus. Input data were 5C data from Bau *et al.*[34] in GM12878 lymphoblastoid cells; a bead represents an individual restriction fragment. No lamin ChIP-seq data are needed for this modeling.

We describe here the generation of a single 3D genome structural model using Chrom3D. We note that a single model is informative for visualization purposes, but for quantifiable estimations of positioning of loci in the nucleus, users will need to run the same simulation (Step 15) multiple times. Specifically, we recommend generating 50–500 independent models to obtain estimates of the distribution of radial positioning in selected regions (such as TADs). To investigate more specific structural positioning and pairwise distance distributions between loci, we recommend 1,000–5,000 independent models. To study interchromosomal pairwise distance distributions, ~1,000 models will typically be needed. Such independent model simulations are initiated from distinct random chromosome configurations by specification of different random seeds. As the construction of a full-genome model can take several days to complete (depending on resolution), we recommend generating independent models in parallel on a computer cluster. We have previously shown that multiple independent Chrom3D models recapitulate some of the spatial variation in genome conformation seen in single cells[24].

Level of expertise required for implementation. This protocol is designed for users experienced with the Unix command-line interface.

MATERIALS

EQUIPMENT

Starting data

- Hi-C matrices for each chromosome in COO format
- Arrowhead domain BED file used as the basis for bead genomic coordinates
- Lamin-associated domains (LAD) BED file
- Scripts necessary to run the protocol (provided in the 'chrom3d_pipeline_data' directory below)
- Example of Arrowhead domains and LAD data from the fibroblast cell lines for use as tutorial

Software

- Operating system: Linux and Macintosh
- Python v2.7 (https://www.python.org/downloads/)
- Python library—pybedtools: https://daler.github.io/pybedtools/main.html
- Python library—statsmodels: https://pypi.python.org/pypi/statsmodels
- Boost library: http://www.boost.org/users/download/
- bedtools: http://bedtools.readthedocs.io/en/latest/content/installation.html
- NCHG: https://folk.uio.no/jonaspau/hic/NCHG_hic.zip
- Chrom3D v1.0.1: https://github.com/Chrom3D/Chrom3D/releases/v1.0.1
- Chimera: https://www.cgl.ucsf.edu/chimera/download.html

EQUIPMENT SETUP

Required data We describe this protocol using Hi-C data from Rao *et al.*[27] and lamin ChIP-seq data from Lund *et al.*[44] for human fibroblast cell lines. Adding LAD information and interchromosomal interactions is optional. Input data for this PROCEDURE are freely available for download in the archived file 'chrom3d_pipeline_data.tar.gz' and are described in the section 'Downloading and organizing the data'.

Downloading and installing software Follow the instructions in the installation links provided in the Software section above to install common software such as Python, BedTools and Chimera.

To install NCHG, download the 'NCHG_hic.zip' file, unzip the file and change to the source directory:

```
curl -O http://folk.uio.no/jonaspau/hic/NCHG_hic.zip
unzip NCHG_hic.zip
cd NCHG_hic
make
export PATH=$PATH:${PWD}
```

Before installing Chrom3D, the Boost library ≥1.54 must be installed. To install the Boost library, go to the Chrom3D GitHub page (https://github.com/CollasLab/Chrom3D) for detailed installation instructions.

To install Chrom3D, clone it from the GitHub page (assuming git is installed), and change to the directory:

```
  curl -O -L https://github.com/Chrom3D/Chrom3D/
archive/v1.0.1.tar.gz
  tar -xvzf v1.0.1.tar.gz
  cd Chrom3D-1.0.1
  make
  export PATH=$PATH:${PWD}
```

▲ CRITICAL All commands and scripts in this protocol should be executed in the Unix shell prompt (we suggest that Bash be used) using a terminal window. Furthermore, BedTools is assumed to be in the executable path. To carry out the protocol, we recommend creating a separate directory and downloading all scripts and data into this directory.

Downloading and organizing the data The size of the compressed example datasets used to execute the PROCEDURE is too large to be provided as supplementary files. Instead, we provide a directory containing all scripts and data necessary to carry out the PROCEDURE. Download the compressed example data and scripts, and extract them:

```
curl -O http://folk.uio.no/tmali/NatProtSuppl/
chrom3d_pipeline_data.tar.gz
tar -xvzf chrom3d_pipeline_data.tar.gz
cd chrom3d_pipeline_data
```

The directory 'chrom3d_pipeline_data' contains files and subdirectories listed in **Table 1**. All steps described below are assumed to be run in the directory 'chrom3d_pipeline_data'.

TABLE 1 | 'Chrom3d_pipeline_data' directory.

File or directory	Description
Scripts (Bash and Python) required	
GSE63525_IMR90_Arrowhead_domainlist.txt	Domains based on the Arrowhead algorithm[27] in a BED12 format file
GSM1313399_HSF_AD04_LMNA_rep1.bed	LADs defined using the EDD peak caller[44], in a three-column BED format
hg19.chrom.sizes.sorted	A text file containing the chromosome sizes in the hg19 assembly, from which chrY, chrM and other patch chromosomes are removed. The file is sorted numerically by chromosome names, matching the format of the names in the Hi-C data
intra_chr_RAWobserved/	This subdirectory contains the intrachromosomal Hi-C matrices at 50-kb resolution in the sparse matrix (COO) format
inter_chr_RAWobserved/	This subdirectory contains the interchromosomal Hi-C matrices at 1-Mb resolution in the sparse matrix (COO) format
unmappable_blacklist.bed	This file contains genomic regions (e.g., centromeres and telomeres) that must be removed as described in Step 10
lad_ad04.ids	This file contains identifiers of beads containing LADs
TAD_interaction.ids	This file contains identifiers of interacting beads

PROCEDURE

Aggregation of Hi-C contact counts for all pairs of TADs ● TIMING ~1 h

1| To create the BED file specifying the genomic positions of the beads, Arrowhead domains (TADs) must be merged, and gaps between them must be filled.

To do so, run the following script on the command line:

```
bash arrowhead_to_domains.sh GSE63525_IMR90_Arrowhead_domainlist.txt hg19.chrom.
sizes.sorted
```

```
DESCRIPTION:
bash arrowhead_to_domains.sh [arrowheadFile] [chromSizeFile]
[arrowheadFile]    A text file listing the Arrowhead domains
[chromSizeFile]    A text file containing the chromosome sizes (sorted numerically)
```

2| Concatenate all the .domains to use in Step 7:

```
cat *.chr*.domains > GSE63525_IMR90_Arrowhead_domainlist.domains
```

3| Compute intrachromosomal interaction counts between TADs. This requires .domains files for all chromosomes (example: 'GSE63525_IMR90_Arrowhead_domainlist.chr19.domains') created above in Step 1, and the raw interaction count matrix for each chromosome in COO format. This step aggregates the intrachromosomal interaction counts based on the domains and creates a BEDPE-format file. The runtime for this step depends on chromosome size and on the resolution of the Hi-C matrix. This step can be performed for each chromosome individually or for all chromosomes at once. For a single chromosome, execute the following command (here, for chromosome 19):

```
mkdir intrachr_bedpe
```

```
bash make_NCHG_input.sh GSE63525_IMR90_Arrowhead_domainlist.chr19.domains
intra_chr_RAWobserved/chr19_50kb.RAWobserved chr19 >
intrachr_bedpe/chr19_50kb.domains.RAW.bedpe
```

To carry out the above command for all chromosomes, run the following script:

```
bash intrachr_NCHG_input_auto.sh GSE63525_IMR90_Arrowhead_domainlist
hg19.chrom.sizes.sorted 50kb
```

```
DESCRIPTION:
bash intrachr_NCHG_input_auto.sh [domainBase] [chromSizeFile] [resolution]
[domainBase]       Basename of the domain files
[chromSizeFile]    Text file containing the chromosome sizes (sorted numerically)
[resolution]       Resolution of the interaction matrix as given in the matrix
                   filename (eg. 50kb or 1mb)
```

? TROUBLESHOOTING

4| Once all processes are finished, concatenate all the BEDPE files into a single file:

```
cat intrachr_bedpe/chr*.bedpe > intrachr_bedpe/IMR90_50kb.domain.RAW.bedpe
```

5| Because some domains may contain centromeres and other unmappable regions, data in these domains may introduce artifacts in the statistical test described in Step 6. Therefore, it is important to remove domains that contain centromeres from the BEDPE file.

To do so, once all the BEDPE files are created and concatenated (Step 4), execute the following command:

```
curl -s "http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/cytoBand.txt.gz" |
gunzip -c | grep acen | pairToBed -a intrachr_bedpe/IMR90_50kb.domain.RAW.bedpe -b
stdin -type neither > intrachr_bedpe/IMR90_50kb.domain.RAW.no_cen.bedpe
```

Identification of significant intrachromosomal interactions ● TIMING ~15 min

6| Calculate the *P* values and odds ratios for Hi-C contacts between each pair of domains:

```
NCHG -m 50000 -p intrachr_bedpe/IMR90_50kb.domain.RAW.no_cen.bedpe >
IMR90_50kb.domain.RAW.no_cen.NCHG.out

DESCRIPTION:
NCHG -m[minDistance] -p[NULL] [inputFile]
-m    Minimum genomic distance (in bp) allowed between domains, below which
      interactions are excluded
-p    Print output to stdout
[inputFile] Hi-C contact count data in BEDPE format
```

7| Calculate the FDR and filter significant interactions based on FDR values and odds ratios by executing the following Python script:

```
python NCHG_fdr_oddratio_calc.py IMR90_50kb.domain.RAW.no_cen.NCHG.out fdr_bh 2 0.01 >
IMR90_50kb.domain.RAW.no_cen.NCHG.sig

DESCRIPTION:
python NCHG_fdr_oddratio_calc.py [inputFile] [testMethod] [cutoff] [threshold]
[inputFile]        Input filename (NCHG output; BEDPE format)
[testMethod]       Multiple hypothesis testing method (bonferroni, sidak, holm-sidak,
                   holm, simes-hochberg, hommel, fdr_bh, fdr_by, fdr_tsbh, fdr_tsbky)
[cutoff]           Odds ratio cutoff value
[threshold]        Significance threshold value after multiple testing correction
```

Combining TAD interactions and LAD information into a Model Setup File (GTrack) ● TIMING ~45 min

8| Create the Model Setup File (GTrack format) using the significant interactions. This entails creating identifiers (ids) for the beads and an 'edge' column that specify interactions between the beads (**Box 1**):

```
bash make_gtrack.sh IMR90_50kb.domain.RAW.no_cen.NCHG.sig
GSE63525_IMR90_Arrowhead_domainlist.domains
IMR90_intra_chromosome.gtrack

DESCRIPTION:
bash make_gtrack.sh [sigFile] [domainFile] [outputFile]
[sigFile]          Intra-chromosome significant interactions file (from Step 7)
[domainFile]       Domain file (concatenated; from Step 2)
[outputFile]       Output GTrack file (Model Setup File)
```

? TROUBLESHOOTING

9| (Optional) Add LAD information to the Model Setup File (GTrack). This step intersects the Model Setup File with LADs to add information to the beads, constraining these to the nuclear periphery. Execute the following script:

```
bash make_gtrack_incl_lad.sh IMR90_intra_chromosome.gtrack
GSM1313399_HSF_AD04_LMNA_rep1.bed IMR90_intra_chromosome_w_LADs.gtrack

DESCRIPTION:
bash make_gtrack_incl_lad.sh [inputFile] [ladFile] [outputFile]
[inputFile]        Input GTrack file (without LAD information)
[ladFile]          LAD BED file to be added to GTrack file
[outputFile]       Output GTrack file with LAD information added
```

10| Prepare interchromosomal Hi-C interaction counts. This step requires an interchromosomal interaction Hi-C matrix for each pair of chromosomes:

```
bash ./interchr_NCHG_input_auto.sh hg19.chrom.sizes.sorted unmappable_blacklist.bed
1mb > IMR90_1mb_inter.bedpe

DESCRIPTION:
bash interchr_NCHG_input_auto.sh [chromSizeFile] [blackList] [resolution]
[chromSizeFile]         Text file containing the chromosome sizes (sorted numerically)
[blackList]             BED file containing positions of blacklisted regions
[resolution]            Resolution of the interaction matrix as given in the matrix
                        filename (e.g. 50kb or 1mb)
```

11| Call significant interchromosomal interactions. To run NCHG for interchromosomal interactions, execute the following command:

```
NCHG -i -p IMR90_1mb_inter.bedpe > IMR90_1mb_inter_chr.NCHG.out

DESCRIPTION:
NCHG -i[NULL] -p[NULL] [inputFile] > [outputFile]
-i             Instructs NCHG to use inter-chromosomal interactions
-p             Instructs NCHG to print output to stdout
[inputFile]    Hi-C contact count data (BEDPE format)
[outputFile]   File containing P-values for each TAD pair
```

12| Perform multiple-testing correction on interchromosomal interactions:

```
python NCHG_fdr_oddratio_calc.py IMR90_1mb_inter_chr.NCHG.out fdr_bh 2 0.01 >
IMR90_1mb_inter_chr.NCHG.sig

DESCRIPTION:
python NCHG_fdr_oddratio_calc.py [inputFile] [testMethod] [cutoff] [threshold]
[inputFile]    Input filename (NCHG output; BEDPE format)
[testMethod]   Multiple hypothesis testing method (bonferroni, sidak, holm-sidak,
               holm, simes-hochberg, hommel, fdr_bh, fdr_by, fdr_tsbh, fdr_tsbky)
[cutoff]       Odds ratio cutoff value
[threshold]    Significance threshold value after multiple testing correction
```

13| Add significant interchromosomal interacting beads to the Model Setup File (which already contains intrachromosomal interaction information):

```
bash add_inter_chrom_beads.sh IMR90_intra_chromosome_w_LADs.gtrack IMR90_1mb_
inter_chr.NCHG.sig IMR90_inter_intra_chr_w_LADs.gtrack

DESCRIPTION:
bash add_inter_chrom_beads.sh [inputFile] [sigFile] [outFile]
[inputFile]        Input GTrack file
[sigFile]          Inter-chromosome significant interaction file (from Step 12)
[outFile]          Output GTrack file containing significant inter-chromosomal interactions
```

? TROUBLESHOOTING

14| (Optional) Modify the Model Setup File (GTrack) to allow a diploid model:

```
python make_diploid_gtrack.py IMR90_inter_intra_chr_w_LADs.gtrack >
IMR90_inter_intra_chr_w_LADs.diploid.gtrack

DESCRIPTION:
python make_diploid_gtrack.py [inputFile] > [outputFile]
```

```
[inputFile]    Input GTrack file with constraints specified for single chromosomes
[outputFile]   Output GTrack file with constraints for each chromosome copy
```

Running Chrom3D to generate 3D genome models ● TIMING ~15 h

15| Generate a single 3D model based on the Model Setup File (GTrack):

```
Chrom3D -y 0.15 -r 5.0 -n 2000000 -o IMR90_inter_intra_chr_w_LADs.diploid.cmm IMR90_
inter_intra_chr_w_LADs.diploid.gtrack

DESCRIPTION:
Chrom3D -y[scale] -r[radius] -n[iterations] -o[outputFile] [setupFile]
-y            Scale total volume of the model beads relative to the volume of the nucleus
-r            Radius of the nucleus in micrometers
-n            Number of iterations
-o            Output filename
[setupFile]   Model Setup File in GTrack format
```

? TROUBLESHOOTING

Visualizing 3D genome models ● TIMING ~15 min

16| Use Chimera to visualize the 3D genome model in the resulting CMM file (see **Fig. 3** for example output figures). Start Chimera and open the 3D genome model file by clicking on 'File' → 'Open' and using the 'File type' menu to choose 'Chimera markers'. Navigate to the directory containing the .cmm file (in this case 'interchr_bedpe/IMR90_inter_intra_chr_w_LADs.diploid.cmm') and click on 'Open'.

Alternatively, assuming Chimera is in the PATH and the CMM file is in the current working directory, run the following command in the terminal:

```
chimera IMR90_inter_intra_chr_w_LADs.diploid.cmm
```

17| Run the following command (in the terminal) to render all LAD-containing beads blue; all other beads will be colored gray (**Fig. 3d**):

```
python color_beads.py IMR90_inter_intra_chr_w_LADs.diploid.cmm
lad_ad04.ids 0,0,255 override >
IMR90_inter_intra_chr_w_LADs.diploid.visLAD.cmm

DESCRIPTION:
python color_beads.py [inputFile] [beadIdFile] [color] [colorScheme]
[inputFile]     Input CMM file
[beadIdFile]    File containing selected bead ids
[color]         Comma-separated RGB value
[colorScheme]   'blend' or 'override' color of the beads specified
```

18| Execute the following command to make all Hi-C-constrained beads red:

```
python color_beads.py IMR90_inter_intra_chr_w_LADs.diploid.visLAD.cmm TAD_interaction.
ids 255,0,0 blend > IMR90_inter_intra_chr_w_LADs.diploid.visLADCons.cmm
```

Beads associated with LADs from above (blue) will be colored purple, due to the 'blend' parameter (**Fig. 3e**).
? TROUBLESHOOTING

19| Visualize the output in Chimera:

```
chimera IMR90_inter_intra_chr_w_LADs.diploid.visLADCons.cmm
```

20| Generate publication-ready images of the 3D genome models (Steps 21–24) by loading a 3D model (CMM) into Chimera as described in Step 16.

21| Open the command-line interface by clicking on 'Favorites' → 'Command Line' and finding the interface at the bottom of the Chimera GUI to add commands.

22| Make the background white and set the depth-cuing to black by typing

```
background solid white
set dcColor black
```

23| Add a nuclear envelope (nucleus border) representation to the structure (**Fig. 3a**):

```
shape sphere center 0,0,0 radius 5.0 color #FFE4B5 slab 0.5
mclip #1
transparency 30 #1
```

24| Save the image by clicking on 'File' → 'Save Image' and using the 'File type' menu to choose an image format. Navigate to the directory in which to save the image and click on 'Save'.

? TROUBLESHOOTING

Troubleshooting advice can be found in **Table 2**.

TABLE 2 | Troubleshooting table.

Step	Problem	Possible reason	Solution
1–18	Invalid output or the program stops	Arguments are not in order	Pass arguments in an order specific to the program
3, 8 and 13	Error messages and the program stops	One of the scripts cannot find the input file	Run the scripts from the script directory
15	Error message: std::bad_alloc	One or more chromosomes as specified in the Model Setup File (GTrack) contain too many beads	Generate a new Model Setup File with fewer defined beads
	Chrom3D returns an error message	Wrong format of the Model Setup File (GTrack)	Validate the format of the Model Setup File (GTrack) (**Box 1**)

● TIMING

Executing the entire PROCEDURE for the given whole-genome data takes ~18 h. The most time-consuming sections are Steps 3 and 15. The runtime of these steps depends on the resolution of the Hi-C data and chromosome sizes.

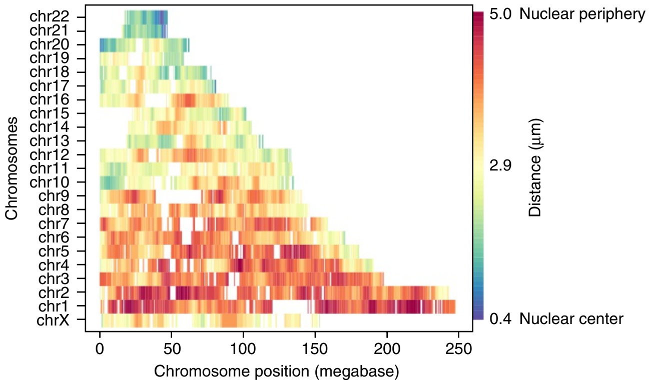

Figure 5 | Heat map of radial distances (nuclear center to periphery) of beads (here, TADs) on each chromosome modeled in IMR90 fibroblasts using Chrom3D. Note that this particular map results from the modeling of a single genome structure and, as such, cannot provide quantified information on the radial placement of TADs; this requires the generation of a large number of models[24]. Distances from the nucleus center to the periphery are shaded from blue to red, respectively, assuming a spherical nucleus radius of 5 µm.

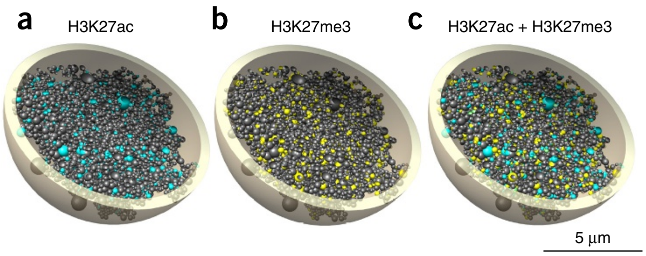

Figure 6 | 3D genome models of IMR90 fibroblasts representing epigenomic data for histone modifications. (**a**–**c**) Datasets shown are for H3K27ac (**a**), H3K27me3 (**b**) and both (**c**). Histone modification ChIP-seq data were downloaded from Gene Expression Omnibus (GEO) accession number GSE16256 with track numbers GSM521887 (H3K27ac) and GSM469968 (H3K27me3).

Steps 1–5, aggregation of Hi-C counts: ~1 h
Steps 6 and 7, identification of significant Hi-C interactions: ~15 min
Steps 8–14, addition of interactions and LADs, and creation of the GTrack file: ~45 min
Step 15, generation of 3D structure models: ~15 h
Steps 16–24, visualizing 3D genome models: ~15 min or more, depending on the user and the application

ANTICIPATED RESULTS

The protocol results in the generation of an XML file (CMM format) containing the *x*, *y* and *z* coordinates of the final, optimized structure (Step 15). This structure can be visualized using, e.g., Chimera (Steps 16–24) (**Fig. 3**). If the user adds an optional `-l` parameter in Step 15, logging information will be provided at fixed intervals during the simulation. By plotting the loss-score for each iteration number in the log, convergence of the optimization can be evaluated (**Fig. 2b**). This function should reach a stable state at the end of the optimization.

The CMM file can be modified after the simulation to enable coloring of selected beads. For example, individual chromosomes can be colored (**Fig. 3a**), rendering chromosome territories evidenced by FISH using chromosome paints and microscopy[26]. Interacting and/or LAD-containing beads can also be colored (**Fig. 3d**,**e**). In this manner, the peripheral placement of the LAD-containing beads (blue) can be evaluated (**Fig. 3d**). Note that not all beads will necessarily be placed within the specified radius of the nucleus (here 5 μm). This is expected because the nucleus is not perfectly spherical. The structure itself can also be converted into a plot showing each bead's Euclidean distance to the nucleus center, to illustrate the nuclear peripheral or central placement of beads. We have found a more peripheral placement of larger, gene-poor chromosomes, and a more central positioning of smaller chromosomes (**Fig. 5**). Furthermore, coloring beads according to selected features such as domains enriched in given post-translational histone modifications (e.g., H3K27me3 or H2K27ac) can reveal the placement of these relative to each other and to the nuclear boundary (**Fig. 6**). In addition to the visualization options presented here, the resulting 3D genome model can also be part of a downstream analysis. As an initial validation of the optimized models, contact matrices can be generated from the pairwise distances between beads by aggregating contacts between beads in close proximity. These contacts can then be correlated with the input Hi-C contact matrices (see ref. 24 for details). Correlation coefficients >0.9 indicate a robust correspondence between the models and the input data. Chrom3D can also be used for local 3D modeling of specific genome regions (**Box 5**). An example structure of the ENCODE ENm008 region containing the α-globin locus is shown in **Figure 4**.

ACKNOWLEDGMENTS P.C. is supported by the Research Council of Norway, the Norwegian Cancer Society, South-East Health Norway, The Norwegian Center for Stem Cell Research and the University of Oslo.

AUTHOR CONTRIBUTIONS J.P. and T.M.L.A. wrote code, performed data analysis and wrote the manuscript. J.P. conceptualized and developed Chrom3D. P.C. supervised the work and wrote the manuscript.

COMPETING INTERESTS
The authors declare no competing interests.

Reprints and permissions information is available online at http://www.nature.com/reprints/index.html. Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Dekker, J., Marti-Renom, M.A. & Mirny, L.A. Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. *Nat. Rev. Genet.* **14**, 390–403 (2013).
2. Lieberman-Aiden, E. *et al.* Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009).
3. Guelen, L. *et al.* Domain organization of human chromosomes revealed by mapping of nuclear lamina interactions. *Nature* **453**, 948–951 (2008).
4. van Steensel, B. & Dekker, J. Genomics tools for unraveling chromosome architecture. *Nat. Biotechnol.* **28**, 1089–1095 (2010).
5. Collas, P., Lund, E.G. & Oldenburg, A.R. Closing the (nuclear) envelope on the genome: how nuclear lamins interact with promoters and modulate gene expression. *BioEssays* **36**, 75–83 (2014).
6. Lund, E.G., Duband-Goulet, I., Oldenburg, A., Buendia, B. & Collas, P. Distinct features of lamin A-interacting chromatin domains mapped by ChIP-sequencing from sonicated or micrococcal nuclease-digested chromatin. *Nucleus* **6**, 30–38 (2015).
7. Gesson, K. *et al.* A-type lamins bind both hetero- and euchromatin, the latter being regulated by lamina-associated polypeptide 2 alpha. *Genome Res.* **26**, 462–473 (2016).
8. Robson, M.I. *et al.* Tissue-specific gene repositioning by muscle nuclear membrane proteins enhances repression of critical developmental genes during myogenesis. *Mol. Cell* **62**, 834–847 (2016).
9. Robson, M.I. *et al.* Constrained release of lamina-associated enhancers and genes from the nuclear envelope during T-cell activation facilitates their association in chromosome compartments. *Genome Res.* **27**, 1126–1138 (2017).
10. Dixon, J.R. *et al.* Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**, 376–380 (2012).
11. Nora, E.P. *et al.* Spatial partitioning of the regulatory landscape of the X-inactivation centre. *Nature* **485**, 381–385 (2012).
12. Nagano, T. *et al.* Cell-cycle dynamics of chromosomal organization at single-cell resolution. *Nature* **547**, 61–67 (2017).
13. Hansen, A.S., Pustova, I., Cattoglio, C., Tjian, R. & Darzacq, X. CTCF and cohesin regulate chromatin loop stability with distinct dynamics. *Elife* **6** e25776 (2017).
14. Barbieri, M. *et al.* Complexity of chromatin folding is captured by the strings and binders switch model. *Proc. Natl. Acad. Sci. USA* **109**, 16173–16178 (2012).
15. Bohn, M. & Heermann, D.W. Repulsive forces between looping chromosomes induce entropy-driven segregation. *PLoS One* **6**, e14428 (2011).
16. Mateos-Langerak, J. *et al.* Spatially confined folding of chromatin in the interphase nucleus. *Proc. Natl. Acad. Sci. USA* **106**, 3812–3817 (2009).
17. Rosa, A. & Everaers, R. Ring polymers in the melt state: the physics of crumpling. *Phys. Rev. Lett.* **112**, 118302 (2014).
18. Vettorel, T., Grosberg, A.Y. & Kremer, K. Statistics of polymer rings in the melt: a numerical simulation study. *Phys. Biol.* **6**, 025013 (2009).

19. Naumova, N. *et al.* Organization of the mitotic chromosome. *Science* **14**, 953 (2013).
20. Cook, P.R. & Marenduzzo, D. Entropic organization of interphase chromosomes. *J. Cell Biol.* **186**, 825–834 (2009).
21. Jerabek, H. & Heermann, D.W. Expression-dependent folding of interphase chromatin. *PLoS One* **7**, e37525 (2012).
22. Kalhor, R., Tjong, H., Jayathilaka, N., Alber, F. & Chen, L. Genome architectures revealed by tethered chromosome conformation capture and population-based modeling. *Nat. Biotechnol.* **30**, 90–98 (2011).
23. Tjong, H. *et al.* Population-based 3D genome structure analysis reveals driving forces in spatial genome organization. *Proc. Natl. Acad. Sci. USA* **113**, E1663–E1672 (2016).
24. Paulsen, J. *et al.* Chrom3D: three-dimensional genome modeling from Hi-C and nuclear lamin-genome contacts. *Genome Biol.* **18**, 21 (2017).
25. Perovanovic, J. *et al.* Laminopathies disrupt epigenomic developmental programs and cell fate. *Sci. Transl. Med.* **8**, 335ra58 (2016).
26. Cremer, T. & Cremer, C. Chromosome territories, nuclear architecture and gene regulation in mammalian cells. *Nat. Rev. Genet* **2**, 292–301 (2001).
27. Rao, S.S. *et al.* A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* **159**, 1665–1680 (2014).
28. Kind, J. *et al.* Genome-wide maps of nuclear lamina interactions in single human cells. *Cell* **163**, 134–147 (2015).
29. Lund, E. *et al.* Lamin A/C-promoter interactions specify chromatin state-dependent transcription outcomes. *Genome Res.* **23**, 1580–1589 (2013).
30. Oldenburg, A. *et al.* A lipodystrophy-causing lamin A mutant alters conformation and epigenetic regulation of the anti-adipogenic MIR335 locus. *J. Cell Biol.* **216**, 2731–2743 (2017).
31. Kind, J. *et al.* Single-cell dynamics of genome-nuclear lamina interactions. *Cell* **153**, 178–192 (2013).
32. Stevens, T.J. *et al.* 3D structures of individual mammalian genomes studied by single-cell Hi-C. *Nature* **544**, 59–64 (2017).
33. Garcia-Nieto, P.E. *et al.* Carcinogen susceptibility is regulated by genome architecture and predicts cancer mutagenesis. *EMBO J.* **36**, 2829–2843 (2017).
34. Bau, D. *et al.* The three-dimensional folding of the alpha-globin gene domain reveals formation of chromatin globules. *Nat. Struct. Mol. Biol.* **18**, 107–114 (2011).
35. Imakaev, M. *et al.* Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat. Methods* **9**, 999–1003 (2012).
36. Serra, F. *et al.* Restraint-based three-dimensional modeling of genomes and genomic domains. *FEBS Lett.* **589**, 2987–2995 (2015).
37. Szalaj, P. *et al.* An integrated 3-dimensional genome modeling engine for data-driven simulation of spatial genome organization. *Genome Res.* **26**, 1697–1709 (2016).
38. Russel, D. *et al.* Putting the pieces together: integrative modeling platform software for structure determination of macromolecular assemblies. *PLoS Biol.* **10**, e1001244 (2012).
39. Bau, D. & Marti-Renom, M.A. Genome structure determination via 3C-based data integration by the Integrative Modeling Platform. *Methods* **58**, 300–306 (2012).
40. Serra, F. *et al.* Automatic analysis and 3D-modelling of Hi-C data using TADbit reveals structural features of the fly chromatin colors. *PLoS Comput. Biol.* **13**, e1005665 (2017).
41. Duan, Z. *et al.* A three-dimensional model of the yeast genome. *Nature* **465**, 363–367 (2010).
42. Li, Q. *et al.* The three-dimensional genome organization of *Drosophila melanogaster* through data integration. *Genome Biol.* **18**, 145 (2017).
43. Dai, C. *et al.* Mining 3D genome structure populations identifies major factors governing the stability of regulatory communities. *Nat. Commun.* **7**, 11549 (2016).
44. Lund, E.G., Oldenburg, A.R. & Collas, P. Enriched Domain Detector: a program for detection of wide genomic enrichment domains robust against local variations. *Nucleic Acids Res.* **42**, e92 (2014).
45. Durand, N.C. *et al.* Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. *Cell Syst.* **3**, 95–98 (2016).
46. Gundersen, S. *et al.* Identifying elemental genomic track types and representing them uniformly. *BMC Bioinformatics* **12**, 494 (2011).
47. Yang, Z. *et al.* UCSF Chimera, MODELLER, and IMP: an integrated modeling system. *J. Struct. Biol.* **179**, 269–278 (2012).
48. de Nooijer, S., Wellink, J., Mulder, B. & Bisseling, T. Non-specific interactions are sufficient to explain the position of heterochromatic chromocenters and nucleoli in interphase nuclei. *Nucleic Acids Res.* **37**, 3558–3568 (2009).
49. Lesne, A., Riposo, J., Roger, P., Cournac, A. & Mozziconacci, J. 3D genome reconstruction from chromosomal contacts. *Nat. Methods* **11**, 1141–1143 (2014).
50. Rousseau, M., Fraser, J., Ferraiuolo, M.A., Dostie, J. & Blanchette, M. Three-dimensional modeling of chromatin structure from interaction frequency data using Markov chain Monte Carlo sampling. *BMC Bioinformatics* **12**, 414–412 (2011).
51. Bau, D. & Marti-Renom, M.A. Structure determination of genomic domains by satisfaction of spatial restraints. *Chromosome Res.* **19**, 25–35 (2011).
52. Mifsud, B. *et al.* GOTHiC, a probabilistic model to resolve complex biases and to identify real interactions in Hi-C data. *PLoS One* **12**, e0174744 (2017).
53. Heinz, S. *et al.* Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Mol. Cell* **38**, 576–589 (2010).
54. Consortium, E.P. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**, 57–74 (2012).
55. Quinlan, A.R. & Hall, I.M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).