Chapter 4

Analysis, Modeling, and Visualization of Chromosome Conformation Capture Experiments

Marco Di Stefano, David Castillo, François Serra, Irene Farabella, Mike N. Goodstadt, and Marc A. Marti-Renom

Abstract

Chromatin Conformation Capture techniques have unveiled several layers of chromosome organization such as the segregation in compartments, the folding in topologically associating domains (TADs), and site-specific looping interactions. The discovery of this genome hierarchical organization emerged from the computational analysis of chromatin capture data. With the increasing availability of such data, automatic pipelines for the robust comparison, grouping, and classification of multiple experiments are needed. Here we present a pipeline based on the TADbit framework that emphasizes reproducibility, automation, quality check, and statistical robustness. This comprehensive modular pipeline covers all the steps from the sequencing products to the visualization of reconstructed 3D models of the chromatin.

Key words Chromosome Conformation Capture data analysis, Read mapping, Interaction filtering, Matrix normalization, Detection of compartments, Detection of TADs, 3D modeling, 3D visualization

1 Introduction

Genome-wide Chromosome Conformation Capture (3C) technologies [1] fostered a huge improvement in the field of genome structural biology [2]. The outcome of these techniques is a set of DNA–DNA frequency interactions reflecting the spatial proximity between pairs of chromatin regions [3, 4].

The applications of 3C technologies to many organisms and cell lines shed light into the hierarchical genome structural organization. In particular, Hi-C confirmed the existence of chromosome territories [5] initially characterized by imaging [6–8] and unveiled the partition of chromosomes into active and inactive compartments [5], into topologically associating domains (TADs) [9, 10],

Marco Di Stefano and David Castillo contributed equally to this work.

Beatrice Bodega and Chiara Lanzuolo (eds.), *Capturing Chromosome Conformation: Methods and Protocols*, Methods in Molecular Biology, vol. 2157, https://doi.org/10.1007/978-1-0716-0664-3_4,
© Springer Science+Business Media, LLC, part of Springer Nature 2021

and in chromatin loops [11], which facilitate interactions between genes and regulatory elements [12]. However, prior to the detection of these structural features, the data from a Hi-C experiment has to be cleaned, normalized, and quantitatively characterized by several steps of bioinformatic analysis. For this purpose, we developed TADbit (https://github.com/3DGenomes/TADbit), a comprehensive Python framework, to perform the analysis of Hi-C interaction datasets covering all the steps from aligning the sequenced reads (paired-end reads) up to the inference and analysis of three-dimensional (3D) models. Specifically, TADbit can perform (1) Hi-C-specific quality control of the reads, (2) mapping to the corresponding reference genome, (3) filtering of artifactual reads, (4) normalization of experimental biases, (5) generation of binned interaction maps, (6) statistical analysis of the differences and consistencies between experiments, (7) identification and comparison of the structural features in the interaction maps, (8) generation and analysis of 3D models using restraint-based methods, and (9) visualization of the 3D models using the companion visualizer TADkit. Here, we present a hands-on protocol for the analysis of any Hi-C datasets with minimal previous knowledge of bioinformatics, including an exhaustive description of each methodological step and a detailed supplementary notebook (https://github.com/3DGenomes/MethodsMolBiol).

2 Materials

2.1 Input Experimental Data

The complete TADbit pipeline has been applied to 5C [13], dilution Hi-C [14, 15], in situ Hi-C [16], and Promoter Capture Hi-C (https://doi.org/10.1101/400291). In this chapter, we will specifically refer to Hi-C experiments, for which we also provide a set of notes and suggestions to help to design the experiment (Materials-1-Design of the HiC experiment). Additionally, other types of 3C-based datasets can be analyzed with ad hoc changes in the parameter choice, mainly after the filtering of artifactual reads. Chromatin feature tracks (such as ChIP-seq, methylation, GC content, chromHMM, or RNA-seq, among others) can be loaded and used to label the genomic structural domains. Information as the nuclear size and the chromatin compaction can be used as additional parameters for the generation of 3D models.

2.2 Hardware Requirements and Performances

The requirements and performances listed below are for a Hi-C experiment of about 200 million reads binned at 50 kilobases (kb) on a mammalian genome of about 2–3 gigabases (Gb):

- Random-access memory (RAM): it is especially important when loading matrices at high resolution. About 32 gigabytes (GB) of RAM should be sufficient.

- Disk space: the data processing occupies around 500 GB per experiment (temporary storage). Some intermediate files are compressed or erased after processing; thus, the storage of the final output files occupies around 50 GB per experiment (long-term storage).
- Calculation (CPU) time: an 8-core computer can perform the data processing in about 1 day. The time for 3D model generation and analysis depends on the size of the considered region and the specific calculations required.

2.3 Software

For data analysis:

- TADbit (https://github.com/3DGenomes/TADbit) [15].
- SciPy (https://www.scipy.org) [17].
- Matplotlib (https://matplotlib.org) [18].
- IMP (https://integrativemodeling.org) [19].
- SAMtools (http://samtools.sourceforge.net) [20].
- SRAtools (https://github.com/ncbi/sra-tools).
- GEM (http://algorithms.cnag.cat/wiki/The_GEM_library) [21].
- OneD (https://github.com/qenvio/dryhic) [22].
- DSRC (http://sun.aei.polsl.pl/dsrc) [23].

For 3D model visualization:

- TADkit (https://github.com/3DGenomes/TADkit).

For more details on hardware requirement and software installation, see the notebook Materials-2-Preparing your computer for the Hi-C data analysis.

3 Methods

Here, we provide a detailed explanation of TADbit framework for Chromosome Conformation Capture dataset analysis. The task of each processing step is described, and the core functions needed to run the corresponding TADbit commands are provided in code blocks. In these blocks, the input values to functions are in italic, strings or lists of strings in green, numbers or list of numbers in red, and Python constants in orange (Table 1). The complete code to perform all the steps of the pipeline is provided in the supplementary notebooks (https://github.com/3DGenomes/MethodsMolBiol). In this chapter, for demonstration purposes, the pipeline is applied on two experiments carried out on two mouse

Table 1
List and description of all the variable used in the code blocks

Variable name	Description
bam_file	Path to the compressed and indexed bam file containing the intersection of the pair of mapped reads file after filtering
counts_per_column	List with the sum of counts per column (bin) of the matrix
Crm	Python object containing one or more experiments
Cutoff	Distance threshold (in nm) to determine if two particles interact. The default value is twice the particle size
destination_dir	Path to store the results of the function
end_bin	Number of the last bin in the matrix for the region we want to model
Exp	Python object containing the experiment and its description
factor_eq_positions	Factor to define the percentage of equivalent positions to be considered in the clustering
file_reads1,file_reads2	Paths to the mapped reads files
gc_content_per_column	List with GC content ratio per column
hic_data, hic_data1, hic_data2	Python objects containing the Hi-C matrix and its properties
input_fastq	Path to the FASTQ file containing the read-ends
list_of_chromosomes	List of chromosomes names
list_of_cutoffs	List of cutoffs in number of particles
list_of_experiments	List of the experiment names included in the chromosome object
list_of_filters_numbers	List of numbers of filter's categories
Mappability	List with the mappability score per column (bin)
Masked	List of paths of the files containing the reads filtered
max_fragment_size	Maximum fragment size of reads considered as being too long in the filtering
max_molecule_length	Maximum distance in nucleotides of two read-ends that are coming from two different REs. It is used to define the extra-dangling-ends category
minimum_distance_to_re	Minimum distance of the reads to a RE cut site
Models	Python object containing the structural models and their properties
nbr_cpus	Number of cpu cores to use
nbr_models	Number of models to compute
nbr_models_keep	Number of best models to keep
number_ev_to_compare	Number of top eigenvectors to compare
number_of_iterations	Number of iterations for the ICE algorithm

(continued)

Table 1
(continued)

Variable name	Description
number_of_re_sites_per_column	List with the numbers of restriction sites per column (bin)
number_of_reads	Number of reads to use in the function
optimal_params	Python dictionary containing the list of best parameters after the optimization
path_to_gem_index	Path to the indexed GEM file
read_length	Read length in nucleotides
reads_file	Path to the file containing the intersection of the pairs of mapped read-ends
Resolution	Resolution in base-pairs to use for binning the Hi-C matrix
restriction_enzyme	Name of the restriction enzyme used in the Hi-C experiment
rich_in_A_marker	Path to a BED or BEDGraph file with a list of genes or active epigenetic marks
start_bin	Number of the first bin in the matrix for the region we want to model
(start_lowfreq, end_lowfreq, step_lowfreq)	Range of *lowfreq values* from *start_lowfreq* to *end_lowfreq* in steps of *step_lowfreq*
(start_maxdist, end_maxdist, step_maxdist)	Range of maxdist values from *start_maxdist* to *end_maxdist* in steps of *step_maxdist*
(start_upfreq, end_upfreq, step_upfreq)	Range of upfreq values from *start_upfreq* to *end_upfreq* in steps of *step_upfreq*
valid_reads_file	Path to the file containing the intersection of the pair of mapped reads file after filtering
win1_start,win1_end, win2_start,win2_end	Starts and ends of the windows of nucleotides to consider in the read for the iterative mapping strategy

cell types (B-cells and induced pluripotent stem cells (iPSC)) each analyzed in two replicas, which were down-sampled at 150 million (M) reads each (Accession number GSE53463) [16].

3.1 Data Management

3.1.1 Hi-C Data

FASTQ files, either input by the user or downloaded from public repositories (see Methods-1-Retrieve published Hi-C datasets), contain the sequenced paired-end reads to be mapped to a reference genome (*see* **Note 1**).

3.1.2 Other Data

A reference genome file contains the reference genome for the species of interest in a FASTA format. Several databases provide reference genomes for downloading, including the National Center for Biotechnology Information (NCBI) that we use in the online tutorial (Methods-2-Preparation of the reference genome). The following information is generated accordingly to the reference genome of interest:

- *Genome index file:* to speed up the mapping process, the FASTA file of the reference genome is converted into an indexed file used to efficiently map each read-end.
- *Positions of restriction enzyme (RE) cut-site file:* to assign each mapped read-end to a given RE fragment and filter for nonspecific products of the chromosome capture experiment.
- *GC content file:* to identify A and B compartments. It has been shown that active genomic regions tend to have higher GC content and are thus used to label type A compartments [24]. However, if available, cell-specific markers of activity (e.g., ChIP-seq for chromatin marks or RNA-seq data) should be used.
- *Mappability file*: it stores the mappability score, which is the probability that a genomic region produces uniquely mapped reads. This is one of the biases fitted by the OneD normalization procedure, and it is computed a priori for a given read length [25].

3.1.3 Hi-C-Specific FASTQ Quality Check

TADbit computes and plots several metrics from the FASTQ files to assess the quality of the Hi-C experiment and the sequencing (Fig. 1 and **Note 2**). Such metrics are:

- The PHRED score and the number of unidentified nucleotides (*Ns*) in the read sequence, which are routinely computed to address the quality of high-throughput sequencing experiments.
- The numbers of undigested and unligated RE sites per nucleotide along the read to assess the quality of the Hi-C experiment.
- The overall percentage of digested sites, which relates directly to the RE efficiency.
- The percentage of non-ligated digested (dangling-ends), which relates to the ligation efficiency.
- The percentage of read-ends with a ligation site, which is negatively correlated with the percentage of dangling-ends.

These quality measurements are performed on a subset of reads, usually *number_or_reads* = 1000000 (Methods-3-Hi-C quality check):

```
quality_plot(input_fastq, r_enz=restriction_enzyme,
nreads=number_of_reads)
```

3.2 Paired-End Read Mapping

The first step in the analysis of a Hi-C experiment consists on uniquely mapping all reads in the input FASTQ files to a location on the reference genome. In TADbit this is achieved using the GEM mapper [21] (Fig. 2a). Currently, three mapping strategies are implemented in TADbit:

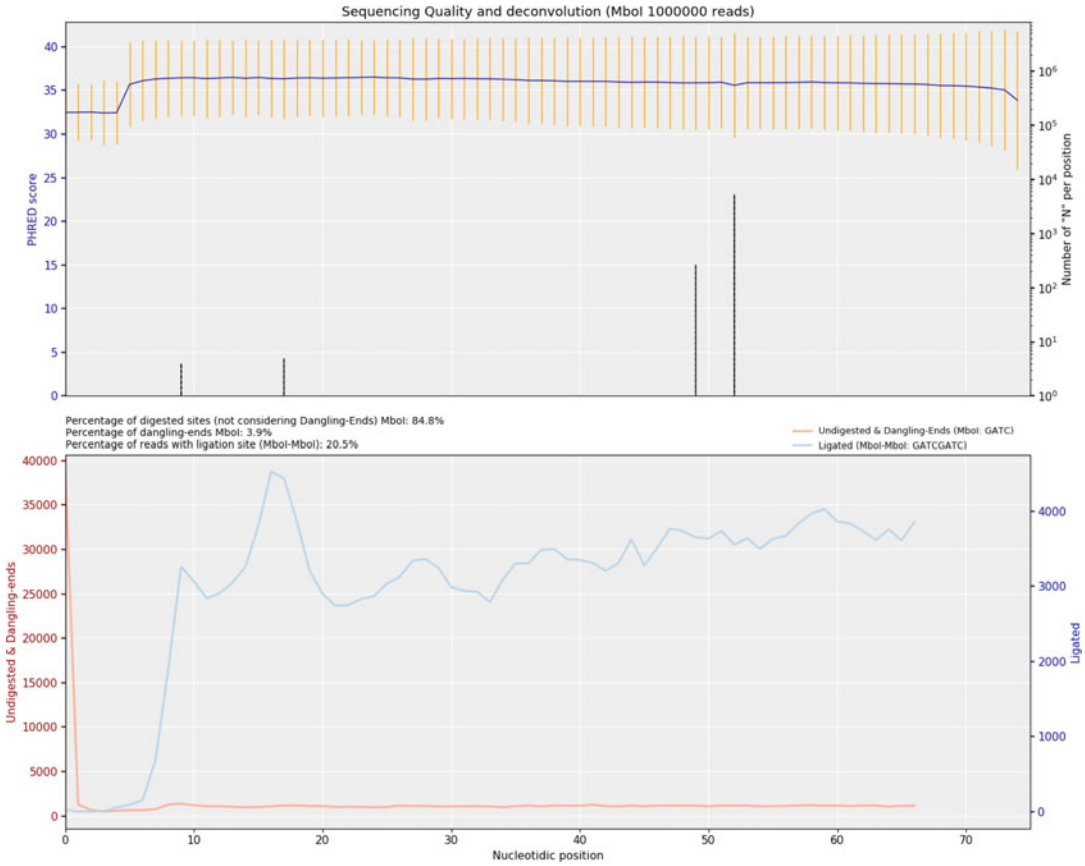

Fig. 1 Hi-C quality check plots. The upper plot shows two classic metrics of NGS experiments: the PHRED score (average in blue and standard deviation in yellow) and the proportion of *Ns* at each position of the read. The lower panel is specific to Hi-C experiments (in the example for MboI restriction enzyme). The two curves indicate the number of undigested sites (dangling-ends) in red, which is expected to be peaked at the beginning of the read, and the number of digested and ligated products in blue, which usually have a more homogeneous distribution along the read. For MboI, the patterns of undigested (GATC) and of digested and ligated (GATCGATC) sites are similar, but for other RE (e.g., HindIII), these patterns are distinct. Above the graph, three quantities are displayed: the proportions of ligated sites (84.8%), undigested sites (dangling-ends) (3.9%), and reads carrying a digestion site (20.5%). Note that TADbit can perform the same analysis in experiments where multiple REs are used at the same time

- *Full-length mapping*: The mapping of the read-end is attempted once (single iteration) taking into account the full length of the read. Unmapped read-ends are discarded.

```
full_mapping(gem_index_path=path_to_gem_index,
             out_map_dir=destination_dir, fastq_path=input_fastq,
             frag_map=False)
```

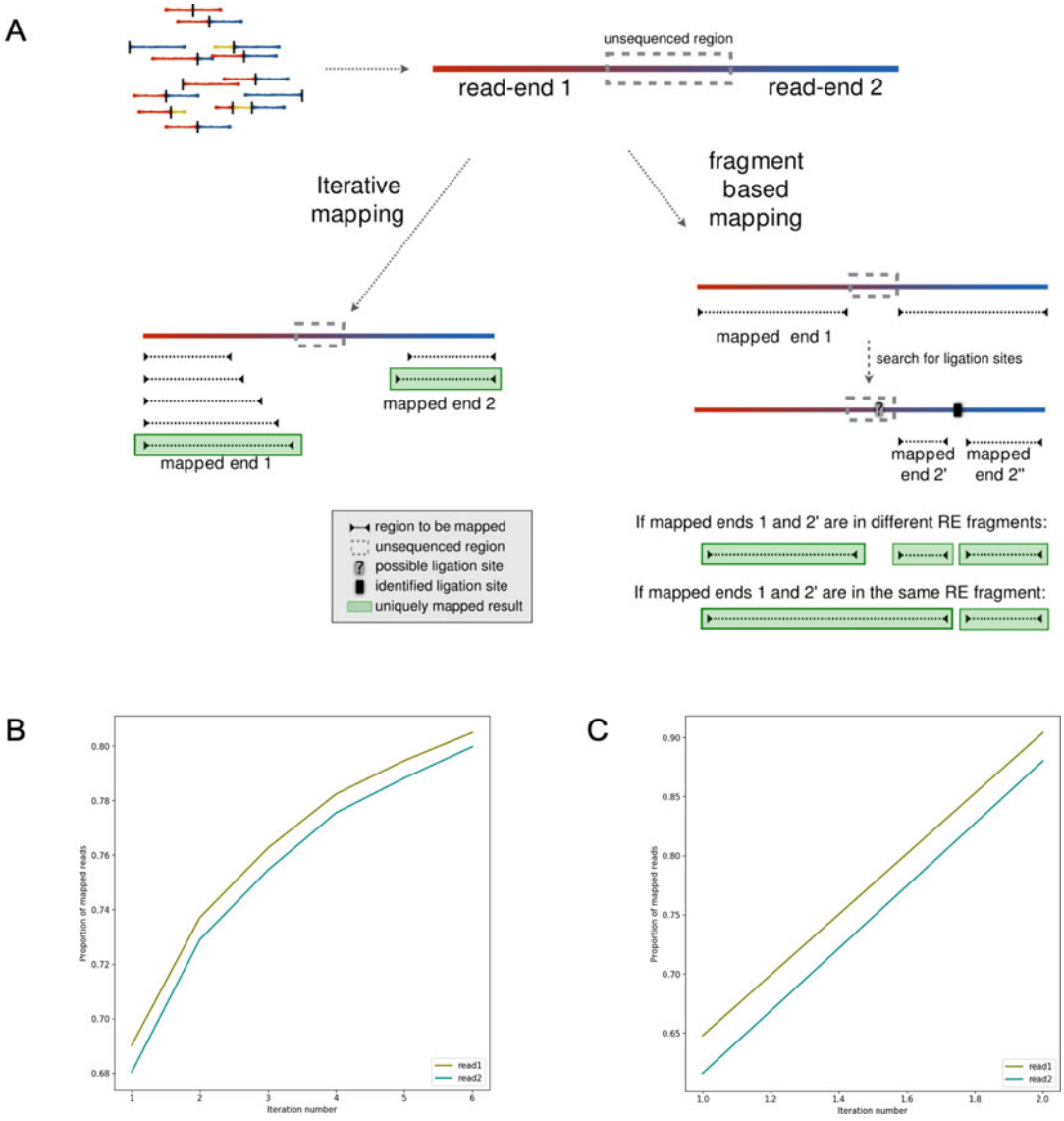

Fig. 2 Mapping of paired-end reads. (**a**) Schematic cartoons of the iterative mapping *(left)* and the fragment-based mapping *(right)* approaches. (**b** and **c**) The proportion of the read-ends mapped in each round of the iterative mapping (**b**, here 6 steps are shown) and of the two rounds of the fragment-based mapping, that is, full and fragment-based iterations (**c**)

- *Iterative mapping*: The mapping of the read-end is attempted with iteratively increasingly larger reads. Therefore, to use this option the user has to define the window parameter as a set of different ranges of nucleotide indexes all starting from 1 and of increasing sizes. Usually (in the human genome) the first section of the read-end goes from 1 to 25 bp and is followed by ranges that are incrementally extended by 5 bp ((1, 30), (1, 35), etc.).

The mapping procedure is applied iteratively until the read is uniquely mapped using the ranges defined in the set [26]. Note that TADbit accepts any combination of window ranges.

```
full_mapping(gem_index_path=path_to_gem_index,
             out_map_dir=destination_dir, fastq_path=input_fastq,
             frag_map=False,
             windows=((win1_start,win1_end),(win2_start,win2_end),
             ...))
```

- *Fragment-based mapping*: This procedure consists of two iteration steps per read-end: (1) the full-length is attempted, and (2) if the read-end is unmapped and contains ligation sites, it is split at the ligation sites, and the split sequences are separately mapped.

```
full_mapping(gem_index_path=path_to_gem_index,
             out_map_dir=destination_dir, fastq_path=input_fastq,
             r_enz=restriction_enzyme_name, frag_map=True)
```

A detailed account of the commands to perform the mapping step is provided in the notebook Methods-4-Mapping. A good metric for assessing the quality of the mapping procedure is the proportion of reads that have been uniquely mapped to the reference genome (Fig. 2b). To render the number of reads mapped at each iteration of the mapping, one can use the command:

```
plot_iterative_mapping(file_reads1, file_reads2, number_of_reads)
```

3.3 Intersection of Paired-End Reads

After mapping, a browser extensible data (BED) file is created with the reads that have been uniquely mapped on both ends. This contains the genomic coordinates of each pair of mapped read-ends as well as their position relative to the closest RE site. Note that for reliably assembled genomes, more than 80% of the reads are typically mapped for any of the two ends and more than 60% on both ends.

```
get_intersection(file_reads1, file_reads2, reads_file)
```

3.3.1 Descriptive Statistics after the Intersection of the Pair-End Reads

- *Interaction count vs. genomic separation*. Given the intrinsic polymeric structure of chromatin, the number of captured interactions is expected to decay as the genomic separation between the interacting sequences increases. In mammals, this decay typically follows a power law with an exponent around −1 in the range between hundreds of kb and tens of Mb [5] (Fig. 3a).

```
plot_distance_vs_interactions(reads_file, resolution=resolution)
```

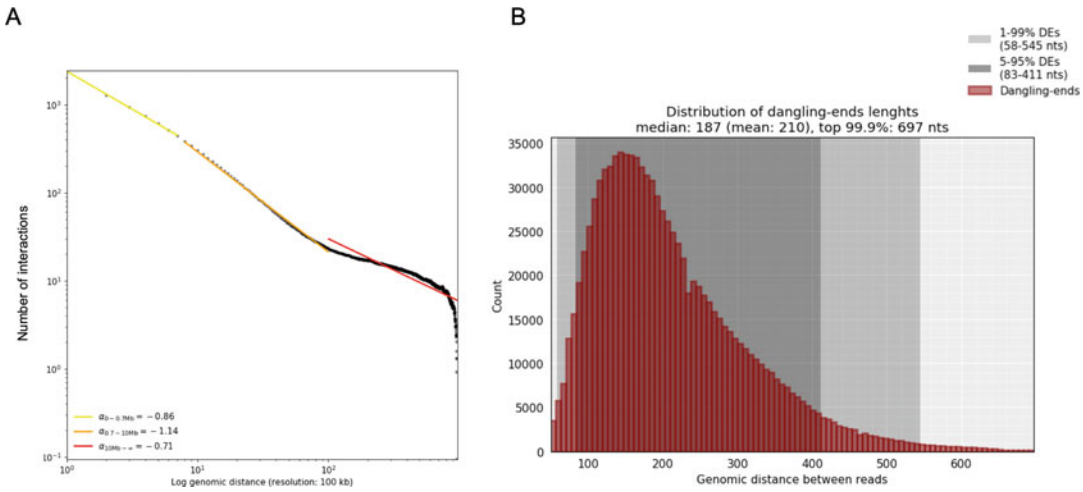

Fig. 3 Descriptive statistics of the mapped read-ends. (**a**) Decay of the number of Hi-C interactions with respect to the genomic distance between interacting *loci*. (**b**) The distribution of RE fragment lengths is estimated considering fragments mapped in a single RE fragment and in *facing* orientation (see dangling-ends in Subheading 3.4). Although the dangling-ends are expected to be shorter than most fragments in the library, we use their average length as a conservative estimate of the mapped fragments for the application of filters

- *Coverage per bin*. The genome is partitioned into regions of the same size (called "bins"), and the number of mapped reads per bin (i.e., coverage) is computed. In an ideal situation, the coverage is expected to be homogeneous across the genome. However, variations in the distribution of GC content, mappability, or the number of RE sites along the genome can cause heterogeneity. The source of such biases will be corrected during the normalization of the interaction matrices (Subheading 3.6).

```
plot_genomic_distribution(reads_file, resolution=resolution,
show=True)
```

- *Sequenced DNA fragment size*. One can measure the genomic length of the sequenced DNA strands using only the reads that were digested, but not ligated and therefore belong to the same DNA linear segment between two consecutive RE cut sites (e.g., the two paired-ends have been mapped within the same RE fragment). These reads (dangling-ends) are mapped in *facing* orientation (Fig. 3b and **Note 3**).

```
fragment_size(reads_file, show=True, nreads=number_of_reads)
```

For details see Methods-5-Parsing mapped reads.

3.4 Filtering of Mapped Reads

At this stage, the intersection file contains uniquely mapped paired-end reads of DNA sequences, including those pairs with no relevant structural information. After the filtering step, typically around

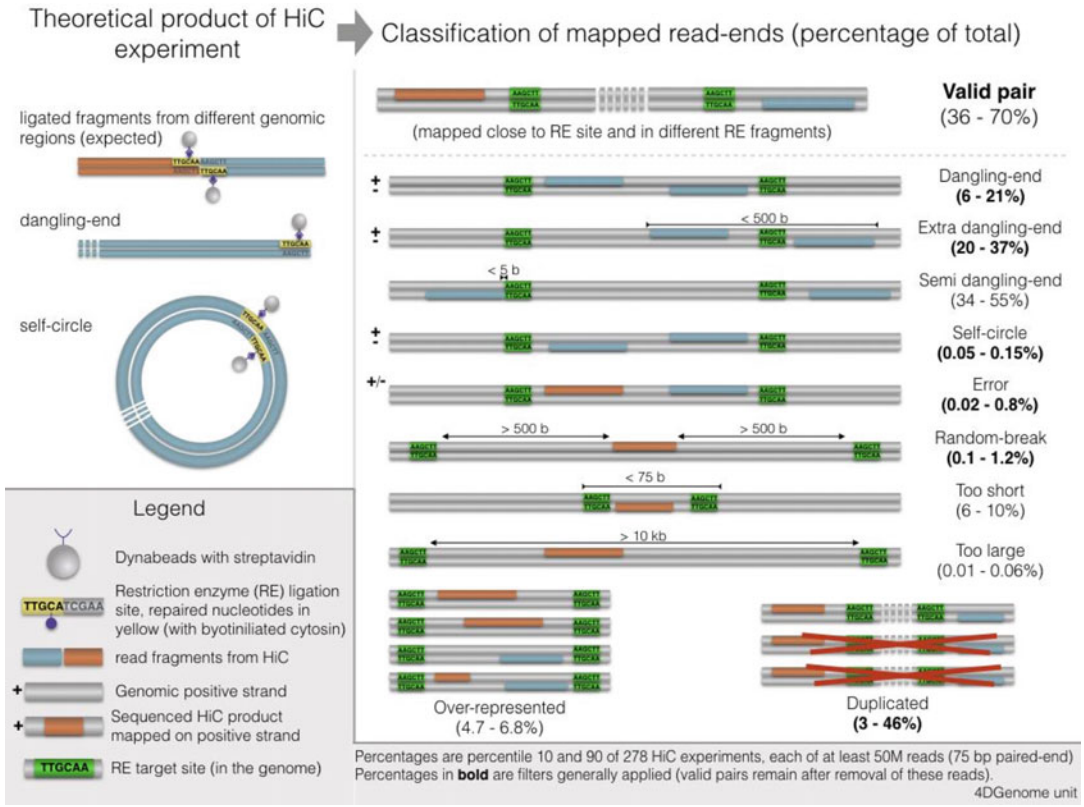

Fig. 4 Summary of the different products of a Hi-C experiment (top left) and the categories of mapped reads (right). Here, the proportions are estimated from the analysis of 278 Hi-C experiments produced by the 4DGenome unit of the CRG. All experiments were performed using 75 bp paired-end sequencing. Fair quality experiments would have proportions falling within the ranges displayed here

40–70% of the pairs are considered informative (also called "valid pairs"). TADbit will classify the remaining (or discarded) mapped pairs into one (or several) of the following categories (Fig. 4 and **Note 4**):

- *Self-circle*: both read-ends are mapped to the same RE fragment in *opposed* orientation.
- *Dangling-end*: both read-ends are mapped to the same RE fragment in *facing* orientation.
- *Error*: both read-ends are mapped to the same RE fragment in the same orientation.
- *Extra dangling-end*: the read-ends are mapped to different RE fragments in *facing* orientation, but are close enough (< *max_molecule_length* bp) from the RE cut site to be considered part of adjacent RE fragments that were not separated by digestion. The *max_molecule_length* parameter can be inferred from the *fragment_size* function previously detailed.

- *Too close from RE sites (or semi-dangling-end)*: the start position of one of the read-end is too close (5 bp by default) from the RE cutting site (*see* **Note 5**).
- *Too short*: one of the read-ends is mapped to RE fragments of less than 75 bp. These are removed since there is ambiguity on where the read-end is mapped as it could also belong to any of the two neighboring RE fragments.
- *Too large*: the read-ends are mapped to long RE fragments (default: 100 kb, $P < 10^{-5}$ to occur in a randomized genome), and they likely represent poorly assembled or repetitive regions.
- *Overrepresented*: the read-ends coming from the top 0.5% most frequently detected RE fragments may represent PCR artifacts, random breaks, or genome assembly errors.
- *PCR artifacts or duplicated*: the combination of the start positions, mapped length, and strands of both read-ends is identical. In this case, only one copy is kept.
- *Random breaks*: the start position of one read-end is too far (>*minimum_distance_to_RE*) from the RE cut site. These are produced most probably by non-canonical enzyme activity or by random physical breakage of the chromatin. Note, that to filter all these types of fragments, the *minimum_distance_to_RE* parameter should be larger than the *maximum_fragment_length*.

Once TADbit has classified each read-end pair (*filter_reads* command below), one can define valid genomic interactions (valid pairs) by removing the paired-end reads belonging to a set of non-informative selected categories (*apply_filter* command).

```
masked = filter_reads(reads_file,
                      max_molecule_length=max_molecule_length,
                      max_frag_size=max_fragment_size,
                      min_dist_to_re=minimum_distance_to_re)

apply_filter(reads_file, valid_reads_file, masked,
             filters=list_of_filters_numbers)
```

For the detailed list of commands to perform the filtering step, see the notebook Methods-6-Filtering mapped reads.

In all the step performed so far, the mapped read-ends were stored in tab-separated value (TSV) files. At this stage, it is convenient for faster processing to convert the TSV file to binary alignment map (BAM) format:

```
bed2D_to_BAMhic(valid_reads_file, valid=True, outbam=bam_file)
```

3.5 Generation of the Raw Interaction Matrix

The interaction matrices are generated by partitioning the genome in *loci* of equal length (bins) and assigning each end of the read to its binned genomic location. We normally refer to this process as binning, which will define the resolution of the Hi-C matrix. Therefore, it is important to choose a suitable bin size (*see* **Note 6**).

3.5.1 Plotting the Interaction Matrix

Plotting and visually inspecting the interaction matrix is a key quality control of a Hi-C experiment. In TADbit, this is achieved by the *hic_map* function:

```
hic_map(hic_data, resolution=resolution, show=True)
```

In this view, TADbit also provides some statistics on the quality of these interactions as (1) the ratio between the intra-chromosome and the total number of interactions (i.e., cis-to-trans ratio), which is expected to be at least 50% in mammals, and (2) the three first eigenvectors of the matrix that summarize the principal structural features of the matrix (Fig. 5a and Subheading 3.7).

3.5.2 Filter Bins with Low Interaction Counts

Very low coverage bins are identified and filtered out from the data analysis (Fig. 5b). For example, these removed bins can contain no RE site (or no mappable region around its RE sites) and so result in zero valid pairs. However, those bins may still contain some valid pairs as the results of mapping errors or random breakage of the DNA before the cross-linking. It is also likely that very low coverage bins present a cis-to-trans ratio typical of random ligation events [26]. Such bins could also cause technical problems, as they will introduce a strong variability in most of the normalization strategies. To visualize the number of bins to be filtered, TADbit can produce a histogram plot using the *filter_columns* function (Methods-7-Bin filtering and normalization).

```
hic_data.filter_columns(draw_hist=True, by_mean=True)
```

3.6 Normalization

Interaction matrices derived from Hi-C experiments contain different types of biases [26, 27], which need to be removed (Fig. 5c). This process is referred to as normalization, which can be done within TADbit using different normalization procedures:

3.6.1 Normalization by Visibility

- *Iterative Correction and Eigenvector decomposition (ICE)*: ICE assumes equal experimental visibility of each bin and seeks iteratively for biases that equalize the sum of counts per bin in the matrix [26]. At each iteration, a new matrix is generated by dividing each cell by the product of the sum of counts in its row times the sum of counts in its column. The process converges to a matrix in which all bins have identical sums.

```
hic_data.normalize_hic(iterations=number_of_iterations)
```

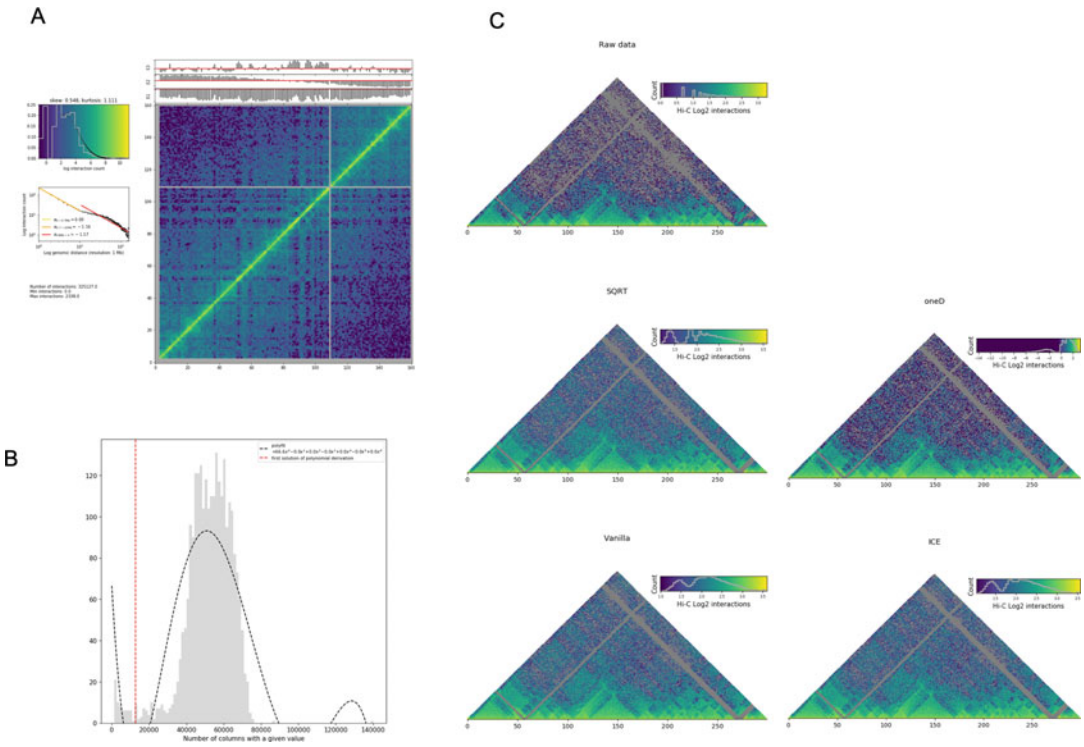

Fig. 5 Matrix binning and normalization. (**a**) The standard Hi-C map representation in TADbit also summarizes other relevant information. On the left, the histogram of the number of interactions (*top*), the plot of the interaction counts vs genomic distance (*middle*), and descriptive statistics (the sum, the minimum and the maximum of the pairwise interactions) (*bottom*) are shown. On the right, there is the binned interaction matrix represented in Log2 color scale and the first three eigenvectors of the matrix on top. (**b**) Histogram of the number of entries with non-zero values per bin. The histogram is expected to be bimodal. One peak is usually close to zero and indicates artifactual bins that are almost empty. The other peak appears at higher values and corresponds to the expected number of non-zero entries for informative bins. A polynomial function is fitted to this distribution in order to filter out bins falling into the peak with almost all the cells empty. (**c**) The Hi-C maps are shown in the normalized forms obtained with the different normalization algorithms

- *Vanilla coverage*: is a variation of the ICE where a single iteration is performed [5].

```
hic_data.normalize_hic(iterations=0)
```

- *Square root vanilla coverage*: is a variation of the vanilla coverage where each element in the matrix is divided by the square root of the product of sums of counts [11].

```
hic_data.normalize_hic(iterations=0, sqrt=True)
```

3.6.2 Normalization Via Individual Bias Estimation

- *OneD*: is based on fitting a non-linear model between the total amount of contacts per bin and the known biases, which are by default the GC content, the number of RE sites, and the

mappability [22].

```
biases = oneD(tot=counts_per_column, map=mappability,
              res=number_of_re_sites_per_column,
              cg=gc_content_per_column) hic_data.bias =
              list(biases)
```

The detailed list of command to normalize a Hi-C matrix is given in the notebook Methods-7-Bin filtering and normalization.

3.7 Call for Structural Features

3.7.1 Compartments

Previous analysis of Hi-C experiments [5, 11] showed that the genome is structurally organized in two major types of compartments named A and B, which are enriched with active and inactive chromatin, respectively. Various approaches exist to determine genomic sub-compartments using Hi-C data [5, 11, 26, 27]. TADbit uses the change in sign in the components of the first eigenvector (calculated on the autocorrelation matrix of the normalized Hi-C matrix) to detect boundaries between A and B compartments [5] (Fig. 6a and **Note 7**). To define the compartment to the corresponding type, TADbit can use various type of genetic or epigenetic activity markers such as GC content (Subheading 3.1.2 and Fig. 6a).

```
hic_data.find_compartments(show_compartment_labels=True, show=True,
                           crms=list_of_chromosomes,
                           rich_in_A=rich_in_A_marker)
```

3.7.2 Topologically Associating Domains

Topologically Associating Domains (TADs) constitute the next level of organization of the genome structure. Several TAD callers exist based on a variety of metrics and statistics. Briefly they consist of two types: (1) tools that detect TAD borders (also known as breakpoints) assuming that the whole genome consists of a succession of TADs (e.g., methods based on the insulation score [28]) and (2) tools that identify TADs as denser interacting regions of the genome (e.g., methods using the directionality index [9]). Importantly, the choice of the suitable TAD caller is mainly dependent on the hypothesis to be tested. However, it has been shown that the results among the most used tools are overall consistent [29]. TADbit implements several TAD callers giving the user the possibility to choose the one that best fits its needs. Thus, it allows the comparison between different strategies (Fig. 6b–d). The available TAD callers in TADbit are:

- *TADbit*: is the TAD caller algorithm after which TADbit framework was named. It is a breakpoint detection algorithm that defines TADs within a chromosome (or genomic region)

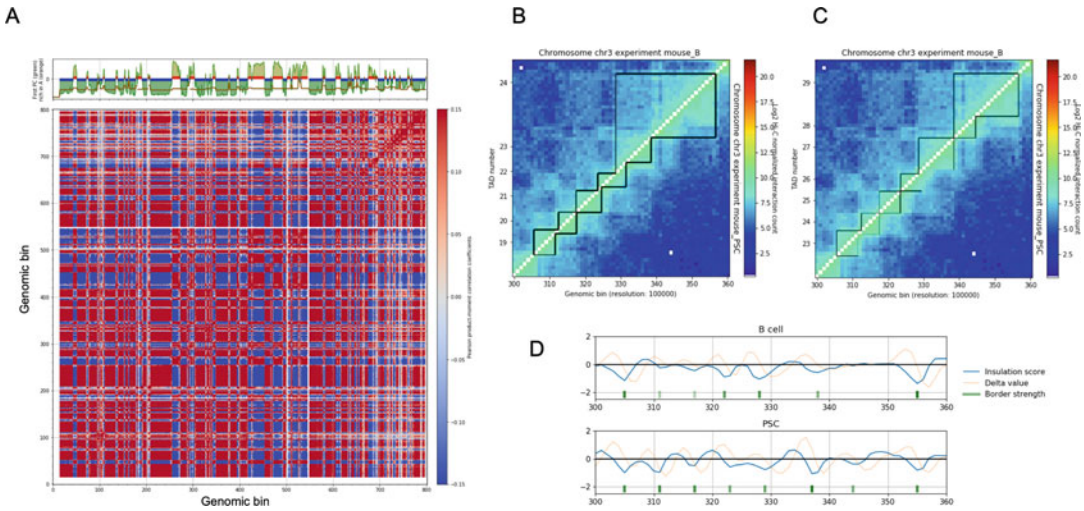

Fig. 6 Detection of the structural features in Hi-C interaction matrices. (**a**) Autocorrelation matrix of a OneD normalized matrix of mouse chromosome 3 at 200 kb resolution. The following steps were performed to obtain the autocorrelated matrix: (1) the raw Hi-C matrix was normalized with the OneD algorithm to correct for experimental and genomic biases, (2) the decay of the number of interactions with the genomic distance was corrected by dividing each cell by the average interactions in its diagonal, and (3) the obtained matrix was used to compute each element (a_{ij}) of the autocorrelation matrix as the Pearson correlation of the *i*-th row and the *j*-th column. On the top panel, we show the first eigenvector of the autocorrelation matrix, the average GC content in each compartment, and the prediction of compartment labeling (based on the average GC content). The changes in sign of the eigenvector components mark the boundaries between compartments (*green*), but the assignment of each compartment to type A or B is based on the average GC content with A compartments on top of the red bands (high CG content) and B compartments on top of blue bands (low GC content). (**b**) and (**c**) TAD border detections obtained using tadbit (**b**) and TopDom (**c**) in a region of mouse chromosome 3 for two different cell types: B cells (*top left*) and iPS cells (*bottom right*). The width of the partitioning lines is proportional to the confidence of the predicted border. (**d**) Insulation score in blue, the delta inferred from it in orange, and the prediction of TAD border in scales of green depending on the strength of the border. The two panels correspond to the same region of chromosome 3 and the same cell types of panels (**b**) and (**c**), respectively

under the BIC-penalized (Bayesian Information Criterion) likelihood [15]. The detected TAD borders are associated with a score (from 1 to 10) quantifying the accuracy of the border detection (Fig. 6b).

```
crm.find_tad(list_of_experiments)
```

- *TopDom*: identifies TAD borders based on the assumption that contact frequencies between regions upstream and downstream of a border are lower than those between two regions within a TAD. The algorithm only depends on a single parameter corresponding to the window size [30]. The algorithm provides a measure (from 0 to 10) of confidence on the accuracy of the border detection (Fig. 6c).

```
crm.find_tad(list_of_experiments, use_topdom=True)
```

- *Insulation score*: can be used to build an insulation profile of the genome and, with a simple transformation, to identify TAD borders [28, 31] (Fig. 6d).

```
insulation_values = insulation_score(input_matrix, locus_size,
delta_span)
borders = insulation_to_borders(insulation_values,
minimum_strength)
```

TAD borders called using *TADbit* or *TopDom* can be saved to text files and visualized using the *visualize* function. More details are in notebook Methods-8-Compartments and TADs detection.

```
crm.visualize(list_of_experiments, paint_tads=True)
```

3.8 Comparison between Hi-C Experiments

An important aspect of the Hi-C data analysis is the comparison of different experiments (e.g., technical or biological replicas, different time points or conditions, etc.) that can unveil fundamental biological insights. The comparison can be carried out at different levels of the data processing from the interaction matrices to the structural features or between 3D models of the chromatin (Methods-9-Compare and merge Hi-C experiments, Subheading 3.9).

3.8.1 Comparison Between Interaction Matrices

In TADbit the user can monitor several quantities to compare Hi-C matrices:

- The Spearman rank correlation of the matrix diagonals at increasing genomic distances and the stratum-adjusted correlation coefficient (SCC) score [32]. Both quantities range from −1 (anti-) to 1 (direct correlation) (Fig. 7a).

```
spearmans, dists, scc, std = correlate_matrices(hic_data1,
                                                hic_data2, show=True,
                                                normalized=True)
```

- The correlation of the eigenvectors. Since the eigenvectors of a matrix capture its internal correlations [26], two matrices with highly correlation of eigenvectors are considered to have similar structure (Fig. 7b).

```
eig_correlate_matrices(hic_data1, hic_data2, show=True,
normalized=True)
```

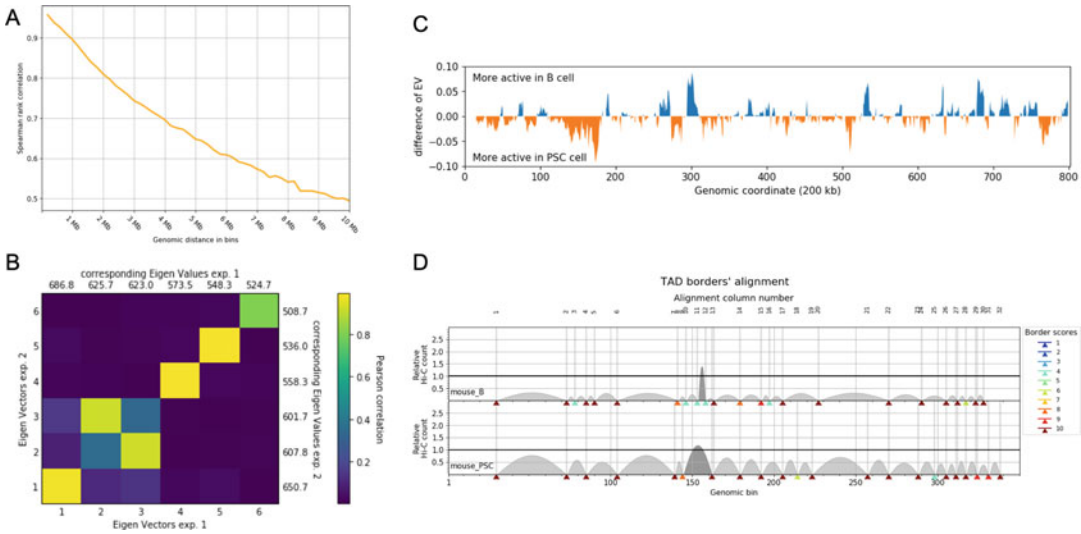

Fig. 7 Comparison between Hi-C experiments. (**a**) Spearman correlation coefficient between diagonals of the two replicates matrices in B cells. The profile in this plot is typical of highly similar matrices. (**b**) Pearson correlation of all combinations of the first six eigenvectors of these same two matrices. As in panel (**a**), these eigenvector correlations are typical of similar matrices. (**c**) The comparison of compartment partitions in B vs iPS cells is shown as the difference between the eigenvector of the autocorrelation matrix (Fig. 6a) computed in each of the two conditions. Blue peaks indicate increased activity in B cells and red peaks in iPS cells. (**d**) Alignment of TAD borders in B vs iPS cells. Each arc represents a TAD and each colored triangle a TAD border. The confidence score of each border is conveyed by the color scale from 1 (low in blue) to 10 (high confidence in red)

- The reproducibility score. Computed as in HiC-spector [33], it is also based on comparing eigenvectors. The reproducibility score ranges from 0 (low similarity) to 1 (identity).

```
reprod = get_reproducibility(hic_data1, hic_data2,
                             num_evec=number_ev_to_compare,
                             normalized=True)
```

For details see Methods-9-Compare and merge Hi-C experiments.

3.8.2 Comparison of the Structural Features

- *Compartments*. The eigenvector analysis performed to detect compartments can also be used to compare experiments. For instance, the simple difference between the first eigenvector (i.e., the one that captures the compartmentalization of the genome) of the different experiments allows detecting changes in the type of compartments between two different conditions (Fig. 7c).

- *TADs*. TAD border conservation across experiments can be used as an additional comparative measure. However, the assessment of co-occurrence of TAD borders in two different matrices is not trivial due to the border strength variability and their dynamic behavior under different conditions. For an effective TAD border comparison, TADbit allows the alignment of TAD borders using a reciprocal best hit strategy (Fig. 7d).

```
ali = crm.align_experiments(list_of_experiments)
```

For details on structural features comparison, see the notebook Methods-8-Compartments and TADs detection.

3.9 3D Modeling

In TADbit, 3D models of genomic regions can be generated via a restraint-based modeling approach [34], by transforming the normalized Hi-C interaction matrix (Subheading 6) into a set of spatial restraints that are satisfied using the Integrative Modeling Platform (IMP) [19]. Note, that TADbit provides an assessment score named Matrix Modeling Potential (MMP) score [35] to predict whether an interaction matrix can be used to produce reliable 3D models. It is recommended to assess such score before 3D models generation.

3.9.1 TADbit Modeling Approach

TADbit modeling approach is based on three main steps. These steps are briefly described [34]:

- *System representation*. Each bin of the matrix is described as a spherical particle of size proportional to its DNA content.
- *Scoring*. Each normalized Hi-C interaction count is Z-score transformed (Fig. 8a) and associated with a harmonic restraint. The simple rationale behind this transformation is that the more two particles interact, the stronger is the spring constant and the smaller is the equilibrium distance of the associated harmonics [34]. Briefly, this transformation depends on three parameters: *maxdist* (i.e., the maximal target distance between two non-interacting particles), as well as *upfreq* and *lowfreq*, which determine particle pairs that interact more than expected as well as those that do less than expected, respectively. Pairs of particles with a Z-score larger than the *upfreq* will be restrained to be close in 3D space, and particles with a Z-score smaller than *lowfreq* will be kept apart in 3D space. The sum of all the harmonic restraints is the so-called scoring function. The lower the *scoring function*, the closer the corresponding 3D structure is to the target pattern of interactions [34].
- *Sampling*. To find the 3D arrangement of the particles that better represent the input Hi-C interactions pattern, the scoring function is minimized using a Monte Carlo simulated annealing sampling protocol with the Metropolis criterion [34].

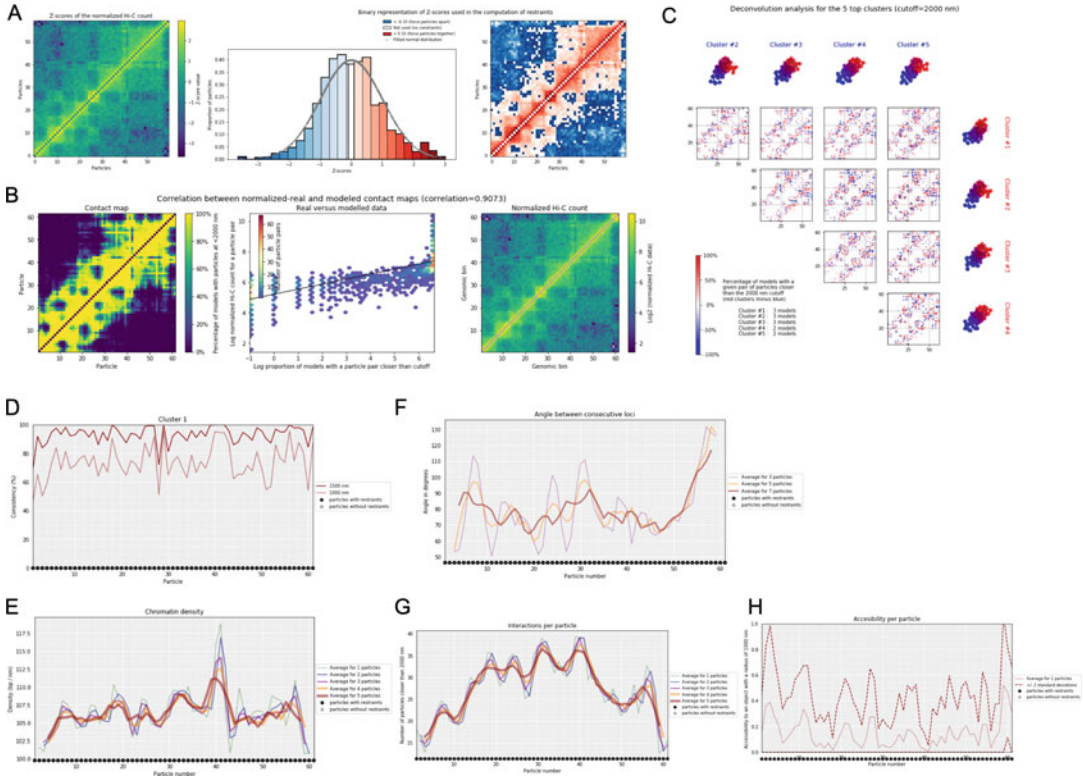

Fig. 8 3D model generation and analysis. (**a**) Interaction matrix for B cells (*left*) and its Z-score transformation (*right*). The central panel shows the distribution of the Z-score values divided into three categories: the pairs of *loci* that are restrained to be far in blue, unrestrained pairs in white, and pairs with higher interaction counts restrained to be close to each other in red. (**b**) Quantitative comparison between the contact map computed from the generated 3D models using a distance cutoff of 2000 nm at 100 kb resolution (*left*) and the original Hi-C interaction matrix of B cells (*right*). The central panel shows the scatter plot of the number of contacts observed in the simulated models (the virtual contact matrix) and the original Hi-C interaction matrix. The Spearman correlation coefficient between the two matrices is 0.91 in this specific case. (**c**) Deconvolution analysis generated ensembles of models, which are compared and clustered to identify local differences between subpopulations of 3D models. (**d**), (**e**), (**f**), (**g**), and (**h**) Structural measures from the 3D model ensemble. As discussed in the main text, they are the consistency, chromatin density, walking angles, number of interactions (*dcutoff* = 2000 nm), and accessibility, respectively

3.9.2 Parameter Optimization

In TADbit, the optimal parameters *(maxdist, lowfreq, upfreq)* are determined empirically via a grid search defined by the user. Other parameters used in the reconstruction of the model can also be optimized at the same time, like the distance cutoff that defines contact between particles. However, those can be ignored by the end-user and are less sensitive to the specifics of the Hi-C matrix experiment to be modeled. The optimization command can be run in parallel in different computers using in each case a predefined number of CPUs:

```
optimal_params = exp.optimal_imp_parameters(start=start_bin,
                                            end=end_bin, n_models=nbr_models,
                                            n_keep=nbr_models_keep,
                                            n_cpus=nbr_cpus,
                                            upfreq_range=(start_upfreq,
                                            end_upfreq, step_upfreq),
                                            lowfreq_range=(start_lowfreq,
                                            end_lowfreq, step_lowfreq),
                                            maxdist_range=(start_maxdist,
                                            end_maxdist, step_maxdist),
                                            dcutoff_range=list_of_cutoffs)
```

For each possible combination of the three parameters, TADbit will produce a set of models (*nbr_models*) from which it will keep the ones best satisfying the imposed restraints (top *nbr_models_keep*). The ratio between the numbers of kept and generated models is typically 20%. During the search of the optimal parameters, *nbr_models_keep* = 100 and *nbr_models* = 500 are the recommended values. For assessing the best combination of parameters, a contact matrix is computed from the best models choosing an optimal distance cutoff (*dcutoff*) and compared with the input normalized Hi-C matrix using the Spearman correlation coefficient. The results of the optimization grid search can be visualized as a 2D matrix:

```
optimal_params.plot_2d(show_best=5)
```

The best set of parameters, those associated with the highest Spearman correlation coefficient, can be retrieved and used for the next modeling step. For details, see notebook Methods-10-Modeling parameters optimization:

```
optimal_params.get_best_parameters_dict()
```

3.9.3 Model Building

The optimized modeling parameters are used next to build a more exhaustive *ensemble* of 3D models usually ten times larger than the one used in the optimization step (i.e., *nbr_models* = 5000 and *nbr_models_keep* = 1000). The larger number of models is meant to

represent the heterogeneity of chromatin conformations in the cell population of the Hi-C experiment; for details see the notebook Methods-11-3D Models production and analysis.

```
models = exp.model_region(start=start_bin, end=end_bin,
                          n_models=nbr_models, n_keep=nbr_models_keep,
                          n_cpus=nbr_cpus, config=optimal_params)
```

3.9.4 3D Model Assessment and Analysis

TADbit offers a set of functions for the evaluation and analysis of the generated ensemble of models. For example, one can:

- Visualize how the experimental data were converted into harmonic restraints (Fig. 8a).

```
models.zscore_plot()
```

- Verify the convergence of the Monte Carlo optimization step by checking that the IMP scoring function decays as a function of the Monte Carlo iteration and reaches a stable *plateau*.

```
models[0].objective_function(log=True)
```

- Compare a set of models with the original experimental data using Spearman correlation coefficient (Fig. 8b).

```
models.correlate_with_real_data(plot=True, cutoff=cutoff)
```

3.9.5 Subpopulations of Models

TADbit can measure similarities between the models in the generated ensemble via the structural alignment of each pair of models using a pairwise rigid-body superposition that minimizes their root-mean-square deviation (RMSD).

```
models.align_models(in_place=True)
```

Using the aligned models and clustering them based on 3D comparative metrics, the ensemble of models can be deconvolved into subpopulations of structures that might represent the variability in the subpopulations of cells in the experimental sample (Fig. 8c).

```
models.deconvolve(fact=factor_eq_positions, dcutoff=cutoff)
```

This analysis generates differential contact maps between each pair of clusters and allows the detection of shared or exclusive sets of contacts within the ensemble.

Alternatively, it is possible to cluster the model ensemble using other measures as the RMSD, distance-RMSD, number of equivalent positions, or a combination of all the above measures. Each cluster will represent a given subpopulation of chromatin structures.

```
models.cluster_models(fact=factor_eq_positions, dcutoff=cutoff)
```

3.9.6 Quantitative Characterization of the 3D Models

TADbit also provides a set of measures averaged over the models (or in a given subset of models such as those in a cluster) to extract useful biological insights (Fig. 8d–h). Next, we briefly describe such measures:

- The consistency score (Fig. 8d) is the percentage of models that have a given particle superimposed within a predefined distance cutoff. The lower is the consistency value, the less deterministic (more variable) are the models in the corresponding position. The consistency score should be measured independently for each cluster.

```
models.model_consistency()
```

- The density (or local compactness, Fig. 8e) is the number of base-pairs per nanometer (nm) and is computed as the ratio of the number of base-pairs per particle and the distance between consecutive particles.

```
models.density_plot()
```

- The walking angle (Fig. 8f) is the angle between triplets of contiguous particles. The higher are these values, the straighter are the models.

```
models.walking_angle()
```

- The number of interactions (Fig. 8g) measures contacts made by each particle within a given cutoff distance.

```
models.interactions(cutoff=cutoff)
```

- The accessibility score (Fig. 8h) measures how accessible a particle is to a spherical object of a predefined radius. Note that the accessibility is measured only for the internal space of the models as the 3D neighborhood of the modeled region is normally not determined.

```
models.accessibility()
```

3.10 3D Visualization

The 3D models generated by TADbit can be saved in different formats, which can then be input to other external software for visualization. TADbit currently outputs models in three main formats:

- JSON file is the most efficient to store the results of the TADbit analysis. This file contains the particle coordinates, the experimental data structure (models, clusters, centroids, normalized interaction matrix, and restraints), and the project metadata. These can be visualized alongside genomic features and matrix

datasets within the TADkit visualization tool (https://github.com/3DGenomes/TADkit).

```
models.write_json(filename=output_file)
```

- XYZ/CMM file of particle coordinates. These can be examined and rendered in the majority of the molecular visualization software, e.g., Chimera [36], Delta [37], PyMOL [38], or VMD [39].

```
models.write_cmm(directory=output_directory)

models.write_xyz(directory=output_directory)
```

4 Notes

1. To speed up the data preprocessing, we recommend using split FASTQ files, one for each read-end. We also suggest working with compressed files to save disk space. To this end, TADbit accepts FASTQ files compressed in many formats such as zip, bzip2, gzip, tar, and DSRC [23].
2. The Hi-C-specific quality plots provided by TADbit (Fig. 1) use the first million paired-ends of the input FASTQ file to assess the quality of the entire dataset. From the quality plot measures, one can infer useful indicators of the quality of the data:
   (a) The first plot, that is common in next-generation sequencing (NGS), describes the quality of the sequencing with PHRED scores and proportion of *Ns* per position. A PHRED score larger than 30 (which represents a 99.9% accuracy of the base call) and a number of *Ns* close to zero are indicative of high confidence in the sequenced product.
   (b) The percentage of digested sites (the ratio of digested over undigested sites) should be larger than 60%. Lower values indicate a problem in the digestion step of the Hi-C experiment.
   (c) The percentage of dangling-ends (the number of time a digested site is found at the beginning of a read) should be between 1% and 10%. This percentage varies also depending on the average length of the restriction fragments; the larger they are, the fewer dangling-ends should be expected. Large percentages indicate a problem in ligation efficiency.
   (d) The percentage of ligation sites (the number of times a ligation site is found in the processed reads) should be higher than 15% for read length of 75 nucleotides (nt) and

sequenced fragment size ~300 nt. Low numbers here indicate a problem in ligation efficiency. These percentages are calculated as follows. Assuming that the ligation sites are homogeneously distributed along the genome, their average number per sequenced fragment can be estimated by extrapolating it to the full length of the sequenced fragment ~300 nt. For example, in the case of Figs. 1, 20.5% of the reads present (at least) one ligation site. Given that our search space for this set of reads is 67 nt (read length is 75, minus the size of the 8 nt ligation pattern GATCGATC), and assuming a constant number of ligation sites in all the 300 nt fragment, we would expect to find a ligation site in about 92% of the reads. Usually, we expect that all the reads in the library contain biotinylated nucleotides. It means that each read has either a ligation site or a dangling-end. In this case, we have calculated that the proportions of ligation sites and dangling-ends sum up to about 100% as expected.

3. The inference of the average RE fragment size in the sequenced library is relevant for the interpretation of the Hi-C data because valid interactions consist of read-ends mapped close to a RE cut site, and this definition of proximity relies on the estimated size of the sequenced DNA fragments.
4. Some of the filters applied to the set of pairs of mapped read-ends may considerably reduce the number of valid interactions while not increasing the general quality of the resulting interaction matrix. In the TADbit implementation, none of the filters is mandatory but can be switched on or off making it easy to adapt to other 3C-based methods. According to our experience, for a Hi-C experiment, the user should define a minimum set of filters to apply including (1) any paired-end reads mapped in a single restriction fragment or very close to the diagonal (self-circle, dangling-end, error, and extra dangling-end), (2) read-ends mapped in too short or too long defined RE fragments, (3) PCR artifacts (duplicates), and (4) random breaks.
5. This filter may be too conservative for the in situ Hi-C protocol [11], and it is usually not applied.
6. In Hi-C data analysis, the choice of the binning is limited usually by the amount and the quality of data or more rarely by the available computational power. However, some of the structural features that can be detected may appear only at certain scales above or below the chosen binning. For example, in mammalian genomes, the compartments are usually detected optimally at 100 kb binning. Depending on the interaction density, it may be more difficult to determine them at

finer scales. To ensure that a given binning is adequate to the scale of the measured structure or, more simply, does not entail too much noise, a good strategy is to check for the consistency between replicates or different resolutions. In this chapter, we provide an example of the detection of TAD borders in the notebook Methods-8-Compartments and TADs detection and Fig. 6. Additionally, binning is important in the modeling step. The genomic region to model could vary in size from few kb to Mb because it can be defined as a chromosome part, an entire chromosome, or also a set of chromosomes. Hence, the resolution used in each situation should be tailored to the computational power to avoid having too many particles and restraints during modeling. Normally the number of particles to model should be maintained below 5000.

7. The sign variation of the first eigenvector of the autocorrelated Hi-C matrix usually represents the borders between compartments (Fig. 6a), but, especially in the case of low-quality Hi-C experiments or aberrant karyotypes, it could also describe other correlations in the matrix. In such cases, the A/B compartments can be explained by the second or the third eigenvector. To this end, it is essential to visually check the correspondence between the checkerboard pattern characteristic of compartments and the eigenvectors, especially when working with atypical datasets. Other more automatic metrics can be used to confirm the choice of a given eigenvector, like the level of correlation with some activity marker (GC content, RNA-seq, or epigenetic marks), or the consistency with the compartment call done at a lower resolution.

   In the case of the detection of TAD borders, a typical problem is the choice of the adequate binning. The first concern is that an increased resolution is correlated with increased noise [29]. A good strategy is to test the consistency of several resolutions between independent replicates. An example is provided using the insulation score strategy (Fig. 9, Methods-8-Compartments and TADs detection). It shows that in these datasets a resolution between 50 kb and 100 kb is optimal to call TAD border, while that at higher resolutions (<50 kb) the consistency between time points is lost.

Acknowledgments

We thank all the current and past members of the Marti-Renom lab for their continuous discussions and support to the development of TADbit. I.F. was supported by the Ministerio de Ciencia, Innovación y Universidades of Spain (IJCI-2015-23352). M.A.M-R was supported by the European Research Council under the seventh

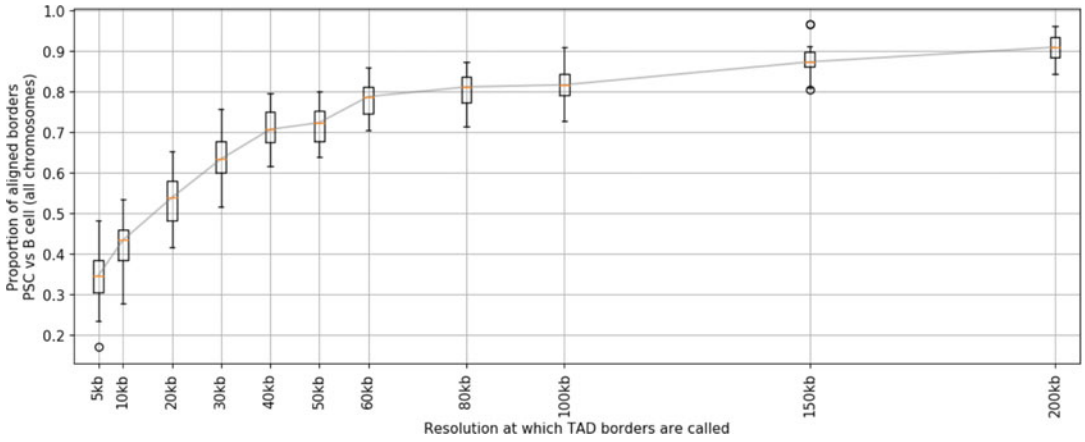

Fig. 9 TAD border consistency as a function of matrix binning. Comparison of the number of TAD borders that are shared (plus-minus one bin) between the two replicates in two different cell types (B and iPS cells) when TAD borders are called at a different matrix resolution. The TAD borders are called using the insulation score with a window of 500 kb and a delta of 100 kb

Framework Program FP7/2007-2013 (ERC grant agreement 609989), the European Union's Horizon 2020 research and innovation program (grant agreement 676556), and the Spanish Ministry of Economy and Competitiveness (BFU2013-47736-P and BFU2017-85926-P). We also acknowledge support from "Centro de Excelencia Severo Ochoa 2013-2017," SEV-2012-0208, and the CERCA Programme/Generalitat de Catalunya to the CRG. Marco Di Stefano and David Castillo contributed equally to this work.

References

1. Dekker J, Rippe K, Dekker M, Kleckner N (2002) Capturing chromosome conformation. Science 295:1306–1311
2. Dekker J, Marti-Renom MA, Mirny LA (2013) Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nat Rev Genet 14:390–403
3. Grob S, Cavalli G (2018) Technical review: a Hitchhiker's guide to chromosome conformation capture. Methods Mol Biol 1675:233–246
4. Kim TH, Dekker J (2018) 3C-based chromatin interaction analyses. Cold Spring Harb Protoc 2018:pdb top097832
5. Lieberman-Aiden E, van Berkum NL, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO, Sandstrom R, Bernstein B, Bender MA, Groudine M, Gnirke A, Stamatoyannopoulos J, Mirny LA, Lander ES, Dekker J (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science 326:289–293
6. Zink D, Cremer T, Saffrich R, Fischer R, Trendelenburg MF, Ansorge W, Stelzer EH (1998) Structure and dynamics of human interphase chromosome territories in vivo. Hum Genet 102:241–251
7. Cremer T, Cremer C (2001) Chromosome territories, nuclear architecture and gene regulation in mammalian cells. Nat Rev Genet 2:292–301
8. Boyle S, Gilchrist S, Bridger JM, Mahy NL, Ellis JA, Bickmore WA (2001) The spatial organization of human chromosomes within the nuclei of normal and emerin-mutant cells. Hum Mol Genet 10:211–219
9. Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, Hu M, Liu JS, Ren B (2012)

Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature 485:376–380

10. Nora EP, Lajoie BR, Schulz EG, Giorgetti L, Okamoto I, Servant N, Piolot T, van Berkum NL, Meisig J, Sedat J, Gribnau J, Barillot E, Bluthgen N, Dekker J, Heard E (2012) Spatial partitioning of the regulatory landscape of the X-inactivation Centre. Nature 485:381–385
11. Rao SS, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, Sanborn AL, Machol I, Omer AD, Lander ES, Aiden EL (2014) A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell 159:1665–1680
12. Rowley MJ, Corces VG (2018) Organizational principles of 3D genome architecture. Nat Rev Genet 19:789–800
13. Bau D, Sanyal A, Lajoie BR, Capriotti E, Byron M, Lawrence JB, Dekker J, Marti-Renom MA (2011) The three-dimensional folding of the alpha-globin gene domain reveals formation of chromatin globules. Nat Struct Mol Biol 18:107–114
14. Le Dily F, Bau D, Pohl A, Vicent GP, Serra F, Soronellas D, Castellano G, Wright RH, Ballare C, Filion G, Marti-Renom MA, Beato M (2014) Distinct structural transitions of chromatin topological domains correlate with coordinated hormone-induced gene regulation. Genes Dev 28:2151–2162
15. Serra F, Bau D, Goodstadt M, Castillo D, Filion GJ, Marti-Renom MA (2017) Automatic analysis and 3D-modelling of hi-C data using TADbit reveals structural features of the fly chromatin colors. PLoS Comput Biol 13: e1005665
16. Stadhouders R, Vidal E, Serra F, Di Stefano B, Le Dily F, Quilez J, Gomez A, Collombet S, Berenguer C, Cuartero Y, Hecht J, Filion GJ, Beato M, Marti-Renom MA, Graf T (2018) Transcription factors orchestrate dynamic interplay between genome topology and gene regulation during cell reprogramming. Nat Genet 50:238–249
17. Virtanen P, Gommers R, Oliphant TE, et al. (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17(3):261-272
18. Hunter JD (2007) Matplotlib: a 2D graphics environment. Comput Sci Eng 9:90–95
19. Russel D, Lasker K, Webb B, Velazquez-Muriel J, Tjioe E, Schneidman-Duhovny D, Peterson B, Sali A (2012) Putting the pieces together: integrative modeling platform software for structure determination of macromolecular assemblies. PLoS Biol 10:e1001244
20. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, 1000 Genome Project Data Processing Subgroup (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25:2078–2079
21. Marco-Sola S, Sammeth M, Guigo R, Ribeca P (2012) The GEM mapper: fast, accurate and versatile alignment by filtration. Nat Methods 9:1185–1188
22. Vidal E, le Dily F, Quilez J, Stadhouders R, Cuartero Y, Graf T, Marti-Renom MA, Beato M, Filion GJ (2018) OneD: increasing reproducibility of hi-C samples with abnormal karyotypes. Nucleic Acids Res 46:e49
23. Roguski L, Deorowicz S (2014) DSRC 2--industry-oriented compression of FASTQ files. Bioinformatics 30:2213–2215
24. Dekker J (2007) GC- and AT-rich chromatin domains differ in conformation and histone modification status and are differentially modulated by Rpd3p. Genome Biol 8:R116
25. Derrien T, Estelle J, Marco Sola S, Knowles DG, Raineri E, Guigo R, Ribeca P (2012) Fast computation and applications of genome mappability. PLoS One 7:e30377
26. Imakaev M, Fudenberg G, McCord RP, Naumova N, Goloborodko A, Lajoie BR, Dekker J, Mirny LA (2012) Iterative correction of hi-C data reveals hallmarks of chromosome organization. Nat Methods 9:999–1003
27. Yaffe E, Tanay A (2011) Probabilistic modeling of hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. Nat Genet 43:1059–1065
28. Crane E, Bian Q, McCord RP, Lajoie BR, Wheeler BS, Ralston EJ, Uzawa S, Dekker J, Meyer BJ (2015) Condensin-driven remodelling of X chromosome topology during dosage compensation. Nature 523:240–244
29. Forcato M, Nicoletti C, Pal K, Livi CM, Ferrari F, Bicciato S (2017) Comparison of computational methods for hi-C data analysis. Nat Methods 14(7):679–685
30. Shin H, Shi Y, Dai C, Tjong H, Gong K, Alber F, Zhou XJ (2016) TopDom: an efficient and deterministic method for identifying topological domains in genomes. Nucleic Acids Res 44:e70
31. Mizuguchi T, Fudenberg G, Mehta S, Belton JM, Taneja N, Folco HD, FitzGerald P, Dekker J, Mirny L, Barrowman J, Grewal SI (2014) Cohesin-dependent globules and heterochromatin shape 3D genome architecture in S. pombe. Nature 516:432–435
32. Yang T, Zhang F, Yardimci GG, Song F, Hardison RC, Noble WS, Yue F, Li Q (2017)

HiCRep: assessing the reproducibility of hi-C data using a stratum-adjusted correlation coefficient. Genome Res 27:1939–1949

33. Yan KK, Yardimci GG, Yan C, Noble WS, Gerstein M (2017) HiC-spector: a matrix library for spectral and reproducibility analysis of hi-C contact maps. Bioinformatics 33:2199–2201
34. Baù D, Marti-Renom MA (2012) Genome structure determination via 3C-based data integration by the integrative Modeling platform. Methods 58:300–306
35. Trussart M, Serra F, Bau D, Junier I, Serrano L, Marti-Renom MA (2015) Assessing the limits of restraint-based 3D modeling of genomes and genomic domains. Nucleic Acids Res 43:3465–3477
36. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, Ferrin TE (2004) UCSF chimera--a visualization system for exploratory research and analysis. J Comput Chem 25:1605–1612
37. Tang B, Li F, Li J, Zhao W, Zhang Z (2018) Delta: a new web-based 3D genome visualization and analysis platform. Bioinformatics 34:1409–1410
38. DeLano WL (2002) The PyMOL molecular graphics system on world wide web. http://www.pymol.org/in
39. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14:33–38