This is the peer reviewd version of the followng article:

Analysis of HiChIP Data / Dori, M., Forcato, M. (METHODS IN MOLECULAR BIOLOGY). - In: Methods in Molecular Biology[s.l] : Humana Press Inc., 2022. - ISBN 978-1-0716-1389-4. - pp. 209-234 [10.1007/978-1-0716-1390-0_11]

Humana Press Inc.
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

08/10/2026 18:44

(Article begins on next page)

Analysis of HiChIP data

Martina Dori* and Mattia Forcato*

Department of Life Sciences, University of Modena and Reggio Emilia; Via G. Campi, 287, 41125, Modena, Italy

* Corresponding authors: martina.dori@unimore.it, mattia.forcato@unimore.it

Silvio Bicciato and Francesco Ferrari (eds.), Hi-C Data Analysis: Methods and Protocols, Methods in Molecular Biology, vol. 2301, https://doi.org/10.1007/978-1-0716-1390-0_11, © Springer Science+Business Media, LLC, part of Springer Nature 2022

Running head

Analysis of HiChIP data

Summary

HiChIP is a novel method for the analysis of chromatin interactions based on *in situ* Hi-C that adds an immuno-precipitation (ChIP) step for the investigation of chromatin structures driven by specific proteins. This approach has been shown to be very efficient as it reliably reproduces Hi-C results and displays a higher rate of informative reads with a required lower amount of input cells when compared with other ChIP-based techniques (as ChIA-PET). Although HiChIP data pre-processing can be performed with the same methods developed for other Hi-C techniques, the identification of chromatin interactions needs to take into account specific biases introduced by the ChIP step. In this chapter we describe a computational pipeline for the analysis of HiChIP data obtained with the immuno-precipitation of Rad21 (part of the cohesin complex) in human embryonic stem cells before and after heat-shock treatment. We provide a detailed description of the pre-processing of raw data, the identification of chromatin interactions, the evaluation of the alterations induced by treatment and, finally, the visualization of differential loops.

Key words

HiChIP, Chromatin interactions, Differential interactions, Bioinformatics

1. Introduction

HiChIP [1] is a recently published protocol for protein-centered chromatin structure studies. It is part of 3C technologies [2,3] and recapitulates the classic *in situ* Hi-C protocol with the addition of a chromatin immuno-precipitation (ChIP) step after DNA digestion and ligation, hence isolating and sequencing solely fragments cross-linked with a specific protein. This allows focusing the analysis on loci associated with the protein of interest rather than measuring all contacting fragments. HiChIP performed with an antibody for the cohesin subunit Smc1a has been shown to reliably overlap *in situ* Hi-C results and to improve signal-to-background ratio. Moreover, when compared to ChIA-PET, another ChIP-based protocol, HiChIP has an overall higher rate of informative reads requiring a lower amount of input cells [1]. Despite the methodological advantages, the addition of the immuno-precipitation step leads to an uneven and restriction-fragment-biased distribution of sequenced reads that has to be taken into account especially when identifying locally enriched interactions (loops) [4,5].

In this chapter we present a pipeline for the analysis of HiChIP data and the identification and visualization of loops (**Figure 1**). Raw sequencing data from a public HiChIP dataset comprising different experimental conditions [6] are initially processed with *HiC-Pro* [7], a pipeline that performs alignment and filtering of read pairs. For the identification of locally enriched interactions, filtered reads are analyzed with *hichipper* [4], a modified version of *Mango* algorithm [8] that accounts for the sequencing biases associated with the ChIP step and that uses ChIP-Seq data to increase the accuracy of interactions call. We further analyze identified loops with *diffloop* [9] to search for significant variations in topology (i.e. gains and losses of loops) between samples. Finally, we visualize local interactions using *diffloop* and the *WashU Epigenome Browser* [10] to explore variations in chromatin interactions between conditions and compare them with epigenomic data.

2. Materials

Materials used for the analysis of HiChIP data include computers running Unix operating system and various software to perform preprocessing of raw data, interaction calling, identification of differential loops and visualization. All the necessary software is listed in subsection 2.1.2, along with the installed version.

2.1 Required resources

2.1.1 Hardware

UNIX operating system with AWK package and standard CPUs.

2.1.2 Software

The following software and tools are required to perform the analyses on the example data set:

SRA toolkit 2.9.2: http://ncbi.github.io/sra-tools/

samtools 1.9: http://samtools.sourceforge.net

CrossMap 0.4.0: http://crossmap.sourceforge.net/

bowtie 0.12.7: http://bowtie-bio.sourceforge.net/index.shtml

macs2 2.1.0.20150731: https://github.com/taoliu/MACS

HiC-Pro 2.11.1: http://nservant.github.io/HiC-Pro/

hichipper 0.7.7: https://hichipper.readthedocs.io/en/latest/index.html

R 3.4.4: https://cran.r-project.org

diffloop 1.14.0: https://bioconductor.org/packages/release/bioc/html/diffloop.html

ggrepel 0.8.1: https://cran.r-project.org/web/packages/ggrepel/index.html

WashU Epigenome Browser: http://epigenomegateway.wustl.edu/browser/

With the exception of *diffloop*, *ggrepel* and *WashU Epigenome Browser*, all tools run on Unix via the command line. *diffloop* and *ggrepel* are R packages and need to be run from R (see **Note 1**), whereas *WashU Epigenome Browser* is a web browser for the

visualization of epigenomic data. *CrossMap* requires python (it works with both python version 2.7 and 3). We assume that all tools are installed in the `/Software` folder (either as binary or from source code) and that each program has been added to the user path. For the installation, refer to the help page of each tool.

2.2 Dataset

2.2.1 Example dataset: generalities

In this chapter we analyze the Lyu *et al.* dataset, containing HiChIP data for different architectural proteins from human embryonic stem cells (hESC) before and after heat shock treatment [6]. The aim of the study was to assess the involvement of chromatin rearrangement in the transcriptional response to temperature stress. In particular, we focus on data derived from the immuno-precipitation of Rad21, part of the cohesin complex that has a fundamental role in the control of the functional architecture of the genome [11]. HiChIP protocol was carried out as described in [1] with some modifications. Samples were digested with DpnII, an isoschizomer of MboI, that recognizes the GATC restriction site. Sequencing was performed with a paired-end layout and read length of 50bp. Since we want to identify topologies which are significantly altered by the heat shock treatment, we need to retrieve at least 2 replicates for each condition. All the selected samples (for a total of 4 files) are reported in **Table 1**. For the sake of simplicity, we illustrate the pre-processing and the identification of loops on a single HiChIP file (Rad21_Rep1).

2.2.2 Downloading HiChIP data

The entire dataset from Lyu *et al.* is available for download at the Sequence Read Archive (SRA; **Table 1** and **Note 2**). Raw reads can be downloaded and converted into FASTQ format using the *SRA Toolkit* [12] as follows:

```
WORKDIR=/home/HiChIP-analyses
```

```
cd $WORKDIR
mkdir fastq
fastq-dump --split-files -O fastq/ --gzip SRR6206783
```

where `--split-files` is used to separate the paired reads into two distinct files, `-O` specifies the folder in which the FASTQ will be saved and `SRR6206783` is the run accession number of Rad21 replicate 1 sample. For practical reasons, we rename the FASTQ files of each replicate as reported in **Table 1**:

```
mv fastq/SRR6206783_1.fastq.gz fastq/Rad21_Rep1_1.fastq.gz
mv fastq/SRR6206783_2.fastq.gz fastq/Rad21_Rep1_2.fastq.gz
```

2.2.3 Downloading additional data

The alignment of the raw reads requires the FASTA sequence and the *Bowtie2* indexes of the reference genome (here we use the UCSC hg19 assembly), that can be downloaded from Illumina's iGenomes collection at the *Bowtie2* website (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml). To perform interactions call with *hichipper*, we additionally download from Lyu *et al.* [6] the FASTQ files relative to Rad21 and control IgG ChIP-Seq experiments performed in the same cell line and experimental conditions as the HiChIP dataset (**Table 2**). The raw files can be downloaded and renamed as shown in the previous paragraph. To remove ChIP-Seq peaks falling in repetitive regions or other anomalous loci, we download in our working directory a BED file containing the ENCODE DAC Blacklisted Regions (https://www.encodeproject.org/annotations/ENCSR636HFF/). In the "Files" section of the web page select the "File Details" tab and download the hg19 BED/BED6 file; after de-compressing the .gz file, we rename it from `ENCFF001TDO.bed` to `hg19_DAC_blacklist.bed`.

Finally, to compare differential loops with variations in Rad21 binding and in the activation level of the interacting loci, we download from Gene Expression Omnibus (GEO) the signal files (BigWig) relative to the ChIP-Seq experiment for Rad21 and

H3K27ac, also produced in Lyu *et al.* [6] (**Table 3**). Since Lyu *et al.* used a different assembly, we will liftover the BigWig files to hg19 using *CrossMap* (see **Note 3**).

3. Methods

3.1 Preprocessing of raw data with *HiC-Pro*

HiC-Pro is a pipeline for the processing of Hi-C data that, starting from sequencing reads (FASTQ files), performs several steps such as alignment, filtering, binning, and normalization. The final output files are the raw and normalized contact matrices together with additional information regarding the quality of the input data [7]. Since *hichipper* (the HiChIP-specific loop caller used in our pipeline) takes as input the aligned and filtered read pairs generated by *HiC-Pro*, we need to perform only the alignment and filtering steps. These can be carried out as in the analysis of a Hi-C experiment. We recommend the user to carefully read the experimental protocol used to generate the HiChIP data and to take note of the adopted restriction enzyme and any other specification (e.g. fill-in strategy, library preparation) that can affect the computational analysis. Before proceeding with *HiC-Pro*, it is also advisable to perform standard quality controls and eventually trimming on the raw data, as commonly done in any other sequencing experiment (see **Note 4**).

3.1.1 Input and configuration files

HiC-Pro requires the raw sequencing files to be organized in per-sample subdirectories. To this end, we create one folder for each sample within the `fastq/` folder and move the relative FASTQ files as follows:

```
mkdir fastq/Rad21_Rep1
mv fastq/Rad21_Rep1_*.fastq.gz fastq/Rad21_Rep1/
```

The use of the asterisk will move both mates of a sample with a single command. To run *HiC-Pro*, four annotation files are required: i) *Bowtie2* indexes, ii) a table including the chromosomes sizes, iii) a file reporting the coordinates of the restriction fragments after

digestion and iv) a configuration file. All these files must refer to the same version of the genome chosen as reference, in our case hg19. *HiC-Pro* already provides some of the annotation files required, such as chromosome sizes for hg19 and the restriction fragments of few restriction enzymes. If the enzyme used for digestion in the HiChIP protocol is not included, as in our case with DpnII, it is possible to generate the restriction fragment file with the *digest_genome.py* script available in the `HiC-Pro-2.11.1/bin/utils` folder. *digest_genome.py* takes as input the sequence recognized by the restriction enzyme (`^GATC` for DpnII, where the '^' character indicates the cutting site) and the sequence of the reference genome in FASTA format and returns the restriction fragment coordinates file:

```
/home/Programs/HiC-Pro-2.11.1/bin/utils/digest_genome.py -r
^GATC -o DpnII_resfrag_hg19.bed
/home/Annotation/Homo_sapiens/UCSC/hg19/Sequence/WholeGenomeFast
a/genome.fa
```

The resulting `DpnII_resfrag_hg19.bed` file needs to be moved into the `/annotation` folder of *HiC-Pro*, where it will be automatically found during the preprocessing steps:

```
mv DpnII_resfrag_hg19.bed /home/Programs/HiC-
Pro_2.11.1/annotation/
```

HiC-Pro configuration file includes all the settings and the paths to the annotation files that are required for the analysis. As for the restriction fragment file, the installation folder of *HiC-Pro* provides a pre-compiled configuration file called `config-hicpro.txt`. Although most of the parameters can be left as default, others need to be modified according to the user settings to properly run the analysis. In particular:

- in the DATA section, we need to specify how the FASTQ files for each mate are named, in our case `_1` and `_2`;
- in the ALIGNMENT section, we have to provide the path to the *Bowtie2* indexes;

- in the ANNOTATION file section, we have to specify the name of the reference genome, which has to be the same as the one appearing as prefix of the *Bowtie2* index files;
- in the DIGESTION section, we have to indicate the name of the restriction fragment file (no need for the complete path as *HiC-Pro* will look for it in its own `annotation/` folder) and the ligation site. Note that the ligation site is the sequence that results from the ligation of two digested fragments and it depends on the fill-in strategy adopted in the experiment. In our case, the ligation site is `GATCGATC`.

The updated fields of the configuration file are as follows:

```
PAIR1_EXT = _1
PAIR2_EXT = _2
BOWTIE2_IDX_PATH =
/home/Annotation/Homo_sapiens/UCSC/hg19/Sequence/Bowtie2Index
REFERENCE_GENOME = genome
GENOME_FRAGMENT = DpnII_resfrag_hg19.bed
LIGATION SITE = GATCGATC
```

The modified configuration file can be saved as `config-HiChIP.txt` in our `$WORKDIR`.

3.1.2 Alignment

HiC-Pro can process samples in a single run or can be operated in a sequential mode that requires the user to call each subset of steps (specified with the `-s` option). Since we want to check the quality of the results at each step and we do not need to perform the entire pipeline (we will not generate the contact matrices), we take advantage of the possibility of running *HiC-Pro* in sequential mode. For the alignment step (`-s mapping`) we use the following commands:

```
HiC-Pro=/home/Programs/HiC-Pro-2.11.1/bin/HiC-Pro
$HiC-Pro -c config-HiChIP.txt -i fastq -o HiC_Pro -s mapping -s
quality_checks
```

which will create the output folder `HiC_Pro/` containing the results of the alignments in the `bowtie_results/` subfolder. Through the `-s quality_checks`, *HiC-Pro* will create two additional subfolders (`hic_results/stats/ and hic_results/pic/`) where all the alignment statistics are stored and graphically represented; all of these result folders are further divided into a subfolder for each sample found in `fastq/`.

HiC-Pro adopts a two-step alignment strategy using *Bowtie2*. Initially, reads for the two mates that align on the genome are saved in a BAM file. Next, unmapped reads are trimmed according to the ligation site indicated in the configuration file, *Bowtie2* is run again on the trimmed reads and alignments are saved in a second BAM file. Finally, BAM files are combined and stored in the `bwt2` folder. Since the two mates are aligned independently, statistics are reported separately for each mate. As shown in **Table 4**, the vast majority of the reads are aligned on the genome (more than 90% on average) and about 10% of reads on average is mapped after trimming. This means that 10% of the reads were chimeric, i.e. spanned the ligation junction. The proportion of chimeric reads depends on the size of the fragments selected during sample preparation and on the read length. A high number of unmapped reads (more than 20%) could indicate problems with the library (e.g. read-through adapters or contamination) or that the wrong ligation site was given as input to *HiC-Pro*. Considering the alignment performance, we can proceed with filtering.

3.1.3 Filtering

The filtering step takes as input the configuration file and the final combined BAM file for each mate and performs reads pairing and assignment to the restriction fragments (`-s proc_hic`) using the following command:

```
$HiC-Pro -c config-HiChIP.txt -i HiC_Pro/bowtie_results/bwt2 -o
HiC_Pro -s proc_hic -s quality_checks
```

where `-s quality_checks` saves statistics and diagnostic plots relative to this second step. This command generates a `data/` subfolder inside the `hic_results/` folder with the coordinates of all valid interaction products (`.allValidPairs` file). Since *HiC-Pro* performs all the filters defined in the configuration file, it is possible to re-run the filtering step with a set of modified filters. First, read pairs are filtered based on their alignment and pairing. Next, read pairs are mapped on the restriction fragments and are classified as valid and invalid pairs based on their position and orientation on the restriction fragments. Then, PCR duplicates are removed and de-duplicated valid pairs are saved in the `.allValidPairs` file. As summarized in **Table 4**, in our dataset we retain on average about 28% of sequenced reads. For more details on the interpretation and fine tuning of *HiC-Pro* filtering parameters and results, please refer to the *HiC-Pro* documentation (http://nservant.github.io/HiC-Pro) and [13].

3.2 Identification of loops

3.2.1 ChIP-Seq peak calling

To accurately identify interacting regions from HiChIP data with *hichipper*, it is useful to provide, when available, protein binding information from ChIP-Seq experiments. *hichipper* takes as input the ChIP-Seq peak coordinates in a BED format; we will generate such files starting from the raw reads of Rad21 ChIP-Seq in hESC before and after HS treatment. After downloading the FASTQ files in the `fastq_ChIP` folder, we apply a standard pipeline which includes alignment, conversion in BAM format, sorting, duplicates removal and finally peak calling (see **Note 5**). More details about the analysis of ChIP-Seq data can be found in [14] and [15]. Since we are again dealing with multiple samples, the following code chunks are relative only to Rad21 and IgG control ChIP-Seq in the untreated condition. The first step is the alignment with *bowtie* on hg19:

```
mkdir bowtie
mkdir ChIP_peaks
```

```
export
BOWTIE_INDEXES=/home/Annotation/Homo_sapiens/UCSC/hg19/Sequence/
BowtieIndex/
bowtie -S -t -p 16 -m 1 genome -1 fastq_ChIP/Rad21_1.fastq -2
fastq_ChIP/Rad21_2.fastq > bowtie/Rad21.sam
bowtie -S -t -p 16 -m 1 genome -1 fastq_ChIP/IgG_1.fastq -2
fastq_ChIP/IgG_2.fastq > bowtie/IgG.sam
```

Next, we take advantage of *samtools* for the SAM to BAM conversion and the subsequent sorting and de-duplication:

```
samtools view -S -h -b bowtie/Rad21.sam -o bowtie/Rad21.bam
samtools view -S -h -b bowtie/IgG.sam -o bowtie/IgG.bam
samtools sort bowtie/Rad21.bam -o bowtie/Rad21.sort.bam
samtools sort bowtie/IgG.bam -o bowtie/IgG.sort.bam
samtools rmdup bowtie/Rad21.sort.bam bowtie/Rad21.sort.noDup.bam
samtools rmdup bowtie/IgG.sort.bam bowtie/IgG.sort.noDup.bam
rm bowtie/*.sam
rm bowtie/Rad21.bam
rm bowtie/IgG.bam
rm bowtie/*.sort.bam
```

Finally, we use *MACS2* [16] to call ChIP-Seq peaks:

```
macs2 callpeak -t bowtie/Rad21.sort.noDup.bam -c
bowtie/IgG.sort.noDup.bam --keep-dup all -g 2685511504 -n
ChIP_peaks/Rad21
```

This command will produce several outputs; for the identification of loops we need the narrowPeak file (a type of BED format). Finally, we filter the narrowPeak file to remove peaks overlapping the ENCODE blacklisted regions, generating the final `Rad21_peaks.noBL.narrowPeak` file with the following command:

```
intersectBed -a ChIP_peaks/Rad21_peaks.narrowPeak -b
hg19_DAC_blacklist.bed -v >
ChIP_peaks/Rad21_peaks.noBL.narrowPeak
```

We obtain 44,190 and 53,219 significant Rad21 peaks in untreated and HS condition, respectively.

3.2.2 Identification of HiChIP loops

Because of the immunoprecipitation step in the HiChIP experimental setup, loops can be defined as interactions between regions bound by the targeted protein (anchor regions). Thus, these regions should be characterized by the presence of a ChIP peak. *Hichipper* requires ChIP peaks information as a starting point for the definition of anchors regions; these peaks can be either user-supplied and derived from independent ChIP-Seq experiment or inferred by *hichipper* from a subset of reads of the HiChIP itself (see **Note 6**). Unlike other ChIP-based assays, HiChIP protocol also involves a digestion with a restriction enzyme that causes the read-density landscape to be strongly biased by the proximity to the enzyme cutting site, hindering the correct identification of peaks representing loop anchors (and, as a consequence, of loops) [4]. In fact, in cases where the ChIP peak falls in the middle of a long restriction fragment, no supporting HiChIP Paired End Tags (PETs) are found close to the peak, as they mainly cluster in proximity to the restriction site, leading to an underestimation of identified loops. To overcome this limitation, *hichipper* implements a fragment-aware padding approach. First, peaks are padded by a fixed window (default is 500bp on both left and right) to take account of the uncertainty in the peak definition; then, loop anchors are defined by expanding each peak to the end of the overlapping restriction fragment or fragments [4]. The effect of this approach is to increase the number of PETs supporting interactions between anchors. Additionally, *hichipper* subsets loops based on the genomic location and distance between interacting anchors. In particular, only intra-chromosomal loops are retained; the minimum distance between anchors is set to 5kb, to avoid the inclusion of self-ligated loops, whereas the maximum distance is set to 2Mb, as interactions that occur at greater distances are less likely to be biologically relevant. Finally, *hichipper* implements

Mango loop significance model [8] to boost the accuracy of the identification of interactions in HiChIP data. *Mango* models the likelihood of interactions between genomic regions as a function of their genomic separation distance and their read depth. The resulting probability of observing interacting loci is then corrected for multiple tests using the Benjamini-Hochberg method [8]. The input for *hichipper* is the *HiC-Pro* output folder where *hichipper* is able to locate the files containing the coordinates of the aligned and valid read pairs. *hichipper* can be run either by providing a sample description file (or manifest file, `.yaml`) or by directly specifying all the input files in the command (`call` mode). The sample description file must include three variables: i) the ChIP-Seq peak file or the indication of how the peaks should be inferred directly from the HiChIP data (see **Note 6**); ii) the restriction fragment coordinates file used for *HiC-Pro*, and iii) the path to *HiC-Pro* output folder. In this protocol we will use the sample description file method. Since in the manifest file we can specify only a single ChIP-Seq peak file, we will prepare one manifest file for each ChIP-Seq experiment (see **Note 7**). The `.yaml` file referring to untreated Rad21 ChIP-Seq peaks is reported below, this file is stored in a folder named `yaml` and it assumes that *hichipper* is run from `$WORKDIR`:

```
peaks:
 - ChIP_peaks/Rad21_peaks.noBL.narrowPeak
resfrags:
 - /Programs/HiC-Pro_2.11.1/annotation/DpnII_resfrag_hg19.bed
hicpro_output:
 - HiC_Pro
```

This manifest file (`Rad21.yaml`) can be used for both replicates of untreated Rad21 HiChIP as follows:

```
cd $WORKDIR
mkdir Hichipper
hichipper --keep-samples Rad21_Rep1 -l 50 --out RAD21_1
Rad21.yaml
```

```
mv RAD21_1 Hichipper/
hichipper --keep-samples Rad21_Rep2 -l 50 --out RAD21_2
Rad21.yaml
mv RAD21_2 Hichipper/
```

where `--keep-samples` tells *hichipper* to work only on the sample that is specified (Rad21_Rep1) and to ignore any other sample found in the `HiC-Pro/hic_results/data` folder. `-l` specifies the HiChIP read length while the `--out` parameter instructs *hichipper* on where to save all the generated output; if the folder already exists, it will return an error requiring to either change the name or remove the existing folder. Here, we create one folder for each sample and then move it into `Hichipper/` folder for convenience.

hichipper generates two per-run files (`RAD21_1.log` and an html QC report file, `RAD21_1.hichipper.qcreport.html`) and seven per-sample ones:

- `Rad21_Rep1.stat` contains overall statistics of PETs.
- `Rad21_Rep1.anchors.bed` includes the genomic coordinates of all loop anchors that have been identified.
- `Rad21_Rep1.inter.loop_counts.bedpe` and `Rad21_Rep1.intra.loop_counts.bedpe` include inter-chromosomal and intra-chromosomal loops, respectively.
- `Rad21_Rep1.filt.intra.loop_counts.bedpe` contains all the intra-chromosomal loops with the anchors distance falling within the min/max cut-off values.
- `Rad21_Rep1.interactions.all.mango` includes the same set of loops of `Rad21_Rep1.filt.intra.loop_counts.bedpe` with the addition of a per-loop FDR value computed using the bias correction algorithm implemented in

Mango [8]. These interactions are also saved as `Rad21_Rep1-HiChIP.rds`, a file containing an R "loop" object.

In our example, we did not modify any of the default parameters. The default values for the min/max distance between anchors are 5kb and 2Mb, but it is possible to modify them through the `--min-dist` and `--max-dist` parameters, respectively. The user can also modify the fragment-aware padding approach e.g. by tuning the left/right peak extension (`--peak-pad`) to increase the number of reads pairs that can be mapped to loops. In **Table 5** we report several quality metrics visible in the html report and in the `.stat` files, that allow an evaluation of the overall quality of the HiChIP, together with the total and significant (FDR ≤ 0.01) number of loops identified per sample. The "Reads in Anchors" are the read pairs that are unique, mapped and with both reads overlapping with a loop anchor (note that here are also counted pairs filtered out by *HiC-Pro* such as those falling on the same restriction fragment); this should be a good estimation of the quality of the HiChIP. The "Mapped unique quality valid" are the de-duplicated valid PETs stored in the `.allValidPairs` file saved by *HiC-Pro*. The "Valid intra 5kb-2Mb" represent PETs involved in intra-chromosomal interactions within the specified distance range and is an evaluation of the efficacy of the *in situ* ligation. Finally, the "Reads in Loops" are the pairs included in loops, i.e. PETs that are unique, intra-chromosomal, with a width between 5kb and 2Mb and anchor-mapped. If the fraction of "reads in loops" is low, but the proportion of "valid intra 5kb-2Mb" is high, this indicates a relatively low ChIP efficacy. In our case, all the percentages are comparable to other good quality HiChIP (see http://aryee.mgh.harvard.edu/hichipper/ for some QC reports of different HiChIP datasets).

3.2.3 Calling differential loops

To identify variations in chromatin 3D conformation due to heat shock, we will perform a differential analysis using the R package *diffloop*, which implements two strategies to

assess significance of variable DNA loops, either negative-binomial regression (from *edgeR* R package) or weighted t-test (from *limma* R package). For *diffloop* to properly perform statistical analyses, a minimum of two replicates per condition is needed (in case of a single sample per condition, see **Note 8**). The input files for *diffloop* are the `.interactions.all.mango` files generated by *hichipper*, therefore we create a new folder in which we copy all the interaction files:

```
cd $WORKDIR
mkdir Diffloop
mkdir Diffloop/input
all_dir=$(ls Hichipper/)
for dir in $all_dir; do
   cp Hichipper/$dir/*.interactions.all.mango Diffloop/input/
done
```

This creates first the `Diffloop/` folder and then the `input/` subfolder, next we create a variable (`all_dir`) that holds all the per-sample folder names that were created by *hichipper* and finally we iterate among all folders to copy the desired file into our `Diffloop/input/` location. Now we start R and load the required packages:

```
library(diffloop)
library(ggplot2)
library(ggrepel)
setwd("Diffloop")
```

To read the interactions files for the four analyzed samples, we use the `loopsMake.mango` function:

```
bed_dir <- paste(getwd(),"input", sep="/")
sample_names <-
c("Rad21_Rep1","Rad21_Rep2","HS_Rad21_Rep1","HS_Rad21_Rep2")
full <- loopsMake.mango(bed_dir, samples=sample_names)

dim(full)
     anchors  interactions      samples      colData      rowData
```

```
       51022        404794             4             2             1
```

Since we had two different sets of ChIP-Seq peaks, anchors were defined independently for treated and untreated samples. In order to create a single list of unique anchor regions, the `loopsMake.mango` function also merges nearby anchors (default gap value of 500bp). With the `dim` function we can have an overview of the resulting "loop" object (`full`), which consists of 5 different slots that store the information relative to the genomic location of unique loop anchor regions (`full@anchors` returns a summarized view of all the coordinates), the total number of interactions among all samples that have been read (`full@interactions` prints all the interactions), the names and grouping information of all samples (`full@colData`), the read pairs count supporting each interaction for each sample (`full@counts`) and, finally the loop width information (`full@rowData`). As summarized by `dim`, `full` consists of 4 samples, with a total of 51,022 unique loop anchors and 404,794 identified interactions. Without specifying anything but the input directory, the import function assigns all samples to the same group. Therefore, we set the correct type for each sample with the `updateLDGroups` function: we use "NT" and "HS" for untreated and heat-shock-treated samples, respectively; note that we also specify the labels order with the `levels` parameter. While self-ligated loops are removed in the final steps of *hichipper*, the import function can create new ones because of the merging of nearby loop anchors. Therefore, we will first remove these self-ligated loops and then filter out all non-significant interactions (as defined by *Mango* implementation in *hichipper*):

```
full <- updateLDGroups(full, factor(c("NT","NT","HS","HS"),
levels=c("NT","HS")))
full <- subsetLoops(full, full@rowData$loopWidth >= 5000)
full_qc <- mangoCorrection(full, FDR = 0.01)

dim(full)
```

```
     anchors  interactions       samples       colData       rowData
       51022        404734             4             2             1

dim(full_qc)
     anchors  interactions       samples       colData       rowData
       40921        120001             4             2             3
```

As shown by `dim`, the `mangoCorrection` function adds two columns in the rowData slot, which consist of the p-value and FDR calculated by *Mango*. As a final filtering step, we want to remove in each group loops that are strongly (PETs ≥ 5) present in one replicate, but absent in the other one (PETs = 0):

```
counts <- full_qc@counts
Rad21_dis <- ((counts[,"Rad21_Rep1"]>=5 &
counts[,"Rad21_Rep2"]==0)|(counts[,"Rad21_Rep2"]>=5 &
counts[,"Rad21_Rep1"]==0))
HS_Rad21_dis <- ((counts[,"HS_Rad21_Rep1"]>=5 &
counts[,"HS_Rad21_Rep2"]==0)|(counts[,"HS_Rad21_Rep2"]>=5 &
counts[,"HS_Rad21_Rep1"]==0)
qc_filt <- subsetLoops(full_qc, !(Rad21_dis | HS_Rad21_dis))

dim(qc_filt)
     anchors  interactions       samples       colData       rowData
       40822        118050             4             2             3
```

The resulting `qc_filt` object includes now a total of 118,050 unique interactions. To evaluate the variability among replicates and samples, we can calculate and plot the Principal Components using the following function:

```
pcaPlot(qc_filt) + geom_text_repel(aes(label=sample_names)) +
ggtitle("PC Plot with Size Factor Correction") +
theme(legend.position="none")
```

By exploring this plot (**Figure 2**), it is possible to see that the untreated Rad21 HiChIP replicates cluster together, while the HS-treated samples show greater variability. As expected, treated samples are clearly separated from untreated ones. Through the

`loopMetrics` functions, that counts the number of PETs supporting loops for each sample, we can evaluate the impact of the applied filters:

```
loopMetrics(full)
           Rad21_Rep1    Rad21_Rep2  HS_Rad21_Rep1  HS_Rad21_Rep2
 unique       1258821       1121515         647992        1246922

loopMetrics(qc_filt)
           Rad21_Rep1    Rad21_Rep2  HS_Rad21_Rep1  HS_Rad21_Rep2
 unique       1018178        937120         459087         895830
```

Once the significant loops have been retained, it is possible to perform the differential analysis and, since we are working with only two groups (NT and HS), we can use the *edgeR*-based `quickAssoc` function:

```
nt_hs_Rad21_res <- quickAssoc(qc_filt)

dim(nt_hs_Rad21_res)
     anchors  interactions       samples       colData       rowData
       40822        118050             4             2             8
```

The `quickAssoc` function creates a new "loop" object (`nt_hs_Rad21_res`), where the `rowData` has been extended with 5 additional columns to include the log2FC (with the Fold Change calculated as HS vs NT according to the labels order), the log2CPM, the linear regression value and the p-value and FDR calculated for each tested loop. In case of an experiment with a more complicated set up, it is necessary to use the `loopAssoc` function (refer to *diffloop* manual for more details). An additional function, `topLoops`, allows to subset the `nt_hs_Rad21_res` object by FDR. In our case, we use a 1% FDR cut-off and obtain 6,477 significant differential loops, with 4,696 increased and 1,781 decreased after heat shock treatment:

```
nt_hs_Rad21_res_sig <- topLoops(nt_hs_Rad21_res, FDR=0.01)

dim(nt_hs_Rad21_res_sig)
     anchors  interactions       samples       colData       rowData
```

```
        8763          6477              4              2              8
```

Finally, we write in a tab-delimited format the resulting significant differential loops. The `summary` function converts the "loop" object into a `data.frame` that can be saved using `write.table`:

```
write.table(summary(nt_hs_Rad21_res_sig),
"diffLoop_sig_Rad21.txt", row.names=F, sep="\t", quote=F)
```

3.3 Loop visualization

To visualize HiChIP loops, we can use the approach commonly adopted for Hi-C data: the entire Hi-C normalized matrices are shown as heatmaps and the pixels corresponding to significant loops are highlighted [13]. Here we will instead favor a representation that focuses only on the significant interactions and allows to easily explore their genomic and epigenomic landscape.

3.3.1 Plotting loops with diffloop

The *diffloop* package offers the possibility to plot a specific region starting directly from the "loop" object. The `loopPlot` function takes the "loop" object and the region to display as GRanges object (to date, only human hg19 and mouse mm9 are supported; default is hg19) and returns an arc view of the loops with both anchors within the selected region, with the arc thickness proportional to the number of PETs supporting the interaction. **Figure 3** depicts an example of Rad21-associated significant differential loops in each replicate, obtained as follows:

```
library(GenomicRanges)
regToPlot <-
GRanges(seqnames=c("3"),ranges=IRanges(start=c(193767907),end=c(
193860343)))
loopPlot(nt_hs_Rad21_res_sig, regToPlot)
```

The depicted genomic region contains a HS-induced gene (HES1) [6]: we clearly see that, after temperature stress, HES1 promoter increases contacts with distant regions, in

both replicates (see **Note 9**). The arcs can be colored to visually discriminate between the type of elements that are brought into contact by the loop (e.g. promoter-enhancer loops are reported red, promoter-promoter are orange…), once this information is provided (see *diffloop* vignette). Otherwise, they are all set to the default black color.

3.3.2 Exploring interactions with WashU Epigenome Browser

Although the plotting function from *diffloop* allows to easily display differences in loops, it is not possible to interactively browse the genome and load additional data. To overcome this limitation, we can use the *WashU Epigenome Browser* to visualize our HiChIP differential loops and compare them with other genomic tracks (like the BigWig of the ChIP-Seq experiments for Rad21 and H3K27ac in the same cell type and experimental conditions). To be loaded into the browser, loops need to be formatted in the longrange interaction format consisting of four tab-separated fields: the first three represent the first interacting region and the fourth contains the second interacting region and the associated score in a "chr:start-end,score" format (refer to https://eg.readthedocs.io/en/latest/tracks.html#longrange for additional information). Since we want to explore the differential loops, we will convert the `diffLoop_sig_Rad21.txt` produced by *diffloop* and use the logFC as the score (see **Note 10**). To generate the tracks, we exit the R environment and use `awk` from command line as follows:

```
cd $WORKDIR
awk 'NR>1{print "chr"$1,$2,$3,"chr"$4":"$5"-"$6","$14"}'
OFS="\t" DiffLoop/diffLoop_sig_Rad21.txt >
diffLoop_sig_Rad21.interactions.txt
```

At the *WashU Epigenome Browser* page, we first select the reference genome (Human → hg19 → Go). The browser will automatically load the repeat masker and the GENCODE v29 track; we can remove them by right-clicking on the track name and then select "Remove". Now we can upload our custom tracks. To do so, we need to select the

“Tracks” button and choose “Text Tracks”; this will open a new page where we first specify the file format (long-range text) and then we browse our local folder and select the appropriate file (`diffLoop_sig_Rad21.interactions.txt`). By default, the visualization method is a heatmap, but by right-clicking on the track name it is possible to change the view (we want either “Arc” or “Flatarc”) and also to set the primary color for positive (red) and the secondary for negative (blue) scores (i.e. significantly enriched and depleted loops after heat-shock treatment, respectively). Now we can load the BigWig files of the ChIP-Seq experiments. Again, we have to select the “Tracks” button, this time choosing the “View Local Tracks” option. As for the loop file, we first instruct the browser on the file format (BigWig) and then we select all four files together. For these files, we keep the default display mode (“Bar”). To discriminate between the histone mark and Rad21, we set the primary color for Rad21 (both NT and HS) as blue while we use orange color for the H3K27ac tracks. Finally, we need to add a gene annotation track, that can be found in the “Tracks” menu, in the “Annotation Tracks” section. Here we choose, among the Gene annotation tab, the RefSeq track. In the menu bar, it is possible to navigate the browser either by gene name or by coordinates. Finally, in the “Apps” menu, it is possible to take a screenshot of the region currently featured in the browser and to download it in svg or pdf format. **Figure 4** shows the screenshots of two genomic loci on chromosome 3 (chr3:45,258,804-45,383,015 in **4A**; chr3:193,767,907-193,860,343 in **4B**). Figure 4A depicts again the enriched loops after HS around the HES1 locus. Thanks to the additional epigenomic information we can observe that the distal regions (blue boxes) involved in loops with the HES1 promoter (orange box) are in fact enhancer regions, marked by H3K27ac, and that the enriched interactions are accompanied by an increased histone acetylation level after temperature stress. Figure 4B depicts an opposite situation, where a loop connecting the TMEM158 gene (orange box) with a Rad21 peak in an enhancer region (blue box) is lost after heat shock. In

parallel, Rad21 binding signal and especially the enhancer acetylation level are decreased. Accordingly, TMEM158 has been shown to be down-regulated in response to HS [6].

4. Notes

1. In the installation process of *hichipper*, a relatively old version of *diffloop* (v. 1.6) is installed. We recommend downloading from Bioconductor and manually installing the latest version (building from source is the best option). *ggrepel* (Automatically Position Non-Overlapping Text Labels with *ggplot2*) is used to optimize plots generated with ggplot2.
2. When dealing with a high number of samples it is convenient to prepare a sample info file, where each line represents one sample and reports the following information (in a TAB-delimited format): SRA run accession number, sample name, immunoprecipitated protein and replicate number. A possible file is given as example below:

```
SRR6206783  Rad21_Rep1  Rad21   1
SRR6206784  Rad21_Rep2  Rad21   2
SRR6206788  HS_Rad21_Rep1   HS_Rad21    1
SRR6206789  HS_Rad21_Rep2   HS_Rad21    2
```

This can be saved as `infofile.txt` in the `$WORKDIR` and then we can make use of either `for` or `while` loops to automatically download the FASTQ files, create the sample directory and then rename and move the FASTQ files in the appropriate folder. Below is an example `while` loop:

```
while IFS=$'\t' read -r SRA name IP REP; do
   fastq-dump --split-files -O fastq/ --gzip $SRA
   mkdir fastq/$name
   mv fastq/$SRA"_1.fastq.gz" fastq/$name/$name"_1.fastq.gz"
```

```
    mv fastq/$SRA"_2.fastq.gz" fastq/$name/$name"_2.fastq.gz"
done < infofile.txt
```

where the first line splits each line of the info file according to the field separator specified by `IFS` and saves each part into a variable. Next, we retrieve the FASTQ using the SRA run accession number saved in `$SRA` (e.g. `SRR6206783`), we create a folder named as the sample (e.g. `Rad21_Rep1`) and finally we move the FASTQ files into this new folder and rename them according to the sample name (e.g. `Rad21_Rep1_2.fastq.gz` and `Rad21_Rep1_2.fastq.gz`).

3. To liftover, *i.e.* to convert the genomic coordinates of the four BigWig files to hg19, we first move them to a folder named `BigWig`. Then, we download the UCSC chain file relative to the hg38 to hg19 conversion using `wget` and the link provided below. Finally, the coordinates conversion can be performed with *CrossMap*. All this can be performed by running the following commands:

```
mkdir BigWig
mv *.bw BigWig/
cd BigWig
wget
http://hgdownload.soe.ucsc.edu/goldenPath/hg38/liftOver/hg38To
Hg19.over.chain.gz
CrossMap.py bigwig hg38ToHg19.over.chain GSE105028_Rad21_HS.bw
HS_Rad21
```

With the last command we launch the BigWig module of *CrossMap* for converting the reference genome. The second argument tells *CrossMap* which chain to use for the coordinate conversion, the third is the input file and the last is the output file name.

4. Before starting the preprocessing with *HiC-Pro*, it is highly recommended to first check the quality of the FASTQ files using e.g. FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) or multiQC (https://multiqc.info). FastQC performs several quality checks and can highlight the

presence of contamination or read adapters, quality loss and high duplication levels. According to the library preparation protocol that is used in the HiChIP experiment, resulting reads might include adapter sequences. In particular, if the original protocol from [1] is followed, library preparation includes the use of Tn5 transposase, possibly resulting in the presence of Nextera Transposase Sequence in the sequenced reads, as in ATAC-seq. This information can be found in the "Adapter Content" section of FastQC html report. To avoid issues in the alignment steps, it is necessary to trim reads in a paired-end setup (e.g. using *Trimmomatic* [17] or similar tools) to remove the adapter sequences. In the case of Lyu *et al.* [6] HiChIP samples, library preparation was performed following the standard Hi-C protocol, therefore we do not have to take care of Tn5 specific sequences.

5. In this particular case, ChIP-Seq experiments have been performed in paired-end mode (while typically done in single end) and this has to be taken into account when aligning with *bowtie*. For the peak calling step, *MACS2* includes the `-f/--format` parameter that allows to specify the format of the input file. If nothing is specified and the input is a paired-end BAM (as in our case), the input will be treated as a single end BAM, i.e. only the 5' end reads of each pair will be considered. Only if `-f BAMPE` is specified, *MACS2* will use the actual insert size of read pairs to build the fragment pileup instead of using position and orientation of the aligned reads to predict the fragment size. Additionally, since duplicated pairs were previously identified and removed with *samtools*, we set the `--keep-dup` parameter to `all` in order to retain the 5' reads that share the same position, thus maximizing the number of reads used for peak calling. For more details, see https://github.com/taoliu/MACS.
6. If support ChIP-Seq peaks for loop calling are not available, *hichipper* implements a standard peak calling procedure to define anchor regions starting directly from HiChIP data. ChIP peaks can be inferred using either all available reads from the HiChIP

experiment (`ALL`) or limit to those resulting from self-ligation (`SELF`). Additionally, either all samples can be used to generate a unique peak file (`COMBINED`) or each sample can be treated individually (`EACH`). The combination of these two choices results into four possible strings (`COMBINED,ALL` - `COMBINED,SELF` - `EACH,ALL` - `EACH,SELF`) that can be specified in the sample description file in the peaks section, see an example below:

```
peaks:
 - EACH,ALL
resfrags:
 - /Programs/HiC-Pro_2.11.1/annotation/DpnII_resfrag_hg19.bed
hicpro_output:
 - HiC_Pro
```

hichipper uses *MACS2* [16] as peak caller and through the `--macs2-genome` and `--macs2-string` parameters it is possible to modify the ChIP peak calling parameters. To address the restriction-fragment bias, *MACS2* background model has been modified in order to account for the distance of a putative ChIP peak from the restriction site.

7. As an alternative to the sample description file method, it is possible to run *hichipper* setting each parameter directly in the `call` mode, thus specifying through command line the right combination of HiChIP/ChIP-Seq input files. An example of this mode is provided below:

```
cd $WORKDIR
hichipper --input-vi
HiC_Pro/hic_results/data/Rad21_Rep1/Rad21_Rep1.allValidPairs -
l 50 -p ChIP_peaks/Rad21_peaks.noBL.narrowPeak --restriction-
frags /Programs/HiC-
Pro_2.11.1/annotation/DpnII_resfrag_hg19.bed --basic-qc --out
RAD21_1 call
```

With this approach, we need to make explicit all the required parameters. Note that, to date, this mode cannot produce the html file but only a simple QC report (`--basic-qc`).

8. As explained in *diffloop* vignette (http://aryee.mgh.harvard.edu/diffloop-paper/vignette/diffloop_vignette.html), performing the differential analysis with only one sample per group is not supported, but it is still possible to calculate a simple count difference among conditions or samples.
9. The `loopPlot` function displays the same set of loops in all samples, even when the loop count is 0 (in this case the minimum possible thickness is used). This has to be taken in consideration when visually exploring the plot.
10. *hichipper* provides the possibility to generate a longrange track file for each sample suitable for visualization in *WashU Epigenome Browser* by adding the `--make-washu` option when running *hichipper*. To properly load these track files, it is necessary to create a `.json` file (please refer to https://eg.readthedocs.io/en/latest/local.html for further information on how to load local tracks). Note that with such approach the tracks will display all loops identified in each sample and not the differential loops.

Acknowledgments

This work was supported by PRIN 2017 Project 2017HWTP2K of the Italian Ministry of Education, University and Research (to M.F.).

References

1. Mumbach MR, Rubin AJ, Flynn RA, Dai C, Khavari PA, Greenleaf WJ, Chang HY (2016) HiChIP: efficient and sensitive analysis of protein-directed genome architecture. Nat Methods 13 (11):919-922. doi:10.1038/nmeth.3999

2. Schmitt AD, Hu M, Ren B (2016) Genome-wide mapping and analysis of chromosome architecture. Nat Rev Mol Cell Biol 17 (12):743-755. doi:10.1038/nrm.2016.104

3. de Wit E, de Laat W (2012) A decade of 3C technologies: insights into nuclear organization. Genes Dev 26 (1):11-24. doi:10.1101/gad.179804.111

4. Lareau CA, Aryee MJ (2018) hichipper: a preprocessing pipeline for calling DNA loops from HiChIP data. Nat Methods 15 (3):155-156. doi:10.1038/nmeth.4583

5. Bhattacharyya S, Chandra V, Vijayanand P, Ay F (2019) Identification of significant chromatin contacts from HiChIP data by FitHiChIP. Nat Commun 10 (1):4221. doi:10.1038/s41467-019-11950-y

6. Lyu X, Rowley MJ, Corces VG (2018) Architectural Proteins and Pluripotency Factors Cooperate to Orchestrate the Transcriptional Response of hESCs to Temperature Stress. Mol Cell 71 (6):940-955 e947. doi:10.1016/j.molcel.2018.07.012

7. Servant N, Varoquaux N, Lajoie BR, Viara E, Chen CJ, Vert JP, Heard E, Dekker J, Barillot E (2015) HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. Genome Biol 16:259. doi:10.1186/s13059-015-0831-x

8. Phanstiel DH, Boyle AP, Heidari N, Snyder MP (2015) Mango: a bias-correcting ChIA-PET analysis pipeline. Bioinformatics 31 (19):3092-3098. doi:10.1093/bioinformatics/btv336

9. Lareau CA, Aryee MJ (2018) diffloop: a computational framework for identifying and analyzing differential DNA loops from sequencing data. Bioinformatics 34 (4):672-674. doi:10.1093/bioinformatics/btx623

10. Zhou X, Maricque B, Xie M, Li D, Sundaram V, Martin EA, Koebbe BC, Nielsen C, Hirst M, Farnham P, Kuhn RM, Zhu J, Smirnov I, Kent WJ, Haussler D, Madden PAF, Costello JF, Wang T (2011) The Human Epigenome Browser at Washington University. Nat Methods 8 (12):989-990. doi:10.1038/nmeth.1772

11. Merkenschlager M, Nora EP (2016) CTCF and Cohesin in Genome Folding and Transcriptional Gene Regulation. Annu Rev Genomics Hum Genet 17:17-43. doi:10.1146/annurev-genom-083115-022339

12. SRA Toolkit Development Team. http://ncbi.github.io/sra-tools/.

13. Forcato M, Bicciato S (2019) Computational analysis of Hi-C data. In: Methods in Molecular Biology (in press). Capturing Chromosome Conformation.

14. Feng J, Liu T, Qin B, Zhang Y, Liu XS (2012) Identifying ChIP-seq enrichment using MACS. Nat Protoc 7 (9):1728-1740. doi:10.1038/nprot.2012.101

15. Harvard Chan Bioinformatics Core (HBC). https://hbctraining.github.io/Intro-to-ChIPseq/lessons/05_peak_calling_macs.html.

16. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS (2008) Model-based analysis of ChIP-Seq (MACS). Genome Biol 9 (9):R137. doi:10.1186/gb-2008-9-9-r137

17. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics (Oxford, England) 30 (15):2114-2120. doi:10.1093/bioinformatics/btu170

Figure and Table Legends

Figure 1. Workflow of HiChIP data analysis presented in this chapter. Each box depicts in bold the main steps, while the tool used to perform each step is reported in the lower part of each box.

Figure 2. Principal Component Plot of HiChIP data according to the identified loops. PCA plot generated starting from the “loop” object after the *diffloop* filtering steps.

Figure 3. Visualization of loops with *diffloop*. Arc representation of significant differential loops obtained with *diffloop* and found in the 0.09Mb region located on chr3:193,767,907−193,860,343. The plot was generated with the `plotLoops` function. The upper panels show the loops identified in each replicate of NT Rad21 HiChIP experiment, while the lower panels show the same region for HS-treated Rad21 HiChIP data. Thickness of each arc line is proportional to the number of read pairs supporting the loop. Boxes were manually added to easily discriminate between NT and HS panels.

Figure 4. Visualization of differential topology with *WashU Epigenome Browser*. Arc representation of differential 3D architecture before and after HS-treatment in Rad21 HiChIP experiment (data obtained with *diffloop*) together with BigWig tracks for H3K27ac and Rad21 ChIP-Seq in the same experimental condition in two genomic regions. Arc colors represent the logFC calculated by *diffloop*, with blue indicating a loss and red a gain of interaction. H3K27ac ChIP-Seq tracks are represented in orange, while Rad21 tracks are depicted in blue. **(A)** 0.09Mb region on chromosome 3 (chr3:193,767,907−193,860,343) characterized by the gain of Rad21 mediated loops that bring in proximity a HS-activated gene (HES1, orange box) and two enhancer regions activated upon HS (blue boxes). **(B)** 0.12Mb window

on chromosome 3 (chr3:193,767,907-193,860,343), where a Rad21 associated loop connects TMEM158 (orange box) with an enhancer region adjacent to a Rad21 ChIP-Seq peak (blue box). This interaction, together with the Rad21 ChIP-Seq peaks, is lost after HS. Interestingly, the enhancer region is also de-activated in response to the treatment and all these changes associate with the reported down-regulation of TMEM158 expression. Orange and blue boxes were manually added for visualization purposes.

Table 1. HiChIP datasets. List of files for controls (NT) and heat shock treated (HS) HiChIP samples with SRA accessions.

Table 2. Rad21 ChIP-Seq datasets. List of files for controls (NT) and heat shock treated (HS) Rad21 ChIP-Seq experiments with SRA accessions.

Table 3. BigWig files for ChIP-Seq datasets. List of BigWig files for controls (NT) and heat shock treated (HS) Rad21 and H3K27ac ChIP-Seq with GEO accessions.

Table 4. Overall statistics after *HiC-Pro* preprocessing. The *Total Reads* sequenced per sample are reported together with the mapping statistics per mate (R1 and R2). Then the table reports the number of all mapped read pairs that are then filtered and finally de-duplicated. Percentages were calculated using the *Total Reads* as 100%.

Table 5. Summary quality metrics. Summary quality metrics as reported in the QC html file generated by *hichipper* together with the number of total and significant (FDR ≤ 0.01) loops identified per sample. *Reads in Anchors* represents the pairs that are unique, mapped and with both reads overlapping with a loop anchor (including also reads falling on the same restriction fragment). *Mapped unique quality valid* are the de-duplicated valid pairs

determined by *HiC-Pro*, which are then filtered according to the genomic location (*Valid intra*) and distance (*Valid intra 5kb-2Mb*) of the interacting loci. Finally, *Reads in Loops* is the number of pairs that are included in loops: unique, intra-chromosomal, with a width between 5kb and 2Mb and with both mates mapped to anchors. Percentages were calculated using the *Total PETs* as 100%.

Tables

Table 1

Target protein	Condition	Rep	SRA run Accession	Link to SRA
Rad21	NT	Rep1	SRR6206783	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6206783
		Rep2	SRR6206784	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6206784
Rad21	HS	Rep1	SRR6206788	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6206788
		Rep2	SRR6206789	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6206789

Table 2

Target protein	Condition	SRA run Accession	Link to SRA
IgG	NT	SRR6177932	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6177932
	HS	SRR6177933	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6177933
Rad21	NT	SRR6177934	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6177932
	HS	SRR6177936	https://www.ncbi.nlm.nih.gov/sra/?term=SRR6177936

Table 3

Target protein	Condition	File name	Link to GEO
Rad21	NT	GSE105028_Rad21_NT.bw	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE105028
	HS	GSE105028_Rad21_HS.bw	
H3K27ac	NT	GSE105028_H3K27ac_NT.bw	
	HS	GSE105028_H3K27ac_HS.bw	

Table 4

		Rad21_Rep1		Rad21_Rep2		HS_Rad21_Rep1		HS_Rad21_Rep2	
		number	%	number	%	number	%	number	%
Total Reads		187,751,149	100	140,089,640	100	93,931,067	100	149,416,527	100
Mapped	R1	177,307,925	94.4	131,253,091	93.7	88,056,906	93.7	140,421,363	94
	R2	168,056,676	89.5	123,855,705	88.4	80,640,374	85.9	130,385,917	87.3
Full	R1	154,788,612	82.4	112,615,926	80.4	79,351,929	84.5	126,017,070	84.3
	R2	150,853,459	80.3	110,885,068	79.2	72,947,043	77.7	118,023,406	79
Trimmed	R1	22,519,313	12	18,637,165	13.3	8,704,977	9.3	14,404,293	9.6
	R2	17,203,217	9.1	12,970,637	9.3	7,693,331	8.2	12,362,511	8.3
Paired pairs		130,113,269	69.3	96,395,054	68.8	6,3291,926	67.4	102,941,739	68.9
Valid pairs		80,313,959	42.8	50,146,034	35.8	37,699,101	40.1	60,464,247	40.5
De-duplicated valid pairs		38,054,344	20.3	35,611,266	25.4	33,133,951	35.3	45,826,888	30.6

Table 5

	Rad21_Rep1		Rad21_Rep2		HS_Rad21_Rep1		HS_Rad21_Rep2	
	number	%	number	%	number	%	number	%
Total PETs	187,751,149	100	140,089,640	100	93,931,067	100	149,416,527	100
Reads in Anchors	99,421,968	53	75,966,757	54.2	38,259,408	40.7	79,673,145	53.3
Mapped unique quality valid	38,054,344	20.3	35,611,266	25.4	33,133,951	35.3	45,826,888	30.6
Valid intra	35,682,744	19	33,447,924	23.9	29,173,403	31.1	41,812,852	28
Valid intra 5kb-2Mb	22,093,351	11.8	18,686,254	13.3	18,437,504	19.6	26,170,082	17.5
Reads in Loops	1,259,385	0.7	1,121,933	0.8	648,104	0.7	1,247,108	0.8
Total Loops	195,864		176,677		188,647		251,254	
Loops (FDR ≤ 0.01)	60,366		55,133		57,630		70,823	