
HiChiP Documentation

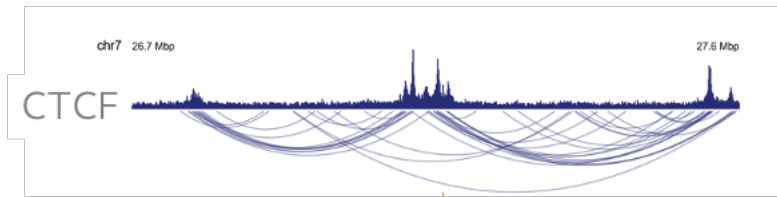
Release 0.1

Dovetail

Aug 01, 2023

CONTENTS:

1	Overview	3
1.1	Before you begin	3
1.2	Pre-Alignment	5
1.3	From fastq to final valid pairs bam file	6
1.4	Library QC	11
1.5	Generating Contact Matrix	24
1.6	HiChIP Loop Calling	29
1.7	HiChIP Comparative Analyses	37
1.8	Plotting HiChIP Data	38
1.9	Calling 1D peaks with MACS2 on HiChIP data	45
1.10	HiChIP Data Sets	47
1.11	Support	50
2	Indices and tables	51



OVERVIEW

- The Dovetail™ HiChIP MNase Kit combines the benefits of ChIP-seq with Hi-C, a proximity ligation method that captures long-range interactions using standard Illumina paired-end sequencing, enabling researchers to query protein-directed chromatin conformation mediated by specific proteins of interest.
- Key benefits of HiChIP:
 - Capture ChIP-seq and Hi-C data together in a single library
 - Map chromatin interactions at nucleosome level resolution
- The unique combination of the Dovetail™ Micro-C Proximity Ligation Assay with the Dovetail HiChIP approach enables the use of micrococcal nuclease (MNase) to fragment chromatin uniformly and without sequence bias prior to proximity ligation, eliminating the need for finicky sonication procedures and offering the maximal resolution (down to mono-nucleosome size) of chromatin interactions.
- Enrichment of protein-directed chromatin features enables high-resolution contact map generation with less read depth. Compared to a high resolution restriction enzyme-based Hi-C, Dovetail HiChIP data enables visualization of higher-order chromatin features, such as loops and chromatin interactions, at a fraction of the read depth leading to significant sequencing costs savings.
- This guide will take you step by step on how to QC your HiChIP library, how to interpret the QC results and how to call and plot significant interactions. If you don't yet have a sequenced HiChIP library and you want to get familiar with the data, you can download HiChIP sequences libraries from our publicly available [data sets](#).
- The QC process starts with aligning the reads to a reference genome then retaining high quality mapped reads. From there the mapped data will be used to generating a pairs file with pairtools, which categorizes pairs by read type and insert distance, this step both flags and removes PCR duplicates. Once pairs are categorized, counts of each class are summed and reported.
- If this is your first time following this tutorial, please check the [Before you begin page](#) first.

1.1 Before you begin

1.1.1 Have a copy of the HiChIP scripts on your machine:

Clone this repository:

```
git clone https://github.com/dovetail-genomics/HiChIP.git
```

And make the `enrichment_stats.sh` script executable:

```
chmod +x ./HiChIP/enrichment_stats.sh
```

1.1.2 Dependencies

Make sure that the following dependencies are installed:

- `pysam`
- `tabulate`
- `bedtools`
- `deeptools`
- `matplotlib`
- `pandas`
- `numpy`
- `bwa`
- `pairtools`
- `samtools`

If you are facing any issues with the installation of any of the dependencies, please contact the supporter of the relevant package.

`python3` and `pip3` are required, if you don't already have them installed, you will need `sudo` privileges.

- Update and install `python3` and `pip3`:

```
sudo apt-get update
sudo apt-get install python3 python3-pip
```

- To set `python3` and `pip3` as primary alternative:

```
sudo update-alternatives --install /usr/bin/python python /usr/bin/python3 1
sudo update-alternatives --install /usr/bin/pip pip /usr/bin/pip3 1
```

If you are working on a new machine and don't have the dependencies, you can use the `installDep.sh` script in this repository for updating your instance and installing the dependencies and `python3`. This process will take approximately 10' and requires `sudo` privileges. The script was tested on Ubuntu 18.04 with the latest version as of 04/11/2020

If you choose to run the provided installation script you will first need to set the permission to the file:

```
chmod +x ./HiChiP/installDep.sh
```

And then run the installation script:

```
./HiChiP/installDep.sh
```

Remember!

Once the installation is completed, sign off and then sign back to your instance to refresh the database of applications.

1.1.3 Input files

For this tutorial you will need:

- **fastq files** R1 and R2, either fastq or fastq.gz are acceptable
- **reference in a fasta file format**, e.g. hg38
- **peak calls from ChIP-seq experiment** (e.g. your own experiment or ENCODE gold standard in bed or narrow-peak format, as explained [here](#)), more details and links to ENCODE files can be found [here](#).

If you don't already have your own input files or want to run a test on a small data set, you can download sample fastq files from the [HiChIP Data Sets section](#). The 2M data set is suitable for a quick testing of the instructions in this tutorial.

The following files are suitable for testing, you can download them as follows:

```
wget https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/HiChIP_CTCF_2M_R1.fastq.gz
wget https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/HiChIP_CTCF_2M_R2.fastq.gz
wget https://www.encodeproject.org/files/ENCFF017XLW/@download/ENCFF017XLW.bed.gz
```

For zipped bed files, unzip them after download is completed (no need to unzip fastq.gz files)

Example:

```
gunzip ENCFF017XLW.bed.gz
```

1.2 Pre-Alignment

For downstream steps you will need a genome file, genome file is a tab delimited file with chromosome names and their respective sizes. If you don't already have a genome file follow these steps:

1. Generate an index file for your reference, a reference file with only the main chromosomes should be used (e.g. without alternative or unplaced chromosomes).

Command:

```
samtools faidx <ref.fasta>
```

Example:

```
samtools faidx hg38.fasta
```

Faidx will index the ref file and create <ref.fasta>.fai on the reference directory.

2. Use the index file to generate the genome file by printing the first two columns into a new file.

Command:

```
cut -f1,2 <ref.fasta.fai> > <ref.genome>
```

Example:

```
cut -f1,2 hg38.fasta.fai > hg38.genome
```

In line with the 4DN project guidelines and from our own experience optimal alignment results are obtained with Burrows-Wheeler Aligner (bwa). Prior to alignment, generate a bwa index file for the chosen reference.

```
bwa index <ref.fasta>
```

Example:

```
bwa index hg38.fasta
```

No need to specify an output path, the bwa index files are automatically generated at the reference directory. Please note that this step is time consuming, however you need to run it only once for a reference.

To avoid memory issues, some of the steps require writing temporary files into a temp folder, please generate a temp folder and remember its full path. Temp files may take up to x3 of the space that the fastq.gz files are taking, that is, if the total volume of the fastq files is 5Gb, make sure that the temp folder can store at least 15Gb.

Command:

```
mkdir <full_path/to/tmpdir>
```

Example:

```
mkdir /home/ubuntu/ebs/temp
```

In this example the folder *temp* will be generated on a mounted volume called *ebs* on a user account *ubuntu*.

1.3 From fastq to final valid pairs bam file

fastq to final valid pairs bam file - for the impatient!

If you just want to give it a shot and run all the alignment and filtering steps without going over all the details, we made a shorter version for you, with all the steps piped, outputting a final bam file with its index file and a dup stats file, otherwise move to the next section *fastq to final valid pairs bam file - step by step*

Command:

```
bwa mem -5SP -T0 -t<cores> <ref.fa> <HiChiP.R1.fastq.gz> <HiChiP.R2.fastq.gz> | \
pairtools parse --min-mapq 40 --walks-policy 5unique \
--max-inter-align-gap 30 --nproc-in <cores> --nproc-out <cores> --chroms-path <ref.
↪genome> | \
pairtools sort --tmpdir=<full_path/to/tmpdir> --nproc <cores>|pairtools dedup --nproc-in
↪<cores> \
--nproc-out <cores> --mark-dups --output-stats <stats.txt>|pairtools split --nproc-in
↪<cores> \
--nproc-out <cores> --output-pairs <mapped.pairs> --output-sam -|samtools view -bS -@
↪<cores> | \
samtools sort -@<cores> -o <mapped.PT.bam>;samtools index <mapped.PT.bam>
```

Example:

```
bwa mem -5SP -T0 -t16 hg38.fasta HiChiP_CTCF_2M_R1.fastq.gz HiChiP_CTCF_2M_R2.fastq.gz |
↪pairtools parse --min-mapq 40 --walks-policy 5unique --max-inter-align-gap 30 --nproc-
↪in 8 --nproc-out 8 --chroms-path hg38.genome | pairtools sort --tmpdir=/home/ubuntu/
↪ebs/temp/ --nproc 16|pairtools dedup --nproc-in 8 --nproc-out 8 --mark-dups --output-
↪stats stats.txt|pairtools split --nproc-in 8 --nproc-out 8 --output-pairs mapped.pairs
↪--output-sam -|samtools view -bS -@16 | samtools sort -@16 -o mapped.PT.bam;samtools
↪index mapped.PT.bam
```



The full command above, with 2M read pairs on an Ubuntu 18.04 machine with 16 CPUs and 64GiB was completed in less than 5 minutes. On the same machine type.

1.3.1 fastq to final valid pairs bam file - step by step

Alignment

Now that you have a genome file, index file and a reference fasta file you are all set to align your HiChiP library to the reference. Please note the specific settings that are needed to map mates independently and for optimal results with our proximity library reads.

Parameter	Alignment function
mem	set the bwa to use the BWA-MEM algorithm, a fast and accurate alignment algorithm optimized for sequences in the range of 70bp to 1Mbp
-5	for split alignment, take the alignment with the smallest coordinate (5' end) as primary, the mapq assignment of the primary alignment is calculated independent of the 3' alignment
-S	skip mate rescue
-P	skip pairing; mate rescue performed unless -S also in use
-T0	The T flag set the minimum mapping quality of alignments to output, at this stage we want all the alignments to be recorded and thus T is set up to 0, (this will allow us to gather full stats of the library, at later stage we will filter the alignments by mapping quality)
-t	number of threads, default is 1. Set the numbers of threads to not more than the number of cores that you have on your machine (If you don't know the number of cores, used the command <code>lscpu</code> and multiply Thread(s) per core x Core(s) per socket x Socket(s))
*.fasta or *.fa	Path to a reference file, ending with .fa or .fasta, e.g, hg38.fasta
*.fastq or *.fastq.gz	Path to two fastq files; path to read 1 fastq file, followed by fastq file of read 2 (usually labeled as R1 and R2, respectively). Files can be in their compressed format (.fastq.gz) or uncompressed (.fastq). In case your library sequence is divided to multiple fastq files, you can use a process substitution < with the cat command (see example below)
-o	sam file name to use for output results [stdout]. You can choose to skip the -o flag if you are piping the output to the next command using ' '

Bwa mem will output a sam file that you can either pipe or save to a path using -o option, as in the example below (please note that version 0.7.17 or higher should be used, older versions do not support the -5 flag)

Command:

```
bwa mem -5SP -T0 -t<threads> <ref.fasta> <HiChiP_R1.fastq> <HiChiP_R2.fastq> -o <aligned.
↪sam>
```

Example (one pair of fastq files):

```
bwa mem -5SP -T0 -t16 hg38.fasta HiChiP_CTCF_2M_R1.fastq.gz HiChiP_CTCF_2M_R2.fastq.gz -
↪o aligned.sam
```

Example (multiple pairs of fastq files):

```
bwa mem -5SP -T0 -t16 hg38.fasta <(zcat file1.R1.fastq.gz file2.R1.fastq.gz file3.R1.
↪fastq.gz)> <(zcat file1.R2.fastq.gz file2.R2.fastq.gz file3.R2.fastq.gz)> -o aligned.sam
```

Recording valid ligation events

We use the `parse` module of the `pairtools` pipeline to find ligation junctions in HiChIP (and other proximity ligation) libraries. When a ligation event is identified in the alignment file the `pairtools` pipeline will record the outer-most (5') aligned base pair and the strand of each one of the paired reads into `.pairsam` file (pairsam format captures SAM entries together with the Hi-C pair information). In addition, it will also assign a pair type for each event. e.g. if both reads aligned uniquely to only one region in the genome, the type UU (Unique-Unique) will be assigned to the pair. The following steps are necessary to identify the high quality valid pairs over low quality events (e.g. due to low mapping quality):

`pairtools parse` options:

Parameter	Value	Function
<code>min-mapq</code>	40	Mapq threshold for defining an alignment as a multi-mapping alignment. Alignment with mapq <40 will be marked as type M (multi)
<code>walks-policy</code>	5unique	Walks is the term used to describe multiple ligations events, resulting three alignments (instead of two) for a read pair. However, there are cases in which three alignment in read pairs are the result of one ligation event, pairtool parse can rescue this event. walks-policy is the policy for reporting un-rescuable walk. 5unique is used to report the 5'-most unique alignment on each side, if present (one or both sides may map to different locations on the genome, producing more than two alignments per DNA molecule)
<code>max-inter-align-gap</code>	30	In cases where there is a gap between alignments, if the gap is 30 or smaller, ignore the gap, if the gap is >30bp, mark as "null" alignment
<code>nproc-in</code>	integer, e.g. 16	pairtools has an automatic-guess function to identify the format of the input file, whether it is compressed or not. When needed, the input is decompressed by <code>bgzip/lz4c</code> . The option <code>nproc-in</code> set the number of processes used by the auto-guessed input decompressing command, if not specified, default is 3
<code>nproc-out</code>	integer, e.g. 16	pairtools automatic-guess the desired format of the output file (compressed or not compressed, based on file name extension). When needed, the output is compressed by <code>bgzip/lz4c</code> . The option <code>nproc-out</code> set the number of processes used by the auto-guessed output compressing command, if not specified, default is 8
<code>chroms-path</code>		path to <code>.genome</code> file, e.g. <code>hg38.genome</code>
<code>*.sam</code>		path to sam file used as an input. If you are piping the input (stdin) skip this option
<code>*pairsam</code>		name of pairsam file for writing output results. You can choose to skip and pipe the output directly to the next command (<code>pairtools sort</code>)

`pairtools parse` command example for finding ligation events:

Command:

```
pairtools parse --min-mapq 40 --walks-policy 5unique --max-inter-align-gap 30 --nproc-in
↪ <cores> \
--nproc-out <cores> --chroms-path <ref.genome> <aligned.sam> > <parsed.pairsam>
```

Example:

```
pairtools parse --min-mapq 40 --walks-policy 5unique --max-inter-align-gap 30 --nproc-in ↪
↪ 8 --nproc-out 8 --chroms-path hg38.genome aligned.sam > parsed.pairsam
```

At the parsing step, pairs will be flipped such that regardless of read1 and read2, pairs are always recorded with first side of the pair having the lower genomic coordinates.

Sorting the pairsam file

The parsed pairs are then sorted using *pairtools sort*

`pairtools sort` options:

Parameter	Function
<code>--tmpdir</code>	Provide a full path to a temp directory. A good rule of thumb is to have a space available for this directory at a volume of x3 of the overall volume of the fastq.gz files. Using a temp directory will help avoid memory issues
<code>--nproc</code>	Number of processes to split the sorting work

Command:

```
pairtools sort --nproc <cores> --tmpdir=<path/to/tmpdir> <parsed.pairsam> > <sorted.
↪pairsam>
```

Example:

```
pairtools sort --nproc 16 --tmpdir=/home/ubuntu/ebs/temp/ parsed.pairsam > sorted.
↪pairsam
```

Important!

Please note that an absolute path for the temp directory is required for `pairtools sort`, e.g. path of the structure `~/ebs/temp/` or `./temp/` will not work, instead, something of this sort is needed `/home/user/ebs/temp/`

Removig PCR duplicates

`pairtools dedup` detects molecules that could be formed via PCR duplication and tags them as “DD” pair type. These pairs should be excluded from downstream analysis. Use the `pairtools dedup` command with the `--output-stats` option to save the dup stats into a text file.

`pairtools dedup` options:

Parameter	Function
<code>--mark-dups</code>	If specified, duplicate pairs are marked as DD in “pair_type” and as a duplicate in the sam entries
<code>--output-stats</code>	Output file for duplicate statistics. Please note that if a file with the same name already exists, it will be opened in the append mode

Command:

```
pairtools dedup --nproc-in <cores> --nproc-out <cores> --mark-dups --output-stats <stats.
↪txt> \
--output <dedup.pairsam> <sorted.pairsam>
```

Example:

```
pairtools dedup --nproc-in 8 --nproc-out 8 --mark-dups --output-stats stats.txt --output-
↳ dedup.pairsam sorted.pairsam
```

Generate .pairs and bam files

The `pairtools split` command is used to split the final `.pairsam` into two files: `.sam` (or `.bam`) and `.pairs` (`.pairsam` has two extra columns containing the alignments from which the HiChIP pair was extracted, these two columns are not included in `.pairs` files)

`pairtools split` options:

Parameter	Function
<code>--output-pairs</code>	Output pairs file. If the path ends with <code>.gz</code> or <code>.lz4</code> the output is <code>pbgzip/lz4c</code> -compressed. If you wish to pipe the command and output the pairs files to stdout use <code>-</code> instead of file name
<code>--output-sam</code>	Output sam file. If the file name extension is <code>.bam</code> , the output will be written in bam format. If you wish to pipe the command, use <code>-</code> instead of a file name. please note that in this case the sam format will be used (and can be later converted to bam file e.g. with the command <code>samtools view -bS -@16 -o temp.bam</code>)

Command:

```
pairtools split --nproc-in <cores> --nproc-out <cores> --output-pairs <mapped.pairs> \
--output-sam <unsorted.bam> <dedup.pairsam>
```

Example:

```
pairtools split --nproc-in 8 --nproc-out 8 --output-pairs mapped.pairs --output-sam
↳ unsorted.bam dedup.pairsam
```

The `.pairs` file can be used for generating *contact matrix*

Generating the final bam file

For downstream steps, the bam file should be sorted, using the command `samtools sort`

`samtools sort` options:

Parameter	Function
<code>-@</code>	number of threads to use
<code>-o</code>	file name. Write final output to FILE rather than standard output
<code>-T</code>	path to temp file. Using a temp file will help avoiding memory issues

Command:

```
samtools sort -@<threads> -T <path/to/tmpdir/tempfile.bam> -o <mapped.PT.bam> <unsorted.
↳ bam>
```

Example:

```
samtools sort -@16 -T /home/ubuntu/ebs/temp/temp.bam -o mapped.PT.bam unsorted.bam
```

For future steps an index (.bai) of the bam file is also needed. Index the bam file:

Command:

```
samtools index <mapped.PT.bam>
```

Example:

```
samtools index mapped.PT.bam
```

The mapped.PT.bam is the final bam file that will be used downstream steps.

The above steps resulted in multiple intermediate files, to simplify the process and avoid intermediate files, you can pipe the steps as in the example above (*fastq to final valid pairs bam file - for the impatient*)

1.4 Library QC

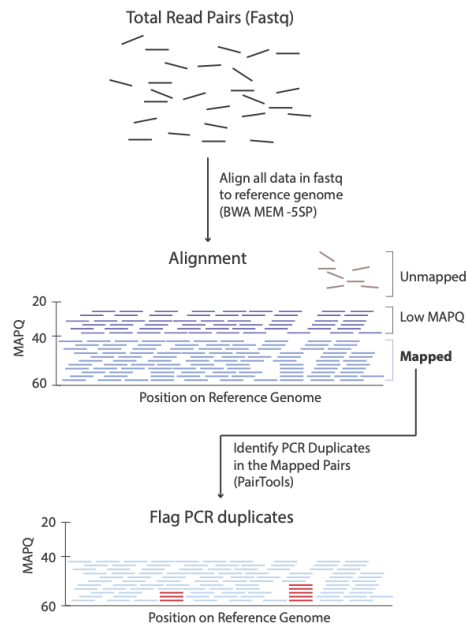
1.4.1 Proximity-ligation assessment

At step *Removing PCR duplicates* you used the flag `-output-stats`, generating a stats file in addition to the pairsam output (e.g. `-output-stats stats.txt`). The stats file is an extensive output of pairs statistics as calculated by pairtools, including total reads, total mapped, total dups, total pairs for each pair of chromosomes etc'. Although you can use directly the pairtools stats file as is to get informed on the quality of the HiChIP library, we find it easier to focus on a few key metrics. We include in this repository the script `get_qc.py` that summarize the paired-tools stats file and present them in percentage values in addition to absolute values.

The images below explains how the values on the QC report are calculated:

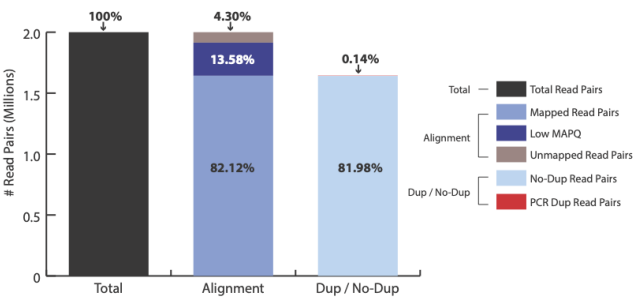
I. Aligning and filtering to remove low mapping quality and PCR duplicate read pairs

Process



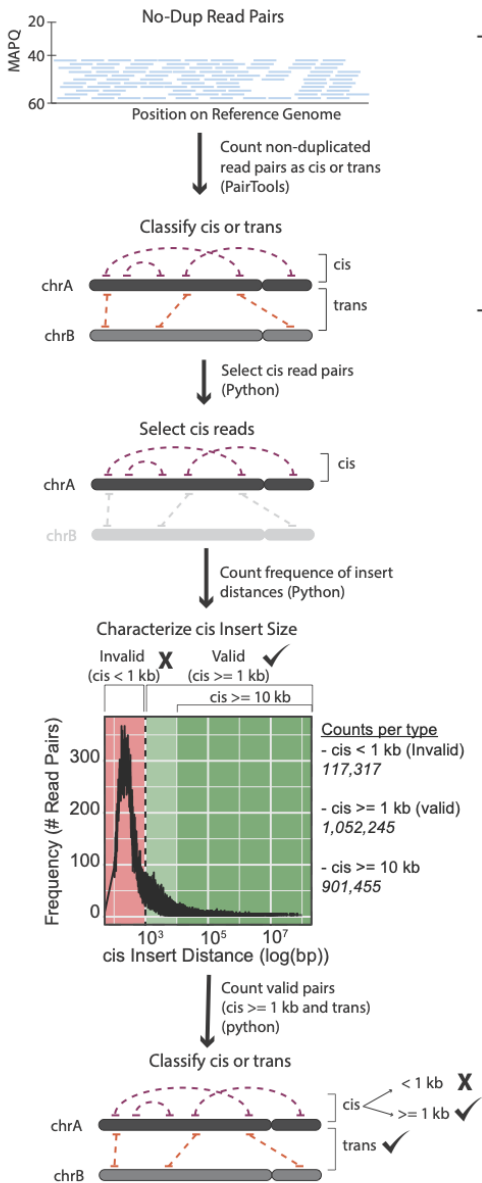
Results

Category	Count	Percent
Total Read Pairs	2,000,000	100.00%
Unmapped Read Pairs	86,000	4.30%
Mapped Read Pairs	1,642,400	82.12%
PCR Dup Read Pairs	2,800	0.14%
No-Dup Read Pairs	1,640,800	81.98%
No-Dup Cis Read Pairs	1,169,563	71.28%
No-Dup Trans Read Pairs	471,317	28.72%
No-Dup Valid Read Pairs (cis >= 1 kb + trans)	1,523,483	92.85%
No-Dup Cis Read Pairs < 1kb	117,317	7.15%
No-Dup Cis Read Pairs >= 1kb	1,052,245	64.13%
No-Dup Cis Read Pairs >=10kb	901,455	54.94%



II. Classifying read pairs (cis or trans), characterizing insert size, and identifying valid pairs

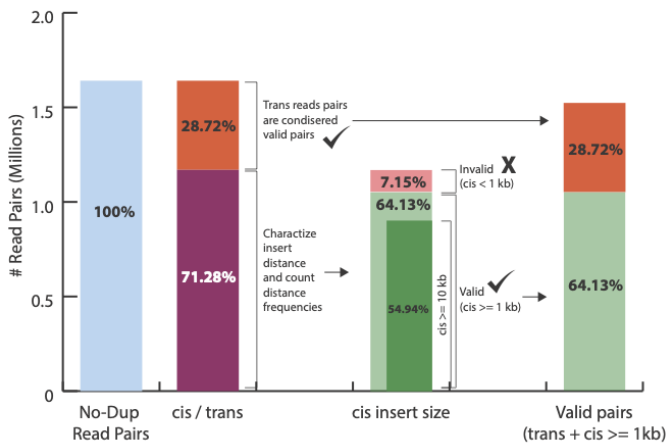
Process



Results

Category	Count	Percent
Total Read Pairs	2,000,000	100.00%
Unmapped Read Pairs	86,000	4.30%
Mapped Read Pairs	1,642,400	81.98%
PCR Dup Read Pairs	2,800	0.14%
No-Dup Read Pairs	1,640,800	82.04%
No-Dup Cis Read Pairs	1,169,563	71.28%
No-Dup Trans Read Pairs	471,317	28.72%
No-Dup Valid Read Pairs (cis >= 1 kb + trans)	1,523,483	92.85%
No-Dup Cis Read Pairs < 1 kb	117,317	7.15%
No-Dup Cis Read Pairs >= 1 kb	1,052,245	64.13%
No-Dup Cis Read Pairs >= 10kb	901,455	54.94%

Proportion of No-Dup Read Pairs



- No-Dup Read Pairs
- No-Dup Cis Read Pairs
- No-Dup Trans Read Pairs
- No-Dup Cis Read Pairs >= 1kb
- No-Dup Cis Read Pairs >= 10kb
- No-Dup Cis Read Pairs < 1kb
- Valid Read Pair
- Invalid Read Pair

Command:

```
python3 ./HiChIP/get_qc.py -p <stats.txt>
```

Example:

```
python3 ./HiChIP/get_qc.py -p stats.txt
```

After the script completes, it will print:

Total Read Pairs	2,000,000	100%
Unmapped Read Pairs	75,832	3.79%
Mapped Read Pairs	1,722,285	86.11%
PCR Dup Read Pairs	4,507	0.23%
No-Dup Read Pairs	1,717,778	85.89%
No-Dup Cis Read Pairs	1,385,238	80.64%
No-Dup Trans Read Pairs	332,540	19.36%
No-Dup Valid Read Pairs (cis >= 1kb + trans)	875,804	50.98%
No-Dup Cis Read Pairs < 1kb	841,974	49.02%
No-Dup Cis Read Pairs >= 1kb	543,264	31.63%
No-Dup Cis Read Pairs >= 10kb	193,061	11.24%

We consider a library prepared from a **mammalian** sample to be acceptable if: - Mapped nondupe pairs cis > 1,000 bp is greater than 20% of the total mapped No-Dup pairs.

1.4.2 ChIP enrichment

Calculating enrichment stats

Another key step in evaluating the quality of the HiChIP library is assessing the enrichment of HiChIP reads at protein binding sites, when protein binding sites correspond to a list of ChIP-Seq peaks.

Our QC pipeline supports as an input both peaks in a simple bed file format (containing three columns: chr, start, end) or [ENCODE narrow peak format](#). For your convenience we include here [links](#) to some key examples of peak files from ENCODE ChIP-Seq experiments. All are of proteins for which Dovetail™ HiChIP MNase Kit has [validated antibodies](#).

You can obtain gold-standards ChIP-Seq peaks from databases, such as ENCODE, or generate your own list of peaks based on ChIP-Seq experiments, e.g. using [MACS2](#).

To calculate stats of reads enrichment around ChIP peaks, we provide the `enrichment_stats.sh` script:

Reminder!

Did you remember to make the `enrichment_stats.sh` script executable?

If not, run the following command:

```
chmod +x ./HiChIP/enrichment_stats.sh
```

If you already ran this command, no need to run it again the execution permission is saved

Parameter	Function
-g	Input <i>genome file</i>
-b	Input <i>final bam file</i>
-p	Input (either in asimple bed format or narrow peak format)
-t	no. of threads
-x	Prefix for output file, enrichment stats will be saved to <prefix>_hichip_qc_metrics.txt

Command:

```
./HiChIP/enrichment_stats.sh -g <ref.genome> -b <mapped.PT.bam> -p <peaks.bed> -t <cores>
↪ -x <prefix>
```

Example:

```
./HiChIP/enrichment_stats.sh -g hg38.genome -b mapped.PT.bam -p ENCFF017XLW.bed -t 16 -x_
↪CTCF
```

Tip!

If your peak file is zipped make sure to unzip it before running the `enrichment_stats.sh` script, e.g.:

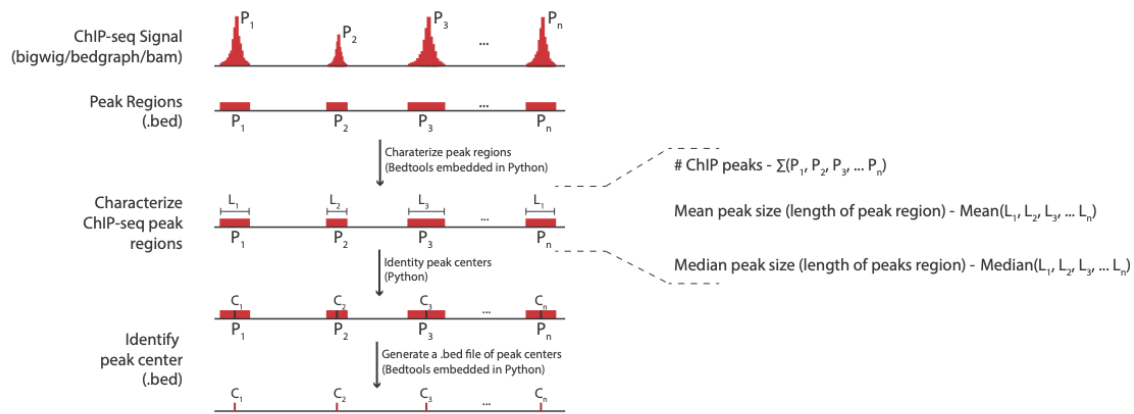
```
gunzip peak.bed.gz
```

In this example an output file `CTCF_hichip_qc_metrics.txt` will be created with the below information:

Total ChIP peaks	41,017	
Mean ChIP peak size	309 bp	
Median ChIP peak size	356 bp	
Total reads in 500 bp around center of peaks	321,368	7.91%
Total reads in 1000 bp around center of peaks	458,843	11.3%
Total reads in 2000 bp around summits	673,628	16.59%
Observed/Expected ratio for reads in 500 bp around center of peaks	11.92	
Observed/Expected ratio for reads in 1000 bp around center of peaks	8.51	
Observed/Expected ratio for reads in 2000 bp around center of peaks	6.25	

The following image illustrates how enrichment around ChIP-Seq peaks is calculated:

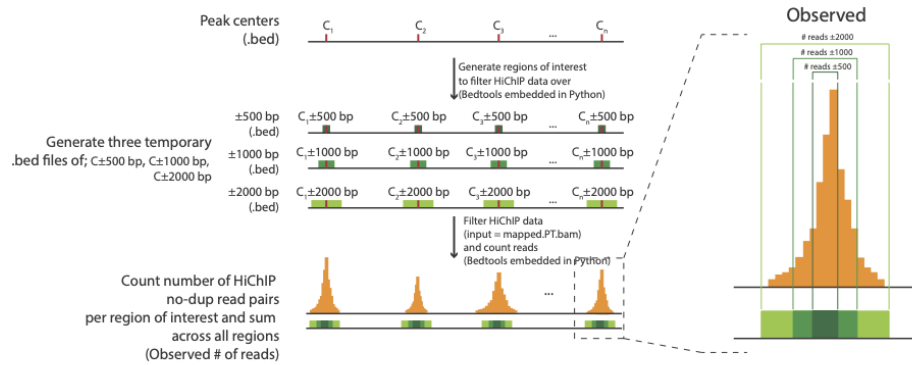
Step 1 - Count and characterize ChIP-seq peak regions (User provided or collected from ENCODE)



Category	Value	Percent
Total ChIP peaks	41,017	NA
Mean ChIP peak size	309 bp	NA
Median ChIP peak size	309 bp	NA
Total reads in 500 bp around center of peaks	393,163	9.46%
Total reads in 1000 bp around center of peaks	519,272	12.49%
Total reads in 2000 bp around center of peaks	692,305	16.66%
Observed/Expected ratio of reads in 500 bp around center of peaks	14.25	NA
Observed/Expected ratio of reads in 1000 bp around center of peaks	9.41	NA
Observed/Expected ratio of reads in 2000 bp around center of peaks	6.27	NA

Step 2 - Assess observed HiChiP coverage enrichment around peak centers and compare to expected value

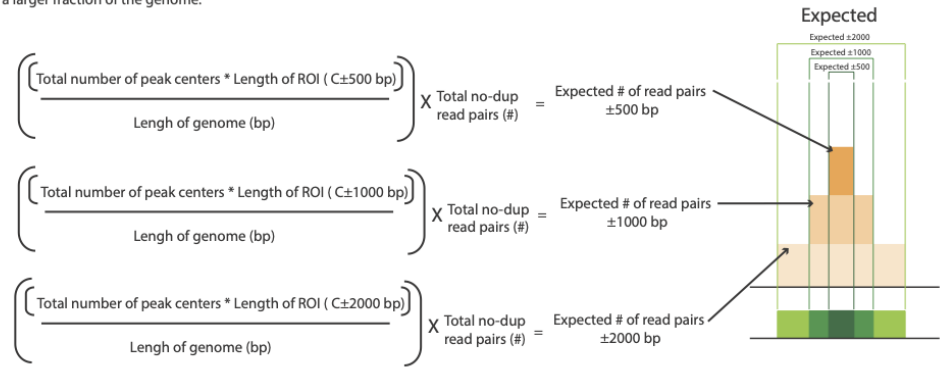
Step 2a - Count number of **observed** HiChiP read pairs in regions surrounding ChIP-seq peak centers



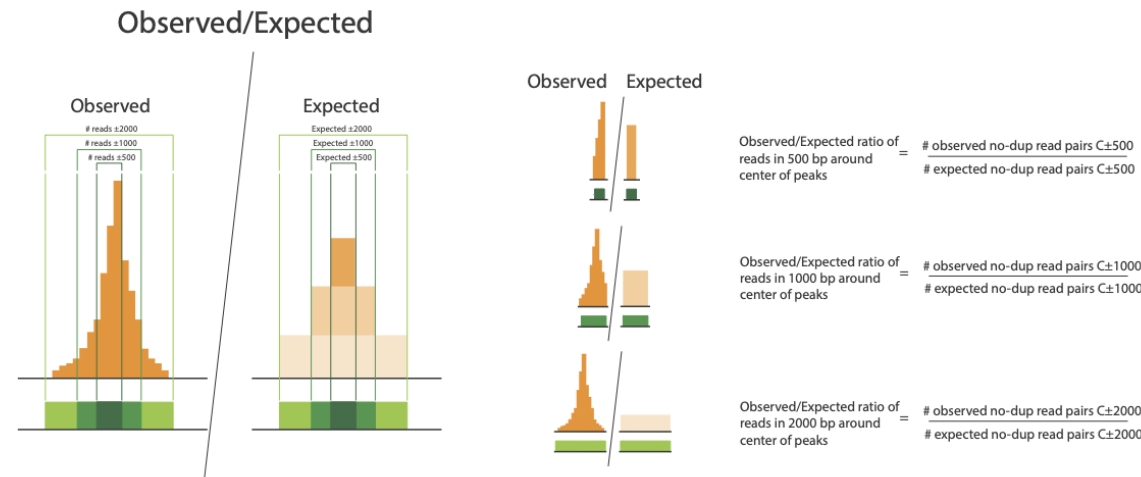
Category	Value	Percent
Total ChIP peaks	41,017	NA
Mean ChIP peak size	309 bp	NA
Median ChIP peak size	309 bp	NA
Total reads in 500 bp around center of peaks	393,163	9.46%
Total reads in 1000 bp around center of peaks	519,272	12.49%
Total reads in 2000 bp around center of peaks	692,305	16.66%
Observed/Expected ratio of reads in 500 bp around center of peaks	14.25	NA
Observed/Expected ratio of reads in 1000 bp around center of peaks	9.41	NA
Observed/Expected ratio of reads in 2000 bp around center of peaks	6.27	NA

Step 2b - Calculate the number of **expected** HiChIP read pairs in regions surrounding ChIP-seq peak centers

Expected value is calculated based on the assumption that the total number of no-dup read pairs is evenly distributed across the ROI proportion of the genome. This is achieved by 1) determining what fraction of the genome length do the ROIs account for, then 2) evenly distributing the total no-dup read pairs across this fraction of the genome. Resulting in a constant coverage value at each bed entry. The expected coverage value decreases with increasing ROI length as the same number of no-dup read pairs are being distributed across a larger fraction of the genome.



Step 2c - Calculate the observed to expected ratio



Category	Value	Percent
Total ChIP peaks	41,017	NA
Mean ChIP peak size	309 bp	NA
Median ChIP peak size	309 bp	NA
Total reads in 500 bp around center of peaks	393,163	9.46%
Total reads in 1000 bp around center of peaks	519,272	12.49%
Total reads in 2000 bp around center of peaks	692,305	16.66%
Observed/Expected ratio of reads in 500 bp around center of peaks	14.25	NA
Observed/Expected ratio of reads in 1000 bp around center of peaks	9.41	NA
Observed/Expected ratio of reads in 2000 bp around center of peaks	6.27	NA

Plotting global enrichment around ChiP peaks

The `plot_chip_enrichment.py` and `plot_chip_enrichment_bed.py` scripts provide global evaluation of enrichment around known ChiP peaks. The script identifies the regions of ChiP peaks, sets a window of 1kb upstream and downstream of the peak's center, and based on the *bam file* of the valid pairs, calculates the aggregated read coverage within this window and plots the global fold coverage change based on the observed coverage divided by the mean coverage, as *illustrated*.

`plot_chip_enrichment.py` is intended to be used when a `narrowPeak` file is available and `plot_chip_enrichment_bed.py` accept a simple bed file with peaks intervals as an input. Other than that, the two scripts accept the same parameters:

Parameter	Function
-bam	Input <i>final bam file</i>
-peaks	Input peaks in <code>narrowPeak</code> format (<code>plot_chip_enrichment.py</code>) or in simple chr,start,end bed format (<code>plot_chip_enrichment_bed.py</code>)
-output	output file name to save the enrichment plot .png image

Command:

```
python3 plot_chip_enrichment.py -bam <mapped.PT.bam> -peaks <peaks.bed> -output
↳<enrichment.png>
```

or

```
python3 plot_chip_enrichment_bed.py -bam <mapped.PT.bam> -peaks <peaks.bed> -output
↳<enrichment.png>
```

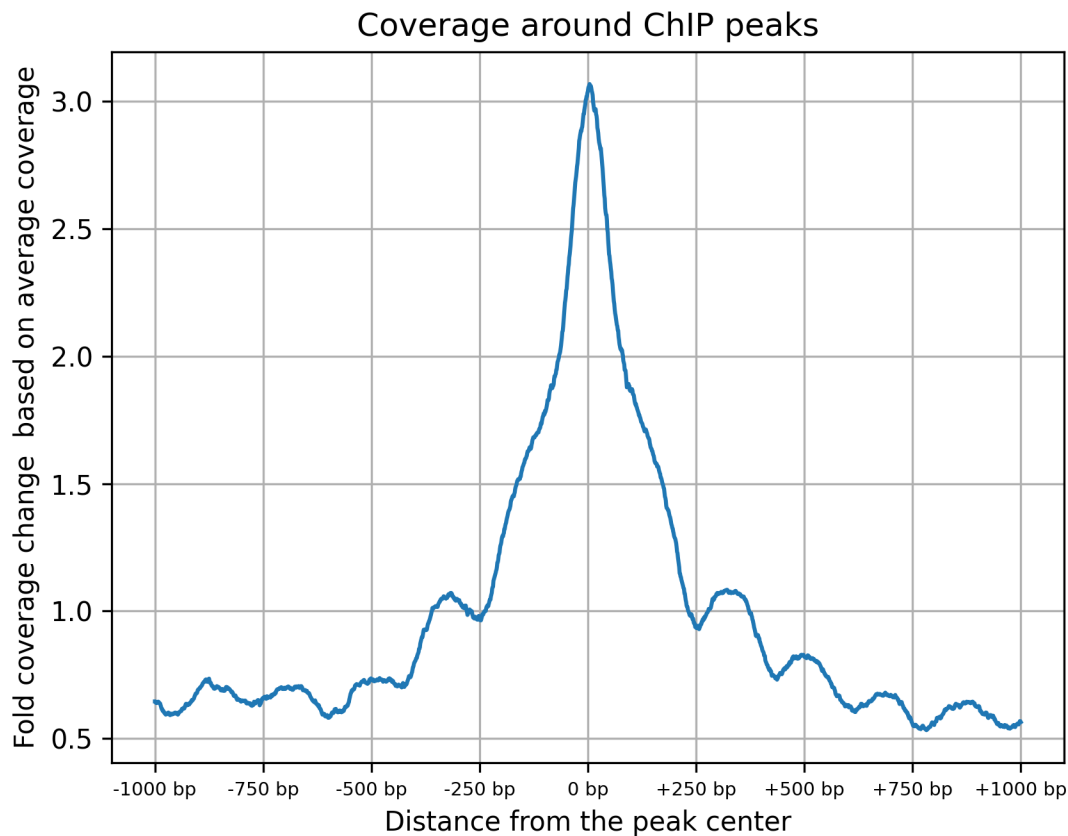
Example:

```
python3 ./HiChiP/plot_chip_enrichment.py -bam mapped.PT.bam -peaks ENCFF017XLW.bed -
↳output enrichment.png
```

or

```
python3 ./HiChiP/plot_chip_enrichment_bed.py -bam mapped.PT.bam -peaks peaks.bed -output
↳enrichment.png
```

Output plot:



Important!

- `plot_chip_enrichment.py` will accept only `narrowPeak` format which has to include 10 columns, with the following specifications: - chromosome, start, end, in the three first columns - Peak Signal value at column #7 - Peak offset value at column #10 (when offset is the distance between the start position and the center of the peaks)
- If your peak file does not follow the above structure you can modify it into a simple bed file by extracting only the three first columns into a new file that can be used with the `plot_chip_enrichment_bed.py` script.
- `plot_chip_enrichment_bed.py` will accept only bed files with 3 columns. If your bed file includes more than three columns, extract the three first columns into a new file
- Example, how to extract only the first three columns:

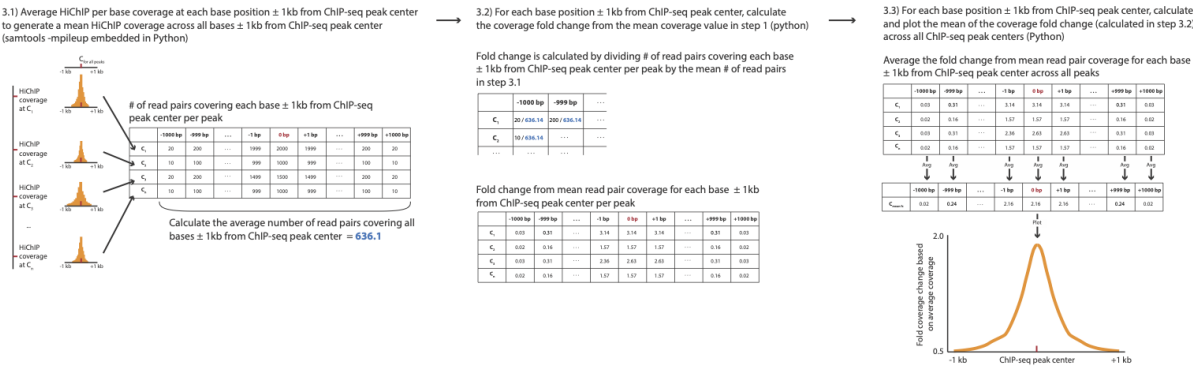
```
cut -f1,2,3 input.bed > output.bed
```

There are two minor differences between the two scripts:

- `plot_chip_enrichment.py` calculates the center of the peak according to `start + offset`
`plot_chip_enrichment_bed.py` chooses the center of the peak as the middle point between `start` and `end`. Both will calculate the aggregated enrichment -1kb and +1kb of the center of the peak (no matter the length of the peak)
- All intervals in the bed files are used for the meta-analysis when `plot_chip_enrichment_bed.py` is used
`narrowPeak` format includes information on peak signal, this information is used to filter out peaks with extreme

values (either very low or very high signals) prior to meta-analysis

Step 3 - Read density enrichment of HiChIP data over ChIP-seq peaks



1.4.3 QC Assessment

Pass/No Pass Metrics

Now that you have successfully completed the QC scripts, it is time to determine if the HiChIP library is of high quality. The QC metrics calculated above can be distilled down to three key quantitative metrics and one qualitative step to help you assess the quality of the library before proceeding into deep sequencing.

Output from get_qc.py

	Total Read Pairs	2,000,000	100%
	Unmapped Read Pairs	75,832	3.79%
	Mapped Read Pairs	1,722,285	86.11%
	PCR Dup Read Pairs	4,507	0.23%
1.	No-Dup Read Pairs	1,717,778	85.89%
	No-Dup Cis Read Pairs	1,385,238	80.64%
	No-Dup Trans Read Pairs	332,540	19.36%
	No-Dup Valid Read Pairs (cis >= 1kb + trans)	875,804	50.98%
	No-Dup Cis Read Pairs < 1kb	841,974	49.02%
2.	No-Dup Cis Read Pairs >= 1kb	543,264	31.63%
	No-Dup Cis Read Pairs >= 10kb	193,061	11.24%

Output from enrichment_stats.sh

	Total ChIP peaks	41,017	
	Mean ChIP peak size	309 bp	
	Median ChIP peak size	356 bp	
	Total reads in 500 bp around center of peaks	321,368	7.91%
3.	Total reads in 1000 bp around center of peaks	458,843	11.3%
	Total reads in 2000 bp around summits	673,628	16.59%
	Observed/Expected ratio for reads in 500 bp around center of peaks	11.92	
	Observed/Expected ratio for reads in 1000 bp around center of peaks	8.51	
	Observed/Expected ratio for reads in 2000 bp around center of peaks	6.25	

1. No-Dup Read Pairs – This value is reflective of the alignment rate and PCR duplication rate. It should be noted that this value scales inversely with sequencing depth.
2. No-dup cis read pairs 1kb – This value demonstrates that the proximity-ligation step was successful, and the majority of the data are useful in downstream analyses (e.g. loop calling). This value can be dependent on the protein of interest, for example CTCF has a very long-range contact profile while POL2A has a much more localized contact domain. The cut-off used here is applicable across different protein targets.
3. Total reads in 1000 bp around center of peaks – This value demonstrates that chromatin enrichment was successful. This metric is very similar to Fraction of Reads in Peaks (FRiP) score that is used to assess ChIP-seq data as defined by ENCODE. Our defined cut-off value is slightly more stringent than the ENCODE standard of 1%. It should be noted that this value is dependent on the peak set used, the value can be underestimated if you used a peak set that is not reflective of your experiment (e.g publicly available).
4. Visual assessment of HiChIP coverage in IGV – This step provides you a visual peace of mind that your IP-enrichment was successful. Alignments (.bam) should be converted into bigwig format with deepTools bamCoverage <https://deeptools.readthedocs.io/en/develop/content/tools/bamCoverage.html>).

While the QC process can be boiled down to these key values, the remaining values of the QC process are used to diagnose and troubleshoot a library that falls into the “No Pass” category. Therefore, it is important to generate all the values in the QC process in case there is a need for troubleshooting.

Pass/No Pass Values

The table below summarizes the minimum passing values for the metrics defined above. The cut-off values were determined for both shallow sequenced (20 million read pairs 2 x 150 bp) and deep sequenced data (100-200 Million read pairs 2 x 150 bp), as the percentage of mapped no-dup pairs changes with the sequencing depth.

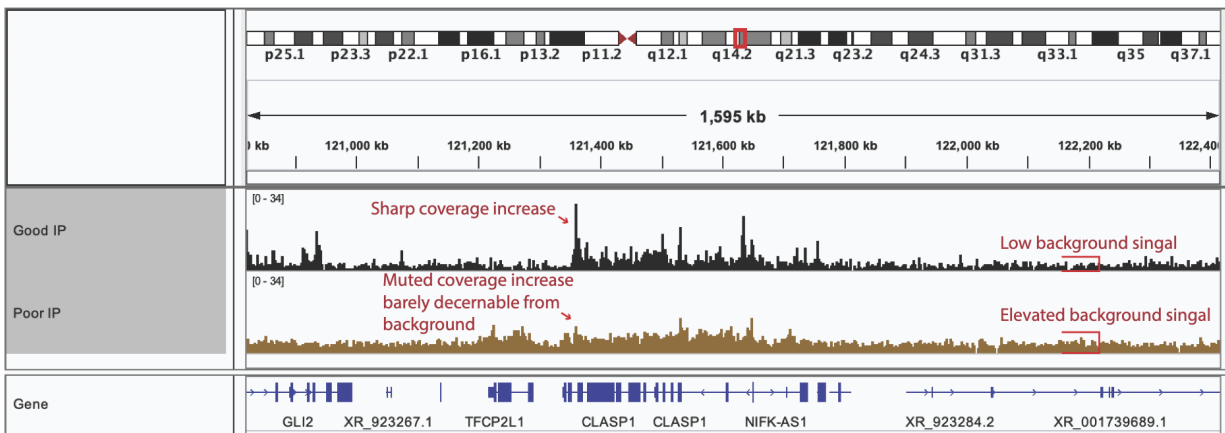
Metric	Shallow Seq (20M)	Deep Seq (100-200M)
No-Dup Read Pairs	>75%	>50%
No-dup cis read pairs 1kb	>20%	>20%
Total reads in 1000 bp around center of peaks	>2%	>2%

Visual Inspection Of The Alignments

Once you have compared your library QC values to the minimal quantitative requirements for a library to pass QC, you can move on visual assessment in IGV. Here we used the Integrated Genome Viewer (which can be downloaded an installed [here](#)). IGV is standard genome browser for visualizing NGS data in track format. Simply load your bigwigs into IGV then zoom in to a 1-2 Mbp window. In this step, we are looking to see if the data suggest that there has been enrichment.

- Good IP – exhibit distinct signals of sharply increased coverage from a low background indicating the location of the protein-DNA binding Site.
- Poor IP – exhibit no or weak coverage increases and are often accompanied by an elevated background signal.

Below is an annotated screenshot from IGV showing examples of both good and bad IP of shallow sequenced (20 M read pairs) libraries. The library exhibiting good IP characteristics (top track in black) shows clear, sharp coverage enrichment, and low background signal, where the library with poor IP, (bottom track in brown) has a high background signal and muted coverage enrichment.



Final Determination

If your libraries pass the minimum threshold for each of the 3 quantitative metrics, and the visual inspection:

- For shallow sequenced libraries - proceed to deep sequencing (~150 M read pairs per library)
- For deep sequencing – proceed with downstream analyses

If the libraries fail one or more of the 3 quantitative metrics or the visual inspection - please reach out to our support team at: support@dovetail-genomics.com

1.5 Generating Contact Matrix

There are two common formats for contact maps, the [Cooler format](#) and [Hic](#) format. Both are compressed and sparsed formats to avoid large storage volumes; For a given n number of bins in the genome, the size of the matrix would be n^2 , in addition, typically more than one resolution (bin size) is being used.

In this section we will guide you on how to generate both matrices types, [HiC](#) and [cool](#) based on the [.pairs file](#) that you generated in the [previous section](#) and how to visualize them.

1.5.1 Generating HiC contact maps using Juicer tools

Additional Dependencies

- [Juicer Tools](#) - Download the JAR file for juicertools and place it in the same directory as this repository and name it as `juicertools.jar`. You can find the link to the most recent version of Juicer tools [here](#) e.g.:

```
wget https://s3.amazonaws.com/hicfiles.tc4ga.com/public/juicer/juicer_tools_1.22.01.jar
mv juicer_tools_1.22.01.jar ./HiChiP/juicertools.jar
```

- Java - If not already installed, you can install Java as follows:

```
sudo apt install default-jre
```

From .pairs to .hic contact matrix

- [Juicer Tools](#) is used to convert [.pairs](#) file into a [HiC](#) contact matrix.
- HiC is highly compressed binary representation of the contact matrix
- Provides rapid random access to any genomic region matrix
- Stores contact matrix at 9 different resolutions (2.5M, 1M, 500K, 250K, 100K, 50K, 25K, 10K, and 5K)
- Can be programmatically manipulated using straw python API

The [.pairs](#) file that you generated in the [From fastq to final valid pairs bam file](#) section can be used directly with Juicer tools to generate the [HiC](#) contact matrix:

Parameter	Function
-Xmx	The flag Xmx specifies the maximum memory allocation pool for a Java virtual machine, from our experience 48000m works well when processing human data sets. If you are not sure how much memory your system has, run the command <code>free -h</code> and check the value under <i>total</i>
Djava.awt.headless=true	Java is ran in a headless mode when the application does not interact with a user (if not specified, the default is Djava.awt.headless=false)
pre	The pre command allows users to create .hic files from their own data
--threads	Specifies the numbers of threads to be used (integer number)
*.pairs or *.pairs.gz	input file for generating the contact matrix
*.genome	genome file, listing the chromosomes and their sizes
*.hic	hic output file, containing the contact matrix

Tip no.1

Please note that if you have an older version of `Juicer tools`, generating contact map directly from `.pairs` file may not be supported. We recommend updating to a newer version. As we tested, the `pre` utility of the version 1.22.01 support the `.pairs` to HiC function.

Command:

```
java -Xmx<memory> -Djava.awt.headless=true -jar <path_to_juicer_tools.jar> pre --
↳ threads <no_of_threads> <mapped.pairs> <contact-map.hic> <ref.genome>
```

Example:

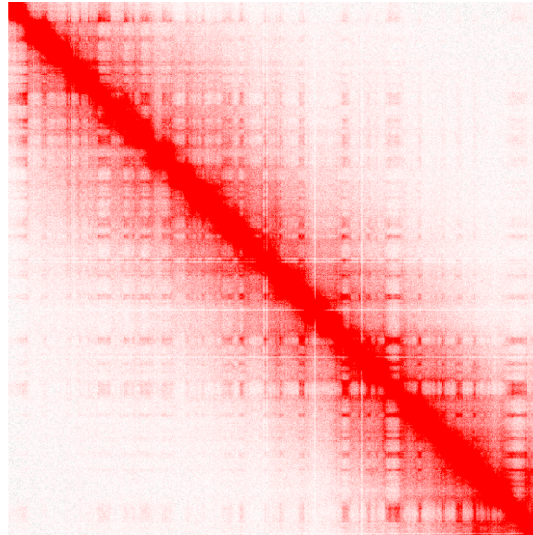
```
java -Xmx48000m -Djava.awt.headless=true -jar ./HiChiP/juicertools.jar pre --threads 16
↳ mapped.pairs contact_map.hic hg38.genome
```

Tip no.2

`Juicer tools` offers additional functions that were not discussed here, including matrix normalization and generating matrix for only specified regions in the genome. To learn more about advanced options, please refer to the [Juicer Tools documentation](#).

Visualizing .hic contact matrix

The visualization tool Juicebox can be used to visualize the contact matrix. You can either [download](#) a local version of the tool to your computer as a Java application or use a [web](#) version of Juicebox. Load your .hic file to visualize the contact map and zoom in to areas of interest.



1.5.2 Generating cooler contact maps

Additional Dependencies

Installing Cooler and its dependencies

- `libhdf5` - `sudo apt-get install libhdf5-dev`
- `h5py` - `pip3 install h5py`
- `cooler` - `pip3 install cooler`

For any issues with cooler installation or its dependencies, please refer to the [cooler installation documentation](#)

Installing Pairix

`Pairix` is a tool for indexing and querying on a block-compressed text file containing pairs of genomic coordinates. You can install it directly from its github repository as follows:

```
git clone https://github.com/4dn-dcic/pairix
cd pairix
make
```

Add the bin path, and utils path to PATH and exit the folder:

```
PATH=~/pairix/bin/:~/pairix/util:~/pairix/bin/pairix:$PATH
cd ..
```

Important!

make sure to modify the following example with the path to your *pairix* installation folder. If you are not sure what is the path you can check it with the command *pwd* when located in the *pairix* folder.

For any issues with *pairix*, please refer to the [pairix documentation](#)

From .pairs to cooler contact matrix

- *Cooler tools* is used to convert **indexed** .pairs file into *cool* and *mcool* contact matrices
- *Cooler* generates a sparse, compressed, and binary persistent representation of proximity ligation contact matrix
- Store matrix as *HDF5* file object
- Provides python API to manipulate contact matrix
- Each cooler matrix is computed at a specific resolution
- Multi-cool (mcool) files store a set of cooler files into a single HDF5 file object
- Multi-cool files are helpful for visualization

Indexing the .pairs file

We will use the *cloud pairix* utility of *Cooler* to generate contact maps. This utility requires the .pairs file to be indexed. *Pairix* is used for indexing compressed .pairs files. The files should be compressed with *bgzip* (which should already be installed on your machine). If your .pairs file is not yet *bgzip* compressed, first compress it as follows:

Command:

```
bgzip <mapped.pairs>
```

Example:

```
bgzip mapped.pairs
```

Following this command *mapped.pairs* will be replaced with its compressed form *mapped.pairs.gz*

Note!

Compressing the .pairs file with *gzip* instead of *bgzip* will also result in a compressed file with the .gz suffix, but due to format differences it will not be accepted as an input for *pairix*.

Next, index the file .pairs.gz file:

Command:

```
pairix <mapped.pairs.gz>
```

Example:

```
pairix mapped.pairs.gz
```

Generating single resolution contact map files

As mentioned above, we will use the `cooler cload pairix` utility of Cooler to generate contact maps:

`cooler cload pairix` usage:

Parameter	Function
<genome_files>:<bin size>	Specifies the reference <i>.genome file</i> , followed with ``:`` and the desired bin size in bp
-p	Number of processes to split the work between (integer), default: 8
*.pairs.gz	Path to bgzip compressed and indexed <i>.pairs</i> file
*.cool	Name of output file

Command:

```
cooler cload pairix -p <cores> <ref.genome>:<bin_size_in_bp> <mapped.pairs.gz> <matrix.  
↪cool>
```

Example:

```
cooler cload pairix -p 16 hg38.genome:1000 mapped.pairs.gz matrix_1kb.cool
```

Generating multi-resolutions files and visualizing the contact matrix

When you wish to visualize the contact matrix, it is highly recommended to generate a multi-resolution *.mcool* file to allow zooming in and out to inspect regions of interest. The `cooler zoomify` utility allows you to generate a multi-resolution cooler file by coarsening. The input to `cooler zoomify` is a single resolution *.cool* file, to allow zooming in into regions of interest we suggest to generate a *.cool* file with a small bin size, e.g. 1kb. Multi-resolution files uses the suffix *.mcool*.

`cooler zoomify` usage:

Parameter	Function
--balance	Apply balancing to each zoom level. Off by default
-p	Number of processes to use for batch processing chunks of pixels, default: 1
*.cool	Name of contact matrix input file

Command:*

```
cooler zoomify --balance -p <cores> <matrix.cool>
```

Example:

```
cooler zoomify --balance -p 16 matrix_1kb.cool
```

The example above will result in a new file named *matrix_1kb.mcool* (no need to specify output name)

Tip

Cooler offers additional functions that were not discussed here, including generating a cooler from a pre-binned matrix, matrix normalization and more. To learn more about advanced options, please refer to the cooler [documentation](#)

[HiGlass](#) is an interactive tool for visualizing *.mcool* files. To learn more about how to set up and use HiGlass follow the HiGlass [tutorial](#)

1.6 HiChIP Loop Calling

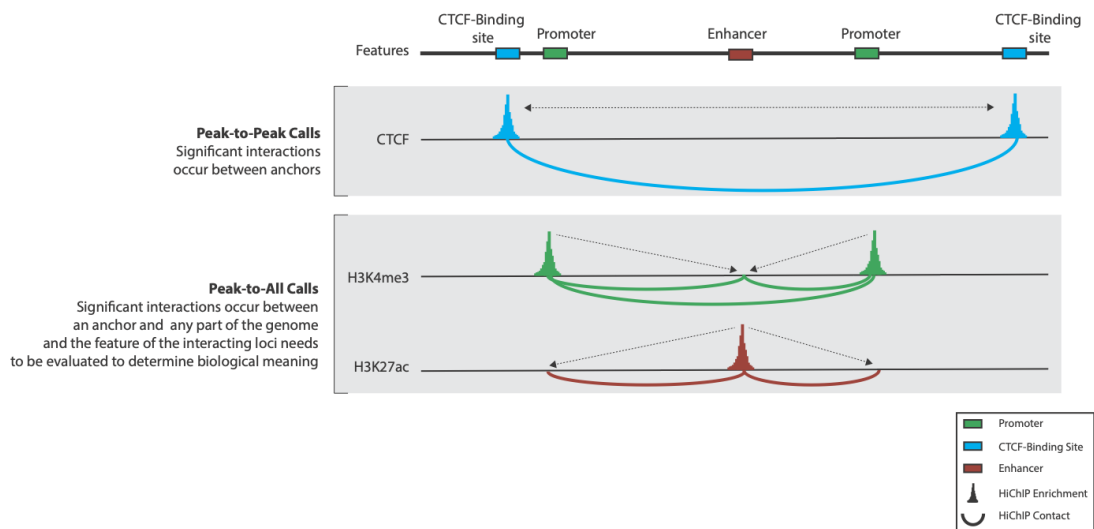
1.6.1 Introduction

This workflow is a simple guide to identify loops in HiChIP data. Before you get started please read this short introduction which will help you better understand what loops are in the context of HiChIP assays and why we're going to focus on [FitHiChIP tool](#) as the tool to use. I would like preface this work by saying there is no "one correct way" to analyze HiChIP data. This is just an example workflow that will enable you identify significant interactions in HiChIP data. The biological implications of those interactions should be interpreted through the lenses of the protein target or biological question you're asking!

What are chromatin Loops in the context of HiChIP?

- HiChIP loops are - significant interactions between a protein-anchor and the surrounding genome.
- The biological interpretations of the interactions is based on the protein of interest for example:
 - CTCF – Identifying actual chromatin loops. Meaning, CTCF-cohesin mediated looping.
 - H3K27ac or H3K4me3 – means identifying regions that interact with the enhancer or active promotor marker respectively. These interactions do not necessarily reflect the canonical loop formation but could reflect short range folding.
- As these interactions can reflect more than just canonical "loops", they will simply be referred to as significant interactions for the rest of the documentation.
- Types of significant HiChIP interactions
 - Peak-to-Peak - ChIP-seq anchors will only interact with other anchors like CTCF chromatin loops
 - Peak-to-All - ChIP-seq anchors will interact with any of the surrounding genome whether or those regions correspond with an anchor site, like H3K27ac, H3K4me3, or PolII
 - All-to-All - This is for a non-targeted Hi-C data contact matrix and will not be used here.

HiChIP interactions type:



1.6.2 Resolution

Resolution can play an odd roll in HiChIP significant interaction detection. Typically, only 5-10 reads-pairs per interaction are required to statistically identify an interaction with HiChIP as opposed to the 100-1000 read-pairs required for non-targeted Hi-C assays. However, many loop callers work on contact matrices which require binning at a specified resolution(s) to build. Typically, 1 kb, 5 kb, or 10 kb for HiChIP data. As such, the fewer reads you have, the larger the bin size must be to have enough read support to run statistics on.

Additionally, the biological nature of the protein target can impact the resolution of that you are interested in looking at. Proteins with a larger footprint (PolII) might require larger bin size, where smaller footprints (CTCF) might need a smaller bin. What becomes problematic, is when the bin size is so large that many anchors are captured in a single bin. This is important to keep in mind.

With all that being said, most HiChIP/ChIA-PET analyses are conducted between 2.5 - 5 kb. Our best recommendation is to try calling significant interactions at different resolutions. Generate lists of significant interactions at multiple resolutions and filter to keep only unique entries.

1.6.3 Tool landscape

There are many tools available to identify significant interactions. Below is a table that outlines just a subset of tools, where to get them, requirement to specify a resolution, ability to select the type of interaction, and input file structure of the HiChIP data. They all have their own pro's and con's, but there is no clearly established way to analyze HiChIP data, and it largely depends on your biological questions. So always keep that in mind and make sure the tool you're using makes sense with the biological question you are asking!

Tool	Repo	Input format	Resolution Option	Considers type of interaction
FitHiChIP	https://ay-lab.github.io/FitHiChIP/	HiC-Pro Valid-Pairs	Yes	Yes
Cloops	https://github.com/YaqiangCao/cLoops	Bedpe	No	No
HiChIP-PER	https://github.com/aryeelab/hichipper	HiC-Pro Valid-Pairs	Yes	No
HiC-CUPS	https://github.com/aidenlab/juicer/wiki/HiCCUPS	.hic	Yes	No

1.6.4 Why FitHiChIP?

We have chosen FitHiChIP for this workflow for a few reasons:

1. The install is very easy, and it can manage all the dependencies through Docker or Singularity (if you don't have sudo privileges)
2. It is very flexible in term of input, .pairs, or interaction tabel in .bedpe format.
3. Has the ability to select bias type
4. Can specify the type of interaction to assess
5. Output is easily integrate-able to other workflows

Below is an annotated configuration file with some of the key parameters to consider

```

1  #####
2  # Sample configuration file for running FitHiChIP
3  #####
4
5  #####
6  # important parameters
7  #####
8
9  # File containing the valid pairs from HiCPro pipeline
10 # Can be either a text file, or a gzipped text file
11 ValidPairs=../TestData/Sample_ValidPairs.txt.gz
12
13 # File containing the bin intervals (according to a specified bin size)
14 # which is an output of HiC-pro pipeline
15 # If not provided, this is computed from the parameter 1
16 Interval=
17
18 # File storing the contact matrix (output of HiC-pro pipeline)
19 # should be accompanied with the parameter 2
20 # if not specified, computed from the parameter 1
21 Matrix=
22
23 # Pre-computed locus pair file
24 # of the format:
25 # chr1 start1 end1 chr2 start2 end2 contactcounts
26 Bed=
27
28 #####
29 # File containing reference ChIP-seq / HiChIP peaks (in .bed format)
30 # mandatory parameter
31 PeakFile=../TestData/Sample.Peaks.gz
32
33 #####
34 # Output base directory under which all results will be stored
35 OutDir=../TestData/results/
36
37 #####
38 # Interaction type - 1: peak to peak 2: peak to non peak 3: peak to all (default) 4: all to all 5: everything from 1 to 4.
39 IntType=3
40
41 #####
42 # Size of the bins [default = 5000], in bases, for detecting the interactions.
43 BINSIZE=5000
44
45 #####
46 # Lower distance threshold of interaction between two segments
47 # (default = 20000 or 20 Kb)
48 LowDistThr=20000
49
50 #####
51 # Upper distance threshold of interaction between two segments
52 # (default = 2000000 or 2 Mb)
53 UpDistThr=2000000
54
55 #####
56 # Applicable only for peak to all output interactions - values: 0 / 1
57 # if 1, uses only peak to peak loops for background modeling - corresponds to FitHiChIP(S)
58 # if 0, uses both peak to peak and peak to nonpeak loops for background modeling - corresponds to FitHiChIP(L)
59 UseP2PBackgrnd=1
60
61 #####
62 # parameter signifying the type of bias vector - values: 1 / 2
63 # 1: coverage bias regression 2: ICE bias regression
64 BiasType=1
65
66 #####
67 # following parameter, if 1, means that merge filtering (corresponding to either FitHiChIP(L+M) or FitHiChIP(S+M))
68 # depending on the background model, would be employed. Otherwise (if 0), no merge filtering is employed. Default: 1
69 MergeInt=1
70
71 #####
72 # FDR (q-value) threshold for loop significance
73 QVALUE=0.01
74
75 #####
76 # File containing chromosome size values corresponding to the reference genome.
77 ChrSizeFile=../TestData/chrom_hg19.sizes
78
79 #####
80 # prefix string of all the output files (Default = 'FitHiChIP')
81 PREFIX=FitHiChIP
82
83 #####
84 # Binary variable 1/0: if 1, overwrites any existing output file. otherwise (0), does not overwrite any output file.
85 OverWrite=1
86

```

full path to HiC-Pro format valid pairs file

full path to ChIP-seq peaks

full path to output directory

Resolution:
2.5 - 5 Kb is a good start

Lower distance:
This depends on the anchor in question -
CTCF 20 kb is a good start, H3K27ac or PolII a lowering 10 kb is good idea

Interaction type:
1 - peak to peak (e.g. CTCF)
2 - peak to non-peak (e.g. PolII)
3 - peak to all (H3K4me3, H3K27ac)
4 - All to all (Hi-C data)
5 - run everything

Stringency:
1 - FitHiChIP(S) - More stringent (peak to peak)
2 - FitHiChIP(L) - Less stringent (peak to all, peak to non-peak)

Bias factor
1 - Coverage (reccomended)
2 - ICE

full path to .genome file

prefix file name for results files

1.6.5 Input files

This workflow assumes you have completed the *Step-by-step guide to Process HiChIP data*. The two key files required are:

- *Filtered Pairs file* - output from *From fastq to final valid pairs* workflow.
- Bed file of ChIP-seq anchors for your protein of interest, e.g. as you used in the *QC step*. We included in the *datasets section* links to some useful ChIP-seq bed files from the Encode project.

Testing!

If you are looking for a dataset to practice this walkthrough, I recommend the GM12878 CTCF (deep sequencing) from our publicly available *datasets*

1.6.6 Tools

- **FitHiChIP**
 - Full documentation is [here](#)
 - I recommend you read it! It is very well documented!


1.6.7 Workflow Overview

- *Convert filtered pairs file to Hi-C Pro valid pairs format.*
- *Modify FitHiChIP configuration file*
- *Run FitHiChIP through docker* - FitHiChIP is a single executable that:
 - Builds a table of interactions (bedpe-like version of a contact matrix)
 - Corrects for biases (coverage or ICE)
 - Filters data for the type of interactions (Peak-to-Peak, Peak-to-All, or All-to-All)
 - Builds a contact frequency to insert distance model from the filtered interactions.
 - Assigns P-values and Q-values (false discovery rate - FDR) to interactions.
 - Will merge near-by interaction that pass a Q-value threshold.
 - Report a bedpe-like file of total and merged interactions filtered by a Q value.
- *Inspect the output report*

1.6.8 Workflow

- Convert filtered pairs file to Hi-C Pro valid pairs format

Command:

```
grep -v '#' <*.pairs> | awk -F"\t" '{print $1"\t"$2"\t"$3"\t"$6"\t"$4"\t"$5"\t"$7}' |  gzip -c > <output.pairs.gz>
```

Example:

```
grep -v '#' mapped.pairs| awk -F"\t" '{print $1"\t"$2"\t"$3"\t"$6"\t"$4"\t"$5"\t"$7}' |
↪gzip -c > hicpro_mapped.pairs.gz
```

- Modify the configuration file to desired specifications:
 - We'll be using coverage bias because these data are MNase based, not RE-based
 - If using CTCT use Peak-to-Peak as outlined earlier, CTCF data is a peak to peak interaction, other protein like H3K27ac and H3K4me3 you're going to want to use Peak-to-All.

Adjusting the configuration file . Entries that need to be adjusted are highlighted:

```
#=====
# Sample configuration file for running FitHiChIP
#=====
#*****
# important parameters
#*****
# File containing the valid pairs from HiCPro pipeline
# Can be either a text file, or a gzipped text file
ValidPairs=/path_to_hicpro_pairs/prefix.hicpro.valid.pairs.gz
# File containing the bin intervals (according to a specified bin size)
# which is an output of HiC-pro pipeline
# If not provided, this is computed from the parameter 1
Interval=
# File storing the contact matrix (output of HiC-pro pipeline)
# should be accompanied with the parameter 2
# if not specified, computed from the parameter 1
Matrix=
# Pre-computed locus pair file
# of the format:
# chr1 start1 end1 chr2 start2 end2 contactcounts
Bed=
# File containing reference ChIP-seq / HiChIP peaks (in .bed format)
# mandatory parameter
PeakFile=/path_to_ChIP_peaks/peaks.bed
# Output base directory under which all results will be stored
OutDir=/path_to_output/fithip_test_1kb
#Interaction type - 1: peak to peak 2: peak to non peak 3: peak to all (default) 4: all
↪to all 5: everything from 1 to 4.
IntType=1
# Size of the bins [default = 5000], in bases, for detecting the interactions.
BINSIZE=2500
# Lower distance threshold of interaction between two segments
# (default = 20000 or 20 Kb)
LowDistThr=20000
# Upper distance threshold of interaction between two segments
# (default = 2000000 or 2 Mb)
UppDistThr=2000000
# Applicable only for peak to all output interactions - values: 0 / 1
# if 1, uses only peak to peak loops for background modeling - corresponds to
↪FitHiChIP(S)
# if 0, uses both peak to peak and peak to nonpeak loops for background modeling -
↪corresponds to FitHiChIP(L)
```

(continues on next page)

(continued from previous page)

```

UseP2PBackgrnd=1
# parameter signifying the type of bias vector - values: 1 / 2
# 1: coverage bias regression    2: ICE bias regression
BiasType=1
# following parameter, if 1, means that merge filtering (corresponding to either
↳ FitHiChIP(L+M) or FitHiChIP(S+M))
# depending on the background model, would be employed. Otherwise (if 0), no merge
↳ filtering is employed. Default: 1
MergeInt=1
# FDR (q-value) threshold for loop significance
QVALUE=0.01
# File containing chromosome size values corresponding to the reference genome.
ChrSizeFile=/path_to_genome_file/hg38.genome
# prefix string of all the output files (Default = 'FitHiChIP').
PREFIX=prefix.2.5kb
# Binary variable 1/0: if 1, overwrites any existing output file. otherwise (0), does
↳ not overwrite any output file.
OverWrite=1

```

- Run FitHiChIP through docker

Command:

```
FitHiChIP_Docker.sh -C config.txt
```

- Inspect the report

FitHiChIP summary Report

Parameter list for the current execution of FitHiChIP: /home/ubuntu/fithichip_CTCF_2.5kb/Parameters.txt

List of CIS interactions within the specified distance thresholds (lower threshold: 20000 upper threshold: 2000000):
/home/ubuntu/fithichip_CTCF_2.5kb/HiChIP_Pro_Matrix_BinSize25001_20000_U2000000/CTCF_DS.2.5kb.cis.interactions.DistThr.bed

Number of interacting (non zero contacts) bin pairs (all to all): 20992259 **← Total number of bins in the genome**

Bin specific HiChIP coverage, coverage bias, and peak / non-peak boolean values: /home/ubuntu/fithichip_CTCF_2.5kb/NormFeatures/Coverage_Bias/CTCFDS.2.5kb.coverage_Bias.bed

Statistics for peak to peak loops

Statistics for Coverage bias regression

All interactions with their FitHiChIP significance values:

/home/ubuntu/fithichip_CTCF_2.5kb/FitHiChIP_Peak2Peak_b2500_L20000_U2000000/Coverage_Bias/FitHiChIP_BiasCorr/CTCF_DS.2.5kb.interactions_FitHiChIP.bed

Number of interactions considered: 249490 **← Total number of bins in the that occur overlap a ChIP-seq peak**

Significant interactions: /home/ubuntu/fithichip_CTCF_2.5kb/FitHiChIP_Peak2Peak_b2500_L20000_U2000000/Coverage_Bias/FitHiChIP_BiasCorr/CTCF_DS.2.5kb.interactions_FitHiChIP_Q0.01.bed

Total number of significant interactions (FDR 0.01): 21791 **← Total number significant interactions between bins that overlap a ChIP-seq peak**

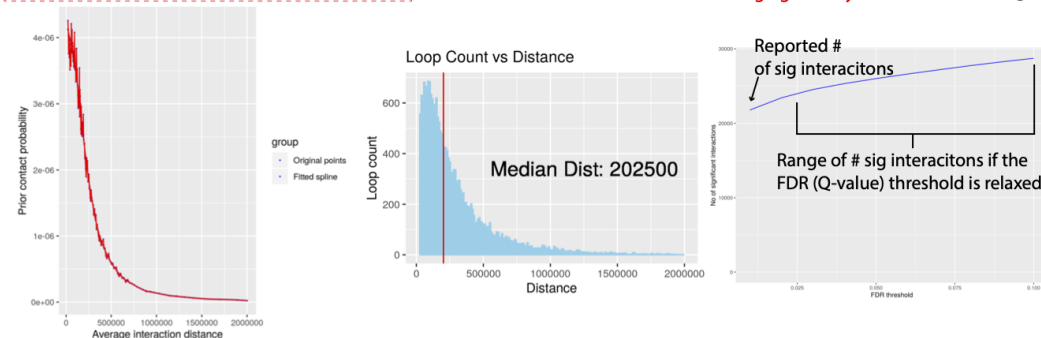
After merging adjacent loops (connected component modeling) significant interactions:

/home/ubuntu/fithichip_CTCF_2.5kb/FitHiChIP_Peak2Peak_b2500_L20000_U2000000/Coverage_Bias/FitHiChIP_BiasCorr/Merge_Nearby_Interactions/CTCF_DS.2.5kb.interactions_FitHiChIP_Q0.01_MergeNearContacts.bed

Total number of significant interactions (FDR 0.01) after merging adjacent loops: 17570 **← Total number of interactions after merging near-by interactions**

Where to find bed file
containing significant
interactions

Where to find bed file
containing merged
significant interactions



1/2

1.6.9 Output

FitHiChIP merged interactions output

1.6.10 What if?

- I don't have a bed file of ChIP-seq anchors or I can't find a representative bed file for my antibody or sample type?
 - Follow our guide to *calling 1-Dimensional peaks with HiChIP data using MACS2*
- I want to use a different tool to identify significant interactions.
 - That is great! This is just one way please refer to tool you'd like to use for documentation. This is just one example of how to find significant interactions in HiChIP data. The key things to consider are the input formats of the data the tool requests.
- I need to do differential analyses.
 - The output of this workflow is nice because the output is a bed file and if you have two samples one could just do a `bedtools intersect` to classify interactions as shared or unique to each sample.

1.6.11 What next?

- Visualization
 - Continue with *plotting HiChIP interactions in R*
 - Import to the Wash-U epigenome browser (more information [in this link](#))

1.7 HiChIP Comparative Analyses

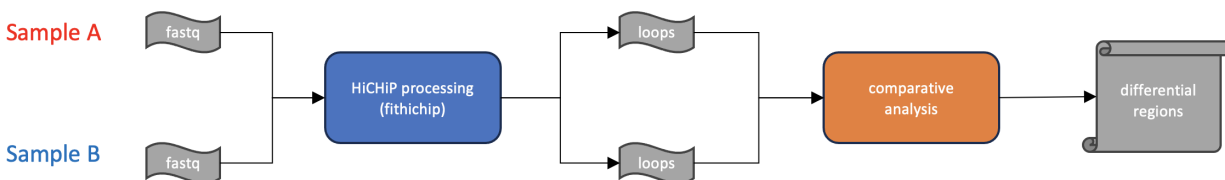
1.7.1 Introduction

Biological questions are seldom answered by analysing single samples in isolation. It is often the case that an experiment aims to make comparisons between two (or more) biological conditions, such as:

- 1) Untreated wild type vs treatment
- 2) Wild type vs knockout
- 3) Normal sample vs tumor

In all cases the goal is to produce a list of differentially interacting regions in one condition relative to the other. The main output for comparative analyses is analogous to what is expected for differential gene expression, where the primary result is a table of regions, the fold change between conditions, and a statistical measure of significance. For HiChIP, the unit for comparison are the loop calls identified using the FitHiChIP software as described in *previous steps*.

Figure 1:



1.7.2 Differential Analysis

Question: How do I perform differential analyses for HiChIP?

Process: Results files from FitHiChIP are used to construct a differential design, and comparison is performed using the scripts bundled with fithichip software.

Results: Final results consist of a table of differentially interacting regions, fold change, and measure of statistical significance.

Files and tools needed:

- FitHiChIP loop calls for each condition: *PREFIX.interactions_FitHiC.bed*
- *FitHiChIP* differential analysis software and scripts
- Associated ChIP-seq peak files [optional]

As the design of differential analysis experiments are unique to each biological question, there are multiple possibilities for how the analysis can be set up. A common scenario is to compare two conditions where each condition has two replicates, and is described in the *FitHiChIP documentation pages*.

Interpreting results:

FitHiChIP differential analysis produces a number of intermediate in addition to the final results table. The most important is the list of significant loops and is named “Loops_EdgeR_Default_SIG.bed”. In general, the interpretation of differential loop analysis is the same as what is familiar for gene expression analysis, where intereactions can be prioritized based on the fold change and statistical significance. An example output file is given below.

chr1	start1	end1	chr2	start2	end2	group1	group2	group3	group4	group5	group6	group7	group8	group9	group10	group11	group12	group13	group14	group15	group16	group17	group18	group19	group20	group21	group22	group23	group24	group25	group26	group27	group28	group29	group30	group31	group32	group33	group34	group35	group36	group37	group38	group39	group40	group41	group42	group43	group44	group45	group46	group47	group48	group49	group50	group51	group52	group53	group54	group55	group56	group57	group58	group59	group60	group61	group62	group63	group64	group65	group66	group67	group68	group69	group70	group71	group72	group73	group74	group75	group76	group77	group78	group79	group80	group81	group82	group83	group84	group85	group86	group87	group88	group89	group90	group91	group92	group93	group94	group95	group96	group97	group98	group99	group100	group101	group102	group103	group104	group105	group106	group107	group108	group109	group110	group111	group112	group113	group114	group115	group116	group117	group118	group119	group120	group121	group122	group123	group124	group125	group126	group127	group128	group129	group130	group131	group132	group133	group134	group135	group136	group137	group138	group139	group140	group141	group142	group143	group144	group145	group146	group147	group148	group149	group150	group151	group152	group153	group154	group155	group156	group157	group158	group159	group160	group161	group162	group163	group164	group165	group166	group167	group168	group169	group170	group171	group172	group173	group174	group175	group176	group177	group178	group179	group180	group181	group182	group183	group184	group185	group186	group187	group188	group189	group190	group191	group192	group193	group194	group195	group196	group197	group198	group199	group200	group201	group202	group203	group204	group205	group206	group207	group208	group209	group210	group211	group212	group213	group214	group215	group216	group217	group218	group219	group220	group221	group222	group223	group224	group225	group226	group227	group228	group229	group230	group231	group232	group233	group234	group235	group236	group237	group238	group239	group240	group241	group242	group243	group244	group245	group246	group247	group248	group249	group250	group251	group252	group253	group254	group255	group256	group257	group258	group259	group260	group261	group262	group263	group264	group265	group266	group267	group268	group269	group270	group271	group272	group273	group274	group275	group276	group277	group278	group279	group280	group281	group282	group283	group284	group285	group286	group287	group288	group289	group290	group291	group292	group293	group294	group295	group296	group297	group298	group299	group300	group301	group302	group303	group304	group305	group306	group307	group308	group309	group310	group311	group312	group313	group314	group315	group316	group317	group318	group319	group320	group321	group322	group323	group324	group325	group326	group327	group328	group329	group330	group331	group332	group333	group334	group335	group336	group337	group338	group339	group340	group341	group342	group343	group344	group345	group346	group347	group348	group349	group350	group351	group352	group353	group354	group355	group356	group357	group358	group359	group360	group361	group362	group363	group364	group365	group366	group367	group368	group369	group370	group371	group372	group373	group374	group375	group376	group377	group378	group379	group380	group381	group382	group383	group384	group385	group386	group387	group388	group389	group390	group391	group392	group393	group394	group395	group396	group397	group398	group399	group400	group401	group402	group403	group404	group405	group406	group407	group408	group409	group410	group411	group412	group413	group414	group415	group416	group417	group418	group419	group420	group421	group422	group423	group424	group425	group426	group427	group428	group429	group430	group431	group432	group433	group434	group435	group436	group437	group438	group439	group440	group441	group442	group443	group444	group445	group446	group447	group448	group449	group450	group451	group452	group453	group454	group455	group456	group457	group458	group459	group460	group461	group462	group463	group464	group465	group466	group467	group468	group469	group470	group471	group472	group473	group474	group475	group476	group477	group478	group479	group480	group481	group482	group483	group484	group485	group486	group487	group488	group489	group490	group491	group492	group493	group494	group495	group496	group497	group498	group499	group500	group501	group502	group503	group504	group505	group506	group507	group508	group509	group510	group511	group512	group513	group514	group515	group516	group517	group518	group519	group520	group521	group522	group523	group524	group525	group526	group527	group528	group529	group530	group531	group532	group533	group534	group535	group536	group537	group538	group539	group540	group541	group542	group543	group544	group545	group546	group547	group548	group549	group550	group551	group552	group553	group554	group555	group556	group557	group558	group559	group560	group561	group562	group563	group564	group565	group566	group567	group568	group569	group570	group571	group572	group573	group574	group575	group576	group577	group578	group579	group580	group581	group582	group583	group584	group585	group586	group587	group588	group589	group590	group591	group592	group593	group594	group595	group596	group597	group598	group599	group600	group601	group602	group603	group604	group605	group606	group607	group608	group609	group610	group611	group612	group613	group614	group615	group616	group617	group618	group619	group620	group621	group622	group623	group624	group625	group626	group627	group628	group629	group630	group631	group632	group633	group634	group635	group636	group637	group638	group639	group640	group641	group642	group643	group644	group645	group646	group647	group648	group649	group650	group651	group652	group653	group654	group655	group656	group657	group658	group659	group660	group661	group662	group663	group664	group665	group666	group667	group668	group669	group670	group671	group672	group673	group674	group675	group676	group677	group678	group679	group680	group681	group682	group683	group684	group685	group686	group687	group688	group689	group690	group691	group692	group693	group694	group695	group696	group697	group698	group699	group700	group701	group702	group703	group704	group705	group706	group707	group708	group709	group710	group711	group712	group713	group714	group715	group716	group717	group718	group719	group720	group721	group722	group723	group724	group725	group726	group727	group728	group729	group730	group731	group732	group733	group734	group735	group736	group737	group738	group739	group740	group741	group742	group743	group744	group745	group746	group747	group748	group749	group750	group751	group752	group753	group754	group755	group756	group757	group758	group759	group760	group761	group762	group763	group764	group765	group766	group767	group768	group769	group770	group771	group772	group773	group774	group775	group776	group777	group778	group779	group780	group781	group782	group783	group784	group785	group786	group787	group788	group789	group790	group791	group792	group793	group794	group795	group796	group797	group798	group799	group800	group801	group802	group803	group804	group805	group806	group807	group808	group809	group810	group811	group812	group813	group814	group815	group816	group817	group818	group819	group820	group821	group822	group823	group824	group825	group826	group827	group828	group829	group830	group831	group832	group833	group834	group835	group836	group837	group838	group839	group840	group841	group842	group843	group844	group845	group846	group847	group848	group849	group850	group851	group852	group853	group854	group855	group856	group857	group858	group859	group860	group861	group862	group863	group864	group865	group866	group867	group868	group869	group870	group871	group872	group873	group874	group875	group876	group877	group878	group879	group880	group881	group882	group883	group884	group885	group886	group887	group888	group889	group890	group891	group892	group893	group894	group895	group896	group897	group898	group899	group900	group901	group902	group903	group904	group905	group906	group907	group908	group909	group910	group911	group912	group913	group914	group915	group916	group917	group918	group919	group920	group921	group922	group923	group924	group925	group926	group927	group928	group929	group930	group931	group932	group933	group934	group935	group936	group937	group938	group939	group940	group941	group942	group943	group944	group945	group946	group947	group948	group949	group950	group951	group952	group953	group954	group955	group956	group957	group958	group959	group960	group961	group962	group963	group964	group965	group966	group967	group968	group969	group970	group971	group972	group973	group974	group975	group976	group977	group978	group979	group980	group981	group982	group983	group984	group985	group986	group987	group988	group989	group990	group991	group992	group993	group994	group995	group996	group997	group998	group999	group1000	group1001	group1002	group1003	group1004	group1005	group1006	group1007	group1008	group1009	group1010	group1011	group1012	group1013	group1014	group1015	group1016	group1017	group1018	group1019	group1020	group1021	group1022	group1023	group1024	group1025	group1026	group1027	group1028	group1029	group1030	group1031	group1032	group1033	group1034	group1035	group1036	group1037	group1038	group1039	group1040	group1041	group1042	group1043	group1044	group1045	group1046	group1047	group1048	group1049	group1050	group1051	group1052	group1053	group1054	group1055	group1056	group1057	group1058	group1059	group1060	group1061	group1062	group1063	group1064	group1065	group1066	group1067	group1068	group1069	group1070	group1071	group1072	group1073	group1074	group1075	group1076	group1077	group1078	group1079	group1080	group1081	group1082	group1083	group1084	group1085	group1086	group1087	group1088	group1089	group1090	group1091	group1092	group1093	group1094	group1095	group1096	group1097	group1098	group1099	group1100	group1101	group1102	group1103	group1104	group1105	group1106	group1107	group1108	group1109	group1110	group1111	group1112	group1113	group1114	group1115	group1116	group1117	group1118	group1119	group1120	group1121	group1122	group1123	group1124	group1125	group1126	group1127	group1128	group1129	group1130	group1131	group1132	group1133	group1134	group1135	group1136	group1137	group1138	group1139	group1140	group1141	group1142	group1143	group1144	group1145	group1146	group1147	group1148	group1149	group1150	group1151	group1152	group1153	group1154	group1155	group1156	group1157	group1158	group1159	group1160	group1161	group1162	group1163	group1164	group1165	group1166	group1167	group1168	group1169	group1170	group1171	group1172	group1173	group1174	group1175	group1176	group1177	group1178	group1179	group1180	group1181	group1182	group1183	group1184	group1185	group1186	group1187	group1188	group1189	group1190	group1191	group1192	group1193	group1194	group1195	group1196	group1197	group1198	group1199	group1200	group1201	group1202	group1203	group1204	group1205	group1206	group1207	group1208	group1209	group1210	group1211	group1212	group1213	group1214	group1215	group1216	group1217	group1218	group1219	group1220	group1221	group1222	group1223	group1224	group1225	group1226	group1227	group1228	group1229	group1230	group1231	group1232	group1233	group1234	group1235	group1236	group1237	group1238	group1239	group1240	group1241	group1242	group1243	group1244	group1245	group1246	group1247	group1248	group1249	group1250	group1251	group1252	group1253	group1254	group1255	group1256	group1257	group1258	group1259	group1260	group1261	group1262	group1263	group1264	group1265	group1266	group1267	group1268	group1269	group1270	group1271	group1272	group1273	group1274	group1275	group1276	group1277	group1278	group1279	group1280	group1281	group1282	group1283	group1284	group1285	group1286	group1287	group1288	group1289	group1290	group1291	group1292	group1293	group1294	group1295	group1296	group1297	group1298	group1299	group1300	group1301	group1302	group1303	group1304	group1305	group1306	group1307	group1308	group1309	group1310	group1311	group1312	group1313	group1314	group1315	group1316	group1317	group1318	group1319	group1320	group1321	group1322	group1323	group1324	group1325	group1326	group1327	group1328	group1329	group1330	group1331	group1332	group1333	group1334	group1335	group1336	group1337	group1338	group1339	group1340	group1341	group1342	group1343	group1344	group13
------	--------	------	------	--------	------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	---------

The most relevant fields from the output will be:

- logFC – the log fold change in coverage between the two conditions
- FDR – a p-value, after correction for multiple hypothesis testing, on the statistical significance of the observed fold change

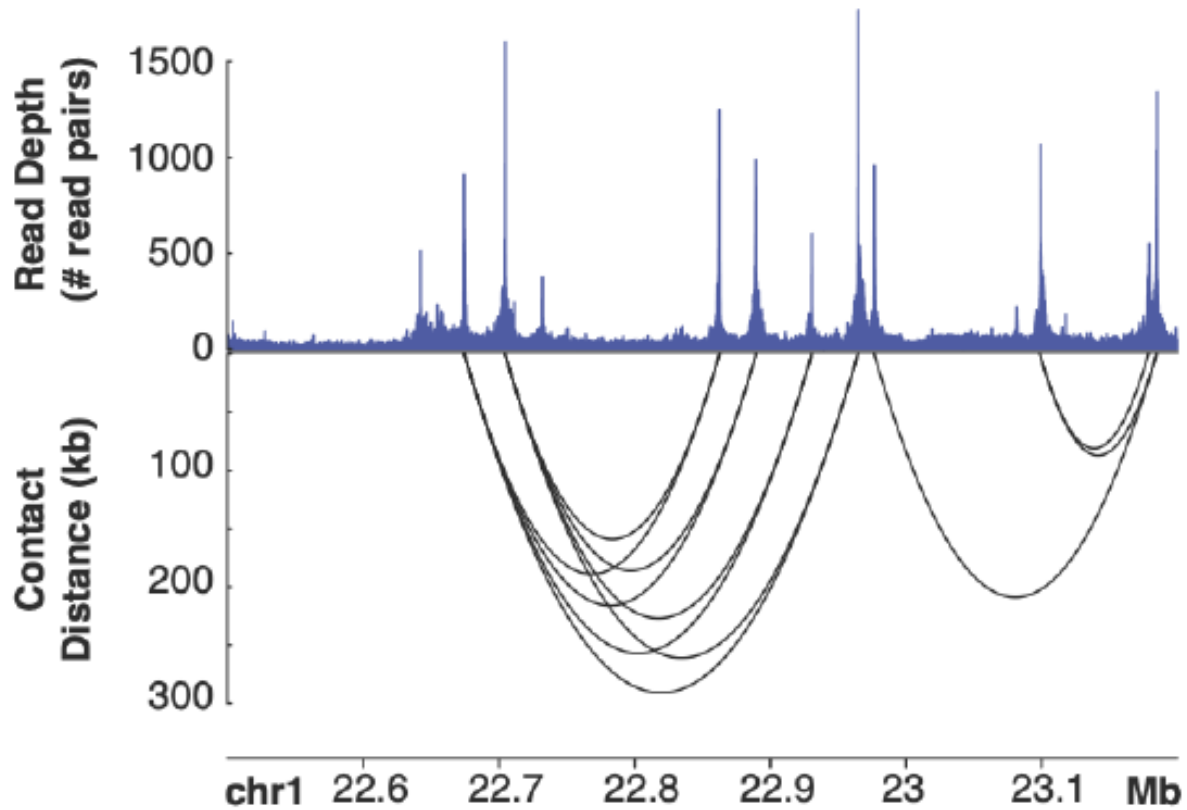
Considerations:

- **Replication** – It is generally advisable to have technical replicates for differential analyses, as this will produce more statistically robust results. FitHiChIP is still able to perform differential analysis with single-replicate samples, and in this case reverts to the square-root-dispersion method used by EdgeR.
- **Paired ChIP-seq experiments** – As mentioned above, it is best practices to have paired ChIP-seq experiments. If that is not do-able, FitHiChIP is bundled with a script that can call peaks *de novo* from the HiChIP data directly.

1.8 Plotting HiChIP Data

1.8.1 Introduction

The purpose of this document is to provide a step-by-step walkthrough to plot significant interactions or “loops” generated through HiChIP data at regions of interests with minimal computational expertise, as seen in the figure below. This workflow assumes you have completed the previous steps *From fastq to final valid pairs bam file*, *Library QC* and *FitHiChIP Loop Calling*. This guide will use the output bam file generated during the data processing and the merged interactions file from FitHiChIP walkthrough. We will be using the bioconductor package *Sushi in R* to plot both the coverage and contact arcs, as in the image below.



1.8.2 Inputs

- *Pairtools bam file*
- *FitHiChIP merged interactions output* e.g. `interactions_FitHiC_Q0.01_MergeNearContacts.bed`

Testing!

If you are looking for a dataset to practice this walkthrough, I recommend the GM12878 CTCF (deep sequencing) from our publicly available [datasets](#)

1.8.3 Tools and Data Used

- `deepTools`
- `R` with the following packages:
 - `Sushi`

1.8.4 Basic Workflow

The basic workflow is as follows:

1. Converting the alignments to a contact matrix and a coverage bedgraph
2. Open R and load libraries
3. Import coverage bedgraph and merged contacts files into R and add a column for distance in merged contacts
4. Set genomic regions
5. Plot and Print

1.8.5 Walkthrough

1. Convert bam to bedgraph with `deepTools bamCoverage -b mapped.bam -of bedgraph -p 36 -o prefix.coverage.bedgraph`

2. Open R

Command:

```
R
```

3. Load libraries

Command:

```
library("Sushi")
```

4. Load data

Command:

```
cov <- read.table("prefix.coverage.bedgraph")
arc <- read.table("prefix_interactions_FitHiC_Q0.01_MergeNearContacts.bed", header=TRUE)
```

5. Inspect arc file structure

Command:

```
head(arc)
```

```
> head(arc)
chr1      s1      e1 chr2      s2      e2 cc P.Value_Bias Q.Value_Bias bin1_low bin1_high bin2_low bin2_high sumCC StrongConn
1 chr1 24202000 24203000 chr1 24391000 24392000 38 5.349360e-28 7.922064e-26 24201000 24203000 24390000 24393000 106 0.8333333
2 chr1 178031000 178032000 chr1 178190000 178191000 10 5.394300e-05 1.008103e-03 178031000 178033000 178189000 178191000 37 1.0000000
3 chr1 204565000 204566000 chr1 204764000 204765000 19 1.193927e-11 5.279233e-10 204565000 204567000 204763000 204765000 62 1.0000000
4 chr1 181190000 181191000 chr1 181423000 181424000 26 2.515409e-18 1.966821e-16 181189000 181191000 181423000 181426000 85 0.6666667
5 chr1 74772000 74773000 chr1 75203000 75204000 58 1.482219e-64 1.187291e-61 74771000 74773000 75203000 75205000 123 1.0000000
6 chr1 12124000 12125000 chr1 12209000 12210000 26 3.558372e-20 3.176450e-18 12124000 12126000 12208000 12210000 89 1.0000000
```

Here we see that the structure of the significant interactions is structured like a bedpe file with position 1 as - chr1, start1, end1 and position 2 – chr2, start2, end2, make up the first six column entries. This is the key structure sushi needs to plot bedpe as “arcs” or “loops”.

The other key factor needed is the height of the arc that Sushi will plot. The rest of the columns point to stats regarding the interactions between position 1 and position 2 that could be used as a height scaler. A common way to plot HiChIP interactions that is visually pleasing is scale the height by the distance of the interaction, therefore we need to add a column of the distance between the start of position 1 and end of position 2

6. Add a column for distance in merged contacts file

Command:

```
arc$dist <- abs(arc$e2 - arc$s1)
```

7. Inspect arc file to see distance

Command:

```
head(arc)
```

```
> head(arc)
  chr1      s1      e1 chr2      s2      e2 cc P.Value_Bias Q.Value_Bias bin1_low bin1_high bin2_low bin2_high sumCC StrongConn dist
1 chr1 24202000 24203000 chr1 24391000 24392000 38 5.349360e-28 7.922064e-26 24201000 24203000 24390000 24393000 106 0.8333333 190000
2 chr1 178031000 178032000 chr1 178190000 178191000 10 5.394300e-05 1.008103e-03 178031000 178033000 178189000 178191000 37 1.0000000 160000
3 chr1 204565000 204566000 chr1 204764000 204765000 19 1.193927e-11 5.279233e-10 204565000 204567000 204763000 204765000 62 1.0000000 200000
4 chr1 181190000 181191000 chr1 181423000 181424000 26 2.515409e-18 1.966821e-16 181189000 181191000 181423000 181426000 85 0.6666667 234000
5 chr1 74772000 74773000 chr1 75203000 75204000 58 1.482219e-64 1.187291e-61 74771000 74773000 75203000 75205000 123 1.0000000 432000
6 chr1 12124000 12125000 chr1 12209000 12210000 26 3.558372e-20 3.176450e-18 12124000 12126000 12208000 12210000 89 1.0000000 86000
```

8. Set region of interest for this example a 1.5 Mb region on chr8

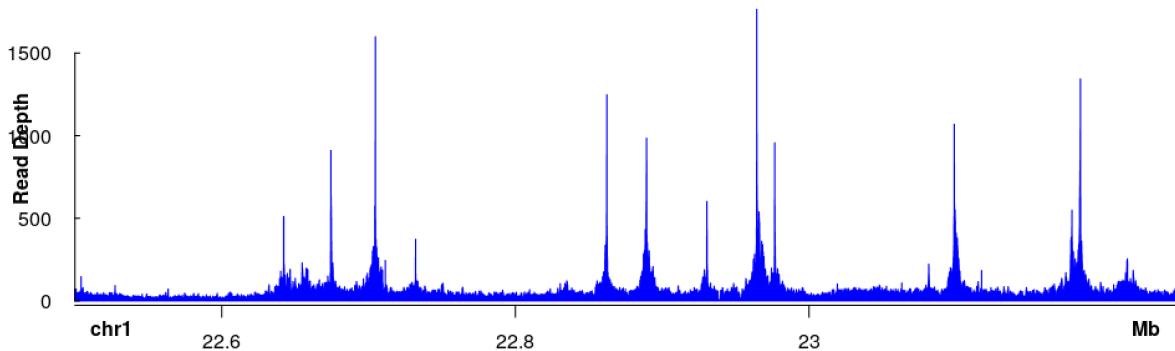
Command:

```
chrom = "chr8"
chromstart = 225000000
chromend = 232000000
```

9. Inspect coverage plot

Command:

```
plotBedgraph(cov, chrom, chromstart, chromend)
labelgenome(chrom, chromstart, chromend, n=4, scale="Mb")
mtext("Read Depth", side=2, line=1.75, cex=1, font=2)
axis(side=2, las=2, tcl=.2)
```



10. Plot arcs with arc heights based on distance

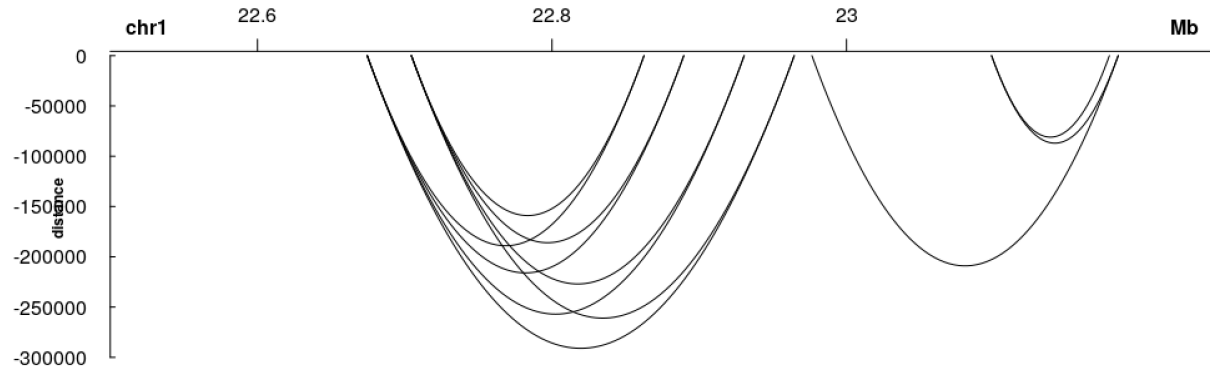
Command:

```
plotBedpe(arc, chrom, chromstart, chromend, heights = arc$dist, plottype="loops", flip=TRUE)
labelgenome(chrom, chromstart, chromend, side=3, n=3, scale="Mb")
```

(continues on next page)

(continued from previous page)

```
axis(side=2,las=2,tcl=.2)
mtext("distance",side=2,line=1.75,cex=.75,font=2)
```

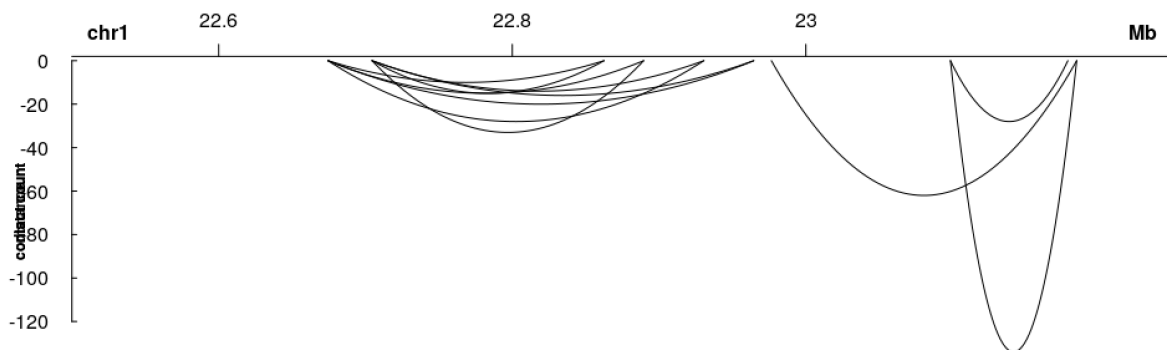


While aesthetically pleasing, the arc file has much more informative information than the distance which is already captured on the x-axis. One could scale the height to the P or Q-values. Or could even add a color scale based on those statistical qualifiers (see the Sushi documentation for other variations on this). To demonstrate an additional layer of information in the arc plot, we can scale the arc height to the number of contacts interacting between position 1 and position 2.

11. Plot arcs with arc heights based on contact frequency

Command:

```
plotBedpe(arc,chrom,chromstart,chromend,heights = arc$sumCC,plottype="loops", flip=TRUE)
labelgenome(chrom, chromstart,chromend,side=3, n=3,scale="Mb")
axis(side=2,las=2,tcl=.2)
mtext("contact freq",side=2,line=1.75,cex=.75,font=2)
```



Finally, we want to generate a PDF file for our records or to clean up in a PDF editor such as Adobe Illustrator.

12. Align and print both plots to a PDF file

Tip!

where “{ }” I’d recommend pasting line-by-line rather than bulk copy and paste

Command:

```
pdfname <- "hichip.cov.arcs.pdf"
makepdf = TRUE
if(makepdf==TRUE)
{
  pdf(pdfname , height=10, width=12)
}

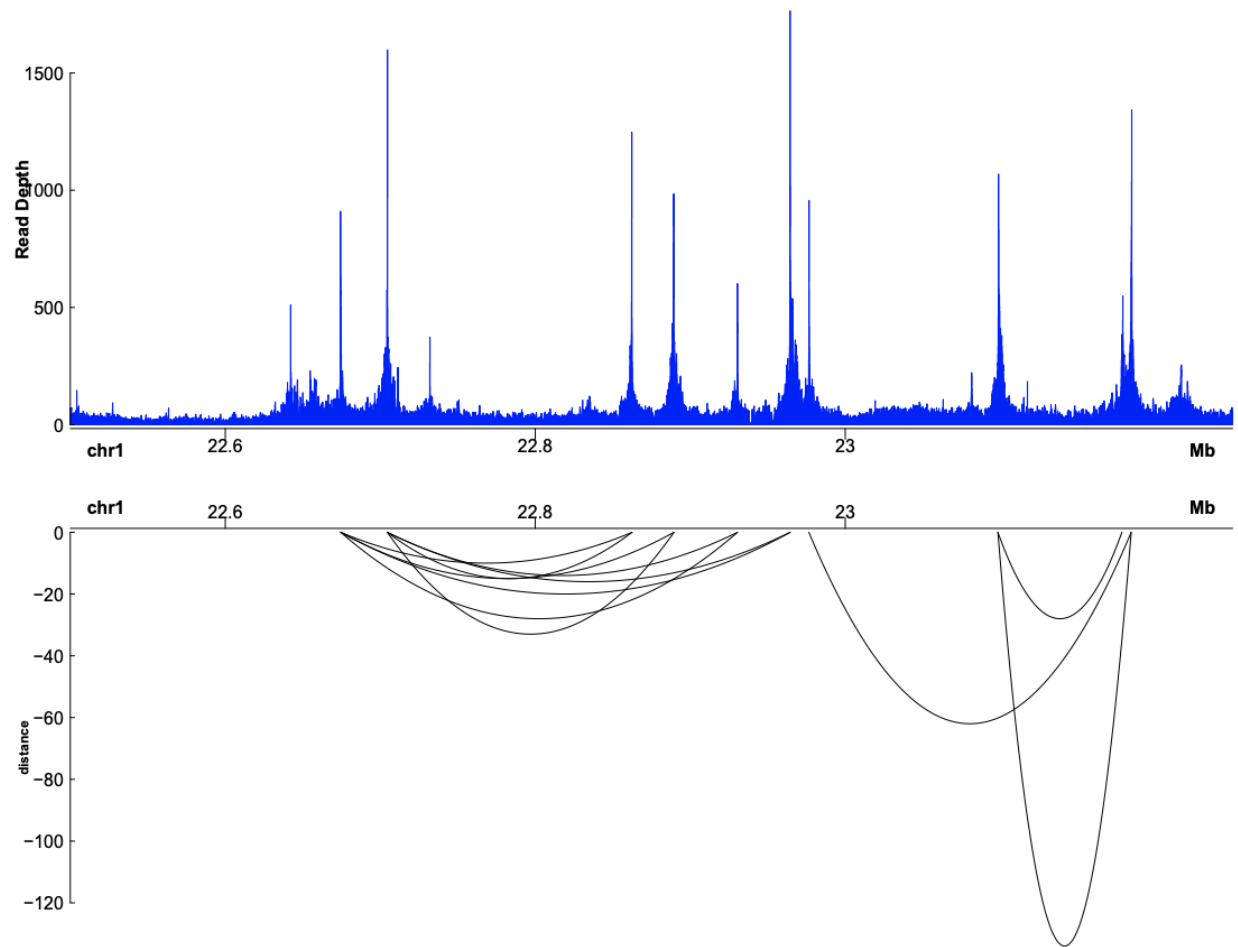
##set layout
layout(matrix(c(1,
  2
), 2,1, byrow=TRUE))
par(mgp=c(3,.3,0))

##plot coverage
par(mar=c(3,4,2,2))
plotBedgraph(cov,chrom,chromstart,chromend)
labelgenome(chrom,chromstart,chromend,n=4,scale="Mb")
mtext("Read Depth",side=2,line=1.75,cex=1,font=2)
axis(side=2,las=2,tcl=.2)

##plot arcs with height based on contact frequency
par(mar=c(3,4,2,2))
plotBedpe(arc,chrom,chromstart,chromend,heights = arc$sumCC,plottype="loops", flip=TRUE)
labelgenome(chrom, chromstart,chromend,side=3, n=3,scale="Mb")
axis(side=2,las=2,tcl=.2)
mtext("distance",side=2,line=1.75,cex=.75,font=2)

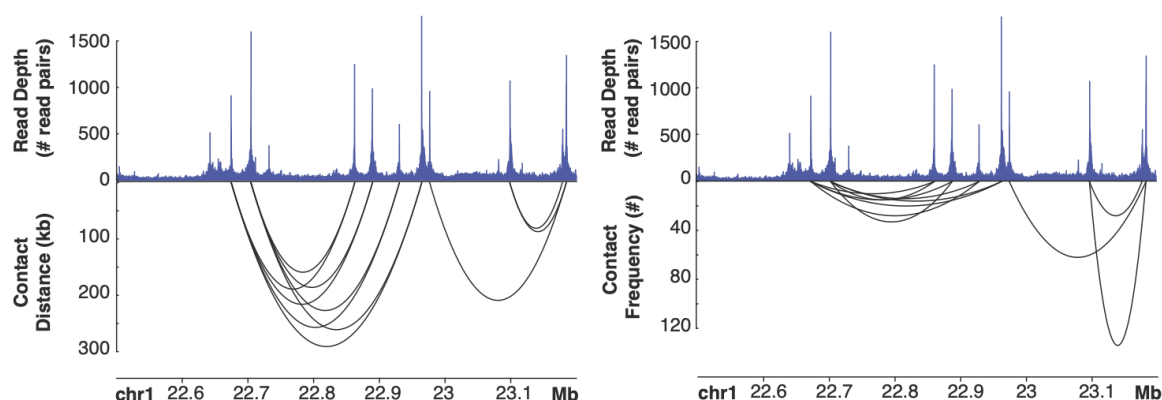
if (makepdf==TRUE)
{
  dev.off()
}
```

The resulting figure should look like the one below:



There're figures, then there are Figures

The outlined workflow provides a rudimentary plot that illustrates the coverage and proximity-ligation links contained in HiChIP data. There is a lot more you can do to beautify the plots or to place the data in context of additional findings. In other words, there is more that should be done to generate a publishable figure. The Bioconductor package ‘Sushi’ has a plethora of ways to customize plots. Further documentation on this can be found [here](#). Alternately the one could clean up the figure in a PDF editor, such as Adobe Illustrator. A few extra minutes in Illustrator provides the final figure below where contact arcs are plotted both by height in reference to the coverage (left) and by contact frequency (right):

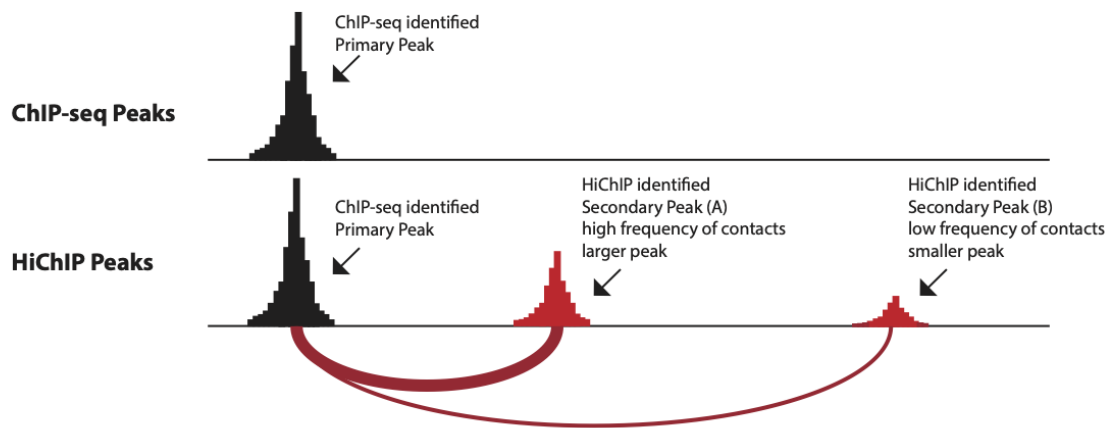


1.9 Calling 1D peaks with MACS2 on HiChIP data

1.9.1 Introduction

Understanding where proteins bind the DNA is a hallmark of ChIP-seq experiments. Typically, [MACS2](#) is used on ChIP-seq data to identify peak signal from the background noise and confirm where these binding sites are located. When it comes to HiChIP these binding sites (or anchors) are important to understand which molecule of the proximity-ligation step occurs at an anchor site or non-anchor site. This information is used to QC the HiChIP library and is a requirement for identifying significant chromatin interactions. You may not have done any ChIP-seq work on a particular sample or maybe there is no publicly available data that reflects your specific sample type or experimental conditions. While it is most ideal to use ChIP-seq derived peak signals, it is possible to use the HiChIP data to call 1-dimensional peaks like you normally would in the ChIP-seq experiment.

There are a few things to keep in mind when using HiChIP data to call 1-D peaks: 1. You may be identifying secondary peaks along with the primary peaks (see figure below) and without a ChIP-seq dataset, it would be hard to discern one peak type from the other. 2. Using .bam files that were processed and filtered through pairtools do not integrate nicely with MACS2. There is a simple solution for that which we will cover here. 3. If you do call peaks with the HiChIP data, you should [run FitHiChIP](#) on both peak-to-peak and peak-to-all settings.



1.9.2 Input files

- *.bam* file generated at the *from fastq to final valid pairs bam file* step.

Testing!

If you are looking for a dataset to practice this walkthrough, I recommend the GM12878 CTCF (deep sequencing) from our publicly available [datasets](#)

1.9.3 Additional tools needed

- [MACS2](#)

1.9.4 Workflow Overview

1. Select the primary alignment in the bam file and convert to bed format.
2. Run MACS2.

1.9.5 Workflow

1. Select the primary alignment in the bam file and convert to bed format.

Command:

```
samtools -view -h -F 0x900 mapped.bam | bedtools bamtobed -i stdin > prefix.primary.aln.  
↪bed
```

Here we're using `samtools -view` function to retain the header (-h) and filter and keep (-F) the primary alignment (flag ID – 0x900) of the input bam file. Then the filtered alignments are being piped into `bedtools` to convert the alignment (bam format) to bed format using the input flag for a UNIX piped input (stdin). Resulting in a final bed file.

2. Run MACS2.

Command:

```
Macs2 callpeak -t prefix.primary.aln.bed -n prefix.macs2
```

Here we are using `macs2 callpeak` function the treatment file (-t) which is the primary alignment bed file, with a particular prefix assigned to the outputs (-n).

1.10 HiChIP Data Sets

To download one of the data sets, simply use the `wget` command:

```
wget https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/HiChIP-CTCF_2M_R1.fastq.gz  
wget https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/HiChIP-CTCF_2M_R2.fastq.gz
```

For testing purposes, we recommend using the 2M reads data sets, for any other purpose we recommend using the 800M reads data set.

1.10.1 Sequenced (human) libraries:

Library	Link
GM12878 CTCF 2M	<ul style="list-style-type: none">• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/HiChIP_CTCF_2M_R1.fastq.gz• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/HiChIP_CTCF_2M_R2.fastq.gz
GM12878 CTCF (deep sequencing)	<ul style="list-style-type: none">• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/CTCF-DS_R1.fastq.gz• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/CTCF-DS_R2.fastq.gz
GM12878 H3K27Ac (deep sequencing)	<ul style="list-style-type: none">• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/H3K27Ac_R1.fastq.gz• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/H3K27Ac_R2.fastq.gz
GM12878 H3K4me3 (deep sequencing)	<ul style="list-style-type: none">• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/H3K4me3_R1.fastq.gz• https://s3.amazonaws.com/dovetail.pub/HiChIP/fastqs/H3K4me3_R2.fastq.gz

1.10.2 Human, hg38, Peak files from ENCODE project

Sample	Target	Accession	URL	Output type
GM12878	CTCF	ENCFF017XLW	https://www.encodeproject.org/files/ENCFF017XLW/@@download/ENCFF017XLW.bed.gz	conservative IDR thresholded peaks
IMR-90	H3K4ac	ENCFF823NUO	https://www.encodeproject.org/files/ENCFF823NUO/@@download/ENCFF823NUO.bed.gz	replicated peaks
GM12878	H3K4me3	ENCFF188SZS	https://www.encodeproject.org/files/ENCFF188SZS/@@download/ENCFF188SZS.bed.gz	replicated peaks
IMR-90	H3K14ac	ENCFF106EAN	https://www.encodeproject.org/files/ENCFF106EAN/@@download/ENCFF106EAN.bed.gz	replicated peaks
GM12878	H3K27ac	ENCFF367KIF	https://www.encodeproject.org/files/ENCFF367KIF/@@download/ENCFF367KIF.bed.gz	replicated peaks
GM12878	H3K27me3	ENCFF153VOQ	https://www.encodeproject.org/files/ENCFF153VOQ/@@download/ENCFF153VOQ.bed.gz	replicated peaks
GM12878	H3K36me3	ENCFF268HMO	https://www.encodeproject.org/files/ENCFF268HMO/@@download/ENCFF268HMO.bed.gz	replicated peaks
GM12878	SMC3	ENCFF534PUK	https://www.encodeproject.org/files/ENCFF534PUK/@@download/ENCFF534PUK.bed.gz	bed
MCF-7	Klf4	ENCFF287QDZ	https://www.encodeproject.org/files/ENCFF287QDZ/@@download/ENCFF287QDZ.bed.gz	conservative IDR thresholded peaks
GM23338	Nanog	ENCFF897LBK	https://www.encodeproject.org/files/ENCFF897LBK/@@download/ENCFF897LBK.bed.gz	conservative IDR thresholded peaks
GM12878	POLR2A	ENCFF794VYB	https://www.encodeproject.org/files/ENCFF794VYB/@@download/ENCFF794VYB.bed.gz	conservative IDR thresholded peaks

1.10.3 Data used for HiChIP Comparative Analysis (Mouse, mm10)

To get a list of all the files generated from the HiChIP Comparative Analysis tutorial, including the required reference genomes, you can use the command:

```
aws s3 ls s3://dovetail.pub/HiChIP/compare_samples/
```

Use wget to download any given file, replacing “s3://” with “https://s3.amazonaws.com/”, followed by the remaining path to the file. For example:

```
wget https://s3.amazonaws.com/dovetail.pub/HiChIP/compare_samples/Reference_Genome/mm10.  
↪fa
```

Data Set	Link
Fastqs (Sample A)	<ul style="list-style-type: none"> • https://s3.amazonaws.com/dovetail.pub/HiChIP/compare_samples/fastq_inputs/sampleA_R1.fastq.gz • https://s3.amazonaws.com/dovetail.pub/HiChIP/compare_samples/fastq_inputs/sampleA_R2.fastq.gz
Fastqs (Sample B)	<ul style="list-style-type: none"> • https://s3.amazonaws.com/dovetail.pub/HiChIP/compare_samples/fastq_inputs/sampleB_R1.fastq.gz • https://s3.amazonaws.com/dovetail.pub/HiChIP/compare_samples/fastq_inputs/sampleB_R2.fastq.gz

Note: The full dataset, including input files and generated output is ~183Gb (roughly 5h with a network speed of 10Mb/s).

1.11 Support

For help or questions related please open a new issue on the github repository or send an email to: support@dovetail-genomics.com

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`