Analyzing Hi-C data with the HiTC BioC package

Nicolas Servant

May 1, 2024

1 Introduction

The Hi-C technic was first introduced by Lieberman-Aiden et al. [2009] to simultaneously detect all chromosomal interactions in a single experiment. The Hi-C aims at measuring the population-averaged frequency at which two genomic loci physically interact in three-dimensional space. Briefly, after a first crosslink and digestion with a restriction enzyme, all genomic fragments are labeled with a biotinylated nucleotide before ligation. These junctions can then be purified efficiently by streptavidin-coated magnetic beads, and finally sequenced using an Illumina paired-end protocol.

After sequencing, raw reads have to be processed to generate the inter/intrachromosomal contact maps. The main steps of this processing are described in Imakaev et al. [2012]. The raw paired-end reads are first independently aligned on a reference genome. The two mates from the same DNA fragment therefore define the two interacting loci on the genome. Imakaev et al. also proposed an iterative mapping procedure to rescue the reads spanning the ligation junction (and thus containing the two interactors in a single read).

After the reads alignment, the Hi-C molecule generated from the DNA digestion and the ligation products are reconstructed using the position and direction of the sequenced mates. Self-circle ligations, single side reads and dangling ends are discarded, and the valid ligation products aligned to different restriction fragments and face toward the restriction site are used to reconstruct the contact maps. The interaction frequencies are therefore estimated by counting how many times two genomic bins (at a given resolution) were found as interactors. The data resolution usually depends on the sequencing depth. In their first paper, Lieberman-Aiden et al. [2009] generated data at a resolution of 1Mb (up to 100kb) and reveal the compartmentalization of the genome into regions of open and closed (active and inactive) chromatin as well as the three-dimensional structure of the genome (fractal globule). More recently, Dixon et al. [2012] generated 20 to 40kb contact maps in order to go deeper in the conformation structure and to study topological domains (Nora et al. [2012]).

The HiTC is designed to import processed data as contact maps. In order to generate these maps from raw sequencing reads a couple of pipeline are available on the web. We have recently developed the HiC-Pro software which is an optimized pipeline to process Hi-C data from raw reads to normalized contact maps Servant et al. [2015]. The outputs of HiC-Pro is fully compatible with the HiTC package.

This vignette is based on the analysis of the Dixon et al. [2012] contact maps, at
Analyzing Hi-C data with the HiTC BioC package

a resolution of 40kb. These data are stored as a HTClist object, i.e. a list of inter/intrachromosomal contact maps, one for each pair of chromosomes. The goal of this vignette is to describe how the HiTC R package can be used to explore such data.

If you use HiTC for analyzing your data, please cite:


2 Working with Hi-C data

The HTCexp (High-Throughput 'C' experiment) class aims at representing a single 'C' experiment, characterized by:

- A contact map (i.e a Matrix)
- Two GRanges objects that describe each feature of the contact map, respectively, the x (i.e. columns) and y (i.e. rows) labels of the matrix. In the context of 5C, these two GRanges objects will describe the set of forward and reverse primers, and for the Hi-C the binned genomic intervals at a given resolution.

Whereas a 5C dataset is usually composed of a single intrachromosomal contact map (i.e. HTCexp object), a Hi-C dataset is represented by a list of inter/intrachromosomal contact maps, characterized by the physical interactions of each pair of chromosomes. The HTClist was designed as a list of HTCexp objects, with a couple of dedicated methods.

Working at a resolution of 40kb (or even at a lower resolution) can result in an intensive memory usage. Assuming that every restriction fragment could ligate to any other, there are on the order of $10^{11}$ possible HindIII restriction fragment pairs in the Human genome. In addition to the fact that the generation of a Hi-C library with enough complexity or sequence depth to cover all possible restriction fragment interactions is difficult, this will result in contact maps with a very high level of sparsity. The HiTC package provides an efficient memory storage of the data, based on the sparseMatrix class and the Matrix R package. In addition, binned intrachromosomal maps are expected to be symmetrical around the diagonal and can thus be stored as symmetrical matrix (dsCMatrix Matrix class). In the same way, the interchromosomal maps are also stored once as the chr1-chr2 map is expected to be the transposed of the chr2-chr1 map.

> require(HiTC)
> require(HiCDataHumanIMR90)
Analyzing Hi-C data with the HiTC BioC package

```r
> data(Dixon2012.IMR90)
> show(hic_imr90.40)

HTClist object of length 325
25 intra / 300 inter-chromosomal maps

> class(intdata(hic_imr90.40$chr1chr1))
[1] "dsCMatrix"
attr(,"package")
[1] "Matrix"

> object.size(hic_imr90.40)
1209797816 bytes
```

3 Description of the Hi-C data

The HiTC package provides several methods to describe a HTClist object.

```r
> ## Show data
> show(hic_imr90.40)

HTClist object of length 325
25 intra / 300 inter-chromosomal maps

> ## Is my data complete (i.e. composed of intra + the related inter chromosomal maps)
> isComplete(hic_imr90.40)

[1] TRUE

> ## Note that a complete object is not necessarily pairwise
> ## (is both chr1-chr2 and chr2-chr1 stored ?)
> isPairwise(hic_imr90.40)

[1] FALSE

> ## Which chromosomes ?
> seqlevels(hic_imr90.40)

[1] "chr1" "chr2" "chr3" "chr4" "chr5" "chr6" "chr7" "chr8" "chr9"
[10] "chr10" "chr11" "chr12" "chr13" "chr14" "chr15" "chr16" "chr17" "chr18"
[19] "chr19" "chr20" "chr21" "chr22" "chrX" "chrY" "chrM"

> ## Details about a given map
> detail(hic_imr90.40$chr6chr6)

HTC object
Focus on genomic region [chr6:1-171115067]
CIS Interaction Map
Matrix of Interaction data: [4278-4278]
```
Binned data - window size = 40000
4278 genome intervals
Total Reads = 21565450
Number of Interactions = 3657661
Median Frequency = 1
Sparsity = 0.1

> ## Descriptive statistics
> head(summary(hic_imr90_40))

| seq1 seq2 nbreads nbinteraction averagefreq medfreq sparsity |
|-------------------|-----------------|-----------------|-----------------|-----------------|
| chr1chr1 chr1 chr1 | 25914788 | 4524734 | 5.7274 | 1 | 0.8835 |
| chr1chr2 chr1 chr2 | 504332 | 497291 | 1.0142 | 1 | 0.9869 |
| chr1chr3 chr1 chr3 | 440865 | 434917 | 1.0137 | 1 | 0.9859 |
| chr1chr4 chr1 chr4 | 456924 | 450005 | 1.0154 | 1 | 0.9849 |
| chr1chr5 chr1 chr5 | 399067 | 393926 | 1.0131 | 1 | 0.986 |
| chr1chr6 chr1 chr6 | 382580 | 377654 | 1.013 | 1 | 0.9858 |

4 Hi-C Visualization

The Hi-C data can be visualized as contact maps, representing the contact frequencies between two chromosomes, or at the level of the genome. By default, objects from the HTClist class will be represented as a heatmap, whereas object from the HTCexp class (i.e. single map) as a triangle. Depending on what you want to visualize the resolution of the map can also be changed (from high to lower resolution).

> ## Go back to a smaller dataset (chr21, 22, X) at lower resolution
> sset <- reduce(hic_imr90_40, chr=c("chr5","chr6","chr7"))
> imr90_500 <- HTClist(lapply(sset, binningC,
+               binsize=500000, bin.adjust=FALSE, method="sum", step=1))
> mapC(imr90_500)
Analyzing Hi-C data with the HiTC BioC package

As we can see on this example, only half of the inter-chromosomal maps are stored and thus plotted. To display the full pairwise maps, methods such as `forcePairwise`, or `forceSymmetric` can be used to switch from a pairwise (and more memory consuming) form to a reduced form.

```r
> mapC(forcePairwise(imr90_500), maxrange=200)
```
5 Hi-C Normalization

5.1 Back to restriction fragments

In addition to descriptive methods on the HTCList object, the HiTC package provides functions to extract biological information related to the data processing. These functions are useful for data normalization, in order to extract the expected bias at the level of the restriction fragment.

```r
> ## Example on chromosome X
> ## GRanges of restriction fragments after HindIII digestion
> resFrag <- getRestrictionFragmentsPerChromosome(resSite="AAGCTT", overhangs5=1, +
> chromosomes="chr6", +
> genomePack="BSgenome.Hsapiens.UCSC.hg18")
> resFrag

[[1]]
GRanges object with 51298 ranges and 0 metadata columns:

<table>
<thead>
<tr>
<th>seqnames</th>
<th>ranges</th>
<th>strand</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6</td>
<td>1-10366</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>10367-18359</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>18360-19010</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>19011-24020</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>24021-25008</td>
<td>+</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>chr6</td>
<td>170887666-170888246</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>170888247-170891362</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>170891363-170893508</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>170893509-170896737</td>
<td>+</td>
</tr>
<tr>
<td>chr6</td>
<td>170896738-170899992</td>
<td>+</td>
</tr>
</tbody>
</table>

seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

5.2 Local genomic feature (LGF) normalization

As any sequencing application, the Hi-C library preparation contains bias, which can be broadly classified as ligation bias and sequence content bias. These effects were first described by Yaffe and Tanay [2011] and require appropriate normalization methods.

Hu et al. [2012] recently proposed a linear model strategy to normalize the data. Their method (named HiCNorm) requires that the bias was inferred from the restriction fragments and then used at the Hi-C resolution. The getAnnotatedRestrictionSites function aims at annotating the restriction fragments according to their mappability (optional), GC content and effective length features. The local genomic features can be then assign to each genomic region, and normalized using the Hu et al. [2012]
Analyzing Hi-C data with the \textit{HiTC BioC} package

method. In the following example, we will focus on chromosome 6 only. The same code can be easily applied on a \textit{HTClist} object using the \texttt{lapply} function. In the same way, we will not use here the mappability information for space issue. In practice, the mappability track can be downloaded from the ENCODE project data, and is important to normalize the Hi-C data.

```r
> ## Annotation of genomic features for LGF normalization
> ## Example on chromosome 6
>
> ## Load mappability track
> require(rtracklayer)
> map_hg18 <- import("wgEncodeCrgMapabilityAlign100mer_chr6.bw",format="BigWig")
> map_hg18 <- NULL
> cutSites <- getAnnotatedRestrictionSites(resSite="AAGCTT", overhangs5=1,
+ chromosomes="chr6",
+ genomePack="BSgenome.Hsapiens.UCSC.hg18",
+ wingc=200, mappability=map_hg18, winmap=500)
> head(cutSites)

GRangesList object of length 1:
$chr6
GRanges object with 51297 ranges and 6 metadata columns:

<table>
<thead>
<tr>
<th>seqnames</th>
<th>ranges</th>
<th>strand</th>
<th>len_U</th>
<th>len_D</th>
<th>GC_U</th>
<th>GC_D</th>
<th>map_U</th>
<th>map_D</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6</td>
<td>10367-10370</td>
<td>*</td>
<td>10370</td>
<td>7996</td>
<td>0.370</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>chr6</td>
<td>18360-18363</td>
<td>*</td>
<td>7996</td>
<td>654</td>
<td>0.385</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>chr6</td>
<td>19011-19014</td>
<td>*</td>
<td>654</td>
<td>5013</td>
<td>0.300</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>chr6</td>
<td>24021-24024</td>
<td>*</td>
<td>5013</td>
<td>991</td>
<td>0.395</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>chr6</td>
<td>25009-25012</td>
<td>*</td>
<td>991</td>
<td>4638</td>
<td>0.345</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

... ... ... ... ... ...
[51293] chr6 170887666-170887669 * | 269 584 0.555
[51294] chr6 170888247-170888250 * | 584 3119 0.555
[51295] chr6 170891363-170891366 * | 3119 2149 0.350
[51296] chr6 170893509-170893512 * | 2149 3232 0.405
[51297] chr6 170896738-170896741 * | 3232 3254 0.555

GC_D | map_U | map_D

| [1] | 0.370 | NA | NA |
| [2] | 0.350 | NA | NA |
| [3] | 0.260 | NA | NA |
| [4] | 0.340 | NA | NA |
| [5] | 0.335 | NA | NA |

... ... ...
[51293] 0.465 NA NA
[51294] 0.415 NA NA
```
Analyzing Hi-C data with the *HiTC* BioC package

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[51295]</td>
<td>0.290</td>
<td>NA</td>
</tr>
<tr>
<td>[51296]</td>
<td>0.330</td>
<td>NA</td>
</tr>
<tr>
<td>[51297]</td>
<td>0.475</td>
<td>NA</td>
</tr>
</tbody>
</table>

-------

seqinfo: 1 sequence from an unspecified genome; no seqlengths

```r
> ## Annotation of Hi-C object
> imr90_500_chr6annot <- setGenomicFeatures(imr90_500$chr6chr6, cutSites)
> y_intervals(imr90_500_chr6annot)
```

GRanges object with 343 ranges and 3 metadata columns:

<table>
<thead>
<tr>
<th>seqnames</th>
<th>ranges</th>
<th>strand</th>
<th>len</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6:1-5e+05</td>
<td>chr6 1-500000</td>
<td>*</td>
<td>127453</td>
</tr>
<tr>
<td>chr6:500001-1e+06</td>
<td>chr6 500001-1000000</td>
<td>*</td>
<td>124817</td>
</tr>
<tr>
<td>chr6:1000001-1500000</td>
<td>chr6 1000001-1500000</td>
<td>*</td>
<td>103076</td>
</tr>
<tr>
<td>chr6:1500001-2e+06</td>
<td>chr6 1500001-2000000</td>
<td>*</td>
<td>127725</td>
</tr>
<tr>
<td>chr6:2000001-2500000</td>
<td>chr6 2000001-2500000</td>
<td>*</td>
<td>122886</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>seqnames</th>
<th>ranges</th>
<th>strand</th>
<th>len</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6:169000001-169500000</td>
<td>chr6 169000001-169500000</td>
<td>*</td>
<td>110009</td>
</tr>
<tr>
<td>chr6:169500001-1.7e+08</td>
<td>chr6 169500001-1700000000</td>
<td>*</td>
<td>118434</td>
</tr>
<tr>
<td>chr6:170000001-1705000000</td>
<td>chr6 170000001-1705000000</td>
<td>*</td>
<td>68293</td>
</tr>
<tr>
<td>chr6:170500001-1.71e+08</td>
<td>chr6 170500001-1710000000</td>
<td>*</td>
<td>114877</td>
</tr>
<tr>
<td>chr6:171000001-171115066</td>
<td>chr6 171000001-171115066</td>
<td>*</td>
<td>0</td>
</tr>
</tbody>
</table>

GC map

<table>
<thead>
<tr>
<th>seqnames</th>
<th>&lt;numeric&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6:1-5e+05</td>
<td>0.419  NaN</td>
</tr>
<tr>
<td>chr6:500001-1e+06</td>
<td>0.419  NaN</td>
</tr>
<tr>
<td>chr6:1000001-1500000</td>
<td>0.433  NaN</td>
</tr>
<tr>
<td>chr6:1500001-2e+06</td>
<td>0.400  NaN</td>
</tr>
<tr>
<td>chr6:2000001-2500000</td>
<td>0.394  NaN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>seqnames</th>
<th>&lt;numeric&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr6:169000001-169500000</td>
<td>0.424  NaN</td>
</tr>
<tr>
<td>chr6:169500001-1.7e+08</td>
<td>0.411  NaN</td>
</tr>
<tr>
<td>chr6:170000001-1705000000</td>
<td>0.473  NaN</td>
</tr>
<tr>
<td>chr6:170500001-1.71e+08</td>
<td>0.418  NaN</td>
</tr>
<tr>
<td>chr6:171000001-171115066</td>
<td>NaN  NaN</td>
</tr>
</tbody>
</table>

-------

seqinfo: 1 sequence from an unspecified genome; no seqlengths

```r
> ## LGF normalization
> imr90_500_LGF <- normLGF(imr90_500_chr6annot)
```
5.3 Iterative correction and eigenvector decomposition (ICE) normalization

The ICE normalization is one of the most popular methods to correct data from Hi-C bias. This method is based on the assumption of equal visibility of each genomic locus. The matrix of contact probabilities, $M$, for all given pairs of regions $(i, j)$ is thus normalized such as $\sum_{i \neq j \neq j \pm 1} M_{ij} = 1$ for each region $j$. Note that running the ICE normalization method can be memory consuming because it uses the full genome matrix, and then stores the bias vectors. If we advice to use apply the ICE normalization on a full Hi-C dataset (using inter and intrachromosomal maps), the HiTC package also allows to run it on a single intrachromosomal map.

```r
> imr90_500_ICE <- normICE(imr90_500, max_iter=10)
> mapC(HTClist(imr90_500_ICE$chr6chr6), trim.range=.95, + col.pos=c("white", "orange", "red", "black"))
```

6 Detection of Topological Domains

Recent studies on a high resolution human and mouse Hi-C dataset have discovered that the genome organization can be further divided into megabase-long and evolutionarily conserved topological domains (TADs), with high frequencies of intra-domain chromatin interactions but infrequent inter-domain chromatin interactions (Nora et al. [2012], Dixon et al. [2012]). More recently, Phillips-Cremins et al. [2013] have demonstrated that the cell-type-specific chromatin organization seems to occur at the sub-megabase scale involving different ligation proteins and epigenomic mech-
Analyzing Hi-C data with the HiTC BioC package

The following code shows how to focus on TADs, such as the ones described in IMR90 around the Hox genes.

\[
> \text{hox} \leftarrow \text{extractRegion(hic.imr90.40$chr6\_chr6, chr="chr6", from=50e6, to=58e6)} \\
> \text{plot(hox, maxrange=50, col.pos=c("white", "orange", "red", "black"))}
\]

Different algorithms have been proposed to detect TADs. The directionality index was proposed by Dixon et al. [2012] as an input to their HMM model.

\[
> \text{di<-directionalityIndex(hox)} \\
> \text{barplot(di, col=ifelse(di>0,"darkred","darkgreen"))}
\]
Analyzing Hi-C data with the *HiTC* BioC package

### Package versions

This vignette was generated using the following package versions:

- **R version 4.4.0 beta (2024-04-15 r86425), x86_64-pc-linux-gnu**
- **Running under: Ubuntu 22.04.4 LTS**
- **Matrix products: default**
- **BLAS: /home/biocbuild/bbs-3.19-bioc/R/lib/libRblas.so**
- **LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0**
- **Base packages: base, datasets, graphics, grDevices, methods, stats, stats4, utils**
- **Other packages: BiocGenerics 0.50.0, BiocIO 1.14.0, Biostrings 2.72.0, BSgenome 1.72.0, BSgenome.Hsapiens.UCSC.hg18 1.3.1000, GenomeInfoDb 1.40.0, GenomicRanges 1.56.0, HiCDataHumanIMR90 1.23.0, HiTC 1.48.0, IRanges 2.38.0, rtracklayer 1.64.0, S4Vectors 0.42.0, XVector 0.44.0**
- **Loaded via a namespace (and not attached): abind 1.4-5, Biobase 2.64.0, BiocManager 1.30.22, BiocParallel 1.38.0, BiocStyle 2.32.0, bitops 1.0-7, cli 3.6.2, codetools 0.2-20, compiler 4.4.0, crayon 1.5.2, curl 5.2.1, DelayedArray 0.30.0, digest 0.6.35, evaluate 0.23, fastmap 1.1.1, GenomeInfoDbData 1.2.12, GenomicAlignments 1.40.0, grid 4.4.0, htmltools 0.5.8.1, httr 1.4.7, jsonlite 1.8.8, knitr 1.46, lattice 0.22-6, Matrix 1.7-0, MatrixGenerics 1.16.0, matrixStats 1.3.0, parallel 4.4.0, R6 2.5.1, RCcolorBrewer 1.1-3, RCurl 1.98-1.14, restfulr 0.0.15, rjson 0.2.21, rlang 1.1.3, rmarkdown 2.26, Rsamtools 2.20.0, S4Arrays 1.4.0, SparseArray 1.4.0, SummarizedExperiment 1.34.0, tools 4.4.0, UCSC.utils 1.0.0, xfun 0.43, XML 3.99-0.16.1, yaml 2.3.8, zlibbioc 1.50.0**
Analyzing Hi-C data with the *HiTC BioC* package

**Acknowledgements**

Many thanks to Nelle Varoquaux and Pierre Gestraud for useful discussion and help in developing this R package. Thank you to Ming Hu who developed the HiCNorm method, and help us in the integration of its method in the HiTC package. A special thanks to the *HiTC* users for useful discussions and idea to improve it.
Analyzing Hi-C data with the HiTC BioC package

References


