Analyzing Hi-C data with the HiTC BioC package

Nicolas Servant

October 13, 2015

1 Introduction

The Hi-C technic was first introduced by Lieberman-Aiden et al. [2009] to simultaneoulsy 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 finaly 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 resulution) 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 optimzed pipeline to process Hi-C data from raw reads to normalized contact maps. 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 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:

- Servant N., Lajoie B.R., Nora E.P., Giorgetti L., Chen C., Heard E., Dekker J., Barillot E. (2012) HiTC : Exploration of High-Throughput 'C' experiments. *Bioinformatics*.

2 Working with Hi-C data

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

- A contact map (i.e a *Matrix*)
- Two *GRanges* objects that describe each features 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)
> 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)
```

```
1161141408 bytes
```

3 Description of the Hi-C data

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

```
> ## 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"  "chr10"
[11] "chr11" "chr12" "chr13" "chr14" "chr15" "chr16" "chr17" "chr18" "chr19" "chr20"
[21] "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
```

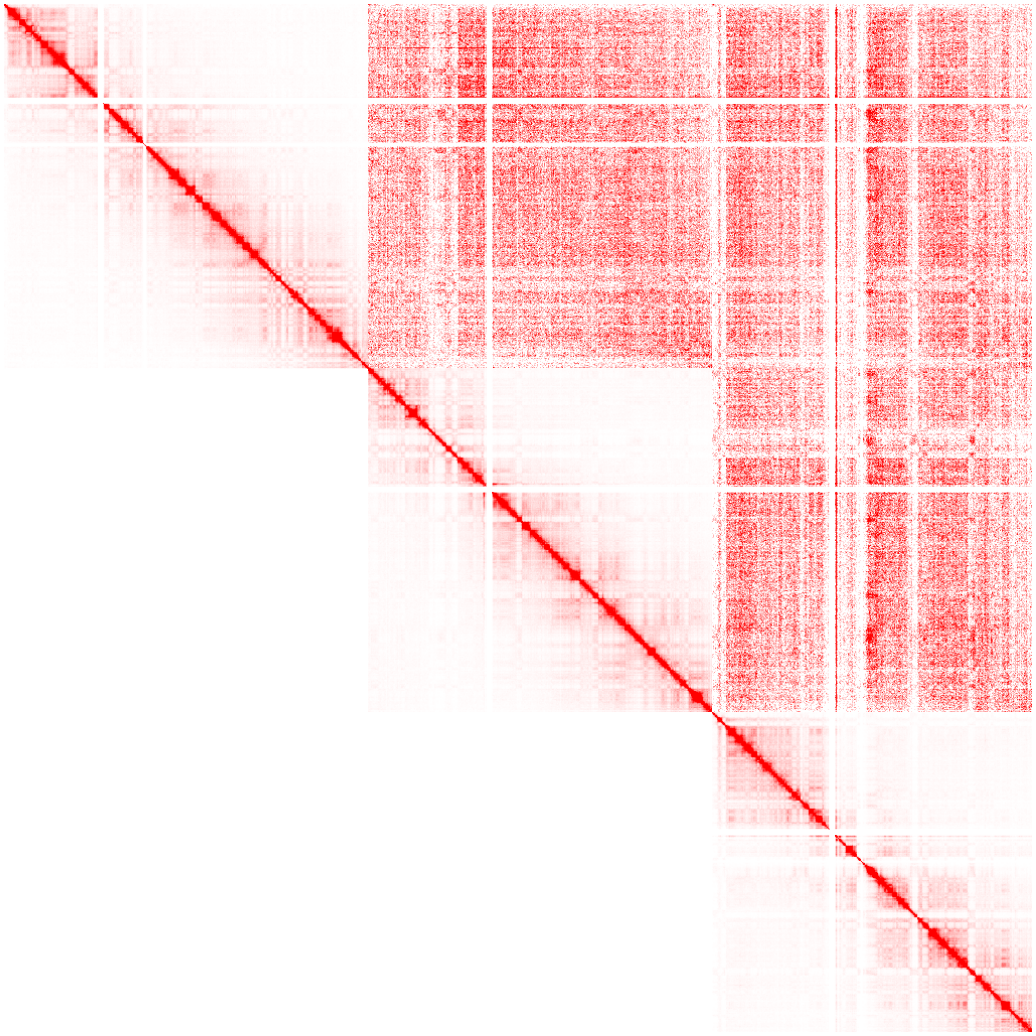
```
> ## Descriptive statistics
> head(summary(hic_imr90_40))
```

```
         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

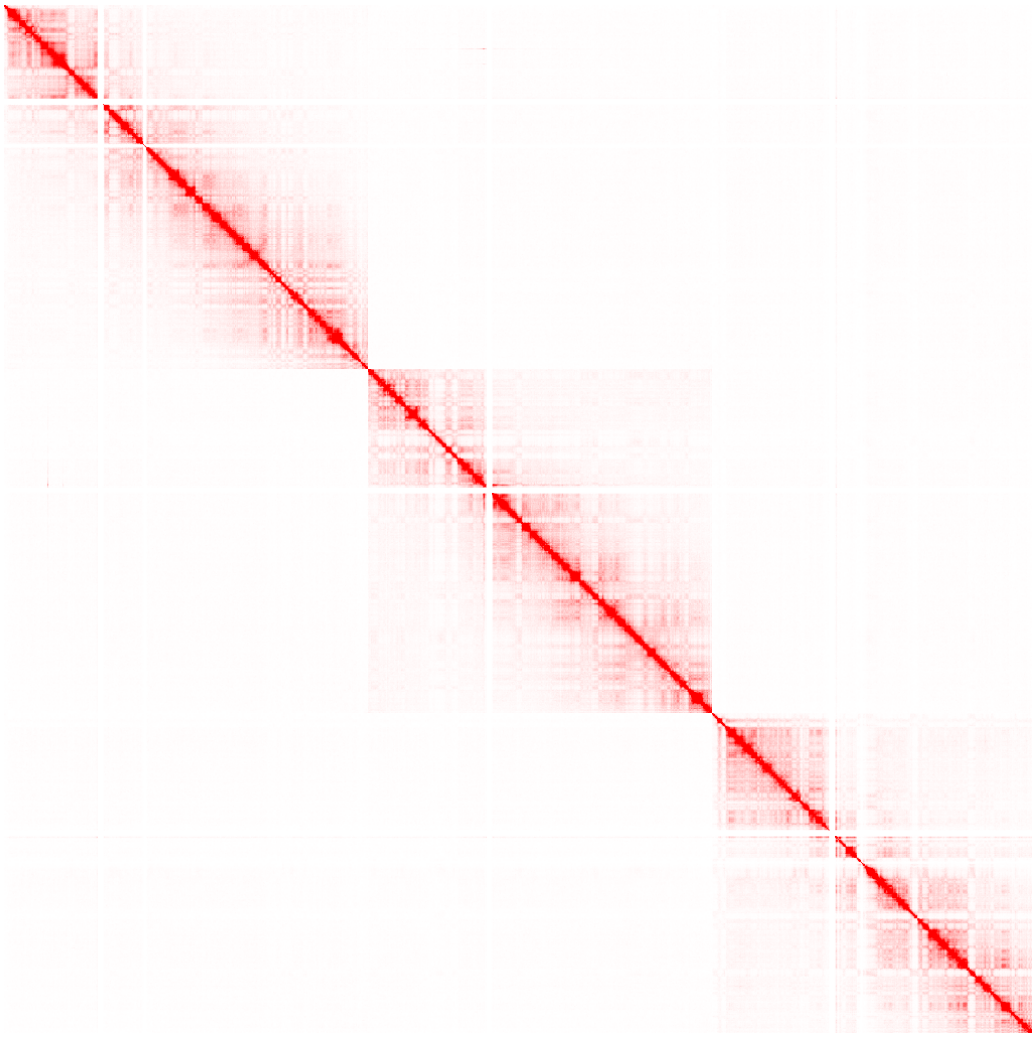
The Hi-C data can be visualized as contact maps, representing the contact frequencies between two chromosomes, or at the level of the genome. By default, objects from the *HTClist* class will be represented as an heatmap, whereas object from the *HTCexp* class (i.e. single map) as a triangle. Depending 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(mclapply(sset, binningC,
+                               binsize=500000, bin.adjust=FALSE, method="sum", step=1))
> mapC(imr90_500)
```



As we can see on this exemple, only half of the inter-chromosomal maps as 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.

```
> mapC(forcePairwise(imr90_500), maxrange=500)
```



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.

```
> ## 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:
          seqnames                ranges strand
             <Rle>             <IRanges>  <Rle>
      [1]     chr6         [    1, 10366]      +
      [2]     chr6         [10367, 18359]      +
```

```
      [3]     chr6         [18360, 19010]      +
      [4]     chr6         [19011, 24020]      +
      [5]     chr6         [24021, 25008]      +
      ...      ...                    ...    ...
  [51294]     chr6 [170887666, 170888246]      +
  [51295]     chr6 [170888247, 170891362]      +
  [51296]     chr6 [170891363, 170893508]      +
  [51297]     chr6 [170893509, 170896737]      +
  [51298]     chr6 [170896738, 170899992]      +
  -------
  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 infered 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] method.
In the following example, we will focus on chromosome 6 only. The same code can be easily applied on a *HTClist* object using the `mclapply` 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.

```
> ## 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:
          seqnames                ranges strand   |     len_U     len_D      GC_U
             <Rle>             <IRanges>  <Rle>   | <numeric> <integer> <numeric>
      [1]     chr6         [10367, 10370]      *   |     10370      7996      0.37
      [2]     chr6         [18360, 18363]      *   |      7996       654     0.385
      [3]     chr6         [19011, 19014]      *   |       654      5013       0.3
      [4]     chr6         [24021, 24024]      *   |      5013       991     0.395
      [5]     chr6         [25009, 25012]      *   |       991      4638     0.345
      ...      ...                    ...    ... ...       ...       ...       ...
  [51293]     chr6 [170887666, 170887669]      *   |       269       584     0.555
```

```
  [51294]     chr6 [170888247, 170888250]      *   |       584      3119     0.555
  [51295]     chr6 [170891363, 170891366]      *   |      3119      2149      0.35
  [51296]     chr6 [170893509, 170893512]      *   |      2149      3232     0.405
  [51297]     chr6 [170896738, 170896741]      *   |      3232      3254     0.555
              GC_D     map_U     map_D
         <numeric> <numeric> <numeric>
      [1]      0.37      <NA>      <NA>
      [2]      0.35      <NA>      <NA>
      [3]      0.26      <NA>      <NA>
      [4]      0.34      <NA>      <NA>
      [5]     0.335      <NA>      <NA>
      ...       ...       ...       ...
  [51293]     0.465      <NA>      <NA>
  [51294]     0.415      <NA>      <NA>
  [51295]      0.29      <NA>      <NA>
  [51296]      0.33      <NA>      <NA>
  [51297]     0.475      <NA>      <NA>

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

```
> ## 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:
                           seqnames                 ranges strand   |       len        GC
                              <Rle>              <IRanges>  <Rle>   | <numeric> <numeric>
              chr6:1-5e+05     chr6      [      1,  500000]      *   |    127453     0.419
         chr6:500001-1e+06     chr6      [ 500001, 1000000]      *   |    124817     0.419
     chr6:1000001-1500000     chr6      [1000001, 1500000]      *   |    103076     0.433
       chr6:1500001-2e+06     chr6      [1500001, 2000000]      *   |    127725       0.4
     chr6:2000001-2500000     chr6      [2000001, 2500000]      *   |    122886     0.394
                      ...      ...                     ...    ... ...       ...       ...
  chr6:169000001-169500000     chr6 [169000001, 169500000]      *   |    110009     0.424
    chr6:169500001-1.7e+08     chr6 [169500001, 170000000]      *   |    110434     0.411
  chr6:170000001-170500000     chr6 [170000001, 170500000]      *   |     68293     0.473
   chr6:170500001-1.71e+08     chr6 [170500001, 171000000]      *   |    114877     0.418
  chr6:171000001-171115066     chr6 [171000001, 171115066]      *   |         0       NaN
                               map
                           <numeric>
              chr6:1-5e+05       NaN
         chr6:500001-1e+06       NaN
     chr6:1000001-1500000       NaN
       chr6:1500001-2e+06       NaN
     chr6:2000001-2500000       NaN
                      ...       ...
  chr6:169000001-169500000       NaN
    chr6:169500001-1.7e+08       NaN
  chr6:170000001-170500000       NaN
   chr6:170500001-1.71e+08       NaN
  chr6:171000001-171115066       NaN
```

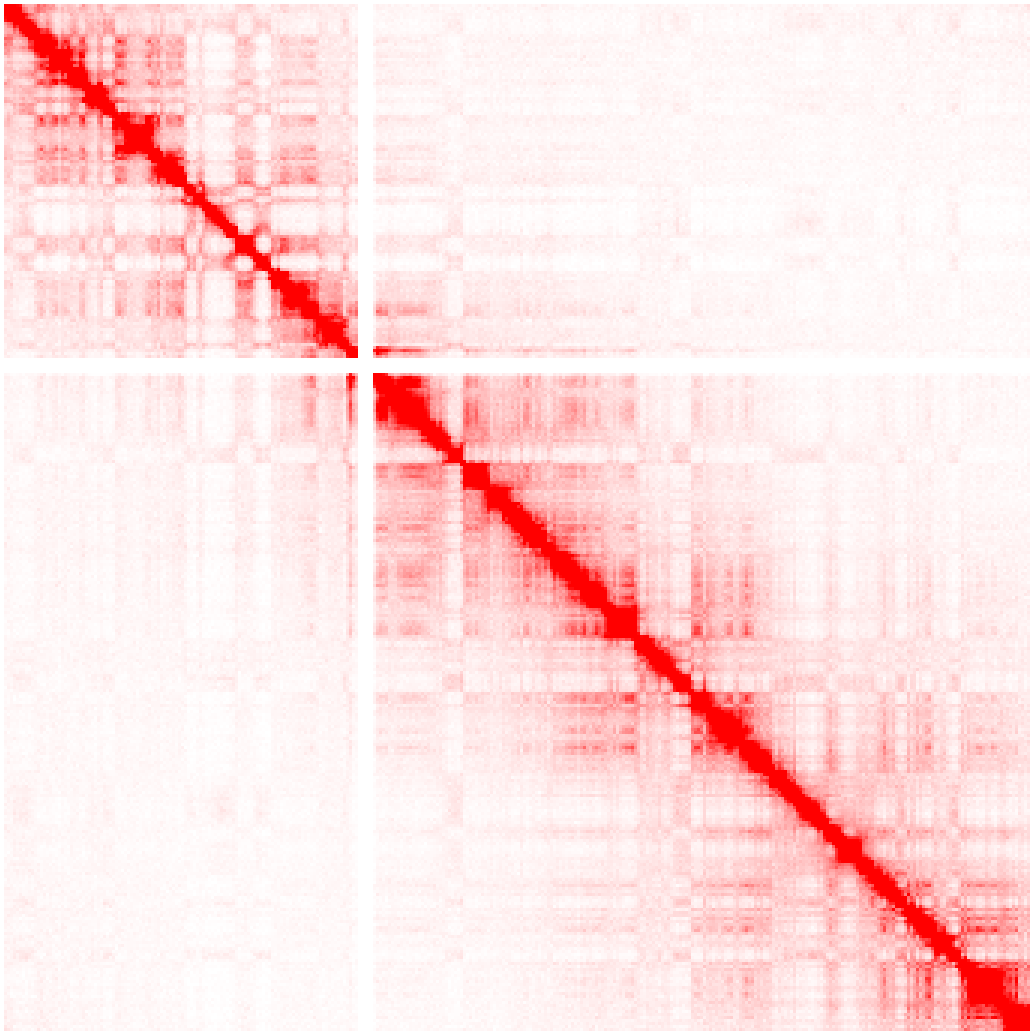
```
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
> ## 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 method 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 $\sum_{i,i\neq j,j\pm 1} M_{ij} = 1$ for each region j. Note that running the ICE normalization method can be memory consuming because it uses the full genome matrix, and then store 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.

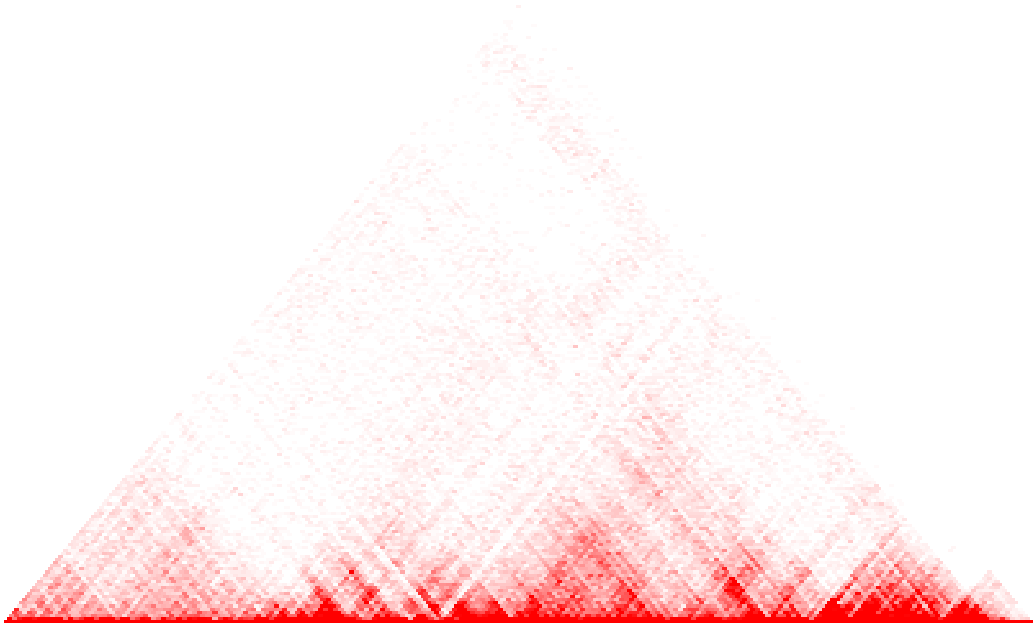
```
> imr90_500_ICE <-  normICE(imr90_500, max_iter=10)
> mapC(HTClist(imr90_500_ICE$chr6chr6), maxrange=500)
```



6 Detection of Topological Domains

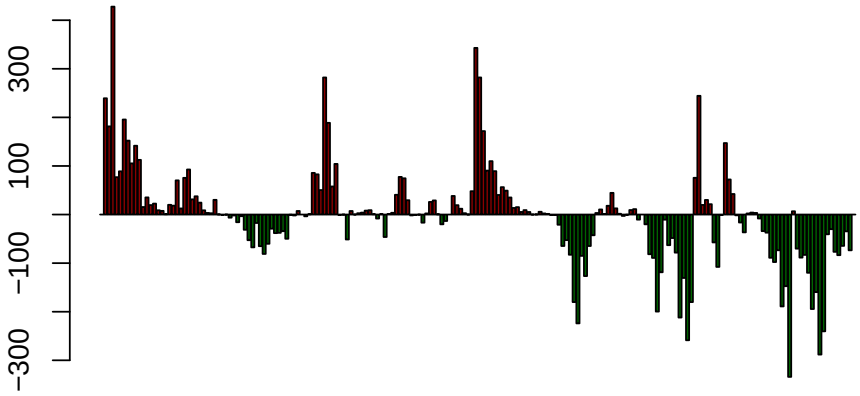
Recent studies on a high resolution human and mouse Hi-C dataset have discovered that the genome organization can be further divided into megabase-long and evolutionarily conserved topological domains (TADs), with high frequencies of intra-domain chromatin interactions but infrequent inter-domain chromatin interactions (Nora et al. [2012], Dixon et al. [2012]). More recently, **?** have demonstrated that the cell-type-specific chromatin organization seems to occur at the sub-megabase scale involving different ligation proteins and epigenomic mechanisms.
The following code shows how to focus on TADs, such as the ones describes in IMR90 around the Hox genes.

```
> hox <- extractRegion(hic_imr90_40$chr6chr6, chr="chr6", from=50e6, to=58e6)
> plot(hox, maxrange=50)
```



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.

```
> di<-directionalityIndex(hox)
> barplot(di, col=ifelse(di>0,"darkred","darkgreen"))
```



Package versions

This vignette was generated using the following package versions:

- R version 3.2.2 (2015-08-14), `x86_64-pc-linux-gnu`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: BSgenome 1.38.0, BSgenome.Hsapiens.UCSC.hg18 1.3.1000, BiocGenerics 0.16.0, Biostrings 2.38.0, GenomeInfoDb 1.6.0, GenomicRanges 1.22.0, HiCDataHumanIMR90 0.103.0, HiTC 1.14.0, IRanges 2.4.0, S4Vectors 0.8.0, XVector 0.10.0, rtracklayer 1.30.0
- Loaded via a namespace (and not attached): Biobase 2.30.0, BiocParallel 1.4.0, BiocStyle 1.8.0, GenomicAlignments 1.6.0, Matrix 1.2-2, RColorBrewer 1.1-2, RCurl 1.95-4.7, Rsamtools 1.22.0, SummarizedExperiment 1.0.0, XML 3.98-1.3, bitops 1.0-6, futile.logger 1.4.1, futile.options 1.0.0, grid 3.2.2, lambda.r 1.1.7, lattice 0.20-33, tools 3.2.2, zlibbioc 1.16.0

Acknowledgements

Many thanks to Nelle Varoquaux and Pierre Gestraud for useful discussion and help in developping this R package. Thank you to Ming Hu who developped 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.

References

J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, Apr 2012. doi: 10.1038/nature11082. URL http://dx.doi.org/10.1038/nature11082.

M. Hu, K. Deng, S. Selvaraj, Z. Qin, B. Ren, and J. S. Liu. Hicnorm: removing biases in hi-c data via poisson regression. *Bioinformatics*, 28(23):3131–3133, Dec 2012. doi: 10.1093/bioinformatics/bts570. URL http://dx.doi.org/10.1093/bioinformatics/bts570.

M. Imakaev, G. Fudenberg, R. P. McCord, N. Naumova, A. Goloborodko, B. R. Lajoie, J. Dekker, and L. A. Mirny. Iterative correction of hi-c data reveals hallmarks of chromosome organization. *Nat Methods*, 9(10):999–1003, Oct 2012. doi: 10.1038/nmeth.2148. URL http://dx.doi.org/10.1038/nmeth.2148.

E. Lieberman-Aiden, N. L. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner, R. Sandstrom, B. Bernstein, M. A. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. A. Mirny, E. S. Lander, and J. Dekker. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, Oct 2009. doi: 10.1126/science.1181369. URL http://dx.doi.org/10.1126/science.1181369.

E. P. Nora, B. R. Lajoie, E. G. Schulz, L. Giorgetti, I. Okamoto, N. Servant, T. Piolot, N. L. van Berkum, J. Meisig, J. Sedat, J. Gribnau, E. Barillot, N. Bluthgen, J. Dekker, and E. Heard. Spatial partitioning of the regulatory landscape of the x-inactivation centre. *Nature*, Apr 2012. doi: 10.1038/nature11049. URL http://dx.doi.org/10.1038/nature11049.

E. Yaffe and A. Tanay. Probabilistic modeling of hi-c contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nat Genet*, 43(11):1059–1065, Nov 2011. doi: 10.1038/ng.947. URL http://dx.doi.org/10.1038/ng.947.