r3Cseq: an R package for the discovery of long-range genomic interactions with chromosome conformation capture and next-generation sequencing data

Supat Thongjuea *

October 14, 2011

Contents

*Bergen Center for Computational Science, Bergen, Norway

1 Abstract

The advent of chromosome conformation capture (3C)-based and next-generation sequencing (NGS) technologies has led to the detection of many long-range genomic interactions via the generation of novel ligation products between DNA sequences that are closely juxtaposed in vivo. These interactions may involve promoter regions, transcription factor binding sites, and enhancers, and play key roles in the regulation of gene expression. To facilitate the identification of genomic regions that physically interact with the given viewpoints of interest, we have developed an R package called "r3Cseq". The package can be used to perform 3C-seq data analysis both in the presence or absence of a control experiment. It can read in a variety of mapped read input formats such as BAM, ELAND, and Bowtie, and it allows the visualization of candidate interaction regions as well as statistical analysis of each separate interaction region, greatly facilitating hypothesis generation and the interpretation of experimental results.

2 Introduction

The document briefly describes how to use the package *r3Cseq*. *r3Cseq* is a Bioconductor-compliant R package designed to facilitate the identification of interaction regions generated by chromosome conformation capture and next-generation sequencing (3C-seq).The fundamental principles of 3C-seq are briefly described in the following Soler et al. (2010) (Figure 1), isolated cells are crosslinked to preserve in vivo nuclear proximity between DNA sequences. Nuclei prepared from these cells are then digested using a primary restriction enzyme. HindIII (6-cutter) is used as the primary restriction endonuclease.The cut DNA fragments are then ligated under dilute conditions. This results in the specific ligation of DNA sequences in close nuclear proximity. After de-crosslinking, DNA fragments are digested again using either NlaIII or DpnII as secondary restriction endonucleases to decrease fragment size. After that they are ligated again under dilute conditions, creating small circular fragments. These fragments are then PCR amplified using primers specific for the viewpoint fragment of choice. The viewpoint is the region of interest, which can be a promoter region of a gene of interest, an enhancer, and a transcrition factor binding region. Then, those amplified fragments are sequenced using massive parallel high-throughput sequencing technology. The primer sequences will be removed from the reads and then reads will be mapped against the reference genome. A mapped read file generated by the mapping software such as ELAND and Bowtie is then analyzed by using *r3Cseq* package.

r3Cseq package has been developed to facilitate the discovery of interaction regions, which physically interact with the given viewpoints (for example a promoter region of the interested gene). The package is built on, and extends the functionality of Bioconductor package such as *IRanges*, *BSgenome*, *ShortRead*, and *rtracklayer*. The package provides classes and methods to facilitate the single-end reads, which are generated by the next-generation sequencing. The package can perform data analysis on both single input file

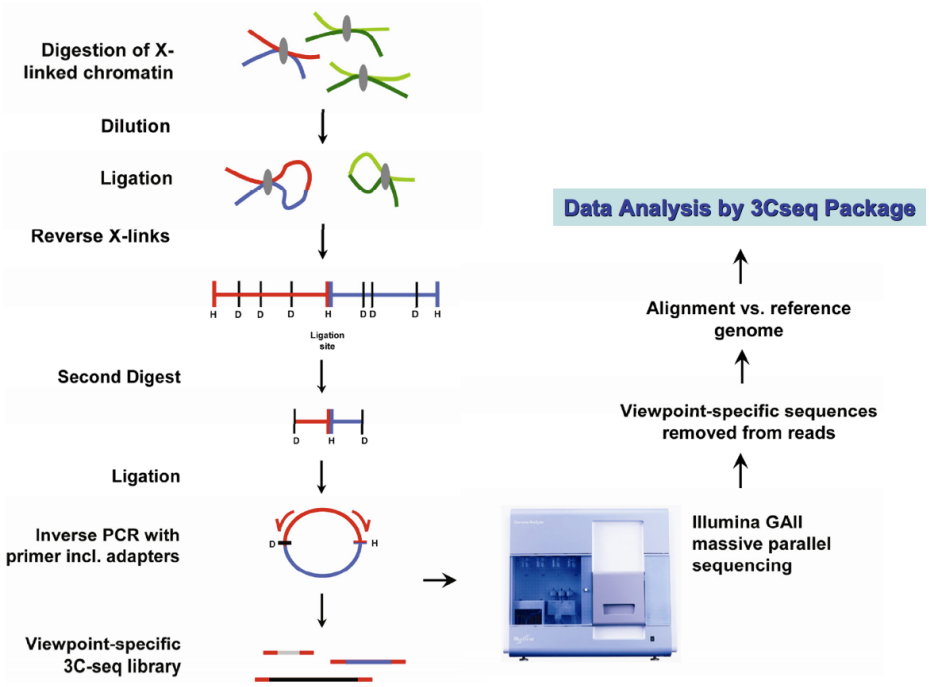

Figure 1: 3C-seq procedures

(single lane from one experiment only) and two input files from both experiment and control. The input of *r3Cseq* is the mapped read file. The package can facilitate a variety of mapped read input formats; ELAND, Bowtie, and those, which are supported by the *ShortRead* package Morgan et al. (2009). The default input format is the Binary Sequence Alignment MAP (BAM).

r3Cseq package can identify candidate interactions regions and it also provides the statistical analysis result. The result reports a number of reads, reads per million (RPM), p-value and fold change (experiment Vs control) per each restriction fragment. The p-value is calculated by using empirical cumulative distribution function. The package contains the function to export data into a UCSC track 'bedgraph' format that is simply uploading to the UCSC genome browser. The package also provides functionalities for plotting to show data analysis result of interaction regions. Plotting functions will be described in the visualization of 3C-seq section. More details about the functions, method and statistical testing are described in this Thongjuea et al. (2011). Here is a list of some of its most important functions.

1. `getCoverage`: a function to read in input formats ELAND, Bowtie, and any other formats supported by Bioconductor's ShortRead package. The default input is the BAM file.

2. `getReadCountPerRestrictionFragment` : a function to count the number of reads per restriction fragment. A user has to specify the name of restriction enzyme. The

package will then automatically generate the genome-wide restriction fragment and counts how many 3C-seq reads are mapped into that particular restriction fragment.

3. `calculateRPM` : a function to calcuate reads per million (RPM) per each restriction fragment

4. `getInteractions` : a function to estimate p-value generated by the empirical cumulative distribution function and to calculate the fold change of experiment compared with the control.

5. *Visualiztion* : the package contains functions for visualizing the interaction regions with the powerful plotting facilities. These functions are `plotOverviewInteractions`, `plotInteractionsNearViewpoint`, `plotInteractionsPerChromosome`, and `plot3Cecdf`.

6. *Exporting the data* : the package contains functions to export the data into tab-delimited text format, which can be easily uploaded to the UCSC genome browser for further visualization and exploration. Currently it supports the bedGraph format. The package can also generate a summary report in PDF format. These functions are `export3Cseq2bedGraph`, `exportInteractions2text`, and `generate3CseqReport`

3 Preparation input files for r3Cseq

The required input file for *r3Cseq* package is the output mapped read file, obtained as an output from the mapping software. The represented identifier for a reference genome shown in each input mapped file is important. In order to run *r3Cseq* properly, the represented identifier for each chromosome must be in "chr[1..19XYM]" format for the mouse reference genome and "chr[1..22XYM]" format for the human reference genome. Therefore, before using *r3Cseq* package, a user has to check the identifier for the reference genome. If the identifier for each chromosome found in the mapped file is not in the proper format for example 'mm9_ref_chr01.fa', the Unix command like 'sed' might be used to replace 'mm9_ref_chr01.fa' to 'chr1'. The example Unix command for 'sed' shows below.
> sed 's/mm9_ref_chr01.fa'/chr1' file

4 Getting started

The subset of 3C-seq data generated by Stadhouders et al. (2011) will be used for demonstration. To load the *r3Cseq* package, type `library(r3Cseq)`.

```
> library(r3Cseq)
```

There are 2 data sets found in the package.

```
> Aligned3CseqMybFetalBrain<-system.file("extdata",
+                 "alignedReads.fetal.brain.subset.bam",package="r3Cseq")
> Aligned3CseqMybFetalLiver<-system.file("extdata",
+                 "alignedReads.fetal.liver.subset.bam",package="r3Cseq")
```

1. alignedReads.fetal.liver.subset.bam, the 3C-seq data contains the aligned reads of Myb's promoter interactions regions in fetal liver.

2. alignedReads.fetal.brain.subset.bam, the 3C-seq data contains the aligned reads of the fetal brain.

In the following, we will perform *r3Cseq* to discover interaction regions, which possibly interact with the promoter region of Myb gene in both fetal liver and brain Stadhouders et al. (2011).

4.1 r3Cseq object creation

In this section, we will analyze 3C-seq data, which was generated by Stadhouders et al. (2011). 3C-seq data were derived from fetal liver and fetal brain. We would like to see the interaction regions of these two tissue types against the Myb's promoter (viewpoint). Myb gene is highly expressed in the fetal liver and it is lowly expressed in the fetal brain. We expect to see that the Myb's promoter highly interacts with the long-range genomic regions, which are mostly candidate enhancers and those drive the expression of gene in the fetal liver. In contrast, there is low level of interaction will be found in the fetal brain.

In order to see the interaction region between Myb promoter and the long-range regulatory elements, we defined the data from fetal liver as an experiment and data from fetal brain as a control with new variables called 'expFile' ane 'contrFile' respectively.

```
> expFile<-Aligned3CseqMybFetalLiver
> contrFile<-Aligned3CseqMybFetalBrain
```

Then, r3Cseq object will be created.

```
> my3Cseq.obj<-new("r3Cseq",organismName='mm9',alignedReadsBamExpFile=expFile,
+                 alignedReadsBamContrFile=contrFile,isControlInvolved=TRUE,
+                 isBamInputFile=TRUE,expLabel="fetal_liver",
+                 contrLabel="fetal_brain",restrictionEnzyme='HindIII')
```

Definition of input parameters is described in the `r3Cseq` object help page.

Type `my3Cseq.obj` to see the r3Cseq object:

```
> my3Cseq.obj
An object of class "r3Cseq"
Slot "organismName":
[1] "mm9"

Slot "restrictionEnzyme":
[1] "HindIII"

Slot "alignedReadsExpFile":
[1] ""

Slot "alignedReadsContrFile":
[1] ""

Slot "alignedReadsBamExpFile":
[1] "/tmp/RtmpvCmJR2/Rinst17c24d0d/r3Cseq/extdata/alignedReads.fetal.liver.subset.bam"

Slot "alignedReadsBamContrFile":
[1] "/tmp/RtmpvCmJR2/Rinst17c24d0d/r3Cseq/extdata/alignedReads.fetal.brain.subset.bam"

Slot "alignedReadsType":
[1] ""

Slot "expLabel":
[1] "fetal_liver"

Slot "contrLabel":
[1] "fetal_brain"

Slot "expLibrarySize":
integer(0)

Slot "contrLibrarySize":
integer(0)

Slot "expReadLength":
integer(0)

Slot "contrReadLength":
integer(0)

Slot "expReadCount":
```

```
RangedData with 0 rows and 0 value columns across 0 spaces

Slot "contrReadCount":
RangedData with 0 rows and 0 value columns across 0 spaces

Slot "expRPM":
RangedData with 0 rows and 0 value columns across 0 spaces

Slot "contrRPM":
RangedData with 0 rows and 0 value columns across 0 spaces

Slot "expInteractionRegions":
RangedData with 0 rows and 0 value columns across 0 spaces

Slot "contrInteractionRegions":
RangedData with 0 rows and 0 value columns across 0 spaces

Slot "expCoverage":
SimpleRleList of length 0
list()

Slot "contrCoverage":
SimpleRleList of length 0
list()

Slot "isControlInvolved":
[1] TRUE

Slot "isBamInputFile":
[1] TRUE
```

4.2 Getting reads per restriction fragments

In order to get number of reads per restriction fragement, function `getReadCount-PerRestrictionFragment` will be performed. `getReadCountPerRestrictionFragment` counts the number of reads for each restriction fragment across the genome.

```
> getReadCountPerRestrictionFragment(my3Cseq.obj)

[1] "start reading in ......"
[1] "making coverage vector......"
[1] "making coverage is done."
[1] "start counting number of reads per each restriction fragment......."
```

```
[1] "chr6 ---> in the experiment is done!"
[1] "chr10 ---> in the experiment is done!"
[1] "chr6 --->  in the control is done!"
[1] "chr10 --->  in the control is done!"
```

4.3 Normalization

Library sizes of the experiment and the control are usually different. In order to make them comparable, normalization is required. Reads per million per restriction fragment size (RPM) can be used as normalized values, and allow for comparison of experiment versus control. *r3Cseq* package provides `calculateRPM` function to calculate RPM.

```
> calculateRPM(my3Cseq.obj)

[1] "RPM calculation is done. Use function 'expRPM' or 'contrRPM' to get the result."
```

4.4 Getting interaction regions

After normalization, the `getInteractions` function will be used to assign the p-value and calculate fold change for each candidate interaction regions.

```
> getInteractions(my3Cseq.obj)

[1] "Calculation is done. Use function 'expInteractionRegions' or 'contrInteractionRegi
```

In order to see the result of interaction regions, Two functions `expInteractionRegions` and `contrInteractionRegions` need to be used to access the r3Cseq object. To get the result of interaction regions for the experiment, `expInteractionRegions` will be performed.

```
> fetal.liver.interactions <-expInteractionRegions(my3Cseq.obj)
> fetal.liver.interactions

RangedData with 2487 rows and 6 value columns across 2 spaces
        space                 ranges   |  expReads
     <factor>              <IRanges>   | <integer>
1        chr6     [3141513, 3143858]   |        12
2        chr6     [3327633, 3332102]   |         1
3        chr6     [3442909, 3445168]   |         5
4        chr6     [3669008, 3669538]   |         1
5        chr6     [3955826, 3957504]   |         1
6        chr6     [4179083, 4181175]   |         4
7        chr6     [4297700, 4300644]   |         2
8        chr6     [4750814, 4755215]   |         1
```

```
9        chr6     [5192227, 5193380]   |         2
...       ...                    ... ...       ...
2479    chr10 [123824449, 123832823]   |         3
2480    chr10 [124700617, 124702503]   |         1
2481    chr10 [125343151, 125343159]   |        37
2482    chr10 [125343164, 125346126]   |        37
2483    chr10 [126739522, 126743080]   |         1
2484    chr10 [127043274, 127046484]   |         8
2485    chr10 [127773509, 127777388]   |         2
2486    chr10 [129705225, 129705597]   |         5
2487    chr10 [129705602, 129708091]   |         5
     contrReads   expRPMs contrRPMs    p_value fold_change
      <integer> <numeric> <numeric>  <numeric>   <numeric>
1             0         5         0 0.09006836           5
2             0         0         0 0.74386811           0
3             0         2         0 0.25412143           2
4             0         0         0 0.74386811           0
5             0         0         0 0.74386811           0
6             0         2         0 0.25412143           2
7             0         1         0 0.46481705           1
8             0         0         0 0.74386811           0
9             0         1         0 0.46481705           1
...         ...       ...       ...         ...         ...
2479          0         1         0 0.46481705           1
2480          0         0         0 0.74386811           0
2481          0        14         0 0.02131082          14
2482          0        14         0 0.02131082          14
2483          0         0         0 0.74386811           0
2484          0         3         0 0.16083635           3
2485          0         1         0 0.46481705           1
2486          0         2         0 0.25412143           2
2487          0         2         0 0.25412143           2
```

And, to get the result of interaction regions for the control, `contrInteractionRegions` will be performed.

```
> fetal.brain.interactions <-        contrInteractionRegions(my3Cseq.obj)
> fetal.brain.interactions

RangedData with 1333 rows and 6 value columns across 2 spaces
        space                 ranges   |  expReads
     <factor>              <IRanges>   | <integer>
1        chr6     [4440236, 4443294]   |         0
```

```
2        chr6     [4512587, 4517292]   |         0
3        chr6     [4573839, 4575813]   |         0
4        chr6     [5029613, 5039856]   |         0
5        chr6     [5049808, 5052569]   |         0
6        chr6     [5337449, 5343124]   |         0
7        chr6     [5643429, 5643685]   |         0
8        chr6     [6529970, 6532698]   |         0
9        chr6     [7034518, 7035053]   |         0
...       ...                     ... ...       ...
1325    chr10 [127977794, 127983400]   |         0
1326    chr10 [128152388, 128159191]   |         0
1327    chr10 [128303580, 128309665]   |         0
1328    chr10 [128349688, 128352614]   |         0
1329    chr10 [128878917, 128880682]   |         0
1330    chr10 [129513654, 129513696]   |         0
1331    chr10 [129513701, 129516520]   |         0
1332    chr10 [129688877, 129689828]   |         0
1333    chr10 [129750979, 129752147]   |         0
     contrReads   expRPMs contrRPMs    p_value fold_change
      <integer> <numeric> <numeric>  <numeric>   <numeric>
1             1         0         0 0.84246062           0
2            10         0         3 0.29557389           3
3            46         0        14 0.03825956          14
4             2         0         1 0.52513128           1
5            11         0         3 0.29557389           3
6            18         0         5 0.17254314           5
7            10         0         3 0.29557389           3
8             3         0         1 0.52513128           1
9            12         0         4 0.20705176           4
...         ...       ...       ...        ...         ...
1325         14         0         4  0.2070518           4
1326          4         0         1  0.5251313           1
1327          5         0         1  0.5251313           1
1328         11         0         3  0.2955739           3
1329          4         0         1  0.5251313           1
1330          2         0         1  0.5251313           1
1331          1         0         0  0.8424606           0
1332          1         0         0  0.8424606           0
1333          4         0         1  0.5251313           1
```

4.5 Getting the viewpoint information

In order to see the viewpoint information, in this case study the viewpoint is the promoter region of Myb's promoter, `getViewpoint` function can be used. `getViewpoint` will return the RangedData object of the viewpoint information.

```
> viewpoint<-getViewpoint(my3Cseq.obj)
> viewpoint

RangedData with 1 row and 2 value columns across 2 spaces
     space               ranges |  expReads contrReads
  <factor>            <IRanges> | <integer>  <integer>
1    chr10 [20879877, 20882210] |      8130      14928
```

5 Visualization of 3C-seq data

r3Cseq package provides visualization functions. Those functions are `plotOverviewInteractions`, `plotInteractionsNearViewpoint`, `plotInteractionsPerChromosome`, and `plot3Cecdf`.

5.1 The overview plot of interactions

`plotOverviewInteractions` shows the overview of interaction regions distributed across genome.

```
> plotOverviewInteractions(my3Cseq.obj)
```

5.2 Empirical cumulative distribution plot

`plot3Cecdf` shows the empirical cumulative distribution of interaction regions.

```
> plot3Cecdf(my3Cseq.obj)
```

5.3 Zoom in interactions near the viewpoint

`plotInteractionsNearViewpoint` shows the zoom in of interaction regions located close to the viewpoint.

```
> plotInteractionsNearViewpoint(my3Cseq.obj)
```

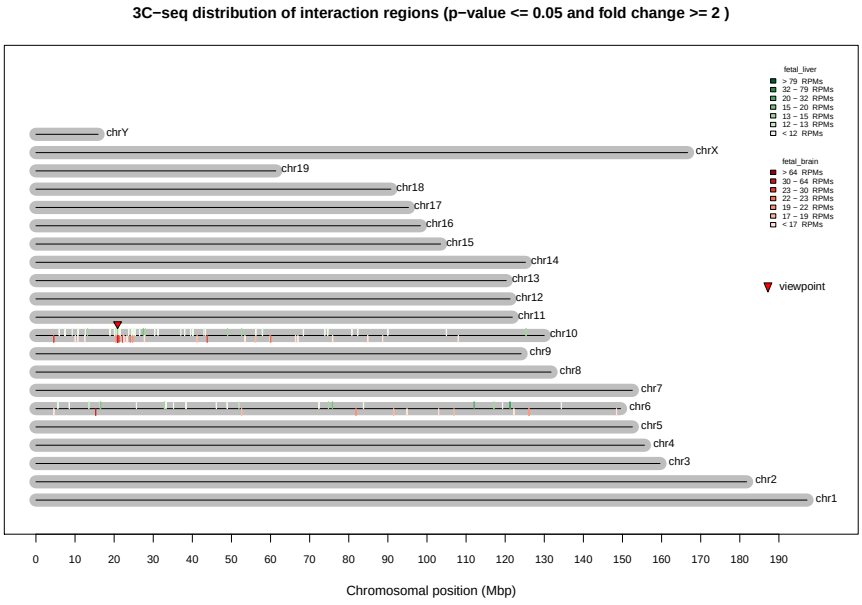

Figure 2: Distribution of interaction regions across genome

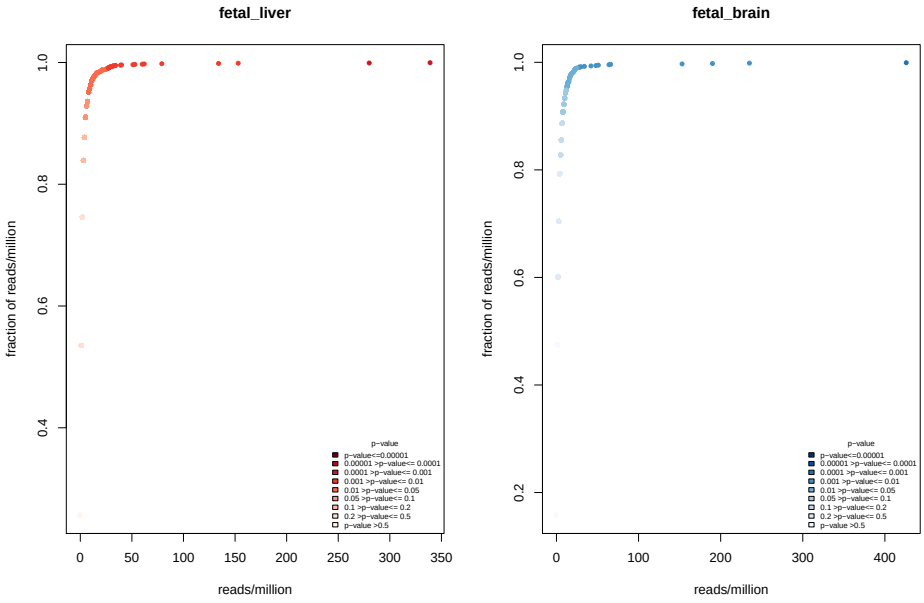

Figure 3: Empirical cumulative distribution of interaction regions

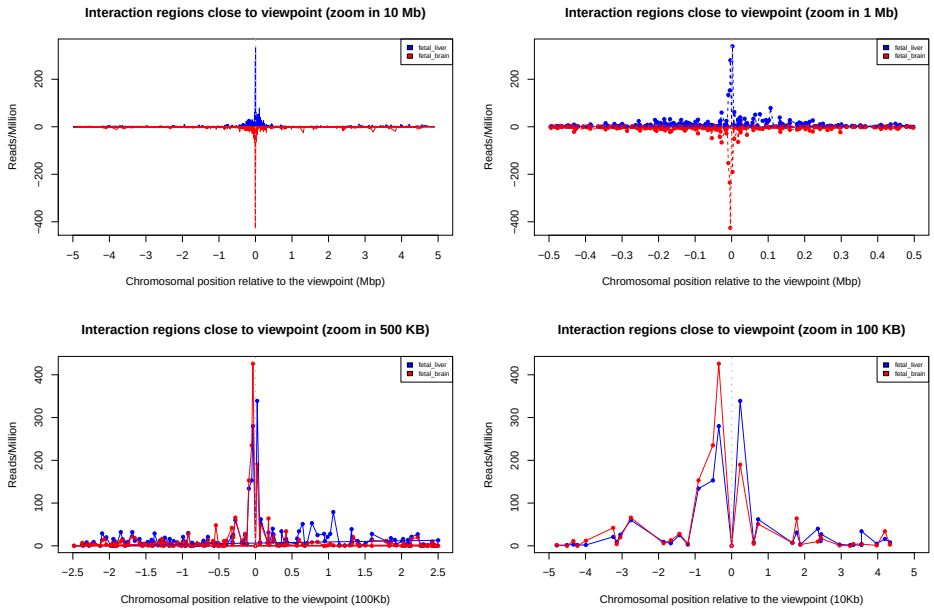

Figure 4: Zoom in interaction regions near the viewpoint

5.4 Visualize interactions in each selected Chromosome

`plotInteractionsPerChromosome` shows the interaction regions found in the chromosome10.

```
> plotInteractionsPerChromosome(my3Cseq.obj,"chr10")
```

5.5 Export interaction regions to the 'bedGraph' format

`export3Cseq2bedGraph` export interaction regions from RagedData to the bedGraph format, which suitable for uploading to the UCSC genome browser.

```
> export3Cseq2bedGraph(my3Cseq.obj)
```

```
[1] "File fetal_liver.bedGraph ' is created."
[1] "File fetal_brain.bedGraph ' is created."
```

5.6 Summary report

`generate3CseqReport` generates the summary report from *r3Cseq* analysis results. The report contains the pdf file for all plots and the text file of interaction regions.

```
> generate3CseqReport(my3Cseq.obj)
```

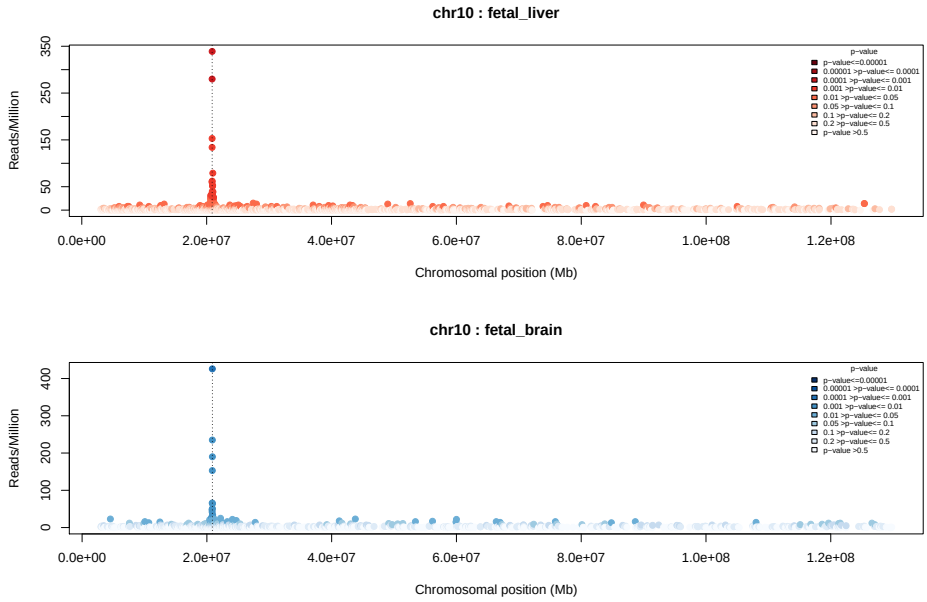

Figure 5: Distribution of interaction regions across chromosome 10

```
[1] "File fetal_liver.txt ' is created."
[1] "File fetal_brain.txt ' is created."
[1] "File fetal_liver.bedGraph ' is created."
[1] "File fetal_brain.bedGraph ' is created."
[1] "Three files are generated : a pdf file of plots, a text file of interaction region
```

6 Session Info

```
> sessionInfo()

R version 2.14.0 (2011-10-31)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=C                 LC_NUMERIC=C
 [3] LC_TIME=C                  LC_COLLATE=C
 [5] LC_MONETARY=C              LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=C                 LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
```

```
[1] stats     graphics  grDevices utils     datasets
[6] methods   base

other attached packages:
 [1] BSgenome.Mmusculus.UCSC.mm9_1.3.17
 [2] r3Cseq_1.0.0
 [3] rtracklayer_1.14.0
 [4] RCurl_1.6-10
 [5] bitops_1.0-4.1
 [6] ShortRead_1.11.45
 [7] latticeExtra_0.6-19
 [8] RColorBrewer_1.0-5
 [9] Rsamtools_1.6.0
[10] lattice_0.20-0
[11] BSgenome_1.22.0
[12] Biostrings_2.22.0
[13] GenomicRanges_1.6.0
[14] IRanges_1.12.0

loaded via a namespace (and not attached):
[1] Biobase_2.14.0 XML_3.4-3      grid_2.14.0
[4] hwriter_1.3    tools_2.14.0   zlibbioc_1.0.0
```

References

Morgan, M., Anders, S., Lawrence, M., Aboyoun, P., PagÃĺs, H., and Gentleman, R. (2009). ShortRead: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics*, 25(19):2607–2608.

Soler, E., Andrieu-Soler, C., de Boer, E., Bryne, J. C., Thongjuea, S., Stadhouders, R., Palstra, R.-J., Stevens, M., Kockx, C., van IJcken, W., Hou, J., Steinhoff, C., Rijkers, E., Lenhard, B., and Grosveld, F. (2010). The genome-wide dynamics of the binding of Ldb1 complexes during erythroid differentiation. *Genes & Development*, 24(3):277–289.

Stadhouders, R., Thongjuea, S., Andrieu-Soler, C., Palstra, R., Bryne, J., De Boer, E., Kockx, C., van der Sloot, A., van den Hout, M., van IJcken, W., Lenhard, B., Grosveld, F., and Soler, E. (2011). Dynamic long-range chromatin interactions control Myb proto-oncogene transcription during erythriod development. *EMBO*, In revision.

Thongjuea, S., Stadhouders, R., Grosveld, F., Soler, E., and Lenhard, B. (2011). r3Cseq- an R package for the discovery of long-range genomic interactions with chromosome conformation capture and next-generation sequencing data. In preparation.