DROMPA Manual
version 3.2.0

January 19, 2016

Contents

1 Overview **2**

2 Getting started **3**

3 Installation **3**
- 3.1 Requirements . . . 3
- 3.2 Install DROMPA . . . 4
- 3.3 Required data: genome table file . . . 4

4 Data format **4**

5 parse2wig: generate bin data **5**
- 5.1 Main options . . . 5
  - 5.1.1 Example . . . 5
  - 5.1.2 Paired-end file . . . 6
  - 5.1.3 Multiple mapped reads . . . 6
  - 5.1.4 Higher resolution with central regions of fragments . . . 6
- 5.2 Statistics file (for quality check) . . . 6
  - 5.2.1 sample.100.xls . . . 6
  - 5.2.2 sample.binarray_dist.xls . . . 7
  - 5.2.3 sample.readlength_dist.xls . . . 7
- 5.3 Filtering reads . . . 7
- 5.4 Total read normalization . . . 7
- 5.5 Mappability . . . 8
- 5.6 GC content . . . 8
  - 5.6.1 GC distribution file . . . 8
  - 5.6.2 Ignore peak regions . . . 9
- 5.7 Cross-correlation analysis . . . 10

6 drompa_peakcall: peak-calling **10**
- 6.1 Modes . . . 10
- 6.2 Examples . . . 10
- 6.3 Significance test . . . 11
- 6.4 Broad mode . . . 11
- 6.5 NCIS Normalization . . . 11
- 6.6 Output bin data of enrichment . . . 12
- 6.7 Annotation . . . 12

7 drompa_draw: visualization **12**
7.1 Output of parameter used . . . 12
7.2 Read distribution visualization (**PC_SHARP**) . . . 13
7.2.1 Specify different parameter for each sample pair . . . 13
7.3 Read distribution visualization (**PC_BROAD**) . . . 15
7.4 Enrichment visualization (**PC_ENRICH**) . . . 15
7.5 Annotation data for **drompa_draw** . . . 16
7.5.1 Gene annotation data . . . 17
7.5.2 Replication origin data . . . 17
7.5.3 Mappability and Gap-region data . . . 17
7.5.4 Showing limited regions . . . 17
7.5.5 Repeat data (RepBase) and GC contents . . . 18
7.5.6 BED annotation and long-range interactions . . . 18

8 drompa_draw: other functions **18**
8.1 Global view (**GV**) . . . 18
8.2 Peak density (**PD**) . . . 19
8.3 Genome Overlook (**GOVERLOOK**) . . . 19
8.4 Accumulate read counts (**FRIP**) . . . 19
8.5 Compare peak intensity (**CI**) . . . 21
8.6 Count reads in each gene (**CG**) . . . 21
8.7 Travelling ratio (**TR**) . . . 21
8.8 Aggregation plot (**PROFILE**, require R) . . . 22
8.9 Heatmap (**HEATMAP**) . . . 23

9 Appendix **24**
9.1 Make mappability and gap files . . . 24
9.2 Gene-density files . . . 24

1 Overview

DROMPA (DRaw and Observe Multiple enrichment Profiles and Annotation) is a program for user-friendly and flexible ChIP-seq pipelining [1]. DROMPA can be used for quality check, PCR-bias filtering, normalization, peak calling, visualization and other multiple analyses of ChIP-seq data. DROMPA is specially designed so that it is easy to handle, and for users without a strong bioinformatics background.

The main features of DROMPA are:

- Any species whose genomic sequence is available can be used;
- Multiple input/output file formats (SAM, BAM, Bowtie, WIG, BED, TagAlign(.gz), bigWig, bedGraph) are available;
- Normalization using mappability and GC content biases that arise in ChIP-seq data;
- Output (in PDF or PNG format) of the read distribution, ChIP/input enrichment and p-values;
- In addtion to typical peak calling, various types of ChIP-seq analysis are available.

Figure 1 shows the workflow of DROMPA. From version 3.0.0 onwards, DROMPA has three internal programs: **parse2wig**, **drompa_peakcall**, and **drompa_draw**. **parse2wig** preprocesses an input mapfile into bin data (the number of mapped read per bin with fixed length). Generated bin data are used as input for both **drompa_peakcall** and **drompa_draw**. **drompa_peakcall** calls peaks (peak calling) and **drompa_draw** executes various types of visualization and quantitative analyses.

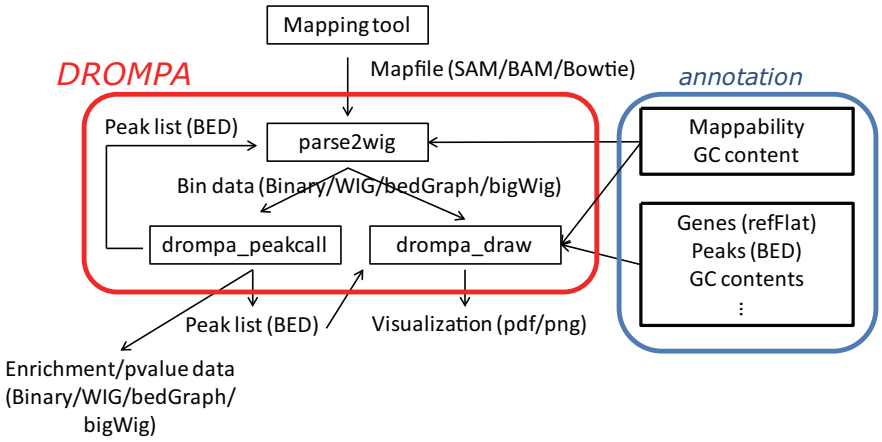

Figure 1: Workflow of DROMPA pipeline.

2 Getting started

This manual describes the usage of DROMPA using many examples. The examples are presented here as command lines to be used in UNIX shell prompts (e.g., bash or csh). Commands are prefixed with '$'. Comments are prefixed with '#' and can be ignored.

The options and default values for the thresholds will change depending on the program version. The examples provided herein uses DROMPA version 3.0.0. The latest versions and options are available at the DROMPA website[1].

3 Installation

3.1 Requirements

DROMPA is written in ANSI C and can be executed on Linux OS. DROMPA requires the following programs and libraries:

- GCC compiler (http://gcc.gnu.org/)
- Cairo libraries (http://www.cairographics.org/)
- GTK library (http://www.gtk.org/)
- GNU Scientific Library (http://www.gnu.org/software/gsl/)
- Coherent PDF (http://community.coherentpdf.com/)
  (changed from pdftk because pdftk is no longer supported on several OS)
- (optional) SAMtools (http://samtools.sourceforge.net/)
- (optional) R (http://www.r-project.org/)

SAMtools is required for BAM-formatted input files and R is required in the PROFILE command (see below). The listed programs and libraries are all freely available. On Ubuntu, these programs can be installed using the following apt-get commands:

[1] http://www.iam.u-tokyo.ac.jp/chromosomeinformatics/rnakato/drompa/

```
$ sudo apt-get install gcc
$ sudo apt-get install libgtk2.0-dev
$ sudo apt-get install libgsl-dev
$ sudo apt-get install samtools
$ sudo apt-get install r-base
```

3.2 Install DROMPA

Install through git:

```
$ git clone git@github.com:rnakato/DROMPA3.git
$ cd DROMPA3
$ make
```

To build from the source:

```
$ tar -xvzf drompa-*.*.*.tar.gz
$ cd drompa-*.*.*
$ make
```

If you get an installation error, make sure that all required libraries are installed.

Add the software directory to your PATH environment variable (consider adding this command to your Bash startup file). For example, if you downloaded DROMPA into the $HOME/my_chipseq_exp directory, type:

```
$ export PATH = $PATH:$HOME/my_chipseq_exp/drompa_*.*.*
```

Use the "-h/--help" and "--version" options to show the help message and program version, respectively.

3.3 Required data: genome table file

DROMPA requires a genome table file, a tab-delimited file describing the name and length of each chromosome. Genome-table files can be generated by **makegenometable.pl** in the *scripts* directory as follows:

```
$ scripts/makegenometable.pl genome.fa > genometable.txt
```

Chromosome names in the genome table file and the reference genome should be identical. Hereafter, "genometable.txt" indicates this genome table file.

4 Data format

- **parse2wig**:
  - Input: SAM, BAM, Bowtie and TagAlign(.gz) formats (default: SAM).
  - Output: binary (.bin), WIG (.wig), compressed WIG (wig.gz), bedGraph (.bedGraph) and bigWig (.bw) formats (default: binary).
- **drompa_peakcall** and **drompa_draw**:

- Input: binary (.bin), WIG (.wig), compressed WIG (wig.gz), bedGraph (.bedGraph) formats (default: binary)[2].
- Output: BED (peak list), PDF, PNG (figures) and tab-delimited text.

Output formats of **parse2wig**, except for binary format, can also be used in other visualization programs and browsers (e.g., IGV and UCSC genome browser). For DROMPA analysis, binary format (.bin) is recommended in order to save computation time and disk space.

5 parse2wig: generate bin data

parse2wig preprocesses an input mapfile into bin data (the number of mapped read per bin). The length of each read is calculated automatically. For single-end mode, mapped reads are extended to the expected DNA-fragment length.

5.1 Main options

```
-f SAM|BAM|BOWTIE|TAGALIGN   format of input file (default:SAM)
-of <int>           output format (default: 0)
                      0: binary (.bin)
                      1: compressed wig (.wig.gz)
                      2: uncompressed wig (.wig)
                      3: bedGraph (.bedGraph)
                      4: bigWig (.bw)
-flen <int>         Expected DNA-fragment length (default: 150 bp)
                    Automatically calculated for paired-end mode
-odir <name>        output directory name (default: 'parse2wigdir')
-binsize <int>      bin size (default: 100 bp)
-rcenter <int>      consider <int> bp around the center of the fragment
-pair               add when the input file is paired-end
-maxins <int>       maximum fragment length (default: 500 bp)
-bed <bedfile>      specify the BED file of enriched regions
```

5.1.1 Example

The command:

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt -binsize 100
```

generates bin files from sample.sam with a bin size of 100 bp. The *parse2wigdir* directory is created and the bin files are outputted into the directory. A bin file is outputted for each chromosome (in the case of binary and WIG outputs) or for the whole-genome (in the case of bedGraph and bigWig outputs).

The command:

```
$ parse2wig -f BAM -i sample.bam -o sample -gt genometable.txt -of 3
```

reads BAM file and outputs bin data in the bedGraph format.

Furthermore, multiple mapfiles can be given as one sample (separated by a ','):

```
$ parse2wig -i sample_rep1.sam,sample_rep2.sam,sample_rep3.sam \
$ -o sample -gt genometable.txt -binsize 100
```

[2]If you want to use an input file in the bigWig format, first convert the file into the bedGraph format using the "bigWigToBedGraph" command of UCSC genome browser (see http://genome.ucsc.edu/goldenPath/help/bigWig.html).

5.1.2 Paired-end file

You must supply the "-pair" option for paired-end files (when parsing paired-end mapfiles with single-end mode, warning messages are outputted). In paired-end mode, each fragment length is calculated from the mapfile automatically. Inter-chromosomal read-pairs and read-pairs longer than the maximum fragment length (specified by the "-maxins" option) are ignored.

5.1.3 Multiple mapped reads

parse2wig automatically recognizes the uniquely mapped and multiple mapped reads. For multiple mapped reads, each mapped locus is weighted equally. Thus, the total number of reads mapped into bin x is $r_x = \sum_{k \in R} 1/n_k$ where n_k is the number of times that read k is mapped onto the reference genome and R is the full set of reads mapped onto bin x.

Note: For SAM and BAM format, while **parse2wig** uses the "NH" flag to check multiple mapped reads, some mapping tools such as Bowtie and BWA do not output the "NH" column. In those cases, all reads are considered as 'uniquely mapped'. Therefore, we recommend the Bowtie format when treating multiple mapped reads.

Note: For TagAlign format, paired-end data is not supported.

5.1.4 Higher resolution with central regions of fragments

When high resolution is required (e.g., nucleosome-seq), it is better to consider only central regions of each fragment. For such purpose, **parse2wig** has the option "-rcenter". The command:

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt -flen 200 -rcenter 50
```

assumes the averaged fragment length is 200 bp and consider only 50 bp around the center of each fragment.

5.2 Statistics file (for quality check)

In addition to the bin files, **parse2wig** also outputs the statistics of the input file into the output directory, which are useful to check the quality of the sample. The commands in subsection 5.1.1 produce three types of statistics files: "sample.100.xls", "sample.binarray_dist.xls" and "sample.readlength_dist.xls".

5.2.1 sample.100.xls

The contents of "sample.100.xls" are the following:

- Input file name;
- Redundancy threshold: threshold used for PCR bias;
- Library complexity [2]
  (the number of reads tested are in parenthesis)
  (if the number of reads tested is insufficient, the score is put in parentheses);
- GC summit: the summit of GC distribution (when -GC is supplied);
- Poisson and Negative binomial: estimated parameter;

- (both whole-genome and chromosomal stats as follows);
- length: total length;
- mappable base and mappability: mappability calculated from specified mappability file;
- total reads: the number of reads in the input file;

- non-redundant reads: the number of reads remaining after PCR-bias filtering;
- redundant reads: the number of reads filtered by PCR-bias filtering
  (mapped on forward, reverse and both strands are outputted);
- reads (GCnormed): read number after GC normalization (when -GC is supplied);
- read depth;
- scaling weight: total weight of the read normalization;
- normalized read number: read number after the total read normalization;
  (this read number is used in **drompa_peakcall** and **drompa_draw**)
- FRiP (fraction of reads in peaks) score [2] (when -bed is supplied).

For the quality check, library complexity and the number of non-redundant reads are especially important.

5.2.2 sample.binarray_dist.xls

"sample.binarray_dist.xls" describes the distribution of read numbers contained in each bin. Distributions of simulated data are also shown.

5.2.3 sample.readlength_dist.xls

"sample.readlength_dist.xls" describes the read length distribution of the input mapfile. Only lengths in which the read number is nonzero are considered.

5.3 Filtering reads

parse2wig filters "redundant reads" (reads starting exactly at the same 5' ends) as "PCR bias" [1]. This filtering step can be omitted by supplying "-nofilter" option.

By default, the threshold of filtering is defined as:

$$thre_{pcr} = max(1, 10 * E_{genome})$$

where E_{genome} is the averaged read depth. This is because E_{genome} can be greater than 1 for a small genome. $thre_{pcr}$ can be supplied manually through the "-thre_pb" option.

The number of redundant/non-redundant reads and library complexity [2] can be checked using the generated statistics file (see section 5.2). Since the library complexity depends on the number of mapped reads, **parse2wig** uses the library complexity for 10 million mapped reads. This default number can be changed through the "-num4cmp" option.

5.4 Total read normalization

For the comparison of multiple ChIP samples, read number normalization is necessary. **parse2wig** has the "-n" option to normalize the bin data with the number of total mapped reads (after PCR-bias filtering).

```
-n {NONE|GR|GD|CR|CD} (default:NONE)
                      NONE; not normalize
                      GR; for whole genome, read number
                      GD; for whole genome, read depth
                      CR; for each chromosome, read number
                      CD; for each chromosome, read depth
-np <int>            read number after normalization
                     (default: 10000000 (10 million))
-nd <double>         depth after normalization (default: 0.1)
```

The users can choose total reads or read depth for normalization. For example, the command:

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt -n GR -np 20000000
```

scales bin data so that the total number of mapped reads (after filtering) onto the whole genome is 20 million. The normalization for each chromosome (CR or CD) is useful when the large difference in one chromosome affects to whole-genome (e.g., rDNA regions in chromosome XII for *Saccharomyces cerevisiae*).

Note: it is not recommended to scale a small number of reads up to a larger number because that will result in plenty of background noise (e.g., 1 million → 10 million).

5.5 Mappability

parse2wig can normalize reads based on the genome mappability [3] by supplying mappability files as follows:

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt \
$ -mp mappability/map_fragL150
```

When "-mp" is not supplied, all bases are considered as mappable. The low mappability regions ("-mpthre" option, < 0.3 (30%) as default) are ignored after ChIP-seq analysis.

DROMPA adopts the mappability files generated through the scripts provided by MOSAiCS [4]. See section 9.1 for details.

5.6 GC content

Sometimes the sequenced data has much GC bias. In those cases, GC normalization is necessary. **parse2wig** can adopt a GC normalization similar to BEADS [5]. This procedure requires the FASTA files of chromosomes and the binary mappability files. The command

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt \
$ -GC <chromosomedir> -mpbin mappability/map -flen4gc 100
```

calculates the GC contents of the input file using the central 100 bp of each fragment.

<chromosomedir> is the directory that contains the FASTA files of all chromosomes described in genometable.txt with corresponding filenames. For example, if "chr1" is in genometable.txt, there should be "chr1.fa" in <chromosomedir>. "-mpbin" specifies the binary mappability text files (see section 9.1 for details).

Note: Since this GC normalization scheme is under development, if a sample has a GC distribution quite different from other samples, it is better to consider re-preparing the sample rather than using it with GC normalization.

5.6.1 GC distribution file

parse2wig uses the longest chromosome described in genometable.txt for GC bias estimation. When using GC normalization, the GC distribution file "sample.GCdist.xls" is also outputted into the output directory. The contents are the following:

- GC: the GC content;
- genome prop: the proportion of the mappable bases containing the GC contents, then $prop_{GC}^{genome} = n_{GC}^{genome}/G$, where n_{GC}^{genome} are the number of positions containing the GC contents and G is the total number of mappable bases;

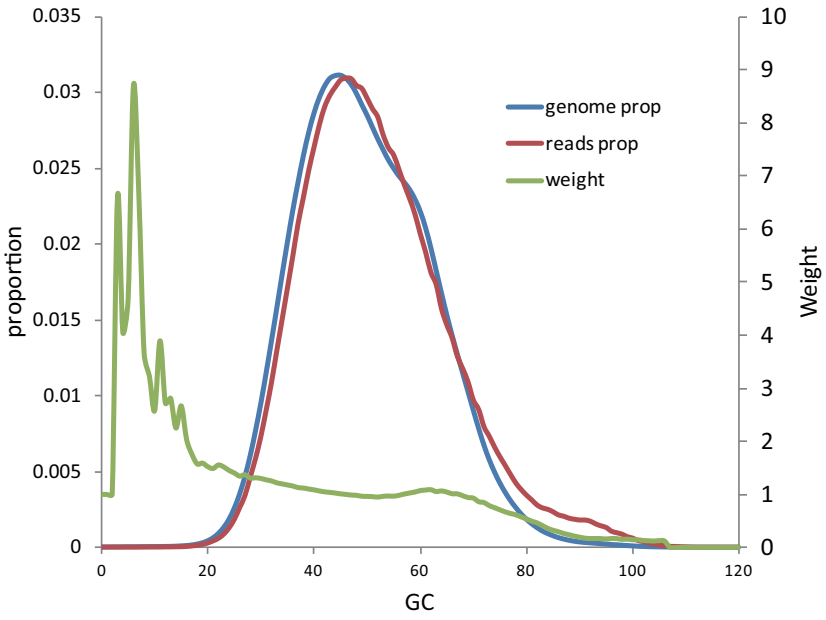

Figure 2: GC distribution generated using "sample.GCdist.xls". X-axis indicates each GC content and y-axis shows both the proportions of genome and reads and weight (genome/reads).

- read prop: the proportion of the reads (fragments) containing the GC contents, then $prop_{GC}^{reads} = n_{GC}^{reads}/N$, where n_{GC}^{reads} are the number of reads containing the GC contents and N is the total number of mapped reads;
- depth: the ratio of GC contents between reads and genome sequence, namely, $depth_{GC} = n_{GC}^{reads}/n_{GC}^{genome}$;
- weight: the ratio of the proportion between reads and genome sequence, namely, $weight = prop_{GC}^{genome}/prop_{GC}^{reads}$;

Because GC contents with low depth ($depth_{GC}$) cause background noise, by default **parse2wig** sets a weight of 1 to the GC content with $depth_{GC}$ less than 0.001, and a weight of 0 to the GC content having $prop_{GC}^{genome}$ less than 0.00001. When supplying the "-gcdepthoff" option, the former threshold is ignored.

Using the GC distribution file, the user can draw GC and weight distribution of the input file and the genome sequence. Figure 2 shows an example.

5.6.2 Ignore peak regions

For ChIP samples, it is necessary to ignore fragments that overlap with peak regions and use background reads only, because ChIP reads can have different GC distribution from the background. To do that, specify a peak list using the "-bed" option:

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt \
$ -GC <chromosomedir> -mpbin mappability/map -flen4gc 100 -bed peaklist.bed
```

5.7 Cross-correlation analysis

Optionally, **parse2wig** can output the cross-correlation profile [2] with a strategy similar to spp [6] by specifying the "-ccp" option.

The command

```
$ parse2wig -i sample.sam -o sample -gt genometable.txt -ccp
```

generates "sample.ccp.xls" in the output directory, which describes the cross-correlation plot between the read number of forward and reverse strands from -500 to 1500 bp with a 5 bp step.

In version 3.0.0, the value of bins that hace above the 95^{th} percentile is reduced to 95^{th} percentile on the cross-correlation analysis.

6 drompa_peakcall: peak-calling

6.1 Modes

drompa_peakcall has three modes as below:

```
PC_SHARP    peak-calling (for sharp mode)
PC_BROAD    peak-calling (for broad mode)
PC_ENRICH   peak-calling (enrichment ratio)
```

These are consistent to the same modes of **drompa_draw** (see section 7).

By default, chromosomes Y and M (Mt) are ignored for the analysis. Supply the "-includeYM" option to include these chromosomes.

6.2 Examples

From version 3.0.0 onwards, both ChIP and input samples are specified with a single "-i" option using a comma as follows:

```
$ drompa_peakcall PC_SHARP -i parse2wigdir/ChIP,parse2wigdir/Input \
$ -p ChIPpeak -gt genometable.txt
```

Since the bin files are chromosome-separated, use the prefix (before the underscore) to specify a sample file as an input file. The peak-list file "ChIPpeak.xls" is outputted in a tab-delimited text file that can be opened in a text editor or Microsoft Excel.

"ChIPpeak.xls" has the following columns: chromosome name, start position, end position, peak summit[3], width, peak intensity, p-values, and FDR.

The above command can be rewritten as:

```
$ IP='parse2wigdir/ChIP'
$ Input='parse2wigdir/Input'
$ drompa_peakcall PC_SHARP -i $IP,$Input -p ChIPpeak -gt genometable.txt
```

To change the parameter for peak-calling, the command:

[3] peak summit is defined as the max position of p_enrich (default), p_inter (when no Input sample) and ChIP/Input enrichment (**PC_ENRICH**).

```
$ IP='parse2wigdir/ChIP'
$ Input='parse2wigdir/Input'
$ drompa_peakcall PC_SHARP -i $IP,$Input -p ChIPpeak -gt genometable.txt \
$ -if 3 -binsize 1000 -sm 2000
```

uses the bedGraph format with a 1 kbp bin size. "-sw 2000" sets the smoothing width to 2 kbp.

When the input sample is omitted, DROMPA calls peaks using the ChIP sample only:

```
$ drompa_peakcall PC_SHARP -i parse2wigdir/ChIP -p ChIPonly -gt genometable.txt
```

Note: We strongly recommend that the ChIP sample is compared with the corresponding input data to decrease the number of false-positive sites.

6.3 Significance test

DROMPA adopts a two-step procedure for significance testing of peak-calling similar to PeakSeq [3]. In the first step, the locations that are significantly enriched compared to a background null model, assuming a negative-binomial distribution, are identified. From these candidates, the second step uses the binomial distribution to identify the significantly enriched site compared to the input control. When using the ChIP sample only, the second step is omitted. Finally, the Benjamini-Hochberg correction is also applied to calculate the false discovery rate (FDR) for multiple testing.

Summarizing the above, these are the thresholds for peak-calling:

- Main thresholds:
  - -pthre_internal: p-value of the first step (ChIP-internal enrichment)
  - -pthre_enrich: p-value of the second step (ChIP/Input enrichment)
  - -qthre: false discovery rate (FDR) (Benjamini-Hochberg correction)
- Additional thresholds:
  - -ethre: IP/Input enrichment of peak summit
  - -ipm: normalized intensity (height) of peak summit

We recommend the "-pthre_enrich" option as the main threshold for peak-calling.

6.4 Broad mode

Several histone modifications and RNA pol2 are distributed broadly. DROMPA version 3.0.0 has the **PC_BROAD** mode that uses the same strategy as **PC_SHARP** but with different default parameter settings customized for such broad peaks. **PC_BROAD** sets the following default values: "-binsize 1000 -sm 10000".

6.5 NCIS Normalization

By default, **drompa_peakcall** normalizes for the total number of reads between ChIP and input samples ("-norm 1"). Optionally, **drompa_peakcall** can normalize reads in a similar way to NCIS [7] ("-norm 2"), which is based on the number of mapped reads on background regions. When using this option the average ratio between ChIP and Input should be one.

Note: NCIS normalization might not work well when the ChIP sample has no peaks or the input sample has a ChIP-like distribution.

6.6 Output bin data of enrichment

In addition to a peak file, **drompa_peakcall** can also output bin data of the ChIP/Input enrichment and p-value distribution with the option "-outputwig". The bin data is outputted into the directory '*drompadir*'.

- -outputwig 1: output ChIP/Input ratio
- -outputwig 2: output ChIP-internal p-value
- -outputwig 3: output ChIP/Input enrichment p-value

6.7 Annotation

When the mappability file is supplied through the "-mp" option, the low mappable regions (defined through the "-mpthre" option) are ignored.

```
$ IP='parse2wigdir/ChIP'
$ Input='parse2wigdir/Input'
$ drompa_peakcall PC_SHARP -i $IP,$Input -p ChIPpeak -gt genometable.txt \
$ -mp mappability/map_fragL150 -mpthre 0.25
```

If the user does not want to output peaks in some regions (e.g., blacklist region), supply a BED file with the "-ignore" option:

```
$ IP='parse2wigdir/ChIP'
$ Input='parse2wigdir/Input'
$ drompa_peakcall PC_SHARP -i $IP,$Input -p ChIPpeak -gt genometable.txt \
$ -ignore blacklist.bed
```

7 drompa_draw: visualization

drompa_draw can visualize multiple ChIP samples with specified genome annotation, using modes for the implementation of various types of ChIP-seq analysis:

```
PC_SHARP    peak-calling (for sharp mode)
PC_BROAD    peak-calling (for broad mode)
PC_ENRICH   peak-calling (enrichment ratio)
GV          global-view visualization
PD          peak density
FRIP        accumulate read counts in bed regions specified
CI          compare peak-intensity between two samples
CG          output ChIP-reads in each gene body
GOVERLOOK   genome-wide overlook of peak positions
PROFILE     make R script of averaged read density
HEATMAP     make heatmap of multiple samples
TR          calculate the travelling ratio (pausing index) for each gene
```

7.1 Output of parameter used

When excuting **drompa_peakcall** and **drompa_draw**, the summary of parameters specified is outputted to STDOUT. The users can check whether the command is specified as expected.

7.2 Read distribution visualization (PC_SHARP)

drompa_draw can take multiple ChIP-input pairs as input. Each pair should be specified with the option "-i", as in the **drompa_peakcall** mode. For example, the command

```
$ IP1='parse2wigdir/ChIP1'
$ IP2='parse2wigdir/ChIP2'
$ IP3='parse2wigdir/ChIP3'
$ IP4='parse2wigdir/ChIP4'
$ Input='parse2wigdir/Input'
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt \
$ -i $ChIP1,$Input,ChIP1 \
$ -i $ChIP2,$Input,ChIP2 \
$ -i $ChIP3,$Input,ChIP3 \
$ -i $ChIP4,$Input,ChIP4 \
$ -gene refFlat.txt -ls 1000 -lpp 2 -show_itag 2 -scale_tag 30
```

generates the PDF files "ChIPseq*.pdf" [4][5] for four ChIP samples (ChIP1, 2, 3 and 4) and using the same Input sample (Input), as shown in Figure 3a.

By default, **drompa_draw** visualizes ChIP-read lines only. The "-show_itag 1" option displays input lines for all ChIP samples while the "-show_itag 2" option displays only the line for first input (Figure 3a). The latter is recommended when the same input sample is used for all ChIP samples.

Similarly, to display the lines of p-value and enrichment line (Figure 3b), type:

```
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt \
$ -i $ChIP1,$Input,ChIP1 \
$ -i $ChIP2,$Input,ChIP2 \
$ -i $ChIP3,$Input,ChIP3 \
$ -i $ChIP4,$Input,ChIP4 \
$ -gene refFlat.txt -showratio 1 -showpinter 1 -showpenrich 1 \
$ -scale_tag 30 -scale_ratio 3 -scale_pvalue 3
```

where the "-scale_tag", "-scale_ratio" and "-scale_pvalue" options change the maximum values for the y axis of the corresponding lines.

7.2.1 Specify different parameter for each sample pair

For **drompa_draw**, the option "-i" can take the following comma-separated multiple fields:

1. ChIP sample (required);
2. Input control sample;
3. Sample name to be shown in figure;
4. peak list to be highlighted;
5. binsize;
6. scale_tag;
7. scale_ratio;
8. scale_pvalue.

Except for the "ChIP sample", all the other fields can be omitted. When the peak list is specified, **drompa_draw** highlights the specified peak regions instead of using the internal peak-calling

[4] By default, both a whole-genome file and chromosome-separated files are generated. Supply the "-rmchr" option to omit the chromosome-separated ones.

[5] If Coherent PDF (cpdf) is not available, some command error will occur when merging pdf files, and the whole-genome file will not be generated. But other than that no problem.

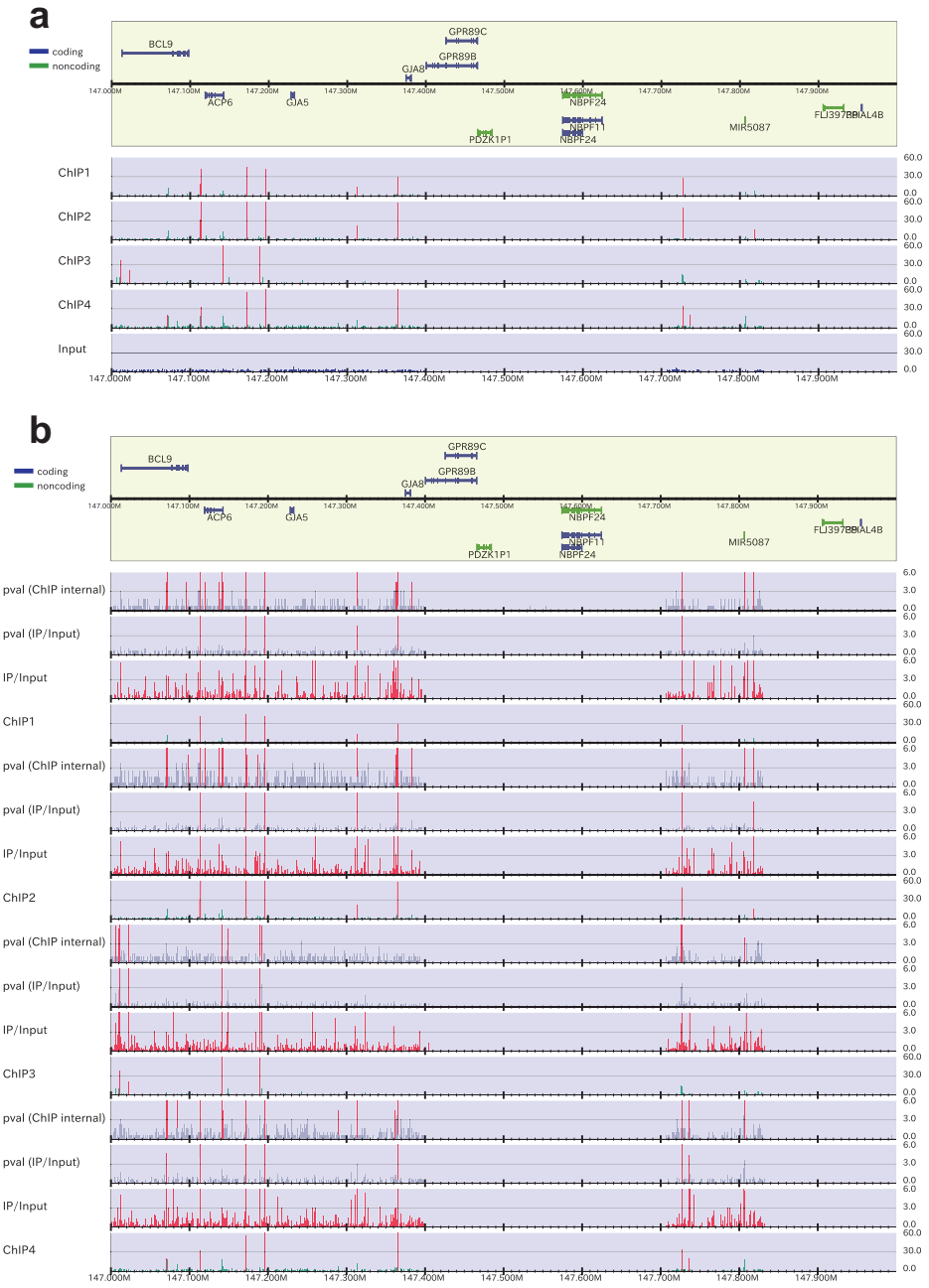

Figure 3: Examples of visualization using DROMPA. (a) Visualizing ChIP and input read lines. The green and blue histograms represent normalized ChIP- and input-read distributions, respectively. Significantly enriched regions (peaks) are shown in red. (b) Visualizing ChIP/input enrichment and -$\log_{10}$(p) values in addition to ChIP lines. The regions that are above the threshold are highlighted in red.

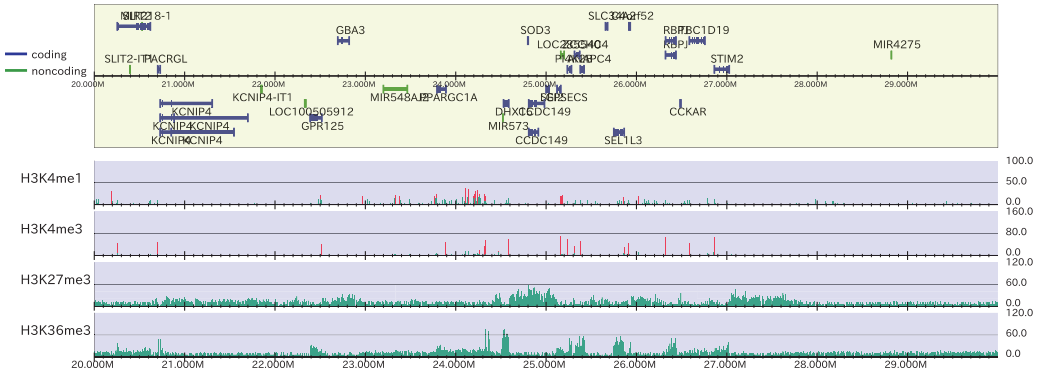

Figure 4: The example using different binsize and y-axis scale for each sample.

engine, which is useful when comparing among multiple peak-calling programs. The rest of the options can be used to specify different parameters for each sample pair. The command:

```
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt \
$ -i $ChIP1,$Input,ChIP1,ChIP1peak.bed,,50 \
$ -i $ChIP2,$Input,ChIP2,ChIP2peak.bed,,80 \
$ -i $ChIP3,$Input,ChIP3,ChIP3peak.bed,1000,60 \
$ -i $ChIP4,$Input,ChIP4,ChIP4peak.bed,1000,60 \
$ -gene refFlat.txt -lpp 1 -chr 4 -ls 10000 -rmchr -sm 2000
```

generates the results presented in Figure 4. The parameter for each sample is superior to the global parameters.

The previous command can be rewritten as:

```
$ s1="-i $ChIP1,$Input,ChIP1,ChIP1peak.bed,,50"
$ s2="-i $ChIP2,$Input,ChIP2,ChIP2peak.bed,,80"
$ s3="-i $ChIP3,$Input,ChIP3,ChIP3peak.bed,1000,60"
$ s4="-i $ChIP4,$Input,ChIP4,ChIP4peak.bed,1000,60"
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -gene refFlat.txt -lpp 1 -chr 4 -ls 10000 -rmchr -sm 2000
```

where "`$s1 $s2 $s3 $s4`" are used as ChIP-input sample pairs.

7.3 Read distribution visualization (PC_BROAD)

To identify broadly enriched region, use the **PC_BROAD** mode as follows:

```
$ drompa_draw PC_BROAD -p ChIPseq_broad -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -gene refFlat.txt -showratio 1 -showpinter 1 -showpenrich 1
```

7.4 Enrichment visualization (PC_ENRICH)

For a small genome, such as the yeast's, the sequencing depth is generally enough (> 10 fold). In such cases, the genome-wide ChIP/Input enrichment distribution is informative because the technical and biological bias in high throughput sequencing can be minimized.

To make a PDF file of the enrichment distribution for *S. cerevisiae* (Figure 5a), type:

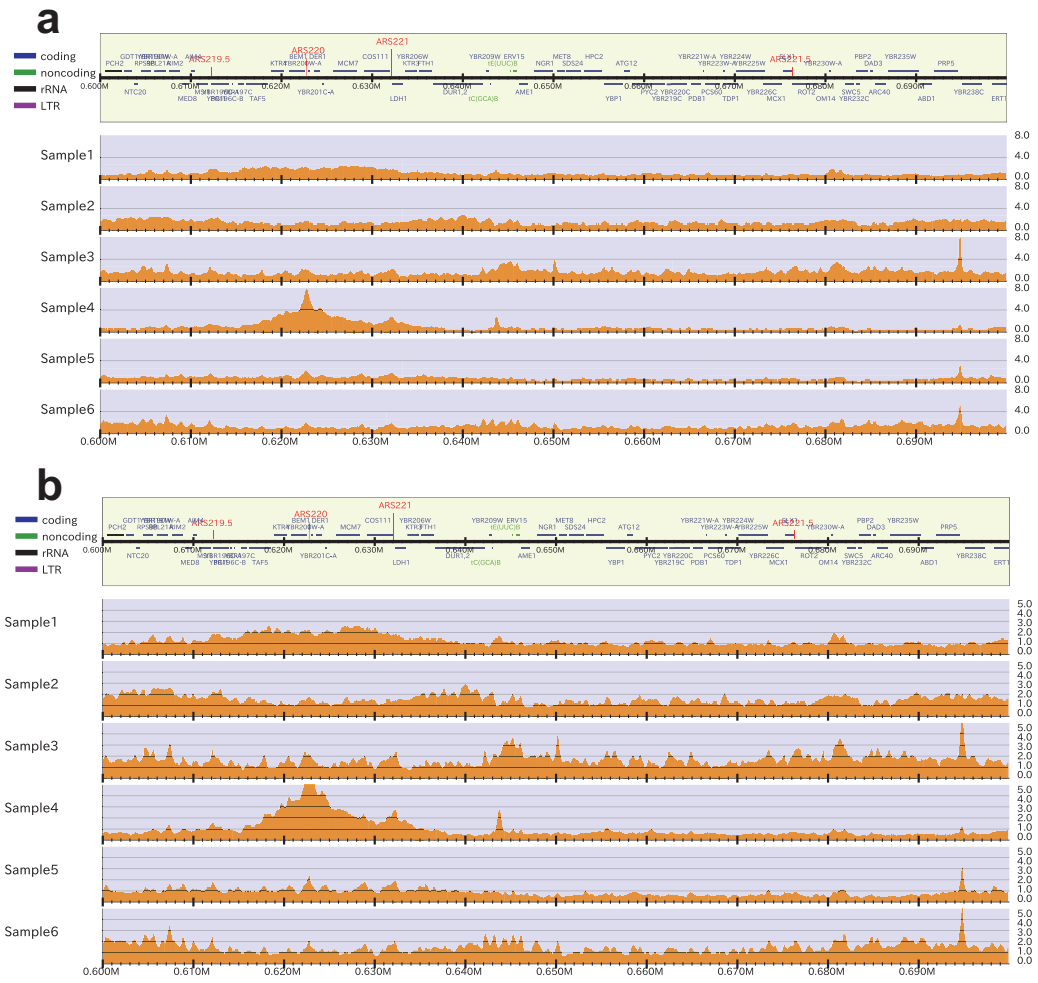

Figure 5: ChIP/Input enrichment distribution for *S. cerevisiae*.

```
$ drompa_draw PC_ENRICH -p ChIPseq_enrich $s1 $s2 $s3 $s4 $s5 $s6 \
$ -gt genometable.txt -gene SGD_features.tab -gftype 3 -ars ARS-oriDB.txt \
$ -lpp 1 -scale_ratio 4 -ls 100
```

Supply "-showratio 2" to use logratio. If you want to check the enrichment precisely, it is good to adjust the y-axis (Figure 5b) as follows:

```
$ drompa_draw PC_ENRICH -p ChIPseq_enrich $s1 $s2 $s3 $s4 $s5 $s6 \
$ -gt genometable.txt -gene SGD_features.tab -gftype 3 -ars ARS-oriDB.txt \
$ -lpp 1 -scale_ratio 4 -ls 100 -bn 5 -ystep 10
```

7.5 Annotation data for drompa_draw

DROMPA accepts annotation data from the publicly accessible websites listed below. These annotation files can also be downloaded from the DROMPA website.

7.5.1 Gene annotation data

DROMPA accepts the following gene annotation data:

- RefSeq annotation (refFlat format) obtained from the UCSC Genome Browser website [8].
- Ensembl gene data. The data for several species can be downloaded from the DROMPA website.
- The genomic annotation data for *S. cerevisiae* "SGD_features.tab" obtained from the Saccharomyces Genome Database (SGD)[6].
- For the gene annotation data of *S. pombe*, download a GFT-formatted file (e.g., "schizosaccharomyces_pombe.EF1.62.gtf") from the Ensembl website.

Supply the option "-gene" to specify gene data.

7.5.2 Replication origin data

DROMPA can visualize DNA replication origin data (ARS) available for *S. cerevisiae* and *S. pombe*. The annotation data can be obtained from OriDB [7]. Download the origin list and supply with the option "-ars".

7.5.3 Mappability and Gap-region data

If the mappability file and/or gap regions (filled with "Ns") are supplied through the "-mp" and "-gap" options, the low mappable regions and gap regions are shaded in purple and gray in the figure, respectively. See section 9.1 for details on how to generate these data.

```
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -gene refFlat.txt -mp mappability/map_fragL150 -gap mappability/N_fragL150
```

7.5.4 Showing limited regions

When the "-chr" option specified, only the specified chromosome is outputted.

```
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -gene refFlat.txt -chr 12
```

This command outputs the result of chromosome 12 only[8].

To focus on specific regions (in this example, the HOX A cluster region), supply a BED file describing the regions to be shown with the option "-r" as follows:

```
$ echo "chr7 27100000 27280000" > HOXA.txt
$ drompa_draw PC_SHARP -gene refFlat.txt $s1 $s2 $s3 $s4 -p HOXA \
$ -gt genometable.txt -r HOXA.txt -ls 300
```

[6] http://www.yeastgenome.org/

[7] http://www.oridb.org/

[8] Chromosome number is identical to the order in the genome table file. For instance, for human, chrX is generally after chr22, and supplying "-chr 23" indicates "show chrX only".

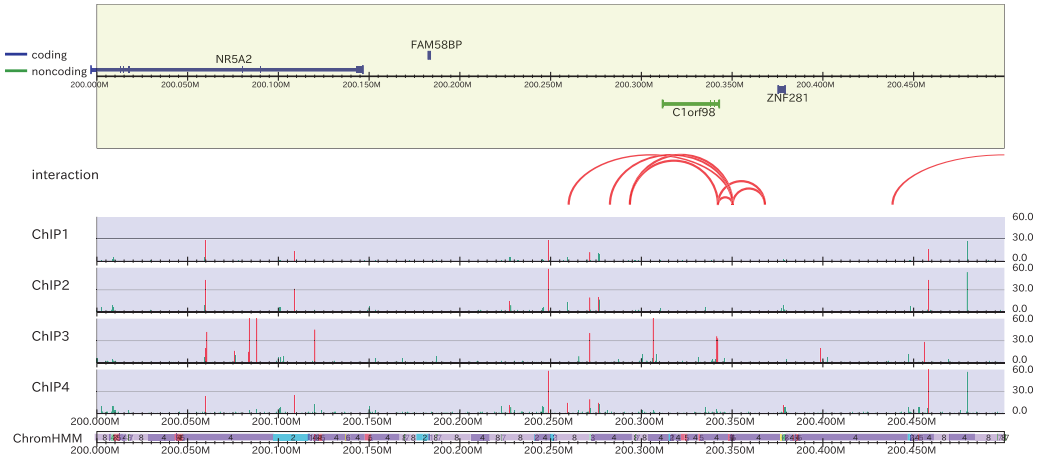

Figure 6: Visualization of ChIA-PET interactions (red arcs) and BED6 file (bottom bars). When using the BED6 format, the name of each row is also shown.

7.5.5 Repeat data (RepBase) and GC contents

DROMPA can incorporate the BED-formatted GC content files and RepBase files using the options "-repeat" and "-GC", respectively. These data can be obtained from the Table Browser of the UCSC Genome Browser [8].

```
$ drompa_draw PC_SHARP -p ChIPseq -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -repeat RepeatMasker_hg19.txt -GC GCcontents/ -gcsize 1000
```

where "-gcsize" specifies the window size of GC contents. GC content files should be chromosome-separated in the specified directory (chr*-bs*).

To supply an arbitrary window size, the DROMPA website provides the program **GCcount.pl** to generate these files from a FASTA-formatted file.

7.5.6 BED annotation and long-range interactions

drompa_draw accepts annotation data in BED or BED6 format (e.g., ChromHMM results [9]) with the "-bed" option. The long-range interactions file such as ChIA-PET results are also allowed with the "-inter" option, which takes tab-separated files with six columns: head_chr, head_start, head_end, tail_chr, tail_start, and tail_end. The intra- and inter-chromosomal interactions are shown in red and green, respectively.

For example, the following command generates the PDF file shown in Figure 6:

```
$ drompa_draw PC_SHARP -p ChIP-seq -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -gene refFlat.txt -bed chromhmm.bed,emission \
$ -inter ChIA-PET.bed,interaction
```

8 drompa_draw: other functions

8.1 Global view (GV)

The "GV" mode shows a chromosome-wide overview of the ChIP-seq data (Figure 7a). Type:

```
$ drompa_draw GV $s1 $s2 $s3 $s4 -p ChIPseq-wholegenome -gt genometable.txt \
$ -GC GCcontents/   -gcsize 500000 \
$ -GD gene_density/ -gdsize 500000
```

where "-GD" specifies the gene-density files (see section 9.2 on how to make files). When specifying the **GV** mode, the binsize is 100k-bp and the ChIP/Input enrichment lines are shown by default. The **GV** mode does not perform the significance test but simply highlights the bins containing ChIP/Input enrichments above the middle of y axis (the value specified with the option "-scale_ratio") in red.

8.2 Peak density (PD)

The **PD** mode shows the concentration of obtained peaks (Figure 7b). This mode requires **peakdensity.pl** in the *scripts* directory. To make a file that contains peak number with fixed length (in this example, 500 kbp) type:

```
$ scripts/peakdensity.pl ChIP1.xls ChIP1 500000 genometable.txt
$ scripts/peakdensity.pl ChIP2.xls ChIP2 500000 genometable.txt
```

and then type:

```
$ drompa_draw PD -p peakdensity -gt genometable.txt \
$ -pd ChIP1,ChIP1 -pd ChIP2,ChIP2  \
$ -GC GCcontents/   -gcsize 500000 \
$ -GD gene_density/ -gdsize 500000
```

"-prop" option is for displaying the proportion of peaks instead of the number for the y-axis.

8.3 Genome Overlook (GOVERLOOK)

When the genome and peak number is small, **GOVERLOOK** mode is also useful, which shows the chromosome bar and highlights the peak regions specified by the "-bed" option (Figure 8a). **GOVERLOOK** can process up to three BED files.

```
$ drompa_draw GOVERLOOK -p overlook -gt genometable.txt \
$ -bed CEN.bed.CEN -bed TER.bed,TER -bed CEN.bed,CEN2
```

8.4 Accumulate read counts (FRIP)

The **FRIP** mode outputs the accumulated read counts within the specified regions. The output file can be used to determine the "FRiP" score [2] of each sample.

```
$ drompa_draw FRIP -p FRiP -gt genometable.txt -peak peaks.xls \
$ -i $IP,,ChIPname -i $IP2,,ChIPname2 -i $IP3,,ChIPname3 -bed region.bed
```

Multiple samples can be processed simultaneously.

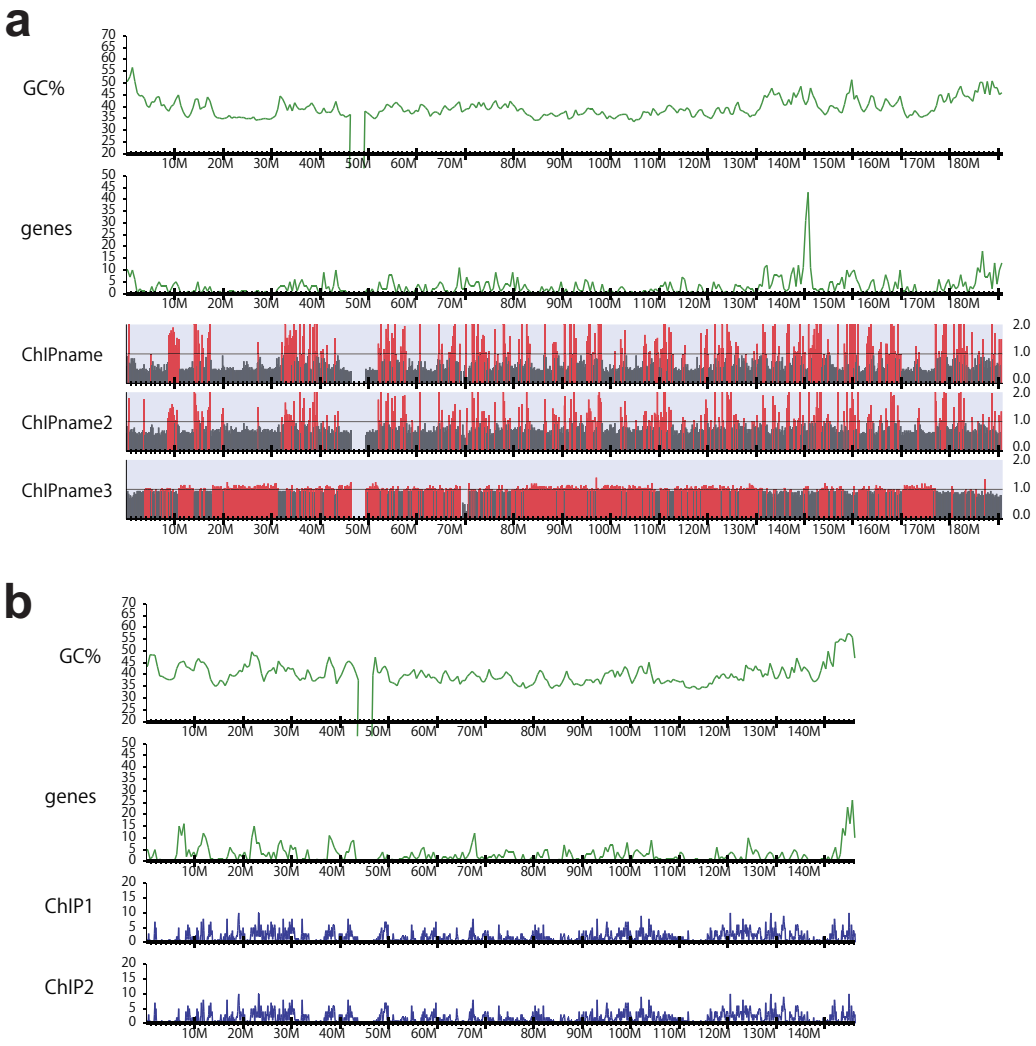

Figure 7: (a) Genome wide view (human chromosome 5) and (b) peak density (human chromosome 8).

8.5 Compare peak intensity (CI)

The **CI** mode can be used for quantitative comparison of ChIP-seq samples. This mode takes two ChIP samples to be compared, and the output file contains the accumulated read number, the average $\log_2$ read density (A), the $\log_2$ ratio of read density between two samples (M) and the significance ($-\log_{10}$(p)) of the difference based on a binomial test, for each peak specified by "-bed". This output file can be used for making a MA plot of overlapped peak regions (Figure 8b).

```
$ drompa_draw CI -p ci -gt genometable.txt -bed sample.bed \
$ -i $IP,,ChIPname -i $IP2,,ChIPname2
```

8.6 Count reads in each gene (CG)

The **CG** mode counts the read number of each gene. This mode is suitable for broad histone marks, such as H3K36me3 and H3K27me3.

```
$ drompa_draw CG -p cg -gt genometable.txt -gene refFlat.txt \
$ -i $IP,,ChIPname -i $IP2,,ChIPname2 -i $IP3,,ChIPname3
```

The output file contains the following columns:

```
name chromosome start end length ChIPname per kbp exon intron  intron/exon
PGLYRP3 chr1 153270337 153283194 12857 117.8 9.16 16.2 113 6.98
```

where the first five columns are the data for each gene. The remaining are:

- total read in the gene;
- read for 1 kbp ($117.8 * 1000/12857$);
- total read in exons;
- total read in introns;
- ratio between intron/exon ($113/16.2$).

When multiple samples are specified, multiple sets of columns are shown.

8.7 Travelling ratio (TR)

The **TR** mode calculates the travelling ratio (or pausing index) [10], the relative ratio of density in the promoter-proximal region, and the gene body. This is especially important for the RNA polII binding. The command:

```
$ drompa_draw TR -p drompaTR -gt genometable.txt $s1 $s2 \
$ -gene refFlat.txt
```

outputs two files: "drompaTR.xls" and "drompaTR.fig.xls". Similarly to the **CG** mode, each row of the "drompaTR.xls" contains the data for each gene and the accumulated read number around TSS, gene body and TSS/body (travelling ratio). "drompaTR.fig.xls" describes the accumulated proportion of genes with a given TR score. Figure 8c can be generated using this file with MS Excel, like Figure 1c of reference [10]. See the reference for further details on the travelling ratio.

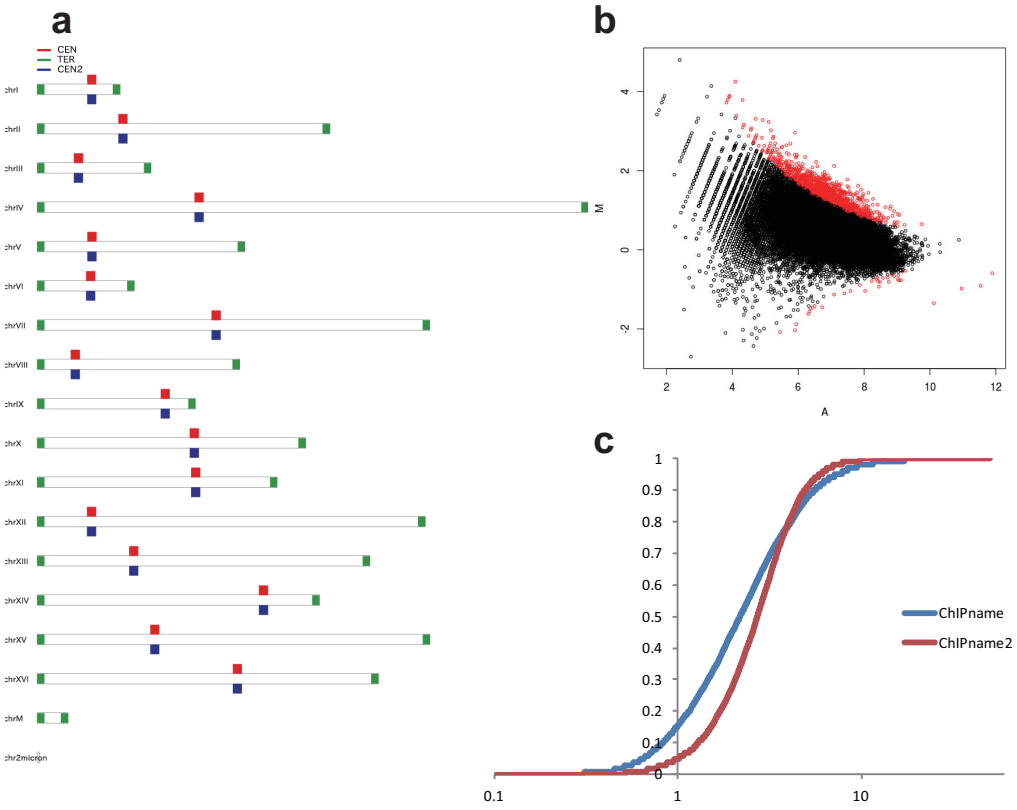

Figure 8: (a) Genome Overlook. Centromere and telomere of *S. cerevisiae* is colored. (b) MAplot between M and A columns in the output file of **CI** mode. This MA plot is generated using the *plot* function of R. The peaks in which -$\log_{10}$(p) is over 10 are colored in red. (c) Distribution of the proportion of genes with a given travelling ratio (pausing index).

8.8 Aggregation plot (PROFILE, require R)

The **PROFILE** mode makes an aggregation plot around the transcription start sites (TSS), transcription termination sites (TTS), gene bodies, and specified peak regions within the 95% confidence interval (Figure 9). The **PROFILE** mode has several important options:

- -ptype: define which types of region are plotted;
- -stype: specify if either ChIP read or ChIP/input enrichment is used;
- -ntype: normalize samples with the number of total reads mapped in target regions (instead of for whole genome);
- -cw: change the width of the plot.

The following commands output a PDF file (aroundgene.pdf) and an R script (aroundgene.R).

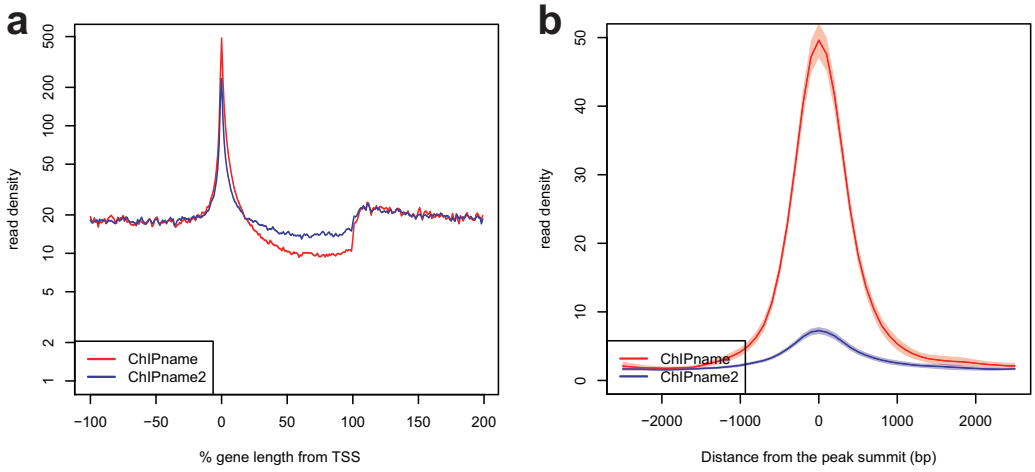

Figure 9: Aggregation plot around gene bodies (a) and peak regions (b) generated by the **PROFILE** mode. Shaded regions indicate a 95% confidence interval.

```
# around gene bodies
$ drompa_draw PROFILE -p aroundgene -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -ptype 3 -gene refFlat.txt

# around peak regions
$ drompa_draw PROFILE -p aroundpeak -gt genometable.txt $s1 $s2 $s3 $s4 \
$ -ptype 4 -bed peaklist.bed
```

If the user wants to modify the parameters of the plot, change the parameters in the generated R script and remake the PDF file as follows:

```
$ R --vanilla < aroundgene.R
$ R --vanilla < aroundpeak.R
```

Optionally, the "-pdetail" option describes the read number of each site in the output file (*.R). This can be used to check for outliers.

8.9 Heatmap (HEATMAP)

The **HEATMAP** mode output the heatmap of target sites. The main options correspond to that of **PROFILE** mode. Additionally, the **HEATMAP** mode has the options "-scale_tag", "-scale_ratio" and "-scale_pvalue" that indicates the maximum value of the scale bar.

The **HEATMAP** mode allows multiple BED files as input. For instance, to show ChIP/input enrichment around the sites in three peak lists (site*.bed), type:

```
$ drompa_draw HEATMAP -p heatmap -gt genometable.txt $s1 $s2 $s3 \
$ -stype 1 -ptype 4 -hmsort 2 -scale_ratio 5 \
$ -bed site1.bed,region1 -bed site2.bed,region2 -bed site3.bed,region3
```

where "-hmsort <int>" defines which sample is used for sorting the order of input sites. "-hmsort 2" means the **HEATMAP** mode sorts the order of sites using the second ChIP sample. When specifying "-hmsort 0" (default), the sites are not sorted.

When showing gene body regions with the "-hmsort" option, the **HEATMAP** mode sorts the read intensity of TSS regions. When supplying the "-sortgbody" option, the **HEATMAP** mode sorts sites based on the summed read number within the whole-gene region.

Note: When the number of sites is large, the output PDF file might become too heavy to load in most PDF viewers. In those case, it is worth to consider using the "-png" option, which outputs the figure in PNG format.

9 Appendix

9.1 Make mappability and gap files

DROMPA accepts the mappability files and gap files generated by scripts provided by MOSAiCS [4], which is based on the code from Peakseq [3].

Using the MOSAiCS scripts with the appropriate fragment length and binsize (here 150 bp and 100 bp, respectively), you can obtain the following files:

- binary mappability files "chr*_map_binary.txt";
- bin-level mappability files "chr*_map_fragL150_bin100.txt";
- gap files "chr*_N_binary.txt".

See the MOSAiCS website[9] for more details. The bin-level mappability files are used for the "-mp" option of **parse2wig**, **drompa_peakcall** and **drompa_draw** while the binary files are used for the "-mpbin" option in GC normalization (see section 5.6).

After generateing those files, change their names for DROMPA:

```
$ for i in $(seq 1 22) X Y M; do
$ mv chr${i}_map_binary.txt map_chr${i}_binary.txt
$ mv chr${i}_map_fragL150_bin100.txt map_fragL150_chr${i}_bin100.txt
$ mv chr${i}_N_fragL150_bin100.txt N_fragL150_chr${i}_bin100.txt
$ done
```

After this, make the "mappability table", a tab-delimited file describing the number of mappable bases for each chromosome, using **makemappabilitytable.pl** in the *scripts* directory.

```
# specify the prefix of binary mappability files for the second arguments
$ scripts/makemappabilitytable.pl genome_table.txt map > map_fragL150_genome.txt
```

The prefix of the "mappability table" should be identical to the bin-level mappability files.

Finally, DROMPA can adopt these files as follows:

```
$ -mp map_fragL150  # means map_fragL150_chr*_bin100.txt
$ -mpbin map        # means map_chr*_binary.txt
$ -gap N_fragL150   # means N_fragL150_chr*_bin100.txt
```

9.2 Gene-density files

Gene density files can be generated through the **makegenedensity.pl** in the *scripts* directory. To use a 500 kbp window, type:

```
$ scripts/makegenedensity.pl genometable.txt refFlat.txt 500000
```

[9] http://www.stat.wisc.edu/~keles/Software/mosaics

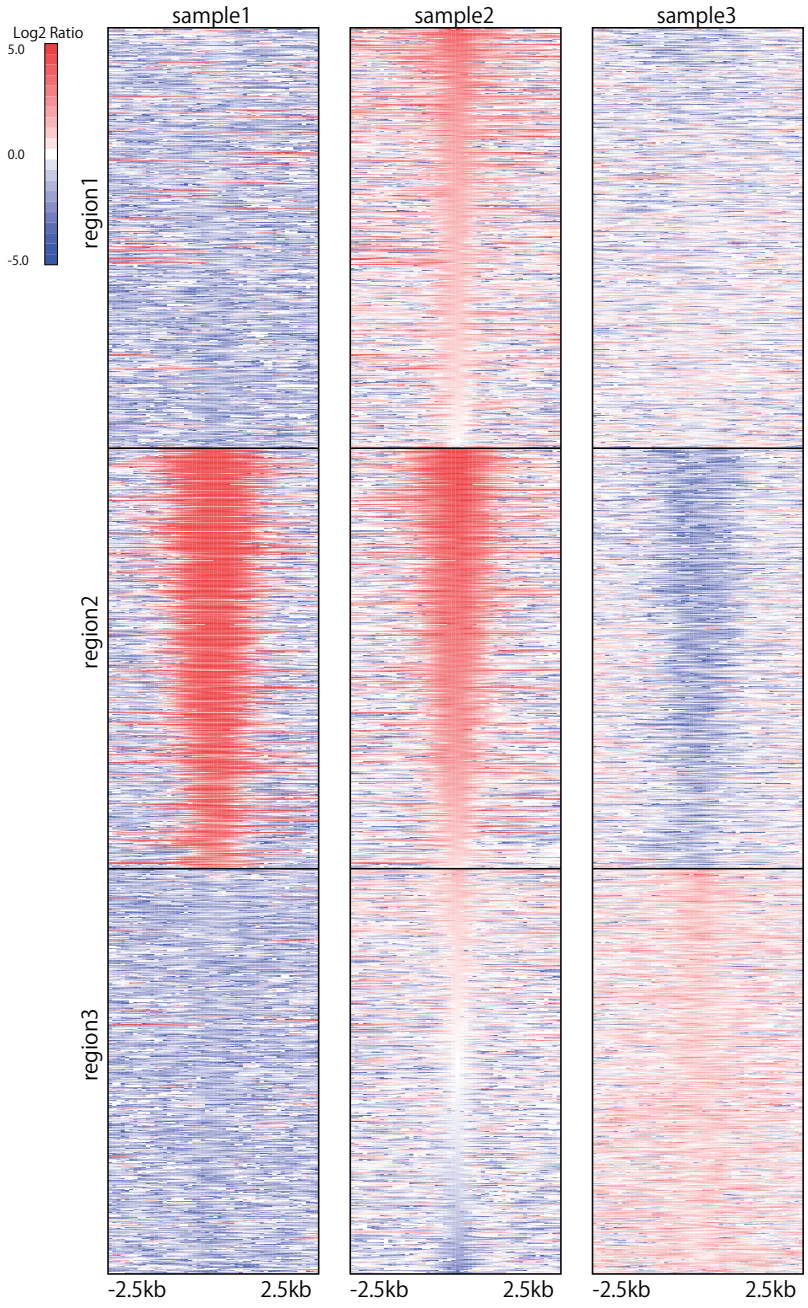

Figure 10: Heatmap for ChIP/input enrichment. Sample2 was used for sorting input sites. Each BED file is sorted individually.

and the gene-density files "chr*-bs<binsize>" in the current directory. These data for several species can also be downloaded from the DROMPA website. Next make a new directory for gene-density files:

```
$ mkdir gene_density_hg19
$ mv chr*-bs* gene_density_hg19
```

and the gene-density files can be specified as follows:

```
$ drompa_draw GV $s1 $s2 $s3 $s4 -p ChIPseq-wholegenome -gt genometable.txt \
$ -GD gene_density_hg19/ -gdsize 500000
```

References

[1] R. Nakato, T. Itoh, K. Shirahige, "DROMPA: easy-to-handle peak calling and visualization software for the computational analysis and validation of ChIP-seq data," *Genes Cells*, vol. 18, no. 7, pp. 589–601, 2013.

[2] S. G. Landt *et al*., "ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia," *Genome Research*, vol. 22, no. 9, pp. 1813–1831, Sept. 2012.

[3] J. Rozowsky *et al*., "PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls," *Nature Biotechnology*, vol. 27, no. 1, pp. 66–75, 2009.

[4] P. F. Kuan *et al*., "A statistical framework for the analysis of ChIP-seq data," *Journal of the American Statistical Association*, vol. 106, no. 495, pp. 891–903, 2011.

[5] M. S. Cheung *et al*., "Systematic bias in high-throughput sequencing data and its correction by BEADS," *Nucleic Acids Res*, vol. 39, no. 15, pp. e103, 2011.

[6] P. V. Kharchenko, M. Y. Tolstorukov, P. J. Park, "Design and analysis of ChIP-seq experiments for DNA-binding proteins," *Nat Biotechnol*, vol. 26, no. 12, pp. 1351–9, 2008.

[7] K. Liang, S. Keles, "Normalization of ChIP-seq data with control," *BMC Bioinformatics*, vol. 13, pp. 199, 2012.

[8] "UCSC Genome Browser," `http://genome.ucsc.edu/`.

[9] J. Ernst, M. Kellis, "ChromHMM: automating chromatin-state discovery and characterization," *Nat Methods*, vol. 9, no. 3, pp. 215–6, 2012.

[10] P. B. Rahl *et al*., "c-Myc regulates transcriptional pause release," *Cell*, vol. 141, no. 3, pp. 432–45, 2010.