

Using PING with Paired-End sequencing data

Xuekui Zhang*, Sangsoon Woo†, Raphael Gottardo‡ and Renan Sauteraud§

October 11, 2012

This vignette presents a workflow to use PING for analyzing paired-end sequencing data.

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | Licensing and citing | 2 |
| 2 | Introduction | 2 |
| 3 | PING analysis steps | 2 |
| 4 | Data Input and Formatting | 2 |
| 5 | PING analysis | 3 |
| 5.1 | Genome segmentation | 3 |
| 5.2 | Parameter estimation | 3 |
| 6 | Post-processing PING results | 4 |
| 7 | Analyzing the prediction | 4 |

*ubcxzhang@gmail.com

†swoo@fhcrc.org

‡rgottard@fhcrc.org

§rsautera@fhcrc.org

1 Licensing and citing

Under the Artistic License 2.0, you are free to use and redistribute this software.

If you use this package for a publication, we would ask you to cite the following:

Xuekui Zhang, Gordon Robertson, Sangsoon Woo, Brad G. Hoffman, and Raphael Gottardo. (2012). Probabilistic Inference for Nucleosome Positioning with MNase-based or Sonicated Short-read Data. PLoS ONE 7(2): e32095.

2 Introduction

For an introduction to the biological background and PING method, please refer to the other vignette: ‘The PING user guide’. Because the structure of paired-end sequencing data requires a slightly different treatment, we are separately presenting how to use PING for these data in this vignette.

3 PING analysis steps

A typical PING analysis consists of the following steps:

1. Extract reads and chromosomes from bam files.
2. Segment the genome into candidate regions that have sufficient aligned reads via ‘segmentPING’
3. Estimate nucleosome positions and other parameters with PING
4. Post-process PING predictions to correct certain predictions

As with any R package, you should first load it with the following command:

```
> library(PING)
```

4 Data Input and Formatting

As with the Single-End PING, the input used for the segmentation step is a `GRanges` object.

Because Paired-End sequencing data often comes in the form of BAM files, we provide a function called `bam2gr` to convert these files into `GRanges` objects with all the appropriate information. A small BAM file including two chromosomes of the yeast is provided to be used as an example in this vignette.

```
> yeastBam <- system.file("extdata/yeastChrI_M.bam", package = "PING")  
  
> gr <- bam2gr(bamFile = yeastBam)  
  
Chromosome chrI  
Chromosome chrM
```

`gr` is a `GRanges` object containing all the reads from the `.bam` file.

5 PING analysis

5.1 Genome segmentation

PING is used the same way for paired-end and single-end sequencing data. The function `segmentPING` will decide which segmentation method should be used based on the arguments provided. When dealing with paired-end data, four new arguments have to be passed to the function: `chr` for a chromosome, `islandDepth`, `min_cut` and `max_cut` for candidate region selection. These arguments control the size and required coverage for a region to be considered as a candidate.

Parallelisation will also work with paired-end data. In what follows, we assume that `parallel` is installed on your machine. If it is not, the first line should be omitted and calculations will occur on a single CPU.

```
> library(parallel)  
  
> segPE <- segmentPING(gr, chr = "chrM", islandDepth = 3, min_cut = 50,  
  max_cut = 1000)
```

It returns a `segReadsListPE` object.

5.2 Parameter estimation

For paired-end data in PING, the users have to set the argument of `PE` as `TRUE`.

```
> ping <- PING(segPE, PE = TRUE)
```

The returned object is a `pingList`, which will go through a post-processing step using `postPING` function.

6 Post-processing PING results

Here again, the only difference with single-end data is that the argument `PE` should be set to `TRUE`.

```
> {
  sigmaB2 = 3600
  rho2 = 15
  alpha2 = 98
  beta2 = 2e+05
}
> PS = postPING(ping, segPE, rho2 = rho2, alpha2 = alpha2, beta2 = beta2,
  sigmaB2 = sigmaB2, PE = TRUE)
```

The 5 Regions with following IDs are reprocessed for singularity problem:

| | | | | | | | | | |
|-------------|----|-----------|-----|-----------|-----|-----------|-----|-----------|-----|
| (0.783,110] | 84 | (110,218] | 37 | (110,218] | 50 | (110,218] | 76 | (110,218] | 79 |
| | 84 | | 146 | | 159 | | 185 | | 188 |

The 1 Regions with following IDs are reprocessed for atypical delta:

```
[1] 149
[1] "No predictions with atypical sigma"
```

The 147 regions with following IDs are reprocessed for Boundary problems:

```
[1] 4 17 25 37 39 46
```

The result output of `postPING` is a dataframe that contains estimated parameters of each nucleosome.

7 Analyzing the prediction

PING comes with a set of tools to export or visualize the prediction. Here, we only show how to export the results into bed format for further analysis and how to make a quick plot to summarize the nucleosome prediction. For more information on how to export the results or make more complex figures, please refer to the section ‘Result output’ of PING vignette.

The function `makeRangedDataOutput` offers a simple way to convert the prediction results into a `RangedData` object that can be exported into a file using the `rtracklayer` package.

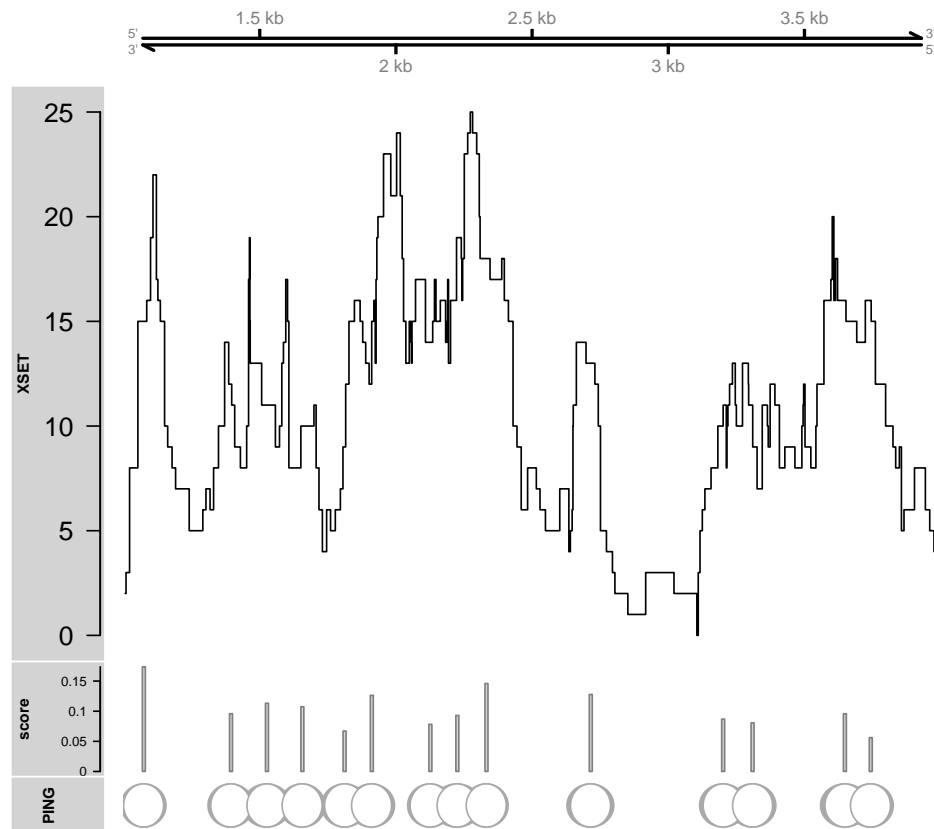
```
> rdBed <- makeRangedDataOutput(PS, type = "bed")
> library(rtracklayer)
> export(rdBed, "nucPrediction.bed")
```

The exported file includes all information about the predicted nucleosomes, which are already automatically ranked by their score.

For paired-end sequencing data, the built-in plotting function `plotSummary` can be used to visualize the predicted nucleosome positions obtained from `postPING` function.

```
> plotSummary(PS, gr, chr = "chrM", from = 1000, to = 4000, PE = TRUE)
```

chrM:1000–4000(3000bps)



Note that the argument `PE` should be set to `TRUE`. All the arguments for this function will work for Paired-end data as well. Refer to `PING` vignette and the man page `?plotSummary` for more information.