NucDe Package Example Version 1.0

Pei Fen Kuan, Xin Li, Sündüz Keleş

May 28, 2010

1 Overview

NucDe is an R package mapping nucleosome-linker boundaries from both MNase-Chip and MNase-Seq data using a non-homogeneous hidden-state model based on first order differences of experimental data along genomic coordinates (Kuan et al.; 2009). Current version is tailored for MNase-Chip with array resolution similar to Yuan et al. (2005) such that a well positioned nucleosome is represented by 6-8 probes, whereas for MNase-Seq we assume a $\sim$5 bps resolution and a well positioned nucleosome is covered by $\sim$29 probes. To load this package, type

```
> library(NucDe)
```

The *NucDe* package requires a data frame which consists of chromosome ID, start coordinate, and signal for each probe as input. For MNase-Chip type of

data, signal corresponds to log base 2 ratio of the two channels. For MNase-Seq type of data, signal corresponds to the number of reads. Two example data sets are used for illustration of the package functionality. An example of input data:

```
> data(chip)
> colnames(chip)
[1] "Chr"      "Position" "Signal"
> dim(chip)
[1] 102   3
> chip[1:10, ]
    Chr Position     Signal
1  chr1   114471  0.3263090
2  chr1   114491  0.7421770
3  chr1   114511  0.9627163
4  chr1   114531  1.0542657
5  chr1   114551  0.9995793
6  chr1   114571  1.0187690
7  chr1   114591  0.9224163
8  chr1   114611  0.7274253
9  chr1   114631  0.2141027
10 chr1   114651 -0.1720987
```

In this package, there are two functions for users: **nucde** and **nucde.plot**. The function **nucde** is the main function for mapping nucleosome-linker boundaries from both MNase-Chip and MNase-Seq data using a non-homogeneous hidden-state model based on first order differences (Kuan et al.; 2009). The function **nucde.plot** plots the original signals with the nucleosome status. For MNase-Seq type of data, the smoothed signals are also shown.

The following are two examples of using the NucDe package in R. The details of these functions are given in the next section.

1. MNase-Seq type of data

```
> library(NucDe)
> data(seq)
> nucde_seq <- nucde(seq, type = "MNase-Seq", training = FALSE,
+     label = "example_seq", out = "example_seq.gff")
> nucde_seq

$chr1
states:
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
[149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[186] 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
[223] 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[260] 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[297] 1 1 1 1
log Likelihood:[1] -2268.037

$chr2
states:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
 [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1
[112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
[149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
[186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[223] 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
[297] 0 0 0 0
log Likelihood:[1] -2183.047
```

2. MNase-Chip type of data

```
> library(NucDe)
> data(chip)
> nucde_chip <- nucde(chip, type = "MNase-Chip", training = FALSE,
+     label = "example_chip", out = "example_chip.gff")
```

```
> nucde.plot(nucde_seq, "chr2", 2100, 2500)
```

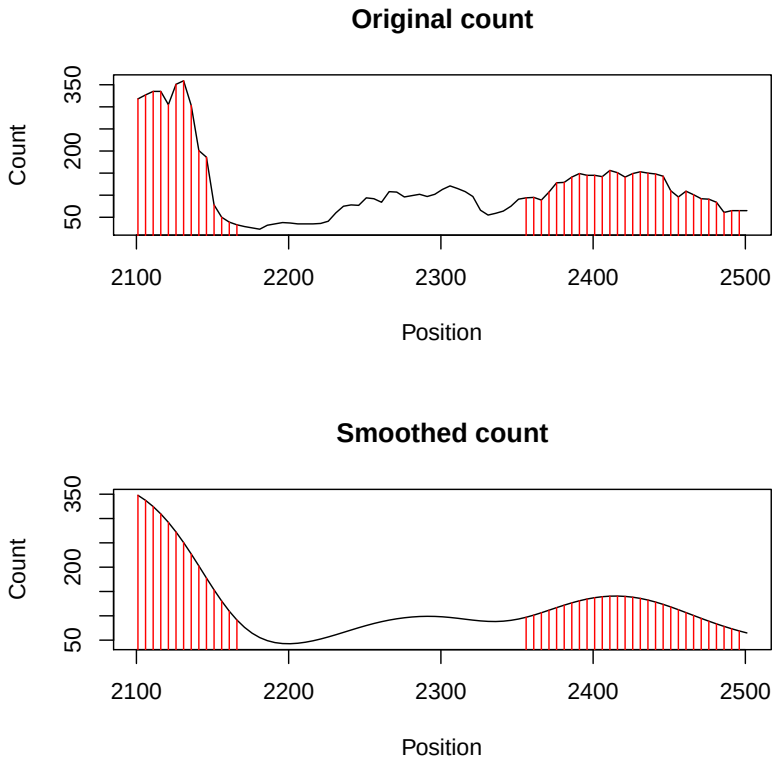

Figure 1: Example of nucde.plot function for MNase-Seq data.

```
> nucde_chip

$chr1

states:

 [1] 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1

[39] 0 0 0 1 1 1 1 1 1 0 0 0

log Likelihood:[1] -49.05753


$chr2

states:

 [1] 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0

[39] 0 1 1 1 1 1 1 1 1 0 0 0 1 1

log Likelihood:[1] -49.37836
```

2 Function descriptions

2.1 nucde

This function decodes the nucleosome status for each probe using a non-homogeneous hidden-state model. It also generates a GFF (general feature format) file and outputs the positions of nucleosome along with some summary statistics such as total signal, mean signal, and posterior probability.

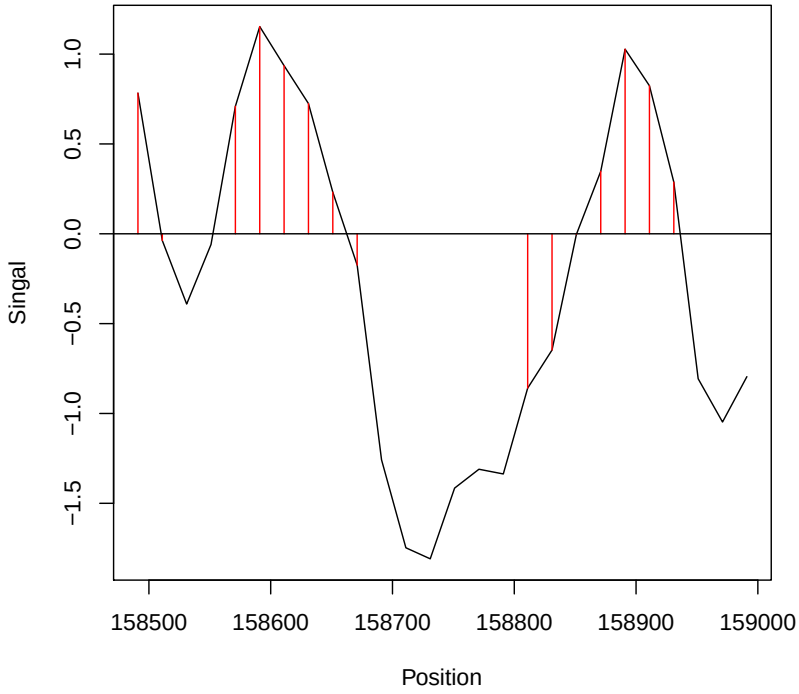

Figure 2: Example of nucde.plot function for MNase-Chip data.

Usage

```
nucde(data, type = "MNase-Seq", training = FALSE, label = 'NucDe',
out = "NucDe.results.gff")
```

Arguments

- `data`: a data frame containing chromosome, position, and signal.
- `type`: a string specifying the type of data. Must be either "MNase-Seq" or "MNase-Chip".
- `training`: logical value. If `training = TRUE`, the emission parameters and transition probabilities are trained by Baum-Welch algorithm. In the current version of *NucDe* training is not available for MNase-Seq.
- `seed`: the seed used for random number generation in the step of parameters' initialization.
- `label`: the label appears in the final GFF output file.
- `out`: the name of the final GFF output file.

Details

The **data** consists of at least three columns with column 1: chromosome ID, column 2: start coordinate, column 3: probe measurement. For MNase-Chip type of data, computation with `training = TRUE` estimates emission parameters and transition probabilities using Baum-Welch algorithm. It takes

longer computational time with `training = FALSE`. For MNase-Seq type of data, training is not available in current version.

Value

- `parameters`: the emission parameters and transition probabilities.
- `q`: the hidden states used in HMM.
- `states`: nucleosome states: 1-Nucleosome, 0-NFR/Linker.
- `loglik`: log-likelihood.
- `posterior`: posterior probability.

2.2 nucde.plot

The function plots the original signals with the nucleosome states. For MNase-Seq type of data, the smoothed signals are also shown.

Usage

```
nucde.plot(nucde, chr, start, stop)
```

Arguments

- `nucde`: output object from function *nucde*.
- `chr`: the chromosome to be plot

- `start`: the start position of the region to be plot
- `stop`: the end position of the region to be plot

Details

It displays the data with nucleosome states.

Value

The results of decoded nucleosome states.

References

[1] P-F. Kuan, D. Huebert, A. Gasch, and S. Keles (2009) . A Non-Homogeneous Hidden-State Model on First Order Differences for Automatic Detection of Nucleosome Positions. Statistical Applications in Molecular Biology and Genetics, 8(1): Article 29.

[2] Yuan, G., Liu, Y., Dion, M., Slack, M., Wu, L., Altschuler, S. and Rando, O. (2005). Genome-scale identification of nucleosome positions in S. cerevisiae, Science 309: 626-630.