Bayesian Analysis of ChIP-chip
The BAC package

Raphael Gottardo

October 22, 2008

Contents

Part I
Licensing

This software is distributed under the Artistic License 2.0. In addition, we would appreciate if you cite the following papers when using this software for publication.

> W. E. Johnson, Li, W., Meyer, C. A., Gottardo, R., Carroll, J. S., Brown, M., and Liu, X. S. (2006). Model-based analysis of tiling-arrays for ChIP-chip. PNAS 103:12457-12462.
>
> Gottardo et al. A Flexible and Powerful Bayesian Hierarchical Model for ChIP-Chip Experiments. Biometrics (2007) In press.

Then if you use the ER data (part of it is included in this package), you should cite the following paper:

> Carroll et al 2006 `http://www.cell.com/content/article/abstract?uid=PIIS0092867405004538`

Part II
Introduction

In our guide, we include example of codes that we hope will help you when using the BAC package. The codes are kept at the basic level for ease of understanding. Some of the options in the functions have been set by default. To learn more about the exact parameters and usage of each function, you may type `help(FUNCTION_NAME)` of the function of interest in R after the BAC package is loaded.

The common goal in analyzing this ChIP-chip data is to detect DNA-protein interactions from ChIP-chip experiments. As of now, the BAC package has mainly be tested with Affymetrix tiling array data. However, we expect it to work with other platforms (e.g. Agilent, Nimblegen, cDNA, etc.). In order to use the BAC package you will need both treatment (IP) and control conditions (e.g. Mock IP) with replicates under each condition. Note that BAC does not deal with normalization, so you will have to normalize your data before hands. For Affymetrix arrays, we refer you to the MATR package which contains efficient normalization procedures.

Part III
Loading the BAC Package

To load the BAC package in R, we type

```
> library(BAC)
```

Part IV
Detecting bound regions

We first load the estrogen receptor data (Carroll et al. 2006).

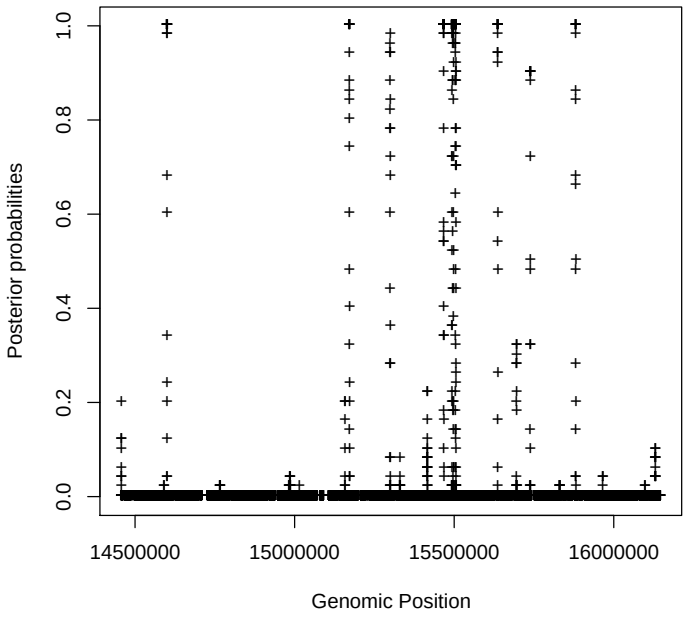
```
> data(ER)
```

then we calculate the (joint) posterior probabilities of enrichment based on 50 iterations for speed up, you should use more when you run the BAC function (see default parameters)

```
> load("bac.rda")
```

where w=5 is the window size, see parameter description for more details. Now you can have a look at these posterior probabilities to see where enriched regions might be, see Figure 1.

```
> plot(ER[, 1], BAConER$jointPP, pch = "+", xlab = "Genomic Position",
+     ylab = "Posterior probabilities")
```



Now regions can be called putative regions using the `CallRegions` function by a applying a 0.5 threshold (other thresholds can be used)

```
> ERregions <- CallRegions(ER[, 1], BAConER$jointPP, cutoff = 0.5,
+     maxGap = 500)
```

Finally, once we are happy with the regions detected we can easily create a BED file, which can be read and visualize in the UCSC genome browser.

```
> nRegions <- max(ERregions)
> BED <- matrix(0, nRegions, 4)
> for (i in 1:nRegions) {
+     BED[i, 2:3] <- range(ER[ERregions == i, 1])
+     BED[i, 4] <- max(BAConER$jointPP[ERregions == i]) * 1000
+ }
> BED <- data.frame(BED)
> BED[, 1] <- "chr21"
> names(BED) <- c("chrom", "chromStart", "chromEnd", "Score")
> print(BED)

   chrom chromStart chromEnd Score
1  chr21   14600350 14600679  1000
2  chr21   15171823 15172238  1000
3  chr21   15299545 15299909   980
```

```
4  chr21   15467160 15467873  1000
5  chr21   15493634 15495171  1000
6  chr21   15497944 15498912  1000
7  chr21   15503734 15504989  1000
8  chr21   15505697 15506008   920
9  chr21   15636775 15637099  1000
10 chr21   15738481 15738680   900
11 chr21   15880913 15881171  1000
```