

The MeDiChI ChIP-chip deconvolution package

David J Reiss, Institute for Systems Biology

February 22, 2008

Chromatin immunoprecipitation followed by hybridization to a genomic tiling microarray (ChIP-Chip) is a routinely used protocol for localizing the genomic targets of DNA-binding proteins. The resolution to which binding sites in this assay can be identified is commonly considered to be limited by two factors: (a) the resolution at which the genomic targets are tiled in the microarray, and (b) the large and variable lengths of the immunoprecipitated DNA fragments.

The **MeDiChI** package uses a generative model of binding sites in ChIP-chip data, and an approach for efficiently and robustly learning that model from diverse data sets. We have evaluated **MeDiChI**'s performance using simulated data, as well as on several diverse ChIP-chip data sets collected on widely different tiling array platforms for two different organisms (*S. cerevisiae* and *H. salinarum* *NRC-1*). We find that **MeDiChI** accurately predicts binding locations to a resolution greater than that of the probe spacing, even for overlapping peaks, and can increase the effective resolution of tiling array data by a factor of $5\times$ or better. Moreover, the method's performance on simulated data provides insights into effectively optimizing the experimental design for increased binding site localization accuracy and efficacy.

The **MeDiChI** package includes functions for constructing a peak profile, using it to deconvolve the ChIP-chip data, and plotting and examining the resulting fits. It also includes functions for generating simulated data for quality assessment.

We encourage you to read this package's publication, listed at the bottom of this document, and to visit the companion website, itemized just below.

<http://baliga.systemsbiology.net/medichi>

1 Technical Background and Notes

The **MeDiChI** package requires a very basic data format, by default a two-column matrix listing the central chromosomal coordinate of each probe and its intensity (*NOTE: intensities should be un-logged*). Multiple replicates may be included in this single data set. For more information, see the online help documentation for this package.

2 Demo 1: Load and deconvolve a small chromosomal segment of ChIP-chip data

In this first demonstration, we

- Initialize the MeDiChI library and load the example low-resolution *H. salinarum* *NRC-1* data set
- Deconvolve a small part of this data set on the main chromosome, including 100 bootstraps
- Plot and examine the resulting fit and bootstrap distribution

```
> library(MeDiChI)
```

Loading MeDiChI, version 0.1.8

```
> data("halo.lowres", package = "MeDiChI")
> fit <- chip.deconv(data.halo.lowres, where = "Chr", fit.res = 10,
+   center = 650000, wind = 20000, max.steps = 100, n.boot = 10,
+   kernel = kernel.halo.lowres, verbose = TRUE)
```

Using 1.520822 as data cutoff!

MEAN PROBE SPACING = 500

Step for min AIC: 33 15 ; BIC: 26 14 ; using: bic

13 coeffs at > 0 for LARS step 26

Number of coeffs: 13 ... Reduced to 8 non-redundant coeffs.

*** BOOTSTRAP ITER: 2 Fri Feb 22 12:34:03 2008 ***

Step for min AIC: 33 16 ; BIC: 30 14 ; using: bic

13 coeffs at > 0 for LARS step 30

Number of coeffs: 13 ... Reduced to 9 non-redundant coeffs.

*** BOOTSTRAP ITER: 3 Fri Feb 22 12:34:05 2008 ***

Step for min AIC: 33 13 ; BIC: 33 13 ; using: bic

12 coeffs at > 0 for LARS step 33

Number of coeffs: 12 ... Reduced to 8 non-redundant coeffs.

*** BOOTSTRAP ITER: 4 Fri Feb 22 12:34:06 2008 ***

Step for min AIC: 30 12 ; BIC: 17 11 ; using: bic

10 coeffs at > 0 for LARS step 17

Number of coeffs: 10 ... Reduced to 6 non-redundant coeffs.

*** BOOTSTRAP ITER: 5 Fri Feb 22 12:34:07 2008 ***

Step for min AIC: 37 15 ; BIC: 34 14 ; using: bic

13 coeffs at > 0 for LARS step 34

Number of coeffs: 13 ... Reduced to 7 non-redundant coeffs.

```

*** BOOTSTRAP ITER: 6 Fri Feb 22 12:34:08 2008 ***
Step for min AIC: 26 12 ; BIC: 26 12 ; using: bic
11 coeffs at > 0 for LARS step 26
Number of coeffs: 11 ... Reduced to 8 non-redundant coeffs.
*** BOOTSTRAP ITER: 7 Fri Feb 22 12:34:09 2008 ***
Step for min AIC: 33 15 ; BIC: 21 13 ; using: bic
12 coeffs at > 0 for LARS step 21
Number of coeffs: 12 ... Reduced to 8 non-redundant coeffs.
*** BOOTSTRAP ITER: 8 Fri Feb 22 12:34:11 2008 ***
Step for min AIC: 41 13 ; BIC: 26 12 ; using: bic
11 coeffs at > 0 for LARS step 26
Number of coeffs: 11 ... Reduced to 7 non-redundant coeffs.
*** BOOTSTRAP ITER: 9 Fri Feb 22 12:34:12 2008 ***
Step for min AIC: 30 12 ; BIC: 30 12 ; using: bic
11 coeffs at > 0 for LARS step 30
Number of coeffs: 11 ... Reduced to 8 non-redundant coeffs.
*** BOOTSTRAP ITER: 10 Fri Feb 22 12:34:13 2008 ***
Step for min AIC: 21 11 ; BIC: 21 11 ; using: bic
10 coeffs at > 0 for LARS step 21
Number of coeffs: 10 ... Reduced to 7 non-redundant coeffs.

```

```
> coef(fit)
```

```

      position intensity
83  640571.5 2.2199189
210 644331.5 2.1581575
380 646521.5 1.0803201
501 649221.5 0.4423681
599 650201.5 1.0666237
706 654261.5 2.4796942
819 657381.5 1.6817031
907 659751.5 0.5821610

```

```
> plot(fit, plot.genes = TRUE, cex = 0.5, cex.lab = 0.8, cex.axis = 0.8)
```

You should see a plot of the chunk of data and the resulting fit appear, similar to this one:

4 Demo 3: Learn the peak profile from data

In the previous two examples, we used a pre-computed ChIP-chip peak profile (the same one that was used in the manuscript [1]). In this demo, we will show how this peak profile is computed from the data set. This method starts with a guess for good starting parameters, generates a binding profile using those parameters, and detects binding sites across the data set as in the previous example. It then identifies the 20 brightest, most isolated peaks, and adjusts the peak profile parameters to better reflect those peaks. This process is repeated several times. It can be sped up by adjusting the parameters of the following function to make its repeated calls to `deconv.entire.genome()` execute faster. In this demonstration, we

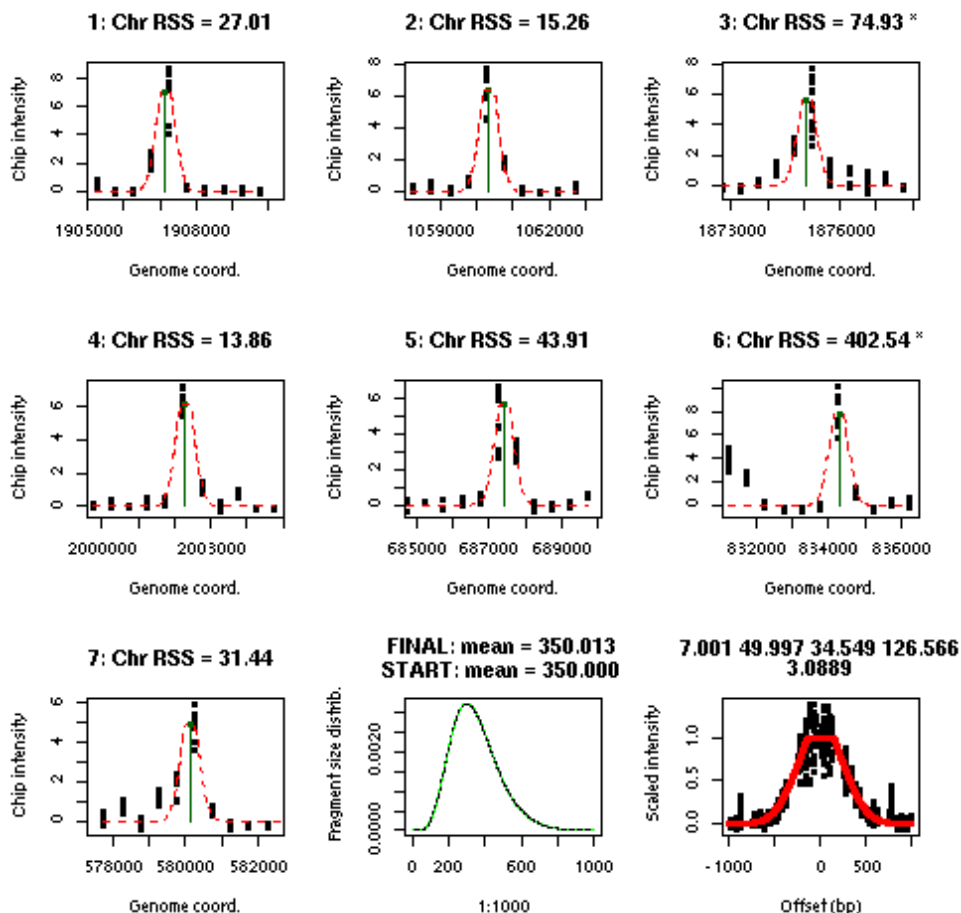
- Fit a peak profile (kernel) to the low-resolution *H. salinarum* *NRC-1* data set
- Plot the resulting fit and learn how to use the output for a run of `chip.deconv()` or `deconv.entire.genome()`

```
> peak.params <- fit.peak.profile(data.halo.lowres, tile.size = 500,  
+   quant.cutoff = "q0.98", chrom = "Chr", fit.res = 30, max.steps = 100,  
+   plot = T, name = "Halo-lowres")
```

The resulting structure can be plotted, which will display seven of the brightest peaks contributing to the fit, the best-fit DNA fragment length distribution, and the final peak profile, including all data data points surrounding the 20 brightest, isolated peaks (shifted and normalized based on their best-fit positions and intensities).

```
> plot(peak.params)
```

This results in something like this for the above example:



`peak.params$kernel` may be used as a kernel for more accurate deconvolution `chip.deconv()` or `deconv.entire.genome()` as in the previous two examples.

5 For More Information

For more information, please visit:

<http://baliga.systemsbiology.net/medichi>

6 References

- Reiss, DJ and Facciotti, MT and Baliga, NS. (2007). "Model-based deconvolution of genome-wide DNA binding", *Bioinformatics*. doi: 10.1093/bioinformatics/btm592.