chipchipnorm: normalization for chip-chip data

Shouyong Peng, Jonathan Dreyfuss, and Peter Park

February 6, 2009

Harvard-Partners Center for Genetics and Genomics, 77 Avenue Louis Pasteur, Boston, MA 02115

1 Introduction

chipchipnorm is a R package that can be incorporated into the normalization workflow for chip-chip data, chromatin immunoprecipitation (ChIP) with microarray technology (chip). It implements the novel normalization scheme from Peng et al. (2007), which was shown to obviate the need for mock control experiments (yielding significant cose saving to the investigator).

The package implements a novel rotation scheme to get rid of major dye trend in the data and then applies a global loess smooth. Because the rotation works on lagged differences of tiled probes, *it is essential that rows of the data are ordered by genomic position.*

You may download the package through Bioconductor,

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("chipchipnorm")
```

and load it using

```
> library("chipchipnorm")
```

2 Setup

If R and G are matrices of raw red and green intensities, respectively, with probes as rows and chips as columns, then it is customary to use $M = log(R/G)$ and $A = log(R * G)/2$ in microarray data analysis, with log base 2. A plot of M versus A is then often used to normalize, since dye bias can make the log ratio (M) dependent on log intensity (A).

An `MA` object with components `M` and `A`, and possibly others, is the main input for functions in *chipchipnorm*. Only components `M` and `A` are ever altered or used. Common Bioconductor packages for preprecessing two-color data which yield an appropriate input object are *limma* and *marray*. For example, if the Genepix image analysis program (which creates GPR files) was used on your data, then using *limma* you could type:

```
> library(limma)
> RG.gpr <- read.maimages(source = "genepix", ext = "gpr")
> MA.gpr <- MA.RG(RG.gpr)
```

which constructs raw (unnormalized) `MA` object `MA.gpr`.

If the `wt.fun` argument in `read.maimages` is used, then the `weights` component of `MA.gpr` becomes populated by a matrix of spot-quality weights. You can input weights to *chipchipnorm* normalization procedures provided all elements are between 0 (no weight) and 1 (full weight). Here we use the convenient wrapper function `lagNorm`.

```
> MA.norm <- lagNorm(MA = MA.gpr, nlag = 8, w = MA.gpr$weights)
```

To choose the lag quantity `nlag`, take the smallest probe lag such that the curves plotted by `lagNoise` flatten out. More information on selecting a lag parameter can be found in Peng et al. (2007).

3 Data

The methodological investigation from Peng et al. (2007) is made possible by a unique data set generated on dosage compensation in Drosophila Alekseyenko et al. (2006). The MSL complex is known to bind specifically to the X chromosome to up-regulate the X-linked genes, while the 2L chromosome is included for comparison. To save disk space, a small portion of the dataset has been included with *chipchipnorm* as an example dataset in workspace `MSLdata`. This workspace contains R object `MA.ex` with components:

M numeric matrix of log ratios with rows corresponding to probes and columns to chips.

A numeric matrix of log intensities with rows corresponding to probes and columns to chips.

The column names of these matrices indicates if they came from replicate 1 or 2 and whether they are mock control (“bkg”) or experiment (“exp”), while the row names are probe identifiers. This workspace also contains `grp`, which is a character vector whose i^{th} element (either “X” or “2L”) gives the chromosome (“group”) corresponding to row i of the above matrices.

4 Example

Now we load the example dataset, get diagnostics to assess noise (here from the 1^{st} chip), and run the normalization.

```
> data(MSLdata)
> noise.mat <- lagNoise(MA = MA.ex, chip = 1, group = grp)
> MA.norm <- lagNorm(MA = MA.ex, nlag = 8, group = grp)
```

`MA.norm` holds all normalized data and `noise.mat` holds the noise estimates for each lag value from chip 1. `lagNoise` always renders a plot of the specified chip, though `lagNorm` can be set not to plot through `lagNorm( , plot.names=NULL)`.

`lagNorm` implements the rotation followed by a global loess smooth. If you prefer to use some other loess-type procedure (such as print-tip loess), you have the option of utilizing the novel rotation in your own workflow. For instance, rather than using the wrapper `lagNorm`, you could execute:

```
> data(MSLdata)
> angle <- getAngle(MA = MA.ex, nlag = 8)
> MA.rot <- rotateMA(MA = MA.ex, angle = angle)
```

followed by your favorite loess-type normalization procedure on `MA.rot`.

References

Artyom A Alekseyenko, Erica Larschan, WR Lai, PJ Park, and Mitzi I Kuroda. ChIP-chip analysis reveals that the Drosophila MSL complex selectively identifies active genes on the male X chromosome. *Genes Dev*, 20(7):848–857, 2006.

Shouyong Peng, Artyom A Alekseyenko, Erica Larschan, Mitzi I Kuroda, and Peter J Park. Normalization and experimental design for ChIP-chip data. *BMC Bioinformatics*, 8(219), 2007.