shinyMethyl: interactive visualization of Illumina 450K methylation arrays

Jean-Philippe Fortin, Kasper Daniel Hansen

May 28, 2014

1 Introduction

Up to now, more than 10,000 methylation samples from the state-of-the-art 450K microarray are already available through The Cancer Genome Atlas portal [1] and the Gene Expression Omnibus (GEO) [2]. Large-scale comparison studies, for instance between cancers or tissues, become possible epigenome-widely. These large studies often require a substantial amount of time spent on data preprocessing, quality control and other exploratory analysis. Moreover, for such studies, it is not rare to encounter significant batch effects, and those can have a dramatic impact on the validity of the biological results [3, 4]. With that in mind, we have developed *shinyMethyl* to make the preprocessing of large 450K datasets intuitive, enjoyable and reproducible. *shinyMethyl* is an interactive visualization tool for Illumina 450K methylation array data based on the packages *minfi* and *shiny* [5, 6].

A few mouse clicks allow the user to appreciate insightful biological inter-array differences on a large scale. The goal of *shinyMethyl* is two-fold: (1) summarize a high-dimensional 450K array experiment into an exportable small-sized R object and (2) launch an interactive visualization tool for quality control assessment as well as exploration of global methylation patterns associated with different phenotypes.

Because a picture is worth a thousand words, we have implemented an online example of *shinyMethyl* ready to use at `http://spark.rstudio.com/jfortin/shinyMethyl/`

2 Example dataset

To take a quick look at how the interactive interface of *shinyMethyl* works, we have included an example dataset in the companion package *shinyMethylData*. The dataset contains the extracted data of 369 Head and Neck cancer samples downloaded from The Cancer Genome Atlas (TCGA) data portal [1]: 310 tumor samples, 50 matched normals and 9 replicates of a control cell lines. The first `shinyMethylSet` was created from the raw data (no normalization) and is stored under the name `summary.tcga.raw.rda`; the second `shinyMethylSet` was created from a `GenomicRatioSet` containing the normalized data and the file is stored under the name `summary.tcga.norm.rda`. The

samples were normalized using functional normalization, a preprocessing procedure that we recently developed for heterogeneous methylation data [7].

To launch *shinyMethyl* with this TCGA dataset, enter the following commands in an *R* session:

```
library(shinyMethyl)
library(shinyMethylData)
runShinyMethyl(summary.tcga.raw, summary.tcga.norm)
```

The interactive interface will take a few seconds to be launched in your default HTML browser.

3 Creating your own dataset visualization

a) For users familiar with RGChannelSet objects

If you are familiar with `RGChannelSet` objects from the *minfi* package, and if you already have an `RGChannelSet` for your experiment, you can launch *shinyMethyl* in two steps:

```
library(shinyMethyl)
summary <- shinySummarize(yourRGSet)
runShinyMethyl(summary)
```

Otherwise go to section **b** to learn how to create an `RGChannelSet` object.

b) To create an RGChannelSet object

An `RGChannelSet` is a *minfi* object containing the raw intensities in the green and red channels of your experiment samples. To create an `RGChannelSet`, you will need to have the raw files of the experiment with extension .IDAT (we refer to those as .IDAT files) If you do not have these files, you might want to ask your collaborators or the processing core of your lab if they have them. You absolutely need them to use both *minfi* and *shinyMethyl*. The vignette in *minfi* describes carefully how to read the data in for different scenarios and how to construct an `RGChannelSet`. Here, we show a quick way to create an `RGChannelSet` from the .IDAT files contained in the package *minfiData*.

First, let's load the packages:

```
library(minfiData)
library(minfi)
```

Second, we need to tell *R* which directory contains the .IDAT files and the experiment sheet:

```
baseDir <- system.file("extdata", package = "minfiData")
# baseDir <- "/home/yourDirectoryPath"
```

Third, we need to read in the experiment sheet:

```
targets <- read.450k.sheet(baseDir)
head(targets)
```

Finally, we construct the RGChannelSet:

```
RGSet <- read.450k.exp(base = baseDir, targets = targets)
```

The function pData() in *minfi* allows to see the phenotype data of the samples:

```
pd <- pData(RGSet)
head(pd)
```

Please see **c** to create a shinyMethylSet necessary to launch *shinyMethyl*.

c) To create a shinyMethylSet object

shinyMethyl requires that you have already created an RGChannelSet. From the RGChannelSet created in the previous section, we create a shinyMethylSet by using the command shinySummarize()

```
myShinyMethylSet <- shinySummarize(RGSet)
```

Please see **d** to launch shinyMethyl

d) To launch the interactive interface

To launch a *shinyMethyl* session, simply pass your shinyMethylSet object to the *runShinyMethyl()* function as follows:

```
runShinyMethyl(myShinyMethylSet)
```

4 How to use the different shinyMethyl panels

5 Advanced option: visualization of normalized data

shinyMethyl also offers the possibility to visualize normalized data that are stored in a GenomicRatioSet object. For instance, suppose we normalize the data by using the quantile normalization algorithm implemented in *minfi*:

```
GRSet.norm <- preprocessQuantile(RGSet)
```

We can then create two separate shinyMethylSet objects corresponding to the raw and normalized data respectively:

```
summary   <- shinySummarize(RGSset)
summary.norm <- shinySummarize(GRSet.norm)
```

To launch the *shinyMethyl* interface, use `runShinyMethyl()` with the first argument being the `shinyMethylSet` extracted from the raw data and the second argument being the `shinyMethylSet` extracted from the normalized data as follows:

```
runShinyMethyl(summary, summary.norm)
```

6 What does a shinyMethylSet contain?

A `shinyMethylSet` object contains the following summary data from a 450K experiment: the names of the samples, a data frame for the phenotype, a list of quantiles for the M and Beta values, a list of quantiles for the methylated and unmethylated channels intensities and a list of quantiles for the copy numbers, the green and red intensities of different control probes, and the principal component analysis (PCA) results performed on the Beta values. One can access the different summaries by using the slot operator `@`. The slot names can be obtained with the function `slotNames` as follows:

```
library(shinyMethyl)
library(shinyMethylData)
data(summary.tcga.raw)
summary.tcga.raw
slotNames(summary.tcga.raw)
```

Session info

Here is the output of `sessionInfo` on the system on which this document was compiled:

- R version 3.1.0 (2014-04-10), `x86_64-apple-darwin10.8.0`
- Locale: `en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils
- Other packages: Biobase 2.25.0, BiocGenerics 0.11.2, Biostrings 2.33.9, bumphunter 1.5.3, foreach 1.4.2, GenomeInfoDb 1.1.6, GenomicRanges 1.17.16, IlluminaHumanMethylation450kanno.ilmn12.hg19 0.2.1, IlluminaHumanMethylation450kmanifest 0.4.0, IRanges 1.99.15, iterators 1.0.7, lattice 0.20-29, locfit 1.5-9.1, minfi 1.11.5, minfiData 0.7.0, S4Vectors 0.0.7, XVector 0.5.6
- Loaded via a namespace (and not attached): annotate 1.43.4, AnnotationDbi 1.27.7, base64 1.1, beanplot 1.1, BiocStyle 1.3.1, codetools 0.2-8, DBI 0.2-7, digest 0.6.4, doRNG 1.6, evaluate 0.5.5, formatR 0.10, genefilter 1.47.5, grid 3.1.0, highr 0.3, illuminaio 0.7.0, knitr 1.6, limma 3.21.4, MASS 7.3-33, matrixStats 0.8.14, mclust 4.3, multtest 2.21.0, nlme 3.1-117, nor1mix 1.1-4, pkgmaker 0.22, plyr 1.8.1, preprocessCore 1.27.0, R.methodsS3 1.6.1, RColorBrewer 1.0-5, Rcpp 0.11.1, registry 0.2, reshape 0.8.5, rngtools 1.2.4, RSQLite 0.11.4,

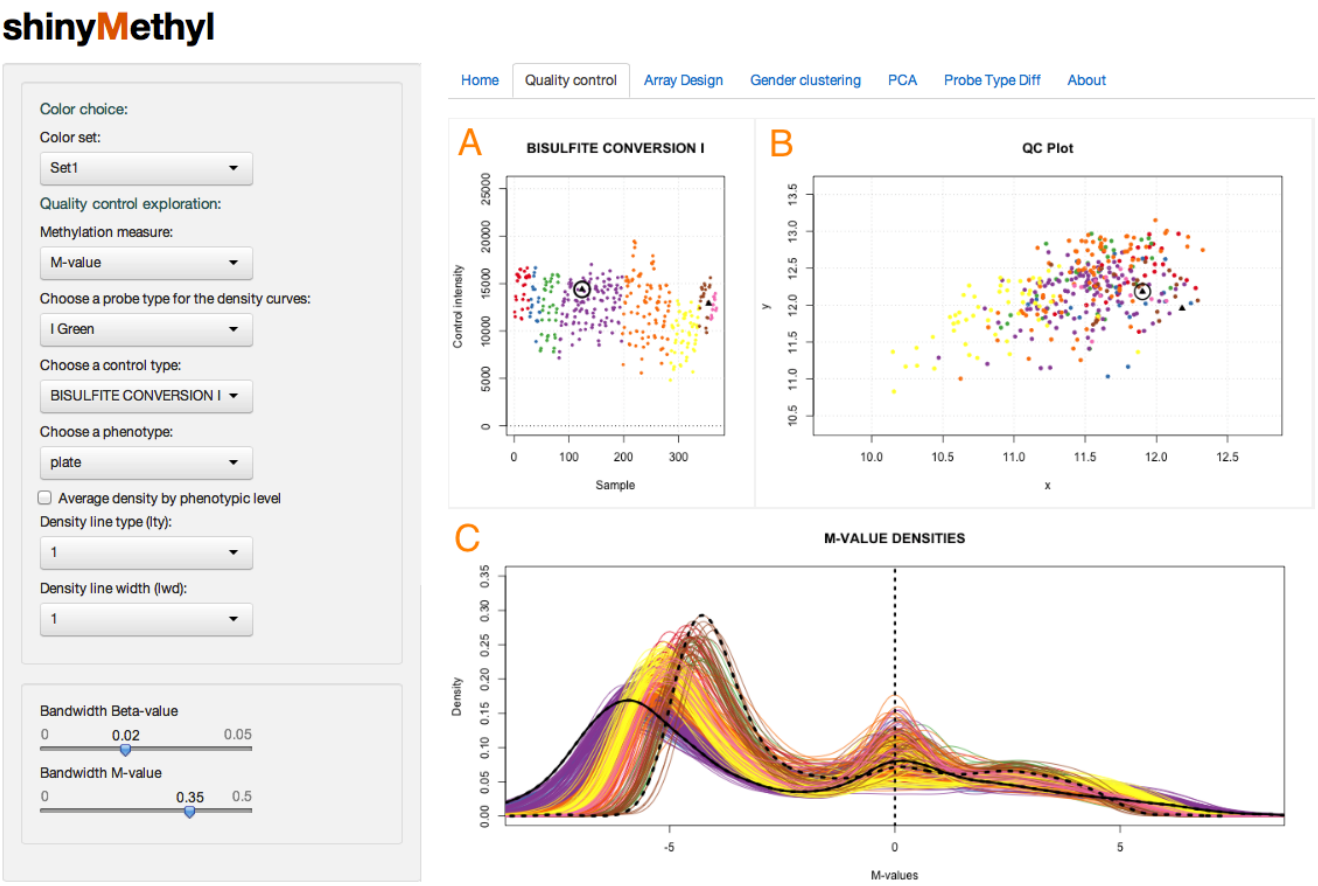

Figure 1: **Example of interactive visualization and quality control assessment** The three plots react simultaneously to the user mouse clicks and selected samples are represented in black. Colors represent batch. (a) Bisulfite conversion probes intensities (b) the median intensity of the M channel against the median intensity of the U channel (c) M-value densities for Infinium I probes for the raw data. The dashed and solid lines in black correspond to the two samples selected by the user and match to the dots circled in black in the left-hand plots. The left-hand-side panel allows users to select different tuning parameters for the plots, as well as different phenotypes for the colors.

siggenes 1.39.0, splines 3.1.0, stats4 3.1.0, stringr 0.6.2, survival 2.37-7, tools 3.1.0, XML 3.98-1.1, xtable 1.7-3, zlibbioc 1.11.1

References

[1] The Cancer Genome Atlas. *Data portal*, 2014. Online. URL: https://tcga-data.nci.nih.gov/tcga/.

[2] R. Edgar, M. Domrachev, and A. E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, 30(1):207–210, Jan 2002.

[3] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead,

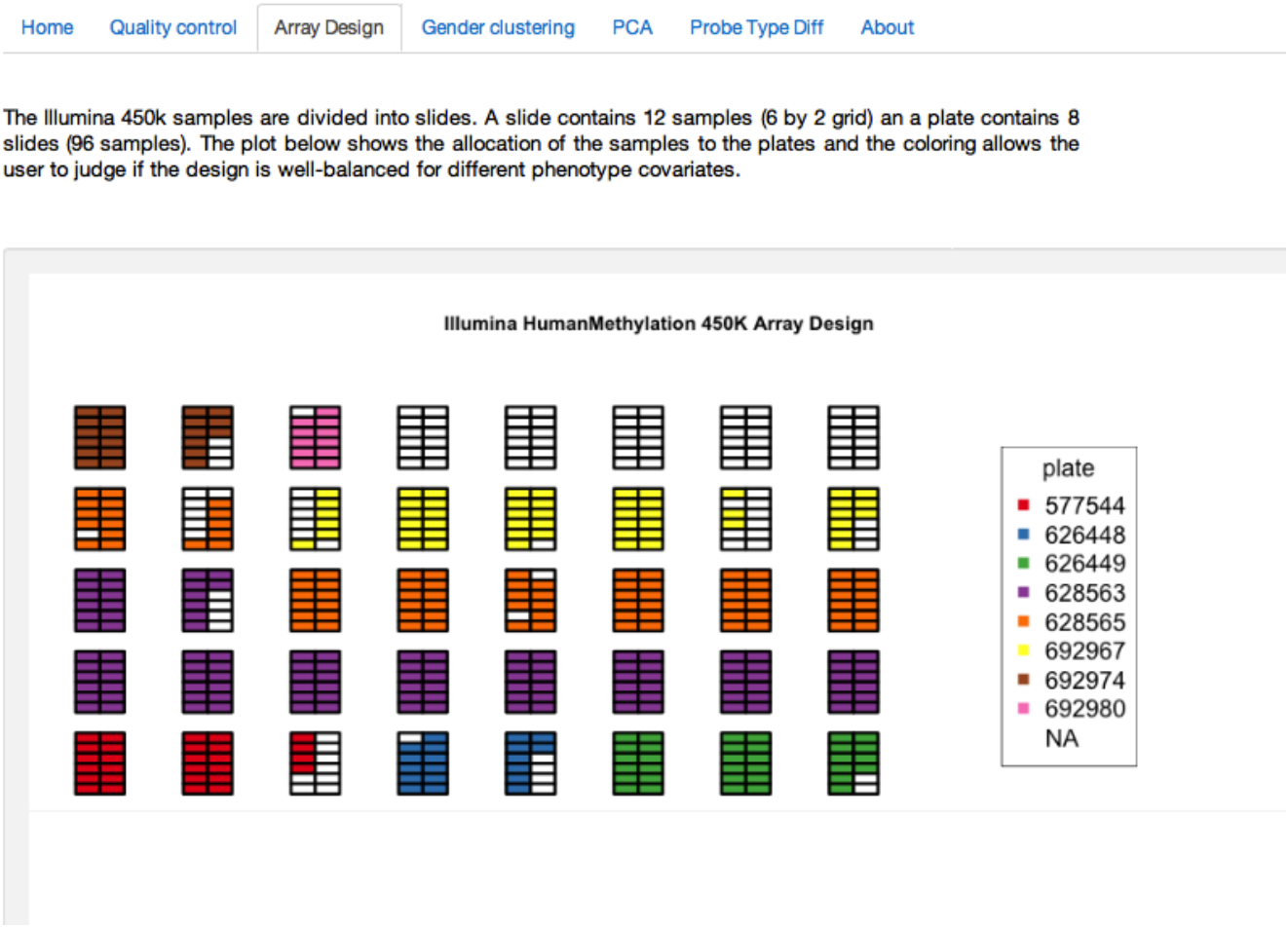

Figure 2: **Array design panel** The plot represents the physical slides (6 $\times$ 2 samples) on which the samples were assayed. The user can select the phenotype and colors allow to distinguish balanced designs.

W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010. doi:10.1038/nrg2825.

[4] Kristin N Harper, Brandilyn A Peters, and Mary V Gamble. Batch effects and pathway analysis: two potential perils in cancer studies involving dna methylation array analysis. *Cancer Epidemiol Biomarkers Prev*, 22(6):1052–60, 2013. doi:10.1158/1055-9965.EPI-13-0114.

[5] Martin J Aryee, Andrew E Jaffe, Hector Corrada Bravo, Christine Ladd-Acosta, Andrew P Feinberg, Kasper D Hansen, and Rafael A Irizarry. Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays. *Bioinformatics*, 30(10):1363–1369, 2014. doi:10.1093/bioinformatics/btu049, PMID:24478339.

[6] RStudio and Inc. *shiny: Web Application Framework for R*, 2014. R package version 0.9.1. URL: http://CRAN.R-project.org/package=shiny.

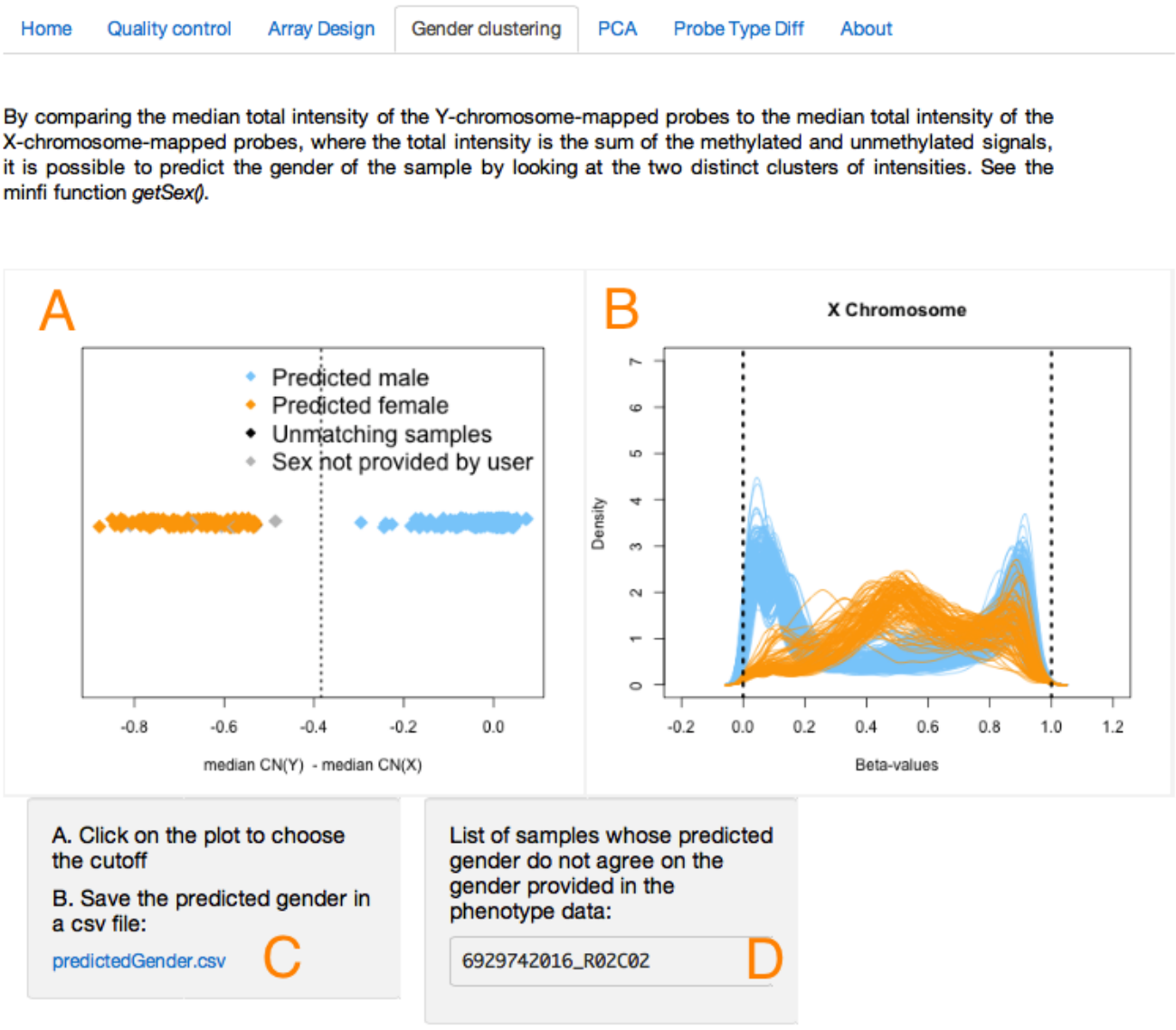

Figure 3: **Sex prediction algorithm panel** The difference of the median copy number intensity for the Y chromosome and the median copy number intensity for the X chromosome can be used to separate males and females. In a), the user can select the vertical cutoff (dashed line) manually with the mouse to separate the two clusters (orange for females, blue for males). Corresponding Beta-value densities appear in b). The predicted sex can be downloaded in a csv file in c), and samples for which the predicted sex differs from the sex provided in the phenotype will appear in d).

[7] Jean-Philippe Fortin, Aurelie Labbe, Mathieu Lemire, Brent W. Zanke, Thomas J. Hudson, Elana J. Fertig, Celia M.T. Greenwood, and Kasper D. Hansen. Functional normalization of 450k methylation array data improves replication in large cancer studies. *bioRxiv*, 2014. URL: http://biorxiv.org/content/early/2014/02/23/002956, arXiv:http://biorxiv.org/content/early/2014/02/23/002956.full.pdf, doi:10.1101/002956.

Figure 4: **Principal component analysis (PCA) panel.** The user can select the principal components to visualize (PC1 and PC2 are shown in the current plot) and can choose the phenotype for the coloring. In the present plot, one can observe that the first two principal components distinguish tumor samples from normal samples for the TCGA example dataset (see example dataset section).

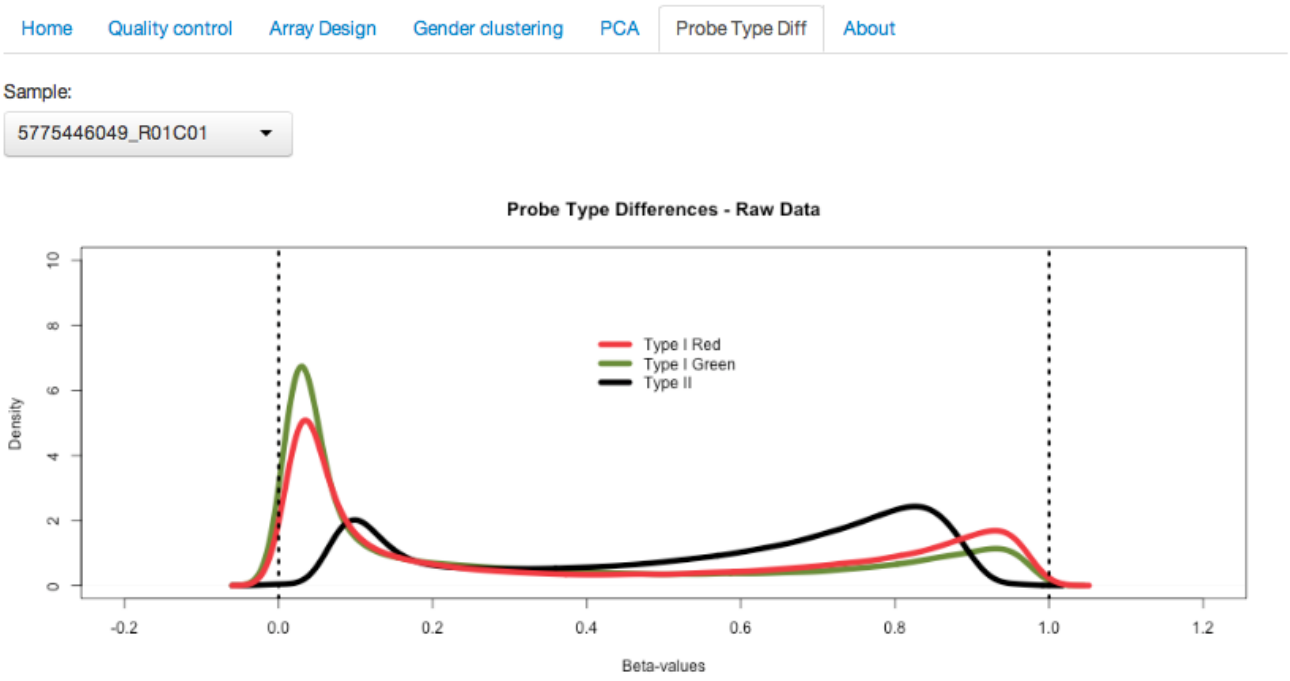
Home
Quality control
Array Design
Gender clustering
PCA
Probe Type Diff
About
Sample:
5775446049_R01C01
Probe Type Differences - Raw Data
Type I Red
Type I Green
Type II
Density
Beta-values
0
2
4
6
8
10
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

Figure 5:

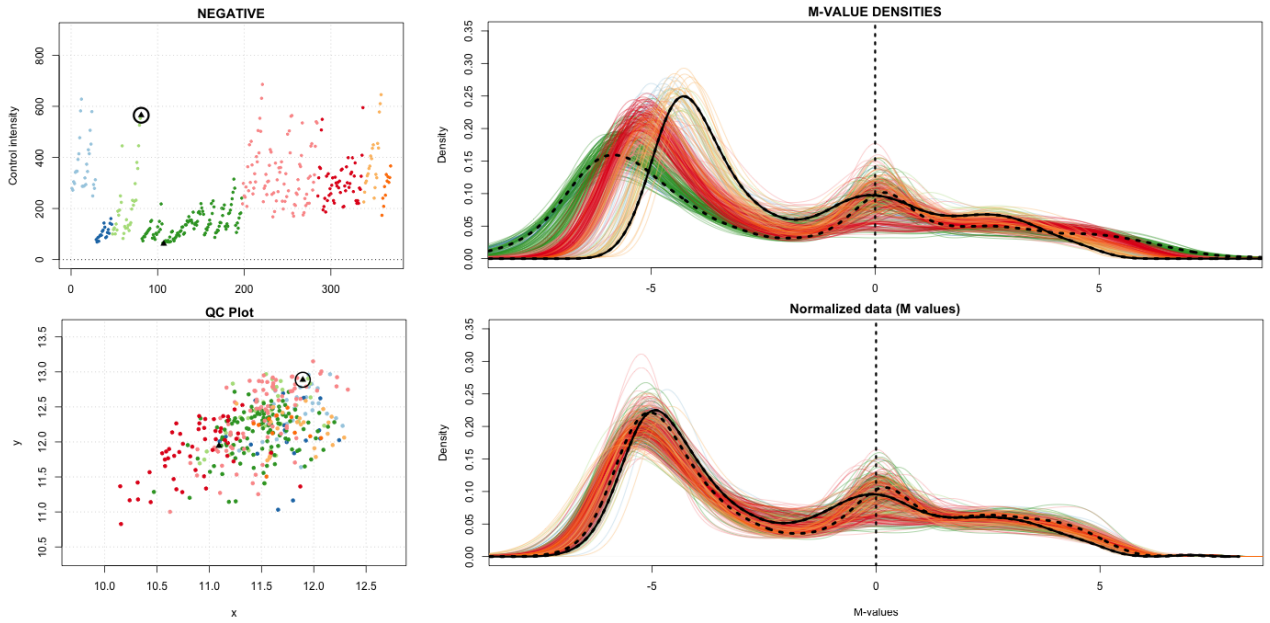

Figure 6: **Example of interactive visualization and quality control assessment** The four plots react simultaneously to the user mouse clicks and selected samples are represented in black. Colors represent batch. (a) Average negative control probes intensities (b) the median intensity of the M channel against the median intensity of the U channel (c-d) M-value densities for Infinium I probes before and after functional normalization. The dashed and solid lines in black correspond to the two samples selected by the user and match to the dots circled in black in the left-hand plots.

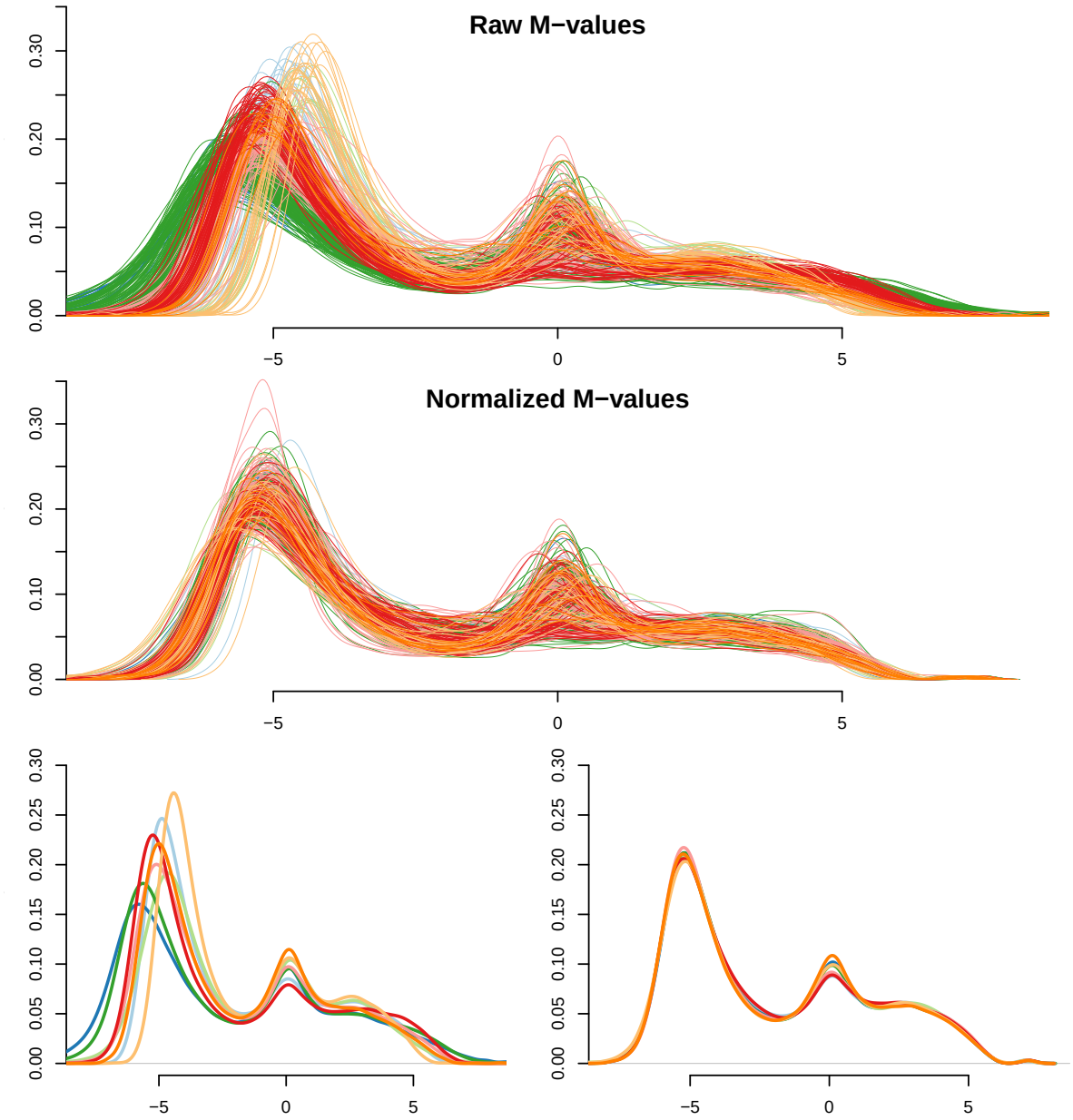

Figure 7: **Visualization of batch effects in the TCGA HNSCC dataset** Densities of the M-values for Type I green probes before (a) and after (b) functional normalization as presented in the *shinyMethyl* interactive interface. Each curve represents one sample and different colors represent different batches. Plots (c-d) show the average density for each plate before and after normalization. One can observe that functional normalization removed significantly global batch effects.

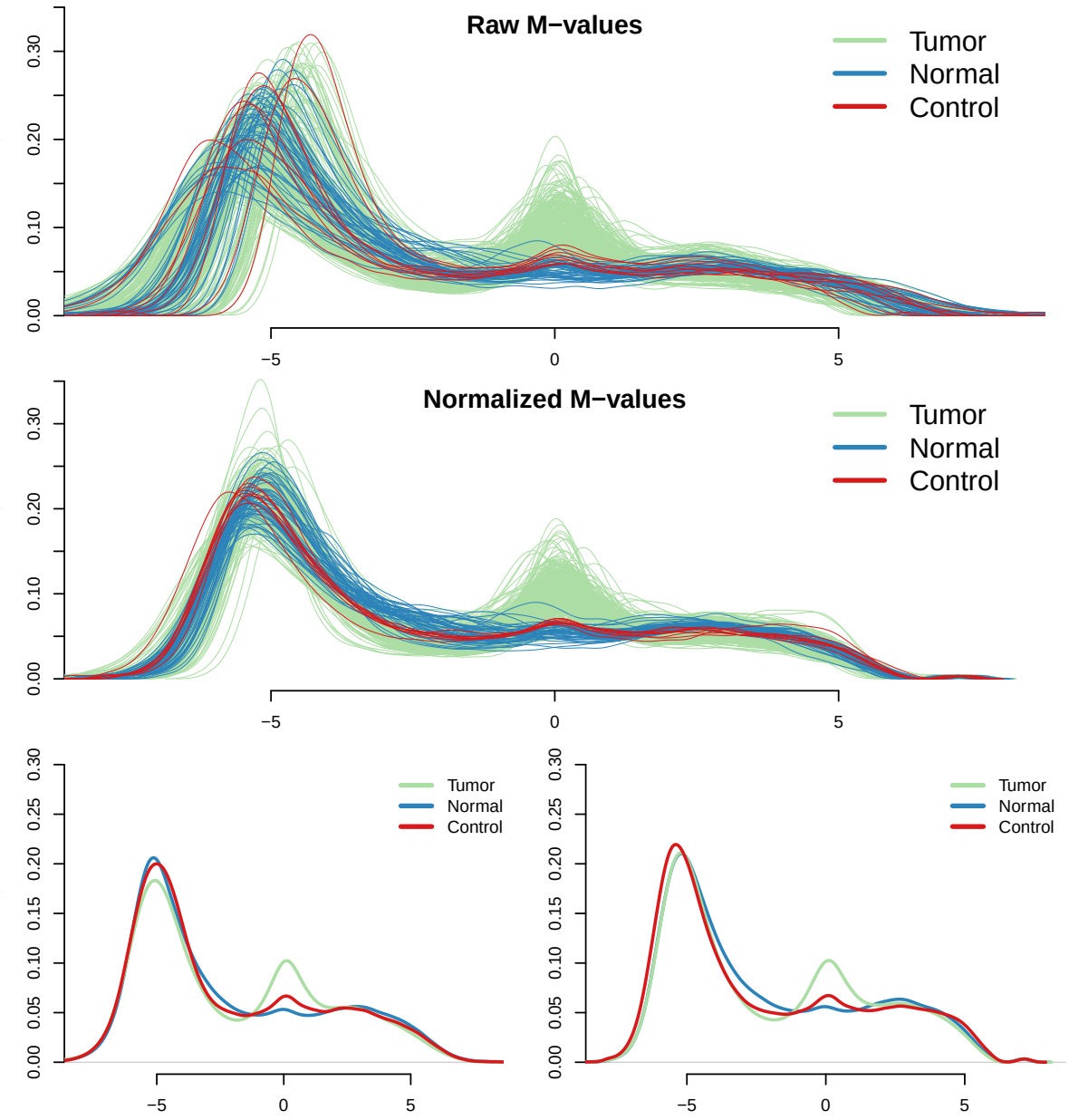

Figure 8: **Visualization of cancer/normal differences in the TCGA HNSCC dataset** Densities of the M-values for Type I green probes before (a) and after (b) functional normalization as presented in the *shinyMethyl* interactive interface. Green and blue densities represent tumor and normal samples respectively, and red densities represent 9 technical replicates of a control cell line. Plots (c-d) show the average density for each sample group before and after normalization. Functional normalization preserves the expected marginal differences between normal and cancer, while reducing the variation between the technical controls (red lines).