Using the charm package to estimate DNA methylation levels and find differentially methylated regions

Martin Aryee[*], Peter Murakami, Rafael Irizarry

March, 2011

Johns Hopkins School of Medicine / Johns Hopkins School of Public Health
Baltimore, MD, USA

1 Introduction

The Bioconductor package charm can be used to analyze DNA methylation data generated using McrBC fractionation and two-color Nimblegen microarrays. It is customized for use with data the from the custom CHARM microarray [2], but can also be applied to many other Nimblegen designs. The preprocessing and normalization methods are described in detail in [1].

Functions include:

- Quality control
- Finding suitable control probes for normalization
- Percentage methylation estimates
- Identification of differentially methylated regions

As input we will need raw Nimblegen data (.xys) files and a corresponding annotation package built with pdInfoBuilder. This vignette uses the following packages:

- charm: contains the analysis functions
- charmData: an example dataset
- pd.charm.hg18.example: the annotation package for the example dataset
- BSgenome.Hsapiens.UCSC.hg18: A BSgenome object containing genomic sequence used for finding non-CpG control probes

[*]aryee@jhu

Each sample is represented by two xys files corresponding to the untreated (green) and methyl-depleted (red) channels. The 532.xys and 635.xys suffixes indicate the green and red channels respectively.

2 Analyzing data from the custom CHARM microarray

Load the charm package:

```
R> library(charm)
R> library(charmData)
```

3 Read in raw data

Get the name of your data directory (in this case, the example data):

```
R> dataDir <- system.file("data", package = "charmData")
R> dataDir
```

```
[1] "/loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data"
```

First we read in the sample description file:

```
R> phenodataDir <- system.file("extdata", package = "charmData")
R> pd <- read.delim(file.path(phenodataDir, "phenodata.txt"))
R> phenodataDir
```

```
[1] "/loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/extdata"
```

```
R> pd
```

```
          filename   sampleID tissue
1    136421_532.xys  441_liver  liver
2    136421_635.xys  441_liver  liver
3    136600_532.xys 449_spleen spleen
4    136600_635.xys 449_spleen spleen
5   3788602_532.xys  449_liver  liver
6   3788602_635.xys  449_liver  liver
7   3822402_532.xys 441_spleen spleen
8   3822402_635.xys 441_spleen spleen
9   5739902_532.xys  624_colon  colon
10  5739902_635.xys  624_colon  colon
11  5875602_532.xys  441_colon  colon
12  5875602_635.xys  441_colon  colon
```

A valid sample description file should contain at least the following (arbitrarily named) columns:

- a filename column
- a sample ID column
- a group label column (optional)

The sample ID column is used to pair the methyl-depleted and untreated data files for each sample. The group label column is used when identifying differentially methylated regions between experimental groups.

The `validatePd` function can be used to validate the sample description file. When called with only a sample description data frame and no further options `validatePd` will try to guess the contents of the columns.

```
R> res <- validatePd(pd)
```

Now we read in the raw data. The `readCharm` command makes the assumption (unless told otherwise) that the two xys files for a sample have the same file name up to the suffixes 532.xys (untreated) and 635.xys (methyl-depleted).

```
R> rawData <- readCharm(files = pd$filename, path = dataDir,
     sampleKey = pd)
```

```
Checking designs for each XYS file... Done.
Allocating memory... Done.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/136421_532.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/136600_532.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/3788602_532.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/3822402_532.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/5739902_532.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/5875602_532.xys.
Checking designs for each XYS file... Done.
Allocating memory... Done.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/136421_635.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/136600_635.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/3788602_635.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/3822402_635.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/5739902_635.xys.
Reading /loc/home/biocbuild/bbs-2.8-bioc/R/library/charmData/data/5875602_635.xys.
```

```
R> rawData
```

```
TilingFeatureSet (storageMode: lockedEnvironment)
assayData: 243129 features, 6 samples
  element names: channel1, channel2
protocolData
  rowNames: 136421 136600 ... 5875602 (6 total)
  varLabels: filenamesChannel1 filenamesChannel2
    dates1 dates2
  varMetadata: labelDescription channel
phenoData
  rowNames: 136421 136600 ... 5875602 (6 total)
  varLabels: sampleID tissue arrayUT arrayMD
  varMetadata: labelDescription channel
featureData: none
experimentData: use 'experimentData(object)'
Annotation: pd.charm.hg18.example
```

4 Array quality assessment

We can calculate array quality scores and generate a pdf report with the `qcReport` command.

A useful quick way of assessing data quality is to examine the untreated channel where we expect every probe to have signal. Very low signal intensities on all or part of an array can indicate problems with hybridization or scanning. The CHARM array and many other designs include background probes that do not match any genomic sequence. Any signal at these background probes can be assumed to be the result of optical noise or cross-hybridization. Since the untreated channel contains total DNA a successful hybridization would have strong signal for all untreated channel genomic probes. The array signal quality score (pmSignal) is calculated as the average percentile rank of the signal robes among these background probes. A score of 100 means all signal probes rank above all background probes (the ideal scenario).

```
R> qual <- qcReport(rawData, file = "qcReport.pdf")
R> qual
```

```
        pmSignal       sd1       sd2
136421  78.56437 0.1950274 0.1932112
136600  81.46541 0.1755225 0.1227921
3788602 83.95419 0.1249030 0.2409803
3822402 81.43751 0.1180708 0.1824810
5739902 82.55727 0.1490854 0.2035761
5875602 79.38069 0.3130266 0.3962373
```

The PDF quality report is shown in Appendix A. Three quality metrics are calculated for each array:

1. Average signal strength: the average percentile rank of untreated channel signal probes among the background (anti-genomic) probes.
2. Untreated channel signal standard deviation. The array is divided into a series of rectangular blocks and the average signal level calculated for each. Since probes are arranged randomly on the array there should be no large differences between blocks. Arrays with spatial artifacts have a larg standard deviation between blocks.

3. Methyl-depleted channel signal standard deviation.

5 Percentage methylation estimates and differentially methylated regions (DMRs)

We now calculate probe-level percentage methylation estimates for each sample. As a first step we need to identify a suitable set of unmethylated control probes from CpG-free regions to be used in normalization.

```
R> library(BSgenome.Hsapiens.UCSC.hg18)
R> ctrlIdx <- getControlIndex(rawData, subject = Hsapiens)
```

The minimal code required to estimate methylation would be `p <- methp(rawData, controlIndex=ctrlIdx)`. However, it is often useful to get `methp` to produce a series of diagnostic density plots to help identify non-hybridization quality issues. The `plotDensity` option specifies the name of the output pdf file, and the optional `plotDensityGroups` can be used to give groups different colors.

```
R> grp <- pData(rawData)$tissue
R> p <- methp(rawData, controlIndex = ctrlIdx, plotDensity = "density.pdf",
     plotDensityGroups = grp)
R> head(p)

        136421    136600   3788602   3822402   5739902
[1,] 0.2275917 0.3906297 0.3882923 0.5602930 0.3249610
[2,] 0.7972779 0.6879132 0.3545020 0.8658204 0.5255016
[3,] 0.1407686 0.1242173 0.2402985 0.2084232 0.3170556
[4,] 0.5812578 0.4807771 0.4824644 0.4907886 0.3605107
[5,] 0.5917244 0.5278607 0.4184537 0.4365131 0.3618830
[6,] 0.6539403 0.7561395 0.7517695 0.7149516 0.8622067
       5875602
[1,] 0.3003832
[2,] 0.8781021
[3,] 0.6940051
[4,] 0.4645142
[5,] 0.3728446
[6,] 0.8177507
```

The density plots are shown in Appendix B.

We can now identify differentially methylated regions using `dmrFinder`:

```
R> dmr <- dmrFinder(rawData, p = p, groups = grp,
     compare = c("colon", "liver", "colon", "spleen"))

R> names(dmr)
```

```
 [1] "tabs"    "p"       "l"       "chr"     "pos"
 [6] "pns"     "index"   "gm"      "groups"  "args"
[11] "comps"   "package"
```

```
R> names(dmr$tabs)
```

```
[1] "colon-liver"  "colon-spleen"
```

```
R> head(dmr$tabs[[1]])
```

```
       chr    start      end        p1        p2
319  chr12 88272817 88273844 0.8478776 0.2011810
358  chr13 27090247 27091263 0.7787445 0.1846656
1754  chr6 52637747 52638747 0.7124013 0.1884187
1120 chr20 60187423 60188227 0.8263859 0.2014856
473  chr15 58673117 58673819 0.8292100 0.3127552
133  chr11 14620645 14621065 0.8473120 0.3551344
                   regionName indexStart indexEnd nprobes
319  chr12:88266873-88274292      40465    40489      25
358  chr13:27090144-27095500      45272    45291      20
1754  chr6:52635302-52638967     160819   160843      25
1120 chr20:60143957-60188418     122600   122623      24
473  chr15:58669815-58674073      57658    57677      20
133  chr11:14620645-14623686      28438    28450      13
          area   ttarea      diff   maxdiff
319  16.167416 784.0852 0.6466966 0.7595101
358  11.881579 722.8358 0.5940789 0.7355009
1754 13.099564 647.1421 0.5239826 0.6613820
1120 14.997607 526.4294 0.6249003 0.8346198
473  10.329097 520.4598 0.5164549 0.6566695
133   6.398309 464.0463 0.4921776 0.6397410
```

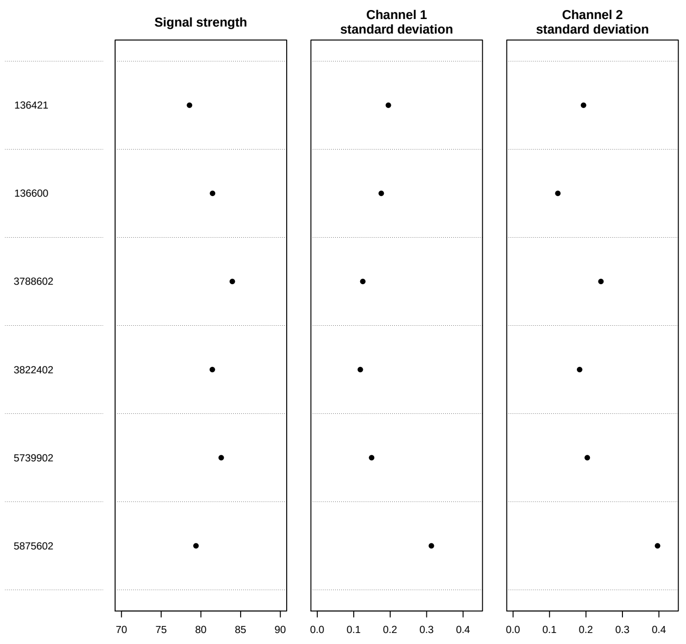
When called without the `compare` option, `dmrFinder` performs all pairwise comparisons between the groups.

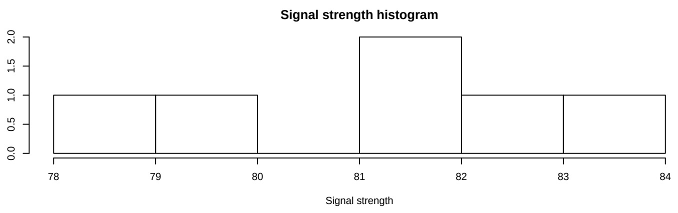
References

[1] Martin J. Aryee, Zhijin Wu, Christine Ladd-Acosta, Brian Herb, Andrew P. Feinberg, Srinivasan Yegnasubramanian, and Rafael A. Irizarry. Accurate genome-scale percentage dna methylation estimates from microarray data. *Biostatistics*, 12(2):197–210, 2011.

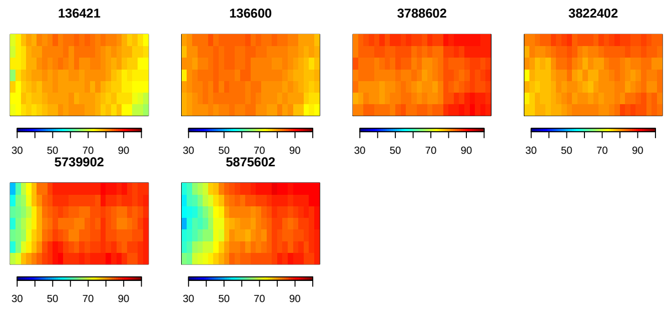
[2] Irizarry et al. Comprehensive high-throughput arrays for relative methylation (charm). *Genome Research*, 18(5):780–790, 2008.

6 Appendix A: Quality report

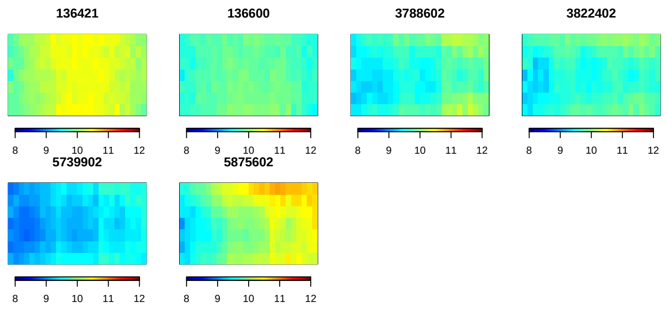

Signal strength
Channel 1 standard deviation
Channel 2 standard deviation
136421
136600
3788602
3822402
5739902
5875602
70
75
80
85
90
0.0
0.1
0.2
0.3
0.4


Signal strength histogram
0.0
0.5
1.0
1.5
2.0
78
79
80
81
82
83
84
Signal strength

Untreated Channel: PM probe quality

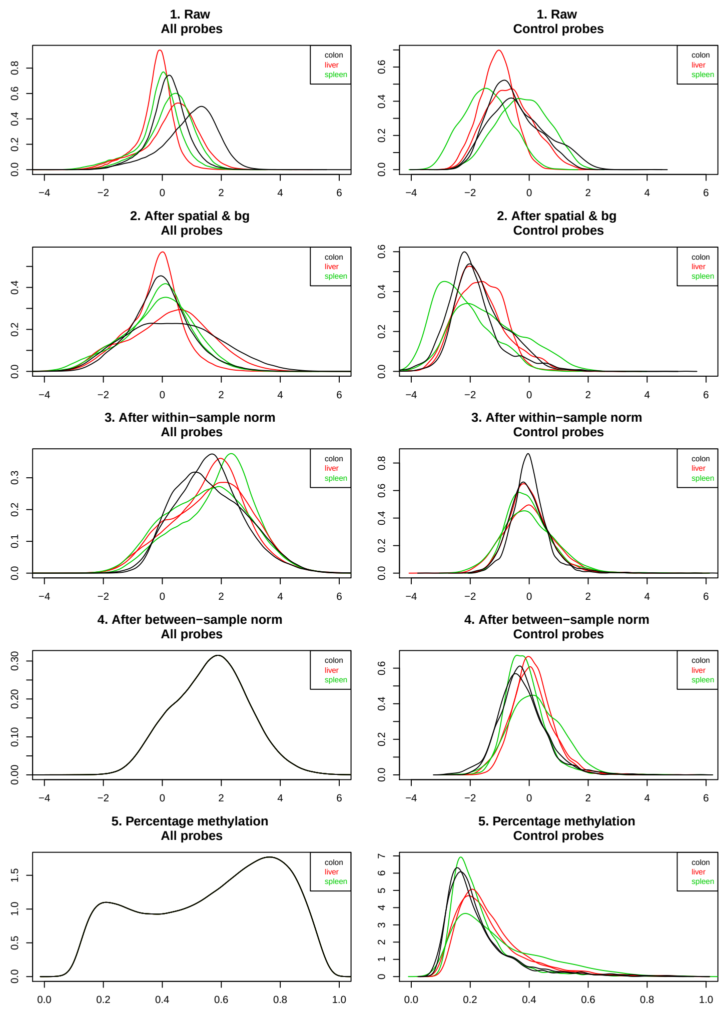


Enriched Channel: PM signal intensity



7 Appendix B: Density plots

Each row corresponds to one stage of the normalization process (Raw data, After spatial and background correction, after within-sample normalization, after between-sample normalization, percentage methylation estimates). The left column shows all probes, while the right column shows control probes.


1. Raw
All probes
1. Raw
Control probes
2. After spatial & bg
All probes
2. After spatial & bg
Control probes
3. After within−sample norm
All probes
3. After within−sample norm
Control probes
4. After between−sample norm
All probes
4. After between−sample norm
Control probes
5. Percentage methylation
All probes
5. Percentage methylation
Control probes
colon
liver
spleen

8 Details

This document was written using:

```
R> sessionInfo()
```

```
R version 2.13.0 (2011-04-13)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=C              LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets
[6] methods   base

other attached packages:
 [1] BSgenome.Hsapiens.UCSC.hg18_1.3.17
 [2] BSgenome_1.20.0
 [3] Biostrings_2.20.0
 [4] GenomicRanges_1.4.0
 [5] IRanges_1.10.0
 [6] charmData_0.99.3
 [7] pd.charm.hg18.example_0.99.2
 [8] oligo_1.16.0
 [9] preprocessCore_1.14.0
[10] oligoClasses_1.14.0
[11] RSQLite_0.9-4
[12] DBI_0.2-5
[13] charm_1.4.0
[14] genefilter_1.34.0
[15] RColorBrewer_1.0-2
[16] fields_6.3
[17] spam_0.23-0
[18] SQN_1.0
[19] nor1mix_1.1-2
[20] mclust_3.4.8
[21] Biobase_2.12.0

loaded via a namespace (and not attached):
 [1] AnnotationDbi_1.14.0 MASS_7.3-12
 [3] affxparser_1.24.0    affyio_1.20.0
 [5] annotate_1.30.0      bit_1.1-6
 [7] ff_2.2-1             gtools_2.6.2
 [9] multtest_2.8.0       siggenes_1.26.0
[11] splines_2.13.0       survival_2.36-5
[13] tools_2.13.0         xtable_1.5-6
```