MLML2R: Maximum Likelihood Estimates for 5-mC and 5-hmC Levels in the DNA

Samara F. Kiihl and Maria Tellez-Plaza

2017-12-07

Package version: MLML2R

Contents

1 Introduction

We present a guide to the Bioconductor package *MLML2R*. The package provides computational efficient maximum likelihood estimates of DNA methylation and hydroxymethylation proportions when data from the DNA processing methods bisulfite conversion (BS), oxidative bisulfite conversion (ox-BS), and Tet-assisted bisulfite conversion (TAB) are available. Estimates can be obtained when data from all the three methods are available or when any combination of only two of them are available. The package does not depend on other *R* packages, allowing the user to read and preprocess the data in any software and import the results into *R* in matrix format, obtain the estimates and use that as input in the other packages for genomic analysis, such as *minfi*, *sva* and *limma*.

In a given CpG site from a single cell we will either have a C or a T after DNA processing conversion methods, with a different interpretation for each of the available methods. This is a binary outcome and we assume a Binomial model and use the maximum likelihood estimation method to obtain the estimates for hydroxymethylation and methylation proportions.

T reads are referred to as converted cytosine and C reads are referred to as unconverted cytosine. Conventionally, T counts are also referred to as unmethylated counts, and C counts as methylated counts. In case of Infinium Methylation arrays, we have intensities representing the methylated (M) and unmethylated (U) channels that are proportional to the number of unconverted and converted cytosines (C and T, respectively). The most used summary from these experiments is the proportion $\beta = \frac{M}{M+U}$, commonly referred to as *beta-value*, which reflects the methylation level at a CpG site. Naïvely using the difference between betas from BS and oxBS as an estimate of 5-hmC (hydroxymethylated cytosine), and the difference between betas from BS and TAB as an estimate of 5-mC (methylated cytosine) can many times provide negative proportions and instances where the sum of 5-C (unmodified cytosine), 5-mC and 5-hmC proportions is greater than one due to measurement errors.

MLML2R package allows the user to jointly estimate hydroxymethylation and methylation consistently and efficiently.

The function `MLML` takes as input the data from the different methods and returns the estimated proportion of methylation, hydroxymethylation and unmethylation for a given CpG site. Table **??** presents the arguments of the `MLML` and Table **??** lists the results returned by the function.

The function assumes that the order of the rows and columns in the input matrices are consistent. In addition, all the input matrices must have the same dimension. Usually, rows represent CpG loci and columns are the samples.

2 Worked examples

2.1 Simulated data

MLML2R includes small example datasets for illustration.

We simulated counts from Binomial model with true proportions of methylation, hydroxymethylation and unmethylated being 0.3, 0.2, and 0.5, respectively. For instance, `MethylatedBS_sim` is a matrix of simulated counts from BS corresponding to 100 CpGs and 2 samples. Similarly we simulated matrices with methylated and unmethylated counts for all the three methods: BS, oxBS and TAB. The rows and columns in the input matrices are consistent.

2.1.1 BS and oxBS methods

Load the package:

```
library(MLML2R)
```

When only two methods are available, the default option returns the exact constrained maximum likelihood estimates using the the pool-adjacent-violators algorithm (PAVA) (Ayer et al. 1955).

```
results_exactBO1 <- MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim)
```

Maximum likelihood estimate via EM-algorithm approach (Qu et al. 2013) is obtained with the option `iterative=TRUE`. In this case, the default (or user specified) `tol` is considered in the iterative method.

```
results_emBO1 <- MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,iterative=TRUE)
```

When only two methods are available, we highly recommend the default option `iterative=FALSE` since the difference in the estimates obtained via EM and exact constrained is very small, but the former requires more computational effort:

```
all.equal(results_emBO1$hmC,results_exactBO1$hmC,scale=1)
## [1] "Mean absolute difference: 6.441136e-07"
```

```
library(microbenchmark)
mbmBO1 = microbenchmark(
   EXACT = MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
                L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim),
   EM =    MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
                L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
                iterative=TRUE),
   times=10)
mbmBO1
## Unit: microseconds
##   expr     min      lq      mean   median       uq      max neval
##  EXACT  66.589  76.501   85.7056   82.284   93.886  117.032    10
##     EM 938.172 980.759 1407.3872 1268.849 1463.352 3259.883    10
```

2.1.2 BS and TAB methods

Using PAVA:

```
results_exactBT1 <- MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim)
```

Using EM-algorithm:

```
results_emBT1 <- MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim,iterative=TRUE)
```

```
all.equal(results_emBT1$hmC,results_exactBT1$hmC,scale=1)
## [1] "Mean absolute difference: 8.860707e-07"
```

```
mbmBT1 = microbenchmark(
   EXACT = MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
                G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim),
   EM =    MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
                G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim,
                iterative=TRUE),
   times=10)
mbmBT1
## Unit: microseconds
##   expr     min      lq      mean    median       uq      max neval
##  EXACT  64.149  67.319   83.2996   75.2715   92.850  135.557    10
##     EM 760.941 783.400 1239.5642 1224.0155 1313.632 2634.893    10
```

2.1.3 oxBS and TAB methods

Using PAVA:

```
results_exactOT1 <- MLML(L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim)
```

Using EM-algorithm:

```
results_emOT1 <- MLML(L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim,iterative=TRUE)
```

```
all.equal(results_emOT1$hmC,results_exactOT1$hmC,scale=1)
## [1] "Mean absolute difference: 2.302158e-05"
```

```
mbmOT1 = microbenchmark(
   EXACT = MLML(L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
                G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim),
   EM =    MLML(L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
                G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim,
                iterative=TRUE),
   times=10)
mbmOT1
## Unit: microseconds
##   expr     min     lq     mean   median      uq     max neval
##  EXACT  61.802  64.92  85.9865  72.3535  94.192 149.805    10
##     EM 223.122 224.08 247.2558 242.6950 253.269 316.616    10
```

2.1.4 BS, oxBS and TAB methods

When data from the three methods are available, the default otion in the MLML function returns the constrained maximum likelihood estimates using an approximated solution for Lagrange multipliers method.

```
results_exactBOT1 <- MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim)
```

Maximum likelihood estimate via EM-algorithm approach (Qu et al. 2013) is obtained with the option iterative=TRUE. In this case, the default (or user specified) tol is considered in the iterative method.

```
results_emBOT1 <- MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim,iterative=TRUE)
```

We recommend the default option iterative=FALSE since the difference in the estimates obtained via EM and the approximate exact constrained is very small, but the former requires more computational effort:

```
all.equal(results_emBOT1$hmC,results_exactBOT1$hmC,scale=1)
## [1] "Mean absolute difference: 1.384806e-06"
```

```
mbmBOT1 = microbenchmark(
   EXACT = MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
                L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
                G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim),
   EM =    MLML(T.matrix = MethylatedBS_sim , U.matrix = UnMethylatedBS_sim,
                L.matrix = UnMethylatedOxBS_sim, M.matrix = MethylatedOxBS_sim,
                G.matrix = UnMethylatedTAB_sim, H.matrix = MethylatedTAB_sim,
                iterative=TRUE),
   times=10)
mbmBOT1
## Unit: microseconds
##   expr     min      lq    mean  median      uq     max neval
##  EXACT  74.864  90.267  94.070  92.983  95.480 122.419    10
##     EM 410.715 453.372 508.125 515.237 545.042 629.934    10
```

2.2 Publicly available data: GSE63179

We will use the dataset from Field (2015), which consists of eight DNA samples from the same DNA source treated with oxBS-BS and hybridized to the Infinium 450K array.

When data is obtained through Infinium Methylation arrays, we recommend the use of the *minfi* package, a well-established tool for reading, preprocessing and analysing DNA methylation data from these platforms. Although our example relies on *minfi* and other *Bioconductor* tools, *MLML2R* does not depend on any packages. Thus, the user is free to read and preprocess the data using any software of preference and then import the intensities (or T and C counts) for the methylated and unmethylated channel (or converted and uncoverted cytosines) into *R* in matrix format.

To start this example we will need the following packages:

```
library(minfi)
library(GEOquery)
```

It is usually best practice to start the analysis from the raw data, which in the case of the 450K array is a .IDAT file.

The raw files are deposited in GEO and can be downloaded by using the getGEOSuppFiles. There are two files for each replicate, since the 450k array is a two-color array. The .IDAT files are downloaded in compressed format and need to be uncompressed before they are read by the read.metharray.exp function.

```
getGEOSuppFiles("GSE63179")
untar("GSE63179/GSE63179_RAW.tar", exdir = "GSE63179/idat")

list.files("GSE63179/idat", pattern = "idat")
files <- list.files("GSE63179/idat", pattern = "idat.gz$", full = TRUE)
sapply(files, gunzip, overwrite = TRUE)
```

The `.IDAT` files can now be read:

```
rgSet <- read.metharray.exp("GSE63179/idat")
```

To access phenotype data we use the `pData` function. The phenotype data is not yet available from the `rgSet`.

```
pData(rgSet)
```

In this example the phenotype is not really relevant, since we have only one sample: male, 25 years old. What we do need is the information about the conversion method used in each replicate: BS or oxBS. We will access this information automatically from GEO:

```
geoMat <- getGEO("GSE63179")
pD.all <- pData(geoMat[[1]])
pD <- pD.all[, c("title", "geo_accession", "characteristics_ch1.1",
                 "characteristics_ch1.2","characteristics_ch1.3")]
pD
```

This phenotype data needs to be merged into the methylation data. The following commands guarantee we have the same replicate identifier in both datasets before merging.

```
sampleNames(rgSet) <- sapply(sampleNames(rgSet),function(x)
  strsplit(x,"_")[[1]][1])
rownames(pD) <- pD$geo_accession
pD <- pD[sampleNames(rgSet),]
pData(rgSet) <- as(pD,"DataFrame")
rgSet
```

The `rgSet` object is a class called *RGChannelSet* used for two color data (green and a red channel). The input in the `MLML` funcion is *MethylSet*, which contains the methylated and unmethylated signals. The most basic way to construct a *MethylSet* is using the function `preprocessRaw`. Here we chose the function `preprocessNoob` for background correction and construction of the *MethylSet*.

```
MSet.noob<- preprocessNoob(rgSet)
```

After the preprocessed steps we can use `MLML` from the *MLML2R* package.

The BS replicates are in columns 1, 3, 5, and 6. The remaining columns are from the oxBS treated replicates.

```
MethylatedBS <- getMeth(MSet.noob)[,c(1,3,5,6)]
UnMethylatedBS <- getUnmeth(MSet.noob)[,c(1,3,5,6)]
MethylatedOxBS <- getMeth(MSet.noob)[,c(7,8,2,4)]
UnMethylatedOxBS <- getUnmeth(MSet.noob)[,c(7,8,2,4)]
```

In this example we only have two methods, therefore we can choose between the EM-algorithm and the exact constrained maximum likelihood estimates (using PAVA).

Estimates via the EM-algorithm:

```
results_emPD1 <- MLML(T.matrix = MethylatedBS , U.matrix = UnMethylatedBS,
                      L.matrix = UnMethylatedOxBS, M.matrix = MethylatedOxBS,
                      iterative = TRUE)
```

The exact constrained MLE (using PAVA):

```
results_exactPD1 <- MLML(T.matrix = MethylatedBS , U.matrix = UnMethylatedBS,
                         L.matrix = UnMethylatedOxBS, M.matrix = MethylatedOxBS)
```

2.3 Publicly available data: GSE73895

We will use the dataset from Johnson et al. (2016), which consists of 30 DNA samples from glioblastoma tumors treated with oxBS-BS and hybridized to the Infinium 450K array.

The raw files are deposited in GEO and can be downloaded and read into *R* by doing:

```
getGEOSuppFiles("GSE73895")
untar("GSE73895/GSE73895_RAW.tar", exdir = "GSE73895/idat")

idatFiles <- list.files("GSE73895/idat", pattern = "idat.gz$", full = TRUE)
sapply(idatFiles, gunzip, overwrite = TRUE)

rgSet <- read.metharray.exp("GSE73895/idat")
```

We need to identify the samples from different methods: BS-conversion, oxBS-conversion. We obtain this information from GEO:

```
geoMat <- getGEO("GSE73895")
pD.all <- pData(geoMat[[1]])
pD <- pD.all[, c("title", "geo_accession", "characteristics_ch1",
                 "characteristics_ch1.2","characteristics_ch1.3")]
head(pD)
```

Keeping only some of the variables from phenotype data:

```
names(pD)[c(1,3,4,5)] <- c("method","gender","survival_months","age_years")
pD$gender <- sub("^gender: ", "", pD$gender)
pD$age_years <- as.numeric(sub("^subject age: ", "", pD$age_years))
pD$survival_months <- as.numeric(sapply(pD$survival_months, function(x)
  strsplit(as.character(x),":")[[1]][2]))
pD$method <- sapply(pD$method, function(x) strsplit(as.character(x),"_")[[1]][3])
```

We now need to merge this pheno data into the methylation data. The following are commands to make sure we have the same row identifier in both datasets before merging.

```
sampleNames(rgSet) <- sapply(sampleNames(rgSet),function(x)
  strsplit(x,"_")[[1]][1])
rownames(pD) <- pD$geo_accession
pD <- pD[sampleNames(rgSet),]
pData(rgSet) <- as(pD,"DataFrame")
rgSet
```

Preprocessing and preparing input matrices for `MLML`:

```
MSet.noob<- preprocessNoob(rgSet)

BS_index <- which(pData(rgSet)$method=="BS")
oxBS_index <- which(pData(rgSet)$method=="oxBS")

MethylatedBS <- getMeth(MSet.noob)[,BS_index]
UnMethylatedBS <- getUnmeth(MSet.noob)[,BS_index]
```

```
MethylatedOxBS <- getMeth(MSet.noob)[,oxBS_index]
UnMethylatedOxBS <- getUnmeth(MSet.noob)[,oxBS_index]
```

In this example we only have two methods, therefore we can choose between the EM-algorithm and the exact constrained maximum likelihood estimates (using PAVA).

Estimates via the EM-algorithm:

```
results_emPD2 <- MLML(T.matrix = MethylatedBS , U.matrix = UnMethylatedBS,
                   L.matrix = UnMethylatedOxBS, M.matrix = MethylatedOxBS,
                   iterative = TRUE)
```

The exact constrained MLE (using PAVA):

```
results_exactPD2 <- MLML(T.matrix = MethylatedBS , U.matrix = UnMethylatedBS,
                      L.matrix = UnMethylatedOxBS, M.matrix = MethylatedOxBS)
```

3 Styles

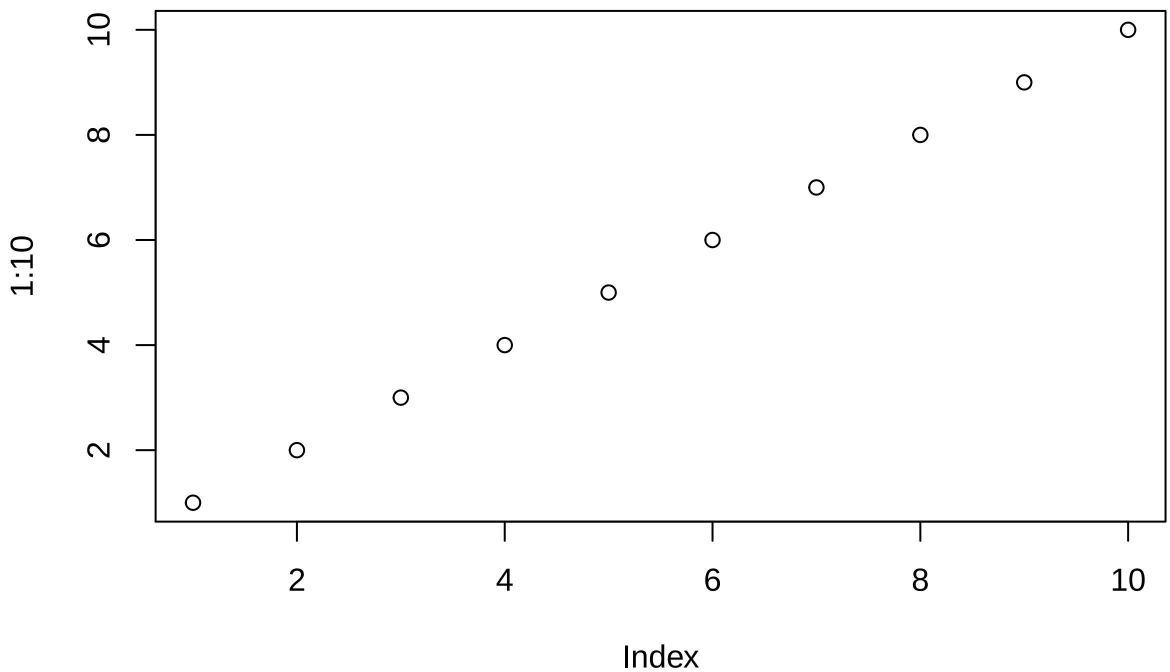
The `html_vignette` template includes a basic CSS theme. To override this theme you can specify your own CSS in the document metadata as follows:

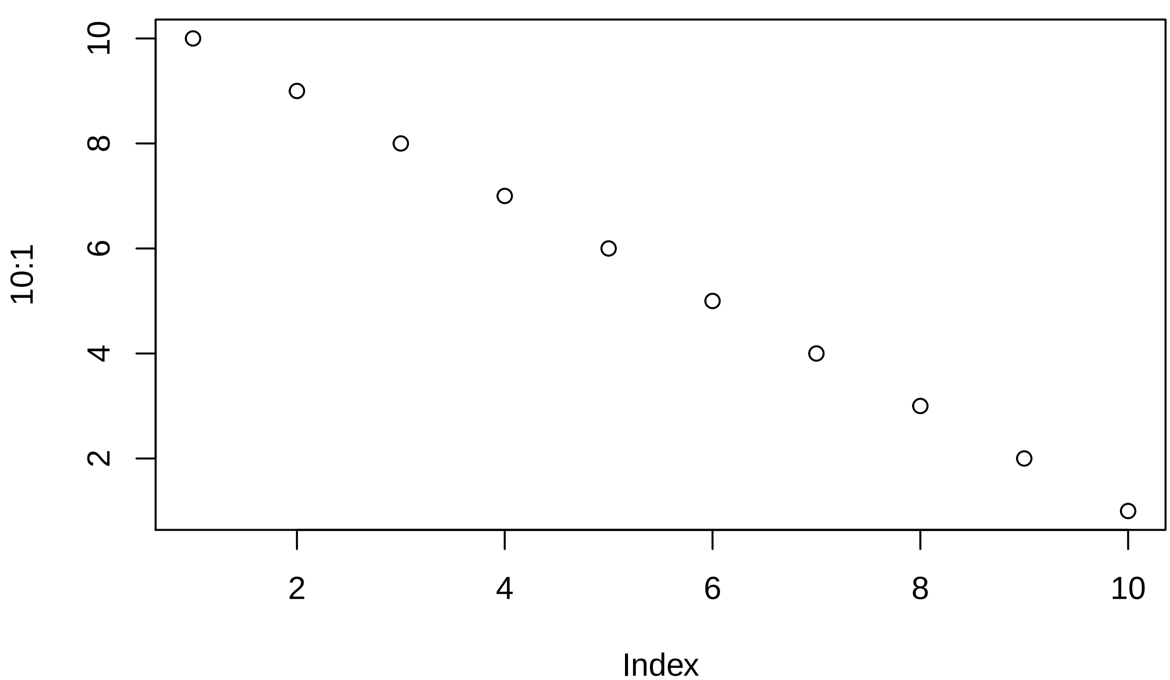
```
output:
  rmarkdown::html_vignette:
    css: mystyles.css
```

3.1 Figures

The figure sizes have been customised so that you can easily put two images side-by-side.

```
plot(1:10)
plot(10:1)
```


1:10
10
8
6
4
2
2
4
6
8
10
Index


10:1
10
8
6
4
2
2
4
6
8
10
Index

You can enable figure captions by `fig_caption: yes` in YAML:

```
output:
  rmarkdown::html_vignette:
    fig_caption: yes
```

Then you can use the chunk option `fig.cap = "Your figure caption."` in **knitr**.

3.2 More Examples

You can write math expressions, e.g. $Y = X\beta + \epsilon$, footnotes[1], and tables, e.g. using `knitr::kable()`.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Also a quote using >:

> "He who gives up [code] safety for [code] speed deserves neither." (via)

References

Ayer, Miriam, H. D. Brunk, G. M. Ewing, W. T. Reid, and Edward Silverman. 1955. "An Empirical Distribution Function for Sampling with Incomplete Information." *Ann. Math. Statist.* 26 (4). The Institute of Mathematical Statistics: 641–47. doi:10.1214/aoms/1177728423.

Field, Dario AND Bachman, Sarah F. AND Beraldi. 2015. "Accurate Measurement of 5-Methylcytosine and 5-Hydroxymethylcytosine in Human Cerebellum Dna by Oxidative Bisulfite on an Array (Oxbs-Array)." *PLOS ONE* 10 (2). Public Library of Science: 1–12. doi:10.1371/journal.pone.0118202.

Johnson, K. C., E. A. Houseman, J. E. King, K. M. von Herrmann, C. E. Fadul, and B. C. Christensen. 2016. "5-Hydroxymethylcytosine localizes to enhancer elements and is associated with survival in glioblastoma patients." *Nat Commun* 7 (November): 13177.

Qu, Jianghan, Meng Zhou, Qiang Song, Elizabeth E. Hong, and Andrew D. Smith. 2013. "MLML: Consistent Simultaneous Estimates of Dna Methylation and Hydroxymethylation." *Bioinformatics* 29 (20): 2645–6. doi:10.1093/bioinformatics/btt459.

[1]A footnote here.