MEDME overview

Mattia Pelizzola and Annette Molinaro

May 2, 2019

1 Introduction

In this document a brief tutorial on the **Model for Experimental Data with MeDIP Enrichment** (MEDME; Pelizzola et al, Genome Research 2008) is provided. This library allows the determination of absolute and relative DNA methylation levels from MeDIP enrichment data. Briefly, MeDIP is a technique developed by Weber et al. (2005) where an antibody specific for mCpGs (CG dinucleotides containing 5-methyl Cytosine) is used to immuno-precipitate genomic methylated fragments. Precipitated fragments are hybridized on one microarray channel while on the other channel the input DNA is hybridized. The resulting logRatio is indicative of the enrichment level, i.e. the DNA methylation level. We have shown how MeDIP enrichment is not linearly correlated to the methylation level. Rather, it can be modeled as a sigmoidal function of the log2(mCpG). The methodology that we developed requires one to perform a calibration experiment where fully-methylated DNA is generated. MeDIP protocol is then applied to this sample followed by microarray hybridization. A logistic model is then fit on the MeDIP enrichment of the fully-methylated sample. The obtained model can then be applied on the experimental data in order to predict both absolute and relative methylation levels from MeDIP enrichment measures.

The model should be applied on calibration data containing MeDIP enrichment of fully methylated DNA, most likely artificially generated. Nevertheless, in case chromosome or genome-wide human tiling arrays are used a regular sample could be used too. In fact, human genomic DNA is known to be hypermethylated but in the promoter regions. Of course the performance of the method is expected to be somehow affected by this approximation. Rather, if the array probes are specific or highly enriched for promoter regions, the model should not be fitted directly on the experimental data.

2 Quick overview

Data may be loaded into R from GFF or SGR files (GFF with probe names is recommended) using the *MEDME.readFiles* function. Usually tiling microarrays are used for this kind of experiments. Genomic position and chromosome need to be provided for each microarray probe. As a first step the *smooth* function can be applied to smooth the MeDIP enrichment data. This is usually necessary to improve the quality of tiling array data. In order to fit the model on the fully methylated dataset one needs to know the expected methylation level. Since

every CpG is methylated in this sample, the CpG content in the region around each probe can be used to determine its expected DNA methylation level. The probe-level CpG content can be determined using the *CGcount* function. The logistic model can be then fitted on the MeDIP smoothed data of the fully-methylated sample using the function *MEDME*.

Finally the model can be applied on the experimental MeDIP data, using the function *MEDME.predict*. This allows the determination of experimental MeDIP smoothed data as well as absolute and relative methylation scores (AMS and RMS respectively). Final results may be exported to GFF or SGR files using the *MEDME.writeFiles* function. This can be useful to visualize the results with genome browser softwares.

3 Step by step

First of all the `MEDME` library and the `test dataset` are loaded:

```
> library(MEDME)
> data(testMEDMEset)
```

The dataset is an object of class *MEDMEset*. This class contains slots to store logR, smoothed, AMS and RMS data. Moreover probe level chromosome assignment as well as genomic position and CGcount information are available too. Some of these slot can be empty at the beginning of the analysis and will be filled by the respective methods (type *class?MEDMEset* at the R prompt for more information).

The test dataset contains calibration as well as experimental data for almost 50000 probes tiled on the human chrX. See the references for the complete dataset and additional details. The logR slot of the test dataset contains the MeDIP enrichment of the fully-methylated sample in the first column. The last two columns contain the MeDIP enrichment of two cell strains. NBMEL are cells from newborn melanocytes, while YUSAC2 is a melanoma strain.

```
> logR(testMEDMEset[1:5,])
```

```
          fullyMet NBMEL YUSAC2
ldX_86705     0.42  0.17   0.54
ldX_86706     0.76 -0.61  -0.14
ldX_86707     0.67  0.17   0.73
ldX_86708     0.29  0.25   0.81
ldX_86709    -0.06  0.53   0.73
```

This dataset contains within and between array normalized data. Nonetheless, this data has not been smoothed. Before applying the model on the fully-methylated data it is more convenient to apply some `smoothing on the data`.

```
> testMEDMEset = smooth(data = testMEDMEset, wsize = 1000, wFunction = 'linear')
```

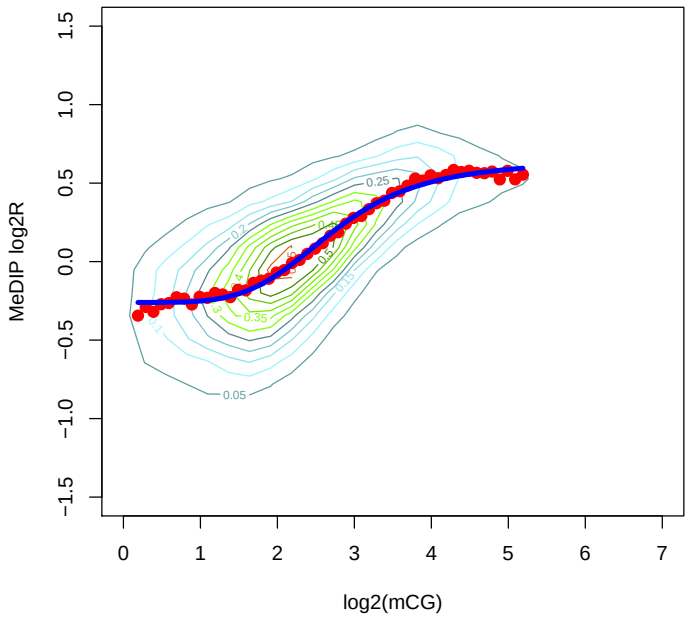
One could optionally try different window sizes and/or weighting functions. The smoothed data are automatically saved in the smoothed slot of the MEDMEset object. As anticipated briefly above, before fitting the model it is necessary to determine the expected methylation level for each probe. This is performed

by determining the probe-level CpG content. The metadata library with the genomic sequence should be already installed and loaded into R. Only human (hsa) and mouse (mmu) genomes are currently supported in this function. Please note that only the last genome release should be used. LiftOver UCSC tool could be used for batch conversion of old genomic position to the last genome release.

```
> library(BSgenome.Hsapiens.UCSC.hg18)
> testMEDMEset = CGcount(data = testMEDMEset)
```

Again, the probe level CpG content is automatically saved on the CGcount slot of the MEDMEset object. Now it is possible to fit the logistic model on the MeDIP enrichment of the fully methylated sample, to determine the relationship between MeDIP enrichment and the expected methylation level. This should be repeated each time the protocol or the microarray platform or design are modified. In case there are many samples on the logR slot of the input object, the sample argument needs to be set to identify the one used for fitting.

```
> MEDMEmodel = MEDME(data = testMEDMEset, sample = 1)
```



Finally, the model can be applied on the experimental data in order to predict probe-level AMS and RMS.

```
> testMEDMEset = MEDME.predict(data = testMEDMEset, MEDMEfit = MEDMEmodel, MEDMEextremes =
```

AMS and RMS have been automatically saved in the AMS and RMS lot of the MEDMEset object. All the generated data can be retrieved using the following methods:

```
> smoothed(testMEDMEset)[1:3,]

           fullyMet        NBMEL    YUSAC2
ldX_86705 0.5438935 -0.039809886 0.4059924
ldX_86706 0.5251526 -0.005187793 0.4582923
ldX_86707 0.4902472  0.122608476 0.5907770

> AMS(testMEDMEset)[1:3,]

          fullyMet    NBMEL   YUSAC2
ldX_86705 20.19321 4.361313 10.70721
ldX_86706 17.71882 4.655687 12.78526
ldX_86707 14.65130 5.838532 32.00000

> RMS(testMEDMEset)[1:3,]

          fullyMet     NBMEL   YUSAC2
ldX_86705 1.740101 0.8054648 2.751128
ldX_86706 1.653976 0.9976744 2.806456
ldX_86707 1.603981 1.3183230 2.817586
```

SGR or GFF files can finally be exported for their visualization on genome browser softwares using the *MEDME.writeFiles* function.

Please note that this oveerview has only been performed on a `small subset of the full dataset`. Hence the model fitting and quality of the resulting data could not be totally satisfactory. See the references and the help for each individual function for additional details.