BPRMeth: Higher order methylation features for clustering and prediction in epigenomic studies

ChantrioInt-Andreas Kapourani*

Modified: 1 August, 2016. Compiled: October 29, 2016

Contents

1 Introduction

DNA methylation is an intensely studied epigenetic mark, yet its functional role is incompletely understood. Attempts to quantitatively associate average DNA methylation to gene expression yield poor correlations outside of the well-understood methylation-switch at CpG islands.

Here we use probabilistic machine learning to extract higher order features associated with the methylation profile across a defined region. These features quantitate precisely notions of shape of a methylation profile, capturing spatial correlations in DNA methylation across genomic regions. Using these higher order features across promoter-proximal regions, we are able to construct a powerful machine learning predictor of gene expression.

2 Background

DNA methylation data produced by High-Throughput Sequencing (HTS) technology can be modelled with a Binomial distribution:

$$m \sim \mathcal{B}inom(t, p) \tag{1}$$

*C.A.Kapourani@ed.ac.uk or kapouranis.andreas@gmail.com

In practical studies we are interested in learning the methylation patterns of genomic regions which can be represented by an observation vector $\mathbf{y}$. Let $f(x) = \Phi\big(g(x)\big)$ be a latent function representing the methylation profiles and $g(x)$ be of the form:

$$g(x) = \sum_{j=0}^{M-1} w_j h_j(x) \tag{2}$$

where $h_j(\cdot)$ can be any basis function, e.g. Radial Basis Function (RBF), and w_j are the coefficients for each basis.

Given $f(x)$ the observations y_l for each CpG site are i.i.d. Binomial variables, so we can define the joint log-likelihood in factorised form:

$$\log p(\mathbf{y}|f) = \sum_{l=1}^{L} \log\Big(\mathcal{B}inom\big(m_l|t_l, \Phi(g(x_l))\big)\Big) \tag{3}$$

We refer to this observation model as the Binomial Probit Regression (BPR) likelihood function. *Figure 1* shows the process of learning the methylation profile for a specific promoter region using the BPR model. For a more detailed explanation of the statistical method, see [1].

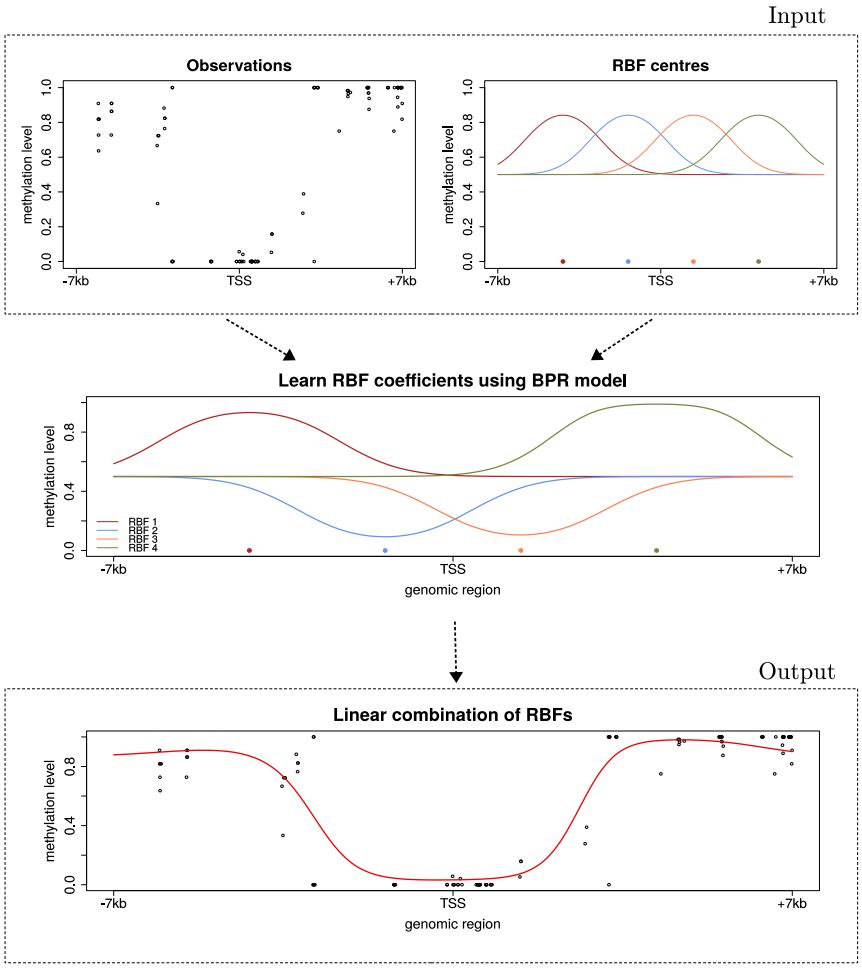

Figure 1: Illustration of the process for learning methylation profiles using the BPR model. The inputs to the model are the observed methylation levels of the CpGs across the promoter region, plus the number, with their corresponding centres, of the Radial Basis Functions (RBFs), which in this example are chosen to be four. Using this information the BPR model will learn the optimal coefficients for each RBF using maximum likelihood. Finally, we obtain the underlying methylation profile by a linear combination of the fitted RBFs. One should note that the higher the number of RBFs the better the resolution for the methylation profile.

3 Analysis Pipeline

3.1 Sample data

To illustrate the functions of the `BPRMeth` package we will use real datasets that are publicly available from the ENCODE project consortium [2]. More specifically we will focus on the K562 immortalized cell line, with GEO: GSE27584 for the RRBS data and GEO: GSE33480 for the RNA-Seq data. We will use directly the preprocessed files, however, we should note that we have converted the RNA-Seq data from `.gtf` to `.bed` format using the `bedops` tool (http://bedops.readthedocs.io). We have kept only the protein coding genes, and for the purpose of this vignette we focus only on `chr12` and `chr13`. Full details of where to download the data and how to preprocess them are included in the Supplementary Material of [1].

3.2 Import and read HTS files

Due to its general approach, the BPR method can be applied both in RRBS and WGBS methylation datasets, provided that we have information about the methylated and unmethylated reads in each CpG location.

The `BPRMeth` package provides methods for reading files generated from HTS experiments with specific formats. For the formats provided by the ENCODE datasets described above; we have implemented `process_haib_caltech_wrap`, which is a wrapper function for performing the preprocessing and obtaining the final objects for downstream analysis. The user can implement his own methods for reading files with different formats, provided that he can create an object similar to what is described below. First, we load and attach the package and then we obtain the paths for the sample RRBS and RNA-Seq files with the following commands:

```
library(BPRMeth)
rrbs_file <- system.file("extdata", "rrbs.bed", package = "BPRMeth")
rnaseq_file <- system.file("extdata", "rnaseq.bed", package = "BPRMeth")
```

Then, we process both files to obtain a `processHTS` object which will be used for downstream analysis:

```
# Preprocess both RRBS and RNA-Seq files
HTS_data <- process_haib_caltech_wrap(rrbs_file, rnaseq_file)
```

Among other information, the `processHTS` object contains the following important slots:

1. A `list` where each entry corresponds to a different promoter methylation region, accessible with **methyl_region**. More specifically, each methylation promoter region is an $L_i \times 3$ dimensional matrix, where L_i denotes the number of CpGs found in region i. The columns contain the following information:
   - 1st column: Contains the locations of CpGs relative to TSS. Note that the actual locations are scaled to the (-1, 1) region.
   - 2nd column: Contains the total reads of each CpG in the corresponding location.
   - 3rd column: Contains the methylated reads each CpG in the corresponding location.
2. A `vector` with the corresponding log2 transformed gene expression levels for each promoter region, accessible with **gex**.
3. A *GRanges* object storing the RNA-Seq data together with annotation information for each promoter region, accessible with **rna_data**.

Now, we give examples on how to access these information on the sample data described in the previous section. Initially, we can access the $16th$ promoter methylation region as follows:

```
HTS_data$methyl_region[[16]]
```

```
##              [,1] [,2] [,3]
##  [1,] -0.65185714    4    2
##  [2,]  0.02985714    4    0
##  [3,]  0.03128571    4    0
```

```
##  [4,]  0.03157143    4    0
##  [5,]  0.03285714    4    4
##  [6,]  0.03714286   38    0
##  [7,]  0.03942857   38    1
##  [8,]  0.03985714   38    0
##  [9,]  0.04014286   38    2
## [10,]  0.04128571   38    0
## [11,]  0.04185714   38    4
## [12,]  0.76028571    4    1
## [13,]  0.95685714   85   52
## [14,]  0.96028571   85   21
```

Below we show the log2 transformed gene expression levels for the first 10 promoter regions:

```
head(HTS_data$gex, 10)
```

```
##  [1] -3.321928  1.587480  1.792014  2.695450 -2.708765  6.715633  2.109909  1.618953
##  [9] -3.321928 -3.321928
```

Finally, the RNA-Seq data which are stored in a *GRanges* object, can be accessed as follows:

```
HTS_data$rna_data
```

```
## GRanges object with 334 ranges and 3 metadata columns:
##         seqnames                ranges strand |      ensembl_id   gene_name gene_fpkm
##            <Rle>             <IRanges>  <Rle> |     <character> <character> <numeric>
##     [1]    chr12     [ 569529,  672675]      + | ENSG00000139044    B4GALNT3 0.0000000
##     [2]    chr12     [ 752147,  755044]      + | ENSG00000177406  AC021054.1 2.9052400
##     [3]    chr12     [1021242, 1058888]      - | ENSG00000002016       RAD52 3.3629800
##     [4]    chr12     [1675158, 1703331]      - | ENSG00000171823      FBXL14 6.3775600
##     [5]    chr12     [1901122, 2027870]      - | ENSG00000151062    CACNA2D4 0.0529609
##     ...      ...                    ...    ... .             ...         ...       ...
##   [330]    chr13 [114303172, 114312501]      - | ENSG00000186009       ATP4B   0.00000
##   [331]    chr13 [114321469, 114438724]      + | ENSG00000185974        GRK1   0.00000
##   [332]    chr13 [114462215, 114514926]      + | ENSG00000184497      FAM70B   0.00000
##   [333]    chr13 [114523523, 114567046]      - | ENSG00000183087        GAS6   8.55437
##   [334]    chr13 [114747193, 114898095]      - | ENSG00000185989       RASA3  15.20510
##   -------
##   seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

The package provides implementation for specific parts of the preprocessing steps described above which can be seen by typing `process_haib_caltech_wrap` on the R console. If the user has files with different formats, he can implement his own functions for reading the data and then combine them with the simple functions present in the BPRMeth package, in order to obtain an object similar to `processHTS`, which should have the slots described above.

3.3 Predict gene expression

After preprocessing the HTS data, we have the following number of unique protein-coding genes belonging in chr12 and chr13:

```
# Obtain the number of gene promoters
length(HTS_data$gex)
```

```
## [1] 334
```

Learning the methylation profiles is equivalent to optimizing the model parameters $\mathbf{w}$ described in Eq. 2. These parameters can be considered as the extracted features which quantitate precisely notions of shape of a methylation profile.

3.3.1 Create basis objects

For each promoter region, we will learn its methylation profile using the BPR model with a specified number of Radial Basis Functions (RBFs) M. For a single input variable x, the RBF takes the form $h_j(x) = exp(-\gamma||x - \mu_j||^2)$, where μ_j denotes the location of the j^{th} basis function in the input space and γ controls the spatial scale. The case when $M = 0$ is equivalent to learning the average methylation level for the given region (i.e. learn a constant function).

For our running example, we will create two RBF objects, one with 9 basis functions and the other with 0 basis functions denoting the mean methylation level approach:

```
# Create basis object with 9 RBFs
basis_profile <- create_rbf_object(M = 9)

# Create basis object with 0 RBFs, i.e. constant function
basis_mean <- create_rbf_object(M = 0)
```

The `rbf` object contains information such as the centre locations μ_j and the value of the spatial scale parameter γ:

```
# Show the slots of the 'rbf' object
basis_profile
```

```
## $M
## [1] 9
##
## $mus
## [1] -0.8 -0.6 -0.4 -0.2  0.0  0.2  0.4  0.6  0.8
##
## $gamma
## [1] 20.25
##
## $eq_spaced_mus
## [1] TRUE
##
## $whole_region
## [1] TRUE
##
## attr(,"class")
## [1] "rbf"
```

The γ is computed by the number of basis M, however the user can tune it according to his liking. Except from RBF basis, the `BPRMeth` pacakge provides polynomial basis which can be created with the `create_polynomial_object` function.

3.3.2 Learn methylation profiles and make predictions

We can now optimize the BPR likelihood function and extract the features $\mathbf{w}_i$ for each promoter region. To quantitatively predict expression at each region, we construct a regression model by taking as input the higher-order methylation features learned from the BPR model. In addition to these features, we consider two supplementary sources of information: (1) the goodness of fit in RMSE and (2) the CpG density. For our analysis an SVM regression model is considered. We will use 70% of the data for training, and we will test the model's performance on the remaining 30%.

All the aforementioned steps are assembled in the `bpr_predict_wrap` wrapper function, which returns a `bpr_predict` object.

```
# Set seed for reproducible results
set.seed(1234)
```

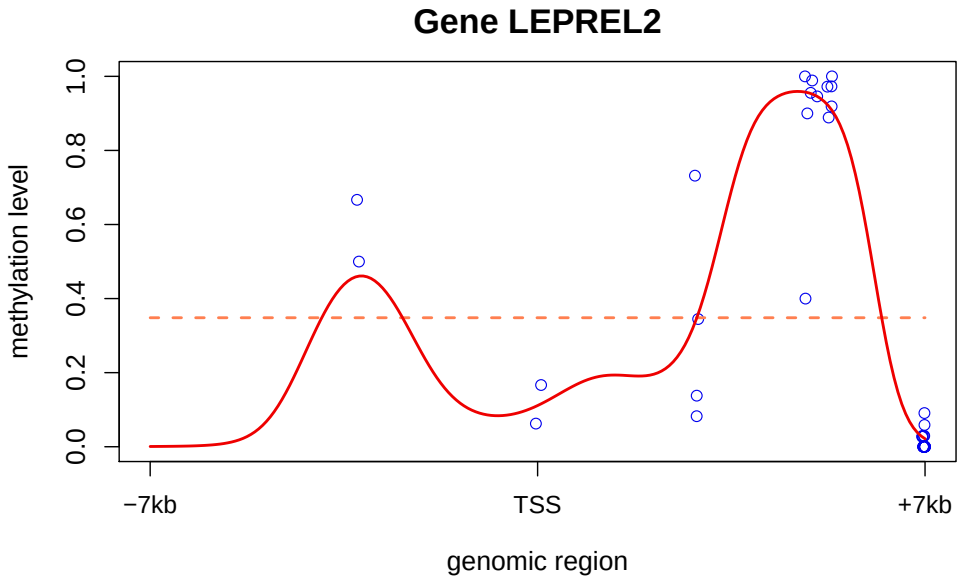

Figure 2: **Methylation pattern for the LEPREL2 gene over $\pm 7kb$ promoter region.** The points represent the DNA methylation level of each CpG site. The shape of the methylation profiles is captured by the red curve, whereas the orange dashed line denotes the mean methylation level.

```
# Perform predictions using methylation profiles
res_profile <- bpr_predict_wrap(x = HTS_data$methyl_region, y = HTS_data$gex,
                                basis = basis_profile, fit_feature = "RMSE",
                                cpg_dens_feat = TRUE, is_parallel = FALSE,
                                is_summary = FALSE)

# Perform predictions using mean methylation level
res_mean <- bpr_predict_wrap(x = HTS_data$methyl_region, y = HTS_data$gex,
                             basis = basis_mean, is_parallel = FALSE,
                             is_summary = FALSE)
```

We can now compare the Pearson's correlation coefficient r for both models and observe that the higher-order methylation features achieve test correlations twice as large as previously reported when using average methylation levels.

```
# Test errors for methylation profiles, PCC = Pearson's r
res_profile$test_errors$pcc
```

```
## [1] 0.7160181
```

```
# Test errors for mean methylation levels
res_mean$test_errors$pcc
```

```
## [1] 0.2953158
```

Figure 2 shows an example promoter region together with the fitted methylation profiles.

```
# Choose promoter region 21 -> i.e. LEPREL2 gene
gene_name <- HTS_data$rna_data$gene_name[21]
plot_fitted_profiles(region = 21, X = HTS_data$methyl_region, fit_prof = res_profile,
                     fit_mean = res_mean, title = paste0("Gene ", gene_name))
```

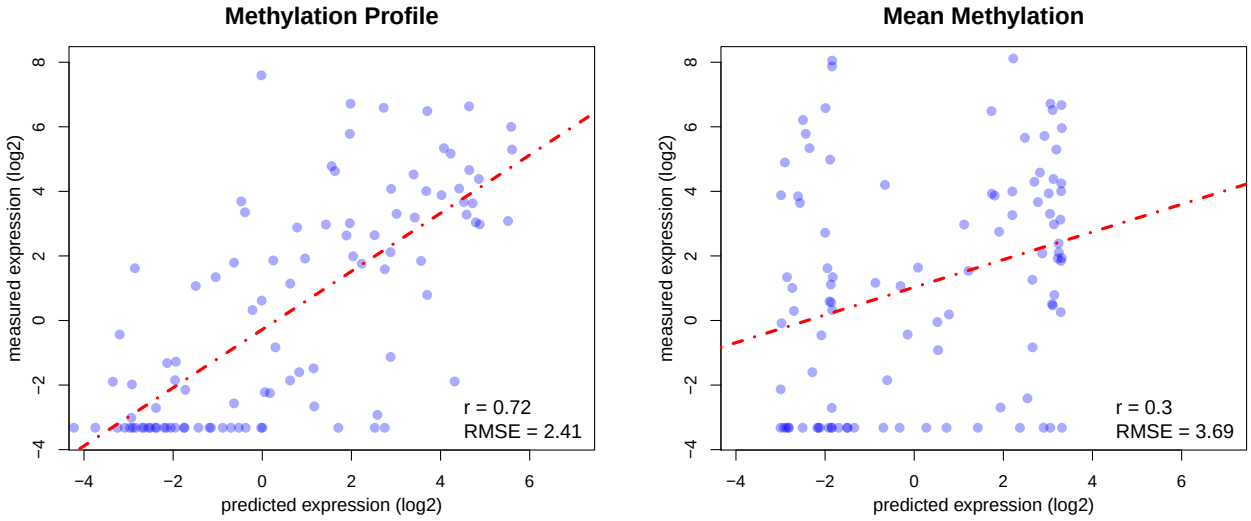

Figure 3: **Quantitative relationship between DNA methylation patterns and gene expression.** Scatter plots of predicted versus measured (log2-transformed) gene expression values: using the BPR model and extracting higher-order features *(left)*, and using the average methylation level *right* as input to the SVM regression model. Each shaded blue dot represents a different gene. The red dashed line indicates the linear fit between the predicted and measured expression values.

Figure 3 shows a scatter plot of the predicted and measured expression values for the `chr12` and `chr13` of the K562 cell line.

```
par(mfrow=c(1,2))
plot_scatter_gex(bpr_predict_obj = res_profile)
plot_scatter_gex(bpr_predict_obj = res_mean, main_lab = "Mean Methylation")
```

3.4 Cluster methylation profiles

Another application of the BPR model is to use the higher-order methylation features to cluster DNA methylation patterns across promoter-proximal regions and examine whether distinct methylation profiles are associated to different gene expression levels. To cluster methylation profiles, we consider a mixture modelling approach and we apply the EM algorithm to estimate the model parameters.

The `BPRMeth` package provides the `bpr_cluster_wrap` function for performing the clustering process, where the user needs to provide the number of clusters K, the methylation regions and a basis object. Since we are interested in capturing broader similarities between profiles rather than fine details, we will model the methylation profiles at a slightly lower resolution:

```
# Set seed for reproducible results
set.seed(1234)
# Create basis object with 4 RBFs
basis_obj <- create_rbf_object(M = 4)
# Set number of clusters K = 5
K <- 5
# Perform clustering
res <- bpr_cluster_wrap(x = HTS_data$methyl_region, K = K, basis = basis_obj,
                        em_max_iter = 15, opt_itnmax = 30, is_parallel = FALSE)
```

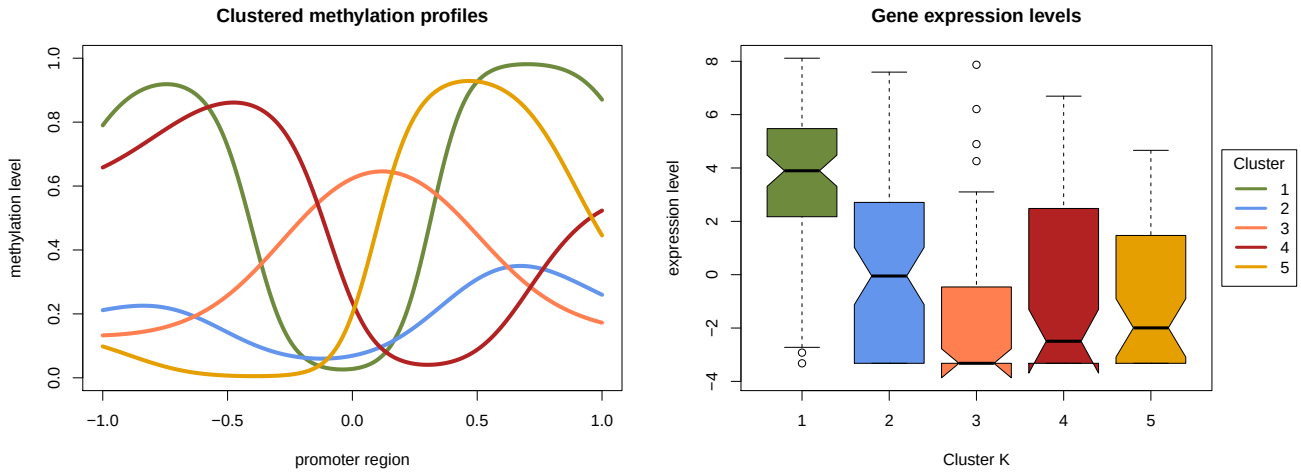

Figure 4: **Clustering methylation profiles across promoter-proximal regions.** *(Left)* Five clustered methylation profiles over $\pm 7kb$ promoter region w.r.t. TSS in the direction of transcription. Each methylation profile is modelled using four RBFs. *(Right)* Boxplots with the corresponding expression levels of the protein-coding genes assigned to each cluster. The colors match with the clustered methylation profiles shown on the left.

Figure 4 shows the fitted methylation profiles for each cluster.

```
par(mfrow=c(1,2))
plot_cluster_prof(bpr_cluster_obj = res)
boxplot_cluster_gex(bpr_cluster_obj = res, gex = HTS_data$gex)
```

4 Session Info

This vignette was compiled using:

```
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows Server 2012 R2 x64 (build 9600)
##
## locale:
## [1] LC_COLLATE=C                           LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets  methods
## [9] base
##
## other attached packages:
## [1] BPRMeth_1.0.0        GenomicRanges_1.26.1 GenomeInfoDb_1.10.0  IRanges_2.8.0
## [5] S4Vectors_0.12.0     BiocGenerics_0.20.0
##
## loaded via a namespace (and not attached):
##  [1] knitr_1.14          XVector_0.14.0      magrittr_1.5        zlibbioc_1.20.0
```

```
##  [5] MASS_7.3-45         doParallel_1.0.10   plotmo_3.2.1        foreach_1.4.3
##  [9] highr_0.6           stringr_1.1.0       tools_3.3.1         data.table_1.9.6
## [13] plotrix_3.6-3       e1071_1.6-7         iterators_1.0.8     class_7.3-14
## [17] assertthat_0.1      randomForest_4.6-12 formatR_1.4         codetools_0.2-15
## [21] evaluate_0.10       earth_4.4.7         stringi_1.1.2       compiler_3.3.1
## [25] TeachingDemos_2.10  BiocStyle_2.2.0     chron_2.3-47
```

5 Acknowledgements

This package was developed at the University of Edinburgh in the School of Informatics, with support from Guido Sanguinetti.

This study was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh, and by the European Research Council through grant MLCS306999.

References

[1] ChantrioInt-Andreas Kapourani and Guido Sanguinetti. Higher order methylation features for clustering and prediction in epigenomic studies. *arXiv:1603.08386*, pages 1–12, 2016. URL: http://arxiv.org/abs/1603.08386, arXiv:1603.08386.

[2] Ian Dunham, Anshul Kundaje, and Encode Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012. URL: http://escholarship.umassmed.edu/sysbio{_}pubs/19/, doi:10.1038/nature11247.An.