An Introduction to the *REMP* Package

Yinan Zheng

November 19, 2017

Contents

1 Introduction

REMP predicts DNA methylation of locus-specific repetitive elements (RE) by learning surrounding genetic and epigenetic information. *REMP* provides genomewide single-base resolution of DNA methylation on RE that are difficult to measure using array-based or sequencing-based platforms, which enables epigenome-wide association study (EWAS) and differentially methylated region (DMR) analysis on RE.

2 Installation

Install *REMP* (release version):

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("REMP")
```

To install devel version:

```
> library(devtools)
> install_github("YinanZheng/REMP")
```

Load *REMP* into the workspace:

```
> library(REMP)
```

3 REMP: Repetitive Element Methylation Prediction

Currently *REMP* supports Human (hg19, build 37) Alu and LINE-1 (L1) repetitive element (RE) methylation prediction using Illumina 450k or EPIC array.

3.1 Groom methylation data

Appropriate data preprocessing including quality control and normalization of methylation data are recommended before running *REMP*. Many packages are available to carry out these data preprocessing steps, for example, *minfi*, *wateRmelon*, and *methylumi*.

REMP is trying to minimize the requirement of the methylation data format. User can maintain the methylation data in *RatioSet* or *GenomicRatioSet* object offered by *minfi*, *data.table*, *data.frame*, *DataFrame*, or *matrix*. User can input either beta value or M-value. There are only two basic requirements of the methylation data:

1. Each row should represent CpG probe and each column should represent sample.
2. The row names should indicate Illumina probe ID (i.e. cg00000029).

However, there are some other common data issues that may prevent *REMP* from running correctly. For example, if the methylation data are in beta value and contain zero methylation values, logit transformation (to create M-value) will create negative infinite value; or the methylation data contain `NA`, `Inf`, or `NaN` data. To tackle these potential issues, *REMP* includes a handy function `grooMethy` which can help detect and fix these issues. We highly recommend to take advantage of this function:

```
> # Get GM12878 methylation data (450k array)
> GM12878_450k <- getGM12878('450k')
> GM12878_450k <- grooMethy(GM12878_450k, verbose = TRUE)
> GM12878_450k

class: RatioSet
dim: 482421 1
metadata(0):
assays(2): Beta M
rownames(482421): cg00000029 cg00000108 ...
  cg27666046 cg27666123
rowData names(0):
colnames(1): GM12878
colData names(0):
Annotation
  array: IlluminaHumanMethylation450k
  annotation: ilmn12.hg19
Preprocessing
  Method: NA
  minfi version: NA
  Manifest version: NA
```

For zero beta values, `grooMethy` will replace them with smallest non-zero beta value. For `NA`/`NaN`/`Inf` values, `grooMethy` will treat them as missing values and then apply KNN-imputation to complete the dataset. If the imputed value is out of the original range (which is possible when `imputebyrow = FALSE`), mean value will be used instead. Warning: imputed values for multimodal distributed CpGs (across samples) may not be correct. Please check package *ENmix* to identify the CpGs with multimodal distribution.

3.2 Prepare annotation data

To run *REMP* for RE methylation prediction, user first needs to prepare some annotation datasets. The function `initREMP` is designed to do the job.

Suppose user will predict Alu methylation using Illumina 450k array data:

```
> data(Alu.demo)
> remparcel <- initREMP(arrayType = "450k", REtype = "Alu",
+                       RE = Alu.demo, ncore = 1)
> remparcel
```

```
REMParcel object
RE type: Alu
Illumina platform: 450k
Valid (max) RE-CpG flanking window size: 1200
Number of RE: 500
Number of RE-CpG: 5039
```

For demonstration, we only use 500 selected Alu sequence dataset which comes along with the package (`Alu.demo`). We specify `RE = Alu.demo`, so that the annotation dataset will be generated for the 500 selected Alu sequences. In real-world prediction, specifying `RE` is not necessary, as the function will pull up the complete RE sequence dataset from package *AnnotationHub*.

All data are stored in the *REMParcel* object. It is recommended to specify a working directory so that the data generated can be preserved for later use:

```
> saveParcel(remparcel)
```

Without specifying working directory using option `work.dir`, the annotation dataset will be created under the temporal directory `tempdir()` by default. User can also turn on the `export` parameter in `initREMP` to save the data automatically.

3.3 Run prediction

Once the annotation data are ready, user can pass the annotation data parcel to `remp` for prediction:

```
> remp.res <- remp(GM12878_450k, REtype = 'Alu',
+                  parcel = remparcel, ncore = 1, seed = 777)
```

If `parcel` is missing, `remp` will then try to search the REMParcel data file in the directory indicated by `work.dir` (use `tempdir()` if not specified).

By default, `remp` uses Random Forest (`method = 'rf'`) model (package *randomForest*) for prediction. Random Forest model is recommended because it offers more accurate prediction results and it automatically enables Quantile Regression Forest (Nicolai Meinshausen, 2006) for prediction reliability evaluation. `remp` constructs 19 predictors to carry out the prediction. For Random Forest model, the tuning parameter `param = 6` (i.e. `mtry` in *randomForest*) indicates how many predictors will be randomly selected for building the individual trees. The performance of random forest model is often relatively insensitive to the choice of `mtry`. Therefore, auto-tune will be turned off using random forest and `mtry` will be set to one third of the total number of predictors. It is recommended to specify a seed for reproducible prediction results.

`remp` will return a *REMPset* object, which inherits Bioconductor's *RangedSummarizedExperiment* class:

```
> remp.res

class: REMProduct
dim: 4808 1
metadata(8): REannotation RECpG ... GeneStats Seed
assays(3): rempB rempM rempQC
rownames: NULL
rowData names(1): RE.Index
colnames(1): GM12878
colData names(1): mtry

> # Display more detailed information
> details(remp.res)

RE type: Alu
Methylation profiling platform: 450k
Flanking window size: 1000
```

```
Prediction model: Random Forest
Using seed: 777
QC model: Quantile Regression Forest
Predicted 4808 CpG sites in 500 Alu

Number of predicted CpGs by chromosome:
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8
 449  276  293  131  179  397  292  102

 chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16
   98   148   254   310    66   127   133   333

chr17 chr18 chr19 chr20 chr21 chr22
  295    81   674    66    37    67

Training information:
  500 profiled Alu by Illumina array are used for model training.
  481 RE-CpGs that have at least 2 neighboring profiled CpGs are used for model training.

Coverage information:
  REMP predicts 500 Alu (4808 RE-CpG).
  Gene coverage by predicted Alu (out of total refSeq Gene):
    492 (1.96%) total genes;
    413 (2.15%) protein-coding genes;
    117 (1.59%) non-coding RNA genes.

Distribution of predicted methylation value (beta value):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02864 0.47292 0.66163 0.59517 0.75088 0.91752

Distribution of prediction reliability score:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.7082  1.4717  1.7806  1.8045  2.0602  5.5515
```

Prediction results can be obtained by accessors:

```
> # Predicted RE-CpG methylation value (Beta value)
> rempB(remp.res)

DataFrame with 4808 rows and 1 column
       GM12878
     <numeric>
1    0.7625485
2    0.7864348
3    0.7966618
4    0.7989569
5    0.8022731
...        ...
4804 0.4572074
4805 0.4557844
4806 0.4648940
4807 0.4883373
4808 0.4941438

> # Predicted RE-CpG methylation value (M value)
> rempM(remp.res)
```

```
DataFrame with 4808 rows and 1 column
         GM12878
       <numeric>
1       1.683196
2       1.880651
3       1.970086
4       1.990613
5       2.020585
...          ...
4804 -0.24755247
4805 -0.25582659
4806 -0.20292303
4807 -0.06731535
4808 -0.03379618
```

```
> # Genomic location information of the predicted RE-CpG
> # Function inherit from class 'RangedSummarizedExperiment'
> rowRanges(remp.res)
```

```
GRanges object with 4808 ranges and 1 metadata column:
         seqnames               ranges strand |    RE.Index
            <Rle>            <IRanges>  <Rle> |       <Rle>
     [1]     chr1     [943927, 943928]      - | Alu_0000527
     [2]     chr1     [943935, 943936]      - | Alu_0000527
     [3]     chr1     [943968, 943969]      - | Alu_0000527
     [4]     chr1     [943974, 943975]      - | Alu_0000527
     [5]     chr1     [943991, 943992]      - | Alu_0000527
     ...      ...                  ...    ... .         ...
  [4804]    chr22 [42095154, 42095155]      - | Alu_1170175
  [4805]    chr22 [42095161, 42095162]      - | Alu_1170175
  [4806]    chr22 [42095170, 42095171]      - | Alu_1170175
  [4807]    chr22 [42095198, 42095199]      - | Alu_1170175
  [4808]    chr22 [42095214, 42095215]      - | Alu_1170175
  -------
  seqinfo: 93 sequences from an unspecified genome; no seqlengths
```

```
> # Standard error-scaled permutation importance of predictors
> rempImp(remp.res)
```

```
DataFrame with 19 rows and 1 column
                 GM12878
               <numeric>
RE.score       8.3661328
RE.Length      6.8405361
RE.CpG.density 5.8717675
RE.InNM        3.0396020
RE.InNR        0.3809575
...                  ...
distance.min2  13.032239
Methy.min      29.625700
Methy.min2     10.924800
Methy.mean     13.161478
Methy.std       3.843843
```

```
> # Retrive seed number used for the reesults
> metadata(remp.res)$Seed
```

```
[1] 777
```

Trim off less reliable predicted results:

```
> # Any predicted CpG values with quality score less than
> # threshold (default = 1.7) will be replaced with NA.
> # CpGs contain more than missingRate * 100% (default = 20%)
> # missing rate across samples will be discarded.
> remp.res <- rempTrim(remp.res, threshold = 1.7, missingRate = 0.2)
> details(remp.res)

RE type: Alu
Methylation profiling platform: 450k
Flanking window size: 1000
Prediction model: Random Forest - trimmed (1.7)
Using seed: 777
QC model: Quantile Regression Forest
Predicted 2108 CpG sites in 392 Alu

Number of predicted CpGs by chromosome:
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8
 215  112  136   73   68  195  135   47

 chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16
   42    80    83   111    22    96    34   182

chr17 chr18 chr19 chr20 chr21 chr22
  141    22   252    34     5    23

Coverage information:
  REMP predicts 392 Alu (2108 RE-CpG).
  Gene coverage by predicted Alu (out of total refSeq Gene):
    375 (1.5%) total genes;
    310 (1.62%) protein-coding genes;
    87 (1.18%) non-coding RNA genes.

Distribution of predicted methylation value (beta value):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.06787 0.67150 0.74475 0.70473 0.79305 0.91615

Distribution of prediction reliability score:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.7082  1.3044  1.4360  1.4240  1.5736  1.6999
```

Aggregate the predicted methylation of CpGs in RE by averaging them to obtain the RE-specific methylation level:

```
> # (ncore is set to 1 only for demostration)
> remp.res <- rempAggregate(remp.res, NCpG = 2, ncore = 1)
> details(remp.res)

RE type: Alu (aggregated by mean: min # of CpGs: 2)
Methylation profiling platform: 450k
Flanking window size: 1000
Prediction model: Random Forest - trimmed (1.7)
```

```
Using seed: 777
QC model: Quantile Regression Forest
Predicted 294 CpG sites in 294 Alu (aggregated by mean: min # of CpGs: 2)

Number of predicted CpGs by chromosome:
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8
  28   17   19   11    7   30   20    8

 chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16
    5    13    11    17     3    10     6    22

chr17 chr18 chr19 chr20 chr21 chr22
   16     5    37     5     1     3

Coverage information:
  REMP predicts 294 Alu (aggregated by mean: min # of CpGs: 2)
  Gene coverage by predicted Alu (aggregated by mean: min # of CpGs: 2) (out of total refSeq Gene):
    271 (1.08%) total genes;
    223 (1.16%) protein-coding genes;
    63 (0.86%) non-coding RNA genes.

Distribution of predicted methylation value (beta value):
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.08809 0.63681 0.72739 0.68464 0.78315 0.90179

Distribution of prediction reliability score:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.7107  1.3567  1.4457  1.4375  1.5440  1.6963
```

It is recommended to aggregate the predicted CpG methylation by RE to obtain RE-wise methylation level. This is beneficial because 1) it greatly reduces the data dimension for downstream analysis and 2) it produces more robust RE methylation estimation. Note that by default, RE with 2 or more predicted CpG sites will be aggregated. Therefore, the downside of doing this is the reduced coverage of RE. The assumption of doing this is the CpG methylation level within each RE are similar.

To add genomic regions annotation of the predicted REs:

```
> # By default gene symbol annotation will be added
> # (ncore is set to 1 only for demostration)
> remp.res <- decodeAnnot(remp.res, ncore = 1)
> rempAnnot(remp.res)
```

Seven genomic region indicators will be added to the annotation data in the input *REMProduct* object:

- InNM: in protein-coding genes (overlap with refSeq gene's "NM" transcripts + 2000 bp upstream of the transcription start site (TSS))
- InNR: in noncoding RNA genes (overlap with refSeq gene's "NR" transcripts + 2000 bp upstream of the TSS)
- InTSS: in flanking region of 2000 bp upstream of the TSS. Default upstream limit is 2000 bp, which can be modified globally using `remp_options`
- In5UTR: in 5'untranslated regions (UTRs)
- InCDS: in coding DNA sequence regions
- InExon: in exon regions

- In3UTR: in 3'UTRs

Note that intron region and intergenic region information can be derived from the above genomic region indicators: if "InNM" and/or "InNR" is not missing but "InTSS", "In5UTR", "InExon", and "In3UTR" are missing, then the RE is strictly located within intron region; if all indicators are missing, then the RE is strictly located in intergenic region.

3.4 Plot prediction

Make a density plot of the predicted methylation (beta values):

```
> plot(remp.res, main = "Alu methylation (GM12878)", col = "blue")
```

4 Session Information

```
R version 3.4.2 (2017-09-28)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.3 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.6-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.6-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    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] parallel  stats4    stats     graphics  grDevices
[6] utils     datasets  methods   base

other attached packages:
 [1] REMP_1.2.3
 [2] IlluminaHumanMethylationEPICanno.ilm10b2.hg19_0.6.0
 [3] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.0
 [4] minfi_1.24.0
 [5] bumphunter_1.20.0
 [6] locfit_1.5-9.1
 [7] iterators_1.0.8
 [8] foreach_1.4.3
 [9] Biostrings_2.46.0
[10] XVector_0.18.0
[11] SummarizedExperiment_1.8.0
[12] DelayedArray_0.4.1
[13] matrixStats_0.52.2
[14] Biobase_2.38.0
[15] GenomicRanges_1.30.0
[16] GenomeInfoDb_1.14.0
[17] IRanges_2.12.0
[18] S4Vectors_0.16.0
[19] BiocGenerics_0.24.0
[20] knitr_1.17

loaded via a namespace (and not attached):
  [1] AnnotationHub_2.10.1
  [2] plyr_1.8.4
  [3] lazyeval_0.2.1
  [4] splines_3.4.2
  [5] BiocParallel_1.12.0
  [6] ggplot2_2.2.1
  [7] digest_0.6.12
  [8] BiocInstaller_1.28.0
  [9] htmltools_0.3.6
```

```
[10] magrittr_1.5
[11] memoise_1.1.0
[12] BSgenome_1.46.0
[13] doParallel_1.0.11
[14] sfsmisc_1.1-1
[15] limma_3.34.1
[16] recipes_0.1.0
[17] readr_1.1.1
[18] annotate_1.56.1
[19] gower_0.1.2
[20] dimRed_0.1.0
[21] siggenes_1.52.0
[22] prettyunits_1.0.2
[23] colorspace_1.3-2
[24] blob_1.1.0
[25] dplyr_0.7.4
[26] settings_0.2.4
[27] RCurl_1.95-4.8
[28] genefilter_1.60.0
[29] impute_1.52.0
[30] bindr_0.1
[31] GEOquery_2.46.8
[32] survival_2.41-3
[33] glue_1.2.0
[34] DRR_0.0.2
[35] registry_0.3
[36] gtable_0.2.0
[37] ipred_0.9-6
[38] zlibbioc_1.24.0
[39] kernlab_0.9-25
[40] ddalpha_1.3.1
[41] DEoptimR_1.0-8
[42] scales_0.5.0
[43] DBI_0.7
[44] rngtools_1.2.4
[45] Rcpp_0.12.13
[46] xtable_1.8-2
[47] progress_1.1.2
[48] bit_1.1-12
[49] mclust_5.3
[50] preprocessCore_1.40.0
[51] lava_1.5.1
[52] prodlim_1.6.1
[53] httr_1.3.1
[54] RColorBrewer_1.1-2
[55] pkgconfig_2.0.1
[56] reshape_0.8.7
[57] XML_3.98-1.9
[58] nnet_7.3-12
[59] caret_6.0-77
[60] rlang_0.1.4
[61] reshape2_1.4.2
[62] AnnotationDbi_1.40.0
[63] munsell_0.4.3
```

```
 [64] tools_3.4.2
 [65] RSQLite_2.0
 [66] ranger_0.8.0
 [67] evaluate_0.10.1
 [68] stringr_1.2.0
 [69] yaml_2.1.14
 [70] org.Hs.eg.db_3.5.0
 [71] ModelMetrics_1.1.0
 [72] bit64_0.9-7
 [73] beanplot_1.2
 [74] robustbase_0.92-8
 [75] purrr_0.2.4
 [76] bindrcpp_0.2
 [77] nlme_3.1-131
 [78] doRNG_1.6.6
 [79] mime_0.5
 [80] nor1mix_1.2-3
 [81] RcppRoll_0.2.2
 [82] xml2_1.1.1
 [83] biomaRt_2.34.0
 [84] compiler_3.4.2
 [85] curl_3.0
 [86] interactiveDisplayBase_1.16.0
 [87] tibble_1.3.4
 [88] stringi_1.1.6
 [89] GenomicFeatures_1.30.0
 [90] lattice_0.20-35
 [91] Matrix_1.2-12
 [92] multtest_2.34.0
 [93] data.table_1.10.4-3
 [94] bitops_1.0-6
 [95] httpuv_1.3.5
 [96] rtracklayer_1.38.0
 [97] R6_2.2.2
 [98] RMySQL_0.10.13
 [99] codetools_0.2-15
[100] MASS_7.3-47
[101] assertthat_0.2.0
[102] CVST_0.2-1
[103] openssl_0.9.9
[104] pkgmaker_0.22
[105] withr_2.1.0
[106] GenomicAlignments_1.14.1
[107] Rsamtools_1.30.0
[108] GenomeInfoDbData_0.99.1
[109] hms_0.3
[110] quadprog_1.5-5
[111] grid_3.4.2
[112] rpart_4.1-11
[113] timeDate_3042.101
[114] tidyr_0.7.2
[115] base64_2.0
[116] class_7.3-14
[117] illuminaio_0.20.0
```

```
[118] BSgenome.Hsapiens.UCSC.hg19_1.4.0
[119] shiny_1.0.5
[120] lubridate_1.7.1
```