

# An Introduction to the *REMP* Package

Yinan Zheng

October 27, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>REMP: Repetitive Element Methylation Prediction</b>	<b>2</b>
3.1	Groom methylation data . . . . .	2
3.2	Prepare annotation data . . . . .	4
3.3	Run prediction . . . . .	4
3.4	Plot prediction . . . . .	10
<b>4</b>	<b>Extract RE-CpG methylation profiled by Illumina BeadChip array</b>	<b>11</b>

# 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 is difficult to measure directly using array-based or sequencing-based platforms, which enables epigenome-wide association study (EWAS) and differentially methylated region (DMR) analysis on RE. *REMP* also provides handy tool to extract methylation data of CpGs that are located within RE sequences.

*REMP* supports both Illumina methylation BeadChip array platforms (450k and EPIC) and sequencing platforms (e.g. TruSeq Methyl Capture EPIC). Both genome build hg19 and hg38 are supported.

## 2 Installation

Install *REMP* (release version):

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+   install.packages("BiocManager")
> BiocManager::install("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/hg38) Alu, LINE-1 (L1), and Long Terminal Repeat (LTR) (including endogenous retroviruses, ERV) repetitive element (RE) methylation prediction using Illumina 450k/EPIC array or sequencing platform.

### 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. Users can maintain the methylation data in *RatioSet* or *GenomicRatioSet* object offered by *minfi*, *data.table*, *data.frame*, *DataFrame*, or *matrix*. Users can input either beta value or M-value. There are only two basic requirements of the methylation array data (450k/EPIC):

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 `groomMethy` 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)
> 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 one beta values, `grooMethy` will replace them with largest non-one 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.

For sequencing data, the users only need to prepare a methylation data matrix (row = CpGs, column = samples). The corresponding CpG location information (either in hg19 or hg38) should be prepared in a separate *GRanges* object and provide it to the `Seq.GR` argument in `grooMethy`. For an example of `Seq.GR`, please run:

```

> library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
> getLocations(IlluminaHumanMethylation450kanno.ilmn12.hg19)

```

GRanges object with 485512 ranges and 0 metadata columns:

	seqnames	ranges	strand
	<Rle>	<IRanges>	<Rle>
	cg00050873	chrY 9363356	*
	cg00212031	chrY 21239348	*
	cg00213748	chrY 8148233	*
	cg00214611	chrY 15815688	*
	cg00455876	chrY 9385539	*
	...	...	...
	ch.22.909671F	chr22 46114168	*
	ch.22.46830341F	chr22 48451677	*
	ch.22.1008279F	chr22 48731367	*
	ch.22.47579720R	chr22 49193714	*
	ch.22.48274842R	chr22 49888838	*

-----  
seqinfo: 24 sequences from hg19 genome; no seqlengths

Note that the row names of the CpGs in `Seq.GR` can be NULL.

### 3.2 Prepare annotation data

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

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

```
> data(Alu.hg19.demo)
> remparcel <- initREMP(arrayType = "450k",
+                       REtype = "Alu",
+                       annotation.source = "AH",
+                       genome = "hg19",
+                       RE = Alu.hg19.demo,
+                       ncore = 1)
> remparcel
```

```
REMPParcel object
RE type: Alu
Genome build: hg19
Illumina platform: 450k
Valid (max) Alu-CpG flanking window size: 1200
Number of RE: 500
Number of Alu-CpG: 4799
```

For demonstration, we only use 500 selected Alu sequence dataset which comes along with the package (`Alu.hg19.demo`). We specify `RE = Alu.hg19.demo`, so that the annotation dataset will be generated for the 500 selected Alu sequences. Most of the time, specifying `RE` is not necessary, as the function will fetch the complete RE sequence dataset from package *AnnotationHub* using `fetchRMSK`. Users can also use this argument `RE` to provide customized RE dataset.

`annotation.source` allows the users to switch the source of the annotation databases, including the RefSeq Gene annotation database and RepeatMasker annotation database. If `annotation.source = "AH"`, the database will be obtained from the AnnotationHub package. If `annotation.source = "UCSC"`, the database will be downloaded from the UCSC website <http://hgdownload.cse.ucsc.edu/goldenpath>. The corresponding build ("`hg19`" or "`hg38`") can be specified in the argument `genome`. Most of the time "`hg19`" is used for array data. But if "`hg38`" is specified, the function will liftover the CpG probe location information to "`hg38`" and obtain annotation databases in "`hg38`".

If `arrayType = "Sequencing"`, users should provide the genomic location information of the CpGs in a *GRanges* object to `Seq.GR`. Note that the genome build of `Seq.GR` provided should match the genome build specified in `genome`.

All data are stored in the *REMPParcel* object:

```
> saveParcel(remparcel)
```

It is recommended to specify a working directory using argument `work.dir` in `initREMP` so that the annotation data generated can be re-used. Without specifying working directory, the annotation dataset will be created under the temporal directory `tempdir()` by default. Users can also turn on the `export` argument in `initREMP` to save the data automatically.

### 3.3 Run prediction

Once the annotation data are ready, users 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, `rem` will then try to search the *REMP*Parcel data file in the directory indicated by `work.dir`. If `work.dir` is also missing, `rem` will try to search the *REMP*Parcel data file in the temporal directory `tempdir()`.

By default, `rem` uses Random Forest (`method = 'rf'`) model (package *ranger* for fast implementation) 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. `rem` constructs predictors to carry out the prediction. For Random Forest model, the tuning parameter `param = 6` (i.e. `mtry` in *ranger* or *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.

Besides random forest, `rem` provides other machine learning engines for users to explore, including Extreme Gradient Boosting, SVM with linear kernel, and SVM with radial kernel).

`rem` will return a *REMP*set object, which inherits Bioconductor's *RangedSummarizedExperiment* class:

```
> rem.res

class: REMProduct
dim: 4619 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(rem.res)
```

```
RE type: Alu
Genome build: hg19
Methylation profiling platform: 450k
Flanking window size: 1000
Prediction model: Random Forest
QC model: Quantile Regression Forest
Seed: 777
Covered 4619 CpG sites in 500 Alu
```

```
Number of Alu-CpGs by chromosome:
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8
 471  316  235  120  271  260  248  128

chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16
 185  128  150  247   77  135  158  280

chr17 chr18 chr19 chr20 chr21 chr22
 257   57  665   71   37  123
```

Training information:

500 profiled Alu are used for model training.

490 Alu-CpGs that have at least 2 neighboring profiled CpGs are used for model training.

Coverage information:

The data cover 500 Alu (4619 Alu-CpG).

Gene coverage by Alu (out of total # of RefSeq genes):

508 (2.04%) total genes;

446 (2.33%) protein-coding genes;

90 (1.25%) non-coding RNA genes.

Distribution of methylation value (beta value):

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.01279493	0.54212918	0.67273730	0.61234316	0.75604032	0.93424969

Distribution of reliability score (lower score = higher reliability):

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.5809995	1.2578860	1.4031167	1.5804923	1.8121280	4.5947357

Prediction results can be obtained by accessors:

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

DataFrame with 4619 rows and 1 column

```
GM12878
<numeric>
1      0.907993
2      0.909524
3      0.929527
4      0.909676
5      0.910390
...
4615  0.600192
4616  0.615479
4617  0.623186
4618  0.720386
4619  0.770255
```

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

DataFrame with 4619 rows and 1 column

```
GM12878
<numeric>
1      3.30286
2      3.32951
3      3.72136
4      3.33218
5      3.34475
...
4615  0.586119
4616  0.678648
4617  0.725810
```

```
4618 1.365334
4619 1.745299
```

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

GRanges object with 4619 ranges and 1 metadata column:

	seqnames	ranges	strand	RE.Index
	<Rle>	<IRanges>	<Rle>	<Rle>
[1]	chr1	942687-942688	+	Alu_0000177
[2]	chr1	942694-942695	+	Alu_0000177
[3]	chr1	942696-942697	+	Alu_0000177
[4]	chr1	942699-942700	+	Alu_0000177
[5]	chr1	942734-942735	+	Alu_0000177
...	...	...	...	...
[4615]	chr22	32768411-32768412	-	Alu_1112204
[4616]	chr22	42343697-42343698	-	Alu_1115852
[4617]	chr22	42343732-42343733	-	Alu_1115852
[4618]	chr22	42343856-42343857	-	Alu_1115852
[4619]	chr22	42343915-42343916	-	Alu_1115852

-----  
seqinfo: 22 sequences from an unspecified genome; no seqlengths

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

DataFrame with 18 rows and 1 column

	GM12878
	<numeric>
RE.score	7.19041
RE.Length	4.75104
RE.CpG.density	12.06688
RE.InTSS	-0.46608
RE.In5UTR	2.41559
...	...
Methy.mean.mov1	16.83022
Methy.mean.mov2	16.38181
Methy.mean.mov3	8.41076
Methy.mean.mov4	7.11607
Methy.std	5.80539

```
> # Retrieve seed number used for the reresults
> 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  
Genome build: hg19  
Methylation profiling platform: 450k  
Flanking window size: 1000  
Prediction model: Random Forest - trimmed (1.7)  
QC model: Quantile Regression Forest  
Seed: 777  
Covered 3289 CpG sites in 422 Alu

Number of Alu-CpGs by chromosome:  
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8  
361 254 164 104 144 210 163 76  
  
chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16  
108 89 121 132 28 85 125 230  
  
chr17 chr18 chr19 chr20 chr21 chr22  
143 55 525 66 15 91

Coverage information:  
The data cover 422 Alu (3289 Alu-CpG).  
Gene coverage by Alu (out of total # of RefSeq genes):  
419 (1.68%) total genes;  
366 (1.91%) protein-coding genes;  
75 (1.04%) non-coding RNA genes.

Distribution of methylation value (beta value):  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.04761694 0.64972352 0.72187453 0.70765058 0.78015014 0.93424969

Distribution of reliability score (lower score = higher reliability):  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
0.5809995 1.1994791 1.3212931 1.3181896 1.4353829 1.6999526

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

```
> remp.res <- rempAggregate(remp.res, NCpG = 2)  
> details(remp.res)
```

RE type: Alu (aggregated by mean: min # of CpGs: 2)  
Genome build: hg19  
Methylation profiling platform: 450k  
Flanking window size: 1000  
Prediction model: Random Forest - trimmed (1.7)  
QC model: Quantile Regression Forest  
Seed: 777  
Covered 386 Alu (aggregated by mean: min # of CpGs: 2)

Number of Alu (aggregated by mean: min # of CpGs: 2) by chromosome:  
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8  
36 28 22 11 15 25 21 9  
  
chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16



```

11    12    16    17    6    10    10    24
chr17 chr18 chr19 chr20 chr21 chr22
18    5    70    8    1    11

```

Coverage information:

The data cover 386 Alu (aggregated by mean: min # of CpGs: 2)  
Gene coverage by Alu (aggregated by mean: min # of CpGs: 2) (out of total # of RefSeq genes):  
380 (1.52%) total genes;  
335 (1.75%) protein-coding genes;  
63 (0.87%) non-coding RNA genes.

Distribution of methylation value (beta value):

```

Min.    1st Qu.    Median    Mean    3rd Qu.    Max.
0.06438777 0.62228067 0.69174782 0.68451281 0.76179442 0.90633885

```

Distribution of reliability score (lower score = higher reliability):

```

Min.    1st Qu.    Median    Mean    3rd Qu.    Max.
0.8067775 1.2336517 1.3383629 1.3484495 1.4730713 1.6843300

```

Aggregating CpGs in the same RE for RE-level methylation data is beneficial because 1) it greatly reduces the data dimension for downstream analysis and 2) it may produce 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
> remp.res <- decodeAnnot(remp.res)
> rempAnnot(remp.res)

```

GRanges object with 386 ranges and 10 metadata columns:

	seqnames	ranges	strand	name	score
	<Rle>	<IRanges>	<Rle>	<character>	<integer>
[1]	chr1	942688-942997	+	AluSq	2448
[2]	chr1	1253757-1254042	+	AluJb	1580
[3]	chr1	1625133-1625434	+	AluSg	2264
[4]	chr1	1885941-1886230	+	AluJb	1962
[5]	chr1	11107877-11108180	+	AluSg	2397
...	...	...	...	...	...
[382]	chr22	17848277-17848585	-	AluSq2	2396
[383]	chr22	25159264-25159566	-	AluSx	2340
[384]	chr22	30277121-30277395	-	AluJb	1948
[385]	chr22	32768148-32768431	-	AluJb	1839
[386]	chr22	42343674-42343938	-	AluJr	1317

  

	Index	InNM.symbol	InNR.symbol	InTSS.symbol
	<Rle>	<character>	<character>	<character>
[1]	Alu_0000177	<NA>	<NA>	<NA>
[2]	Alu_0000241	INTS11	<NA>	<NA>
[3]	Alu_0000460	CDK11B SLC35E2B	<NA>	SLC35E2B
[4]	Alu_0000634	CFAP74	<NA>	<NA>
[5]	Alu_0003424	MASP2	<NA>	MASP2
...	...	...	...	...
[382]	Alu_1106441	CECR2	<NA>	<NA>

[383]	Alu_1109250	PIWIL3	PIWIL3 TOP1P2	TOP1P2
[384]	Alu_1110888	MTMR3	<NA>	MTMR3
[385]	Alu_1112204	<NA>	RFPL3S	RFPL3S
[386]	Alu_1115852	CENPM	<NA>	CENPM
	In5UTR.symbol	InCDS.symbol	InExon.symbol	In3UTR.symbol
	<character>	<character>	<character>	<character>
[1]	<NA>	<NA>	<NA>	<NA>
[2]	<NA>	INTS11	<NA>	<NA>
[3]	CDK11B	CDK11B	<NA>	<NA>
[4]	<NA>	<NA>	CFAP74	CFAP74
[5]	<NA>	<NA>	<NA>	<NA>
...	...	...	...	...
[382]	CECR2	<NA>	<NA>	<NA>
[383]	PIWIL3	<NA>	<NA>	<NA>
[384]	<NA>	<NA>	<NA>	<NA>
[385]	<NA>	<NA>	<NA>	<NA>
[386]	<NA>	<NA>	<NA>	<NA>

-----

seqinfo: 22 sequences from an unspecified genome; no seqlengths

Seven genomic region indicators will be added to the annotation data in the input *REMP* 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):

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

## 4 Extract RE-CpG methylation profiled by Illumina BeadChip array

*REMP* offers a handy tool to extract methylation data of CpGs that are located in RE. Similar as *remp*, users can choose the source of annotation database (AH: AnnotationHub or UCSC: UCSC website) and genome build (hg19 or hg38).

```
> # Use Alu.hg19.demo for demonstration
> remp.res <- remprofile(GM12878_450k,
+                       REtype = "Alu",
+                       annotation.source = "AH",
+                       genome = "hg19",
+                       RE = Alu.hg19.demo)
> details(remp.res)
```

```
RE type: Alu
Genome build: hg19
Methylation profiling platform: 450k
Flanking window size: N/A
Prediction model: Profiled
QC model: N/A
Covered 595 CpG sites in 500 Alu
```

Number of Alu-CpGs by chromosome:

```
chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8
  60  34  33  18  31  40  34  17

chr9 chr10 chr11 chr12 chr13 chr14 chr15 chr16
  17  20  27  28   9  15  13  29

chr17 chr18 chr19 chr20 chr21 chr22
  30   6  99  14   4  17
```

Coverage information:

```
The data cover 500 Alu (595 Alu-CpG).
Gene coverage by Alu (out of total # of RefSeq genes):
  508 (2.04%) total genes;
  446 (2.33%) protein-coding genes;
  90 (1.25%) non-coding RNA genes.
```

Distribution of methylation value (beta value):

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
0.0010000 0.3985000 0.6690000 0.5891748 0.8110000 0.9730000
```

```
> # All accessors and utilites for REMProduct are applicable
> remp.res <- rempAggregate(remp.res)
> details(remp.res)
```

```
RE type: Alu (aggregated by mean: min # of CpGs: 2)
Genome build: hg19
Methylation profiling platform: 450k
Flanking window size: N/A
Prediction model: Profiled
QC model: N/A
Covered 73 Alu (aggregated by mean: min # of CpGs: 2)
```

Number of Alu (aggregated by mean: min # of CpGs: 2) by chromosome:

chr1	chr2	chr3	chr4	chr5	chr6	chr7	chr8
10	3	3	4	3	7	2	1

chr10	chr11	chr12	chr14	chr15	chr17	chr18	chr19
4	6	4	2	1	3	1	13

chr20	chr21	chr22
2	1	3

Coverage information:

The data cover 73 Alu (aggregated by mean: min # of CpGs: 2)

Gene coverage by Alu (aggregated by mean: min # of CpGs: 2) (out of total # of RefSeq genes):

85 (0.34%) total genes;

71 (0.37%) protein-coding genes;

19 (0.26%) non-coding RNA genes.

Distribution of methylation value (beta value):

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.04152111	0.44247940	0.64911063	0.57671567	0.76178351	0.90023474