Genome-wide methylation analysis using coMethDMR via parallel computing

Gabriel J. Odom, Lissette Gomez, and Lily Wang

8/20/2019

1 Introduction

In the previous vignette “Introduction to **coMethDMR**”, we discussed components of the **coMethDMR** pipeline and presented example R scripts for running an analysis with **coMethDMR** serially. However, because identifying co-methylated clusters and fitting mixed effects models on large numbers of genomic regions can be computationally expensive, we illustrate implementation of parallel computing for **coMethDMR** via the **BiocParallel** R package in this vignette.

First, we load these two packages.

```
library(coMethDMR)
library(BiocParallel)
```

In Section 2, we give a brief re-introduction to the example data. In Section 3, we present example scripts for analyzing a single type (e.g. CpG islands) of genomic regions using parallel computing. In Section 4, we present example scripts for analyzing genomic regions from all eleven types on the Illumina arrays (TSS1500, TSS200, UTR5, EXON1, GENEBODY, UTR3, NSHORE, NSHELF, ISLAND, SSHORE and SSHELF) using parallel computing.

2 Example Dataset

For illustration, we use a subset of prefrontal cortex methylation data (GEO GSE59685) described in Lunnon et al. (2014). This example dataset includes beta values for 8552 CpGs on chromosome 22 for a random selection of 20 subjects. We assume quality control and normalization of the methylation dataset have been performed by R packages such as **minfi** or **RnBeads**.

```
data(betaMatrixChr22_df)
betaMatrixChr22_df[1:5, 1:5]
```

```
##            GSM1443279 GSM1443663 GSM1443434 GSM1443547 GSM1443577
## cg00004192  0.9249942  0.8463296  0.8700718  0.9058205  0.9090382
## cg00004775  0.6523025  0.6247554  0.7573476  0.6590817  0.6726261
## cg00012194  0.8676339  0.8679048  0.8484754  0.8754985  0.8484458
## cg00013618  0.9466056  0.9475467  0.9566493  0.9588431  0.9419563
## cg00014104  0.3932388  0.5525716  0.4075900  0.3997278  0.3216956
```

The corresponding phenotype dataset included variables `stage` (Braak AD stage), `subject.id`, `Mplate` (batch effect), `sex`, `Sample` and `age.brain` (age of the brain donor).

```
data(pheno_df)
head(pheno_df)
```

```
##    stage subject.id     Mplate         sex     Sample age.brain
## 3      0          1 6042316048 Sex: FEMALE GSM1443251        82
## 8      2          2 6042316066 Sex: FEMALE GSM1443256        82
## 10    NA          3 6042316066   Sex: MALE GSM1443258        89
## 15     1          4 7786923107 Sex: FEMALE GSM1443263        81
## 21     2          5 6042316121 Sex: FEMALE GSM1443269        92
## 22     1          6 6042316099   Sex: MALE GSM1443270        78
```

Note that only samples with both methylation data and non-missing phenotype data are analyzed by **coMethDMR**. So in this example, the sample with `subject.id = 3` which lacks AD stage information will be excluded from analysis.

3 Analyzing One Type of Genomic Region via BiocParallel

As mentioned previously in "Introduction to **coMethDMR**", there are several steps in the **coMethDMR** pipeline:

1. Obtain CpGs located closely in pre-defined genomic regions (e.g. CpG islands),
2. Identify co-methylated regions, and
3. Test co-methylated regions against the outcome variable (AD stage).

Suppose we are interested in analyzing genomic regions corresponding to the gene promoter regions, specifically the `TSS200` regions. In the following, `closeByTSS200_ls` is a list, where each item includes CpGs located closely (max distance between 2 CpGs is 200bp by default) in the `TSS200` regions.

```
closeByTSS200_ls <- readRDS(
  system.file(
    "extdata",
    "TSS200_3_200.rds",
    package = 'coMethDMR',
    mustWork = TRUE
  )
)

closeByTSS200_ls[1]
```

```
## $`Chr5:16180033-16180076`
## [1] "cg25092681" "cg00339556" "cg01791874" "cg17030173" "cg17712694"
## [6] "cg16150752" "cg21901718"
```

If you are interested in analyzing other types of genomic regions, see "Introduction to **coMethDMR**", Section 2.1 for necessary modifications to this code.

3.1 Finding co-methylated regions

The cluster computing with the **BiocParallel** package involves two steps:

1. Setting up the clusters, and
2. Passing in the cluster as an argument to `CoMethAllRegions()`.

First, we set up the cluster:

```
snow_cl <- SnowParam(workers = 20, type = "SOCK")
```

Now we execute the `CoMethAllRegions()` function using each worker in the cluster, to find co-methylated clusters in the TSS200 regions.

Note that for demonstration, the following script evaluates only the first 500 regions by setting `CpGs_ls = closeByTSS200_ls[1:500]`, to identify co-methylated clusters in all 10907 TSS200 regions, change to `CpGs_ls = closeByTSS200_ls`. This would take about 6 minutes over 20 cores with 64Gb of RAM.

```
a0 <- Sys.time()
coMeth_ls <- CoMethAllRegions(
  betaMatrix = betaMatrixChr22_df,
  method = "spearman",
  arrayType = "450k",
  CpGs_ls = closeByTSS200_ls[1:500],
  cluster = snow_cl
)
Sys.time() - a0
```

```
## Time difference of 43.449 secs
```

Note that argument `regionType = "TSS200"` can also be used to obtain the co-methyalted regions. The following scripts are equivalent to the block of scripts above.

```
a0 <- Sys.time()
coMeth_ls <- CoMethAllRegions(
  betaMatrix = betaMatrixChr22_df,
  method = "spearman",
  arrayType = "450k",
  regionType = "TSS200",
  cluster = snow_cl
)
Sys.time() - a0
```

The object `coMeth_ls` is a list, with each item containing the list of CpGs within an identified co-methylated region.

3.2 Testing the co-methylated regions

Next we test these co-methylated regions against continuous phenotype stage, adjusting for covariates age and sex. We now execute the `lmmTestAllRegions()` function. This completes in a few seconds for the TSS200 regions. Note that computing the linear mixed model can often take more memory than identifying co-methylated regions.

```
res_df <- lmmTestAllRegions(
  beta_df = betaMatrixChr22_df,
  region_ls = coMeth_ls,
  pheno_df = pheno_df,
  contPheno_char = "stage",
  covariates_char = c("age.brain", "sex"),
  modelType = "randCoef",
  arrayType = "450k",
  outLogFile = "res_lmm_log.txt"
)
```

```
## messages for mixed model fittings are in file res_lmm_log.txt
```

Model fit messages and diagnostics for each region will be saved to the log file specified with the `outLogFile` argument. For a single region, this will return a one row of model fit statistics similar to the following:

chrom	start	end	nCpGs	Estimate	StdErr	Stat	pValue
chr22	24823455	24823519	4	-0.0702	0.0290	-2.4184	0.0155

4 Analyzing All Types of Genomic Regions via BiocParallel

Genomic regions on the Illumina arrays can be defined based on their relations to genes or CpG Islands. Genomic regions annotated to genes include TSS1500, TSS200, UTR5, EXON1, GENEBODY and UTR3. Genomic regions annotated to CGIs are NSHORE, NSHELF, ISLAND, SSHORE and SSHELF. In this section, we describe analyzing three of these 11 types of genomic regions using coMethDMR.

Same as before, first, we create the clusters of workers. If you still have the clusters defined, you can reuse them; if not, create them again.

```
snow_cl <- SnowParam(workers = 20, type = "SOCK")
```

4.1 Analyzing multiple region types

For illustration, we will only analyze CpG islands and shores. If you would like to analyse all region types, use the `AllRegions_char` vector instead of `regions_char`.

```
# Create vector of region types
# AllRegions_char <- c(
#   "NSHORE", "NSHELF", "SSHORE", "SSHELF", "TSS1500", "TSS200", "UTR5", "EXON1",
#   "GENEBODY", "UTR3", "ISLAND"
# )
regions_char <- c(
  "NSHORE", "SSHORE", "ISLAND"
)

# Create empty results list for region types
resultsAll_ls <- vector(mode = "list", length = length(regions_char))
names(resultsAll_ls) <- regions_char
```

Finally, we execute the `CoMethAllRegions()` function over each worker on the cluster (this completes in roughly 17 minutes using 20 cores and 64Gb of RAM), then pass these results to `lmmTestAllRegions()` :

```
aAll <- Sys.time()

for (region in regions_char) {

  print(region) # Uncomment for progress updates
  pfcRegions_ls <- CoMethAllRegions(
    betaMatrix = betaMatrixChr22_df,
    method = "spearman",
    arrayType = "450k",
    regionType = region,
    cluster = snow_cl
  )

  res_df <- lmmTestAllRegions(
    beta_df = betaMatrixChr22_df,
    region_ls = pfcRegions_ls,
    pheno_df = pheno_df,
    contPheno_char = "stage",
    covariates_char = c("age.brain", "sex"),
    modelType = "randCoef",
    arrayType = "450k",
    outLogFile = paste0(region, "_lmm_log.txt")
  )

  res_df$regionType <- region

  resultsAll_ls[[region]] <- res_df

}

Sys.time() - aAll
```

Now that we have the results, we need to stack the model fits for each region.

```
lmmResults_df <- do.call(rbind, resultsAll_ls)
row.names(lmmResults_df) <- NULL

head(lmmResults_df)
```

```
##   chrom    start      end nCpGs     Estimate     StdErr       Stat
## 1 chr22 17082770 17082787     3 -0.024283518 0.02282435 -1.0639301
## 2 chr22 18632429 18632476     3 -0.003507545 0.01861621 -0.1884135
## 3 chr22 20861470 20861479     3 -0.013943194 0.02021033 -0.6899042
## 4 chr22 21398553 21398582     3 -0.060684190 0.03448228 -1.7598658
## 5 chr22 21921731 21921833     4  0.044321951 0.01343634  3.2986632
## 6 chr22 23522307 23522474     5 -0.048519527 0.02202002 -2.2034281
##         pValue        FDR regionType
## 1 0.2873603476 0.52682730     NSHORE
## 2 1.0000000000 1.00000000     NSHORE
## 3 0.4902544311 0.77039982     NSHORE
## 4 1.0000000000 1.00000000     NSHORE
## 5 0.0009714639 0.02137221     NSHORE
## 6 0.0275645926 0.10107017     NSHORE
```

We can see this output is similar to the previous output, with the additional column annotating the type of genomic region corresponding to each co-methylated region.

4.2 Annotate results

Finally, we may be interested in the specific probes in these regions, or which genes overlap with these regions. We can use the `AnnotateResults()` function.

```
lmmResAnnotated_df <- AnnotateResults(
  lmmRes_df = lmmResults_df,
  arrayType = "450k"
)
head(lmmResAnnotated_df)
```

```
##   chrom    start      end nCpGs     Estimate     StdErr       Stat
## 1 chr22 17082770 17082787     3 -0.024283518 0.02282435 -1.0639301
## 2 chr22 18632429 18632476     3 -0.003507545 0.01861621 -0.1884135
## 3 chr22 20861470 20861479     3 -0.013943194 0.02021033 -0.6899042
## 4 chr22 21398553 21398582     3 -0.060684190 0.03448228 -1.7598658
## 5 chr22 21921731 21921833     4  0.044321951 0.01343634  3.2986632
## 6 chr22 23522307 23522474     5 -0.048519527 0.02202002 -2.2034281
##         pValue        FDR regionType UCSC_RefGene_Group
## 1 0.2873603476 0.52682730     NSHORE             TSS200
## 2 1.0000000000 1.00000000     NSHORE            TSS1500
## 3 0.4902544311 0.77039982     NSHORE            TSS1500
## 4 1.0000000000 1.00000000     NSHORE             TSS200
## 5 0.0009714639 0.02137221     NSHORE     TSS1500;TSS200
## 6 0.0275645926 0.10107017     NSHORE     TSS1500;TSS200
##          UCSC_RefGene_Accession UCSC_RefGene_Name
## 1                     NR_001591         psiTPTE22
## 2                     NM_017414             USP18
## 3        NM_001003891;NM_015889             MED15
## 4                     NR_002829            P2RX6P
## 5 NM_003347;NR_028436;NR_028437            UBE2L3
## 6           NM_004327;NM_021574               BCR
##                                                   probes
## 1                         cg01660911;cg14505585;cg12431879
## 2                         cg01963623;cg18699242;cg02169692
## 3                         cg19265040;cg26608332;cg26796825
## 4                         cg00004192;cg15169469;cg15362336
## 5            cg17274080;cg25421647;cg26067319;cg26844104
## 6 cg03930964;cg04028010;cg00281467;cg07260820;cg10480239
```

Finally. we can save our results:

```
write.csv(
  lmmResAnnotated_df,
  file = "EWAS_PFC_lmmOut.csv"
)
```

5 Additional Comments on Computational Time and Resources

In this vignette, we have analyzed a small subset of a real EWAS dataset (i.e. only chromosome 22 data on 20 subjects). To give users a more realistic estimate of time for analyzing real EWAS datasets, we also measured time used for analyzing the entire Lunnon et al. (2014) dataset with 110 samples on all chromosomes. These computation times measured on a Dell Precision 5810 with 64Gb of RAM, an Intel Xeon E5-2640 CPU at 2.40Ghz, and using up to 20 cores. More specifically, in Section 4, the entire **coMethDMR** workflow for all 11 types of genomic regions took 50 minutes with 12 cores and used 24Gb of RAM. We're currently working improving the speed and reducing the size of **coMethDMR**, so please check back soon for updates.