Lab 02: Analyzing squamous cell carcinoma methylation data

Sean Cho

Overview

```
library(GEOquery)
library(minfi)
library(limma)
```

For this task, we will be analysing a public dataset on the Gene Expression Omnibus (GEO), GSE67097. We will be doing the following:

1. Download the data
2. Process the metadata
3. Read the raw .idat files
4. Perform QC analysis
5. Normalize the data
6. Get beta-values or M-values
7. Identify differentially methylated probes (DMPs)
8. Identify differentually methylated regions (DMRs)

Download dataset

We will be using the `GEOquery` package to download the GSE67097 dataset. We will first download the supplementary files. This next code chunk will:

1. create the necessary directories for download
2. download and parse data into appropriate directories
3. get a list of idat files to be passed on to be read

```
if(length(list.files('data/scc/GSE67097/idat')) != 26){
    ## functions
    movefiles <- function(from){
      to <- gsub('data/scc/GSE67097','data/scc/GSE67097/idat', from)
      file.rename(from, to)
    }

    ## create directories
    sapply(c('data','rda','output'), dir.create)

    #### get data
    ## download raw data and move data
    dir.create('data/scc', recursive = TRUE) ## create data directory
    getGEOSuppFiles('GSE67097', baseDir = 'data/scc') ## download IDAT files
    untar(tarfile = 'data/scc/GSE67097/GSE67097_RAW.tar', exdir = 'data/scc/GSE67097') ## untar file
    dir.create('data/scc/GSE67097/idat') ## create idat directory to store idat file

    idat_files <- list.files('data/scc/GSE67097/', pattern = 'idat.gz', full.names = TRUE)
    sapply(idat_files, movefiles) ## move files into idat folder
    idat_files <- list.files('data/scc/GSE67097/idat', pattern = 'idat.gz', full.names = TRUE)
```

```
    sapply(idat_files, gunzip, overwrite = TRUE) ## unzip files
}

idat_files <- list.files('data/scc/GSE67097/idat', pattern = 'idat$', full.names = TRUE)
idat_files
```

```
##  [1] "data/scc/GSE67097/idat/GSM1638770_9482801039_R01C01_Grn.idat"
##  [2] "data/scc/GSE67097/idat/GSM1638770_9482801039_R01C01_Red.idat"
##  [3] "data/scc/GSE67097/idat/GSM1638771_9482801039_R02C01_Grn.idat"
##  [4] "data/scc/GSE67097/idat/GSM1638771_9482801039_R02C01_Red.idat"
##  [5] "data/scc/GSE67097/idat/GSM1638772_9482801039_R04C01_Grn.idat"
##  [6] "data/scc/GSE67097/idat/GSM1638772_9482801039_R04C01_Red.idat"
##  [7] "data/scc/GSE67097/idat/GSM1638773_9482801039_R05C01_Grn.idat"
##  [8] "data/scc/GSE67097/idat/GSM1638773_9482801039_R05C01_Red.idat"
##  [9] "data/scc/GSE67097/idat/GSM1638774_9482801039_R01C02_Grn.idat"
## [10] "data/scc/GSE67097/idat/GSM1638774_9482801039_R01C02_Red.idat"
## [11] "data/scc/GSE67097/idat/GSM1638775_9482801039_R02C02_Grn.idat"
## [12] "data/scc/GSE67097/idat/GSM1638775_9482801039_R02C02_Red.idat"
## [13] "data/scc/GSE67097/idat/GSM1638776_9482801039_R04C02_Grn.idat"
## [14] "data/scc/GSE67097/idat/GSM1638776_9482801039_R04C02_Red.idat"
## [15] "data/scc/GSE67097/idat/GSM1638777_9482801039_R05C02_Grn.idat"
## [16] "data/scc/GSE67097/idat/GSM1638777_9482801039_R05C02_Red.idat"
## [17] "data/scc/GSE67097/idat/GSM1638778_9482801099_R01C01_Grn.idat"
## [18] "data/scc/GSE67097/idat/GSM1638778_9482801099_R01C01_Red.idat"
## [19] "data/scc/GSE67097/idat/GSM1638779_9482801099_R02C01_Grn.idat"
## [20] "data/scc/GSE67097/idat/GSM1638779_9482801099_R02C01_Red.idat"
## [21] "data/scc/GSE67097/idat/GSM1638780_9482801099_R04C01_Grn.idat"
## [22] "data/scc/GSE67097/idat/GSM1638780_9482801099_R04C01_Red.idat"
## [23] "data/scc/GSE67097/idat/GSM1638781_9482801099_R01C02_Grn.idat"
## [24] "data/scc/GSE67097/idat/GSM1638781_9482801099_R01C02_Red.idat"
## [25] "data/scc/GSE67097/idat/GSM1638782_9482801099_R04C02_Grn.idat"
## [26] "data/scc/GSE67097/idat/GSM1638782_9482801099_R04C02_Red.idat"
```

Process metadata

Next, we will download the phenotype metadata using GEOquery. `getGEO` will download all the data into a `list` because some GSE entries has multiple associated data sets. GSE67097 only has one entry so we will extract the data using `[[1]]`.

```
## get phenodata
gse <- getGEO('GSE67097')
```

```
## Found 1 file(s)
```

```
## GSE67097_series_matrix.txt.gz
```

```
## Parsed with column specification:
## cols(
##   ID_REF = col_character(),
##   GSM1638770 = col_double(),
##   GSM1638771 = col_double(),
##   GSM1638772 = col_double(),
##   GSM1638773 = col_double(),
##   GSM1638774 = col_double(),
##   GSM1638775 = col_double(),
```

```
##   GSM1638776 = col_double(),
##   GSM1638777 = col_double(),
##   GSM1638778 = col_double(),
##   GSM1638779 = col_double(),
##   GSM1638780 = col_double(),
##   GSM1638781 = col_double(),
##   GSM1638782 = col_double()
## )
```

```
## File stored at:
```

```
## /var/folders/mj/1zdvk5cs1vlc3vjjrwf7zd280000gn/T//Rtmpd4Y2og/GPL13534.soft
```

```
rawpheno <- pData(gse[[1]])
str(rawpheno)
```

```
## 'data.frame':    13 obs. of  36 variables:
##  $ title                  : Factor w/ 13 levels "genomic DNA from arm squamous cell carcinoma",..: 5
##  $ geo_accession          : chr  "GSM1638770" "GSM1638771" "GSM1638772" "GSM1638773" ...
##  $ status                 : Factor w/ 1 level "Public on Mar 23 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ submission_date        : Factor w/ 1 level "Mar 20 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ last_update_date       : Factor w/ 1 level "Mar 24 2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ type                   : Factor w/ 1 level "genomic": 1 1 1 1 1 1 1 1 1 1 ...
##  $ channel_count          : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ source_name_ch1        : Factor w/ 11 levels "arm squamous cell carcinoma",..: 4 9 2 3 1 7 5 6 3 8
##  $ organism_ch1           : Factor w/ 1 level "Homo sapiens": 1 1 1 1 1 1 1 1 1 1 ...
##  $ characteristics_ch1    : Factor w/ 11 levels "tissue: arm squamous cell carcinoma",..: 4 9 2 3 1 1
##  $ characteristics_ch1.1  : Factor w/ 2 levels "gender: female",..: 1 1 1 1 2 1 1 1 1 1 ...
##  $ characteristics_ch1.2  : Factor w/ 6 levels "body site: Arm",..: 4 5 2 3 1 3 5 1 3 4 ...
##  $ treatment_protocol_ch1 : Factor w/ 1 level "preserved in OCT immediately after biopsy": 1 1 1 1 1
##  $ growth_protocol_ch1    : Factor w/ 0 levels: NA NA NA NA NA NA NA NA NA NA ...
##  $ molecule_ch1           : Factor w/ 1 level "genomic DNA": 1 1 1 1 1 1 1 1 1 1 ...
##  $ extract_protocol_ch1   : Factor w/ 1 level "Genomic DNA was purified using EpiCentre MasterPure ki
##  $ label_ch1              : Factor w/ 1 level "Cy5 and Cy3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ label_protocol_ch1     : Factor w/ 1 level "Standard Illumina Protocol": 1 1 1 1 1 1 1 1 1 1 ...
##  $ taxid_ch1              : Factor w/ 1 level "9606": 1 1 1 1 1 1 1 1 1 1 ...
##  $ hyb_protocol           : Factor w/ 1 level "Bisulphite converted DNA was amplified, fragmented and
##  $ scan_protocol          : Factor w/ 1 level "Arrays were imaged using Illumina iScan using standard
##  $ data_processing        : Factor w/ 1 level "Minfi Package": 1 1 1 1 1 1 1 1 1 1 ...
##  $ platform_id            : Factor w/ 1 level "GPL13534": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_name           : Factor w/ 1 level "Andrew,,Feinberg": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_institute      : Factor w/ 1 level "Johns Hopkins University School of Medicine": 1 1 1 1
##  $ contact_address        : Factor w/ 1 level "855 N. Wolfe St": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_city           : Factor w/ 1 level "Baltimore": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_state          : Factor w/ 1 level "Maryland": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_zip/postal_code: Factor w/ 1 level "21205": 1 1 1 1 1 1 1 1 1 1 ...
##  $ contact_country        : Factor w/ 1 level "USA": 1 1 1 1 1 1 1 1 1 1 ...
##  $ supplementary_file     : Factor w/ 13 levels "ftp://ftp.ncbi.nlm.nih.gov/geo/samples/GSM1638nnn/GS
##  $ supplementary_file.1   : Factor w/ 13 levels "ftp://ftp.ncbi.nlm.nih.gov/geo/samples/GSM1638nnn/GS
##  $ data_row_count         : Factor w/ 1 level "467971": 1 1 1 1 1 1 1 1 1 1 ...
##  $ body site:ch1          : chr  "Lip" "Neck" "Calf" "Leg" ...
##  $ gender:ch1             : chr  "female" "female" "female" "female" ...
##  $ tissue:ch1             : chr  "lip squamous cell carcinoma" "normal neck skin" "calf squamous cell
```

There are several columns of information here that we want to capture, including the tissue, gender, body source, tumor type, and if the skin source is exposed to the sun.

The `dl_metadata` data.frame will be used to import the raw data appropriately. The required columns are as follows:

1. `Slide`: The array chip ID
2. `Array`: The position on the chip
3. `Basename`: Location of the idat file

Any other columns will be appended onto the phenotype data when the data is read.

```
## create metadata for reading data using minfi
dl_metadata <- data.frame(do.call(rbind, strsplit(basename(idat_files), split = '_')),
                          stringsAsFactors = FALSE)
colnames(dl_metadata) <- c('Sample_Name', 'Slide', 'Array', 'suffix')
dl_metadata$Basename <- gsub('_[GrnRed]+\\.idat','', idat_files)
dl_metadata$suffix <- NULL
dl_metadata <- unique(dl_metadata)

## annotate metadata
sun <- c('Lip','Neck','Scalp','Arm')
nosun <- c('Calf','Leg')

dl_metadata$OTissue <- rawpheno$`tissue:ch1`
dl_metadata$OGender <- rawpheno$`gender:ch1`
dl_metadata$OBody <- rawpheno$`body site:ch1`
dl_metadata$Tumor <- ifelse(grepl('squamous', rawpheno$`tissue:ch1`), 'SCC', 'normal')
dl_metadata$Sun <- ifelse(dl_metadata$OBody %in% sun, 'sun', 'nosun')

head(dl_metadata)
```

```
##    Sample_Name      Slide  Array
## 1   GSM1638770 9482801039 R01C01
## 3   GSM1638771 9482801039 R02C01
## 5   GSM1638772 9482801039 R04C01
## 7   GSM1638773 9482801039 R05C01
## 9   GSM1638774 9482801039 R01C02
## 11  GSM1638775 9482801039 R02C02
##                                                Basename
## 1  data/scc/GSE67097/idat/GSM1638770_9482801039_R01C01
## 3  data/scc/GSE67097/idat/GSM1638771_9482801039_R02C01
## 5  data/scc/GSE67097/idat/GSM1638772_9482801039_R04C01
## 7  data/scc/GSE67097/idat/GSM1638773_9482801039_R05C01
## 9  data/scc/GSE67097/idat/GSM1638774_9482801039_R01C02
## 11 data/scc/GSE67097/idat/GSM1638775_9482801039_R02C02
##                          OTissue OGender OBody  Tumor   Sun
## 1   lip squamous cell carcinoma  female   Lip    SCC   sun
## 3              normal neck skin  female  Neck normal   sun
## 5  calf squamous cell carcinoma  female  Calf    SCC nosun
## 7   leg squamous cell carcinoma  female   Leg    SCC nosun
## 9   arm squamous cell carcinoma    male   Arm    SCC   sun
## 11             normal leg skin  female   Leg normal nosun
```

Read raw IDAT data

We will use `read.metharray.exp` from the `minfi` package with the `dl_metadata` data.frame to read the raw idat file.

The data is read into an `RGChannelSet` (red/green channel set) which is a structured object that stores the raw signal intensities, annotation information about the array, and phenotype data as provided by the targets argument (in our case `dl_metadata`).

We will extract the phenotype information using `pData` which will be ordered the same way as the signal intensity information.

```
## read data
rawdata <- read.metharray.exp(targets = dl_metadata)
class(rawdata)
```

```
## [1] "RGChannelSet"
## attr(,"package")
## [1] "minfi"
```

```
## save rawdata for future experiments
if(!file.exists('rda/scc_rawdata.rds')){
  saveRDS(rawdata, file = 'rda/scc_rawdata.rds')
}
## start analysis
pheno <- pData(rawdata)
head(pheno)
```

```
## DataFrame with 6 rows and 10 columns
##                              Sample_Name       Slide       Array
##                              <character> <character> <character>
## GSM1638770_9482801039_R01C01  GSM1638770  9482801039      R01C01
## GSM1638771_9482801039_R02C01  GSM1638771  9482801039      R02C01
## GSM1638772_9482801039_R04C01  GSM1638772  9482801039      R04C01
## GSM1638773_9482801039_R05C01  GSM1638773  9482801039      R05C01
## GSM1638774_9482801039_R01C02  GSM1638774  9482801039      R01C02
## GSM1638775_9482801039_R02C02  GSM1638775  9482801039      R02C02
##                                                                                  Basename
##                                                                               <character>
## GSM1638770_9482801039_R01C01 data/scc/GSE67097/idat/GSM1638770_9482801039_R01C01
## GSM1638771_9482801039_R02C01 data/scc/GSE67097/idat/GSM1638771_9482801039_R02C01
## GSM1638772_9482801039_R04C01 data/scc/GSE67097/idat/GSM1638772_9482801039_R04C01
## GSM1638773_9482801039_R05C01 data/scc/GSE67097/idat/GSM1638773_9482801039_R05C01
## GSM1638774_9482801039_R01C02 data/scc/GSE67097/idat/GSM1638774_9482801039_R01C02
## GSM1638775_9482801039_R02C02 data/scc/GSE67097/idat/GSM1638775_9482801039_R02C02
##                                                 OTissue     OGender
##                                               <character> <character>
## GSM1638770_9482801039_R01C01  lip squamous cell carcinoma      female
## GSM1638771_9482801039_R02C01             normal neck skin      female
## GSM1638772_9482801039_R04C01 calf squamous cell carcinoma      female
## GSM1638773_9482801039_R05C01  leg squamous cell carcinoma      female
## GSM1638774_9482801039_R01C02  arm squamous cell carcinoma        male
## GSM1638775_9482801039_R02C02              normal leg skin      female
##                                    OBody       Tumor         Sun
##                              <character> <character> <character>
## GSM1638770_9482801039_R01C01         Lip         SCC         sun
## GSM1638771_9482801039_R02C01        Neck      normal         sun
## GSM1638772_9482801039_R04C01        Calf         SCC       nosun
## GSM1638773_9482801039_R05C01         Leg         SCC       nosun
## GSM1638774_9482801039_R01C02         Arm         SCC         sun
## GSM1638775_9482801039_R02C02         Leg      normal       nosun
```

```
##                                                                   filenames
##                                                                 <character>
## GSM1638770_9482801039_R01C01 data/scc/GSE67097/idat/GSM1638770_9482801039_R01C01
## GSM1638771_9482801039_R02C01 data/scc/GSE67097/idat/GSM1638771_9482801039_R02C01
## GSM1638772_9482801039_R04C01 data/scc/GSE67097/idat/GSM1638772_9482801039_R04C01
## GSM1638773_9482801039_R05C01 data/scc/GSE67097/idat/GSM1638773_9482801039_R05C01
## GSM1638774_9482801039_R01C02 data/scc/GSE67097/idat/GSM1638774_9482801039_R01C02
## GSM1638775_9482801039_R02C02 data/scc/GSE67097/idat/GSM1638775_9482801039_R02C02
```

QC

We can perform a QC analysis using the convenient `qcReport` function from `minfi` which plots the beta-value distributions of all the samples and the intensities of control probes. We can look at the output file to identify outliers that should be excluded from the analysis.

```
qcReport(rawdata, pdf = 'output/ssc_qc.pdf')
```

```
## Loading required package: IlluminaHumanMethylation450kmanifest
```

```
## pdf
##   2
```

There are no outliers in this dataset so we will proceed with the analysis.

Normalization

We will be using `preprocessFunnorm` to perform data normalization across the samples using the functional normalization algorithm. This will change the `RGChannelSet` into a `GenomicRatioSet`, which now contains beta-values.

```
## normalization
fndata <- preprocessFunnorm(rawdata)
```

```
## [preprocessFunnorm] Background and dye bias correction with noob
```

```
## Loading required package: IlluminaHumanMethylation450kanno.ilmn12.hg19
```

```
## [dyeCorrection] Applying R/G ratio flip to fix dye bias
```

```
## [preprocessFunnorm] Mapping to genome
```

```
## [preprocessFunnorm] Quantile extraction
```

```
## [preprocessFunnorm] Normalization
```

Get methylation values

Next, we extract the methylation values as beta-values and M-values. We will be using M-values in a linear model to identify differentially methylated probes. We will be using beta-values to filter the analysis results and interpret the data since differences in beta-values are more intuitive than M-values.

```
## beta values
Bdat <- getBeta(fndata)
head(colnames(Bdat))
```

```
## [1] "GSM1638770_9482801039_R01C01" "GSM1638771_9482801039_R02C01"
## [3] "GSM1638772_9482801039_R04C01" "GSM1638773_9482801039_R05C01"
```

```
## [5] "GSM1638774_9482801039_R01C02" "GSM1638775_9482801039_R02C02"
```

```
## fix column names
colnames(Bdat) <- sapply(colnames(Bdat), function(x) strsplit(x, '_')[[1]][1])
head(colnames(Bdat))
```

```
## GSM1638770_9482801039_R01C01 GSM1638771_9482801039_R02C01
##                 "GSM1638770"                 "GSM1638771"
## GSM1638772_9482801039_R04C01 GSM1638773_9482801039_R05C01
##                 "GSM1638772"                 "GSM1638773"
## GSM1638774_9482801039_R01C02 GSM1638775_9482801039_R02C02
##                 "GSM1638774"                 "GSM1638775"
```

```
bdat <- Bdat
head(bdat)
```

```
##            GSM1638770 GSM1638771 GSM1638772 GSM1638773 GSM1638774
## cg13869341 0.90164216 0.86612952 0.87188389 0.82237180 0.81243716
## cg14008030 0.66582663 0.63074112 0.73009919 0.65907630 0.64797828
## cg12045430 0.05343860 0.03230307 0.03103815 0.04686546 0.03771607
## cg20826792 0.06473237 0.05419466 0.08370047 0.08374139 0.07976441
## cg00381604 0.03333421 0.03195213 0.02748341 0.02594008 0.03777784
## cg20253340 0.48884899 0.44999921 0.53151900 0.38894218 0.57034019
##            GSM1638775 GSM1638776 GSM1638777 GSM1638778 GSM1638779
## cg13869341 0.83117233 0.91293168 0.88974986 0.84771474 0.88265281
## cg14008030 0.63454315 0.55971792 0.60205645 0.61206850 0.61112629
## cg12045430 0.02984643 0.07319306 0.02734739 0.05201216 0.02861094
## cg20826792 0.04960912 0.15504078 0.05779645 0.12346318 0.08834652
## cg00381604 0.03017852 0.03076504 0.03288153 0.03941853 0.03005465
## cg20253340 0.51507997 0.67682911 0.55714039 0.54944082 0.42902538
##            GSM1638780 GSM1638781 GSM1638782
## cg13869341 0.87053683 0.85393401 0.85327260
## cg14008030 0.65782165 0.64134044 0.61498803
## cg12045430 0.03247936 0.03593104 0.02497080
## cg20826792 0.10530053 0.09798425 0.06374052
## cg00381604 0.03064035 0.02578155 0.02764550
## cg20253340 0.52288393 0.51794929 0.54523273
```

```
## M values
Mdat <- getM(fndata)
colnames(Mdat) <- sapply(colnames(Mdat), function(x) strsplit(x, '_')[[1]][1])
Mdat[Mdat==Inf] <- max(Mdat[Mdat!=Inf])
Mdat[Mdat==-Inf] <- min(Mdat[Mdat!=-Inf])
mdat <- Mdat
head(mdat)
```

```
##             GSM1638770 GSM1638771 GSM1638772 GSM1638773 GSM1638774
## cg13869341  3.19644296  2.6937449  2.7666841  2.2109302  2.1148822
## cg14008030  0.99454978  0.7724154  1.4356631  0.9509966  0.8802810
## cg12045430 -4.14674192 -4.9048122 -4.9643255 -4.3460829 -4.6732114
## cg20826792 -3.85282012 -4.1253207 -3.4525116 -3.4517420 -3.5281861
## cg00381604 -4.85794188 -4.9210944 -5.1450901 -5.2307559 -4.6707580
## cg20253340 -0.06436069 -0.2895112  0.1821307 -0.6517532  0.4086279
##             GSM1638775 GSM1638776 GSM1638777 GSM1638778 GSM1638779
## cg13869341  2.29959629  3.3902871  3.0126194  2.4768025  2.9110627
## cg14008030  0.79601719  0.3462718  0.5973350  0.6578912  0.6521688
## cg12045430 -5.02258270 -3.6624902 -5.1524495 -4.1879476 -5.0854106
```

```
## cg20826792 -4.25984368 -2.4462340 -4.0269859 -2.8277338 -3.3672404
## cg00381604 -5.00612545 -4.9774825 -4.8783432 -4.6069622 -5.0122434
## cg20253340  0.08704957  1.0664943  0.3311915  0.2862475 -0.4123636
##            GSM1638780 GSM1638781 GSM1638782
## cg13869341  2.7493637  2.5475043  2.5398684
## cg14008030  0.9429480  0.8384753  0.6756550
## cg12045430 -4.8966971 -4.7458338 -5.2871315
## cg20826792 -3.0868904 -3.2025308 -3.8766258
## cg00381604 -4.9835273 -5.2398345 -5.1363657
## cg20253340  0.1321504  0.1036259  0.2617437
```

Identify DMPs using minfi

`minfi` has a convenience function called `dmpFinder` that can be used to identify differentially methylated probes. It accepts a matrix of beta or M-values and a vector of phenotype to run the comparison against.

In our analysis, that would be `pheno$Tumor`.

```
dmpres <- dmpFinder(mdat, pheno$Tumor)
sum(dmpres$qval <= 0.05)
```

```
## [1] 38145
```

Here, we have `sum(dmpres$qval <= 0.05)` DMPs from running `dmpFinder`.

Identify DMPs using limma

While `dmpFinder` is convenient, it doesn't allow us to include covariates in our linear model. We will use the `limma` package to identify differentially methylated probes (DMPs), which is what goes on under the hood of `dmpFinder`.

Design matrix and contrasts

We will create a design matrix using the `model.matrix` function that captures information about the phenotype of interest and covariates that could capture other sources of biological variation. We will create appropriate contrasts for the design matrix that will calculate differences in the groups of interest.

```
## make model matrix and contrasts
sccmodel <- model.matrix(~ 0 + pheno$Tumor + pheno$OGender)
colnames(sccmodel) <- c('normal','scc','gender')
scc_contrasts <- makeContrasts(scc - normal, levels = sccmodel)
```

Fit the model

We will use `lmFit`, `contrasts.fit`, and `eBayes` to identify the DMPs. Then, we will use `top.table` to extract the DMPs.

Next, we will flag DMPs using the following filters:

1. FDR <= 0.05 in the M-value analysis
2. abs(delta beta) >= 0.3 in the beta-value analysis

```
## fit model
sccfit1 <- lmFit(mdat, design = sccmodel)
sccfit2 <- contrasts.fit(sccfit1, scc_contrasts)
sccfiteb <- eBayes(sccfit2)
scc_res <- topTable(sccfiteb, coef = 1, number = nrow(mdat))

## do the same for beta values
sccfit1b <- lmFit(bdat, design = sccmodel)
sccfit2b <- contrasts.fit(sccfit1b, scc_contrasts)
sccfitebb <- eBayes(sccfit2b)
scc_resb <- topTable(sccfitebb, coef = 1, number = nrow(bdat))

## flag differentially methylated probes
dmps <- rownames(scc_res)[scc_res$adj.P.Val <= 0.05] ## FDR < 0.05 in M-value
dmps <- dmps[abs(scc_resb[dmps,]$logFC) >= 0.3] ## delta beta >= 0.3
scc_resb$DMP <- rownames(scc_resb) %in% dmps

head(scc_resb)
```

```
##                 logFC   AveExpr         t      P.Value    adj.P.Val
## cg17094249 -0.4002074 0.2824954 -19.01831 1.081911e-09 0.0003075845
## cg11118235 -0.2050683 0.1635392 -18.33425 1.593531e-09 0.0003075845
## cg03699566  0.2790467 0.7134223  17.74182 2.253465e-09 0.0003075845
## cg26693553  0.1324031 0.8302075  17.54527 2.534104e-09 0.0003075845
## cg27015773  0.2395286 0.7440730  16.72551 4.191289e-09 0.0003866809
## cg04099543 -0.2576492 0.4786838 -16.51779 4.778636e-09 0.0003866809
##                   B   DMP
## cg17094249 12.60309  TRUE
## cg11118235 12.26711 FALSE
## cg03699566 11.96245 FALSE
## cg26693553 11.85844 FALSE
## cg27015773 11.40797 FALSE
## cg04099543 11.28939 FALSE
```

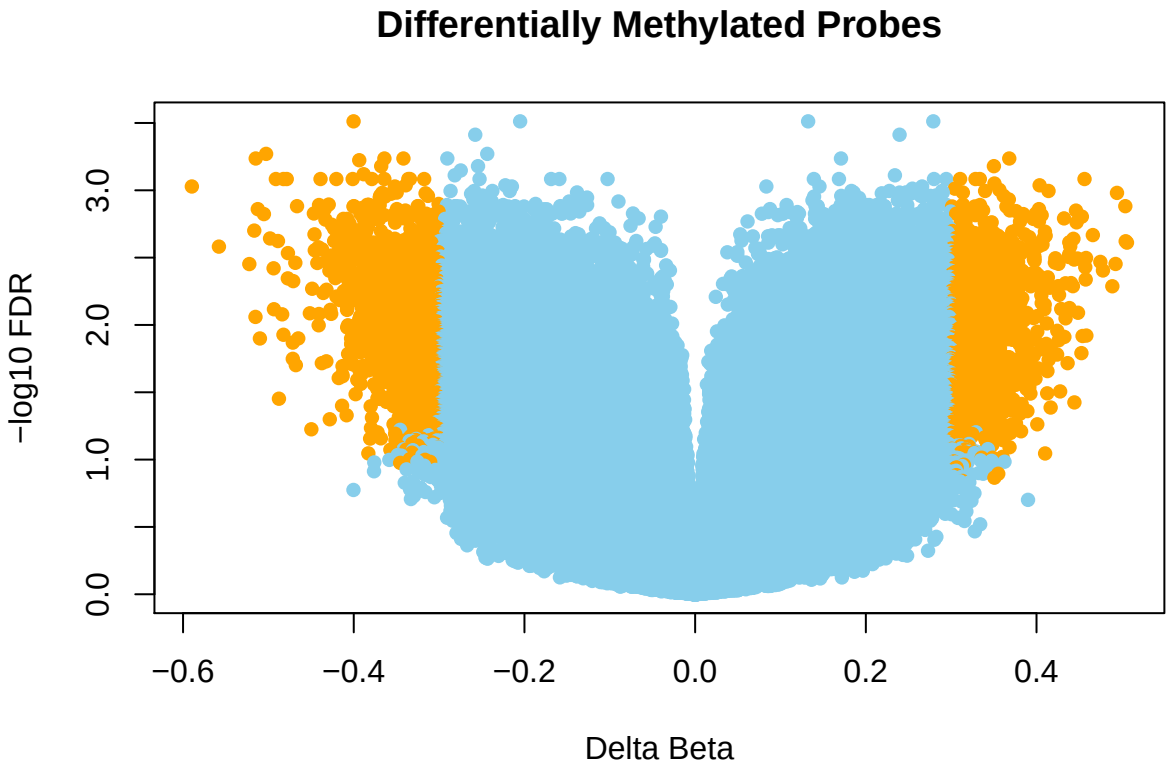
The `topTable` result has 6 columns:

1. logFC: delta-beta of log fold change of M-values
2. AveExpr: average methylation value across all samples
3. t: moderated t-statistic
4. P.Value: p-value
5. adj.P.Val: p-value adjusted using FDR
6. B: B-statistic (log-odds of differential methylation)

Plot result

We will visualise the data using a volcano plot, which plots delta beta on the x-axis and -log10 FDR on the y-axis. We will color DMPs orange and the non-significant probes skyblue.

```
plot(x = scc_resb$logFC, y = -log10(scc_resb$adj.P.Val),
     col = ifelse(scc_resb$DMP, 'orange', 'skyblue'), pch = 16,
     ylab = '-log10 FDR', xlab = 'Delta Beta', main = 'Differentially Methylated Probes')
```



Identify DMRs

The `DMRcate` package can be used to identify differentially methylated regions by comparing DMPs in windows across the genome. As an added bonus, `DMR.plot` also annotates the plot with nearby DNA elements of interest.

```
library(DMRcate)
```

```
## Loading required package: DSS
```

```
## Loading required package: bsseq
```

```
##
## Attaching package: 'bsseq'
```

```
## The following object is masked from 'package:minfi':
##
##     getMeth
```

```
## Loading required package: splines
```

```
## Loading required package: DMRcatedata
```

```
##
```

```
##
```

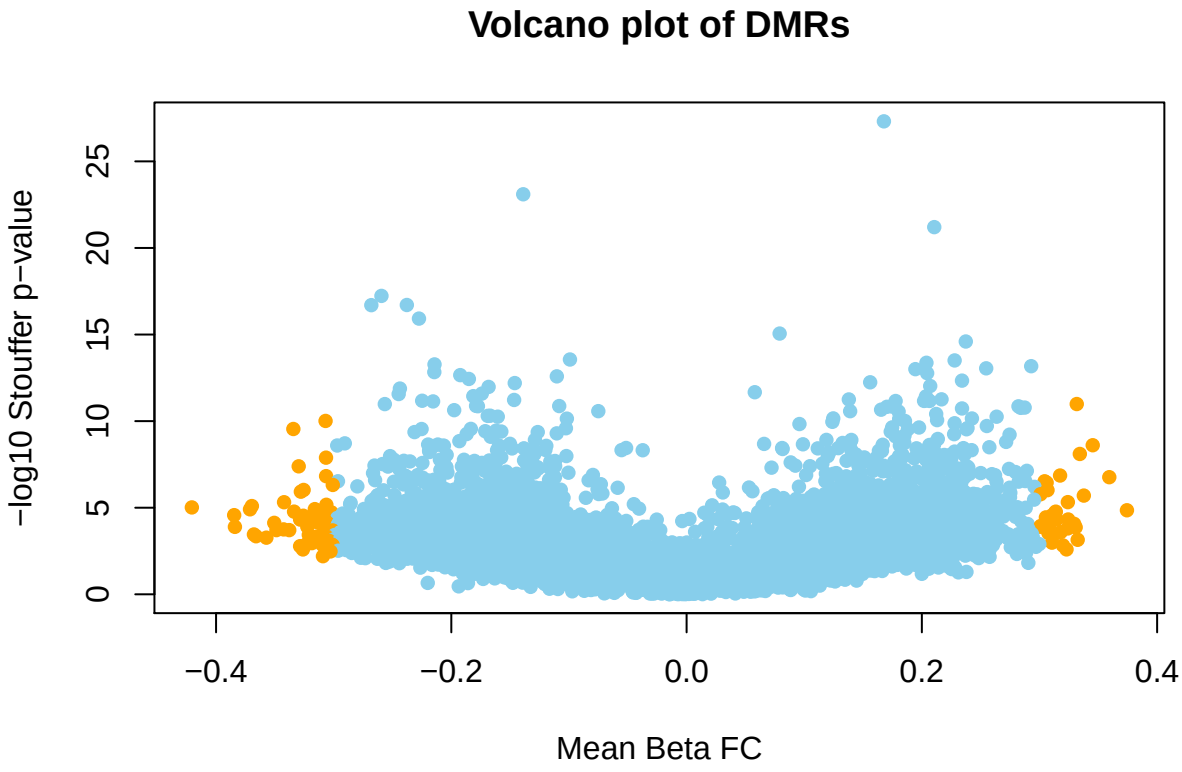
```
## annotate
scc_dmrdat <- cpg.annotate(object = mdat, datatype = 'array',
                           what = 'M', design = sccmodel,
                           arraytype = '450K', contrasts = TRUE,
                           cont.matrix = scc_contrasts,
                           coef = 'scc - normal')
```

```
## Your contrast returned 38815 individually significant probes. We recommend the default setting of pcu
```

```
# contrasts
scc_dmr <- dmrcate(scc_dmrdat)
```

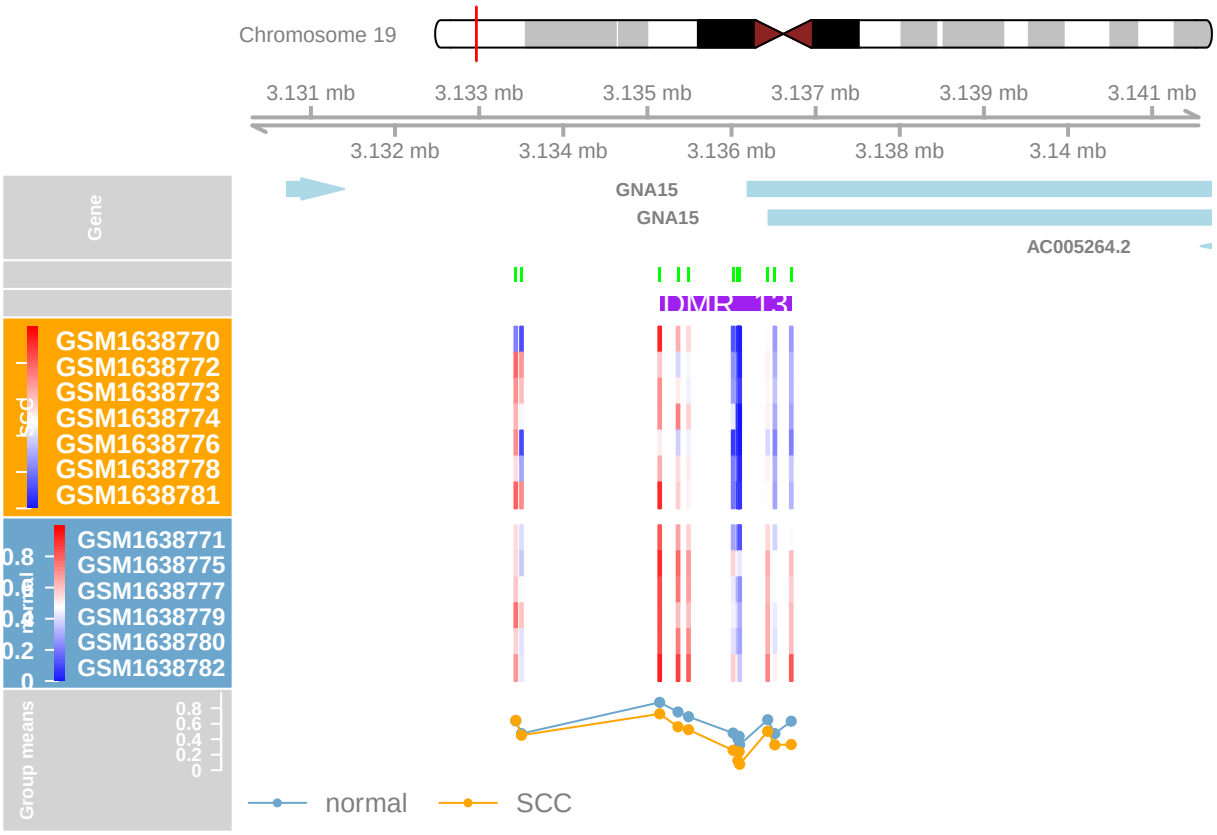
```
## Fitting chr1...
## Fitting chr10...
## Fitting chr11...
## Fitting chr12...
## Fitting chr13...
## Fitting chr14...
## Fitting chr15...
## Fitting chr16...
## Fitting chr17...
## Fitting chr18...
## Fitting chr19...
## Fitting chr2...
## Fitting chr20...
## Fitting chr21...
## Fitting chr22...
## Fitting chr3...
## Fitting chr4...
## Fitting chr5...
## Fitting chr6...
## Fitting chr7...
## Fitting chr8...
## Fitting chr9...
## Fitting chrX...
## Fitting chrY...
## Demarcating regions...
## Done!
```

```
scc_dmranges <- extractRanges(scc_dmr, genome = 'hg19')
plot(x = scc_dmranges$meanbetafc, y = -log10(scc_dmranges$Stouffer),
     col = ifelse(abs(scc_dmranges$meanbetafc) >= 0.3 &
                    scc_dmranges$Stouffer < 0.01,'orange','skyblue'),
     pch = 16, xlab = 'Mean Beta FC', ylab = '-log10 Stouffer p-value',
     main = 'Volcano plot of DMRs')
```




```
## create colors
colgrp <- c('SCC' = 'orange', 'normal' = 'skyblue3')[pheno$Tumor]

DMR.plot(ranges = scc_dmranges, dmr = 13, CpGs=bdat,
         phen.col =  colgrp,
         what = 'Beta', arraytype = '450K', pch = 16, plotmedians = TRUE, genome='hg19',
         samps = 1:ncol(bdat), toscale = TRUE)
```



session info

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
##  [1] splines   stats4    parallel  stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] DMRcate_1.16.0
##  [2] DMRcatedata_1.16.0
##  [3] DSS_2.28.0
##  [4] bsseq_1.16.1
##  [5] IlluminaHumanMethylation450kanno.ilmn12.hg19_0.6.0
##  [6] IlluminaHumanMethylation450kmanifest_0.4.0
##  [7] bindrcpp_0.2.2
```

```
##  [8] limma_3.36.1
##  [9] minfi_1.26.2
## [10] bumphunter_1.22.0
## [11] locfit_1.5-9.1
## [12] iterators_1.0.9
## [13] foreach_1.4.4
## [14] Biostrings_2.48.0
## [15] XVector_0.20.0
## [16] SummarizedExperiment_1.10.1
## [17] DelayedArray_0.6.0
## [18] BiocParallel_1.14.1
## [19] matrixStats_0.53.1
## [20] GenomicRanges_1.32.3
## [21] GenomeInfoDb_1.16.0
## [22] IRanges_2.14.10
## [23] S4Vectors_0.18.2
## [24] GEOquery_2.48.0
## [25] Biobase_2.40.0
## [26] BiocGenerics_0.26.0
##
## loaded via a namespace (and not attached):
##   [1] backports_1.1.2
##   [2] Hmisc_4.1-1
##   [3] plyr_1.8.4
##   [4] lazyeval_0.2.1
##   [5] ggplot2_3.0.0.9000
##   [6] digest_0.6.15
##   [7] ensembldb_2.4.1
##   [8] htmltools_0.3.6
##   [9] GO.db_3.6.0
##  [10] checkmate_1.8.5
##  [11] magrittr_1.5
##  [12] memoise_1.1.0
##  [13] BSgenome_1.48.0
##  [14] cluster_2.0.7-1
##  [15] readr_1.1.1
##  [16] annotate_1.58.0
##  [17] R.utils_2.6.0
##  [18] siggenes_1.54.0
##  [19] htmldeps_0.1.1
##  [20] prettyunits_1.0.2
##  [21] colorspace_1.3-2
##  [22] blob_1.1.1
##  [23] BiasedUrn_1.07
##  [24] dplyr_0.7.5
##  [25] RCurl_1.95-4.10
##  [26] genefilter_1.62.0
##  [27] bindr_0.1.1
##  [28] VariantAnnotation_1.26.1
##  [29] survival_2.41-3
##  [30] glue_1.3.0
##  [31] ruv_0.9.7
##  [32] registry_0.5
##  [33] gtable_0.2.0
```

```
##  [34] zlibbioc_1.26.0
##  [35] Rhdf5lib_1.2.1
##  [36] HDF5Array_1.8.1
##  [37] scales_0.5.0
##  [38] DBI_1.0.0
##  [39] rngtools_1.3.1
##  [40] bibtex_0.4.2
##  [41] Rcpp_0.12.17
##  [42] htmlTable_1.12
##  [43] xtable_1.8-2
##  [44] progress_1.1.2
##  [45] foreign_0.8-70
##  [46] bit_1.1-14
##  [47] mclust_5.4.1
##  [48] preprocessCore_1.42.0
##  [49] Formula_1.2-3
##  [50] missMethyl_1.14.0
##  [51] htmlwidgets_1.2
##  [52] httr_1.3.1
##  [53] RColorBrewer_1.1-2
##  [54] acepack_1.4.1
##  [55] pkgconfig_2.0.1
##  [56] reshape_0.8.7
##  [57] XML_3.98-1.11
##  [58] R.methodsS3_1.7.1
##  [59] Gviz_1.24.0
##  [60] nnet_7.3-12
##  [61] tidyselect_0.2.4
##  [62] rlang_0.2.1
##  [63] AnnotationDbi_1.42.1
##  [64] munsell_0.4.3
##  [65] tools_3.5.0
##  [66] RSQLite_2.1.1
##  [67] evaluate_0.11
##  [68] stringr_1.3.1
##  [69] yaml_2.2.0
##  [70] org.Hs.eg.db_3.6.0
##  [71] knitr_1.20
##  [72] bit64_0.9-7
##  [73] beanplot_1.2
##  [74] methylumi_2.26.0
##  [75] purrr_0.2.5
##  [76] AnnotationFilter_1.4.0
##  [77] nlme_3.1-137
##  [78] doRNG_1.6.6
##  [79] nor1mix_1.2-3
##  [80] R.oo_1.22.0
##  [81] xml2_1.2.0
##  [82] biomaRt_2.36.1
##  [83] rstudioapi_0.7
##  [84] compiler_3.5.0
##  [85] curl_3.2
##  [86] statmod_1.4.30
##  [87] tibble_1.4.2
```

```
##  [88] stringi_1.2.2
##  [89] GenomicFeatures_1.32.0
##  [90] IlluminaHumanMethylationEPICanno.ilm10b2.hg19_0.6.0
##  [91] lattice_0.20-35
##  [92] ProtGenerics_1.12.0
##  [93] Matrix_1.2-14
##  [94] permute_0.9-4
##  [95] multtest_2.36.0
##  [96] pillar_1.2.3
##  [97] data.table_1.11.4
##  [98] bitops_1.0-6
##  [99] rtracklayer_1.40.2
## [100] R6_2.2.2
## [101] latticeExtra_0.6-28
## [102] gridExtra_2.3
## [103] codetools_0.2-15
## [104] dichromat_2.0-0
## [105] MASS_7.3-49
## [106] gtools_3.5.0
## [107] assertthat_0.2.0
## [108] rhdf5_2.24.0
## [109] openssl_1.0.1
## [110] pkgmaker_0.27
## [111] withr_2.1.2
## [112] GenomicAlignments_1.16.0
## [113] Rsamtools_1.32.0
## [114] GenomeInfoDbData_1.1.0
## [115] hms_0.4.2
## [116] quadprog_1.5-5
## [117] grid_3.5.0
## [118] rpart_4.1-13
## [119] tidyr_0.8.1
## [120] base64_2.0
## [121] rmarkdown_1.10.10
## [122] DelayedMatrixStats_1.2.0
## [123] illuminaio_0.22.0
## [124] biovizBase_1.28.1
## [125] base64enc_0.1-3
```