Differential Methylation Analysis

Materials

Tutorial materials for this section is found at my github account:

```
https://github.com/jimmybgammyknee/EpiCSA_Methylation_June2016
```

This part of the differential methylation pipeline is mostly done in R. Much like Ben's methyl array tutorial, you can do all of this work in Rstudio.

Due to the time it takes to create all our nessessary files, I have done the before work for you and all you need to do is download the following file from my University CloudStor+ account:

```
https://cloudstor.aarnet.edu.au/plus/index.php/s/RlS5a9Xn3kwGeqx
```

Annotations

But firstly lets use the command-line and download a bed file with the *Arabidopsis thaliana* genome annotation. The gene annotation will allow us to identify where differentially methylated regions are located (promoters, gene-body, intergenic sequences etc).

Firstly we need to download the gff3 file from the Arabidopsis Information Resource (TAIR) and then convert to Bed format using the program bedops (https://bedops.readthedocs.io/en/latest).

For the Workshop I have provided all the nessessary files, so for today you will not have to download the gff file

In bash:

```
wget
https://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/TAIR1
0_gff3/TAIR10_GFF3_genes.gff

convert2bed -i gff < TAIR10_GFF3_genes.gff > TAIR10_GFF3_genes.gff.bed
```

Installing Required R Packages

Now we can start running our statistical analysis in R. Open up Rstudio and nagivate to where your data is found.

Firstly we need to install methylkit:

```
# install some dependencies
install.packages( c("data.table","devtools", "magrittr"))

# And some more dependencies
source("http://bioconductor.org/biocLite.R")
biocLite(c("GenomicRanges","IRanges"))

# And now Methylkit itself from its github repository
library(devtools)
install_github("al2na/methylKit",build_vignettes=FALSE)
```

Running an Analysis

Ok now we're ready to rock. Lets load our required libraries and read our files in as a file list for each context.

```
library(methylKit)
library(magrittr)

# reading multiple files
cg.list <- list("colWT.TAIR10.sorted.markdup_CpG.methylKit",
                "met1.TAIR10.sorted.markdup_CpG.methylKit")
chg.list <- list("colWT.TAIR10.sorted.markdup_CHG.methylKit",
                "met1.TAIR10.sorted.markdup_CHG.methylKit")
chh.list <- list("colWT.TAIR10.sorted.markdup_CHH.methylKit",
                "met1.TAIR10.sorted.markdup_CHH.methylKit")
```

From a list of these files, we now can make data objects in R, which is a data structure that methylkit uses to analyse the data.

```
CG <- methRead(cg.list,
                sample.id=list("Control", "Test"),
                assembly = "Athal10",
                header = TRUE,
                context = "CpG",
                treatment = c(0, 1))
CHG <- methRead(chg.list,
                sample.id=list("Control", "Test"),
                assembly = "Athal10",
                header = TRUE,
                context = "CHG",
                treatment = c(0, 1))
CHH <- methRead(chh.list,
                sample.id=list("Control", "Test"),
                assembly = "Athal10",
                header = TRUE,
                context = "CHH",
                treatment = c(0, 1))
```

As part the initial part of the pipeline, we should have a quick check of the samples to see how they look. Im only going to look at CpG's in the following examples, but you can try all the contexts to see how they look afterwards.

```
getCoverageStats(CG[[1]], plot = TRUE, both.strands = FALSE)
getCoverageStats(CG[[2]], plot = TRUE, both.strands = FALSE)
```

You can also filter the dataset more if you want to be more stringent in your work. For example, you may have very low coverage, so you want to make sure you sites are legitimate.

```
filtered.CG <- filterByCoverage(CG, lo.count = 10,
                                    lo.perc = NULL,
                                    hi.count = NULL,
                                    hi.perc = 99.9)
```

Now its time to do the differential methylation analysis. First we need to unite the data within the object

```
# On either filtered set or normal set
meth.CG <- unite(CG)
```

We can now plot the correlation of each sample and cluster or create a PCA with the samples. This is a little overkill at the moment considering we only have two samples which

we are expecting to be quite different. However, ideally you will be using replicates, so this step is crucial to see how much variation you have within you biological or technical replicates. We wont run through these today

```
getCorrelation(meth.CG, plot = T)

clusterSamples(meth.CG, dist = "correlation", method = "ward", plot = TRUE)

PCASamples(meth, screeplot = TRUE)
```

Differential Methylation

Now we look at differential methylation. Firstly we can just look at differentially methylated bases:

```
diff.CG <- calculateDiffMeth(meth.CG)
```

For multiple-cores we can use the num.cores option as well

```
## Not to be run
## diff.CG <- calculateDiffMeth(meth.CG, num.cores = 2)
```

For differential methylation, we look for bases that have a 25% change in methylation between samples, with a q-value of 0.01 or less. These are classed as hypermethylated. The exact opposite condition is hypomethylation. You can easily change the percentage difference or qvalue to whatever your needs

```
# get hyper methylated bases
myDiff25p.hyper <- get.methylDiff(diff.CG,
                                  difference = 25,
                                  qvalue = 0.01,
                                  type = "hyper")

# get hypo methylated bases
myDiff25p.hypo <- get.methylDiff(diff.CG,
                                 difference = 25,
                                 qvalue = 0.01,
                                 type = "hypo")

# get all differentially methylated bases
myDiff25p <- get.methylDiff(diff.CG,
                            difference = 25,
                            qvalue = 0.01)
```

Now what if we want to find differentially methylated "regions" rather bases

```
tiles <- tileMethylCounts(CG, win.size = 1000, step.size = 1000)
```

This can be then be fed into the unite() and calculateDiffMeth() functions to get differential methylation.

We can also visualize the distribution of hypo/hyper-methylated bases/regions per chromosome using the following function. In this case, the example set includes only one chromosome.

```
diffMethPerChr(diff.CG,
               plot = TRUE,
               qvalue.cutoff = 0.01,
               meth.cutoff = 25)
```

Annotating Differentially Methylated Sites/Regions

Ok we have some differentially methylated regions/bases etc. Now we can do some fancy annotations. We initially read in the gff bed file as a gene object:

```
gene.obj <- read.transcript.features("TAIR10_GFF3_genes.gff.bed")
```

and then we can just use the native functions within methylkit to produce some basic plots of differentially methylated regions that have 25percent differences (should be a lot!)

```
diffAnn <- annotate.WithGenicParts(myDiff25p, gene.obj)

getTargetAnnotationStats(diffAnn,
                         percentage = TRUE,
                         precedence = TRUE)

plotTargetAnnotation(diffAnn,
                     precedence = TRUE,
                     main = "differential methylation annotation")
```

Of course, once you use these objects, you are free to use other R packages for plotting more sophisticated figures and analyses!

Now try the other contexts and see what results you get