全基因组甲基化差异分析流程

李春晖 2021/7/9

DNA甲基化（DNA methylation）为DNA化学修饰的一种形式，能够在不改变DNA序列的前提下，改变遗传表现。DNA甲基化具体是指在DNA甲基化转移酶的作用下，在基因组CpG二核苷酸的胞嘧啶5号碳位共价键结合一个甲基基团。大量研究表明，DNA甲基化能引起染色质结构、DNA构象、DNA稳定性及DNA与蛋白质相互作用方式的改变，从而控制基因表达。 在成熟体细胞组织中，DNA甲基化一般发生于CpG双核苷酸（CpG dinucleotide）部位；而非CpG甲基化则于胚胎干细胞中较为常见。植物体内胞嘧啶的甲基化则可分为对称的CpG（或CpNpG），或是不对称的CpNpNp形式（C与G是碱基；p是磷酸根；N指的是任意的核苷酸）。DNA甲基化可以理解为基因组上的表观修饰，也就是说甲基化可以导致基因失活，去甲基化则代表基因的激活与表达。

DNA甲基化数据处理所使用的软件是Bismark

> https://github.com/FelixKrueger/Bismark

Bismark needs the following tools to be installed and ideally available in the `PATH` environment:

- Bowtie2 or HISAT2
- Samtools

（图1. 来源：https://github.com/FelixKrueger/Bismark ）

Bismark下载解压后不需要安装，在安装好依赖软件后，可以直接使用

step1

新建一个文件夹，在新建文件夹下把甲基化数据和要分析的物种基因组文件链接过来

> 能链接就链接，节省磁盘空间，链接操作［ln -S 源文件 目标文件］

```
mkdir Bismark
cd Bismark
mkdir 00.data 00.genome
#把甲基化原始数据.fq.gz链接到00.data文件夹下,把基因组fasta文件链接到00.genome下
```

step2

首先要对甲基化原始数据进行质控过滤，使用的软件是fastp，参数设置-n 0（不允许reads中出现gap）

> 一般从公司拿到的数据都是去过接头的clean data,所以一般只需要进行简单的质控处理

直接运行01.filter.py，会得到01.filter.py.sh文件

```
mkdir -p 01.filter
```

```
python3 01.filter.py
sh 01.filter.py.sh
#建议并行运行，质控步骤会花一些时间,用parallel命令时要考虑好资源占用情况，不然程序会被挤占掉
#parallel -j 12 < 01.filter.py.sh
```

如果要质控的数据有点多，建议加入pbs语句投递到后台去跑，可以节省时间
质控后的数据都在01.filter文件夹下，可以打开.json文件查看质控情况

step3

> bismark帮助文档：https://github.com/FelixKrueger/Bismark/tree/master/Docs

首先要对基因组构建索引

```
#直接给文件夹路径就行，程序会自动找到00.genome下的fasta文件
bismark_genome_preparation 00.genome
```

运行bismark，进行比对，该步骤比较耗时，一定要提交PBS到后台去跑

```
python3 02.bismark.py
#生成得到02.bismark.py.sh，提交PBS到后台运行
```

step4

对bismark比对BAM文件进行去重复

> This command will deduplicate the Bismark alignment BAM file and remove all reads but one which align to the the very same position and in the same orientation. This step is recommended for whole-genome bisulfite samples, but should not be used for reduced representation libraries such as RRBS, amplicon or target enrichment libraries.

```
python3 03.deduplicate.py
#生成得到 03.deduplicate.py.sh，提交PBS到后台运行
```

step5

从去重复后的比对文件中提取出甲基化信号

```
python3 04.extractor.py
#生成得到 04.extractor.py.sh，提交PBS到后台运行
```

bismark_methylation_extractor命令中一些参数设置的说明

```
bismark_methylation_extractor --parallel 10 --comprehensive --gzip -o 02.bismark/6-XPYJ --bedGr
#--parallel 设置10个并行，这里的10并不是10个线程的意思，具体可以看--help里面的说明
#--comprehensive 将4个可能甲基化链的结果合并到一起，分别生成CpG context、CHG context、CHH context三
#--gzip 输出文件进行.gz格式压缩
```

```
#-o 输出文件夹
#--bedGraph 将提取出的甲基化信号写入bedGraph文件，报告所有胞嘧啶的位置及其甲基化率，默认情况下只会输出
#--buffer_size 设置缓存区使用的内存大小
#--cytosine_report 全基因组甲基化报告
#--genome_folder 基因组fasta序列路径，要全路径
```

The methylation extractor output looks like this (tab separated):

1. seq-ID
2. methylation state
3. chromosome
4. start position (= end position)
5. methylation call

Methylated cytosines receive a `+` orientation, unmethylated cytosines receive a `-` orientation.

(图2. 来源：https://github.com/FelixKrueger/Bismark/tree/master/Docs)

step6

step5跑完之后，在输出文件目录下找到这个后缀的文件deduplicated.CpG_report.txt.gz

HiC_scaffold_1	91	+	0	4	CG	CGA
HiC_scaffold_1	92	-	0	2	CG	CGG
HiC_scaffold_1	94	+	0	4	CG	CGG
HiC_scaffold_1	95	-	0	2	CG	CGT
HiC_scaffold_1	99	+	0	4	CG	CGA
HiC_scaffold_1	100	-	0	2	CG	CGT
HiC_scaffold_1	107	+	0	4	CG	CGC

(图3. deduplicated.CpG_report.txt.gz文件内容)

每一列对应的内容为

<chromosome> <position> <strand> <count methylated> <count unmethylated> <C-context> <trinucleotide context>

需要对这个文件进行一定的处理得到如下格式

chrBase	chr	base	strand	coverage	freqC	freqT		
HiC_scaffold_1.91	HiC_scaffold_1	91	R	4	0.0	100.0		
HiC_scaffold_1.94	HiC_scaffold_1	94	R	4	0.0	100.0		
HiC_scaffold_1.99	HiC_scaffold_1	99	R	4	0.0	100.0		
HiC_scaffold_1.107	HiC_scaffold_1	107	R	4	0.0	100.0		
HiC_scaffold_1.181	HiC_scaffold_1	181	R	4	0.0	100.0		
HiC_scaffold_1.188	HiC_scaffold_1	188	R	4	0.0	100.0		
HiC_scaffold_1.343	HiC_scaffold_1	343	R	3	0.0	100.0		

（图4. 差异甲基化分析的输入文件格式内容）

```
#对所有deduplicated.CpG_report.txt.gz运行脚本05.2.cpg_report_change.py
python3 05.1.run_cov_change.py
#得到输出文件05.1.run_cov_change.py.sh
parallel -j 12 < 05.1.run_cov_change.py.sh
```

输出文件为deduplicated.CpG_report.txt.gz.filter.out

step7

用R包methyKit进行差异分析；

methylKit 是一个用于分析甲基化测序数据的R包，不仅支持WGBS，RRBS和目的区域甲基化测序，还支持oxBS-sq, TAB-seq等分析5hmc的数据。 其核心功能是差异甲基化分析和差异甲基化位点和区域的注释。

利用methylKit 做差异分析包括3步

1. 读取原始数据

- 纯文本格式，内容如下

chrBase	chr	base	strand	coverage	freqC	freqT		
HiC_scaffold_1.91	HiC_scaffold_1	91	R	4	0.0	100.0		
HiC_scaffold_1.94	HiC_scaffold_1	94	R	4	0.0	100.0		
HiC_scaffold_1.99	HiC_scaffold_1	99	R	4	0.0	100.0		
HiC_scaffold_1.107	HiC_scaffold_1	107	R	4	0.0	100.0		
HiC_scaffold_1.181	HiC_scaffold_1	181	R	4	0.0	100.0		
HiC_scaffold_1.188	HiC_scaffold_1	188	R	4	0.0	100.0		
HiC_scaffold_1.343	HiC_scaffold_1	343	R	3	0.0	100.0		

每一行是一个甲基化位点，coverage 代表覆盖这个位点的reads数，freqC 代表甲基化C的比例，freqT 代表非甲基化C的比例。这种纯文本格式内容非常直观，文件大小相比bam 文件小很多，读取的速度更快。纯文本格式的读取过程如下

```
library(methylKit)

file.list <- list(
    "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/4-YWJ/4-YWJ_1_bismark_bt2_pe.deduplicated.CpG_report.txt.gz.filter.out",
    "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/5-YWJ/5-YWJ_1_bismark_bt2_pe.deduplicated.CpG_report.txt.gz.filter.out",
    "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/6-YWJ/6-YWJ_1_bismark_bt2_pe.deduplicated.CpG_report.txt.gz.filter.out",
    "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/4-ZWJ/4-ZWJ_1_bismark_bt2_pe.deduplicated.CpG_report.txt.gz.filter.out",
    "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/5-ZWJ/5-ZWJ_1_bismark_bt2_pe.deduplicated.CpG_report.txt.gz.filter.out",
    "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/6-ZWJ/6-ZWJ_1_bismark_bt2_pe.deduplicated.CpG_report.txt.gz.filter.out")

myobj <- methRead(
    file.list,
    sample.id = list(
        "4YWJ_1_bismark_bt2_pe",
        "5YWJ_1_bismark_bt2_pe",
        "6YWJ_1_bismark_bt2_pe",
        "4ZWJ_1_bismark_bt2_pe",
        "5ZWJ_1_bismark_bt2_pe",
        "6ZWJ_1_bismark_bt2_pe"
        ),
    assembly = "hg19",
    treatment = c(1,1,1,0,0,0),
    context = "CpG"
    )
```

treatment参数指定样本的分组，0代表control组，1代表treatment组

assembly参数在查了官方文档后发现对结果没有影响，所以写hg19就可以了

2. 合并所有样本的数据

将所有样本的甲基化情况合并，得到所有样本的甲基化表达谱，用法如下

```
meth=unite(myobj, destrand=FALSE)
```

在合并的过程中，默认情况下，只有所有的样本都包含该位点时，才会保留，本质就是取的所有样本的交集，如果你想要取并集，可以修改min.per.group参数的值，该参数的值代表每组中至少有多少个样本覆盖该位点时才保留，如果设置为1，就是取并集。

```
meth.min=unite(myobj,min.per.group=1L)
```

3. 执行差异分析

通过calculateDiffMeth函数来执行差异甲基化分析，用法如下

```
myDiff=calculateDiffMeth(meth)
```

根据甲基化C是变多了还是变少了，可以将差异甲基化的结果分为两大类：

1. hypermethylated
2. hypomethylated

hypermethylated表示相比control组，treatment组中的甲基化C更多；hypomethylated则相反，表示treatment组中的甲基化C比control组中少。采用getMethylDiff函数提取差异分析的结果，用法如下

```
all <- getMethylDiff(myDiff,difference=30,qvalue=0.05)
head(all,10)
```

difference函数表明差异的阈值，只有差异大于该阈值时，才会保留，起始就是meth.diff的值，注意是绝对值大于difference的值。

除了difference阈值之外，还有qvalue阈值，小于该阈值的结果保留。在methylKit中，校正p值采用的是SILM算法，和我们常规的BH算法不同。type参数定义差异的类型，如果你只关注hypermethylated或者hypomethylated，可以设置type 参数的值，单独筛选。
在methylKit中，它的差异分析总是针对合并后的甲基化表达谱，如果你的甲基化表达谱每一行是一个甲基化位点，那么差异分析的结果就是差异甲基化位点；如果你的表达谱每一行是一个甲基化区域，那么差异分析的结果就是差异甲基化区域。上面的例子都是针对差异甲基化位点的，下面看下差异甲基化区域的分析。
首先遇到的问题就是甲基化区域如何界定，在methylKit中，按照滑动窗口的方式定义甲基化区域，默认窗口大小为10000 bp ， 步长为10000bp,通过tileMethylCounts函数实现。
完整的差异甲基化区域分析的代码如下：

```
library(methylKit)

file.list <- list(
   "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/4-YWJ/4-YWJ_1_bismark_bt2_pe.dedu
   "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/5-YWJ/5-YWJ_1_bismark_bt2_pe.dedu
   "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/6-YWJ/6-YWJ_1_bismark_bt2_pe.dedu
   "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/4-ZWJ/4-ZWJ_1_bismark_bt2_pe.dedu
   "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/5-ZWJ/5-ZWJ_1_bismark_bt2_pe.dedu
   "/public/home/lichunhui/project/baoyu/31.BSseq/02.bismark/6-ZWJ/6-ZWJ_1_bismark_bt2_pe.dedu

myobj <- methRead(
   file.list,
   sample.id = list(
           "4-YWJ_1_bismark_bt2_pe",
           "5-YWJ_1_bismark_bt2_pe",
           "6-YWJ_1_bismark_bt2_pe",
           "4-ZWJ_1_bismark_bt2_pe",
           "5-ZWJ_1_bismark_bt2_pe",
           "6-ZWJ_1_bismark_bt2_pe"
           ),
   assembly = "hg19",
   treatment = c(1,1,1,0,0,0),
   context = "CpG"
   )

regions <- tileMethylCounts(myobj,win.size=100,step.size=100)

meth <- unite(regions,destrand=FALSE)
head(meth)

myDiff <- calculateDiffMeth(meth)

```

```
all <- getMethylDiff(myDiff,difference=30,qvalue=0.05,type='all')
head(all)

write.table(all,file='/public/home/lichunhui/project/baoyu/31.BSseq/06.difmethy_regions100bp.ou
```

在单碱基差异甲基化和区域差异甲基化两种分析中，比较偏好区域差异甲基化，能够和基因区域进行关联
在后续的分析中就是将这些区域差异甲基化和基因gff关联起来，然后跑个GO富集和KEGG富集，如果有特别关注的基因，就特别查看这个基因位置附近的甲基化情况如何