测序深度可视化

笔记本:	本科毕业课题		
创建时间:	2019/8/15 10:33	**更新时间:**	2019/8/15 20:52
作者:	黄思源		

1. 求出每一个碱基上的测序深度

需要用到的文件：基因组bed文件

```
$ cat ~/project/ref/ref_v2.0/branap_v2.0.fa.bed
NC_027757.2	0	35822559
NC_027758.2	0	35332830
......
NC_027774.2	0	46347054
NC_027775.2	0	51699379
```

> 1.基因组的bed文件，第三列是染色体长度，只需要列出主要染色体，参考基因组文件可能还有一些没有定位的scaffold序列，这些不考虑
> 2.samtools faidx可以用来求染色体长度，基因组gtf或gff文件也可以看出染色体长度

用到的工具：samtools

```
$ nohup samtools depth -b ~/project/ref/ref_v2.0/branap_v2.0.fa.bed
xxx.smr.bam > bedbase_depth.txt &
#耗时较长，需要2小时以上

#结果查看
$ less bedbase_depth.txt | head
NC_027757.2	32	1
NC_027757.2	33	1
```

```
NC_027757.2	34	2
NC_027757.2	35	2
```

> 注意：该文件的第二列是0开始还是1开始（涉及到基因组中一些表示位置的文件格式，bed第一位是0，gff/gtf第一位是1）
> 怎么看？samtools tview xxx.smr.bam查看一下第一个碱基是在哪个位置即可（需要提前给bam文件建立索引：samtools index xxx.smr.bam）。结论是第一个碱基就在32位处，故是以1开始的。建议转换为bed坐标，即减1。

2. 将染色体拆分为若干个一定大小（比如50k）的窗口

> 得到bedbase_depth.txt文件之后，接下来的思路是：第二列除以50k结果取整，当作窗口的NO.编号（表示这个碱基属于第几个染色体的第几个窗口）。在同一条染色体上当窗口号相同时，第三列相加，当作该窗口的总碱基数。

因此需要知道每条染色体能分为多少个窗口，窗口起止位置和窗口编号的对应关系，用下面的脚本可以实现，得到的chr_to_50kb.final.bed文件是bed坐标的，即第一个位置记为0。

```
$ cat chr_to_50kb_window.pl
#! /usr/bin/perl
use warnings;
use strict;

my $parm_name = $ARGV[0];
open my $file_fh, "<", "$parm_name";
open my $file_fh2, ">", "chr_to_50kb.bed";

while (<$file_fh>) {
        chomp $_;
        my @one_line = (split("\t", $_));
        my $num = int($one_line[2]/50000);
        for(my $i=1; $i <= $num; $i++) {
                my $left = 50000*($i-1);
                my $right = 50000*($i);
                print $file_fh2 "$one_line[0]\t$left\t$right\n";
        }
        my $left2 = $num*50000;
        my $right2 = $one_line[2];
        print $file_fh2 "$one_line[0]\t$left2\t$right2\n";
}
close $file_fh;
close $file_fh2;
```

```
$ perl chr_to_50kb_window.pl branap_v2.0.fa.bed

$ head -n 5 chr_to_50kb.bed
NC_027757.2      0       50000
NC_027757.2      50000   100000
NC_027757.2      100000  150000
NC_027757.2      150000  200000
NC_027757.2      200000  250000

$ less chr_to_50kb.bed | awk '{print $1"\t"$2"\t"$3"\t"$2/50000+1}' >
chr_to_50kb.final.bed

$ head -n 2 chr_to_50kb.final.bed
NC_027757.2      0       50000   1
NC_027757.2      50000   100000  2

#第4列表示第几个窗口
```

3. 可能还需要更改染色体名称

如果染色体的名称不规范，这里需要更改，用到chr_name_change.txt，该文件中有两列，第一列是旧名称，第二列是新名称。原bed文件、chr_to_50kb.final.bed、base_depth.txt三个文件中的染色体名称都要改。

先建立染色体名称的对应关系

```
$ cat chr_name_change.txt
NC_027757.2      chrA1
NC_027758.2      chrA2
......
NC_027774.2      chrC8
NC_027775.2      chrC9
```

再执行脚本

```
$ cat 3.pl
#! /usr/bin/perl
use warnings;
use strict;

my $file1 = $ARGV[0];
my $file2 = $ARGV[1];
```

```
my %oldname_newname = ();
open my $file_fh, "<", "$file1" or die "can't open file '$file1'! $!";
while (<$file_fh>) {
    chomp $_;
    my @one_line = (split(/\t/,$_));
    $oldname_newname{$one_line[0]}=$one_line[1];
}
close $file_fh;

open my $file_fh2, "<", "$file2";
while (<$file_fh2>) {
    chomp $_;
    my @one_line = (split("\t",$_));
    my $one_line_maxindex = $#one_line;
    my $outline = $oldname_newname{$one_line[0]};
    for (my $i = 1;$i <= $one_line_maxindex; $i++) {
        $outline .= "\t$one_line[$i]";
    }
    print "$outline\n";
}
close $file_fh2;

$perl 3.pl chr_name_change.txt chr_to_50kb.final.bed >
chr_to_50kb.final.rename.bed
$perl 3.pl chr_name_change.txt branap_v2.0.fa.bed > branap_v2.0.fa.rename.bed
$perl 3.pl chr_name_change.txt bedbase_depth.txt > bedbase_depth.rename.txt
```

4. 求出每一个窗口中的每一个碱基测序深度之和

将bedbase_depth.rename.txt每一行转换成染色体编号_窗口号 深度 的格式

```
$ awk '{print $1"_"int(($2-1)/50000)+1"\t"$3}' bedbase_depth.rename.txt >
tmp1
$ head tmp1
chrA1_1 1
chrA1_1 1
chrA1_1 2
chrA1_1 2
```

接下来用perl脚本结合哈希表可求出窗口上总的碱基数，除以50000就是平均测序深度了。

```
$ cat 1.pl
#! /usr/bin/perl
use warnings;
use strict;

my %chuangkou_jianji=();
open my $file_fh, "<", "tmp1";
while (<$file_fh>) {
    chomp $_;
    my @one_line = (split("\t",$_));
    if (exists $chuangkou_jianji{$one_line[0]}) {
        $chuangkou_jianji{$one_line[0]} = $chuangkou_jianji{$one_line[0]} +
$one_line[1];
    } else {
        $chuangkou_jianji{$one_line[0]} = $one_line[1];
    }
}
close $file_fh;

open my $file_fh2, ">", "tmp2";
foreach my $i ( sort keys %chuangkou_jianji) {
    print $file_fh2 "$i\t$chuangkou_jianji{$i}\n";
}
close $file_fh2;

$perl 1.pl
#有些慢

$ head -n 5 tmp2
chrA1_1 510848
chrA1_10        913890
chrA1_100       1149640
chrA1_101       1268166
#这时再看看tmp1的内容就更好理解了。注意上面tmp2并不是按照窗口编号由小到大排
列的
```

5. 求出每一个窗口中的平均测序深度

格式是：染色体编号 窗口号 平均测序深度

```
$ awk -F "_|\t" '{print $1"\t"$2"\t"$3/50000}' tmp2 | sort -k1,1 -k2,2n >
tmp3
```

```
$ head -n 3 tmp3
chrA1    1        11.8568
chrA1    2        10.217
chrA1    3        12.202
```

6. 将窗口号替换为窗口的染色体bed坐标

格式是：染色体编号 窗口起始位置 窗口终止位置 平均测序深度
先创建窗口 和窗口位置 的对应关系

```
$ less chr_to_50kb.final.rename.bed | awk '{print $1"-"$4"\t"$2"-"$3}' > tmp4

$ head -n 2 tmp4
chrA1-1 0-50000
chrA1-2 50000-100000
```

再执行脚本

```
$ cat 2.pl
#! /usr/bin/perl
use warnings;
use strict;

my %chuangkou_weizhi=();
open my $file_fh1, "<", "tmp4";
while (<$file_fh1>) {
    chomp $_;
    my @one_line=(split("\t",$_));
    $chuangkou_weizhi{$one_line[0]}=$one_line[1];
}
close $file_fh1;

open my $file_fh2, "<", "tmp3";
open my $file_fh3, ">", "tmp5";
while (<$file_fh2>) {
    chomp $_;
    my @one_line = (split("\t",$_));
    my $chuangkou = $one_line[0]."-".$one_line[1];
    print $file_fh3
"$one_line[0]\t$one_line[1]\t$chuangkou_weizhi{$chuangkou}\t$one_line[2]\n";
}
```

```
close $file_fh3;

$perl 2.pl

$ awk -F "\t|-" '{OFS="\t";$1=$1;print $0}' tmp5 | cut -f1,3,4,5 > tmp6
$ head -n 2 tmp6
chrA1    0        50000    11.8568
chrA1    50000    100000   10.217
```

上述tmp6仍然是bed坐标

> 由于高度同源序列和重复序列的存在，某些区域的深度会非常高，建议人为调整以便于呈现。如，将平均深度大于100的区域深度全重新定义为100。这里的100需要根据整体的测序深度来定，因为前面可以看出窗口的平均测序深度是10+或者20+，这里才定的100。如果本身平均测序深度比较大，30+、40+的样子，这个阈值需要调整，比如150。

```
$ awk '{if($4 > 100) print $1"\t"$2"\t"$3"\t100"}{if($4 <= 100) print $0}'
tmp6 > depth_distribution.bed

$ head -n 2 depth_distribution.bed
chrA1    0        50000    11.8568
chrA1    50000    100000   10.217
```

最终会得到两个文件一个是测序深度分布文件，另一个是更改染色体名称后的bed文件，在RStudio中画图会用到这两个文件。后续画图可以在自己的Windows电脑上面操作，所以需要将上述两个生成的文件传到Windows电脑上去。并且测序深度分布文件需要在第一行加上`chr start end y1`

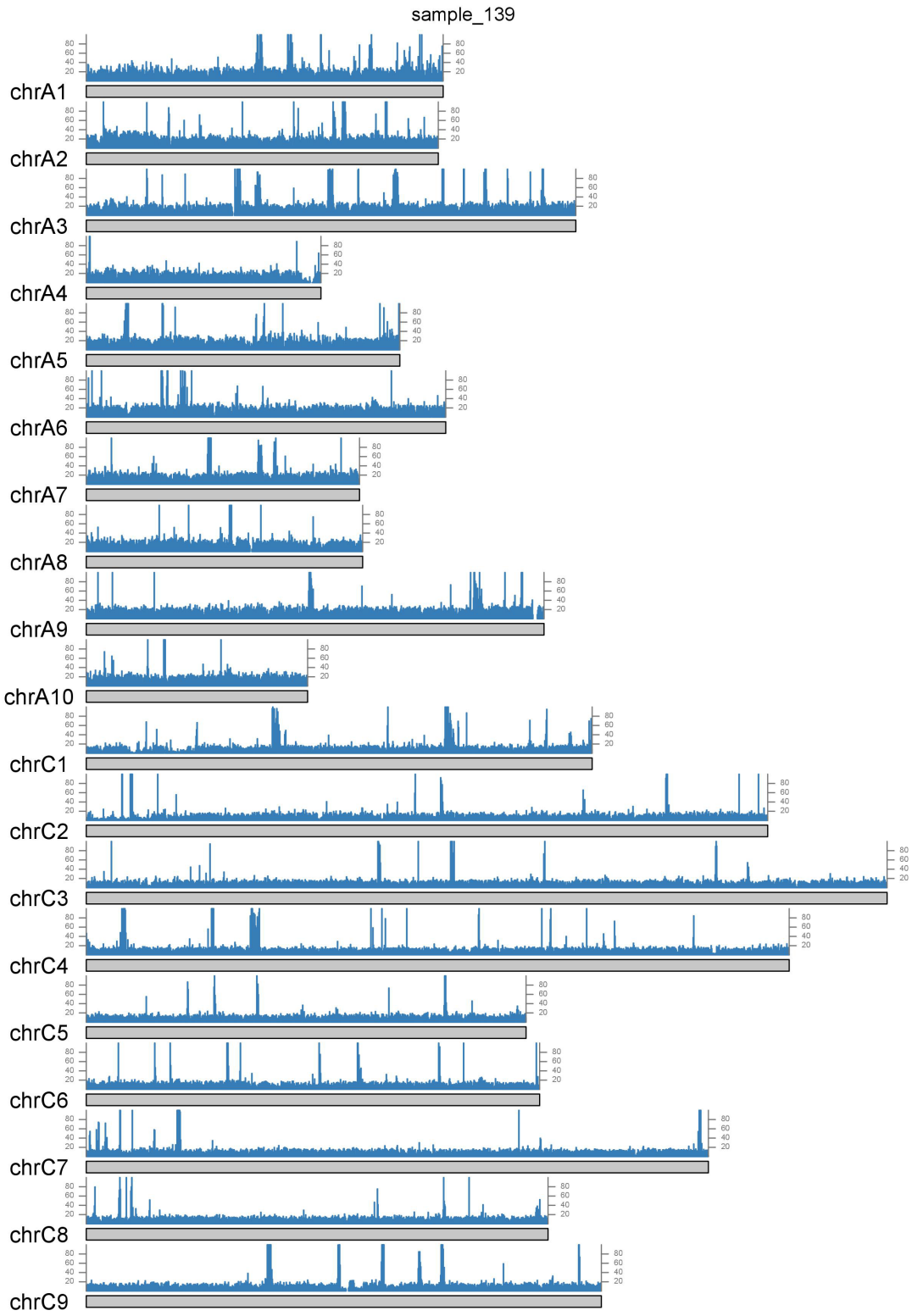
7. 使用R包绘图

R包：karyoploteR
branap_v2.0.fa.bed染色体名称更改后为branap_v2.0.fa.`rename` .bed
假设样本名称为`sample_139`

```
#R代码
###导入文件，需要用到两个文件
chr_bed <- "branap_v2.0.fa.rename.bed"
depth_bed <- "depth_distribution.bed"
sample_name <- "sample_139"
###
```

```
###安装加载R包
a <- installed.packages()
b <- a[,"Package"]
pkg <- setdiff(c("karyoploteR"), b)

if (length(pkg) == 1) {
  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

  BiocManager::install("karyoploteR")
}

library(karyoploteR)
rm(list=c("a","b","pkg"))
###



Bn.genome <- toGRanges(chr_bed)
kp <- plotKaryotype(genome=Bn.genome,labels.plotter =
NULL,plot.type=1,main=sample_name)
kpAddChromosomeNames(kp, cex = 1.2) #cex调整字的大小
sample_bar <- toGRanges(depth_bed)
kpBars(kp, sample_bar,border="#377EB8",data.panel = 1,r0 = 0,r1 = 1,ymin =
0,ymax = 100) #此处的100选自前面数据整理过程中的最大深度为100



#在图形左右添加纵坐标轴，可以结合图形微调
kpAxis(kp, data.panel = 1, ymin = 0, ymax=100, tick.pos = c(20,40,60,80),
labels = c("20","40","60","80"),tick.len = 700000, label.margin = -600000,
side = 2, cex = 0.4, col="#777777")
kpAxis(kp, data.panel = 1, ymin = 0, ymax=100, tick.pos = c(20,40,60,80),
labels = c("20","40","60","80"),tick.len = 700000, label.margin = -600000,
side = 1, cex = 0.4, col="#777777")
```

图片呈现如下


sample_139
chrA1
chrA2
chrA3
chrA4
chrA5
chrA6
chrA7
chrA8
chrA9
chrA10
chrC1
chrC2
chrC3
chrC4
chrC5
chrC6
chrC7
chrC8
chrC9