# Perl和R编程实现fastqc软件的部分功能

fastqc这个软件是一个java软件，但是有些人服务器没有配置好这个java环境，导致无法使用，这里我们用简单的perl脚本和R实现fastqc的部分功能。

统一测试文件是illumina的phred33格式的fastq文件，共100000/4=25000条reads，读长都是101个碱基

目录

[Perl和R编程实现fastqc软件的部分功能 1](#_Toc467086902)

[1 计算每条reads每个碱基的Q值 1](#_Toc467086903)

[程序名 1](#_Toc467086904)

[使用命令 1](#_Toc467086905)

[功能 1](#_Toc467086906)

[统计结果 1](#_Toc467086907)

[2 计算每条reads的平均Q值 2](#_Toc467086908)

[程序名 2](#_Toc467086909)

[使用命令 2](#_Toc467086910)

[功能 2](#_Toc467086911)

[统计结果 2](#_Toc467086912)

[3 统计每个位点质量的五分位数 2](#_Toc467086913)

[程序名 2](#_Toc467086914)

[使用命令 2](#_Toc467086915)

[功能 2](#_Toc467086916)

[统计结果 2](#_Toc467086917)

[4 统计每条reads的GC含量 3](#_Toc467086918)

[程序名 3](#_Toc467086919)

[使用命令 3](#_Toc467086920)

[功能 3](#_Toc467086921)

[统计结果 4](#_Toc467086922)

[5 R画Q值和GC含量图 4](#_Toc467086923)

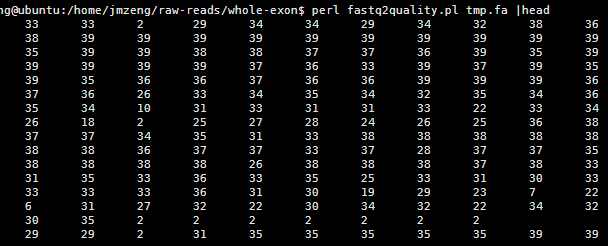
## 1 计算每条reads每个碱基的Q值

程序名-fastq2quality.pl

使用命令**：**perl fastq2quality.pl SRR504517\_1.fastq >quality.txt

功能**：** 把fastq格式的每条原始reads的第四行ascii码质量值，转换为Q值并输出一个矩阵，有多少条reads就有多少行，每条reads的碱基数就是列数。

统计结果**：**

**[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的部分功能-上817.png)**

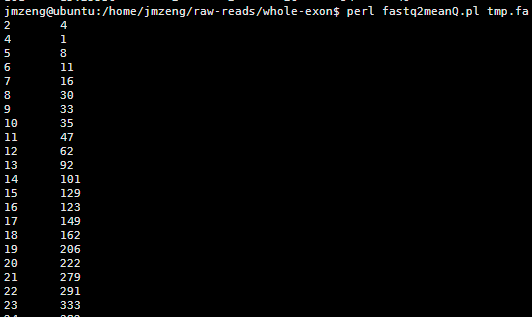
## 2 计算每条reads的平均Q值

程序名-fastq2meanQ.pl

使用命令：perl fastq2meanQ.pl SRR504517\_1.fastq

功能： 把fastq格式的原始reads统计每条reads的平均Q值，并画出Q值1到50各有多少条reads的分布图

统计结果：

**[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的部分功能-上1438.png)**

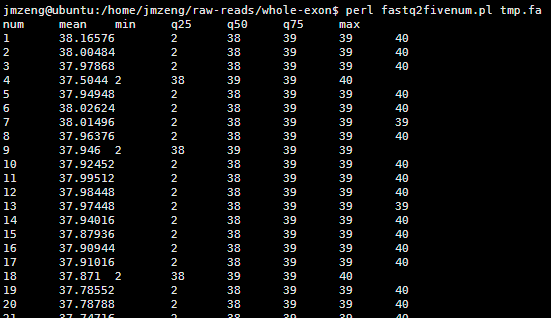
## 3 统计每个位点质量的五分位数

程序名-fastq2fivenum.pl

使用命令：perl fastq2fivenum.pl  SRR504517\_1.fastq

功能**：**把fastq格式的每条原始reads的第四行ascii码质量值，转换为Q值：，并对每一个位点统计所有reads的四分位数，加上平均数。

统计结果：

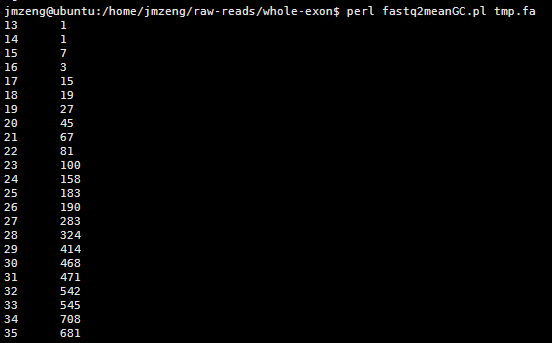
**[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的部分功能-上3024.png)**

## 4 统计每条reads的GC含量

程序名-fastq2meanGC.pl

使用命令：perl fastq2meanGC.pl SRR504517\_1.fastq

功能： 把fastq格式的原始reads统计每条reads的平均Q值，并画出Q值1到50各有多少条reads的分布图

统计结果：  
[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的部分功能-上3633.png)

## 5 R画Q值和GC含量图

a=read.table(“meanQ.txt”)

看看数据结构如下

> head(a)

V1    V2

1  2 93879

2  3 17800

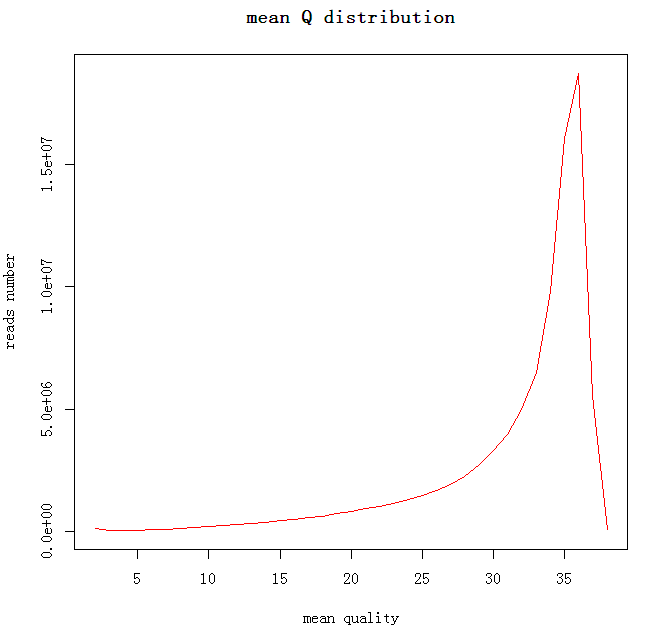
3  4 25295

4  5 33259

5  6 55685

6  7 84866

plot(a,type=’l’,col=’red’,ylab=’reads number’,xlab=’mean quality’,main=’mean Q distribution’)

[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的一些功能-下-R代码755.png)

可以看出绝大部分的reads的Q值都在30-35直接，也就是说本次测序挺符合要求的，但是还是需要对那些平均Q20以下的reads过滤掉。

a=read.table(‘meanGC.txt’)

看看数据结构如下

> head(a)

V1  V2

1  0 503

2  1 151

3  2 163

4  3 179

5  4 315

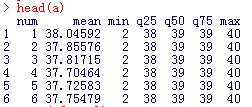
6  5 443

plot(a,type=’l’,col=’red’,ylab=’reads number’,xlab=’reads bp’,main=’GC% distribution’)

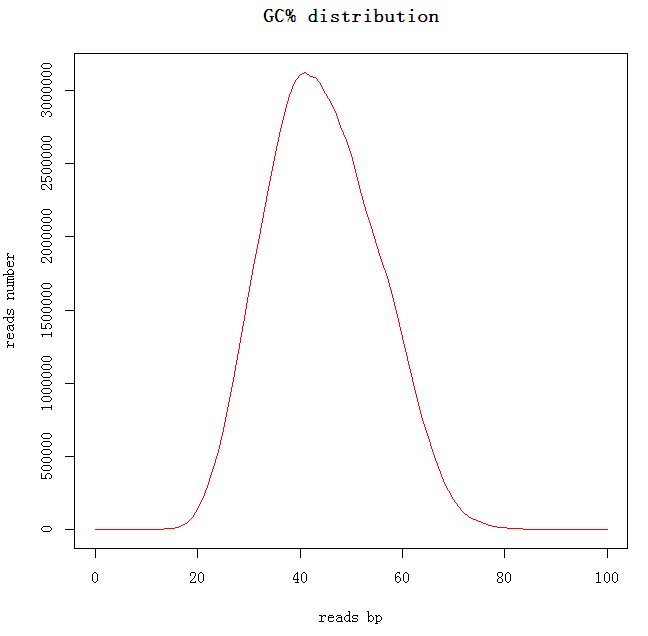
可以看出GC含量的分布看起来挺符合正态分布的，大部分reads的GC含量都是在40%-60%直接

a=read.table(‘fivenum.txt’,header=T)

看看数据结构如下

[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的一些功能-下-R代码1405.png)

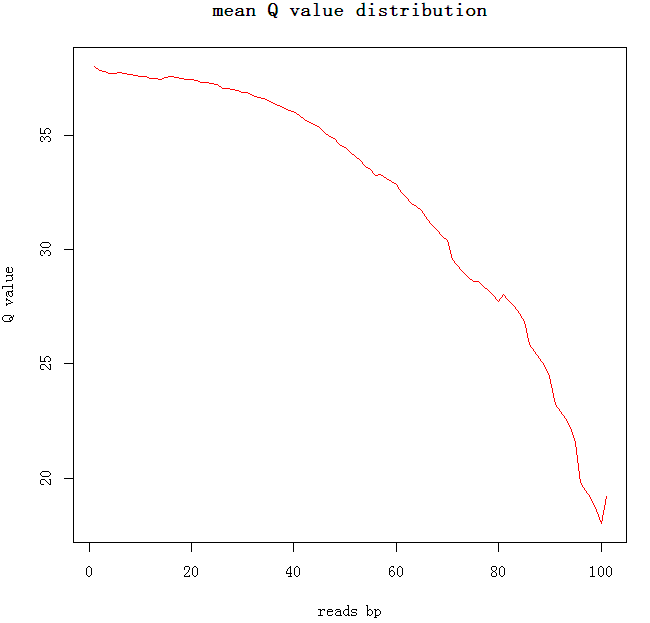
boxplot(t(a[,3:7]),xlab=’reads bp’,ylab=’Q value’,main=’mean Q boxplot’)

[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的一些功能-下-R代码1160.png)

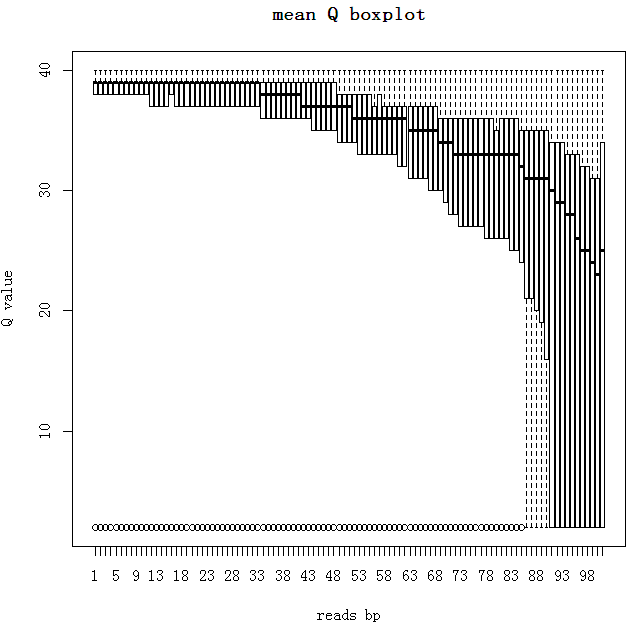
可以看出测序质量从1-100bp过去质量越来越差，但是大部分都是高于Q30，但是88bp之后的碱基测序质量不咋地，可能需要trim掉

对于这个数据还可以画一个图

plot(a[,1:2],type=’l’,col=’red’,ylab=’Q value’,xlab=’reads bp’,main=’mean Q value distribution’)

[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的一些功能-下-R代码1987.png)

可以看到88bp之后的平均Q值小于30，根据我们的阈值可能要把所有的reads的后面约10个bp的碱基要trim掉

[](http://www.bio-info-trainee.com/wp-content/uploads/2015/03/仿写fastqc软件的一些功能-下-R代码1676.png)