## Assignment 3 Author: “LuchaoQi” Email: [lqi9@jhu.edu](mailto:lqi9@jhu.edu)

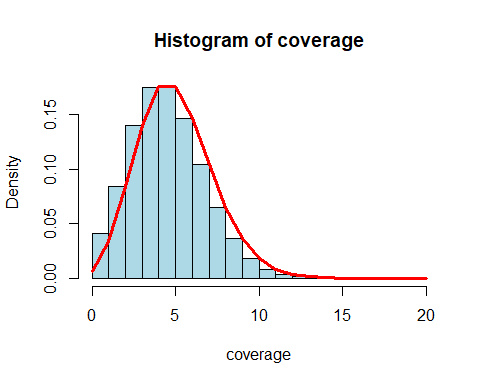
### Q1a.

We need 100bp reads.

### Q1b.

Use following R code to simulate 5x coverage of a 1Mbp genome:

set.seed(100)  
s = 1000000  
n = 5  
a = sample(1:s, n\*s,replace=TRUE)  
coverage = rep(0,s)  
for (i in a ){coverage[i] = coverage[i]+1}  
hist(coverage,prob=T,col="light blue")  
xfit<-seq(min(coverage),max(coverage))  
yfit<-dpois(xfit,n)  
lines(xfit,yfit,col="red",lwd=3)



length(which(coverage==0))

## [1] 6871

mean(coverage)

## [1] 5

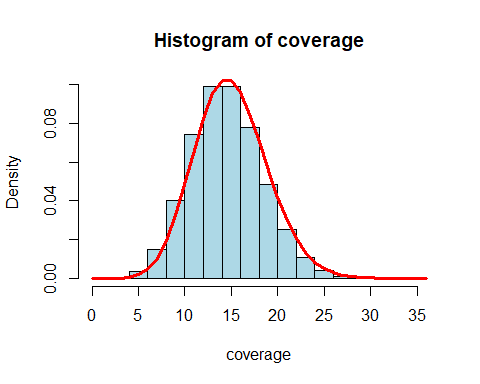
### Q1c.

Theoretically, the Poisson expecation should be the value of coverage:5, which equals to exactly the mean of our simulations.

### Q1d.

Use following R code to simulate 15x coverage:

set.seed(100)  
s = 1000000  
n = 15  
a = sample(1:s, n\*s,replace=TRUE)  
coverage = rep(0,s)  
for (i in a ){coverage[i] = coverage[i]+1}  
hist(coverage,prob=T,col="light blue")  
xfit<-seq(min(coverage),max(coverage))  
yfit<-dpois(xfit,n)  
lines(xfit,yfit,col="red",lwd=3)



length(which(coverage==0))

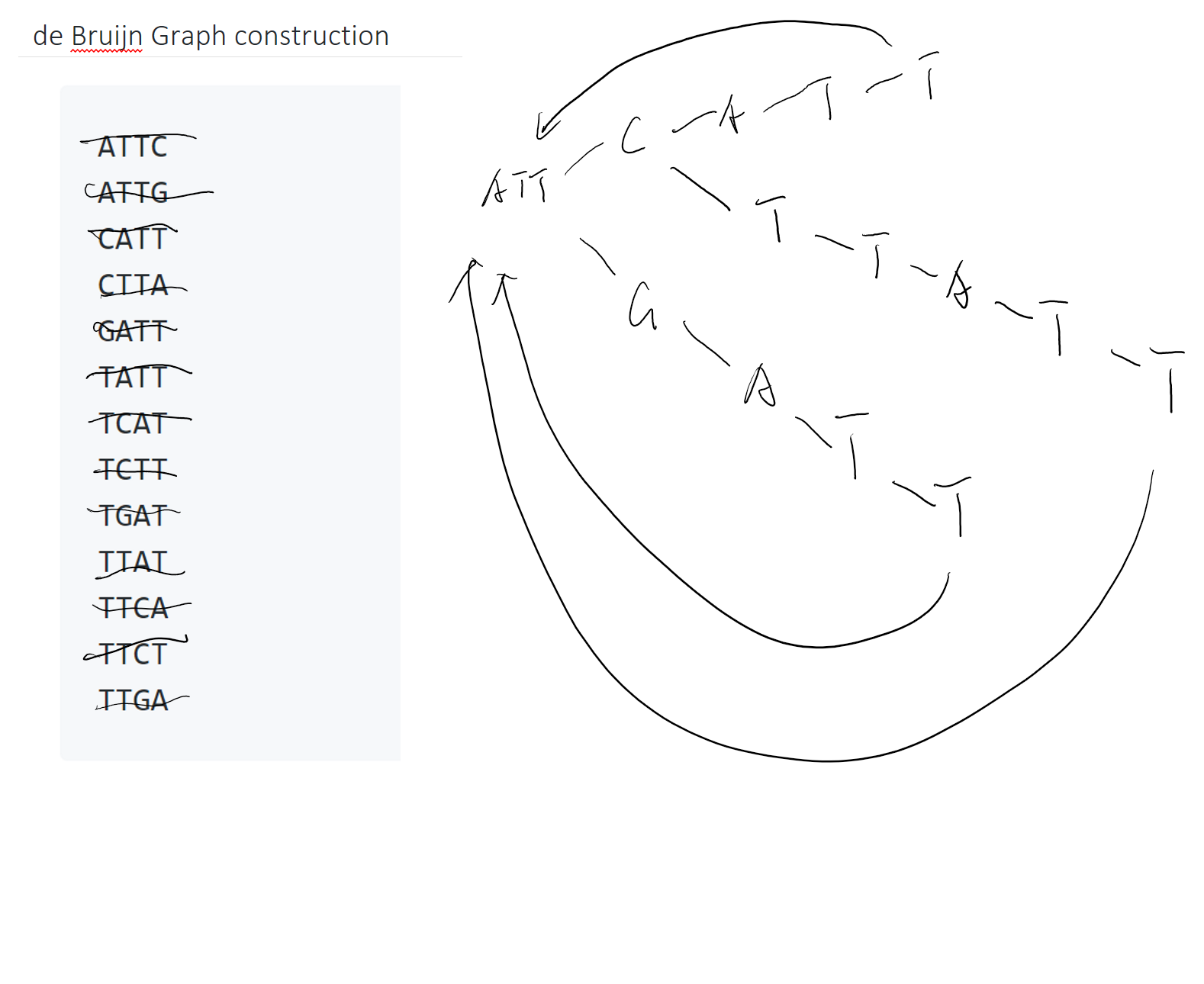
## [1] 1

mean(coverage)

## [1] 15

Theoretically, the Poisson expecation should be the value of coverage: 15, which equals to exactly the mean of our simulations.

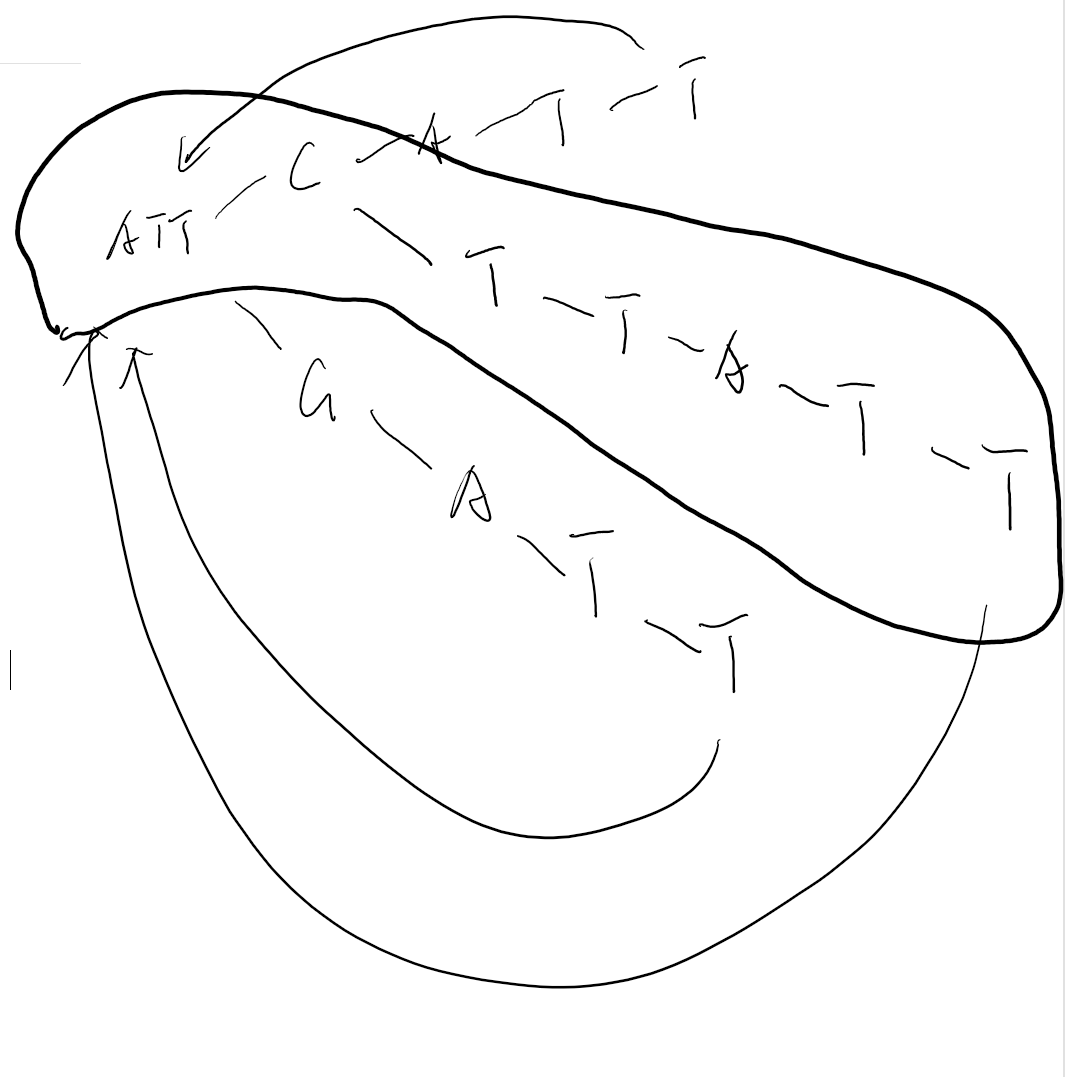
### Q2a.



### Q2b.

One possible genome sequence could be:  
ATTCATTCTTATTG

### Q2c.

The longest repeat should be:  
ATTCTTATT  


### Q3a.

Through following code, We could see the results ouputted at the end of code.

def computeBWT(s):  
 s = s +'$'  
 rows = sorted(s[i:] + s[:i] for i in range(len(s)))  
 bwt = [row[-1:] for row in rows]  
 print("".join(bwt))  
 return "".join(bwt)  
computeBWT('I\_am\_fully\_convinced\_that\_species\_are\_not\_immutable;\_but\_that\_those\_belonging\_to\_what\_are\_called\_the\_same\_genera\_are\_lineal\_descendants\_of\_some\_other\_and\_generally\_extinct\_species,\_in\_the\_same\_manner\_as\_the\_acknowledged\_varieties\_of\_any\_one\_species\_are\_the\_descendants\_of\_that\_species.\_Furthermore,\_I\_am\_convinced\_that\_natural\_selection\_has\_been\_the\_most\_important,\_but\_not\_the\_exclusive,\_means\_of\_modification.')

## .etsense\_\_$.,eIIrftassrse,;emyleeymedntt,ee,fetetssssyeeelftttedfdtsetndntgdort\_ercr\_\_ss\_metdd\_\_\_\_\_vh\_hhhhhcn\_\_a\_\_\_innsseeeeax\_\_neneeeenn\_\_eorvlhsrhhmrmhrmhhnrnmpppplcclglbsbeccgghnnnhiiiiiddi\_\_ooooi\_nd\_\_n\_wtttttttttttttfcccctrd\_\_\_vvtlgttscaabelw\_auaecllaa\_aao\_i\_r\_imooei\_iiiaeeoineeioa\_\_kaaaaooatm\_\_\_\_sii\_lccmphmnn\_nssssmeeeueoaaaaaeoueeeeaentta\_\_eeo\_u\_\_\_\_\_onaosaucaauoaur\_\_\_\_\_\_\_\_\_\_or\_exac\_nnaftFlbbm\_inn\_oeelln

### Q4.

Through following code, We could see the results ouputted at the end of code.

def decodeBWT(r):  
 rows = [""] \* len(r)  
 for i in range(len(r)):  
 rows = sorted(r[i] + rows[i] for i in range(len(r)))  
 s = [row for row in rows if row.endswith("$")][0]  
 print(s.rstrip("$").strip())  
 return s.rstrip("$").strip()  
decodeBWT('.uspe\_gexr\_\_\_\_\_\_\_$..,e.orrs,sdddeedkdsuoden-tf,tyewtktttt,sewteb\_ce\_\_ww\_\_h\_PPsm\_u\_naseueeennnrrlmwwhWcrskkmHwhttv\_no\_nnwttzKt\_l\_ocoo\_be\_\_\_aaaooaAakiiooett\_oooi\_sslllfyyD\_\_uouuueceetenagan\_\_\_rru\_aasanIiatt\_\_c\_\_saacooor\_ootjeae\_\_\_\_\_\_ir\_\_a')

## We\_went\_up,\_saw\_the\_structure,\_we\_came\_back\_to\_Kings\_and\_looked\_at\_our\_Pattersons,\_and\_every\_section\_of\_our\_Pattersons\_we\_looked\_at\_screamed\_at\_you,\_Double\_Helix.\_And\_it\_was\_just\_there.\_-\_once\_you\_knew\_what\_to\_look\_for.\_It\_was\_amazing.