

# ***De novo* whole genome assembly**

## **Lecture 1**

**Qi Sun**

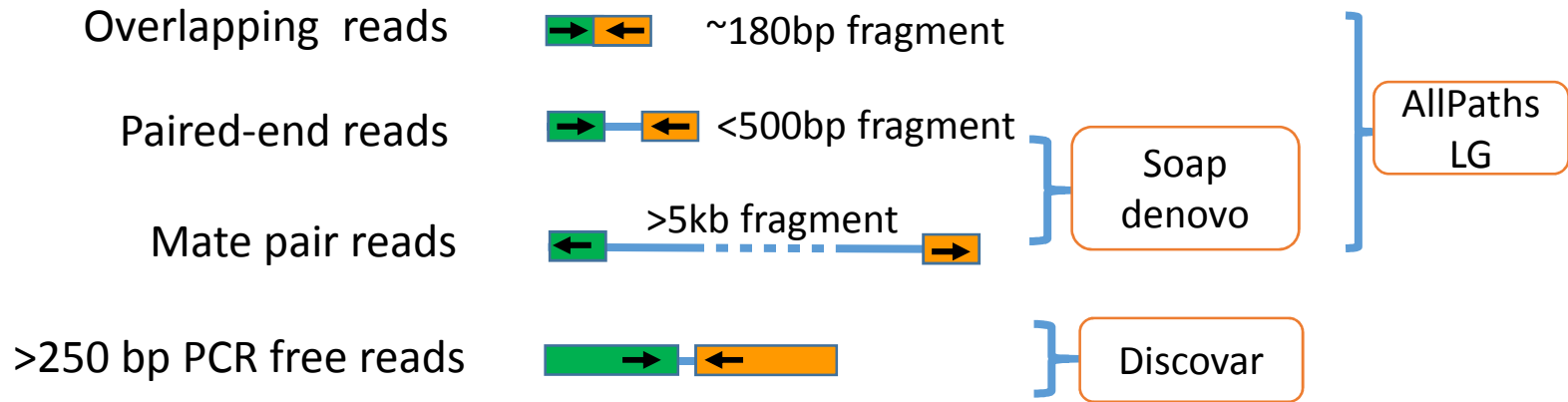
**Minghui Wang**

**Bioinformatics Facility**

**Cornell University**

# DNA Sequencing Platforms

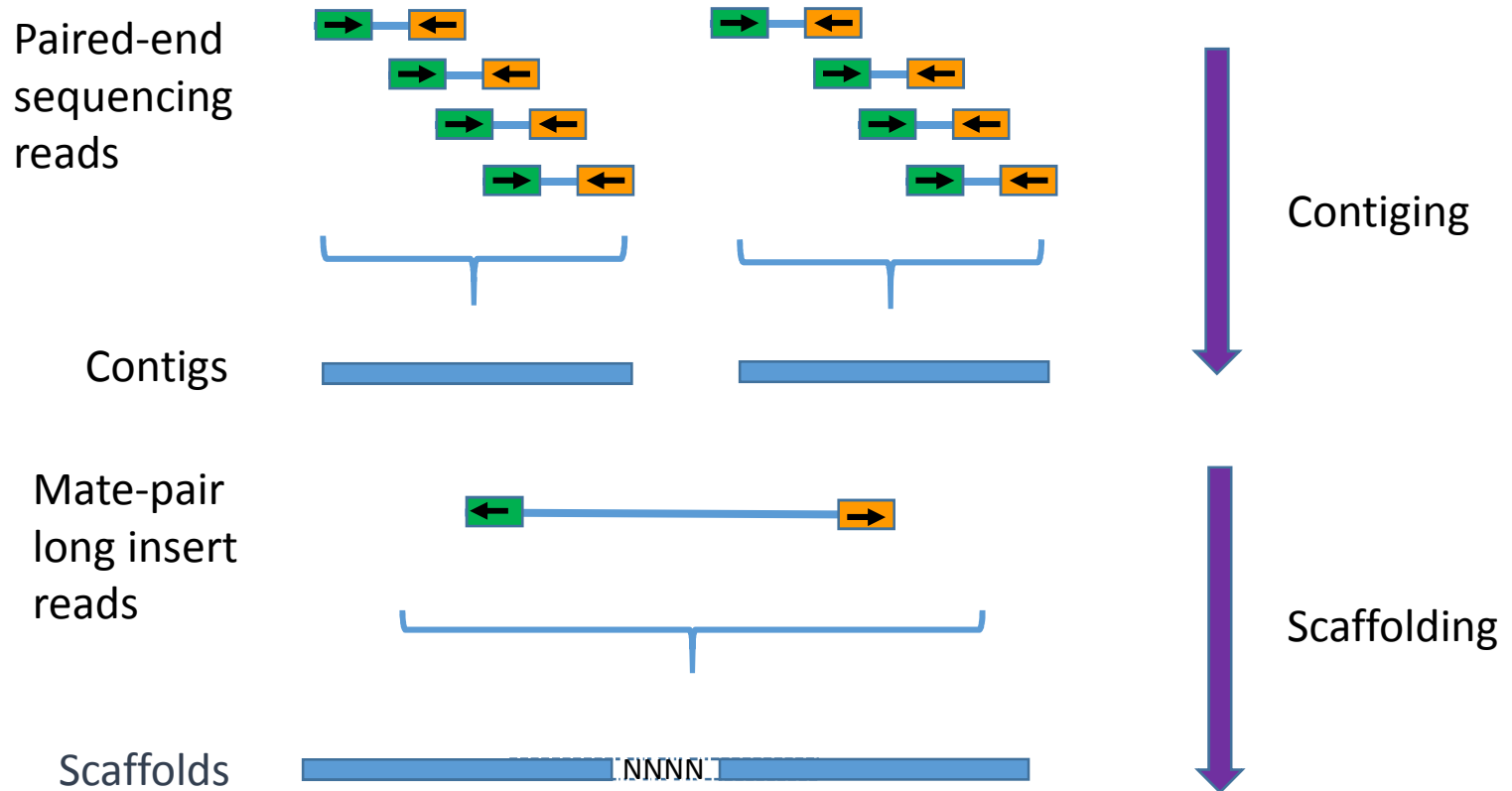
## Illumina sequencing (100 to 300 bp reads)



## PacBio long-read (10 – 15 kb reads)

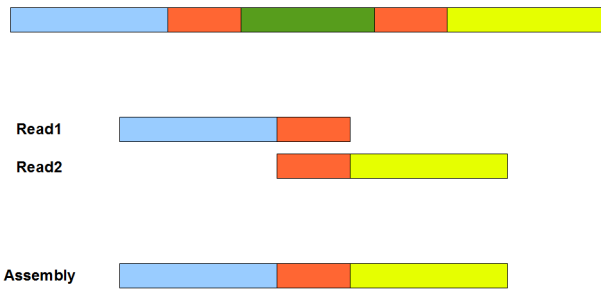


# *De novo* whole genome assembly



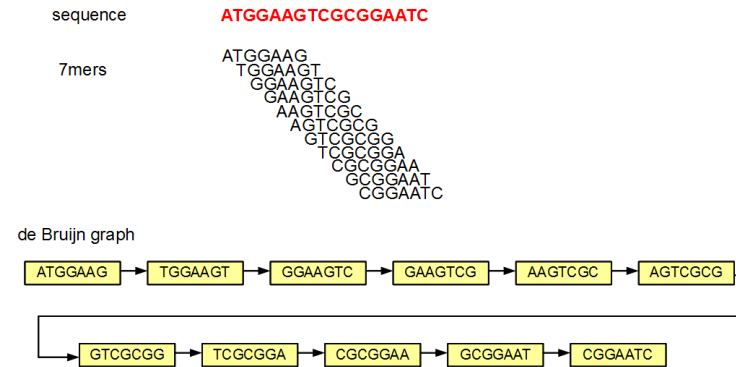
## Two categories of contiging strategies

### Long reads overlap–layout–consensus



Celera assembler, 454 assembler

### Short reads de-bruijn-graph



Velvet, Soap-denovo, Abyss, Trinity et al

## de-bruijn-graph for contiging short reads

sequence

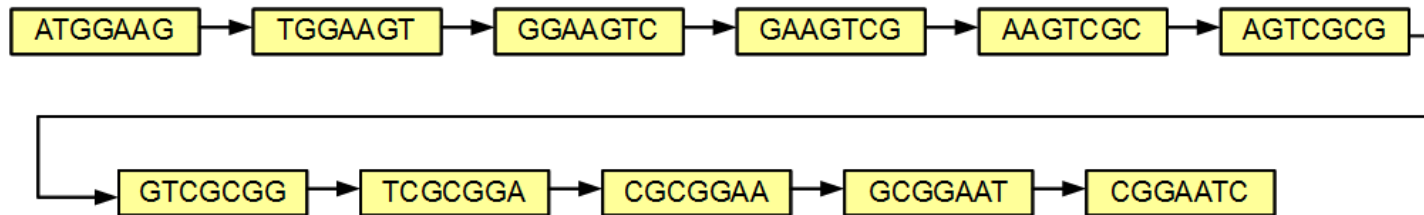
**ATGGAAGTCGCGGAATC**

7mers

ATGGAAG  
TGGAAGT  
GGAAGTC  
GAAGTCG  
AAGTCGC  
AGTCGCG  
GTCGCGG  
TCGCGGA  
CGCGGAA  
GCGGAAT  
CGGAATC

Kmers

de Bruijn graph

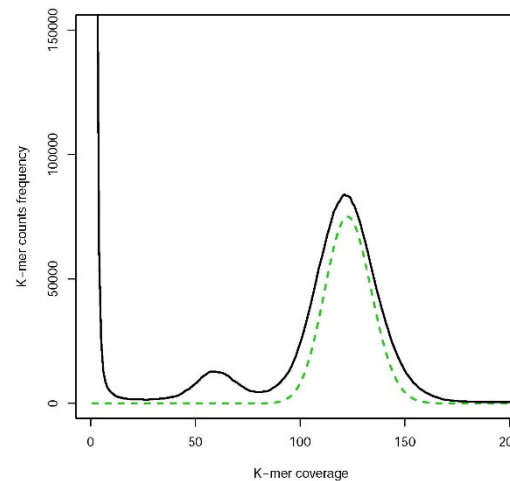
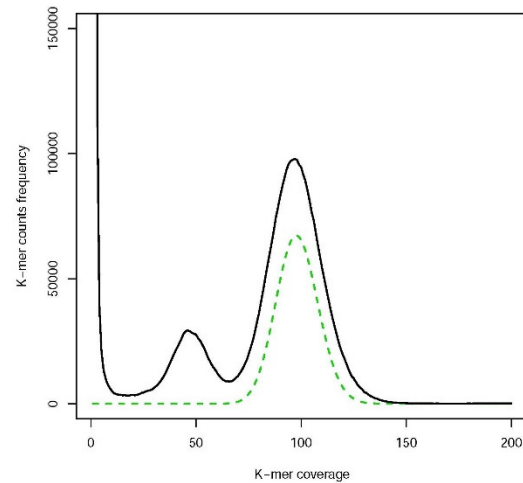


## Kmer counting from sequencing reads

AATTTGTGAATGGCT	1
ATGCAAGACCAATTG	128
ACAAATAGGGGTAGT	1
AACACAAAATAATAA	1
GCTGTAAGCTTTAAC	1
AACCTATGAGTGAAA	1
ATTCGTAGATCTTCC	101
CAGGTATCAGGCATG	146
TATCAGCTGGAGTCA	60
ACCAGAGATATAAAA	1
GGTCCATTTTGTTTA	1
GTCAAAAATTACGA	1
ACATTCCCTCGGTAA	1
AGCATTTCTTAACAC	1
TAAAACCATCTTTA	137
ATTCGATTTGTCAAG	62
ACATTGGAAAAAACT	1
AATGGAAATACAATA	2
...	
CATTCGT <b>T</b> AGATCAA	1
...	
CATTCGT <b>G</b> AGATCAA	128
...	
CCTTGTCATTAACTC	667
...	

(15mers)

## Kmer depth distribution plots



# Impact of kmer-length (1)

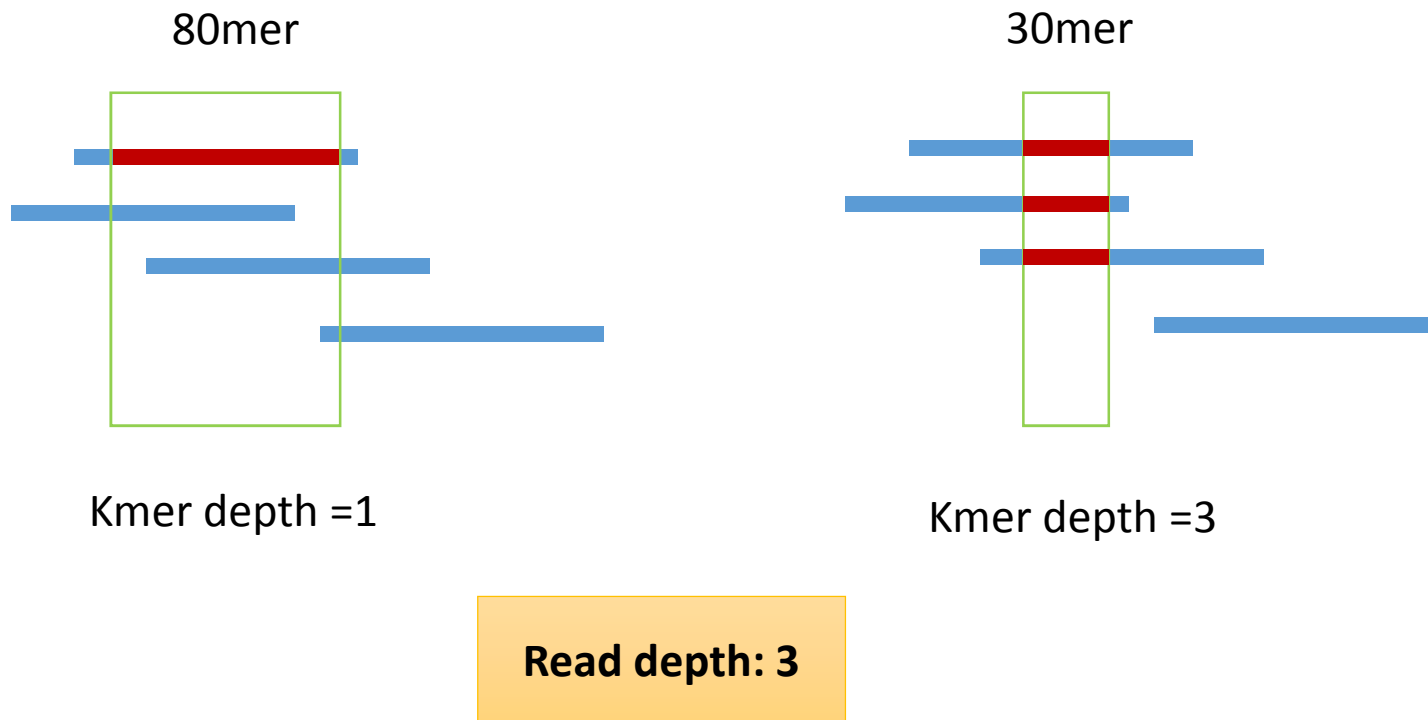
Short kmers would collapse paralogous regions, and result in more crossovers

Paralogous regions      **G**GATGGAAG**TCG**..... **C**GATGGAAG**GAT**



## Impact of kmer-length (2)

### Read depth vs Kmer depth

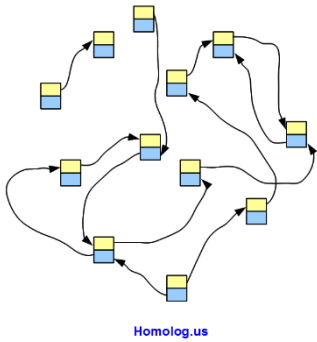


**Longer kmers could results in too little kmer depth**



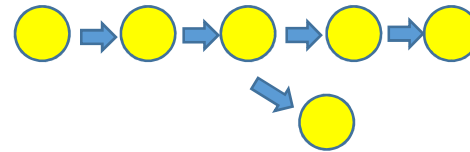
# Sequencing errors and Tips, bubbles and crosslinks

De Bruijn Graphs for NGS Assembly

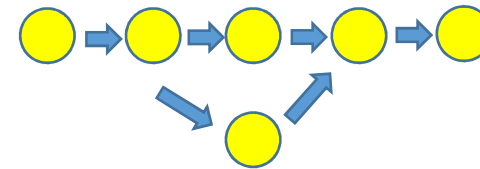


source: <http://www.homolog.us/Tutorials/index.php>

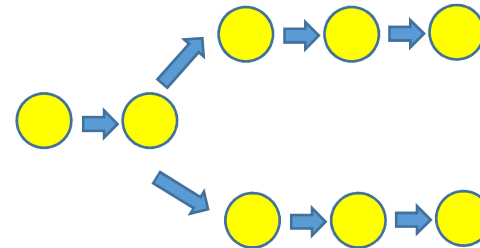
Tips



Bubbles



Crosslinks



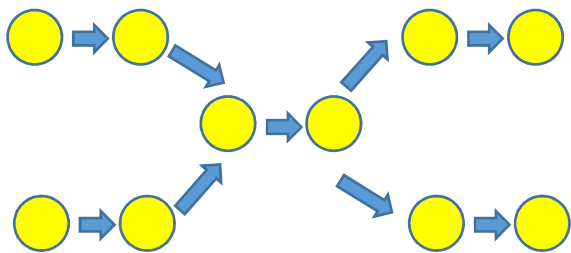
## Deal with sequencing errors and repetitive regions

### 1. Sequencing errors

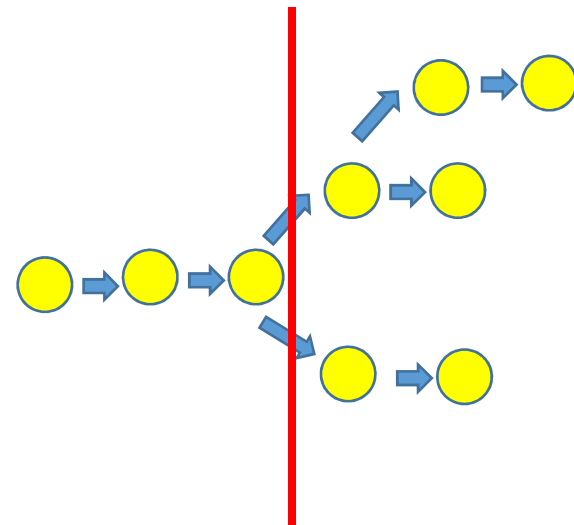
- Remove low depth kmers in a bubble;
- Too long kmers would cause coverage problem;

### 2. Repetitive region

- Longer kmers

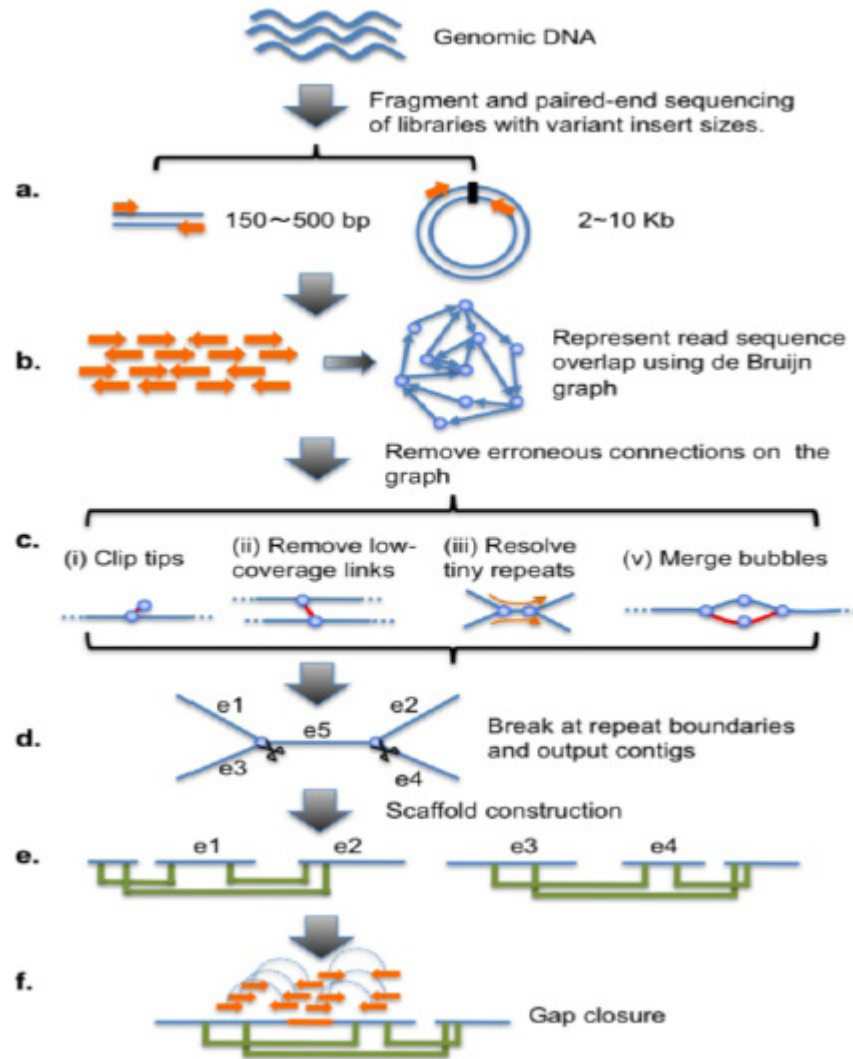


Tiny repeat:  
Separate the path



Break boundary between low  
and high copy regions

# Examine an assembly software 1. Soap denovo



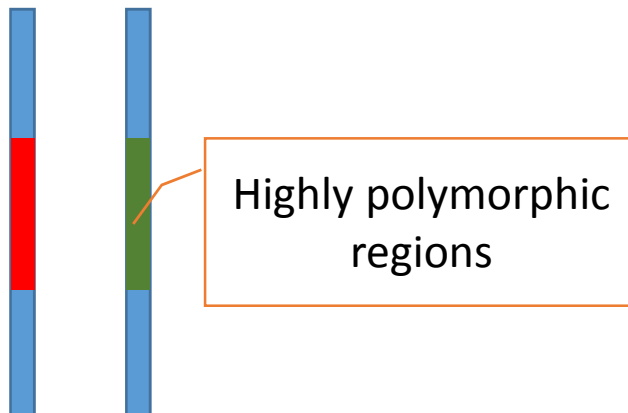
# Heterozygous genomes

## Experimentally:

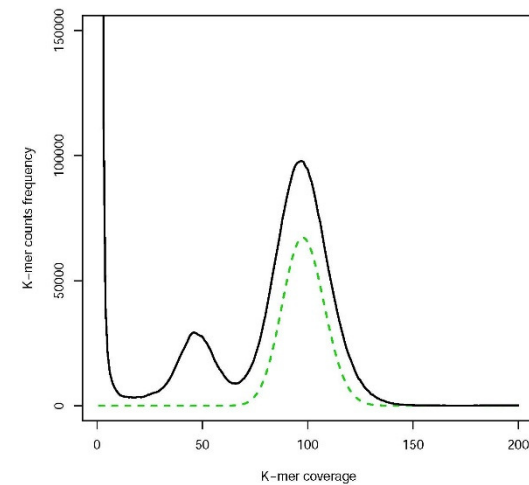
- Create inbred lines or haploid cell culture.
- Assembly of clonal fragments, merging allelic regions.

## Assembly

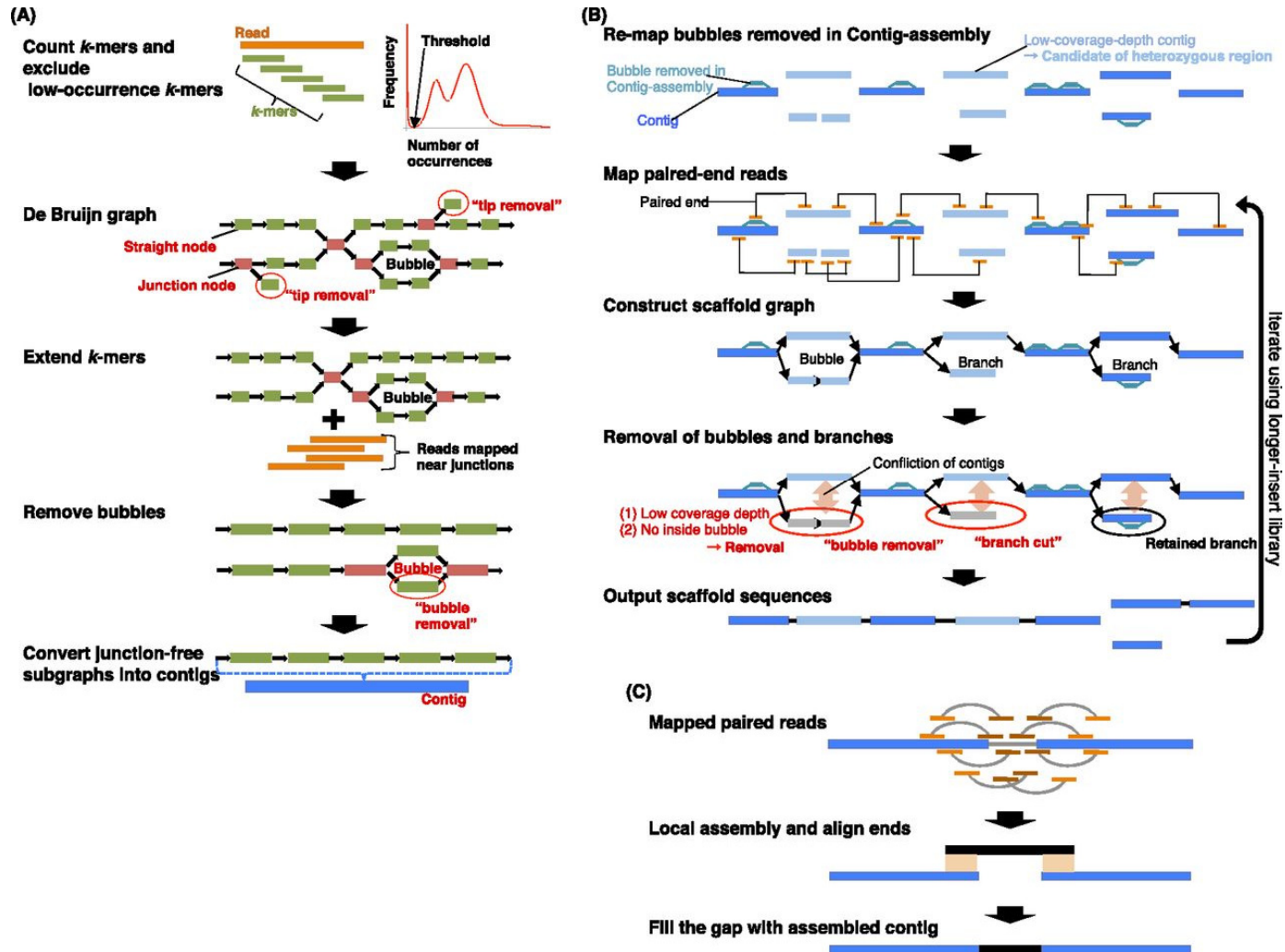
- SNP: merge bubbles
- Highly polymorphic regions



## Kmer coverage distribution



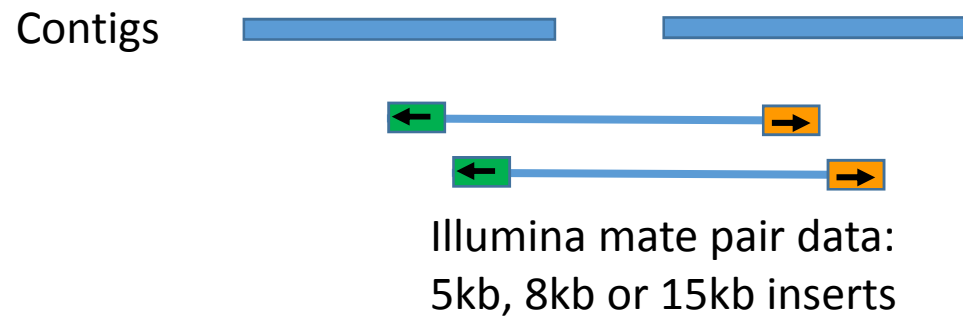
## Examine an assembly software 2. Platanus



Rei Kajitani et al. Genome Res. 2014;24:1384-1395

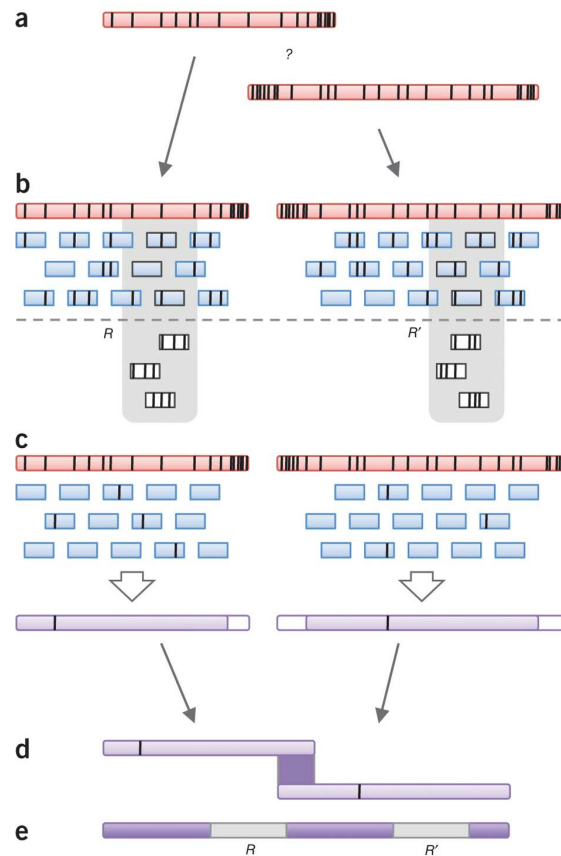


## Scaffolding strategies 1. Mate-pair reads



Software: Soap denovo, et al

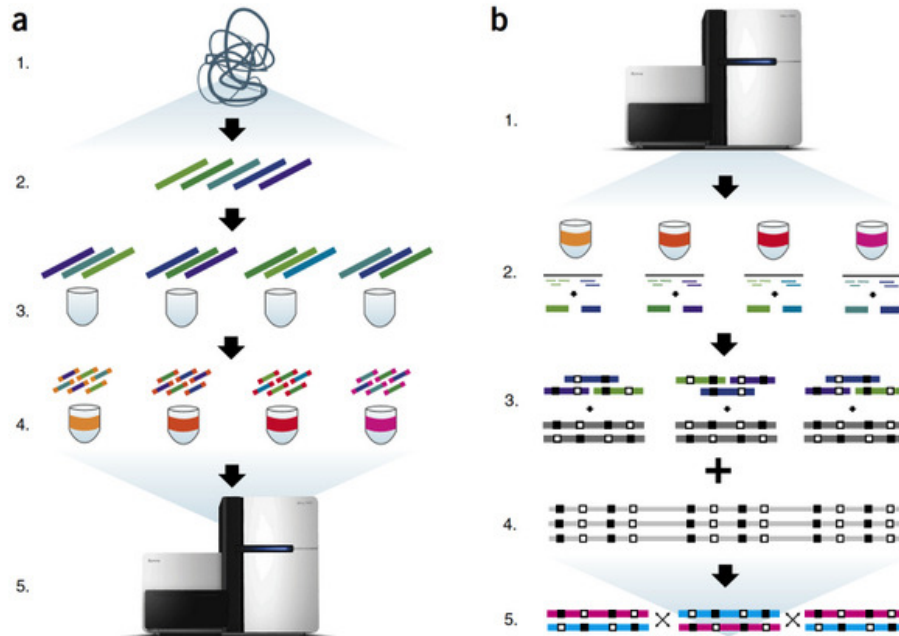
## Scaffolding strategies 2. Long reads



PacBio long reads:  
Long but with high error rate

Current size: 10-15 kb

### Scaffolding strategies 3. Illumina Molecule

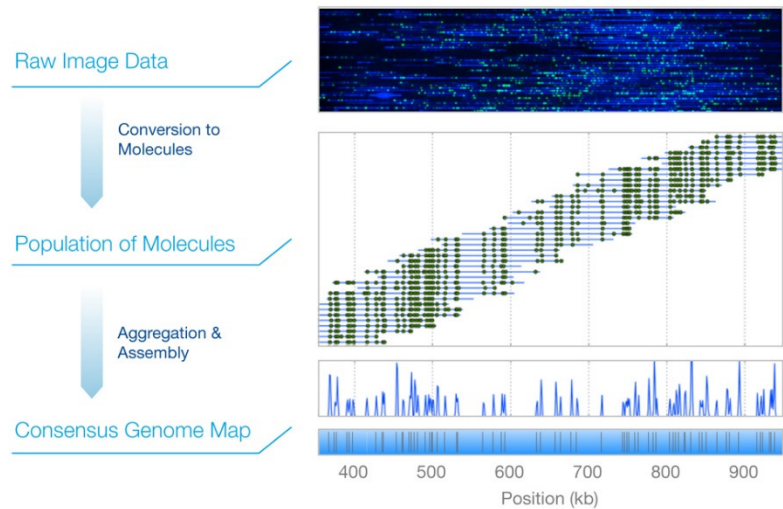


- Generate synthetic long reads for *de novo* assembly and genome finishing applications
- Perform whole human genome phasing to identify co-inherited alleles, haplotype information, and phase *de novo* mutations

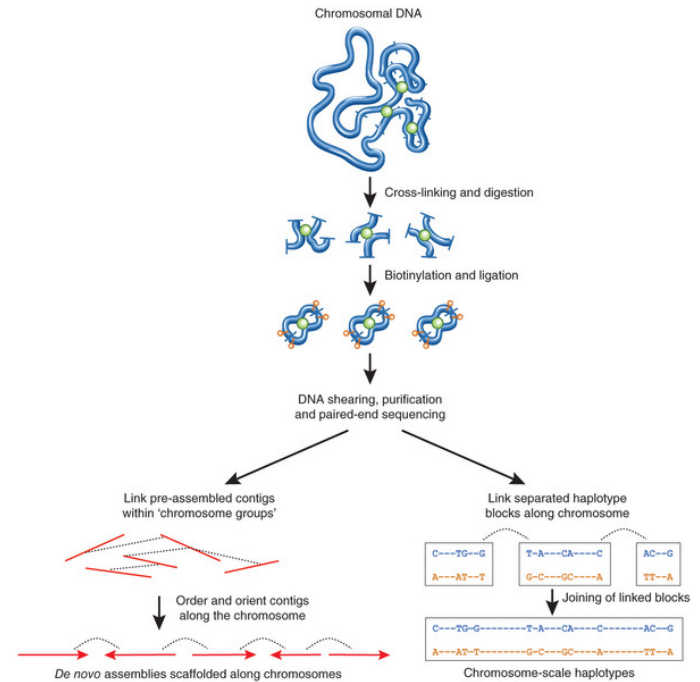


## Scaffolding strategies 4. Physical maps

BioNano:  
 Generating high-quality genome maps  
 by labeling specific 7-mer nickase  
 recognition sites in a genome with a  
 single-color fluorophore



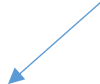
Hi-C:  
 Sequence cross-linked and ligated  
 DNA fragments




## Generate kmer file

```
/programs/jellyfish-2.1.4/bin/jellyfish count -s 1G -m 21 -t 8 -o kmer -C SRR1554178.fastq
```


```
/programs/jellyfish-2.1.4/bin/jellyfish dump -c -t kmer > kmer21.txt
```



CATTATTGGAGTTGATGAAAA	134
CCGCCACCACCACCACCGCCA	52
TAATTAATTTCAAATTTATAA	1
CCCTGTAGAAGCCTGTGTACC	1
AAAAAAAAAATTTGTAGGGGGG	79
GGCATCTAGGTCATGTATTTA	3
AAATGAAAGTACATCAGCTTA	1
AAAATTTGTTCTACCACCGCC	2
CGAGGAGAATTGGGAAAAAAA	103
CTATTTGTGGAATCATAGATC	1
CATCACAAGAACCTATAGAAG	1
CGCCAACCATCCATTATCAAA	1
AAAGTATCGCCAACTTTTAAT	1
CAAAATGAATTTCTTATTTTC	76
GGAAATATGGAGTAGTTACCA	1
CTGATTTCTCTCTATCTCC	1
ATTTATCTCGCAAATTTTAGA	1
CAAAGAGAGAATCACAACAAA	86
...	
...	

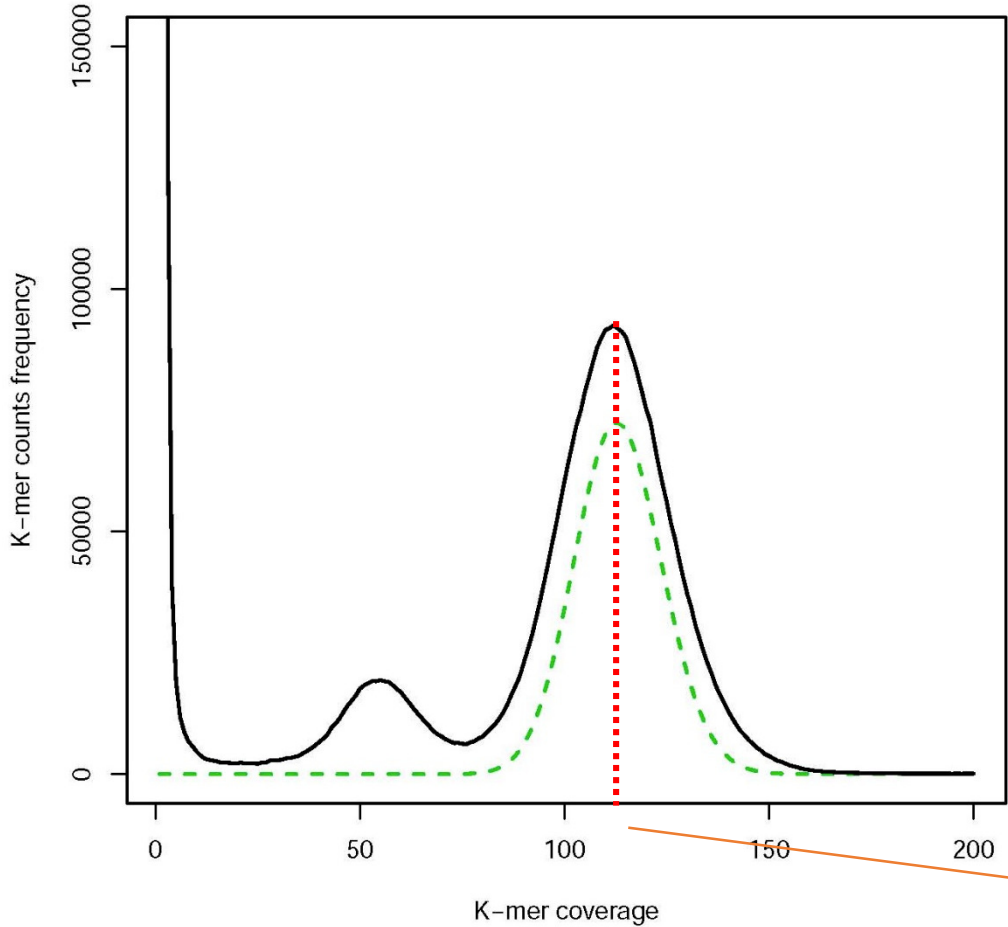


```
awk '{print $2}' kmer21.txt | sort -n | uniq -c \
| awk '{print $2 "\t" $1}' > histogram
```



1	14747539
2	798919
3	131160
4	39150
5	19202
6	11755
7	8734
8	6752
9	5795
10	4679
11	3848
12	3232
13	2978
14	2819
15	2655
16	2373

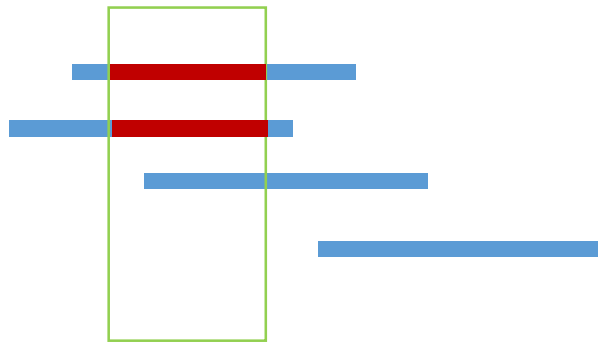
# Estimate genome size based on kmer distribution



100	60272
101	64317
102	68183
103	72164
104	75022
105	78876
106	81907
107	85077
108	88059
109	89946
110	91659
111	91950
112	92535
113	91935
114	91286
115	90194
116	87616
117	85215
118	82041
119	78751
120	75306

Peak kmer depth: 112

# Estimate genome size based on kmer distribution

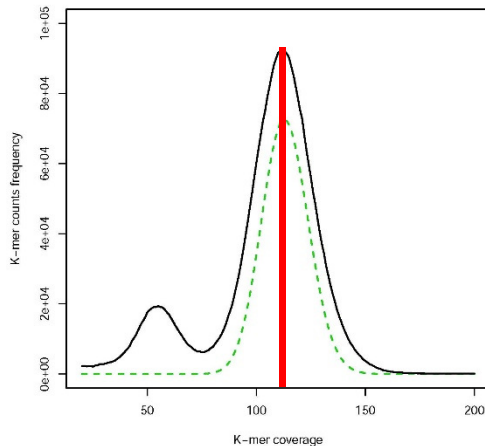


75 mer  
Read depth = 3  
Kmer depth = 2

Step 1: convert read depth to kmer depth

$$N = M * L / (L - K + 1)$$

M: kmer depth = 112  
L: read length = 101 bp  
K: Kmer size = 21 bp  
N: read depth = 140

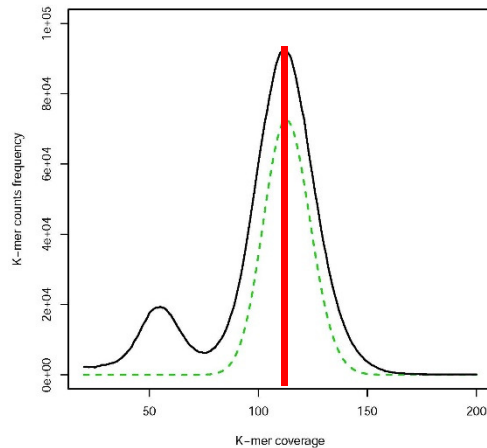


Step 2: genome size is total sequenced basepairs divided by read depth

$$\text{Genome size} = T / N$$

T: total base pairs = 0.505 gb  
N: read depth = 140  
Genome size: 3.6 mb

## Estimate genome size based on kmer distribution

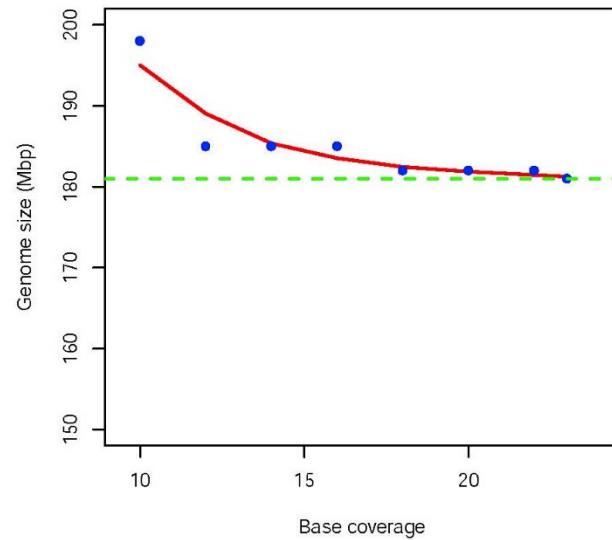
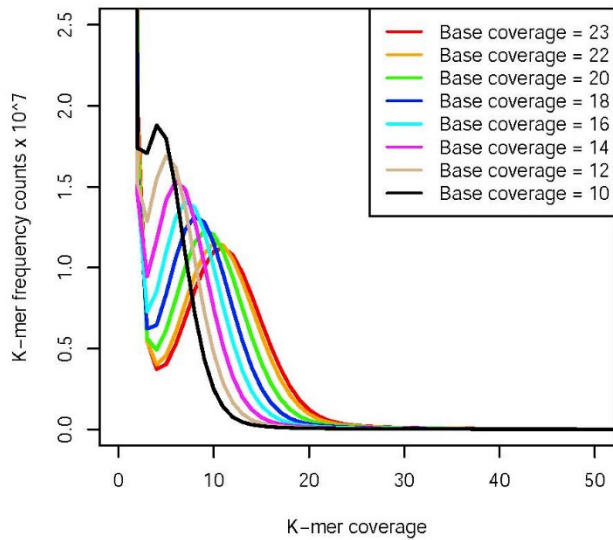
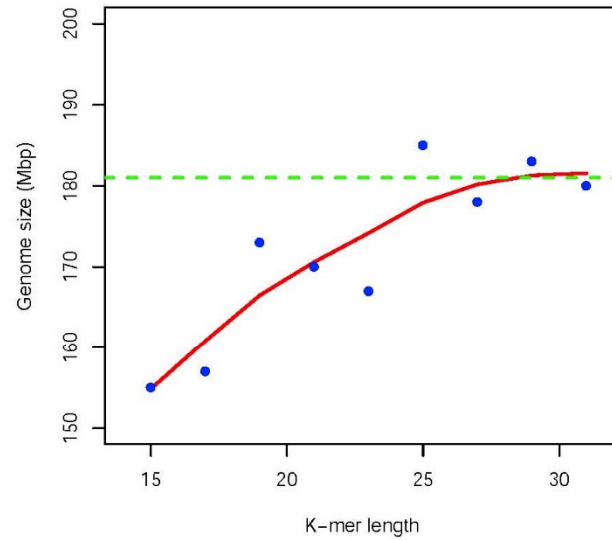
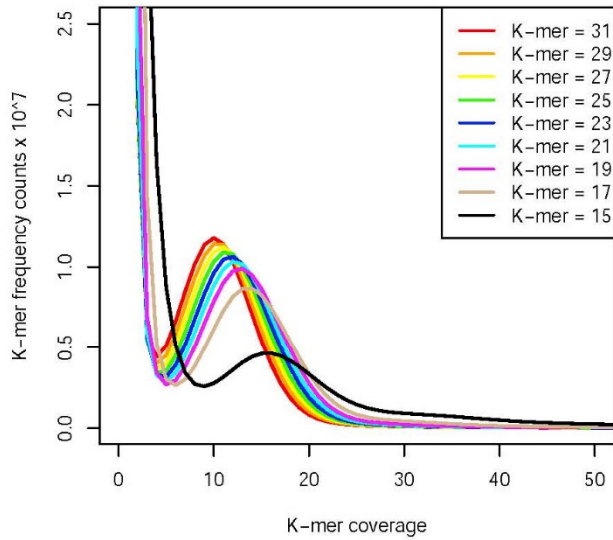


Total kmer bases: 387 mb  
Kmer depth: 113  
Genome size: 3.4 mb

## R code to fit Poisson distribution

```
data <- read.table(file="histogram",header=F)
poisson_expect_K=function(file,start=3,end=200,step=0.1){
diff<-1e20
min<-1e20
pos<-0
total<-sum(as.numeric(file[start:dim(file)[1],1]*file[start:dim(file)[1],2]))
singleCopy_total <- sum(as.numeric(file[10:500,1]*file[10:500,2]))
for (i in seq(start,end,step))
{
singleC <- singleCopy_total/i
a<-sum((dpois(start:end, i)*singleC-file[start:end,2])^2)
if (a < diff){
pos<-i
diff<-a
}
}
print(paste("Total kmer bases: ", total))
print(paste("Kmer depth: ", pos))
print(paste("Genome size: ", total/pos))
pdf("kmerplot.pdf")
plot(1:200,dpois(1:200, pos)*singleC, type = "l", col=3, lwd=2, lty=2,
ylim=c(0,150000), xlab="K-mer coverage",ylab="K-mer counts frequency")
lines(file[1:200,],type="l",lwd=2)
dev.off()
}
poisson_expect_K(data)
```

**Using short kmer could underestimate genome size (<20)**  
**Low kmer depth could overestimate genome size (<20)**



## Exercise: Run kmer analysis tools to estimate genome size

You will be provided with a Fastq file. Using Jellyfish to get the kmer count, then estimate the kmer depth

```
/programs/jellyfish-2.1.4/bin/jellyfish count -s 1G -m 21 -t 4 -o kmer -C  
SRR1554178.fastq  
/programs/jellyfish-2.1.4/bin/jellyfish dump -c -t kmer > kmer21.txt
```