Sequence Assembly Class: Exercises

by
Aureliano Bombarely
ab782@cornell.edu

1. Know your data:

Checking quality and coverage.

2. Usually more is better:

Increasing the read coverage.

3. The key is the library type diversity:

Combine libraries with different insert sizes.

4. Polish the results:

Remove low quality contigs and contaminations

Goal of the Exercise:

Assembly the *Nicotiana benthamiana* chloroplast genome.

Findings:

1- Effects of the read coverage in a genome assembly.

2- Effects of the different insert size library combination

Data Source:

MPMI Vol. 25, No. 12, 2012, pp. 1523–1530. http://dx.doi.org/10.1094/MPMI-06-12-0148-TA.

e-Xtra*

TECHNICAL ADVANCE

A Draft Genome Sequence of *Nicotiana benthamiana* to Enhance Molecular Plant-Microbe Biology Research

Aureliano Bombarely,[1] Hernan G. Rosli,[1] Julia Vrebalov,[1] Peter Moffett,[1,2] Lukas A. Mueller,[1] and Gregory B. Martin[1,3,4]

[1]Boyce Thompson Institute for Plant Research, Ithaca, NY 14853, USA; [2]Département de Biologie, Université de Sherbrooke, Sherbrooke, Quebec J1K 2R1, Canada; [3]Department of Plant Pathology and Plant-Microbe Biology, Cornell University, Ithaca, NY 14853, USA; [4]Genomics and Biotechnology Section, Department of Biological Sciences, Faculty of Science, King Abdulaziz University, P.O. Box 80203 Jeddah 21589, Saudi Arabia

Submitted 7 June 2012. Accepted 28 July 2012.

Data Source:

Table 1. *Nicotiana benthamiana* sequencing, assembly, and gene space evaluation statistics

Sequencing statistics	**Raw data**		**Processed data**	
Library	**Size**	**Coverage**	**Size**	**Coverage**
Paired end 0.4 Kb inserts	181.29 Gb	58 X	160.63 Gb	51 X
Mate pair 2 Kb inserts	46.59 Gb	15 X	39.99 Gb	13 X
Mate pair 5 Kb inserts	40.04 Gb	13 X	28.58 Gb	9 X
Total	267.92 Gb	86 X	229.2 Gb	63 X
Assembly statistics	**Contigs**		**Scaffolds**	
Total assembly size	2.46 Gb		2.63 Gb	
Total assembled sequences	461,463[a]		141,339	
Longest sequence length	208.21 Kb		615.59 Kb	
Average sequence length	5,336 bp		18,610 bp	
N90 index[b]	163,000 sequences		30,261 sequences	
N90 length	3,403 bp		23,201 bp	
N50 index	42,984 sequences		8,897 sequences	
N50 length	16,480 bp		89,778 bp	
Gene space statistics	**Length > 300 bp[c]**		**Coverage > 80%[d]**	
N. benthamiana unigenes	79%		93%	
Solanum lycopersicum gene models	61%		39%	

[a] A total of 485,798 contigs were originally present in the assembly but only contigs >200 bp (461,463) were included in the contig file and in this report. In terms of genome size the 24,335 contigs excluded add approximately 3.72 Mb to the total size.

[b] When ordering all contigs (or scaffolds) by size, the N50 or N90 index indicates the number of the longest sequences (contigs or scaffolds) that contain 50 or 90%, respectively, of the total assembled sequence. The N50 and N90 length indicate the length of the shortest sequence in the set of the largest contigs (or scaffolds) that contain 50 or 90%, respectively, of all the sequences in the assembly.

[c] The percentage of the *N. benthamiana* unigenes (16,024) or tomato gene models (34,739) for which we could identify, in the v0.4.2 draft sequence, a fragment of at least 300 bp.

[d] The percentage of *N. benthamiana* unigenes or tomato gene models that were found, in the v0.4.2 draft sequence, to be represented with a coverage of 80% or more.

Data Source:

Processed reads

Library	Raw data Size	Raw data Coverage
Paired end 0.4 Kb inserts	181.29 Gb	58 X
Mate pair 2 Kb inserts	46.59 Gb	15 X
Mate pair 5 Kb inserts	40.04 Gb	13 X
Total	267.92 Gb	86 X

Quality-Length Filtering

PCR duplications Filtering

Processed data Size	Processed data Coverage
160.63 Gb	51 X
39.99 Gb	13 X
28.58 Gb	9 X
229.2 Gb	63 X

Quality-Length Filtering → Fastq-mcf

minimum quality score 30 (Q30)

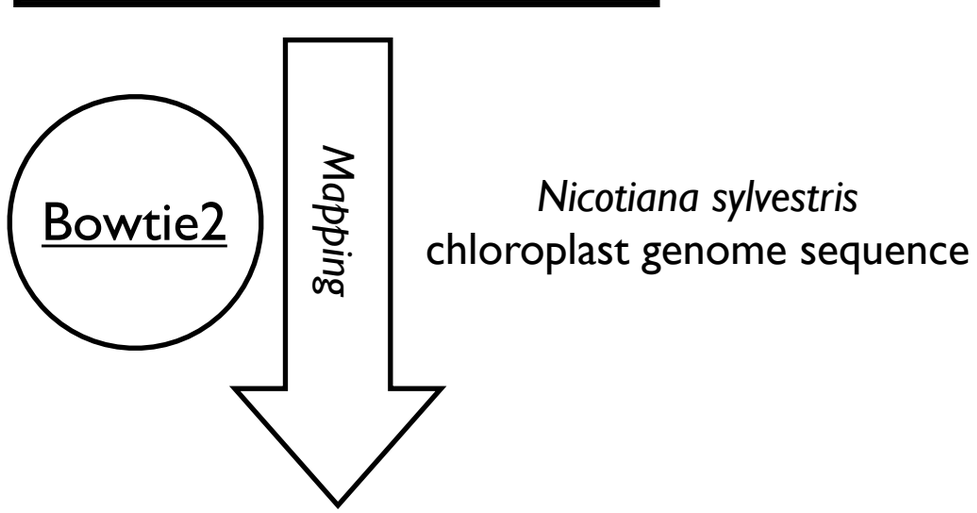
minimum length 50 bp (L50)

PCR duplications Filtering → PrinSeq

Data Source:

Chloroplast reads identification

Processed data	
Size	**Coverage**
160.63 Gb	51 X
39.99 Gb	13 X
28.58 Gb	9 X
229.2 Gb	63 X



Library Name	Type	Insert (bp)
single_*	single reads	NA
pe400_*	pair ends	400
mp2k_*	mate pairs	2000

Data Source:

Chloroplast reads identification

Library Name	Type	Insert (bp)	Set	Subsets	Range
single	single reads	NA	1	12	1K to 22000K
pe400_L1	pair ends	400	1	7	1K to 880K
pe400_L2	pair ends	400	2	1	860K
mp2k_L7	mate pairs	2000	1	6	1K to 870K

Exercise 1: Calculate the estimated coverage per dataset.

Data Needed:

- Each of the dataset files (single, pair ends L1 and mate pairs L7). Data: 1K, 10K, 100K and 500K (200K for mate pairs)
- Size of the chloroplast genome.

1. **Nicotiana** undulata **chloroplast**, complete genome
   155,863 bp circular DNA
   Accession: NC_016068.1 GI: 351653853
   GenBank FASTA Graphics

2. **Nicotiana** sylvestris **chloroplast**, complete genome
   155,941 bp circular DNA
   Accession: NC_007500.1 GI: 78102509
   GenBank FASTA Graphics

3. **Nicotiana** tomentosiformis **chloroplast**, complete genome
   155,745 bp circular DNA
   Accession: NC_007602.1 GI: 81301540
   GenBank FASTA Graphics

4. **Nicotiana** tabacum plastid, complete genome
   155,943 bp circular DNA
   Accession: NC_001879.2 GI: 81238323
   GenBank FASTA Graphics

~155,900 bp

Exercise 1: Calculate the estimated coverage per dataset.

Tools Needed:

- R, bioconductor and package 'ShortRead'.
- Functions:
  - 'readFastq' , to read the fastq file
  - 'sread' , to extract the reads from the object
  - 'width' , to get the length of each read.
  - 'sum' , to sum the lengths

Exercise 1: Calculate the estimated coverage per dataset.

Results Presentation:

- Table with 4 columns (FileName; Reads; TotalBP; EstimatedCoverage) (note: Pair ends and mate pairs library should be presented as a sum of data)

Solution 1: Calculate the estimated coverage per dataset.

1. Open R-Studio, select the Package 'ShortRead'. If this package is not in the list, type in the console:
   - source("http://bioconductor.org/biocLite.R")
   - biocLite("ShortRead") and mark now 'ShortRead' package
2. Read the file:

```
readFastq('Desktop/GenoAssemblyData/single/single_0001K.fq')
```

3. Get the reads:

```
sing1k_rd = sread(sing1k)
```

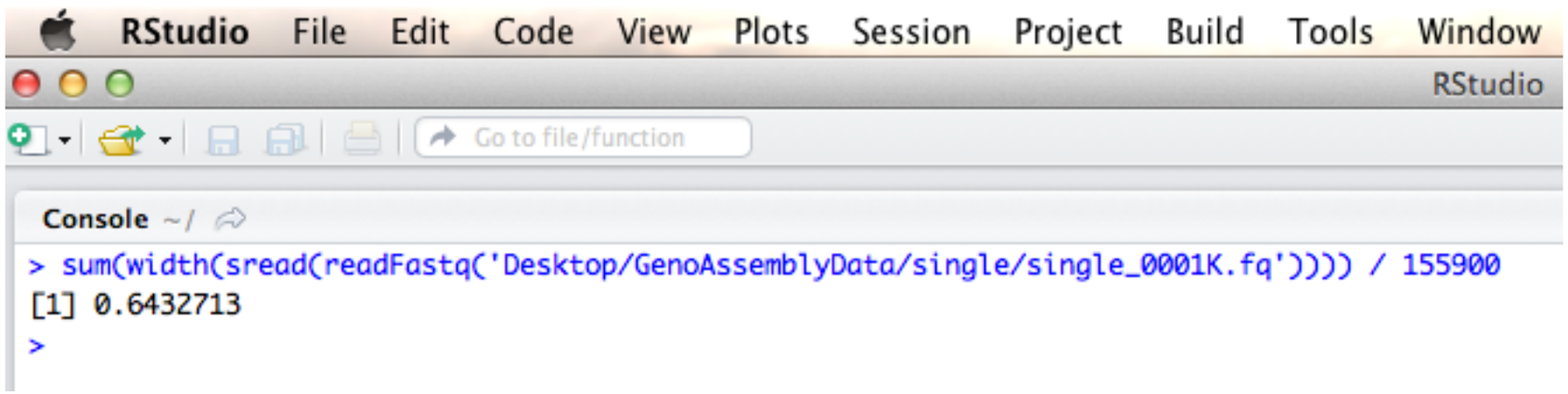
Solution 1: Calculate the estimated coverage per dataset.

4. Get the length for each of them:

```
sing1k_wd = width(sing1k_rd)
```

5. Sum the results for all the reads and divide by 155900

```
sum(sing1k_wd) / 155900
```



Solution 1: Calculate the estimated coverage per dataset.

FileName	Reads	TotalBP	EstimatedCoverage
single_0001K.fq	1000	100286	0.64X
single_0010K.fq	10000	1001962	6.43X
single_0100K.fq	105000	10506611	67.39X
single_0500K.fq	505000	49968467	320.52X
pe400b_L1_001K_p*.fq	1019 pairs	202507	1.30X
pe400b_L1_010K_p*.fq	10429 pairs	2079247	13.34X
pe400b_L1_100K_p*.fq	109022 pairs	21763252	139.60X
pe400b_L1_500K_p*.fq	512531 pairs	98811409	633.81X
mp2kb_L7_001K_p*.fq	1068 pairs	212214	1.36X
mp2kb_L7_010K_p*.fq	10691 pairs	2130010	13.66X
mp2kb_L7_100K_p*.fq	100483 pairs	19955733	128.00X
mp2kb_L7_200K_p*.fq	206678 pairs	40631972	260.62X

Exercise 2: Check quality, sequence length and duplications

Data Needed:

- Each of the dataset files (single, pair ends L1 and mate pairs L7). Data: 10K

Tools Needed:

- FastQC

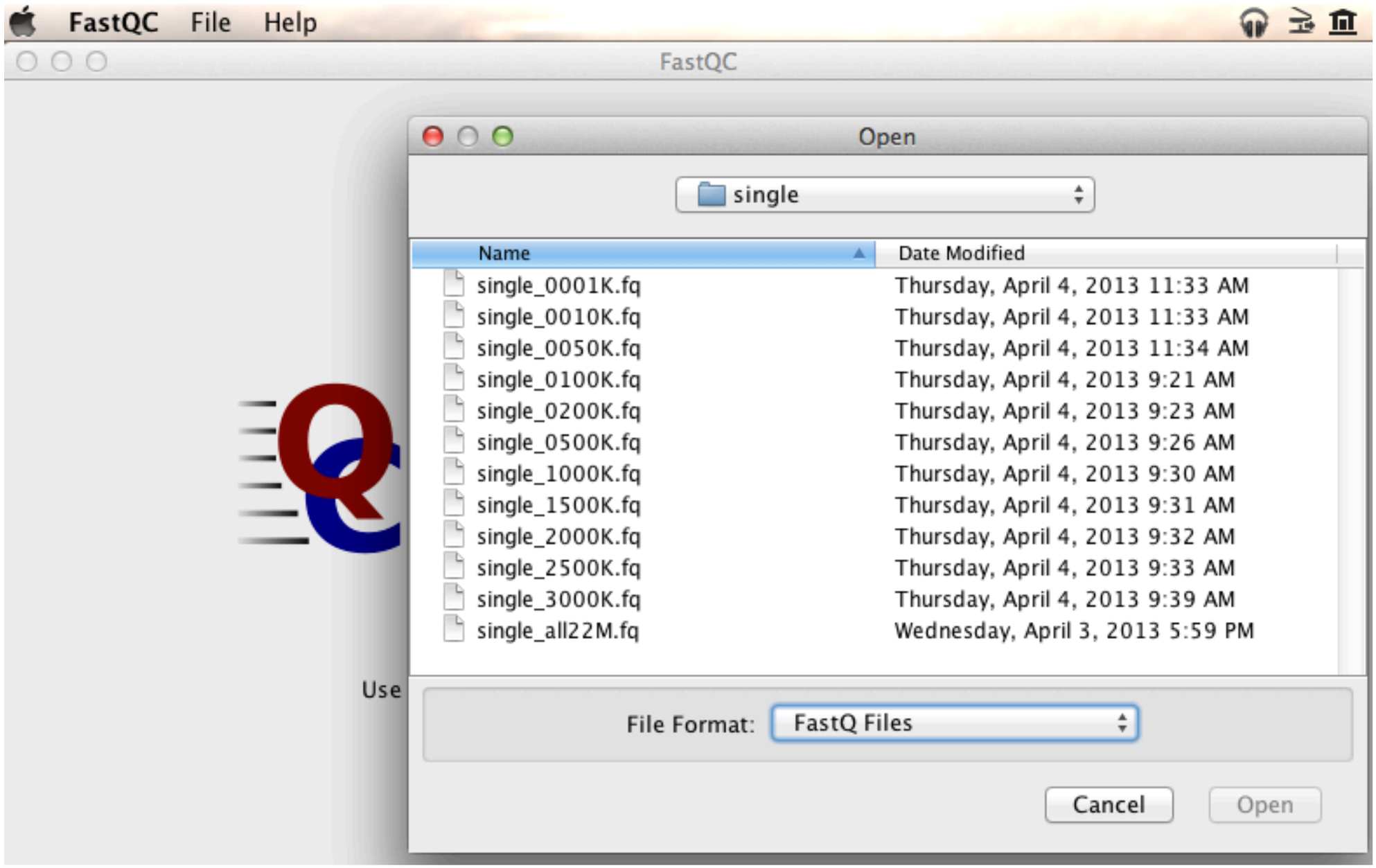
Exercise 2: Check quality, sequence length and duplications.

Results Presentation:

- Answer these questions:
  - Which is the mininum read length ?
  - Which is the minimum quality score ?
  - Do the samples have any overrepresented sequence ?

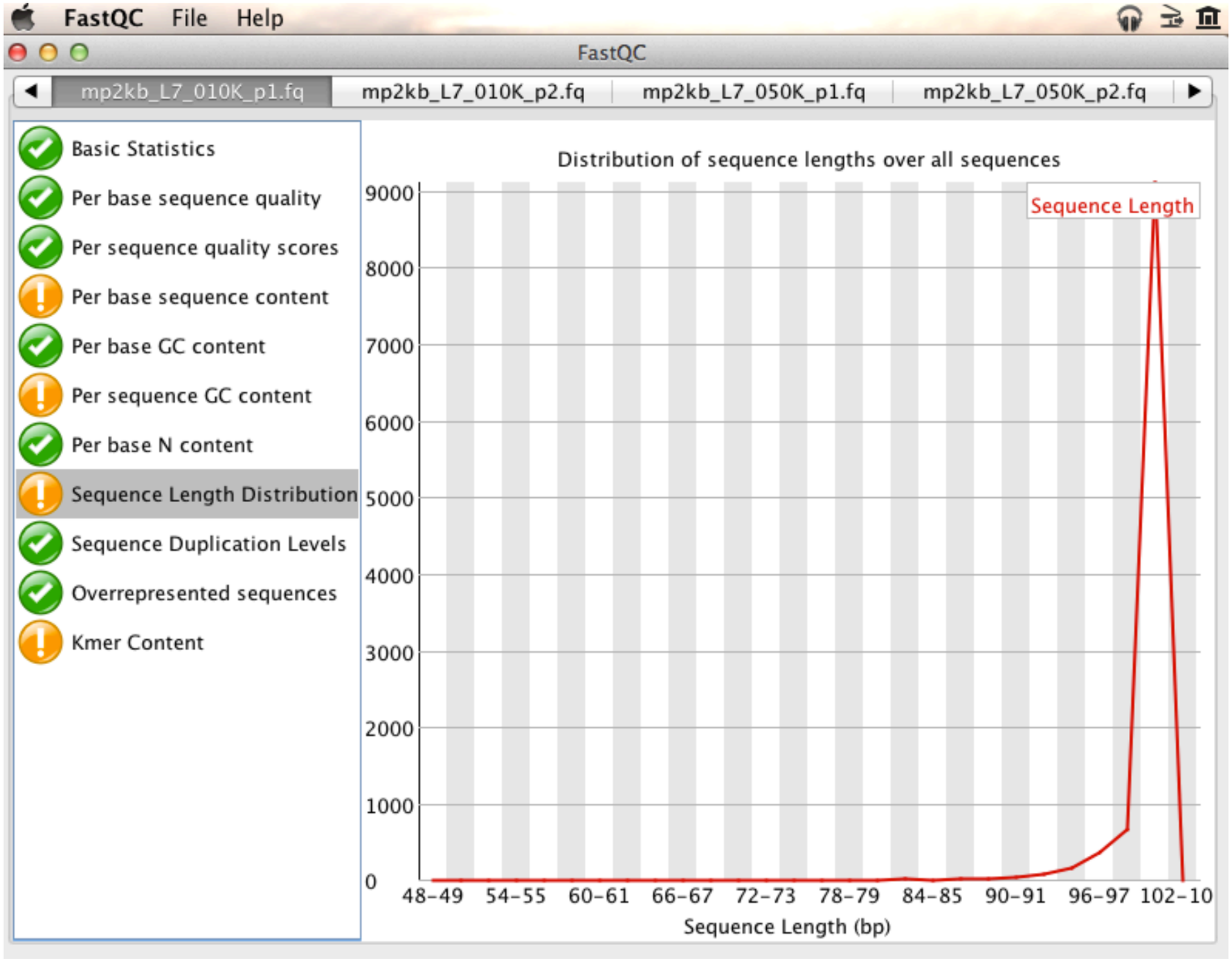
Solution 2: Check quality, sequence length and duplications.

1. Load the reads into the FastQC program



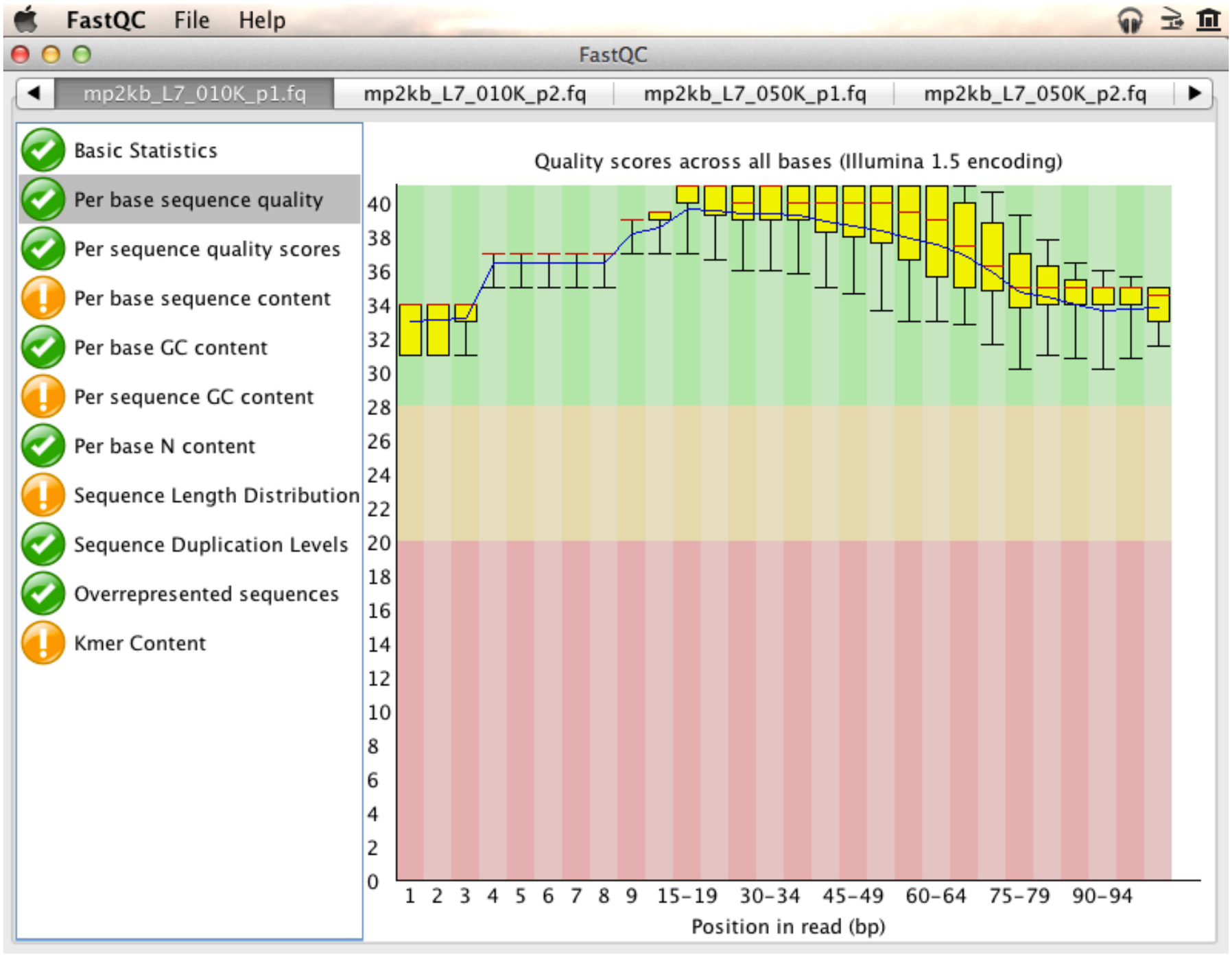
Solution 2: Check quality, sequence length and duplications.

2. Check "Sequence Length Distribution" section



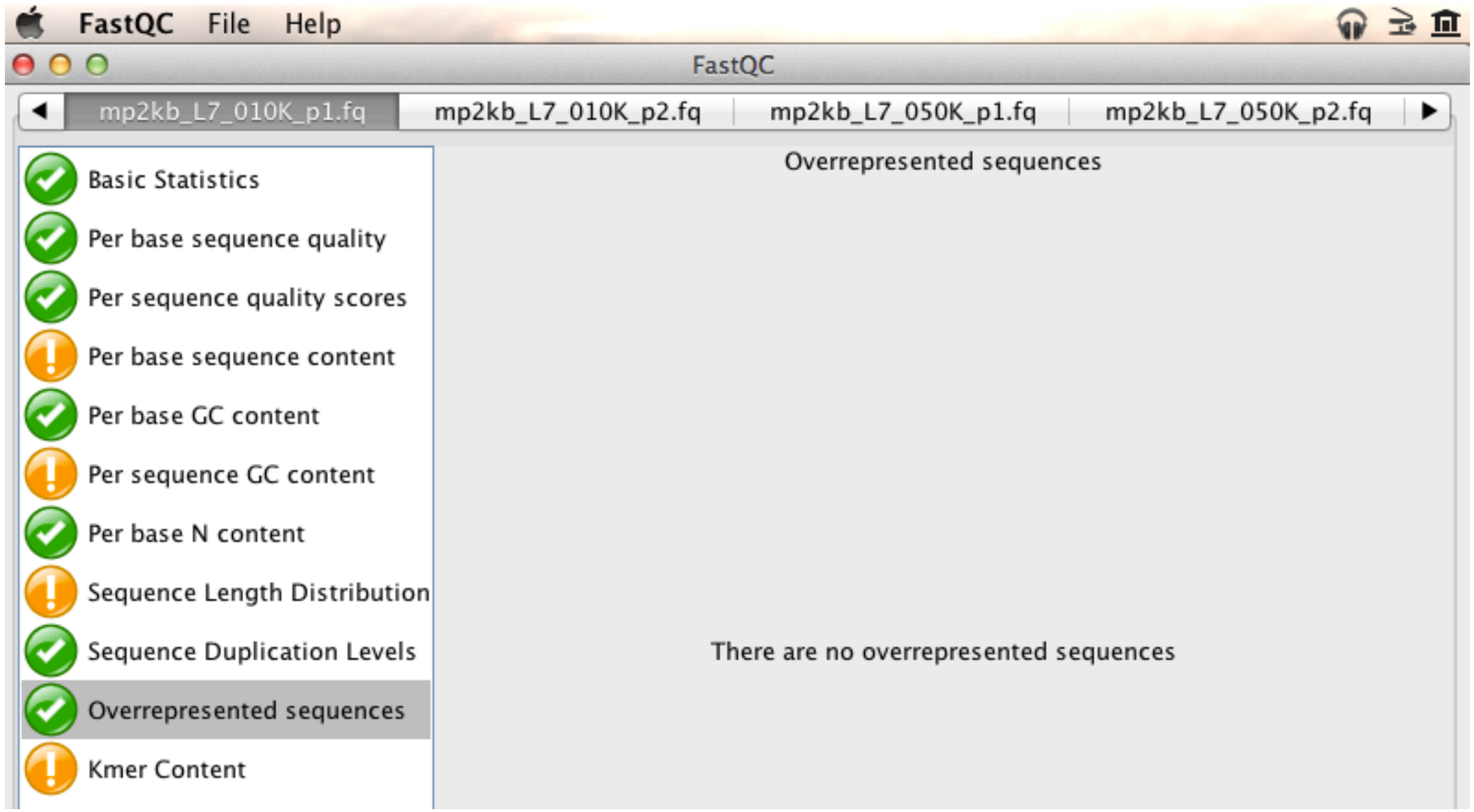
Solution 2: Check quality, sequence length and duplications.

3. Check "Per Base Sequence Quality" section



Solution 2: Check quality, sequence length and duplications.

4. Check "Overrepresented sequences" section



Solution 2: Check quality, sequence length and duplications.

- Answer these questions:
  - Which is the mininum read length ?

    50 bp

  - Which is the minimum quality score ?

    30

  - Do the samples have any overrepresented sequence ?

    No

Exercise 3: Calculate the real coverage using Kmer count

Data Needed:

- Each of the dataset files (single, pair ends L1 and mate pairs L7). Data: 1K, 10K, 100K and 500K (200K for mate pairs).
- Kmer size = 31
- Hash size = 10,000,000

Tools Needed:

- Jellyfish ('count' and 'histo')
- R and R-Studio ('plot', 'lines')

<u>Exercise 3:</u> Calculate the real coverage using Kmer count

Results Presentation:

- 3 Graph with 4 the datasets each (one per library)

Solution 3: Calculate the real coverage using Kmer count.

1. Run 'jellyfish count' with every dataset.

```
jellyfish count -m 31 -s 10000000 -o single1k_kmerc single/single_0001K.fq
jellyfish count -m 31 -s 10000000 -o single10k_kmerc single/single_0010K.fq
jellyfish count -m 31 -s 10000000 -o single100k_kmerc single/single_0100K.fq
jellyfish count -m 31 -s 10000000 -o single500k_kmerc single/single_0500K.fq

jellyfish count -m 31 -s 10000000 -o pe400_L1_1k_kmerc pe400/pe400b_L1_001K_p1.fq pe400/pe400b_L1_001K_p2.fq
jellyfish count -m 31 -s 10000000 -o pe400_L1_10k_kmerc pe400/pe400b_L1_010K_p1.fq pe400/pe400b_L1_010K_p2.fq
jellyfish count -m 31 -s 10000000 -o pe400_L1_100k_kmerc pe400/pe400b_L1_100K_p1.fq pe400/pe400b_L1_100K_p2.fq
jellyfish count -m 31 -s 10000000 -o pe400_L1_500k_kmerc pe400/pe400b_L1_500K_p1.fq pe400/pe400b_L1_500K_p2.fq
jellyfish count -m 31 -s 10000000 -o pe400_L1_500k_kmerc pe400/pe400b_L1_500K_p1.fq pe400/pe400b_L1_500K_p2.fq

jellyfish count -m 31 -s 10000000 -o mp2k_L7_1k_kmerc mp2k/mp2kb_L7_001K_p1.fq mp2k/mp2kb_L7_001K_p2.fq
jellyfish count -m 31 -s 10000000 -o mp2k_L7_10k_kmerc mp2k/mp2kb_L7_010K_p1.fq mp2k/mp2kb_L7_010K_p2.fq
jellyfish count -m 31 -s 10000000 -o mp2k_L7_100k_kmerc mp2k/mp2kb_L7_100K_p1.fq mp2k/mp2kb_L7_100K_p2.fq
jellyfish count -m 31 -s 10000000 -o mp2k_L7_200k_kmerc mp2k/mp2kb_L7_200K_p1.fq mp2k/mp2kb_L7_200K_p2.fq
```

command | kmer | hash size | output | input

Solution 3: Calculate the real coverage using Kmer count.

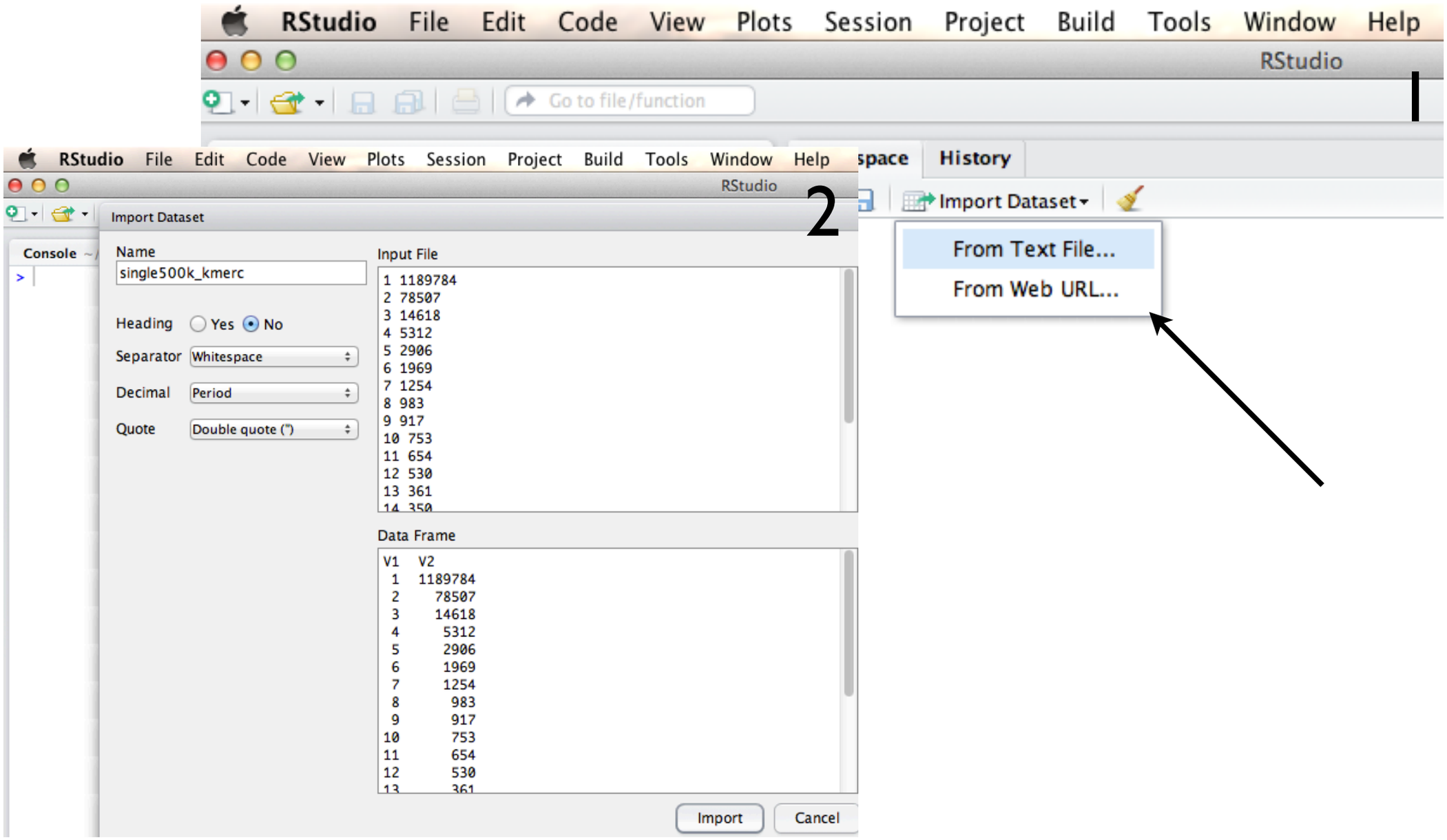
2. Run ‘jellyfish hist’ with every dataset.

```
jellyfish histo single1k_kmerc_0 > single1k_kmerc.hist
jellyfish histo single10k_kmerc_0 > single10k_kmerc.hist
jellyfish histo single100k_kmerc_0 > single100k_kmerc.hist
jellyfish histo single500k_kmerc_0 > single500k_kmerc.hist

jellyfish histo pe400_L1_1k_kmerc_0 > pe400_L1_1k_kmerc.hist
jellyfish histo pe400_L1_10k_kmerc_0 > pe400_L1_10k_kmerc.hist
jellyfish histo pe400_L1_100k_kmerc_0 > pe400_L1_100k_kmerc.hist
jellyfish histo pe400_L1_500k_kmerc_0 > pe400_L1_500k_kmerc.hist

jellyfish histo mp2k_L7_1k_kmerc_0 > mp2k_L7_1k_kmerc.hist
jellyfish histo mp2k_L7_10k_kmerc_0 > mp2k_L7_10k_kmerc.hist
jellyfish histo mp2k_L7_100k_kmerc_0 > mp2k_L7_100k_kmerc.hist
jellyfish histo mp2k_L7_200k_kmerc_0 > mp2k_L7_200k_kmerc.hist
```

command input output

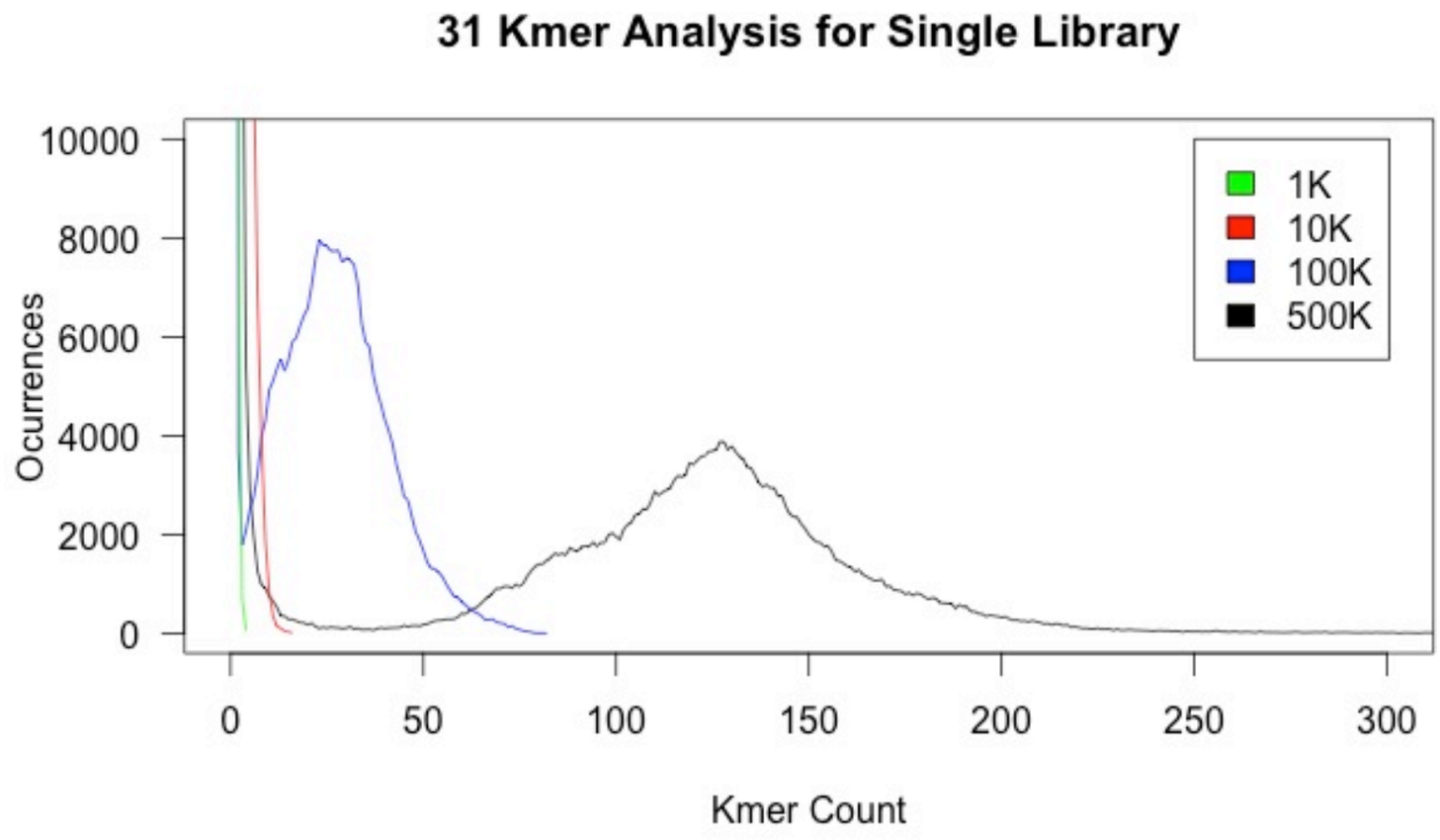
Solution 3: Calculate the real coverage using Kmer count.

3. Load the 'jellyfish hist' output in R-Studio and plot the data using 'plot' and 'lines'.



Solution 3: Calculate the real coverage using Kmer count.

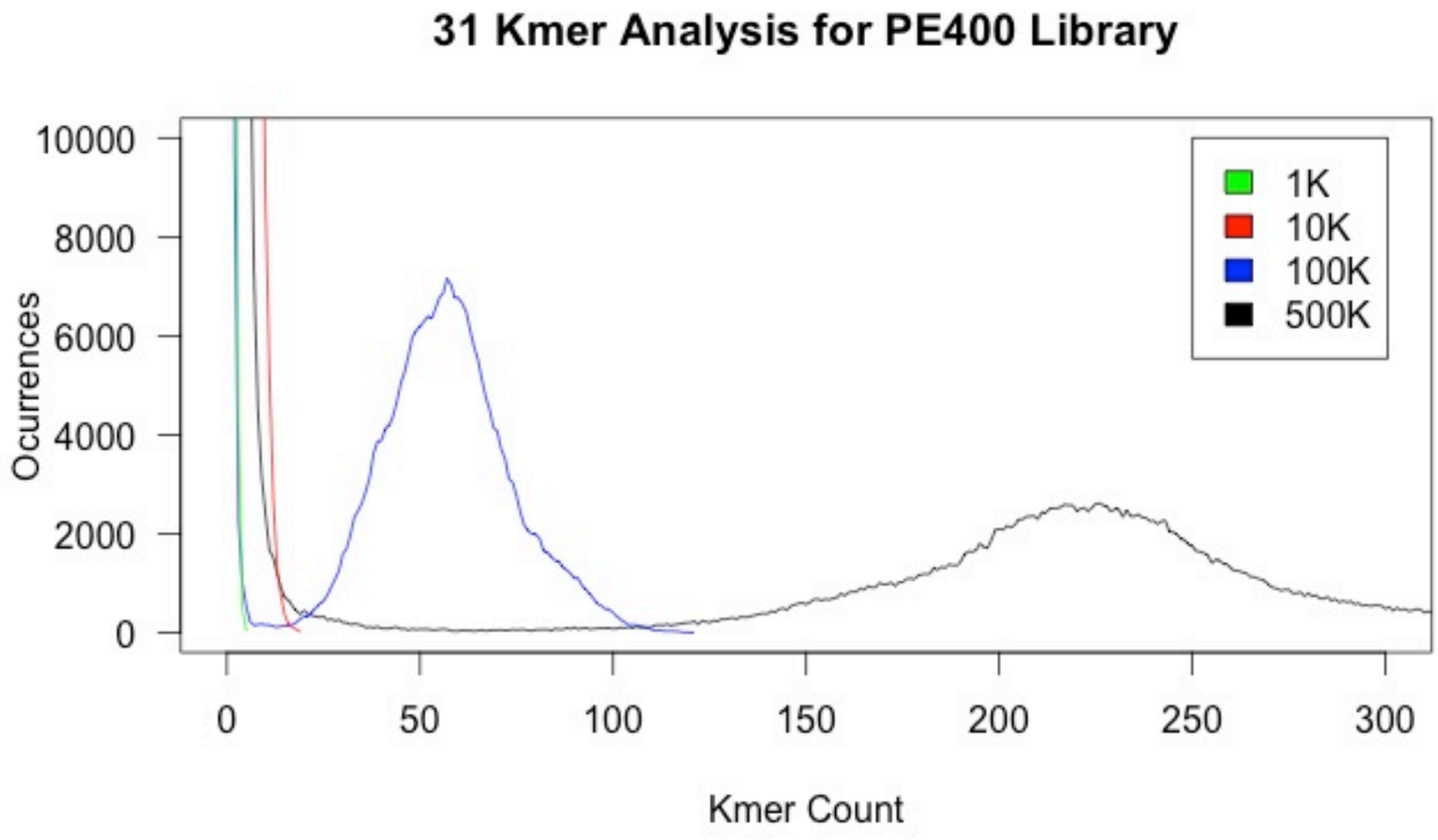
3. Load the 'jellyfish hist' output in R-Studio and plot the data using 'plot' and 'lines'.




```
> plot(single500k_kmerc, type="l", ylim=c(0,10000), las=1, main="31 Kmer Analysis for Single Library", xlab="Kmer Count", ylab="Ocurrences", xlim=c(0,300))
> lines(single100k_kmerc, col="blue")
> lines(single10k_kmerc, col="red")
> lines(single1k_kmerc, col="green")
> legend(250, 10000, c("1K", "10K", "100K", "500K"), fill=c("green", "red", "blue", "black"))
```

Solution 3: Calculate the real coverage using Kmer count.

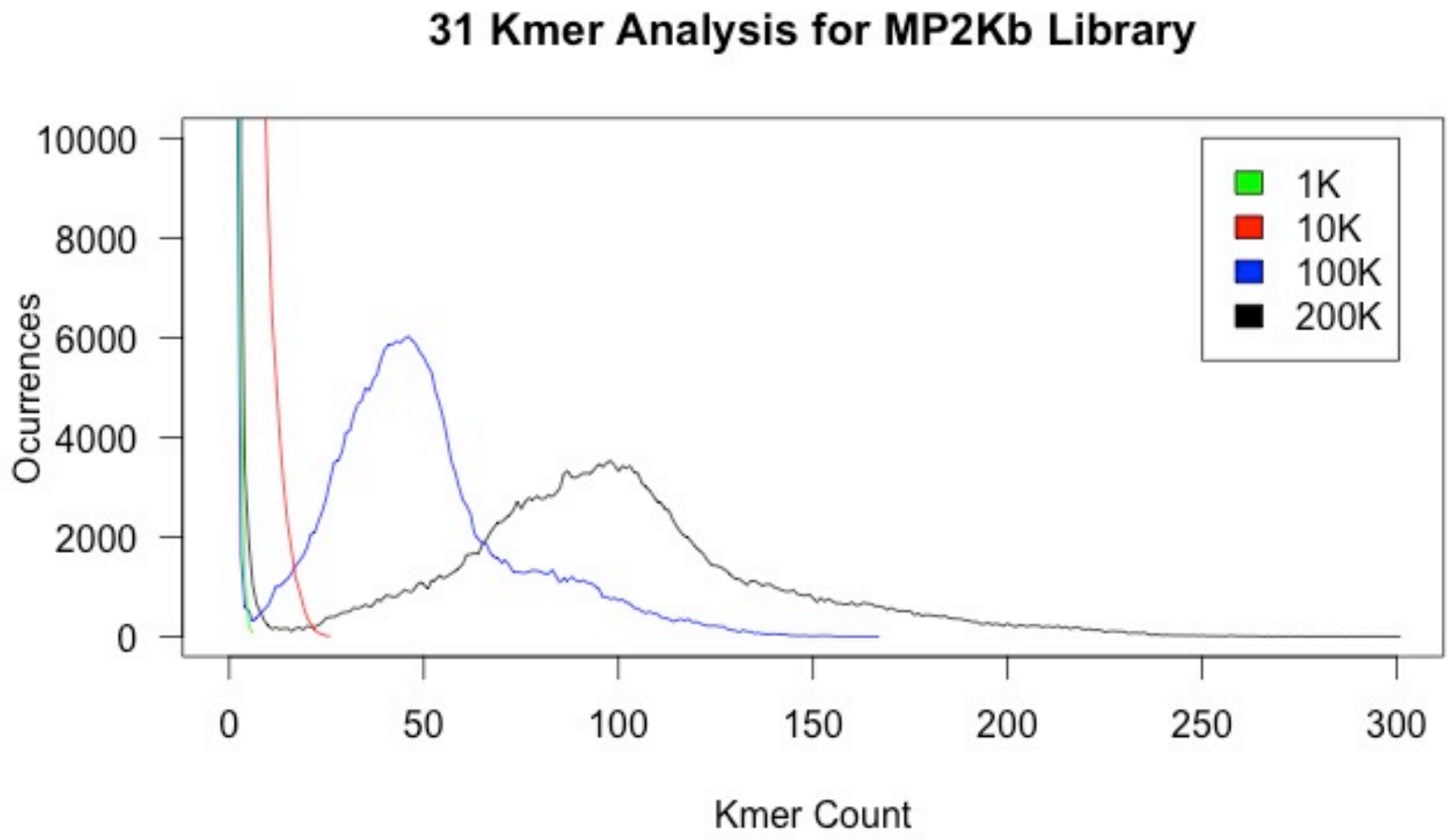
3. Load the 'jellyfish hist' output in R-Studio and plot the data using 'plot' and 'lines'.




```
> plot(pe400_L1_500k_kmerc, type="l", ylim=c(0,10000), las=1, main="31 Kmer Analysis for PE400 Library",
xlab="Kmer Count", ylab="Ocurrences", xlim=c(0,300))
> lines(pe400_L1_100k_kmerc, col="blue")
> lines(pe400_L1_10k_kmerc, col="red")
> lines(pe400_L1_1k_kmerc, col="green")
> legend(250, 10000, c("1K", "10K", "100K", "500K"), fill=c("green", "red", "blue", "black"))
```

Solution 3: Calculate the real coverage using Kmer count.

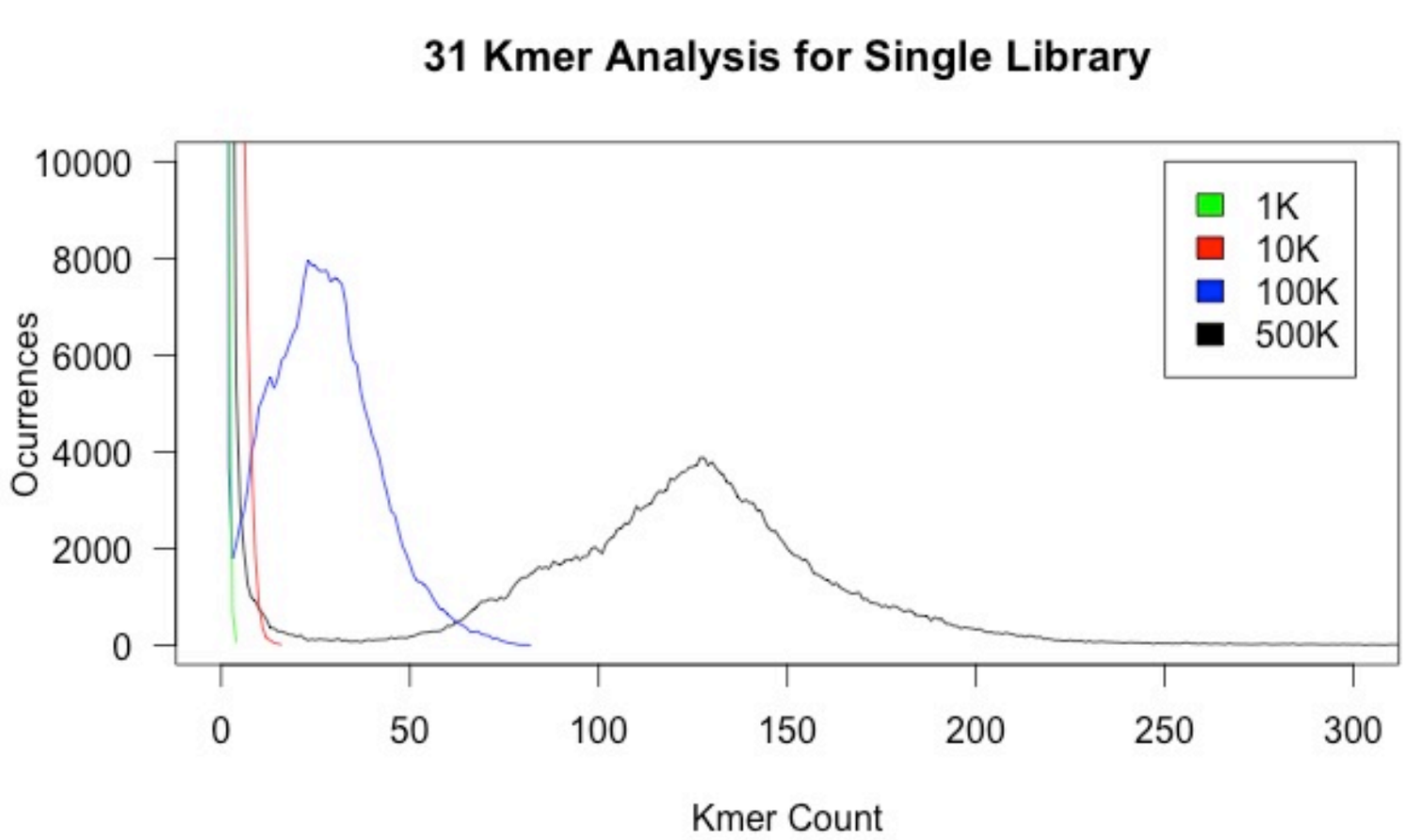
3. Load the 'jellyfish hist' output in R-Studio and plot the data using 'plot' and 'lines'.




```
> plot(mp2k_L7_200k_kmerc, type="l", ylim=c(0,10000), las=1, main="31 Kmer Analysis for MP2Kb Library",
xlab="Kmer Count", ylab="Ocurrences", xlim=c(0,300))
> lines(mp2k_L7_100k_kmerc, col="blue")
> lines(mp2k_L7_10k_kmerc, col="red")
> lines(mp2k_L7_1k_kmerc, col="green")
> legend(250, 10000, c("1K", "10K", "100K", "200K"), fill=c("green", "red", "blue", "black"))
```

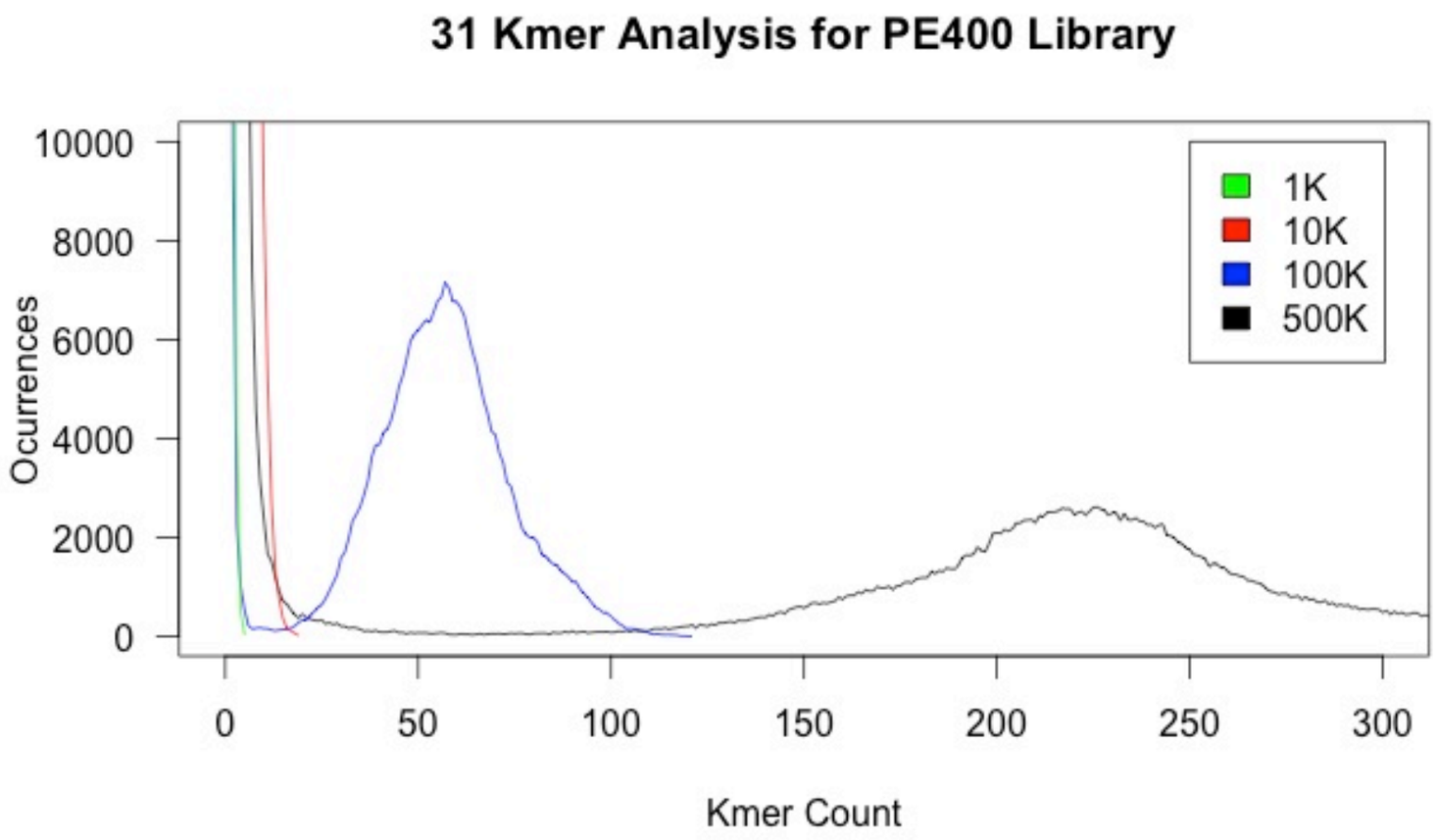
Solution 3: Calculate the real coverage using Kmer count.

FileName	EstimatedCoverage
single_0001K.fq	0.64X
single_0010K.fq	6.43X
single_0100K.fq	67.39X
single_0500K.fq	320.52X



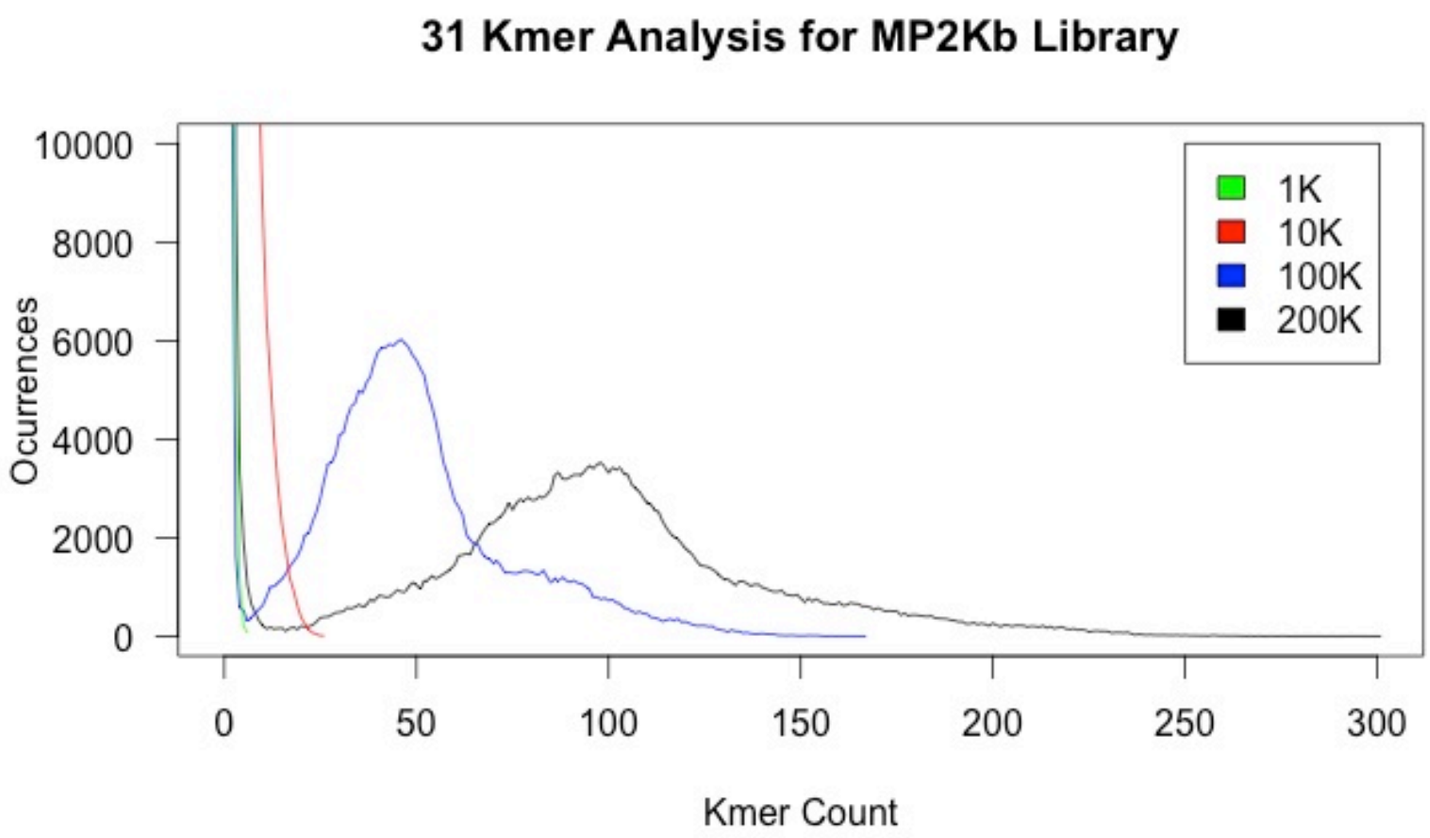
Solution 3: Calculate the real coverage using Kmer count.

FileName	EstimatedCoverage
pe400b_L1_001K_p*.fq	1.30X
pe400b_L1_010K_p*.fq	13.34X
pe400b_L1_100K_p*.fq	139.60X
pe400b_L1_500K_p*.fq	633.81X



Solution 3: Calculate the real coverage using Kmer count.

FileName	EstimatedCoverage
mp2kb_L7_001K_p*.fq	1.36X
mp2kb_L7_010K_p*.fq	13.66X
mp2kb_L7_100K_p*.fq	128.00X
mp2kb_L7_200K_p*.fq	260.62X



Exercise 4: Compare assemblies with different coverage

Data Needed:

- The single library dataset 1K, 10K, 100K and 500K
- Kmer size = 31
- Format the output (unitig/contig) for 100 character lines

<u>Exercise 4:</u> Compare assemblies with different coverage

Tools Needed:

- ABySS (<u>http://www.bcgsc.ca/downloads/abyss/doc</u>)
- Fold
- R-Studio, Bioconductor and 'Biostring' package
  - 'read.DNAstringset'
  - 'width'
  - 'sort', 'length' 'sum', 'cumsum' and 'mean'

Exercise 4: Compare assemblies with different coverage

Results Presentation:

- Table with 5 columns: Assembly, TotalSequences, TotalSize BiggestSeq, Mean and N50.

Solution 4: Compare assemblies with different coverage.

1. Run ‘abyss-pe’ with every dataset.

```
/Applications/ABySS.app/bin/abyss-pe name=single1k_k31     k=31 se='single/single_0001K.fq'
```

command	output	kmer	input
/Applications/ABySS.app/bin/abyss-pe	name=single10k_k31	k=31	se='single/single_0010K.fq'
/Applications/ABySS.app/bin/abyss-pe	name=single100k_k31	k=31	se='single/single_0100K.fq'
/Applications/ABySS.app/bin/abyss-pe	name=single500k_k31	k=31	se='single/single_0500K.fq'

Solution 4: Compare assemblies with different coverage.

2. Format the unitig/contig file for a maximum line length of 100 characters.

```
fold -w 100 single1k_k31-unitigs.fa > single1k_k31-unitigs.form.fa

fold -w 100 single10k_k31-unitigs.fa > single10k_k31-unitigs.form.fa
fold -w 100 single100k_k31-unitigs.fa > single100k_k31-unitigs.form.fa
fold -w 100 single500k_k31-unitigs.fa > single500k_k31-unitigs.form.fa
```

command input output

Solution 4: Compare assemblies with different coverage.

3. Open R-Studio, select the Package 'Biostrings'. If this package is not in the list, type in the console:

   - source("http://bioconductor.org/biocLite.R")
   - biocLite("Biostrings") and mark now 'Biostrings' package

4. Read the files:

```
single1k_k31ctg = read.DNAStringSet('single100k_k31-unitigs.form.fa')
```

5. Get the size of each sequence with decreasing size order:

```
single1k_k31ctg_sizes = sort(width(single1k_k31ctg), decreasing=TRUE)
```

Solution 4: Compare assemblies with different coverage.

6. Get the number of sequences

```
length(single1k_k31ctg_sizes)
```

7. Get the total assembly size

```
sum(single1k_k31ctg_sizes)
```

8. Get the mean

```
mean(single1k_k31ctg_sizes)
```

9. Get the N50

```
single1k_k31ctg_sizes[cumsum(single1k_k31ctg_sizes) >=
sum(single1k_k31ctg_sizes)/2][1]
```

Solution 4: Compare assemblies with different coverage.

Assembly	TotalSequences	TotalSize	BiggestSeq	Mean	N50
Single K1	139	14686	307	106	101
Single K10	108	115845	9537	1073	2343
Single K100	13	130658	39228	10051	31802
Single K500	119	134833	23099	1133	10214

Exercise 5: Compare assemblies with different kmers

Data Needed:

- The single library dataset 100K
- Kmer sizes = 23, 31, 39, 47, 55, 63
- Format the output (unitig/contig) for 100 character lines

<u>Exercise 5:</u> Compare assemblies with different kmers

Tools Needed:

- ABySS
- Fold
- R-Studio, Bioconductor and 'Biostring' package
  - 'read.DNAstringset'
  - 'width'
  - 'sort', 'length, 'sum', 'cumsum' and 'mean'

Exercise 5: Compare assemblies with different kmers

Results Presentation:

- Table with 5 columns: Assembly, TotalSequences, TotalSize BiggestSeq, Mean and N50.

Solution 5: Compare assemblies with different kmers.

Kmer	TotalSequences	TotalSize	BiggestSeq	Mean	N50
23	35	130993	26663	3743	9661
31	13	130658	39228	10051	31802
39	15	130629	40960	8709	30311
47	13	130673	20844	10052	16723
55	14	130754	28700	9340	17900
63	18	130978	37860	7277	17902

Exercise 6: Compare assemblies with different libraries

Data Needed:

- The single library dataset 100K
- Pair ends library 100K
- Mate pair library 100K
- The single library dataset 100K + pair ends library 100K + mate pair library 100K
- Kmer sizes = 31
- Format the output (unitig/contig) for 100 character lines

<u>Exercise 6:</u> Compare assemblies with different libraries

Tools Needed:

- ABySS
- Fold
- R-Studio, Bioconductor and 'Biostring' package
  - 'read.DNAstringset'
  - 'width'
  - 'sort', 'length, 'sum', 'cumsum' and 'mean'

Exercise 6: Compare assemblies with different libraries

Results Presentation:

- Table with 5 columns: Assembly, TotalSequences, TotalSize BiggestSeq, Mean and N50. For pair ends and mate pairs will be used to rows, one for contigs and other for scaffolds.

Solution 6: Compare assemblies with different libraries.

library	TotalSequences	TotalSize	BiggestSeq	Mean	N50
Single	13	130658	39228	10051	31802
Pair Ends, contigs	2	205410	137160	103205	137160
Pair Ends, scaffolds	2	205410	137160	103205	137160
Mate Pairs, contigs	5	206276	109928	41255	109928
Mate Pairs, scaffolds	4	206326	111576	51581	111576
All, contigs	2	206570	137160	103285	137160
All, scaffolds	2	206570	137160	103285	137160

Exercise 7: Interpretation of the results

Data Needed:

- Best assembly produced by the exercise 8.

Tools Needed:

- Bl2seq (http://blast.ncbi.nlm.nih.gov/)

Exercise 7: Interpretation of the results

Results Presentation:

Answer the following questions:

- Which is the best assembly ? Why ?

- Order by importance to optimize a sequence assembly the following parameters: “Coverage”, “Kmer”, “Library Type”. Explain why.

- The best assembly is producing two scaffolds and the total assembly size is bigger than the estimated genome size (~156 Kb). Is this wrong ? Why ?

Solution 7: Interpretation of the results

- Which is the best assembly ? Why ?

  Probably the assembly using single ends, pair ends and mate pairs libraries, with a Kmer = 31 and no more than 100K reads per library, but it is close from the pair end assembly.

  It is the best assembly because it has only two sequences, probably the result closest to the expected result (just one sequence for the whole chloroplast sequence).

Solution 7: Interpretation of the results

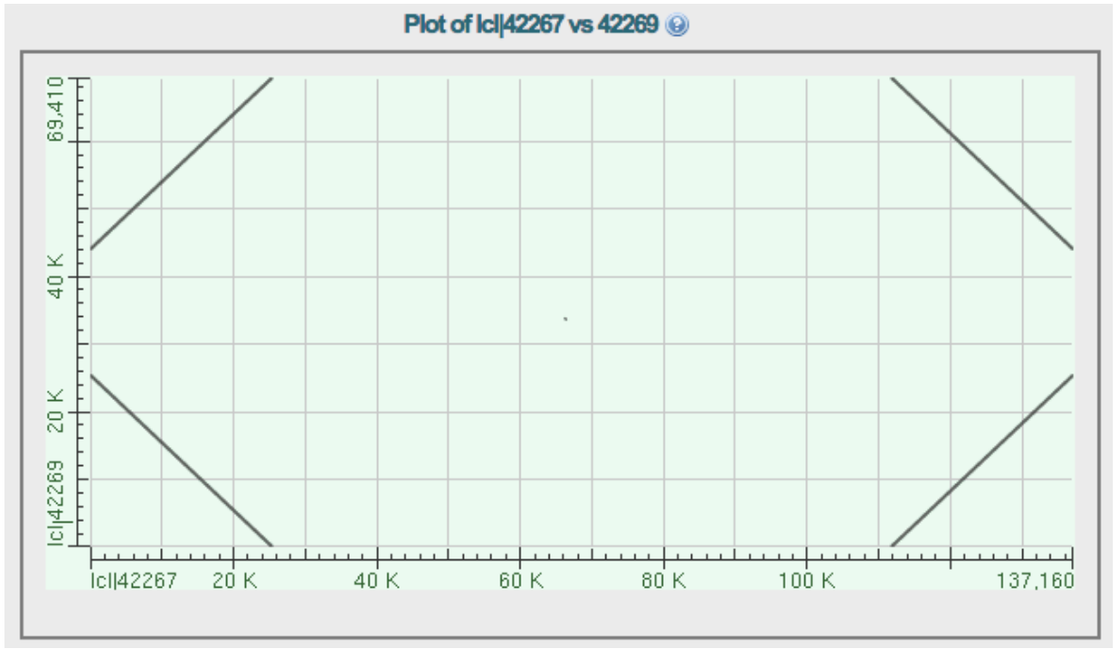
- Order by importance to optimize a sequence assembly the following parameters: "Coverage", "Kmer", "Library Type". Explain why.

Probably "Library Type", "Coverage" and finally "Kmer". It is important to have a combination of pair ends and mate pairs to be able to have scaffolds (single reads don't have pair information). Combination of short (pair ends) and long (mate pairs) pairs lets join read information at different distances.

Solution 7: Interpretation of the results

- The best assembly is producing two scaffolds and the total assembly size is bigger than the estimated genome size (~156 Kb). Is this wrong ? Why ?

No really. Both scaffolds have two overlapping regions in its extremes corresponding to a circular structure. Doing a 2 sequence blast and representing the results as a 'dot matrix view' it is possible to visualize the circular structure.



Solution 7: Interpretation of the results

For other hand, subtracting the overlapping regions from the total assembly size 206,570 - (25,421+25,421) = 155,728 bp