DNA Sequencing

Table of Contents

Nanopore Sequencing

Input Nanopore Sequencing Data (FASTQ File)

```
fastqinfo('lambda_nanopore.fastq')
```

```
ans = struct with fields:
           Filename: 'lambda_nanopore.fastq'
           FilePath: 'C:\Users\kabil\OneDrive - Drexel University\Academic\3 - Pre-Junior\1 - Fall Quarter\BMES 375\
        FileModDate: '14-Oct-2021 22:56:12'
           FileSize: 10926131
    NumberOfEntries: 947
```

```
reads = fastqread('lambda_nanopore.fastq')
```

reads = 1×947 struct

Fields	Header	Sequence	Quality
1	'fdf910a5-7...	'GTTGTGT...	'/+++$$#...
2	'a41fe7de-f...	'AGTATGC...	'#$&#'&*...
3	'baabbce4-9...	'CACTAGG...	'$.%'(.8...
4	'42be897d-8...	'CGTGTAC...	'&$#%'&-...
5	'9a42ec01-c...	'ACTCTAT...	'+&(.752...
6	'3cd8595d-4...	'GACTCGT...	'&/%*.13...
7	'56468e96-9...	'ATTGCTA...	'/**+-11...
8	'5c1f6b93-0...	'GTTACTA...	'+)),*-G...

Fields	Header	Sequence	Quality
9	'863bdf10-0...	'ATGCTGT...	'/##&'+0...
10	'6fa01687-d...	'CAGTATA...	"#%)%#&...
11	'ccc2bb13-2...	'AGCTGAG...	'+%,3698...
12	'4282edf2-b...	'ACTATTG...	'/06556%...
13	'50cd3e3d-3...	'CTGTTTA...	'$"%%&%#...
14	'd10220d5-d...	'CATTCAC...	'$#"$%#%...
15	'1a10a492-6...	'CGGTACT...	'*%)*'6=...
16	'7dc543fb-0...	'GTTGTAC...	')%(()02...
17	'475608a9-b...	'GTATGCT...	'$'$$$,/...
18	'e06a5c44-9...	'ATTGCTG...	'.$%$.76...
19	'57d2b26a-2...	'CGGTGTA...	"$%%%$,...
20	'8bc2f25a-1...	'CACTTTA...	'+/$$$/8...
21	'5bfdc711-8...	'CACTTTA...	'#*356BF...
22	'99bf0895-c...	'GTTGTAC...	",/(-*1...
23	'f30fcb5b-0...	'CAGTATG...	'%")&%#...
24	'2b76be60-3...	'GTTGTAC...	'$'+65;=...
25	'4826d568-6...	'GTTGTAT...	'&%40/3)...
26	'bd6ec2d0-5...	'GTTGTAC...	'%--&$$$...
27	'65decd3f-c...	'CGGTATG...	'+%+-,))...
28	'c0d54cc6-8...	'CGGTATG...	'+$)-*)&...
29	'dc5564fe-8...	'GGTGTAC...	'$&$#+-2...
30	'6b3efec0-3...	'ACTGAGC...	'%')(**6...
31	'5322227c-1...	'CACTCGT...	"+9;;2*...
32	'23705b9e-5...	'CAGTGTA...	'*#('$%,...
33	'7a635d85-3...	'ACTGTAA...	',$$#),+...
34	'6b90d19d-5...	'GGTATAC...	'$###$#%...
35	'9770040d-0...	'CGATGTG...	'&$#%%&%...
36	'75ba1b7c-5...	'GTTGTAC...	'$'($$%&...
37	'918011e9-5...	'GGTAGCT...	'&'(&$#(...
38	'8ae02f94-c...	'GATGTAC...	'%$+'"4...
39	'85fb32ed-c...	'ACTATTG...	'-'0:8F5...
40	'eb98718e-9...	'CGATGTA...	'+(###&....
41	'ac9ae36b-1...	'CGCTTCG...	'##%'++/...

Fields	Header	Sequence	Quality
42	'4a42bec8-6...	'CATTGTA...	'$#%')(&...
43	'71d2f951-b...	'CACTCTA...	',+),06E...
44	'b299d04f-7...	'ATCATTG...	'.%%$%%%...
45	'4e3c1746-a...	'CGCATGG...	"$%-9;@...
46	'53ab9c74-4...	'GTTGTAC...	'+45,,-,...
47	'54c537c1-4...	'CAGTGAC...	'.#($%"%...
48	'51169d15-c...	'ACTGTGG...	'-)+-7?A...
49	'0bc40f6d-9...	'CATTGTG...	'%##+'%#...
50	'035160b1-b...	'CGTATTG...	"%*5+++...
51	'b37d0fc3-8...	'CAATGCT...	'.+&%%00...
52	'705b9c80-4...	'ACTGCTT...	'0344))/...
53	'e494ff3e-a...	'CGGTATC...	'("&,##...
54	'ad586169-2...	'CAATGTA...	',%$$$%*...
55	'4005d8ae-3...	'CAACACT...	'&%%&')+...
56	'2f2db06d-f...	'CGGTAGC...	'%&&%$$#...
57	'9a5f92e1-d...	'CGGTACT...	'*&&('--...
58	'3ed9b28a-b...	'CGGTATT...	'0$(%($%...
59	'c27603ba-2...	'GTTACTG...	'&"2.3'...
60	'ffbedb22-5...	'CGGTATT...	'.&))($%...
61	'e2e40267-f...	'GATGTAC...	'%$&$%%)...
62	'09bf2a04-f...	'GGTATGC...	'&&(&$#%...
63	'05d89c21-c...	'GTTGTAC...	'.67)+)4...
64	'121e0af0-9...	'GTTGTGC...	'$$&%%(5...
65	'9fff5139-9...	'ATTGCTG...	'3$&&/7<...
66	'e12c5c25-c...	'CAAACTT...	'$&$#"$%...
67	'6b7d8b39-4...	'GTTGTAC...	',#%#$$$...
68	'4489ecab-c...	'GATGTGC...	'$#$$(&%...
69	'1b42397a-e...	'CATTGTA...	"'()'.,...
70	'07dc7045-2...	'CATTGTA...	'-##$%*1...
71	'b2dabdf9-f...	'CTTTAGT...	'$$'(('%...
72	'1549c5c2-b...	'CAAACGT...	'/-*%"#$...
73	'6ac83da8-3...	'ACCATTC...	'/#$%$/#...
74	'124bfc77-a...	'AGTATGC...	'#$&#$$'...

Fields	Header	Sequence	Quality
75	'bb462264-8...	'CGGTAGC...	"$%$$/0...
76	'a3ac48d3-7...	'GCGCTTT...	'##$%&"...
77	'5470915f-e...	'GTTCTTC...	'*5/+(-2...
78	'aca5810f-b...	'GTTGTAA...	'.-.00.$...
79	'0025f8a3-9...	'CAGTATA...	'#&'$%'%...
80	'824f8dae-e...	'CATTGTA...	'$#$&%&&...
81	'c0f5ef32-e...	'GTTGTAC...	'%02)(),...
82	'2db70e97-7...	'GTTGTAC...	'-23/366...
83	'fcfa06b7-6...	'CTATGCT...	'#$%%$12...
84	'06d0d008-f...	'CGGTGTA...	')%$%#%%...
85	'fe768141-e...	'ATCTATT...	'.')0:9:...
86	'30604dfc-a...	'GTTGTAC...	'),-$));...
87	'd7801ea9-c...	'GGCATGC...	'%#$#$"(...
88	'f00524da-b...	'ACTCTAT...	'&.28C/+...
89	'08a4e8dc-7...	'CGGTGTA...	"#%#$&0...
90	'ede50cae-6...	'ACTGGGC...	'.-201.'...
91	'7d595274-1...	'GTTGTGC...	'-89))&$...
92	'42cb287b-6...	'CACTATT...	'$.'3812...
93	'07f28d79-a...	'GTTGCTT...	'%#$#'/0...
94	'ac18972e-0...	'CGGTATT...	'&$$%$"$...
95	'a7dd4ba8-f...	'GTTACTT...	'-2(2667...
96	'a497e1cc-b...	'AAATTTG...	'$%#%))&...
97	'ba4d675f-9...	'GCTACTA...	'%#%+-:>...
98	'b44f057a-4...	'CGTTGTA...	'&%%'%$%...
99	'62cde812-b...	'GTTGTAC...	'&#%$')+...
100	'38179797-1...	'CGTATGC...	'%%&&*$%...

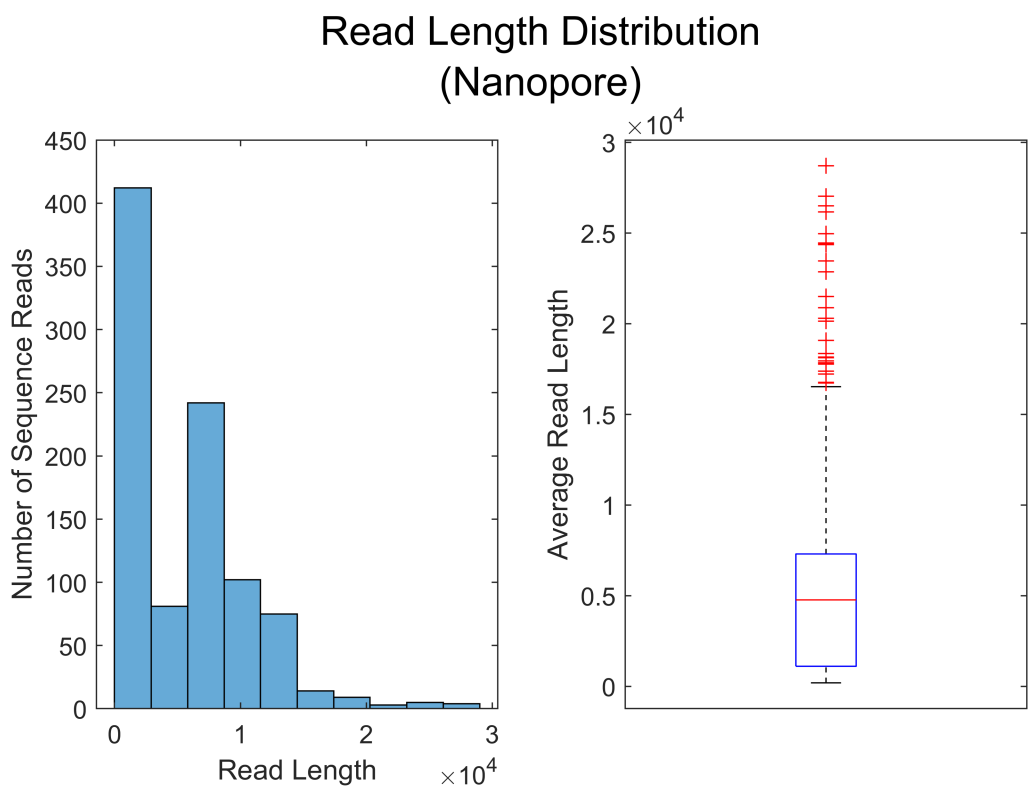
⋮

Read Length Distribution of All Sequencing Reads

```
seqs_nano = {reads.Sequence}; % convert to cell array
readLen_nano = cellfun(@length, seqs_nano); % calculate read length

% Visualize results
figure('Position', [0 0 600 400]), sgtitle({'Read Length Distribution', '(Nanopore)'});
subplot(1,2,1), histogram(readLen_nano, 10);
xlabel('Read Length'), ylabel('Number of Sequence Reads');
```

```
subplot(1,2,2), boxplot(readLen_nano), ylabel('Average Read Length'), xticks([]);
```



Caption: The figure shows the distribution of sequence read lengths from the nanopore sequencing run. On the left is a histogram showing the frequency distribution of read lengths. On the right is a boxplot showing the median, upper and lower quartiles, and outliers of average sequence read length.

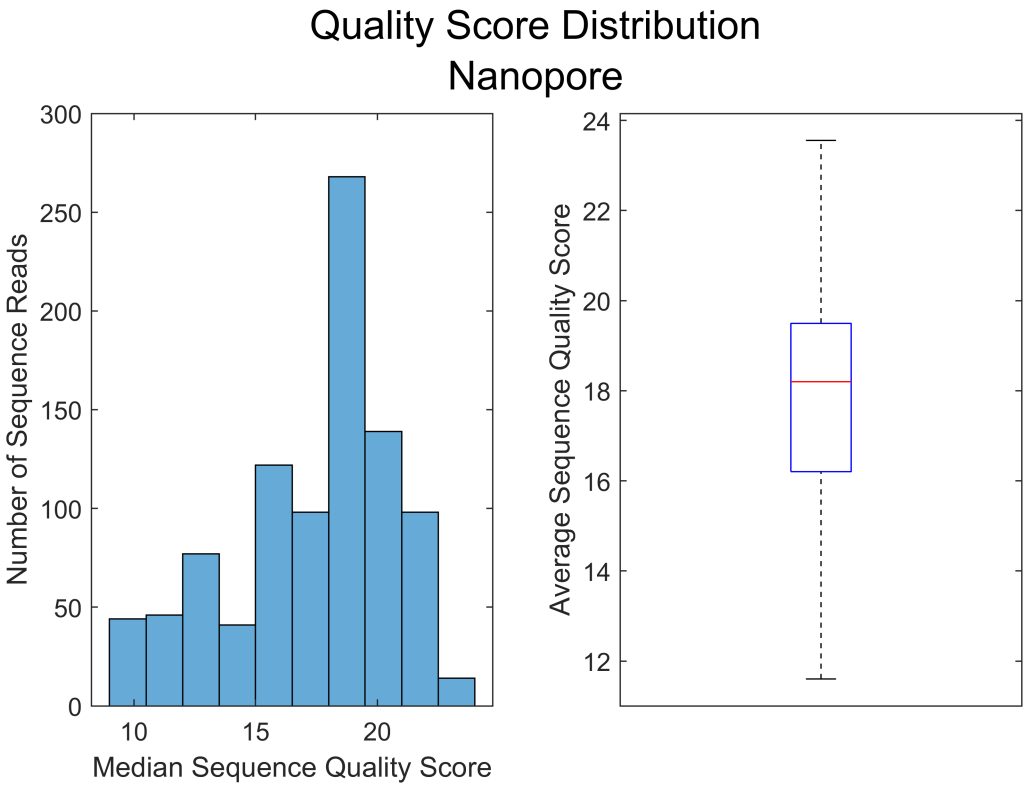
Conclusion: The average read length for the nanopore tends to be around 0.5×10^4. Although, there are some outliers with as high as 2.5×10^4 reads.

Quality Score Distribution per Sequencing Read

```
seqQ_nano = {reads.Quality}; % ASCII format
% Convert ascii to digits
seqQS_nano = cellfun(@(x) double(x) - 33, seqQ_nano, 'UniformOutput', false);

% Average, Median & Standard Deviation
avgQS_nano = cellfun(@mean, seqQS_nano);
medQS_nano = cellfun(@median, seqQS_nano);
stdQS_nano = cellfun(@std, seqQS_nano);

% Plot Distribution of Median and Average Quality
figure('Position', [0 0 600 400]), sgtitle({'Quality Score Distribution', 'Nanopore'});
subplot(1,2,1), histogram(medQS_nano, 10);
xlabel('Median Sequence Quality Score'), ylabel('Number of Sequence Reads');
subplot(1,2,2), boxplot(avgQS_nano), ylabel('Average Sequence Quality Score'), xticks([]);
```



Caption: The figure shows the distribution of sequence quality scores from the nanopore sequencing run. On the left is a histogram showing the frequency distribution of quality scores. On the right is a boxplot showing the median, upper and lower quartiles, and outliers of average sequence quality score.

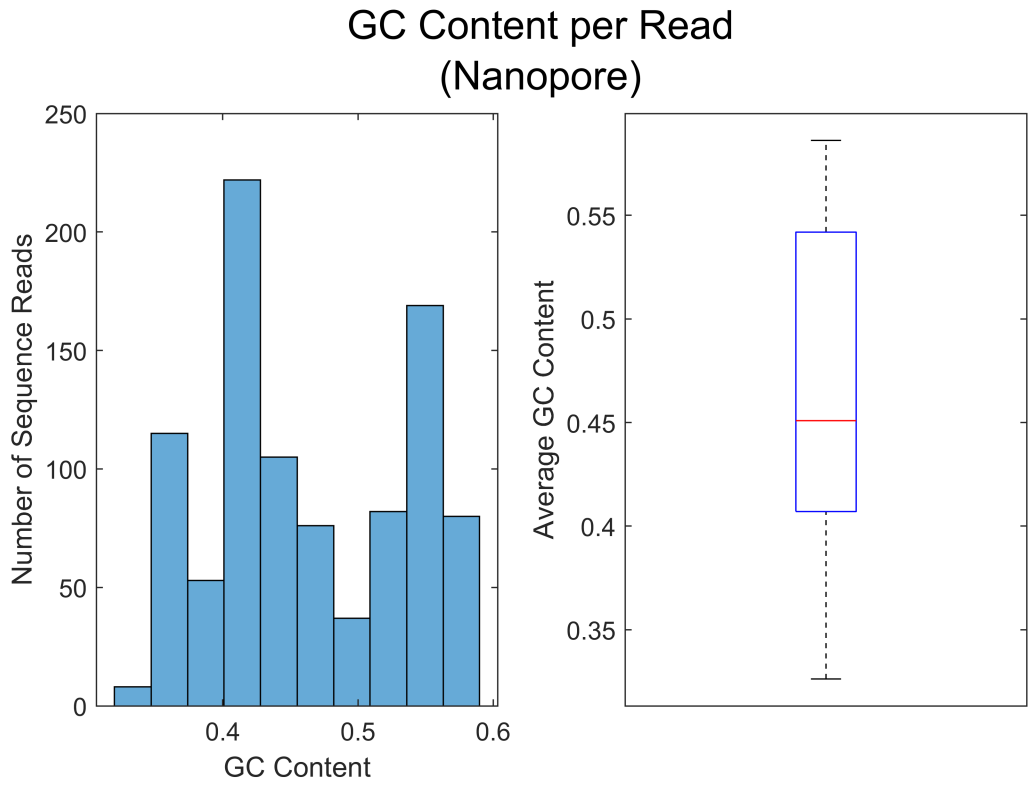
Conclusion: The average quality score for the nanopore tends to be around 18 and seems to follow a slightly skewed normal distribution.

GC Content Histogram per Sequence Read

```
% Determine G and C content of reads
G_nano = cellfun(@(x) numel(strfind(x, 'G')), seqs_nano);
C_nano = cellfun(@(x) numel(strfind(x, 'C')), seqs_nano);

% Determine the GC content
GC_nano = (G_nano + C_nano) ./ readLen_nano;

% Visualize results
figure('Position', [0 0 600 400]), sgtitle({'GC Content per Read', '(Nanopore)'});
subplot(1,2,1), histogram(GC_nano, 10);
xlabel('GC Content'), ylabel('Number of Sequence Reads');
subplot(1,2,2), boxplot(GC_nano), ylabel('Average GC Content'), xticks([]);
```



Caption: The figure shows the distribution of GC content from the nanopore sequencing run. On the left is a histogram showing the frequency distribution of GC content. On the right is a boxplot showing the median, upper and lower quartiles, and outliers of average GC content.

Conclusion: The average GC content for the nanopore tends to be about 45%.

Illumina Sequencing

Input Illumina Sequencing Data (FASTQ File)

```
fastqinfo('lambda_illum.fastq')
```

```
ans = struct with fields:
           Filename: 'lambda_illum.fastq'
           FilePath: 'C:\Users\kabil\OneDrive - Drexel University\Academic\3 - Pre-Junior\1 - Fall Quarter\BMES 375\
        FileModDate: '14-Oct-2021 22:56:55'
           FileSize: 311925841
    NumberOfEntries: 792838
```

```
reads = fastqread('lambda_illum.fastq')
```

reads = 1×792838 struct

Fields	Header	Sequence	Quality
1	'M02486:...	'AGGCAGA...	'CCCCCGG...
2	'M02486:...	'AATCAGC...	'CCCCCGG...
3	'M02486:...	'CCTCAAA...	'CCCCCGG...

Fields	Header	Sequence	Quality
4	'M02486:...	'GTAAGAT...	'CCCCCGG...
5	'M02486:...	'GGGCTGT...	'CCCCCGG...
6	'M02486:...	'TTCCAGC...	'CCCCCGG...
7	'M02486:...	'CTATTTA...	'CCCCCGG...
8	'M02486:...	'GACGTGT...	'CCCCCGG...
9	'M02486:...	'GCGCCGC...	'CCCCCGG...
10	'M02486:...	'GCCCAGC...	'CCCCCGG...
11	'M02486:...	'ATGATGA...	'CCCCCGG...
12	'M02486:...	'ATGCTGC...	'CCCCCGG...
13	'M02486:...	'AAACAAA...	'CCCCCGG...
14	'M02486:...	'GATGTGG...	'CCCCCGG...
15	'M02486:...	'GCTTTAA...	'CCCCCGG...
16	'M02486:...	'CTCACAT...	'CCCCCGG...
17	'M02486:...	'ACACAGA...	'CCCCCGG...
18	'M02486:...	'GCAACCG...	'CCCCCFG...
19	'M02486:...	'GTTCAAA...	'CCCCCGG...
20	'M02486:...	'TTTTAAA...	'CCCCCGG...
21	'M02486:...	'GGTAAAG...	'CCCCCGG...
22	'M02486:...	'GGTGAAT...	'CCCCCGG...
23	'M02486:...	'GACAAGA...	'CCCCCGG...
24	'M02486:...	'CCCCACA...	'CCCCCGG...
25	'M02486:...	'ATGTTGG...	'CCCCCGG...
26	'M02486:...	'CTTGCAG...	'CCCCCGG...
27	'M02486:...	'GACTCAG...	'CCCCCGG...
28	'M02486:...	'TATGATG...	'CCCCCGG...
29	'M02486:...	'TATGATG...	'CCCCCGG...
30	'M02486:...	'TCCCAAA...	'CCCCCGG...
31	'M02486:...	'GTATACC...	'CCCCCGG...
32	'M02486:...	'CTATCAC...	'CCCCCGG...
33	'M02486:...	'ATTCAAA...	'CCCCCGG...
34	'M02486:...	'ATTATGT...	'CCCCCGC...
35	'M02486:...	'ATAACAC...	'CCCCCGG...
36	'M02486:...	'GACCATA...	'CCCCCGG...

Fields	Header	Sequence	Quality
37	'M02486:...	'TGGTACA...	'CCCCCGG...
38	'M02486:...	'GCATAAC...	'CCCCCGF...
39	'M02486:...	'GAATAAT...	'CCCCCGG...
40	'M02486:...	'GATCACC...	'CCCCCGG...
41	'M02486:...	'GACTATA...	'CCCCCGG...
42	'M02486:...	'ACCAGAT...	'CCCCCGG...
43	'M02486:...	'GCTCTTT...	'CCCCCGG...
44	'M02486:...	'GAAAAGG...	'CCCCCGG...
45	'M02486:...	'TTTTTCA...	'CCCCCGC...
46	'M02486:...	'GCCATCA...	'CCCCCGG...
47	'M02486:...	'GTGTGTC...	'CCCC-CF...
48	'M02486:...	'ATTCAAA...	'CCCCCGG...
49	'M02486:...	'CTGCCAC...	'CCCCCGG...
50	'M02486:...	'TGTGTGG...	'CCCCCGG...
51	'M02486:...	'GATCTGG...	'CCCCCGG...
52	'M02486:...	'GGTTTCT...	'CCCCCGG...
53	'M02486:...	'AGCGAGA...	'CCCCCGG...
54	'M02486:...	'GTATTAA...	'CCCCCGG...
55	'M02486:...	'ACATTAG...	'CCCCCGG...
56	'M02486:...	'TCTCTAT...	'CCCCCGG...
57	'M02486:...	'AATGAGT...	'CCCCCGG...
58	'M02486:...	'GGTGCGC...	'CCCCCGG...
59	'M02486:...	'CTTAATT...	'CCCC?FF...
60	'M02486:...	'GCTCCAT...	'CCCCCGG...
61	'M02486:...	'ATTTAAA...	'CCCCCGG...
62	'M02486:...	'GGGCTGA...	'CCCCCGG...
63	'M02486:...	'CTCGATT...	'CCCCCGG...
64	'M02486:...	'ATTTGGT...	'CCCCCGG...
65	'M02486:...	'GCCTCAT...	'CCCCCGG...
66	'M02486:...	'CGGTGTT...	'CCCCCGG...
67	'M02486:...	'GATATAG...	'CCCCCGG...
68	'M02486:...	'GAAGATA...	'CCCCCAF...
69	'M02486:...	'ATTTTCA...	'CCCCCGG...

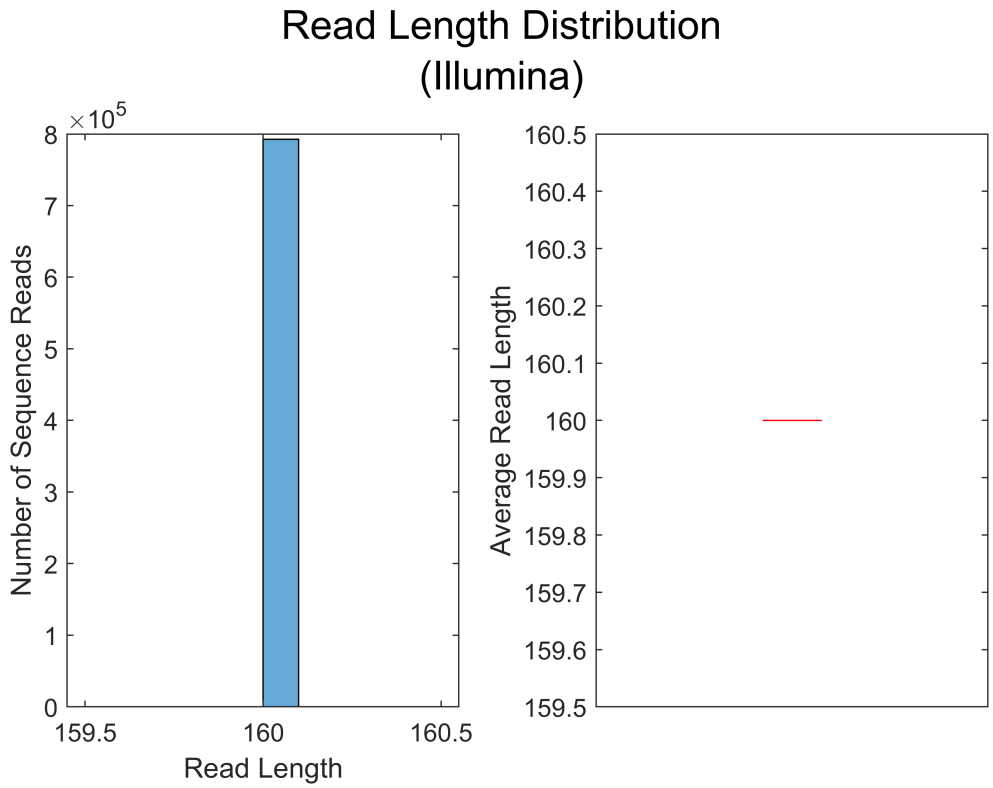
Fields	Header	Sequence	Quality
70	'M02486:...	'TTATTAT...	'CCCCCGG...
71	'M02486:...	'TGCTGAT...	'<CCCCGG...
72	'M02486:...	'AAGTAAT...	'9<ACB9F...
73	'M02486:...	'GGTTTAG...	'B@ACCGF...
74	'M02486:...	'ATATTGC...	'CCCCCGG...
75	'M02486:...	'GCCTGGG...	'CCCCCGG...
76	'M02486:...	'AGAAAAA...	'@C@CCFG...
77	'M02486:...	'CGGGAGA...	'CCCCCGG...
78	'M02486:...	'AAGCTAT...	'CCCCCGG...
79	'M02486:...	'ATGCAAT...	'CCCCCGG...
80	'M02486:...	'GTGAGAT...	'CCCCCGG...
81	'M02486:...	'CGGTGGT...	'CCCCCGD...
82	'M02486:...	'TATTACG...	'CCCCCGG...
83	'M02486:...	'CATGTGC...	'ACCCCGG...
84	'M02486:...	'GCCTTTA...	'CCCCCGG...
85	'M02486:...	'TCGTTTT...	'CCC@CFF...
86	'M02486:...	'GGTATAA...	'CCCCCGG...
87	'M02486:...	'GCACACA...	'CCCCCGG...
88	'M02486:...	'GTACGCC...	'CCCCCGG...
89	'M02486:...	'CTGTTGG...	'CCCCCGG...
90	'M02486:...	'GTATCGT...	'CCCCCGG...
91	'M02486:...	'CATTCCA...	'CCCCCGG...
92	'M02486:...	'GGGATAA...	'<BCCCGG...
93	'M02486:...	'GTGAGTC...	'CCCCCGG...
94	'M02486:...	'AGCTTAT...	'CC8CCGG...
95	'M02486:...	'TGATCGG...	'CCCCCGG...
96	'M02486:...	'ATTCAAC...	'CCCCCGG...
97	'M02486:...	'ACGGGAT...	'CCCCCGG...
98	'M02486:...	'GTATCGG...	'CCCCCGG...
99	'M02486:...	'GCATCAG...	'CCCCCGG...
100	'M02486:...	'GCCTAAT...	'CCCCCGG...

⋮

Read Length Distribution of All Sequencing Reads

```
seqs_ilum = {reads.Sequence};
readLen_ilum = cellfun(@length, seqs_ilum);

% Visualize results
figure('Position', [0 0 600 400]), sgtitle({'Read Length Distribution', '(Illumina)'});
subplot(1,2,1), histogram(readLen_ilum, 10);
xlabel('Read Length'), ylabel('Number of Sequence Reads');
subplot(1,2,2), boxplot(readLen_ilum), ylabel('Average Read Length'), xticks([]);
```



Caption: The figure shows the distribution of sequence read lengths from the Illumina sequencing run. On the left is a histogram showing the frequency distribution of read lengths. On the right is a boxplot showing the median, upper and lower quartiles, and outliers of average sequence read length.

Conclusion: The read length is consistent at 160 bps for all samples.

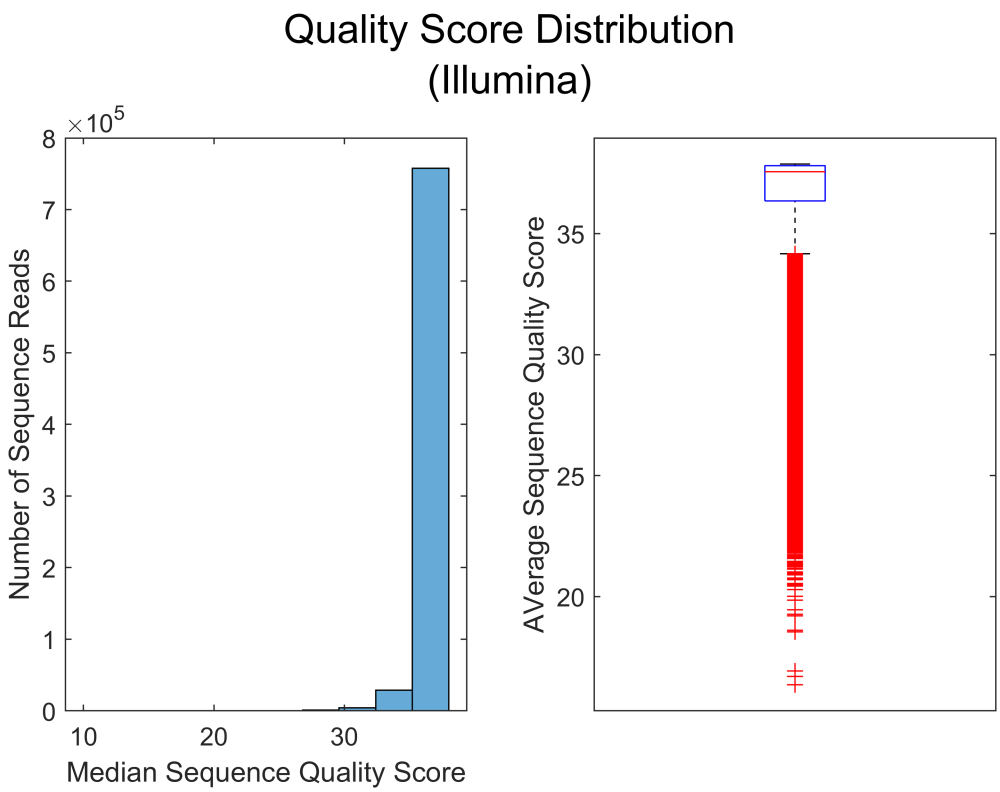
Quality Score Distribution per Sequencing Read

```
seqQ_ilum = {reads.Quality}; % ASCII format
seqQS_ilum = cellfun(@(x) double(x) - 33, seqQ_ilum, 'UniformOutput', false);

% Average, Median & Standard Deviation
avgQS_ilum = cellfun(@mean, seqQS_ilum);
medQS_ilum = cellfun(@median, seqQS_ilum);
stdQS_ilum = cellfun(@std, seqQS_ilum);

% Plot Distribution of Median and Average Quality
figure('Position', [0 0 600 400]), sgtitle({'Quality Score Distribution', '(Illumina)'});
subplot(1,2,1), histogram(medQS_ilum, 10);
xlabel('Median Sequence Quality Score'), ylabel('Number of Sequence Reads');
```

```
subplot(1,2,2), boxplot(avgQS_ilum), ylabel('AVerage Sequence Quality Score'), xticks([]);
```



Caption: The figure shows the distribution of sequence quality scores from the Illumina sequencing run. On the left is a histogram showing the frequency distribution of quality scores. On the right is a boxplot showing the median, upper and lower quartiles, and outliers of average sequence quality score.

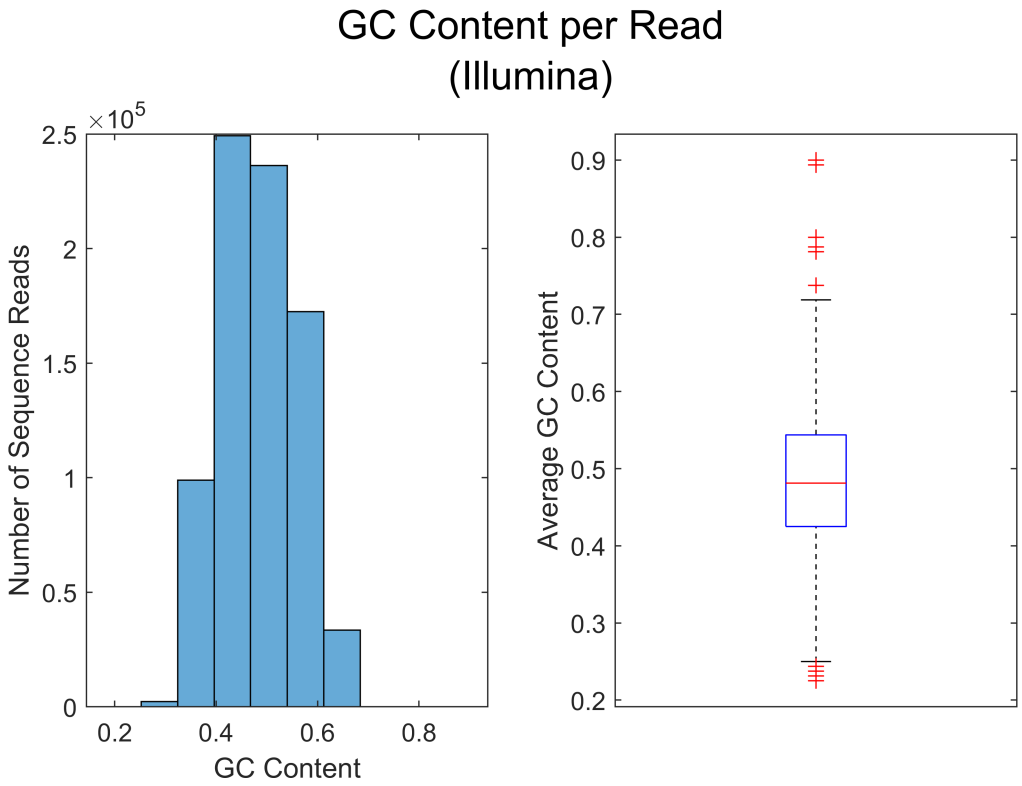
Conclusion: The quality scores remain relatively consistent at 36. However, there are several outliers with quality scores as low as 20.

GC Content Histogram per Sequence Read

```
% Determine G and C content of reads
G_ilum = cellfun(@(x) numel(strfind(x, 'G')), seqs_ilum);
C_ilum = cellfun(@(x) numel(strfind(x, 'C')), seqs_ilum);

% Determine the GC content
GC_ilum = (G_ilum + C_ilum) ./ readLen_ilum;

% Visualize results
figure('Position', [0 0 600 400]), sgtitle({'GC Content per Read', '(Illumina)'});
subplot(1,2,1), histogram(GC_ilum, 10);
xlabel('GC Content'), ylabel('Number of Sequence Reads');
subplot(1,2,2), boxplot(GC_ilum), ylabel('Average GC Content'), xticks([]);
```



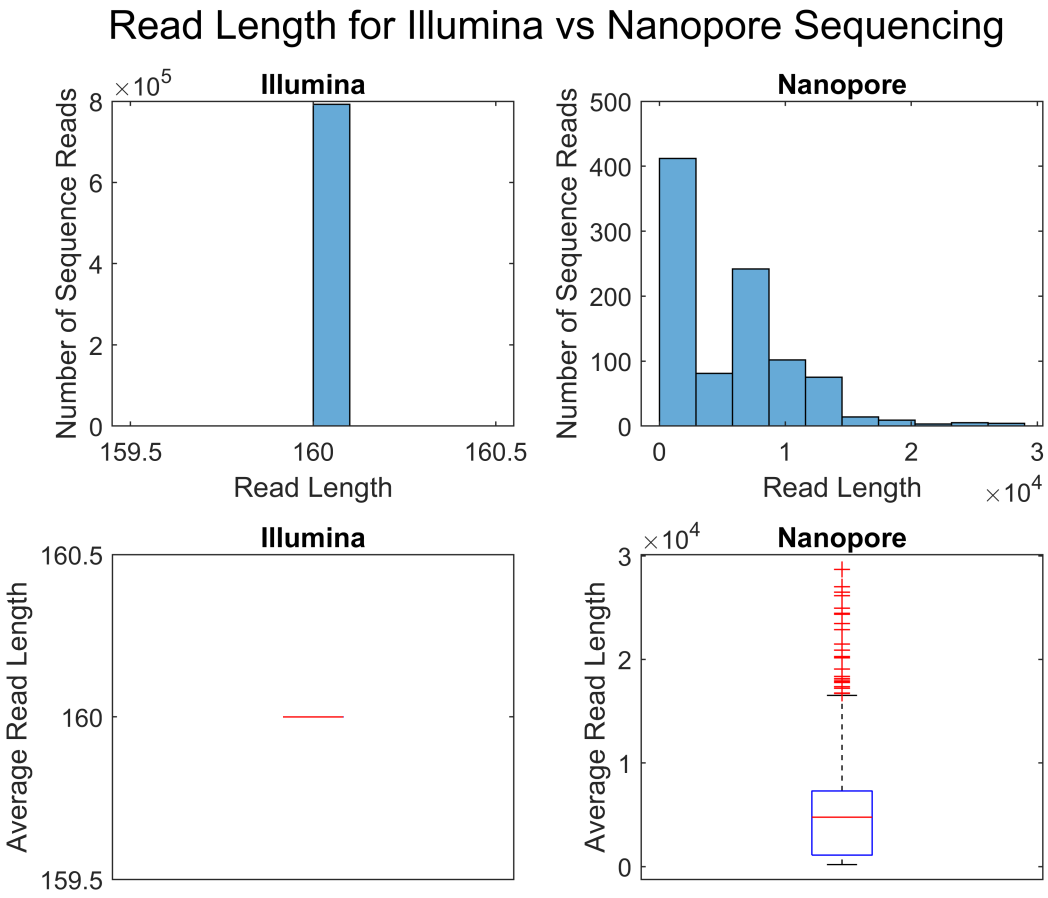
Caption: The figure shows the distribution of GC content from the Illumina sequencing run. On the left is a histogram showing the frequency distribution of GC content. On the right is a boxplot showing the median, upper and lower quartiles, and outliers of average GC content.

Conclusion: The average GC content for the Illumina sequencing run is roughle normally distributed with a mean of about 50%.

Differences Between Illumina and Nanopore Sequencing

First difference: Plot the nanopore and Illumina data together using histogram and box plot (caption and conclusion)

```
figure('Position', [0 0 600 500]), sgtitle('Read Length for Illumina vs Nanopore Sequencing')
subplot(2,2,1), histogram(readLen_ilum, 10), title('Illumina');
xlabel('Read Length'), ylabel('Number of Sequence Reads');
subplot(2,2,3), boxplot(readLen_ilum), title('Illumina'), ylabel('Average Read Length'), xticks
subplot(2,2,2), histogram(readLen_nano, 10), title('Nanopore');
xlabel('Read Length'), ylabel('Number of Sequence Reads');
subplot(2,2,4), boxplot(readLen_nano), title('Nanopore'), ylabel('Average Read Length'), xticks
```



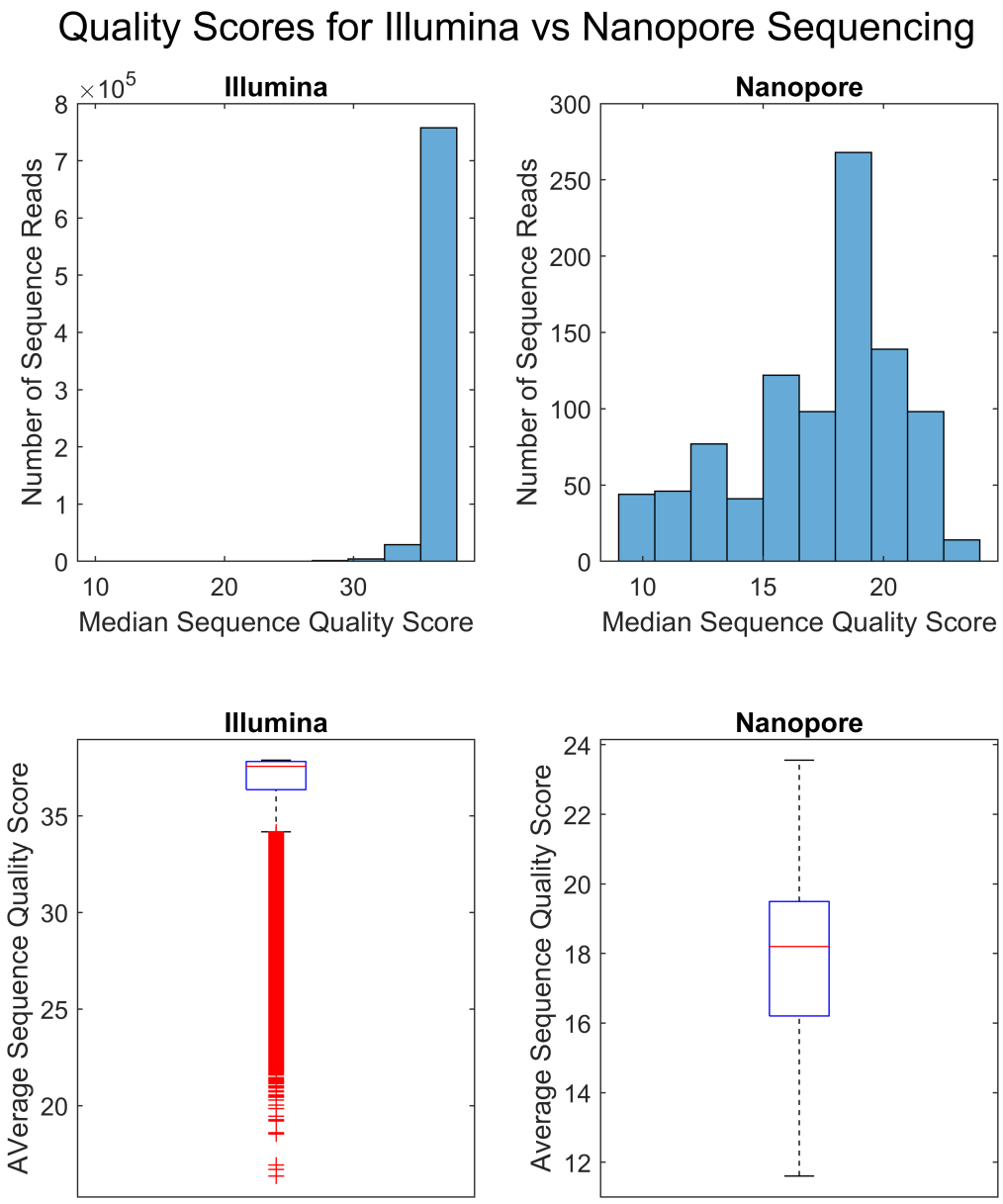
Caption: The figure shows histograms (top left & right) and boxplots (bottom left & right) showing the distibution of read lengths for illumina (left panes) and nanopore (right panes) sequencing.

Conclusion: Illumina sequencing produces reads with a consistent read length (in this case, 160bp), while nanopore sequencing produces reads of varying lengths. The read length for nanopore is also orders of magnitude larger than the read length for Illumina sequencing. This is due to underlying differences in the sequencing technologies. Illumina sequencing kits allow for a specific number of reads per cycle (usually in the range of 150 - 300bps), while nanopore sequencing allows much larger fragments of DNA to be sequenced at a time.

Second difference: Plot the nanopore and Illumina data together using histogram and box plot (caption and conclusion)

```
figure('Position', [0 0 600 700]), sgtitle('Quality Scores for Illumina vs Nanopore Sequencing
subplot(2,2,1), histogram(medQS_ilum, 10), title('Illumina');
xlabel('Median Sequence Quality Score'), ylabel('Number of Sequence Reads');
subplot(2,2,3), boxplot(avgQS_ilum), title('Illumina');
ylabel('AVerage Sequence Quality Score'), xticks([]);
subplot(2,2,2), histogram(medQS_nano, 10), title('Nanopore');
xlabel('Median Sequence Quality Score'), ylabel('Number of Sequence Reads');
subplot(2,2,4), boxplot(avgQS_nano), title('Nanopore');
```

```
ylabel('Average Sequence Quality Score'), xticks([]);
```

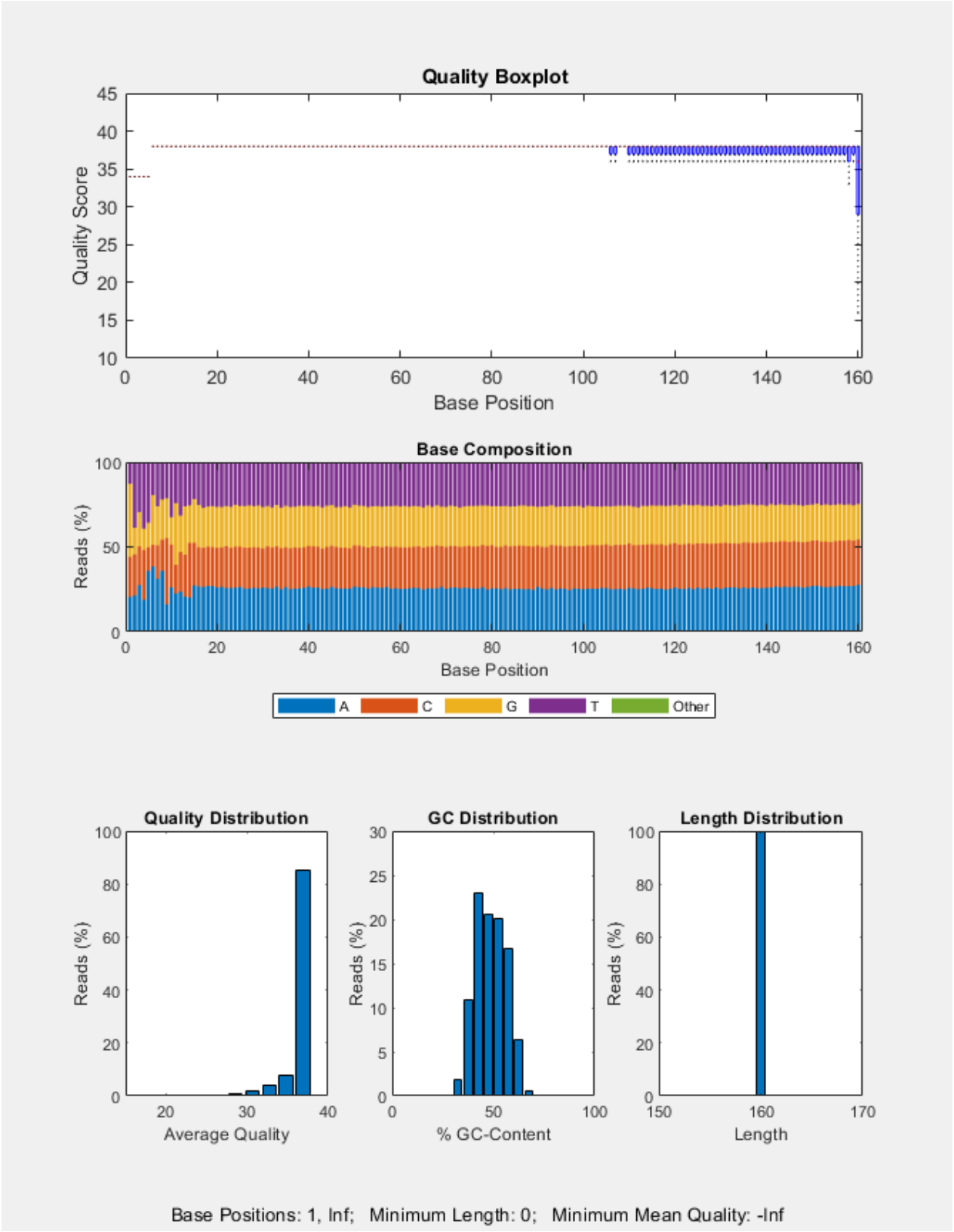


Caption: The figure shows histograms (top left & right) and boxplots (bottom left & right) showing the distibution of quality scores for illumina (left panes) and nanopore (right panes) sequencing.

Conclusion: Illumina sequencing produeces higher quality reads than nanopore sequencing. The quality scores are also significantly more consistent for Illumina sequencing. With nanopore quality cores being roughly normally distributed and illumina quality scores being mostly constant at 36, with a few outliers as low as 20.

`seqqcplot()` on Illumina Data

```
seqqcplot('lambda_illum.fastq');
```




```
%seqqcplot('lambda_nanopore.fastq')  %what happens and why?
```

The `seqqcplot` function does not work for nanopore sequencing data because the function currently does not support the encoding for nanopore.

Filter Sequencing Reads

```
[outFile, in, out] = seqfilter('lambda_nanopore.fastq', 'Method', 'MeanQuality', 'Threshold', 1
```

```
outFile = 1×1 cell array

    {'C:\Users\kabil\OneDrive - Drexel University\Academic\3 - Pre-Junior\1 - Fall Quarter\BMES 375\bmes375.TonyOkel
in = 797
out = 150
```