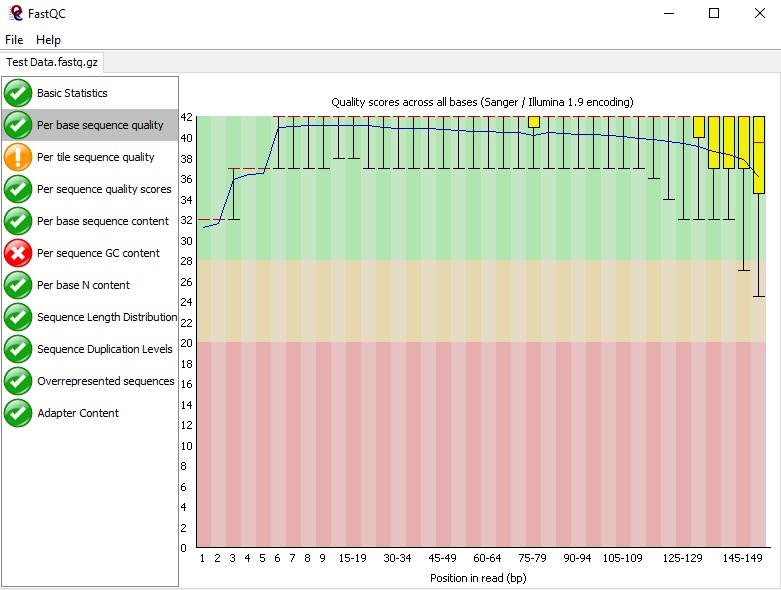


Per Base Sequence Quality

* ***For each base pair position, a box plot is drawn with:***



○ **the median value (red line)**

○ **the interquartile range (25-**

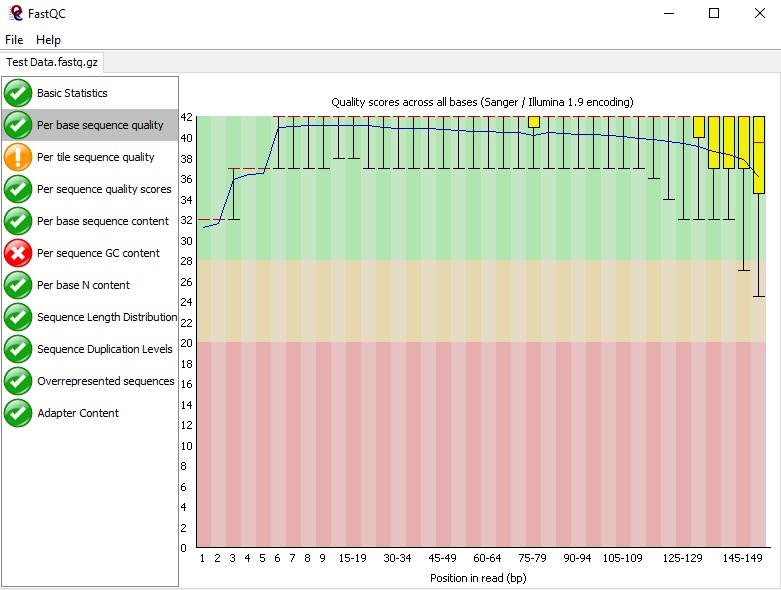
**75%) (yellow box)**

○ **the mean quality (blue line)**

* ***So, rather than looking at quality scores for each individual read, this module looks at quality collectively across all reads within a sample.***

Per Base Sequence Quality

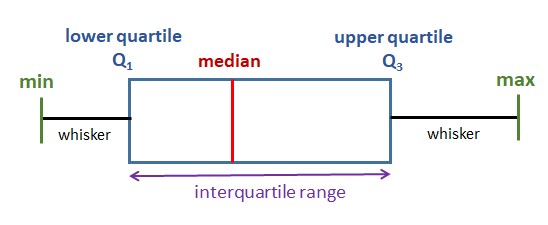
* ***X-axis:* base pair positions**



* ***Y-axis:* quality scores**
* **The background of the graph divides the y-axis into very good quality scores (green), scores of reasonable quality (orange), and reads of poor quality**

**(red).**

# Side Note: Box & Whisker Plot



# Exercise 1

* Write a python script that reads the given FASTQ file and:

◆ **Calculates the “Per base sequence quality” (mean and median only)**

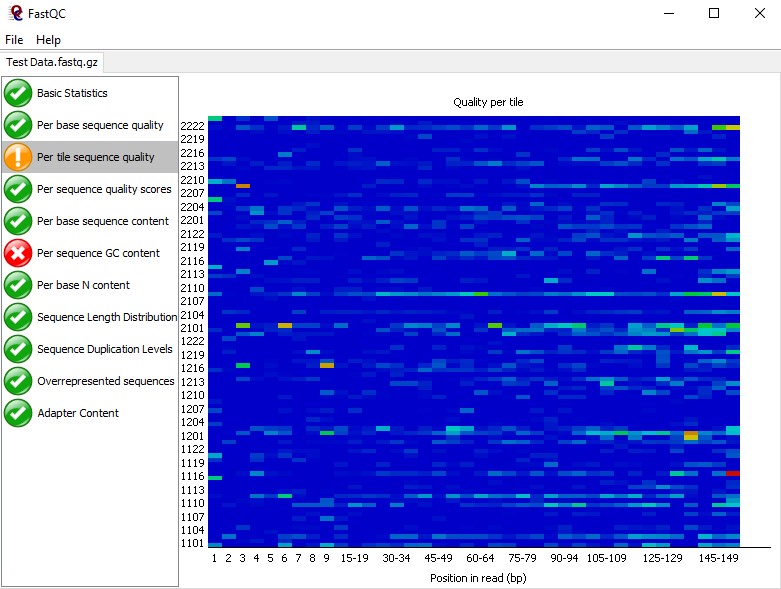
● ***Hint:* Use sorted() built-in function to sort a list and easily get the median.**

◆ **Draws a plot (any kind of plot you prefer) for this metric**

* Open the same FASTQ file in FastQC and compare the results of “Per base sequence quality” to your output.

Per Tile Sequence Quality

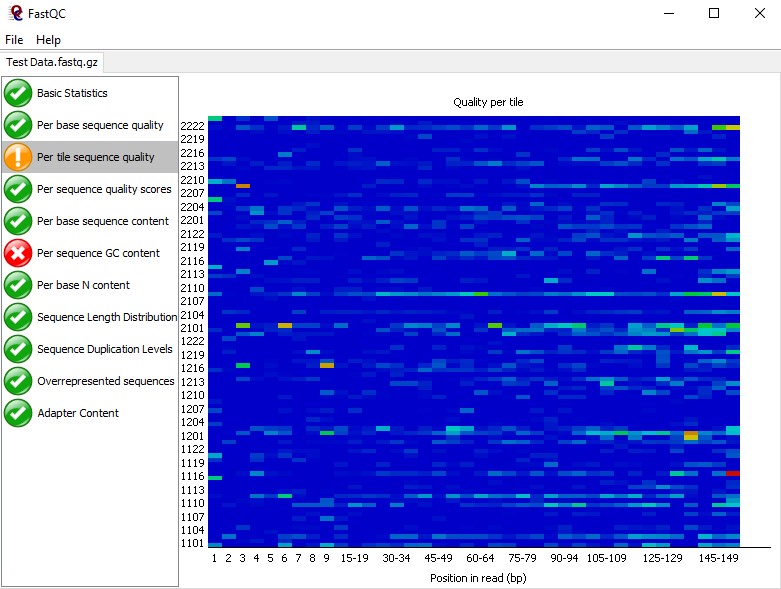
* ***This graph displays the quality scores from each tile across all of the bases to see if there was a loss in quality associated with only one part of the flow cell.***



* **Only available if you're using an Illumina library.**

Per Tile Sequence Quality

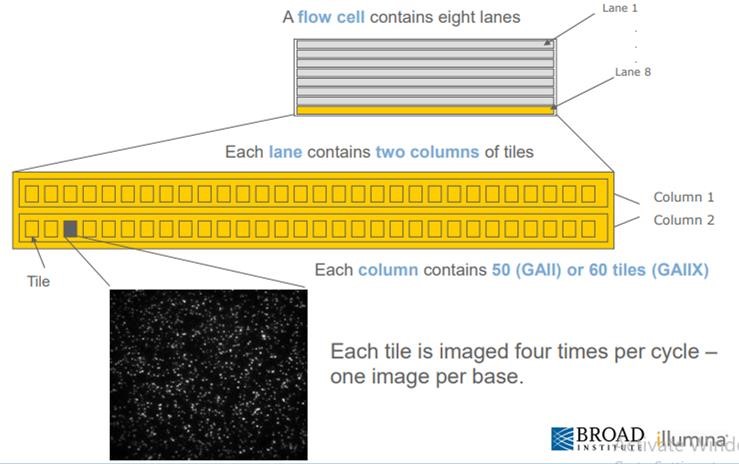
* **The colours are on a cold to hot scale, with cold colours being positions where the quality was at or above the average for that base in the run, and hotter colours indicate that a tile had worse qualities than other tiles for that base.**



* **This means that the plot shows the deviation from the average quality for each tile.**

Side Note: Tiles in a Flow Cell

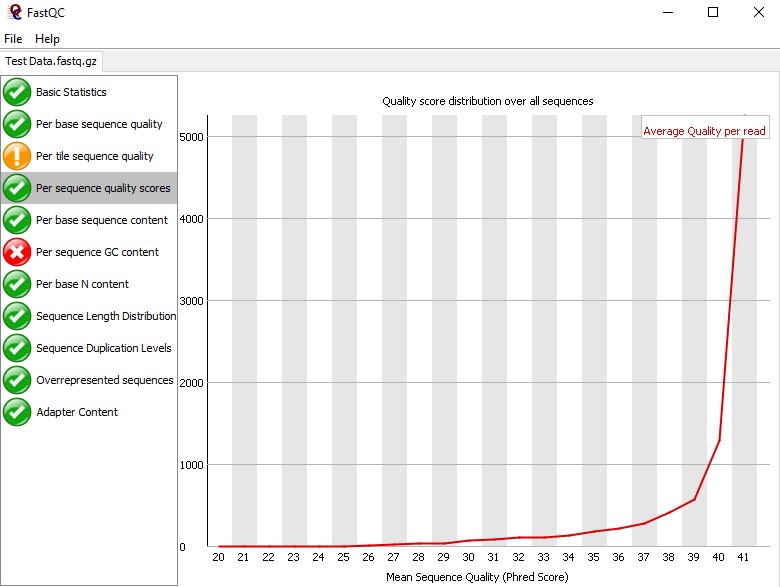
* **The flow cell is a glass slide used during sequencing. Its surface is coated with a dense lawn of oligos.**



* **It contains small fluidic channels through which polymerases can be pumped.**

Per Sequence Quality Scores

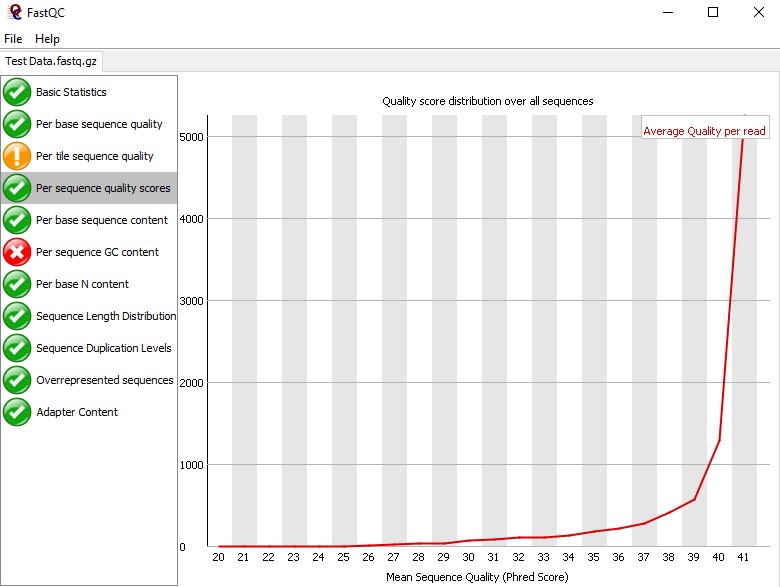
* ***For each read (sequence), this module calculates the average quality score over the full length of the read.***



* ***Then it plots the quality averages against the number of reads having them.***

Per Sequence Quality Scores

* ***X-axis:* average quality scores over the full length of all reads**



* ***Y-axis:* the total number of reads with this score**

# Exercise 2

* In the previously used FASTQ file, write a python script that:

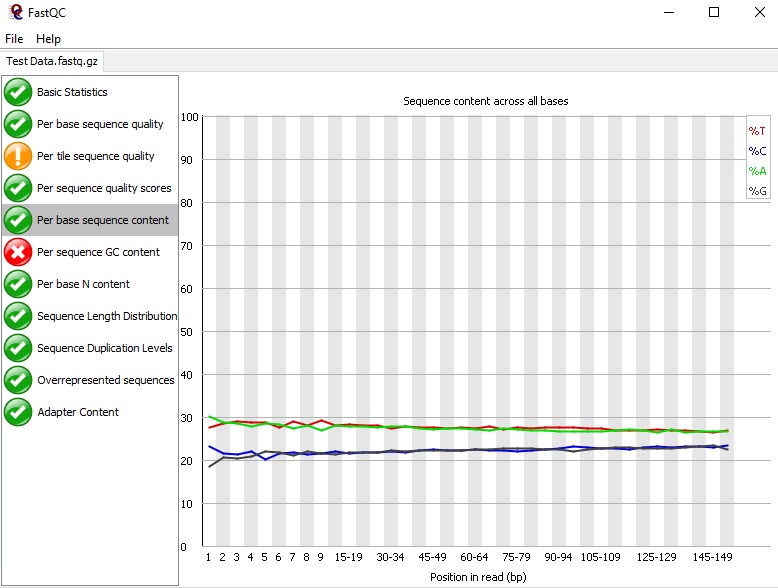
◆ **Calculate the “Per sequence quality scores”**

◆ **Draws a plot (any kind of plot you prefer) for this metric**

* Open the same FASTQ file in FastQC and compare the results of “Per sequence quality scores” to your output.

Per Base Sequence Content

* ***For each position in the reads, this module plots the percentage of each nucleotide at this position across all reads.***



* ***X-axis:* positions**
* ***Y-axis:* percentage of each nucleotide**
* ***Color-coded nucleotides***

# Exercise 3

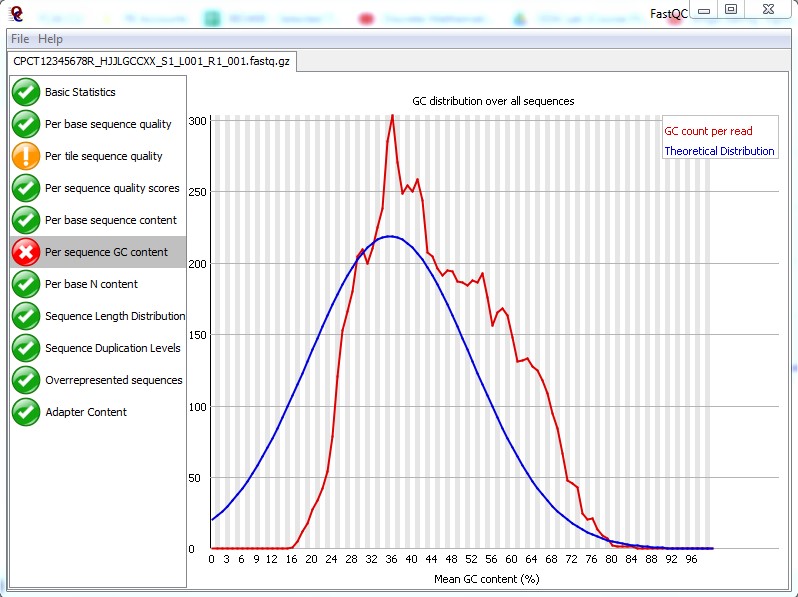
* In the previously used FASTQ file, write a python script that:

◆ **Calculates the “Per base sequence content”**

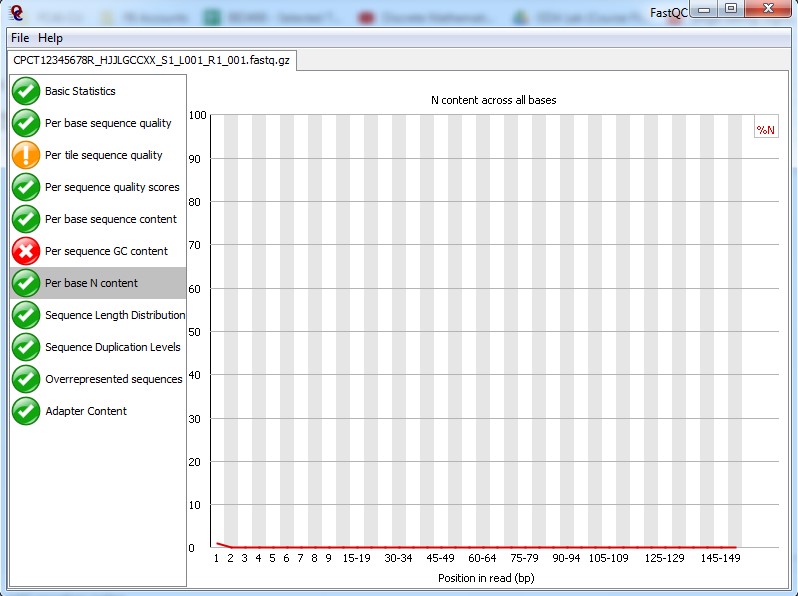
◆ **Draws a plot (any kind of plot you prefer) for this metric**

* Open the same FASTQ file in FastQC and compare the results of “Per base sequence content” to your output.

Per Sequence GC Content

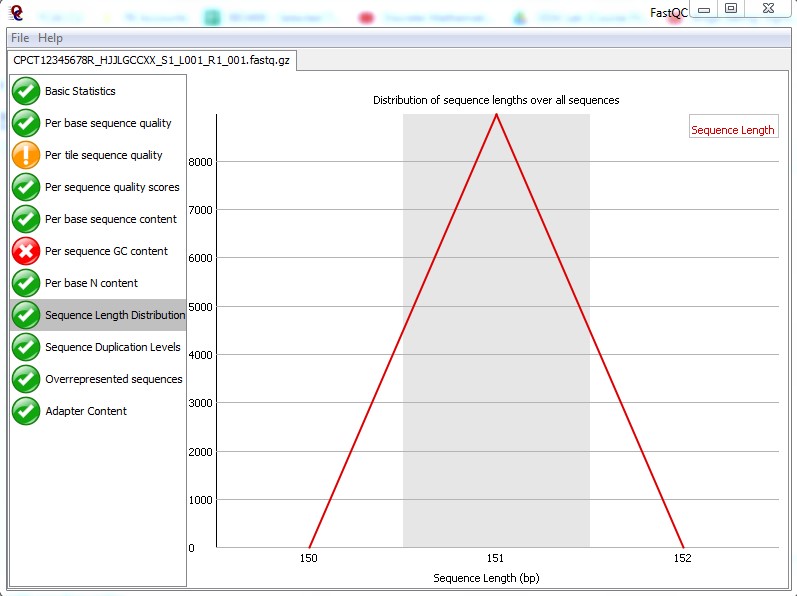
* ***This module plots the distribution of GC content over all sequences.***
* **The blue line is a theoretical normal distribution.**
* **The red line shows the actual distribution of the given data.**
* ***X-axis:* mean GC content (%)**
* ***Y-axis:* number of reads having the corresponding mean GC value**

Per Base N Content

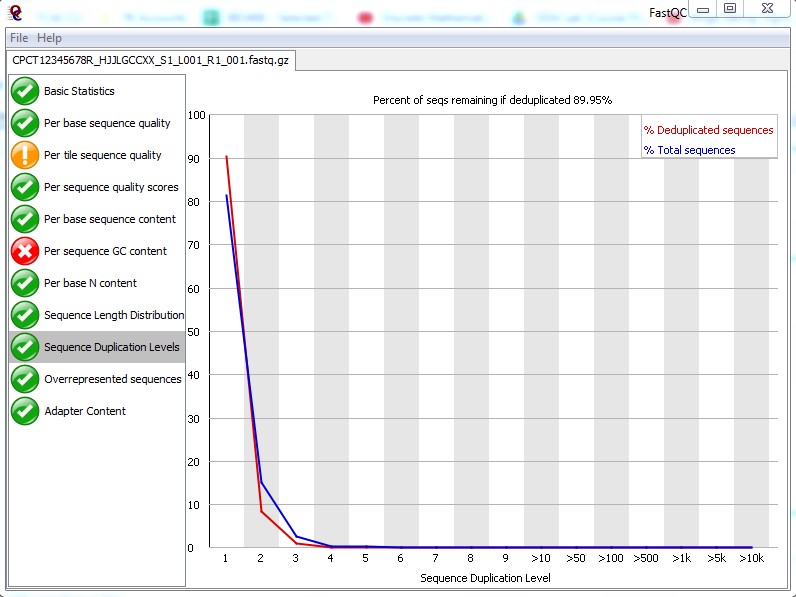
* ***This module indicates the N content over all bases.***
* **It also tells if the given sample has any uncalled bases.**
* ***X-axis:* position in read**
* ***Y-axis:* percentage of N in that position over all**

**sequences**

Sequence Length Distribution

* ***This module indicates the distribution of sequence lengths over all sequences.***
* ***X-axis:* sequence length**
* ***Y-axis:* number of reads having that length**

Sequence Duplication Levels

* ***This module offers a way of finding how unique the sequences are in the given sample.***
* **Most of the sequence should have occurred only once.**
* ***X-axis:* duplication level**

**(occurred once, twice...)**

* ***Y-axis:* percentage of sequences at specific**

**duplication level**

# Exercise 4

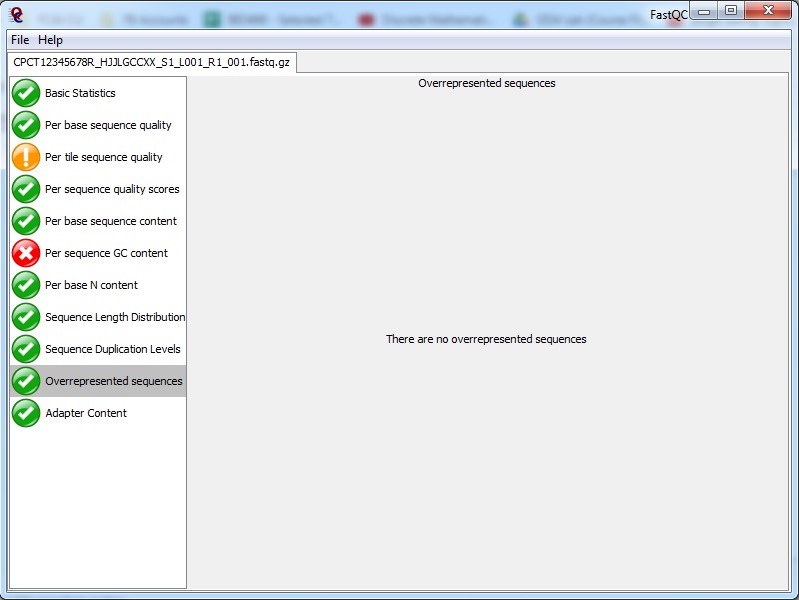
* In the previously used FASTQ file, write a python script that:

◆ **Calculates the “Sequence Duplication Levels”**

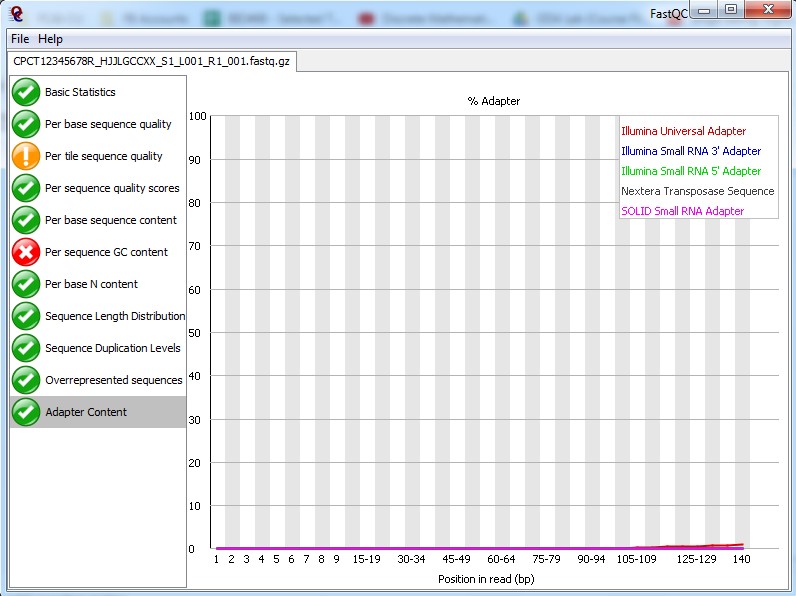
◆ **Draws a plot (any kind of plot you prefer) for this metric**

* Open the same FASTQ file in FastQC and compare the results of “Sequence Duplication Levels” to your output.

Overrepresented Sequences

* ***This module shows the list of sequences which appear more than expected in the file.***
* **A sequence is considered overrepresented if it accounts for** ≥ **0.1% of the total reads.**

Adapter Content

* ***This module shows a cumulative plot of the fraction of reads where the sequence library adapter sequence is identified at the indicated base position.***
* ***X-axis:* position in read**
* ***Y-axis:* percentage of reads in which an adapter**

**is present**