A modular tool to aggregate results from bioinformatics analyses across many samples into a single report.

Report generated on 2020-12-11, 13:50 based on data in:
`/remote-fs/storix2/student/2020-2021/Thema06/project-data/Thema06-groep1/results/fastqc`

## General Statistics

| Sample Name | % Dups | % GC | Length | % Failed | M Seqs |
|---|---|---|---|---|---|
| SRR018013_1 | 41.8% | 53% | 35 bp | 50% | 72.7 |
| SRR018013_2 | 43.7% | 54% | 35 bp | 50% | 67.5 |
| SRR018015 | 4.0% | 50% | 35 bp | 30% | 92.9 |
| SRR057598 | 42.6% | 56% | 40 bp | 9% | 15.0 |
| SRR057599 | 41.9% | 54% | 40 bp | 18% | 15.8 |
| SRR1106118 | 26.7% | 31% | 32 bp | 30% | 6.8 |
| SRR1106138 | 8.3% | 30% | 32 bp | 30% | 5.6 |
| SRR1106139 | 10.0% | 30% | 32 bp | 30% | 6.8 |
| SRR1106140 | 6.4% | 30% | 32 bp | 30% | 4.2 |

## FastQC

FastQC is a quality control tool for high throughput sequence data, written by Simon Andrews at the Babraham Institute in Cambridge.

### Sequence Counts Help

Sequence counts for each sample. Duplicate read counts are an estimate only.

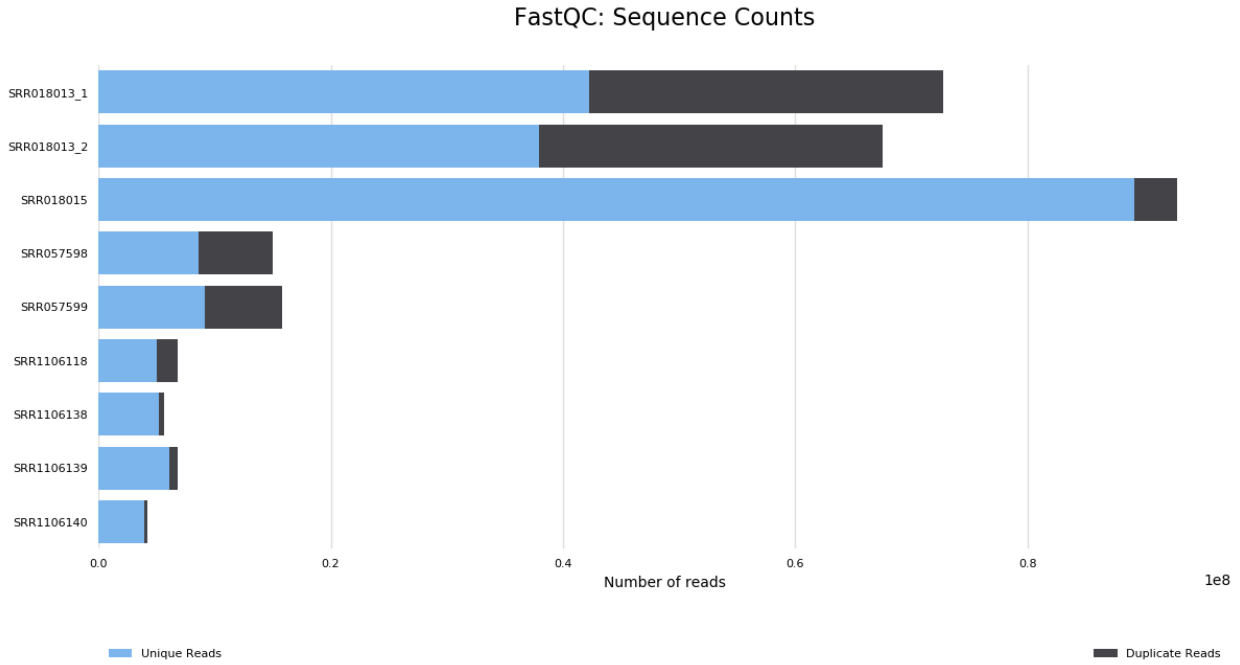
This plot show the total number of reads, broken down into unique and duplicate if possible (only more recent versions of FastQC give duplicate info).

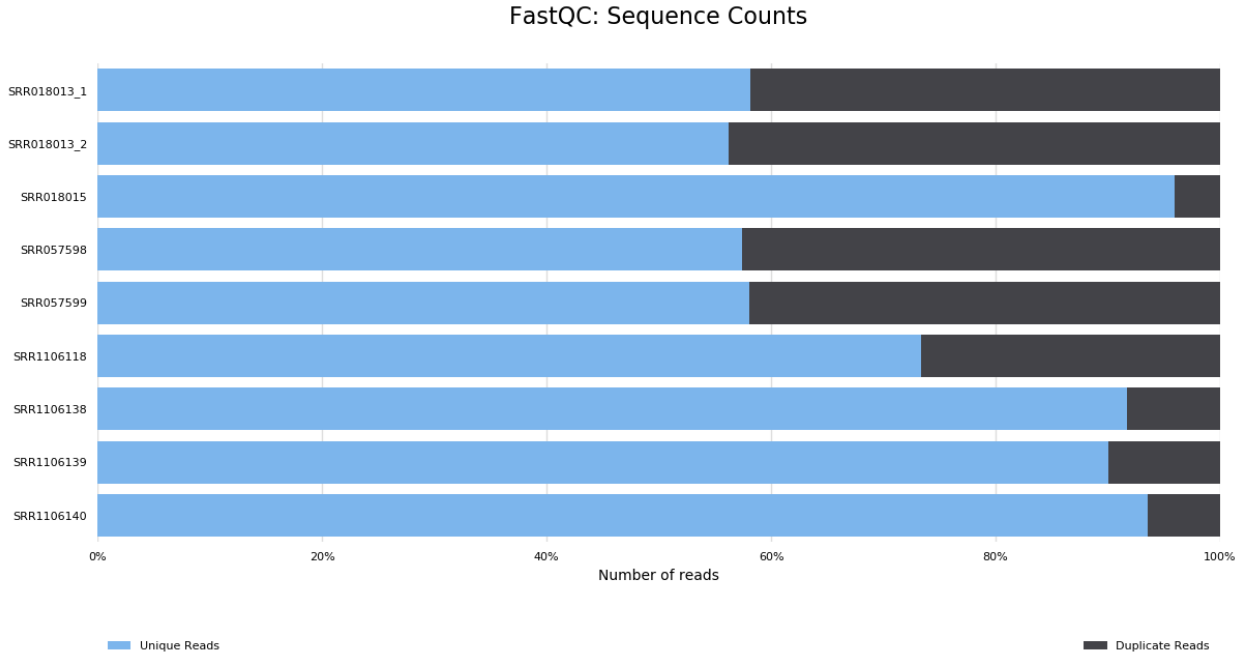
You can read more about duplicate calculation in the FastQC documentation. A small part has been copied here for convenience:

*Only sequences which first appear in the first 100,000 sequences in each file are analysed. This should be enough to get a good impression for the duplication levels in the whole file. Each sequence is tracked to the end of the file to give a representative count of the overall duplication level.*

*The duplication detection requires an exact sequence match over the whole length of the sequence. Any reads over 75bp in length are truncated to 50bp for this analysis.*

Flat image plot. Toolbox functions such as highlighting / hiding samples will not work (see the docs).




---

**Sequence Quality Histograms Help**

The mean quality value across each base position in the read.

To enable multiple samples to be plotted on the same graph, only the mean quality scores are plotted (unlike the box plots seen in FastQC reports).

Taken from the FastQC help:

*The y-axis on the graph shows the quality scores. The higher the score, the better the base call. The background of the graph divides the y axis into very good quality calls (green), calls of reasonable quality (orange), and calls of poor quality (red). The quality of calls on most platforms will degrade as the run progresses, so it is common to see base calls falling into the orange area towards the end of a read.*

loading..

---

**Per Sequence Quality Scores Help**

The number of reads with average quality scores. Shows if a subset of reads has poor quality.

From the FastQC help:

*The per sequence quality score report allows you to see if a subset of your sequences have universally low quality values. It is often the case that a subset of sequences will have universally poor quality, however these should represent only a small percentage of the total sequences.*

loading..

---

**Per Base Sequence Content Help**

The proportion of each base position for which each of the four normal DNA bases has been called.

To enable multiple samples to be shown in a single plot, the base composition data is shown as a heatmap. The colours represent the balance between the four bases: an even distribution should give an even muddy brown colour. Hover over the plot to see the percentage of the four bases under the cursor.

**To see the data as a line plot, as in the original FastQC graph, click on a sample track.**

From the FastQC help:

*Per Base Sequence Content plots out the proportion of each base position in a file for which each of the four normal DNA bases has been called.*

*In a random library you would expect that there would be little to no difference between the different bases of a sequence run, so the lines in this plot should run parallel with each other. The relative amount of each base should reflect the overall amount of these bases in your genome, but in any case they should not be hugely imbalanced from each other.*

*It's worth noting that some types of library will always produce biased sequence composition, normally at the start of the read. Libraries produced by priming using random hexamers (including nearly all RNA-Seq libraries) and those which were fragmented using transposases inherit an intrinsic bias in the positions at which reads start. This bias does not concern an absolute sequence, but instead provides enrichement of a number of different K-mers at the 5' end of the reads. Whilst this is a true technical bias, it isn't something which can be corrected by trimming and in most cases doesn't seem to adversely affect the downstream analysis.*

Click a sample row to see a line plot for that dataset.

**Rollover for sample name**

Export Plot

Position: -

%T: -

%C: -

%A: -

%G: -

---

**Per Sequence GC Content Help**

The average GC content of reads. Normal random library typically have a roughly normal distribution of GC content.

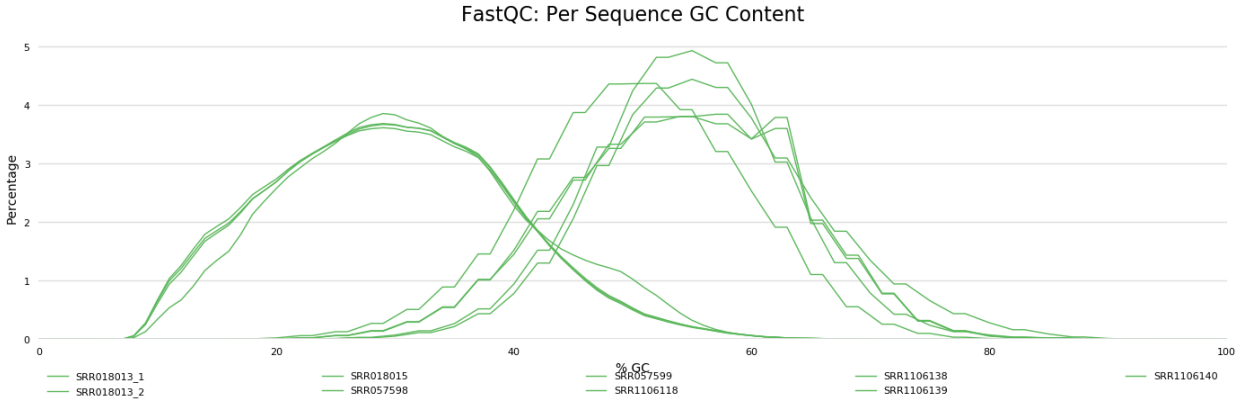
From the FastQC help:

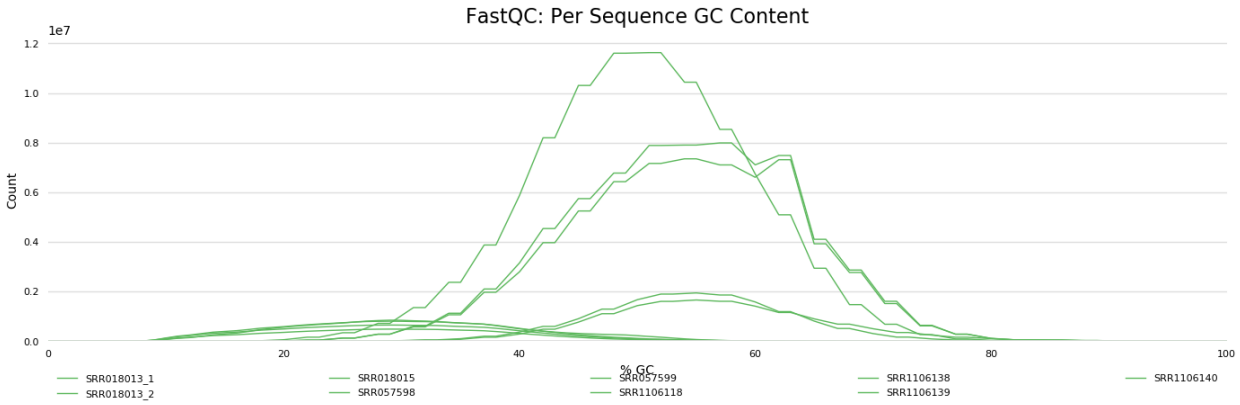
*This module measures the GC content across the whole length of each sequence in a file and compares it to a modelled normal distribution of GC content.*

*In a normal random library you would expect to see a roughly normal distribution of GC content where the central peak corresponds to the overall GC content of the underlying genome. Since we don't know the the GC content of the genome the modal GC content is calculated from the observed data and used to build a reference distribution.*

*An unusually shaped distribution could indicate a contaminated library or some other kinds of biased subset. A normal distribution which is shifted indicates some systematic bias which is independent of base position. If there is a systematic bias which creates a shifted normal distribution then this won't be flagged as an error by the module since it doesn't know what your genome's GC content should be.*

Flat image plot. Toolbox functions such as highlighting / hiding samples will not work (see the docs).




---

**Per Base N Content Help**

The percentage of base calls at each position for which an `N` was called.

From the FastQC help:

*If a sequencer is unable to make a base call with sufficient confidence then it will normally substitute an `N` rather than a conventional base call. This graph shows the percentage of base calls at each position for which an `N` was called.*

*It's not unusual to see a very low proportion of Ns appearing in a sequence, especially nearer the end of a sequence. However, if this proportion rises above a few percent it suggests that the analysis pipeline was unable to interpret the data well enough to make valid base calls.*

loading..

---

**Sequence Length Distribution**

The distribution of fragment sizes (read lengths) found. See the FastQC help

loading..

---

### Sequence Duplication Levels Help

The relative level of duplication found for every sequence.

From the FastQC Help:

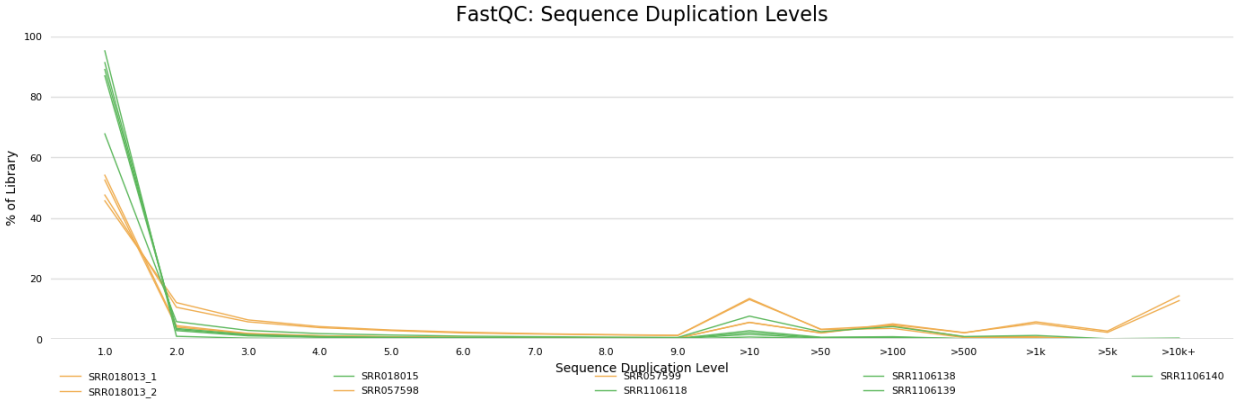
*In a diverse library most sequences will occur only once in the final set. A low level of duplication may indicate a very high level of coverage of the target sequence, but a high level of duplication is more likely to indicate some kind of enrichment bias (eg PCR over amplification). This graph shows the degree of duplication for every sequence in a library: the relative number of sequences with different degrees of duplication.*

*Only sequences which first appear in the first 100,000 sequences in each file are analysed. This should be enough to get a good impression for the duplication levels in the whole file. Each sequence is tracked to the end of the file to give a representative count of the overall duplication level.*

*The duplication detection requires an exact sequence match over the whole length of the sequence. Any reads over 75bp in length are truncated to 50bp for this analysis.*

*In a properly diverse library most sequences should fall into the far left of the plot in both the red and blue lines. A general level of enrichment, indicating broad oversequencing in the library will tend to flatten the lines, lowering the low end and generally raising other categories. More specific enrichments of subsets, or the presence of low complexity contaminants will tend to produce spikes towards the right of the plot.*

Flat image plot. Toolbox functions such as highlighting / hiding samples will not work (see the docs).




---

### Overrepresented sequences Help

The total amount of overrepresented sequences found in each library.

FastQC calculates and lists overrepresented sequences in FastQ files. It would not be possible to show this for all samples in a MultiQC report, so instead this plot shows the *number of sequences* categorized as over represented.

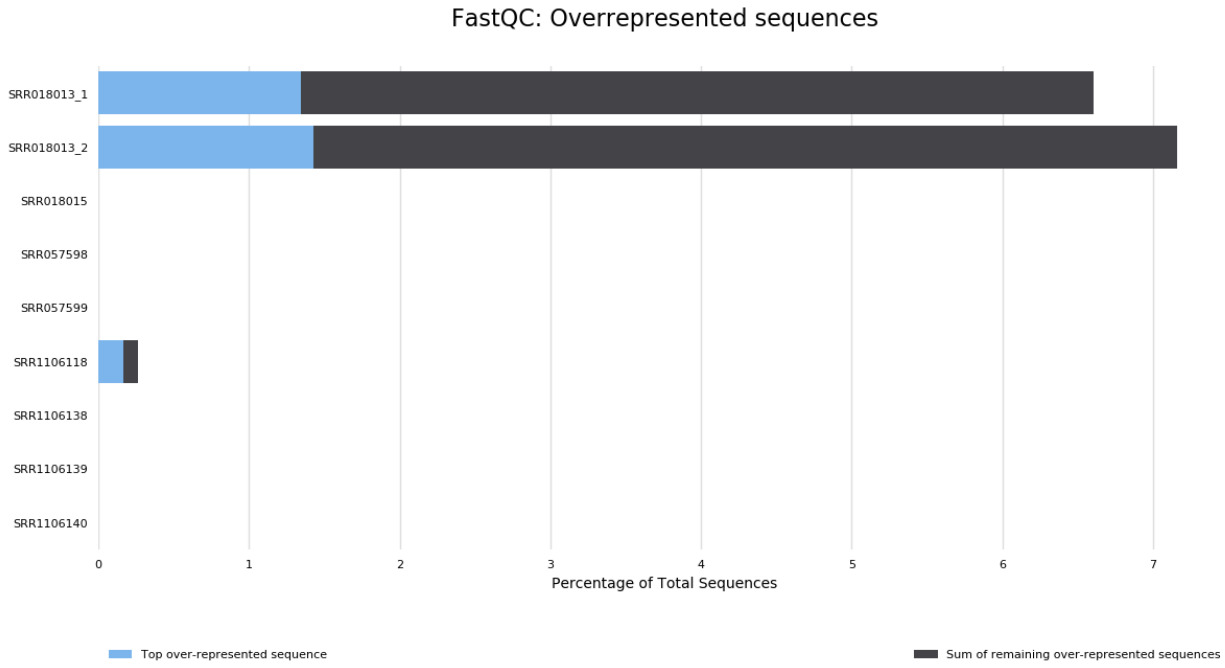
Sometimes, a single sequence may account for a large number of reads in a dataset. To show this, the bars are split into two: the first shows the overrepresented reads that come from the single most common sequence. The second shows the total count from all remaining overrepresented sequences.

From the FastQC Help:

*A normal high-throughput library will contain a diverse set of sequences, with no individual sequence making up a tiny fraction of the whole. Finding that a single sequence is very overrepresented in the set either means that it is highly biologically significant, or indicates that the library is contaminated, or not as diverse as you expected.*

*FastQC lists all of the sequences which make up more than 0.1% of the total. To conserve memory only sequences which appear in the first 100,000 sequences are tracked to the end of the file. It is therefore possible that a sequence which is overrepresented but doesn't appear at the start of the file for some reason could be missed by this module.*

Flat image plot. Toolbox functions such as highlighting / hiding samples will not work (see the docs).




---

**Adapter Content Help**

The cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position.

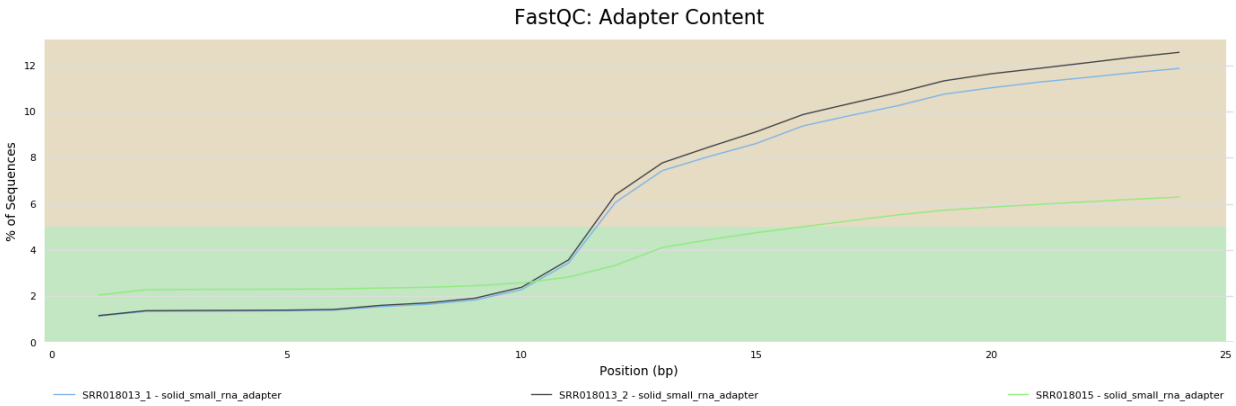
Note that only samples with 0.1% adapter contamination are shown.

There may be several lines per sample, as one is shown for each adapter detected in the file.

From the FastQC Help:

*The plot shows a cumulative percentage count of the proportion of your library which has seen each of the adapter sequences at each position. Once a sequence has been seen in a read it is counted as being present right through to the end of the read so the percentages you see will only increase as the read length goes on.*

Flat image plot. Toolbox functions such as highlighting / hiding samples will not work (see the docs).




---

**Status Checks Help**

Status for each FastQC section showing whether results seem entirely normal (green), slightly abnormal (orange) or very unusual (red).

FastQC assigns a status for each section of the report. These give a quick evaluation of whether the results of the analysis seem entirely normal (green), slightly abnormal (orange) or very unusual (red).

It is important to stress that although the analysis results appear to give a pass/fail result, these evaluations must be taken in the context of what you expect from your library. A 'normal' sample as far as FastQC is concerned is random and diverse. Some experiments may be expected to produce libraries which are biased in particular ways. You should treat the summary evaluations therefore as pointers to where you should concentrate your attention and understand why your library may not look random and diverse.

Specific guidance on how to interpret the output of each module can be found in the relevant report section, or in the FastQC help.

In this heatmap, we summarise all of these into a single heatmap for a quick overview. Note that not all FastQC sections have plots in MultiQC reports, but all status checks are shown in this heatmap.

Sort by highlight

loading..

SciLifeLab

**MultiQC v1.9** - Written by Phil Ewels, available on GitHub.

This report uses HighCharts, jQuery, jQuery UI, Bootstrap, FileSaver.js and clipboard.js.