

PREPROCESSING RIBO-SEQ AND RNA-SEQ SAMPLES in RIBOGALAXY

This is the first part of the tutorial. Here the user can learn how to preprocess raw Ribo-Seq and matching RNA-Seq reads using the RiboGalaxy instance available at <https://ribogalaxy.genomicsdatascience.ie/>. At the end of this tutorial the user will end up with FASTQ files with clean reads that do not contain adapters, untemplated additions, barcodes, UMIs and rRNA/tRNA contamination. Those FASTQ files will be ready for the part 2 and 3 of the tutorial where the user can map reads to genome and transcriptome and analyse and visualise the alignments in GWIPs-viz and Trips-viz browsers.

First, if you want to keep your progress and have a bigger data storage quota, you need to create an account on RiboGalaxy. If you're working on this tutorial for the EMBO course, then try to make sure that you'll have it completed before the first computational class so that you can prepare questions. Also, when attendees of the course run tutorials at different times, it helps to decrease the load on the server and waiting times for the pipelines to be completed.

In this tutorial we assume that you've had some previous basic experience working with any Galaxy instance, e.g. you know how to run tools and how to look at resulting files. If you don't, the interface is pretty intuitive and can be learnt on the fly.

The next 2 parts of the tutorial also contains steps that link processed ribo-seq data with GWIPS-viz (<https://gwips.ucc.ie/>) and Trips-viz (<http://trips.ucc.ie/>) browsers for further analysis and visualisation, however it does not contain the full tutorial for those tools (if you're attending the EMBO course, there will be dedicated sessions explaining how to use them). If you're interested in exploring your data in detail using their full functionality you can take a look at corresponding publications and a walkthrough example for Trips-viz

https://trips.ucc.ie/help/?parent_acc=parent_walkthrough. If you are familiar with the UCSC Genome browser, GWIPS-viz would be very easy to use.

Most tools have a wide range of parameters that can be set.

Text descriptions and screenshots will show you a minimal set of parameters you need to define. Parameters that are not mentioned in text or on screenshots can be skipped. However, If you know what you're doing or want to experiment, feel free to adjust parameters to your liking.

However, we advise you not to output and store alignments to rRNA or tRNAs when you remove ncRNAs using bowtie. This typically leads to overload of the server's storage and may make it crash since those files weigh several Tb. By

default it is set to not output them, so don't worry about that.

Input Ribo-Seq Reads

We created the input file called: '*RIBO_human.fq*'. It is available in the same archive where tutorials are stored or alternatively you can download it from here:

https://www.dropbox.com/s/n71o3u75f7oxik5/RIBO_human.fq?dl=0

It's ribosome profiling of human U2OS cells and it's a subsample of reads that were specifically chosen to map to a selected set of 6 transcripts including ENST00000559916 (B2M), ENST00000371222 (JUN), ENST00000621592 (MYC), ENST00000373316 (PGK1), ENST00000674681 (ACTB), ENST00000396861 (GAPDH).

It was demultiplexed (file contains reads from 1 sample only), however barcodes are still there. This file is not gzipped, however, the gzipped FASTQ files are also accepted in RiboGalaxy.

Now let's move to the read composition.

Here is the structure of a read:

UUU - RPF sequence - **NNNN** - **BBBBB** - AGATCGGAAGAGCACACGTCTGAA

UUU = untemplated additions

RPF sequence = ribosome protected fragment sequence

NNNN = UMIs or Unique Molecular Identifier

BBBBB = barcodes

AGATCGGAAGAGCACACGTCTGAA = 3' adapter

Depending on a protocol, the structure of a read may vary and you need to adjust your parameters accordingly. This tutorial deals with exactly this structure.

Retrieval of the Ribosome Protected Fragment sequence from the raw read

Just to give you a quick overview about the key steps and tools that we will need to use for obtaining clean Ribosome Protected Fragments from raw reads:

- After raw data is uploaded, we would want to look at the quality of the reads using the pretty standard tool called **FastQC**.
- After that we will trim adapters, untemplated additions and barcodes using **Cutadapt**.
- Next, we will move UMIs from the read sequence to the name (header) of the read using **UMIs to Header** tool.
- Then we will deal with typical major contaminants of ribosome profiling - tRNA and rRNAs by using **Bowtie ncRNA Removal**.
- At the end we can again check the quality of the resulting data using **FastQC**.

Let's begin!

1. Upload sample to the RiboGalaxy ('Upload data' at the left panel), choose Type 'fastqsanger' and Genome - 'hg38'. Click 'Start'.

Download from web or upload from disk

Regular Composite Collection Rule-based

You added 1 file(s) to the queue. Add more files or click 'Start' to proceed.

Name	Size	Type	Genome	Settings	Status
RIBO_human.fq	35.4 MB	fastqsa...	Human Dec. 20...		0%

fastqsanger hg38

Type (set all): Auto-detect Genome (set all): unspecified (?)

Choose local files Choose FTP files Paste/Fetch data **Start** Pause Reset Close

You can take a look at reads when you hit an 'eye' button on the right panel with history. Reads are in .fastq format which has 4 lines for each read: read header starting with @; read sequence; symbol '+'; sequencing quality values. During each step of trimming, it is important to track changes and make sure that adapters, untemplated additions and other unrelated to RPF sequences are gone.

1: RIBO_human.fq

35.4 MB

format: fastqsanger, database: hg38

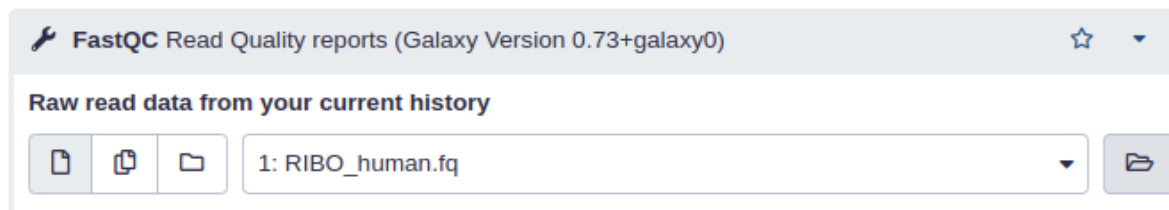
uploaded fastqsanger file

📄 🔗 ⓘ 📊 ? 🗑️ 💬

```
@A01174:295:H7CWGDSX3:4:1648:32081:27962
GGGGGAGCACCCAGTGCTGCTGACCGAGGCCCGCGGACTA
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
@A01174:295:H7CWGDSX3:4:1401:15329:24377
```

```
@A01174:295:H7CWGDSX3:4:1648:32081:2796
2 1:N:0:ATTCAGAA+AGGCTATA
GGGGGAGCACCCAGTGCTGCTGACCGAGGCCCGCGGAC
TAGAAGATCGGAAGAGCACACGTCTGAACTCCAGTCACA
TTCAGAAATCTCGTATGCCGTCTTCTGCTTGAAAAAGGG
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFF:FF:FF::FFFFFFFF:F,:FFFFFF,:FF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
@A01174:295:H7CWGDSX3:4:1401:15329:2437
7 1:N:0:ATTCAGAA+AGGCTATA
GCCCAGATTGTGTGGAATGGTCCTGTGGGGGTCGTCTAG
AAGATCGGAAGAGCACACGTCTGAACTCCAGTCACATTC
AGAAATCTCGTATTCGTCTTCTTATTGCCAAAAAGGGG
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFF:FFF,,F::FFF:F,,FFF:F,F,,::,,FF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

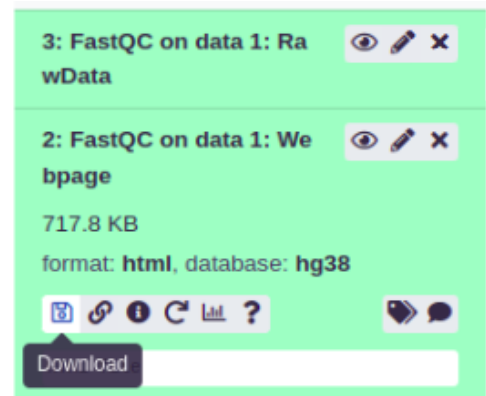
2. Run quality control using the **Fastqc** tool from the **Preprocessing** section. Click **Execute**.



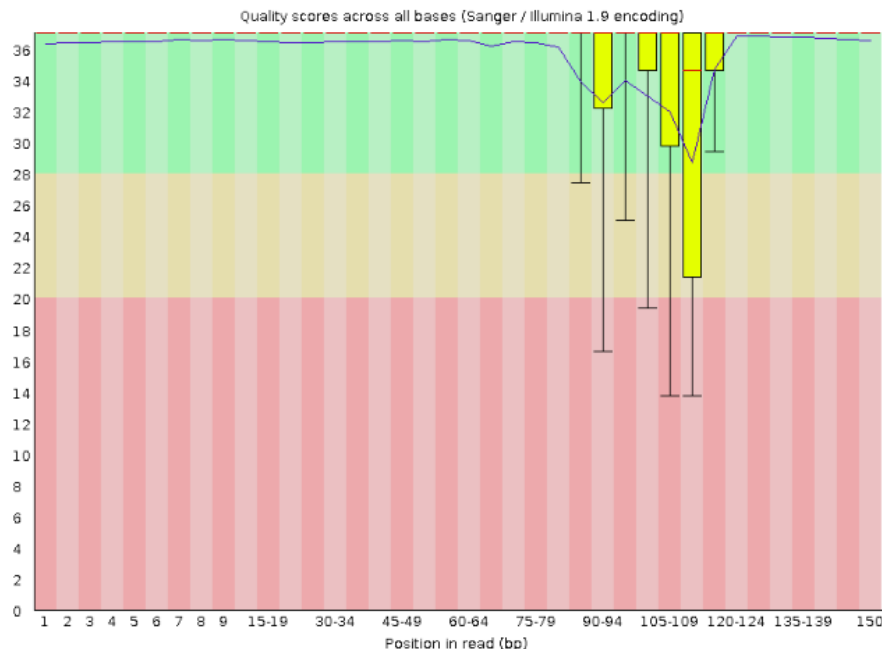
It outputs Webpage and RawData on your right panel. You can download the Webpage by clicking on the 'save' icon. It will download the zip archive containing the file RIBO_human_fq_fastqc.html which you can open in a browser by clicking on it.

You can explore different quality control plots, especially ones that are **failed** or have **warnings**. More information on each type of diagnostic plot can be found in

<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.



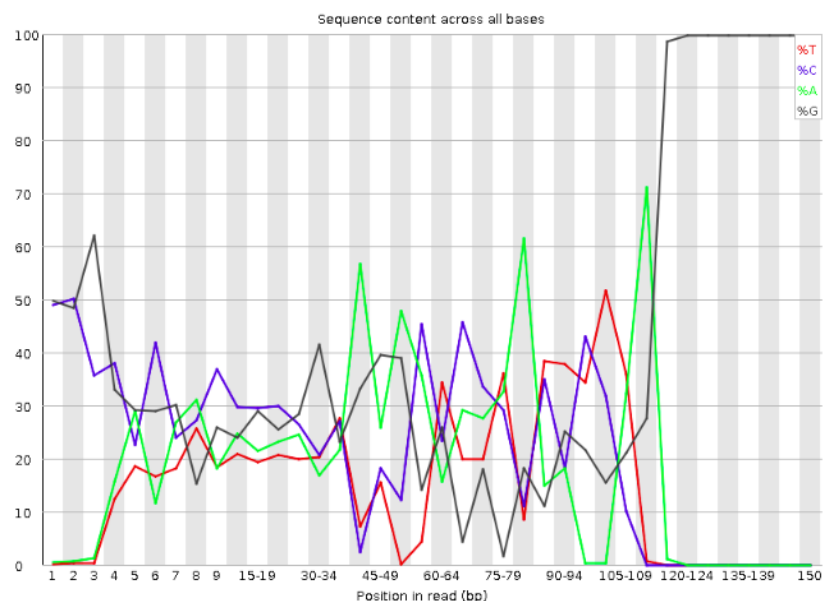
Let's have a look at the **passed** QC plot: **Per base sequence quality**.



X-axis shows position in read (bp), y-axis shows distribution of sequencing quality score per base for all reads which is split by 3 areas: green is high quality, yellow is acceptable, red is poor quality. Here the entire read has either high or acceptable quality. If the entire read sequence has poor quality, you may consider filtering reads based on quality or even choose another dataset to work with.

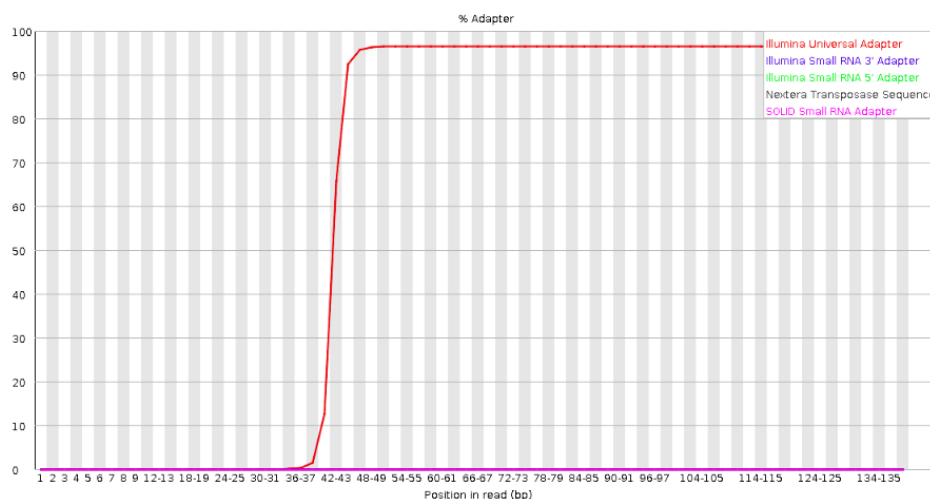
Here we have not yet removed adapters, barcodes, untemplated additions and UMIs, so after all the steps read will be much shorter (~28-30nt). Let's look at the **failed** one: **Per base sequence content**. Y-axis is the fraction of nucleotide per position in read, x-axis is position in read (bp). We have not yet removed untemplated additions (first 3 nucleotides), adapter (last dozens of nucleotides), 5nt of barcode preceding the adapter and 5nt of UMI preceding the barcode. Indeed, RPF starting from position 4 till approximately position 30-34 has relatively uniform distribution of nucleotides as expected (all nucleotides have similar fractions). Adapters, barcodes, UMIs and untemplated additions show clearly huge variation in fractions of nucleotides. We will look at this plot once we remove all of that.

✖ Per base sequence content

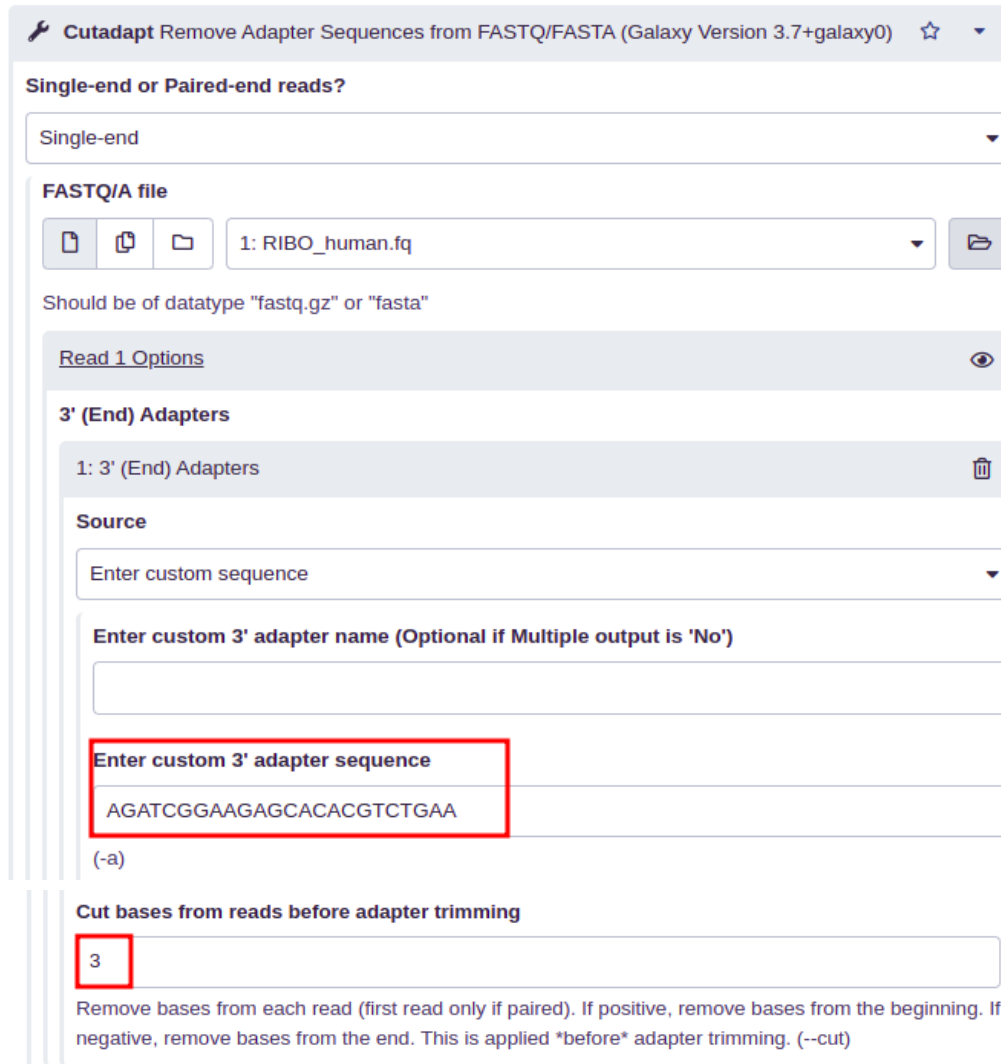


We do also see untrimmed adapters in **Adapter Content** plot (also **failed** one, y-axis is the cumulative fraction of reads where the adapter sequence is identified at the indicated base position and x-axis is position in read in bp):

✖ Adapter Content



3. Trim untemplated additions (first 3nt - **UUU**) and 3' adapter with **Cutadapt** from **Processing**. Choose the input file that you uploaded (RIBO_human.fq). We will trim the **3' (End) Adapter**. Choose **Source**: Enter Custom Sequence. **Enter a custom 3' adapter sequence**: **AGATCGGAAGAGCACACGTCTGAA**. If you'd like, you can also add the name of the adapter, but it's not necessary since we have only 1 adapter. **Cut bases from reads before adapter trimming**: choose 3 (those untemplated additions in the beginning of the read).



Cutadapt Remove Adapter Sequences from FASTQ/FASTA (Galaxy Version 3.7+galaxy0)

Single-end or Paired-end reads?
Single-end

FASTQ/A file
1: RIBO_human.fq

Should be of datatype "fastq.gz" or "fasta"

[Read 1 Options](#)

3' (End) Adapters

1: 3' (End) Adapters

Source
Enter custom sequence

Enter custom 3' adapter name (Optional if Multiple output is 'No')

Enter custom 3' adapter sequence
AGATCGGAAGAGCACACGTCTGAA

(-a)

Cut bases from reads before adapter trimming
3

Remove bases from each read (first read only if paired). If positive, remove bases from the beginning. If negative, remove bases from the end. This is applied *before* adapter trimming. (--cut)

Also you can choose additional files in the **Outputs selector**, e.g. here we will have **Report** for adapter statistics (e.g. how many reads contain adapters).



Outputs selector

Select/Unselect all

☒ Report: Cutadapt's per-adapter statistics. You can use this file with MultiQC.

Click **Execute**. After this step read looks like: **RPF sequence - NNNNN - BBBBB**.

We can take a look at the Report file (hit an 'eye' icon). Almost all reads contain adapters as expected.

```

=== Summary ===

Total reads processed:          101,698
Reads with adapters:          100,032 (98.4%)

== Read fate breakdown ==
Reads that were too short:      37 (0.0%)
Reads written (passing filters): 101,661 (100.0%)

Total basepairs processed:      15,057,845 bp
Total written (filtered):       3,894,151 bp (25.9%)

```

4. Remove barcodes with the **Cutadapt** tool from the **Preprocessing** section. Input - FASTQ file from the previous step. Here you need to define only one parameter: **Cut bases from reads before adapter trimming**. Choose -5 (trim 5 nt from the end of the read). Click **Execute**. Read will look like: **RPF sequence - NNNNN**.

Cutadapt Remove Adapter Sequences from FASTQ/FASTA (Galaxy Version 3.7+galaxy0)

Single-end or Paired-end reads?
Single-end

FASTQ/A file
4: Cutadapt on data 1: Read 1 Output

Cut bases from reads before adapter trimming
-5

Remove bases from each read (first read only if paired). If positive, remove bases from the beginning. If negative, remove bases from the end. This is applied **before** adapter trimming. (--cut)

5. Move UMIs to the header of the read. Use tool **UMI to Header** from **UMI and barcodes**. Input is FASTQ from the previous step. Choose **Number of UMI bases at the 5' end** to 0 and **Number of UMI bases at the 3' end** to 5. Click **Execute**. After this step, the read will contain only RPF sequence.

UMIs to Header (Galaxy Version 0.1.12)

fastqsanger,fastqsanger.gz
6: Cutadapt on data 4: Read 1 Output

Gzip the outputted FASTQ
☒ Yes

Number of UMI bases at the 5' end
0
2 for McGlinch Ingolia Protocol

Number of UMI bases at the 3' end
5
5 for McGlinch Ingolia Protocol

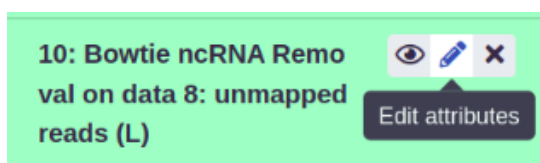
Notice that 5nt UMIs are now in the read header ('_GTGGA')

```
@A01174:295:H7CWGDSX3:4:1648:32081:27962_GCGGA 1:N:0:ATTCAGAA+AGGCTATA
GGAGCACCCAGTGCTGCTGACCGAGGCC
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
@A01174:295:H7CWGDSX3:4:1401:15329:24377_GTCGT 1:N:0:ATTCAGAA+AGGCTATA
CAGATTGTGTGGAATGGTCCTGTGGG
+
```

6. Remove rRNA from the reads using the Bowtie **ncRNA Removal tool** from the **Preprocessing** section. Since it is human reads, we can choose built-in index *Homo sapiens rRNA*. Input file is fastq file from the previous step. Click **Execute**.

Let's have a look at the report file (named 'mapping stats', hit an 'eye' icon): most reads do not align to rRNA (99.96%). We know that this sample is very clean thus we don't expect to see a lot of rRNA reads. However, usually we expect to see ~80% or even more of reads that do map to rRNA in typical ribo-seq samples.

```
# reads processed: 101151
# reads with at least one reported alignment: 615 (0.61%)
# reads that failed to align: 100536 (99.39%)
Reported 8566 alignments to 1 output stream(s)
```



Sometimes galaxy bowtie does not correctly output the datatype of unmapped reads that we are going to use for the next step. We can do the following trick. Choose **'Edit attributes'** on the right panel and in the middle panel choose the **'Datatypes'**

tab and then **'Auto-detect'**. It should change the datatype to the right one, **'fastqsanger'** (or you can type it manually).

7. Remove tRNA from the reads using **Bowtie ncRNA Removal tool** from **Preprocessing**. Since its human reads, we can choose built-in index *Homo sapiens tRNA*. Input file is fastq file from the previous step with unmapped reads. Click **Execute**.

Bowtie ncRNA Removal Remove rRNA and tRNA using Bowtie (Galaxy Version 1.9.0) ☆ ▼

Will you select a reference genome from your history or use a built-in index?

Use a built-in index ▼

Built-ins were indexed using default options

Select a reference index

Homo sapiens tRNA ▼

if your index of interest is not listed - contact Galaxy team

Is this library mate-paired?

Single-end ▼

FASTQ file

10: Bowtie ncRNA Removal on data 8: unmapped reads (L) ▼

8. Finally, we can repeat QC using the **FastQC** tool in **Preprocessing**. Input file is trimmed reads (no additions, adapters, UMIs and barcodes) filtered out from rRNA/tRNA contamination taken from step 7. Click **Execute**.

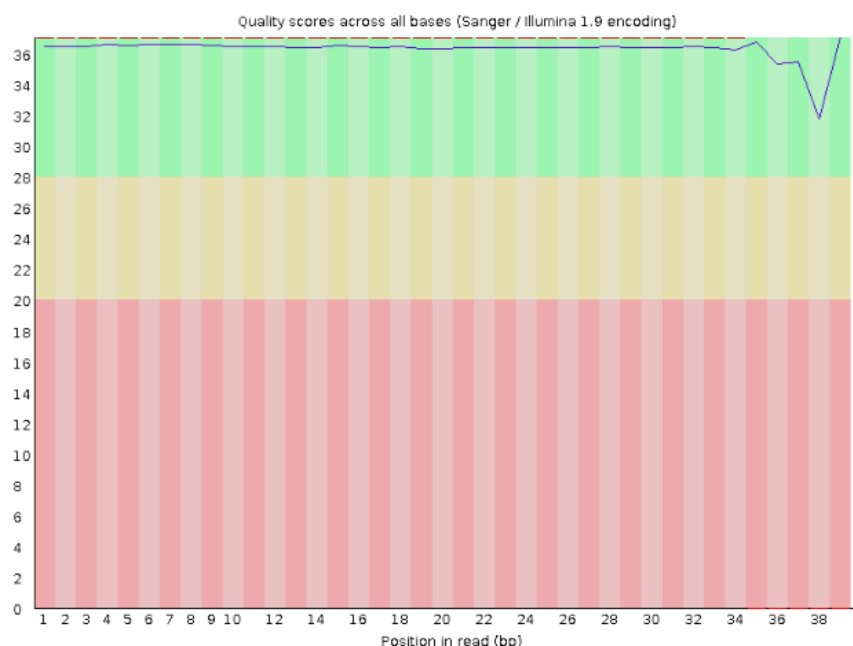
FastQC Read Quality reports (Galaxy Version 0.73+galaxy0) ☆ ▼

Raw read data from your current history

12: Bowtie ncRNA Removal on data 10: unmapped reads (L) ▼

Let's examine the report .html file. **Per base sequence quality** is perfect:

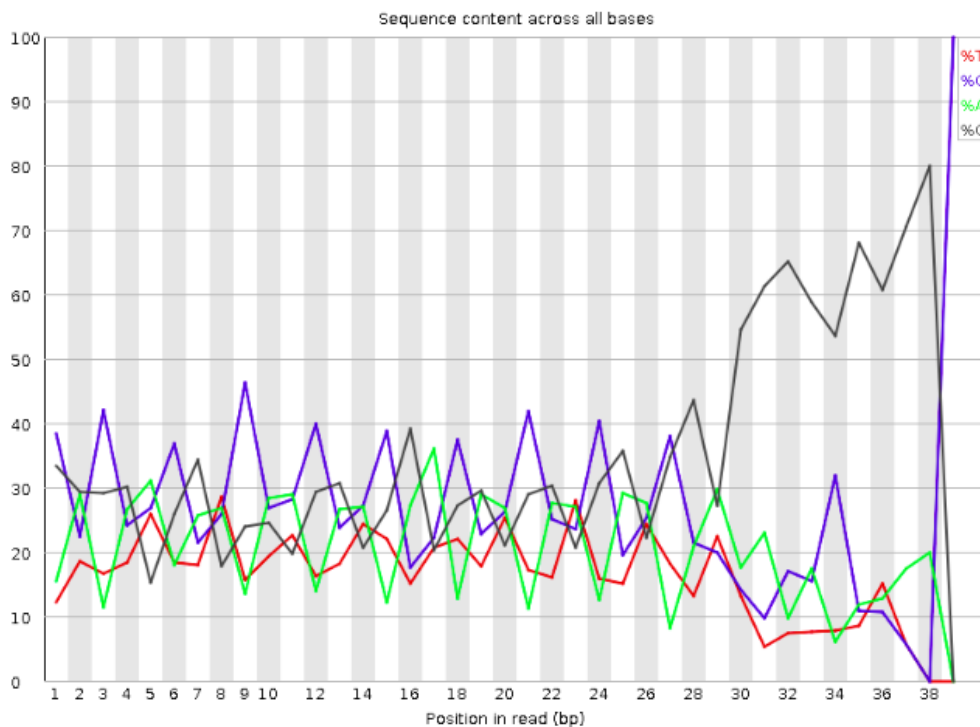
✅ **Per base sequence quality**



Per base sequence content looks much better then for the raw reads, however

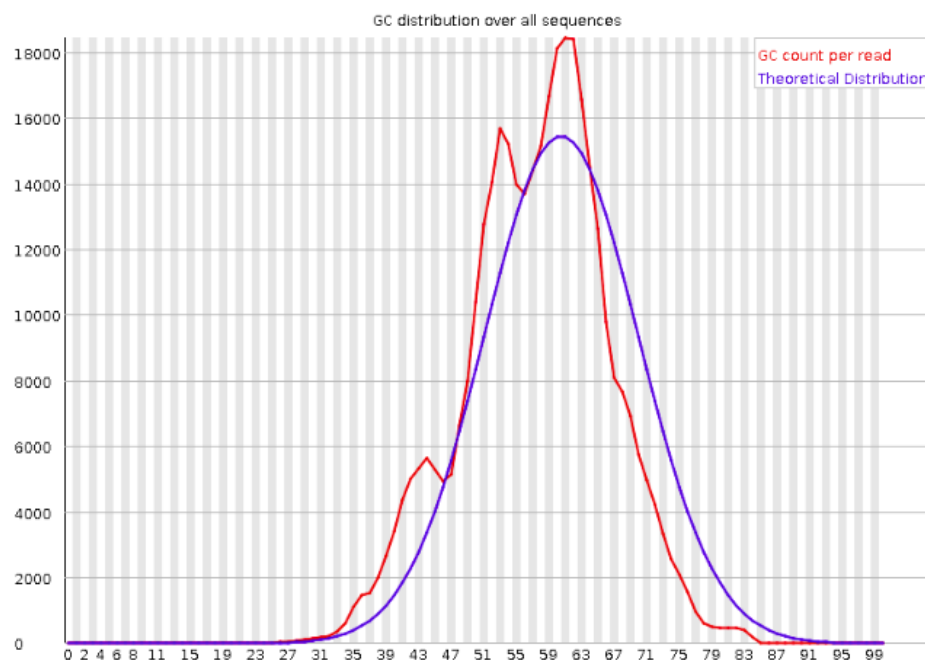
3'end of reads is still wobbly. We are not going to trim it since for deriving A-sites of ribosomes we use 5'end of reads.

✖ Per base sequence content



Per sequence GC content. It shows GC content across the whole length of each sequence in a file and compares it to a modelled normal distribution of GC content. In our case, GC content is significantly different from expected one which can be explained by non-random sampling of reads (reads that map to only 6 transcripts).

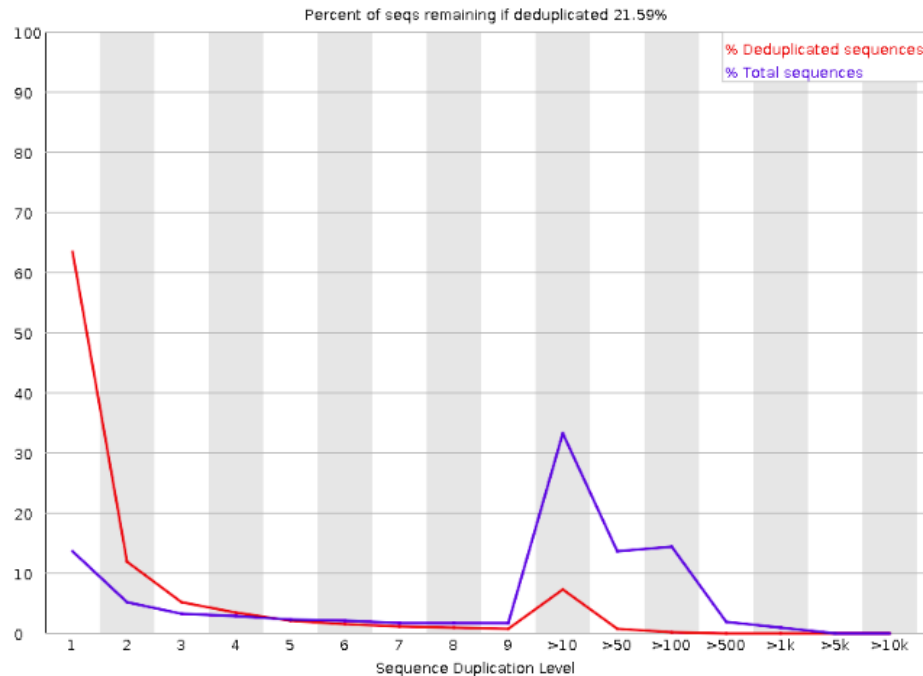
⚠ Per sequence GC content



Sequence Duplication Levels. X-axis is how often a sequence occurs (1 = one

time, 2 = twice, >10 = more than 10 times etc), y-axis is the fraction of such sequences (reads). Typically in RNA-seq and Ribo-seq experiments it's expected to see many duplicated reads. Also we are going to use UMI deduplication at a later stage for getting rid of PCR duplicates.

✖ Sequence Duplication Levels



It's also good to take a look at **Overrepresented sequences** and blast top hits. This way we can spot unexpected contaminations. We had about 100,000 reads and the top overrepresented sequence is about 1%.

✖ Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CAGCTCACCATGGATGATGATATCGCC	1114	1.114122553480883	No Hit
CGCTCAGACACCATGGGGAAGGTGAAGG	847	0.8470931802498275	No Hit
CGCCAGCTCACCATGGATGATGATATCG	648	0.6480712878416626	No Hit
CGCCAGCTCACCATGGATGATGATATC	604	0.604066447309204	No Hit
CGCTCAGACACCATGGGGAAGGTGAAG	413	0.4130454349978498	No Hit
GGGAAACTGTGGCGTGATGGCCGCGGG	407	0.40704477492524177	No Hit
GGGAAACTGTGGCGTGATGGCCGCGGG	387	0.3870425746832151	No Hit
CACACCTTCTACAATGAGCTGCGTGTGG	366	0.3660402644290872	No Hit
GCTCAGACACCATGGGGAAGGTGAAGG	350	0.3500385042354659	No Hit

When we blast the top1 sequence, we can see that it's part of actin beta which is indeed among transcripts we sampled originally.

✓ PREDICTED: Pan paniscus actin_cytoplasmic 1 (LOC100990579). mRNA	Pan paniscus	54.0	54.0	100%	3e-04	100.00%	1840	XM_008963291.2
✓ PREDICTED: Alluopoda melanoleuca actin_cytoplasmic 1 (LOC117800408). mRNA	Alluopoda mel...	54.0	54.0	100%	3e-04	100.00%	1629	XM_034652944.1
✓ PREDICTED: Trachypithecus francoisi actin_cytoplasmic 1-like (LOC117077202). misc_RNA	Trachypithecus...	54.0	54.0	100%	3e-04	100.00%	1804	XR_004434698.1
✓ PREDICTED: Trachypithecus francoisi actin_alpha skeletal muscle (LOC117094377). mRNA	Trachypithecus...	54.0	54.0	100%	3e-04	100.00%	1779	XM_033229843.1
✓ PREDICTED: Trachypithecus francoisi actin-like (LOC117094375). mRNA	Trachypithecus...	54.0	54.0	100%	3e-04	100.00%	1953	XM_033229841.1
✓ PREDICTED: Trachypithecus francoisi actin_cytoplasmic 1-like (LOC117092896). misc_RNA	Trachypithecus...	54.0	54.0	100%	3e-04	100.00%	1293	XR_004440955.1
✓ PREDICTED: Trachypithecus francoisi actin_cytoplasmic 1-like (LOC117085681). misc_RNA	Trachypithecus...	54.0	54.0	100%	3e-04	100.00%	1745	XR_004438050.1
✓ PREDICTED: Chelonoidis abingdonii actin_cytoplasmic 1-like (LOC116820837). partial mRNA	Chelonoidis abi...	54.0	54.0	100%	3e-04	100.00%	1144	XM_032773537.1
✓ PREDICTED: Hylobates moloch actin beta (ACTB). mRNA	Hylobates moloch	54.0	54.0	100%	3e-04	100.00%	1916	XM_032755826.1

Input RNA-Seq Reads

We prepared the artificial training sample with human RNA-seq reads from the same cell line U2OS called: '**RNA_human.fastq**'. It is also provided in the same archive together with tutorials or can be downloaded from

https://www.dropbox.com/s/wqxyzc3v0z99226n/RNA_human.fastq?dl=0.

Matching RNA-seq samples are typically obtained alongside with Ribo-Seq samples for different purposes including Differential Translation Efficiency analysis or building a cell-specific transcriptome.

This sample contains reads that map on a selected subset of 6 transcripts (the same as Ribo-Seq sample).

Unlike Ribo-Seq reads, RNA-seq reads have only 3'end adapters (AGATCGGAAGAGCACACGTCTGAA) and no untemplated additions, UMIs, adapters and barcodes, so their preprocessing take less number of steps.

Retrieval of the read sequence sequence from the raw read

Similarly, here is a quick overview of key preprocessing steps.

- After raw data is uploaded, we will look at the quality of the reads using the pretty standard tool called **FastQC**.
- After that we will trim adapters using **Cutadapt**.
- Then we will deal with typical major contaminants of ribosome profiling - tRNA and rRNAs by using **Bowtie ncRNA Removal**.
- At the end we can again check the quality of the resulting data using **FastQC**.

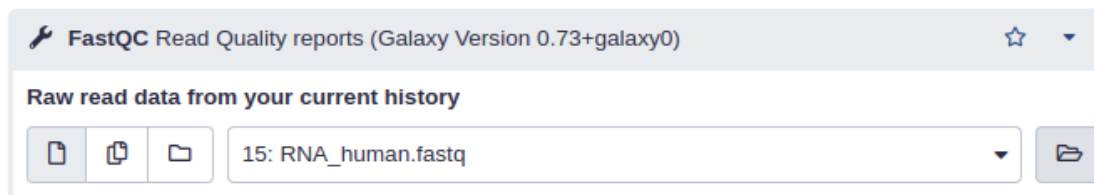
1. Upload sample to the RiboGalaxy ('**Upload data**' at the left panel), choose Type 'fastqsanger' and Genome - 'hg38'. Click '**Start**'.

Download from web or upload from disk

Regular
Composite
Collection
Rule-based

Name	Size	Type	Genome	Settings	Status
RNA_human.fastq	1018.9 KB	fastqsa...	Human Dec. 20...		100%

2. Run quality control using the **Fastqc** tool from the **Preprocessing** section. Push **Execute**.

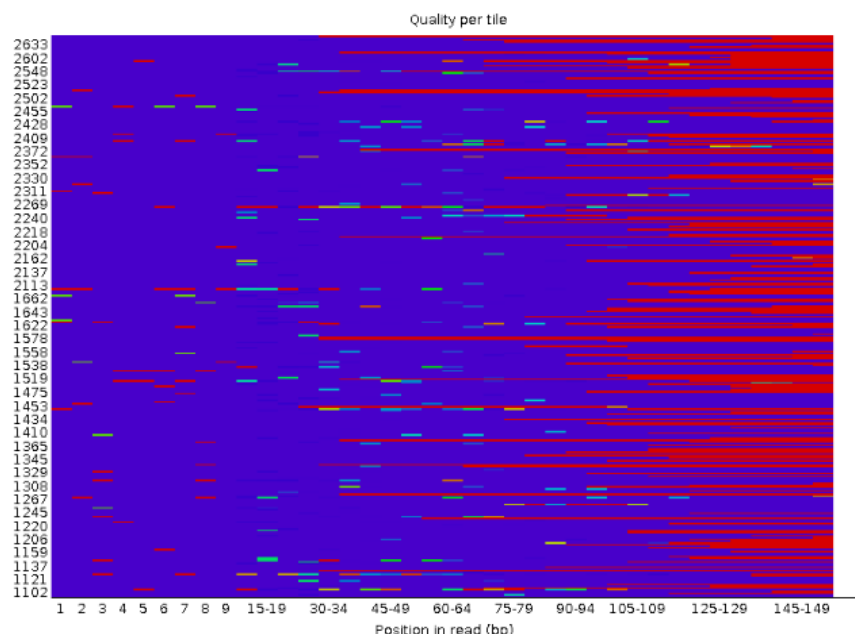


Download the .zip file and click on the .html report.

Per base sequence quality is a **pass**. **Per sequence GC content** is **failed** which is expected from the fact that we have only 6 transcripts in our samples. **Sequence duplication level** and **Overrepresented sequences** are also a **fail** which is expected from RNA-seq. Moreover, we specifically chose housekeeping genes that are highly transcribed in this particular cell line for creating the sample therefore we expect to see overrepresented sequences. By blasting the top-1 sequence (CAGGATTATAAACTGGAACGGTGAAGGTGACAGCAGTCGGTTGGAGCGA) we can see that it corresponds to actin beta.

Per tile sequence quality is another diagnostic plot that **failed**. This type of plot shows the Illumina sequencing quality scores from each tile across all of your bases. It is used to see if there was a loss in quality associated with only one part of the flowcell. X-axis is 'position in read (bp)', the y-axis is tiles. Hotter colours (red) depict tiles that had worse qualities than other tiles for that base. We can see here that quality worsens while progressing to the end of the read where adapters are located (and they are going to be trimmed).

✖ Per tile sequence quality



3. Trim 3' end adapter with **Cutadapt** from **Processing**. Choose the input file that you uploaded (RNA_human.fq). We will trim the 3' (**End**) **Adapter**. Choose **Source**:

Enter Custom Sequence. **Enter a custom 3' adapter sequence:**
AGATCGGAAGAGCACACGTCTGAA. If you'd like, you can also add the name of the adapter, but it's not necessary since we have only 1 adapter.

The screenshot shows the Cutadapt web interface. At the top, it says "Cutadapt Remove Adapter Sequences from FASTQ/FASTA (Galaxy Version 3.7+galaxy0)". Below this, there's a dropdown menu for "Single-end or Paired-end reads?" set to "Single-end". Under "FASTQ/A file", there's a file selection button and a text input field containing "15: RNA_human.fastq". Below this, it says "Should be of datatype 'fastq.gz' or 'fasta'". There's a section for "Read 1 Options" with an eye icon. Under "3' (End) Adapters", there's a list item "1: 3' (End) Adapters" with a trash icon. Below this, there's a "Source" dropdown menu set to "Enter custom sequence". There's a text input field for "Enter custom 3' adapter name (Optional if Multiple output is 'No')". Below that, there's a text input field for "Enter custom 3' adapter sequence" containing "AGATCGGAAGAGCACACGTCTGAA". At the bottom, there's a label "(-a)".

We can also ask for a report. Click **Execute**.

The screenshot shows the "Outputs selector" section. It has a "Select/Unselect all" button. Below it, there's a checkbox labeled "Report" which is checked. To the right of the checkbox, it says "Cutadapt's per-adapter statistics. You can use this file with MultiQC."

By hitting an 'eye' icon on the report file, we can see that only a small fraction of reads actually have 3'end adapters. As a simple explanation, in short-read Illumina sequencing only quite short original molecules will get 3' adaptors sequenced. Since Ribo-seq footprints are naturally quite short, we do see most of them have adapters while for RNA-seq it is often not the case.

```

=== Summary ===

Total reads processed:          3,521
Reads with adapters:           209 (5.9%)

== Read fate breakdown ==
Reads that were too short:      6 (0.2%)
Reads written (passing filters): 3,515 (99.8%)

```

4. Remove rRNA from the reads using the Bowtie **ncRNA Removal tool** from the **Preprocessing** section. Since its human reads, we can choose built-in index *Homo sapiens rRNA*. Input file is fastq file from the previous step. Click **Execute**.

We can take a look at report file as well:

```

# reads processed: 3515
# reads with at least one reported alignment: 0 (0.00%)
# reads that failed to align: 3515 (100.00%)
No alignments

```

We subsampled the reads from the sample where all ncRNAs were removed, therefore we don't see any alignments to rRNA. We are not going to perform the step with removing tRNA.

5. Finally, we can repeat QC using the **FastQC** tool in **Preprocessing**. Input file is trimmed reads (no adapters and ncRNAs) taken from the previous step. Click **Execute**.

Let's quickly examine the fastqc report. We still see all the same categories that failed since there was not so much done in terms of trimming and filtering (only 6% of reads had adapters and there was no rRNA originally).