uORF-Tools Documentation

Release 1.0.0

Rick Gelhausen

Sep 10, 2018

Documentation

CHAPTER 1

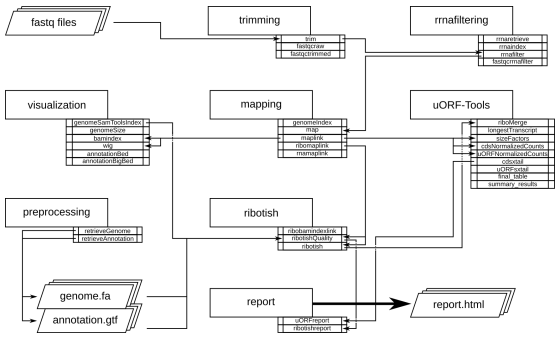
uORF-Tools

1.1 Introduction

uORF-Tools are a workflow and a collection of tools for the analysis of **Upstream Open Reading Frames** (short uORFs). The workflow is based on the workflow management system snakemake and handles installation of all dependencies via bioconda *[GruningDSjodin+17]*, as well as all processings steps. The source code of uORF-Tools is open source and available under the License. Installation and basic usage is described below.

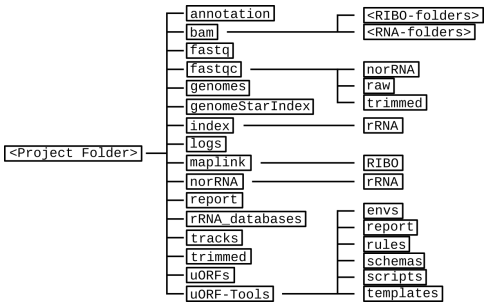
1.2 Program flowchart

The following flowchart describes the processing steps of the workflow and how they are connected. there is a variant of the workflow accepting a preprocessed uORF-annotation file, to skip the time consuming ribotish step for reruns of the workflow.



1.3 Directory table

The output is written to a directory structure that corresponds to the workflow steps, you can decide at the bedginning of the workflow whether you want to keep the intermediary files (default) or only the final result.



- **annotation:** contains the processed user-provided annotation file with genomic features.
  Contents: *annotation.gtf*
- **bam:** contains a subfolder for each input *.fastq* file. These subfolders contain the *.bam* files created using STAR.
  Contents: *Aligned.sortedByCoord.out.bam*, *Log.final.out*, *Log.out*, *Log.progress.out*, *SJ.out.tab*
- **fastq:** contains the user-provided *.fastq* files used as input.
  Contents: *<method-condition-replicate>.fastq*
- **fastqc:** contains the result files of a quality control for the input *.fastq* files, using different processing methods:
  - **norRNA:** quality control after removal of the rRNA.
  - **raw:** quality control for the unprocessed data.
  - **trimmed:** quality control after removal of the adapter sequences.

  Contents: *<method-condition-replicate>_fastqc.zip*, *<method-condition-replicate>-<subfolderName>.html*
- **genomes:** contains the genome file, as well as an according index and sizes file.
  Contents: *genome.fa*, *genome.fa.fai*, *sizes.genome*
- **genomeStarIndex:** contains the files created by STAR during the creation of a genome index.
  Contents: *chrLength.txt*, *chrNameLength.txt*, *chrStart.txt*, *chrName.txt*, *exonGeTrInfo.tab*, *exonInfo.tab*, *genInfo.tab*, *Genome*, *genomeParameters.txt*, *SA*, *SAindex*, *sjdbList.fromGTF.out.tab*, *transcriptInfo.tab*, *sjdbInfo.txt*, *sjdbList.out.tab*
- **index**: contains an index for the rRNA databases created using *indexdb_rna*.
  Contents: *<database-ID>.bursttrie_0.dat*, *<database-ID$>.kmer_0.dat*, *<database-ID>.pos_0.dat*, *<database-ID>.stats*
- **logs:** contains log files for each step of the workflow.
  Contents: *<rule>.o<jobID>*, *<methods>.log*
- **maplink:** contains soft links to the *.bam* files and an according index.
  - **RIBO:** contains soft links to the *.bam* and *.bam.bai* files for RIBO and corresponding parameter files (*.para.py*).

  Contents: *<method-condition-replicate>.bam.bai*, *RIBO/<condition-replicate>.bam.para.py*
- **norRNA:** contains processed *.fastq* files, where the rRNA has been removed.
  - **rRNA:** contains the *reject.fastq* which specifies the rRNA reads that are removed.

Contents: *<method-condition-replicate>.fastq*, *rRNA/reject.fastq*

- **report:** contains *.jpg* plots for the *report.html*.
  Contents: *<condtion-replicate>-qual.jpg*, *xtail_cds_fc.jpg*, *xtail_cds_r.jpg*, *xtail_uORFs_fc.jpg*, *xtail_uORFs_r.jpg*
- **rRNA_databases:** contains the known annotated rRNA sequences for filtering.
  Contents: *<database-ID>.fasta*
- **tracks:** contains *BED (.bed)*, *wig (.wig)* and *bigWig (.bw)* files for visualizing tracks in a genome browser.
  Contents: *annotation.bb*, *annotation.bed*, *annotation.bed6*, *annotationNScore.bed6*, *annotation-woGenes.gtf*, *<method-condition-replicate>.bw*, *<method-condition-replicate>.wig*
- **trimmed:** contains processed *.fastq* files, where the adapter sequences have been trimmed.
  Contents: *<method-condition-replicate>.fastq*
- **uORFs:** contains the main output of the workflow.
  - **uORFs_regulation.tsv:** table summarizing the predicted uORFs with their regulation on the main ORF.
  - **merged_uORFs.bed:** genome browser track with predicted uORFs.
  - **processing_summary.tsv:** table indicating the lost reads per processing step.

  Contents: *longest_protein_coding_transcripts.gtf*, *merged_uORFs.bed*, *merged_uORFs.csv*, *norm_CDS_reads.csv*, *norm_uORFs_reads.csv*, *sfactors_lprot.csv*, *processing_summary.tsv*, *uORF_regulation.tsv*, *xtail_cds.csv*, *xtail_cds_fc.pdf*, *xtail_cds_r.pdf*, *xtail_uORFs.csv*, *xtail_uORFs_fc.pdf*, *xtail_uORFs_r.pdf*

- **uORF-Tools:** contains the workflow tools.
  - **envs:** conda environment files (.yaml).
  - **report:** restructuredText files for the report (.rst).
  - **rules:** the snakemake rules.
  - **schemas:** validation templates for input files
  - **scripts:** scripts used by the snakemake workflow.
  - **templates:** templates for the *config.yaml* and the *samples.tsv*.

1.4 Installation

We recommend to install *uORF-Tools* with all dependencies via conda. Once you have conda installed simply type:

```
conda create -c bioconda -c conda-forge -n uORF-Tools snakemake
source activate uORF-Tools
```

1.5 Usage

Using the workflow requires the *uORF-Tools*, a genome sequence (.fasta), an annotation file (.gtf) and the sequencing results files (.fastq). We recommend retrieving both the genome and the annotation files for mouse and human from GENCODE *[HFG+12]* and for other species from Ensembl Genomes *[ZAA+18]*. The usage of the workflow is first described in general, while a detailed example applied to an example dataset is described here: *example-workflow*.

1.6 Retrieve uORF-Tools

The first step is downloading the latest version of *uORF-Tools* from Github. Open your terminal and create a new directory for your workflow and change into it.

```
mkdir uORFflow; cd uORFflow;
```

Note: All following commands assume that you are located in the workflow folder

Now download and unpack the latest version of the *uORF-Tools* by entering the following commands:

```
wget https://github.com/anibunny12/uORF-Tools/archive/1.0.1.tar.gz
tar -xzf 1.0.1.tar.gz; mv uORF-Tools-1.0.1 uORF-Tools; rm 1.0.1.tar.gz;
```

The *uORF-Tools* are now located in a subdirectory of your workflow.

1.7 Prepare input files

If the genome and the annotation file are compressed, extract them using *gunzip* or any other decompression tool.

```
gunzip <genomeFile>.fa.gz
gunzip <annotationFile>.gtf.gz
```

Copy or move the genome and the annotation file into the workflow folder and name them *genome.fa* and *annotation.gtf*.

```
mv <genomeFile>.fa genome.fa
mv <annotationFile>.gtf annotation.gtf
```

Create a folder *fastq/* and move or copy all of your compressed fastq files into the folder. .. note:: Ensure that you compress the fastq files. The workflow expects compressed files and it saves a lot of disk space.

```
mkdir fastq
mv *.fastq.gz fastq/
```

Now copy the templates of the sample sheet and the configuration file into the *uORF-Tools* folder.

```
cp uORF-Tools/templates/samples.tsv uORF-Tools/
cp uORF-Tools/templates/config.yaml uORF-Tools/
```

Next, customize the *config.yaml*. It contains the following variables:

- **taxonomy** Specify the taxonomic group of the used organism in order to ensure the correct removal of reads mapping to ribosomal genes (Eukarya, Bacteria, Archea).

- **adapter** Specify the adapter sequence to be used. If not set, *Trim galore* will try to determine it automatically.
- **samples** The location of the samples sheet created in the previous step.
- **genomeindexpath** If the STAR genome index was already precomputed, you can specify the path to the files here, in order to avoid recomputation.
- **uorfannotationpath** If the uORF-file was already precomputed, you can specify the path to the files here, in order to avoid recomputation.

Now edit the sample sheet corresponding to your project. It contains the following variables:

- **method** Indicates the method used for this project. RIBO for ribosome profiling or RNA for RNA-seq.
- **condition** Indicates the applied condition (A, B / CTRL, TREAT). Please ensure that you put the control before the treatment alphabetically (e.g. A: Control B: Treatment or CTRL: Control, TREAT: Treatment)
- **replicate** ID used to distinguish between the different replicates (e.g. 1,2, . . .)
- **fastqFile** Indicates the according fastq file for a given sample.

As seen in the *samples.tsv* template:

method	condition	replicate	fastqFile
RIBO	A	1	fastq/FP-ctrl-1-2.fastq.gz
RIBO	B	1	fastq/FP-treat-1-2.fastq.gz
RNA	A	1	fastq/Total-ctrl-1-2.fastq.gz
RNA	B	1	fastq/Total-treat-1-2.fastq.gz

1.8 Executing the workflow

The workflow will first retrieve all required programs and install them. Then it will derive the necessary computation step depending on your input files. You will receive continuous updates about the progress of the workflow execution. Log files of the individual steps will be written to the logs subdirectory and are named according to the workflow step. The intermediary output of the different workflow steps are written to directories as shown in the directory table.

1.8.1 Run the workflow locally

Use the following steps when you plan to execute the workflow on a single server or workstation. Please be aware that some steps of the workflow require a lot of memory, specifically for eukaryotic species.

```
snakemake --use-conda -s uORF-Tools/Snakefile --configfile uORF-Tools/config.yaml --
↪directory ${PWD} -j 20 --latency-wait 60
```

1.8.2 Run Snakemake in a cluster environment

Use the following steps if you are executing the workflow via a queuing system. Edit the configuration file cluster.yaml according to your queuing system setup and cluster hardware. The following system call shows the usage with Grid Engine:

```
snakemake --use-conda -s uORF-Tools/Snakefile --configfile uORF-Tools/config.yaml --
↪directory ${PWD} -j 20 --cluster-config uORF-Tools/cluster.yaml
```

1.8.3 Report

Using any of the presented methods, this will run the workflow on our dataset and create the desired output files. Once the workflow has finished, we can request an automatically generated *report.html* file using the following command:

```
snakemake --report report.html
```

1.9 References

CHAPTER 2

Example workflow

The retrieval of input files and running the workflow locally and on a server cluster via a queuing system is demonstrated using an example with data available from SRA via NCBI. The dataset is available under the GEO accession number *GSE66929*. The retrieval of the data is described in this tutorial.

Note: Ensure that you are in the uORF-Tools conda environment as explained in the installation section.

2.1 Setup

First of all, we start by creating the project directory and changing to it.

```
mkdir tutorial; cd tutorial;
```

We then download the lastest version of the *uORF-Tools* into the newly created project folder and unpack it.

```
wget https://github.com/anibunny12/uORF-Tools/archive/1.0.1.tar.gz
tar -xzf 1.0.1.tar.gz; mv uORF-Tools-1.0.1 uORF-Tools; rm 1.0.1.tar.gz;
```

2.2 Retrieve and prepare input files

Before starting the workflow, we have to acquire and prepare several input files. These files are the annotation file, the genome file, the fastq files, the configuration file and the sample sheet.

2.2.1 Annotation and genome files

First, we want to retrieve the annotation file and the genome file. In this case we can find both on the GENCODE *[HFG+12]* webpage for the human genome.

GTF / GFF3 files

Content	Regions	Description	Download
Comprehensive gene annotation	CHR	• It contains the comprehensive gene annotation on the reference chromosomes only • This is the **main annotation file** for most users	GTF GFF3
Comprehensive gene annotation	ALL	• It contains the comprehensive gene annotation on the reference chromosomes, scaffolds, assembly patches and alternate loci (haplotypes) • This is a **superset** of the main annotation file	GTF GFF3

Fasta files

Content	Regions	Description	Download
Transcript sequences	CHR	• Nucleotide sequences of all transcripts on the reference chromosomes	Fasta
Protein-coding transcript sequences	CHR	• Nucleotide sequences of coding transcripts on the reference chromosomes • Transcript biotypes: protein_coding, nonsense_mediated_decay, non_stop_decay, IG_*_gene, TR_*_gene, polymorphic_pseudogene	Fasta
Protein-coding transcript translation sequences	CHR	• Amino acid sequences of coding transcript translations on the reference chromosomes • Transcript biotypes: protein_coding, nonsense_mediated_decay, non_stop_decay, IG_*_gene, TR_*_gene, polymorphic_pseudogene	Fasta
Long non-coding RNA transcript sequences	CHR	• Nucleotide sequences of long non-coding RNA transcripts on the reference chromosomes	Fasta
Genome sequence (GRCh38.p12)	ALL	• Nucleotide sequence of the GRCh38.p12 genome assembly version on all regions, including reference chromosomes, scaffolds, assembly patches and haplotypes • The sequence region names are the same as in the GTF/GFF3 files	Fasta

On this page, we can directly retrieve both files by clicking on the according download links next to the file descriptions. Alternatively, you can directly download them using the following commands:

```
wget ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_28/gencode.v28.
↪annotation.gtf.gz
wget ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_28/GRCh38.p12.
↪genome.fa.gz
```

Then, we are going to unpack both files.

```
gunzip gencode.v28.annotation.gtf.gz
gunzip GRCh38.p12.genome.fa.gz
```

Finally, we will rename these files to *annotation.gtf* and *genome.fa*.

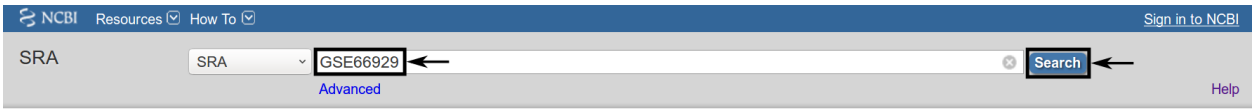
```
mv gencode.v28.annotation.gtf annotation.gtf
mv GRCh38.p12.genome.fa genome.fa
```

Another webpage that provides these files is Ensembl Genomes *[ZAA+18]*. This usually requires searching their file system in order to find the wanted files. For this tutorial, we recommend to stick to GenCode instead.

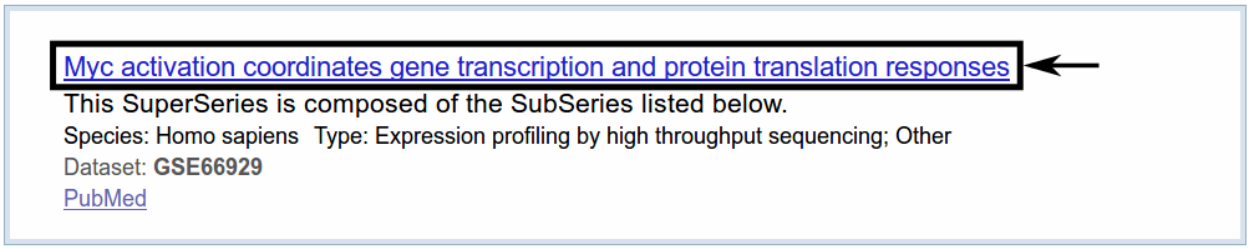
2.2.2 Fastq files

Next, we want to acquire the fastq files. For many datasets, the easiest way to retrieve the fastq files is using the European Nucleotide Archive (ENA) *[SAA+17]* as it provides direct download links when searching for a dataset. Unfortunately, the *GSE66929* dataset is not provided by ENA.

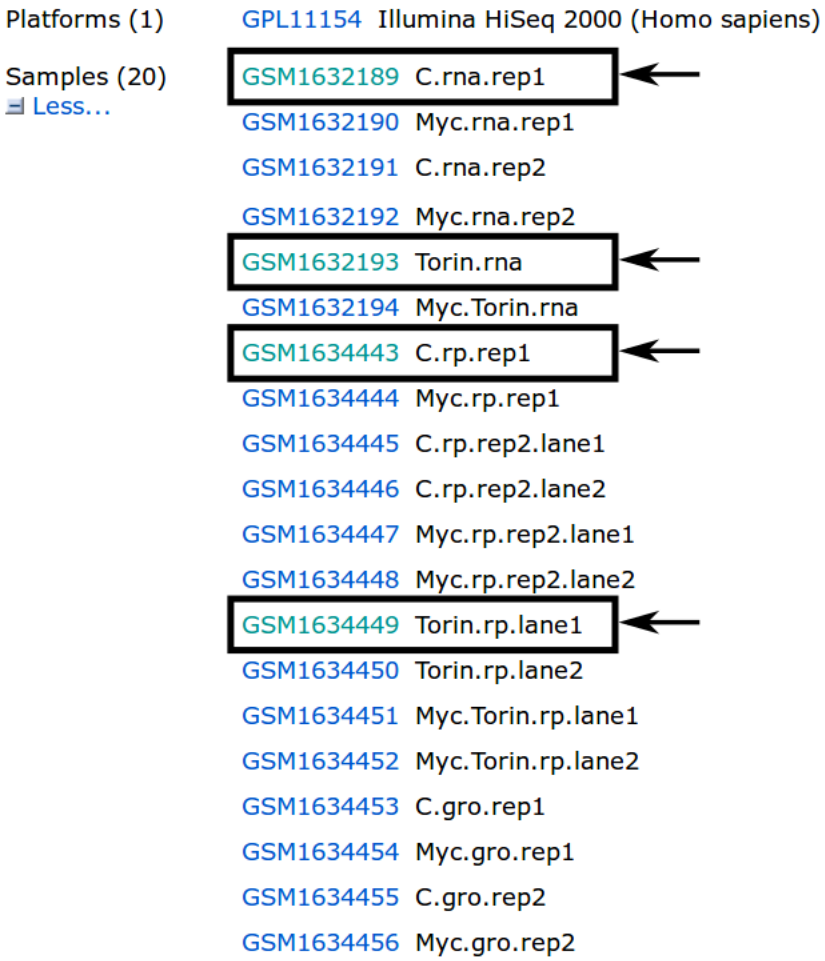
Therefore, we will use the Sequence Read Archive (SRA) *[LSS11]* instead, which is hosted by NCBI. On the NCBI webpage, we search for the GEO accession number, here *GSE66929*.



We receive one search result.

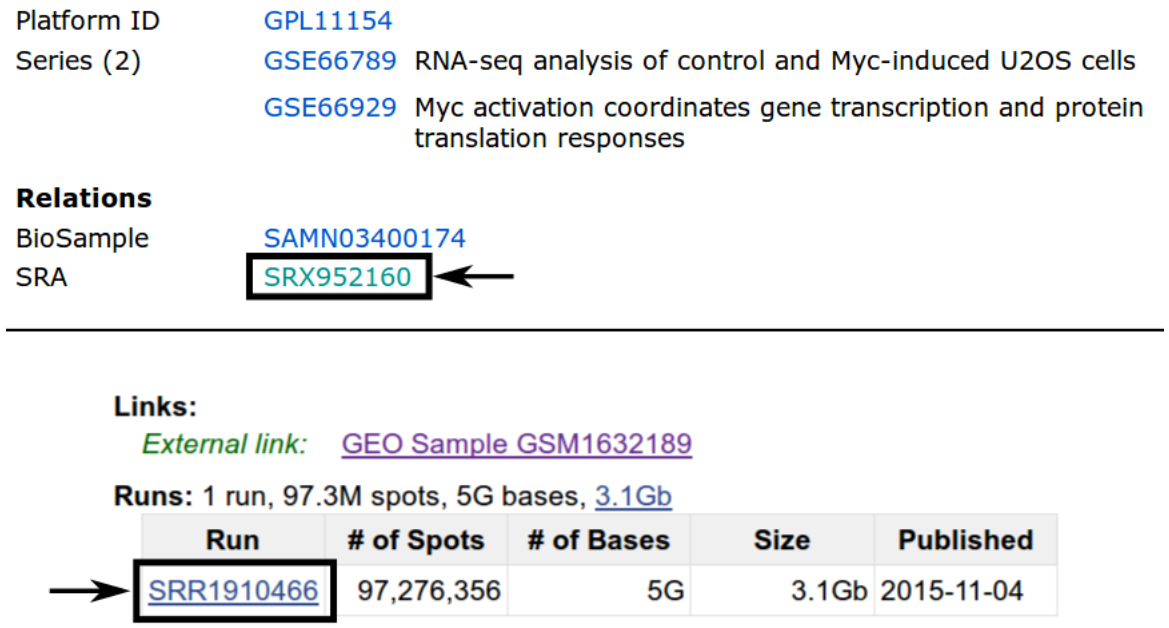


When following the link provided in the search results, we get an overview with all kinds of information about the dataset. We are interested mainly in the samples provided on this page. In this tutorial, we are interested in the highlighted samples.



There are many ways to download fastq files with SRA. For more information about downloading please have a look at the following guide: Downloading SRA data using command line utilities.

The simplest way is most likely the usage of the SRA Toolkit, as it allows direct conversion into *.fastq* files. The figure below shows how to find the *SRR ID* for the example of *C.rna.rep1*. By following the *GSM ID* link (Figure above) and then the *SRX ID* link, the *SRR ID* can be retrieved.



Using the *SRA Toolkit* and the *SRR IDs* for our 4 samples we can use the *fasterq-dump* executable to download the according *.fastq* files.

If you already have an installation of the *SRA Toolkit*, you can use the following commands.

```
./<sraToolkitPath>/bin/fasterq-dump SRR1910466
gzip SRR1910466.fastq

./<sraToolkitPath>/bin/fasterq-dump SRR1916542
gzip SRR1916542.fastq

./<sraToolkitPath>/bin/fasterq-dump SRR1910470
gzip SRR1910470.fastq

./<sraToolkitPath>/bin/fasterq-dump SRR1916548
gzip SRR1916548.fastq
```

If you do not have the *SRA Toolkit*, we suggest using the conda environment:

```
conda install sra-tools -c bioconda -c conda-forge
```

This will install the sra-tools, with all required dependencies, to the current conda environment (uORF-Tools). Then you can use the following commands to generate the required *fastq* files.

```
fasterq-dump SRR1910466; gzip SRR1910466.fastq;
fasterq-dump SRR1916542; gzip SRR1916542.fastq;
fasterq-dump SRR1910470; gzip SRR1910470.fastq;
fasterq-dump SRR1916548; gzip SRR1916548.fastq;
```

Note: Ensure that you compress the fastq files. The workflow expects compressed fastq files and it saves a lot of disk space.

Warning: Be advised that the fastq generation step can take several hours depending on the size of the fastq files and your internet connection.

Now, we create a fastq folder and move all the *.fastq.gz* files into this folder.

```
mkdir fastq; mv *.fastq.gz fastq/;
```

2.2.3 Configuration file and sample sheet

Finally, we will prepare the configuration file (*config.yaml*) and the sample sheet (*samples.tsv*). We start by copying templates for both files from the *uORF-Tools/templates/* into the *uORF-Tools/* folder.

```
cp uORF-Tools/templates/* uORF-Tools/
```

Using any text editor (vim, nano, gedit, atom, . . .), we will first edit the *samples.tsv*.

```
vim uORF-Tools/samples.tsv
```

The template looks as follows:

method	condition	replicate	fastqFile
RIBO	A	1	fastq/FP-ctrl-1-2.fastq.gz
RIBO	B	1	fastq/FP-treat-1-2.fastq.gz
RNA	A	1	fastq/Total-ctrl-1-2.fastq.gz
RNA	B	1	fastq/Total-treat-1-2.fastq.gz

- **method** Indicates the method used for this project. RIBO for ribosome profiling or RNA for RNA-seq.
- **condition** Indicates the applied condition (A, B / CTRL, TREAT). Please ensure that you put the control before the treatment alphabetically (e.g. A: Control B: Treatment or CTRL: Control, TREAT: Treatment)
- **replicate** ID used to distinguish between the different replicates (e.g. 1,2, . . .)
- **fastqFile** Indicates the according fastq file for a given sample.

For this tutorial, the resulting *samples.tsv* will look as follows:

method	condition	replicate	fastqFile
RIBO	A	1	fastq/SRR1916542.fastq.gz
RIBO	B	1	fastq/SRR1916548.fastq.gz
RNA	A	1	fastq/SRR1910466.fastq.gz
RNA	B	1	fastq/SRR1910470.fastq.gz

Warning: Please ensure that you do not replace any tabulator symbols with spaces while changing this file.

Next, we are going to set up the *config.yaml*.

```
vim uORF-Tools/config.yaml
```

This file contains the following variables:

- **taxonomy** Specify the taxonomic group of the used organism in order to ensure the correct removal of reads mapping to ribosomal genes (Eukarya, Bacteria, Archea).
- **adapter** Specify the adapter sequence to be used. If not set, *Trim galore* will try to determine it automatically.
- **samples** The location of the samples sheet created in the previous step.
- **genomeindexpath** If the STAR genome index was already precomputed, you can specify the path to the files here, in order to avoid recomputation.
- **uorfannotationpath** If the uORF-file was already precomputed, you can specify the path to the files here, in order to avoid recomputation.

```
#Taxonomy of the samples to be processed, possible are Eukarya, Bacteria, Archea
taxonomy: "Eukarya"
#Adapter sequence used
adapter: ""
samples: "uORF-Tools/samples.tsv"
genomeindexpath: ""
uorfannotationpath: ""
```

For this tutorial, we can keep the default values for the *config.yaml*. The organism analyzed in this tutorial is *homo sapiens*, therefore we keep the taxonomy at *Eukarya*. We let *Trim galore* determine the correct adapter sequence. The path to *samples.tsv* is correct and we precomputed nothing, therefore we leave the rest empty.

2.3 Running the workflow

Now that we have all the required files, we can start running the workflow, either locally or in a cluster environment.

2.3.1 Run the workflow locally

Use the following steps when you plan to execute the workflow on a single server or workstation. Please be aware that some steps of the workflow require a lot of memory, specifically for eukaryotic species.

```
snakemake --use-conda -s uORF-Tools/Snakefile --configfile uORF-Tools/config.yaml --
↪directory ${PWD} -j 20 --latency-wait 60
```

2.3.2 Run Snakemake in a cluster environment

Use the following steps if you are executing the workflow via a queuing system. Edit the configuration file *cluster.yaml* according to your queuing system setup and cluster hardware. The following system call shows the usage with Grid Engine:

```
snakemake --use-conda -s uORF-Tools/Snakefile --configfile uORF-Tools/config.yaml --
↪directory ${PWD} -j 20 --cluster-config uORF-Tools/cluster.yaml
```

2.3.3 Example: Run Snakemake in a cluster environment

Warning: Be advised that this is a specific example, the required options may change depending on your system.

We ran the tutorial workflow in a cluster environment, specifically a TORQUE cluster environment. Therefore, we created a bash script *torque.sh* in our project folder.

```
vim torque.sh
```

We proceeded by writing the queueing script:

```
#!/bin/bash
#PBS -N <ProjectFolder>
#PBS -S /bin/bash
#PBS -q "long"
#PBS -d <PATH/ProjectFolder>
#PBS -l nodes=1:ppn=1
#PBS -o <PATH/ProjectFolder>
#PBS -j oe
cd <PATH/ProjectFolder>
source activate snakemake
snakemake --latency-wait 600 --use-conda -s uORF-Tools/Snakefile --configfile uORF-
↪Tools/config.yaml --directory ${PWD} -j 20 --cluster-config uORF-Tools/torque.yaml -
↪-cluster "qsub -N {cluster.jobname} -S /bin/bash -q {cluster.qname} -d <PATH/
↪ProjectFolder> -l {cluster.resources} -o {cluster.logoutputdir} -j oe"
```

We then simply submitted this job to the cluster:

```
qsub torque.sh
```

Using any of the presented methods, this will run the workflow on our dataset and create the desired output files.

2.3.4 Report

Once the workflow has finished, we can request an automatically generated *report.html* file using the following command:

```
snakemake --latency-wait 600 --use-conda -s uORF-Tools/Snakefile --configfile uORF-
↪Tools/config.yaml --report report.html
```

2.4 References

Bibliography

[GruningDSjodin+17] Björn Grüning, Ryan Dale, Andreas Sjödin, Jillian Rowe, Brad A. Chapman, Christopher H. Tomkins-Tinch, Renan Valieris, and Johannes Köster. Bioconda: a sustainable and comprehensive software distribution for the life sciences. *bioRxiv*, 2017. URL: https://www.biorxiv.org/content/early/2017/10/27/207092, arXiv:https://www.biorxiv.org/content/early/2017/10/27/207092.full.pdf, doi:10.1101/207092.

[HFG+12] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, I. Barnes, A. Bignell, V. Boychenko, T. Hunt, M. Kay, G. Mukherjee, J. Rajan, G. Despacio-Reyes, G. Saunders, C. Steward, R. Harte, M. Lin, C. Howald, A. Tanzer, T. Derrien, J. Chrast, N. Walters, S. Balasubramanian, B. Pei, M. Tress, J. M. Rodriguez, I. Ezkurdia, J. van Baren, M. Brent, D. Haussler, M. Kellis, A. Valencia, A. Reymond, M. Gerstein, R. Guigo, and T. J. Hubbard. GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Res.*, 22(9):1760–1774, Sep 2012.

[LSS11] R. Leinonen, H. Sugawara, and M. Shumway. The sequence read archive. *Nucleic Acids Res.*, 39(Database issue):19–21, Jan 2011.

[SAA+17] Nicole Silvester, Blaise Alako, Clara Amid, Ana Cerdeño-Tarrága, Laura Clarke, Iain Cleland, Peter W Harrison, Suran Jayathilaka, Simon Kay, Thomas Keane, and others. The european nucleotide archive in 2017. *Nucleic acids research*, 46(D1):D36–D40, 2017.

[ZAA+18] Daniel R Zerbino, Premanand Achuthan, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Jyothish Bhai, Konstantinos Billis, Carla Cummins, Astrid Gall, Carlos García Girón, Laurent Gil, Leo Gordon, Leanne Haggerty, Erin Haskell, Thibaut Hourlier, Osagie G Izuogu, Sophie H Janacek, Thomas Juettemann, Jimmy Kiang To, Matthew R Laird, Ilias Lavidas, Zhicheng Liu, Jane E Loveland, Thomas Maurel, William McLaren, Benjamin Moore, Jonathan Mudge, Daniel N Murphy, Victoria Newman, Michael Nuhn, Denye Ogeh, Chuang Kee Ong, Anne Parker, Mateus Patricio, Harpreet Singh Riat, Helen Schuilenburg, Dan Sheppard, Helen Sparrow, Kieron Taylor, Anja Thormann, Alessandro Vullo, Brandon Walts, Amonida Zadissa, Adam Frankish, Sarah E Hunt, Myrto Kostadima, Nicholas Langridge, Fergal J Martin, Matthieu Muffato, Emily Perry, Magali Ruffier, Dan M Staines, Stephen J Trevanion, Bronwen L Aken, Fiona Cunningham, Andrew Yates, and Paul Flicek. Ensembl 2018. *Nucleic Acids Research*, 46(D1):D754–D761, 2018. URL: http://dx.doi.org/10.1093/nar/gkx1098, arXiv:/oup/backfile/content_public/journal/nar/46/d1/10.1093_nar_gkx1098/2/gkx1098.pdf, doi:10.1093/nar/gkx1098.

[HFG+12] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, I. Barnes, A. Bignell, V. Boychenko, T. Hunt, M. Kay, G. Mukherjee, J. Rajan, G. Despacio-Reyes, G. Saunders, C. Steward, R. Harte, M. Lin, C. Howald, A. Tanzer, T. Derrien, J. Chrast, N. Walters, S. Balasubramanian, B. Pei, M. Tress, J. M. Rodriguez, I. Ezkurdia, J. van Baren, M. Brent, D. Haussler, M. Kellis, A. Valencia, A. Reymond, M. Gerstein, R. Guigo, and T. J. Hubbard. GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Res.*, 22(9):1760–1774, Sep 2012.

[LSS11] R. Leinonen, H. Sugawara, and M. Shumway. The sequence read archive. *Nucleic Acids Res.*, 39(Database issue):19–21, Jan 2011.

[SAA+17] Nicole Silvester, Blaise Alako, Clara Amid, Ana Cerdeño-Tarrága, Laura Clarke, Iain Cleland, Peter W Harrison, Suran Jayathilaka, Simon Kay, Thomas Keane, and others. The european nucleotide archive in 2017. *Nucleic acids research*, 46(D1):D36–D40, 2017.

[ZAA+18] Daniel R Zerbino, Premanand Achuthan, Wasiu Akanni, M Ridwan Amode, Daniel Barrell, Jyothish Bhai, Konstantinos Billis, Carla Cummins, Astrid Gall, Carlos García Girón, Laurent Gil, Leo Gordon, Leanne Haggerty, Erin Haskell, Thibaut Hourlier, Osagie G Izuogu, Sophie H Janacek, Thomas Juettemann, Jimmy Kiang To, Matthew R Laird, Ilias Lavidas, Zhicheng Liu, Jane E Loveland, Thomas Maurel, William McLaren, Benjamin Moore, Jonathan Mudge, Daniel N Murphy, Victoria Newman, Michael Nuhn, Denye Ogeh, Chuang Kee Ong, Anne Parker, Mateus Patricio, Harpreet Singh Riat, Helen Schuilenburg, Dan Sheppard, Helen Sparrow, Kieron Taylor, Anja Thormann, Alessandro Vullo, Brandon Walts, Amonida Zadissa, Adam Frankish, Sarah E Hunt, Myrto Kostadima, Nicholas Langridge, Fergal J Martin, Matthieu Muffato, Emily Perry, Magali Ruffier, Dan M Staines, Stephen J Trevanion, Bronwen L Aken, Fiona Cunningham, Andrew Yates, and Paul Flicek. Ensembl 2018. *Nucleic Acids Research*, 46(D1):D754–D761, 2018. URL: http://dx.doi.org/10.1093/nar/gkx1098, arXiv:/oup/backfile/content_public/journal/nar/46/d1/10.1093_nar_gkx1098/2/gkx1098.pdf, doi:10.1093/nar/gkx1098.