Title

Assess the quality of different transcriptomic small RNA alignment techniques by analyzing multiple sources of variability using a multi-alignment framework.

Authors
Xinwei Zhao, Eberhard Korsching*

Affiliation
University of Münster, University Hospital of Münster (UKM), Institute of Bioinformatics, CCSR Group, DE 48149 Münster, Germany

Xinwei Zhao zhaoxi@uni-muenster.de ,
ORCID https://orcid.org/0009-0006-5293-486X

Eberhard Korsching *eberhard.korsching@uni-muenster.de* ,
ORCID https://orcid.org/0000-0001-9141-1988

* Corresponding author

Abstract

Background DNA and RNA nucleotide sequences are ubiquitous in all biological cells and serve as a comprehensive library of capabilities for the cells and as an impressive regulatory system to control cellular function. The multi-alignment framework (MAF) presented here offers researchers a user-friendly sequence alignment platform as a functional template that is flexible enough to adjust all steps but also comprehensive enough to join many different tools and custom parameter combinations if in-depth analysis is necessary or advised, like in low read rate situations. The framework enables researchers to apply multiple aligners and further processing tools in flexible combinations to analyze the subtle to pronounced differences between different alignment programs and algorithms applied to the same data basis. This flexibility supports sophisticated research concepts and enables the discovery of robust and relevant biological results.

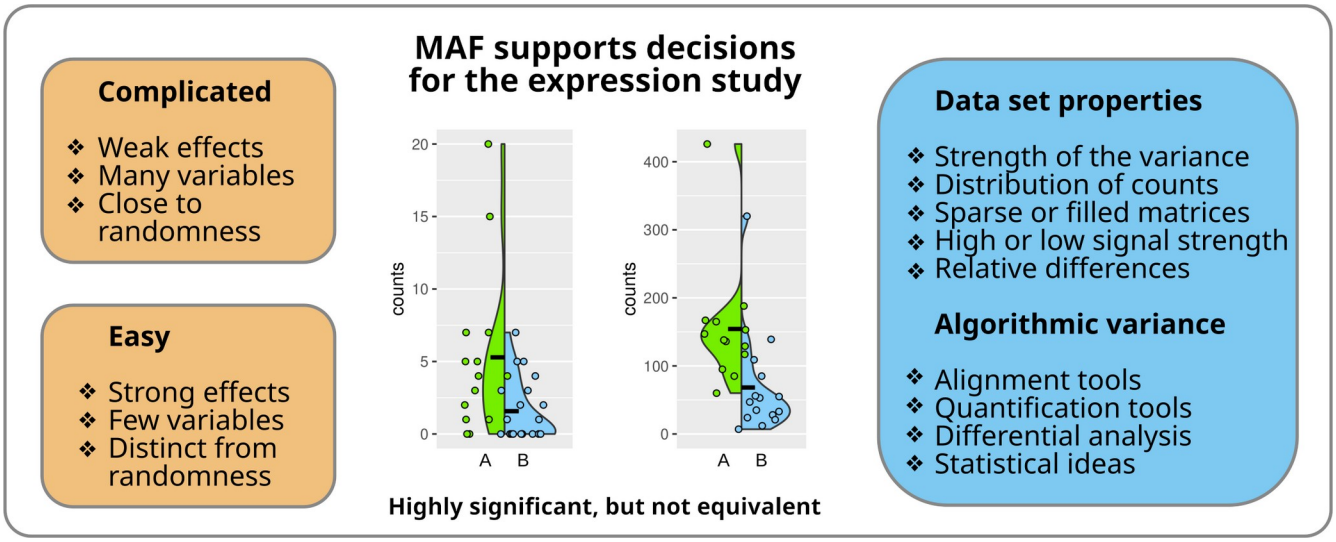
Results MAF is designed for use on the Linux platform, which is widely used in bioinformatics. The script structure allows for the integration of all processing steps, saving time when procedures need to be repeated many times with different parameter combinations or many different data sets. The focus is here on the microRNA analysis but the provided templates allow to generalize the workflow to all transcriptomic and even genomic analyses. The template structure allows to combine pre- and post processing steps in a flexible way and are inherently extendable and completely redesignable to support various sequence analysis scenarios. The exemplary and comprehensive microRNA analysis exemplifies the value of examining the details of alignment and quantification procedures, providing insight into how the quality of the result candidates can be assessed and the false positive rate minimized.

Conclusion This approach is suitable for scientists who really want to delve into the basics of their alignment results and expect flexibility in all aspects of their results. The exemplary microRNA analysis suggests that the STAR and Bowtie2 alignment programs may be more useful than BBMap. Overall, using STAR in combination with the Salmon quantifier, or with some limitations, the Samtools quantification, appears to be the most reliable approach.

Keywords

Sequence alignment, Transcriptome sequences, Genome sequences, Alignment variability, Read quantification, Differential analysis, Quality.

Graphical abstract



Background

The development of sequence aligners began in the 1970s and continues to this day, with algorithms and concepts constantly being adjusted [1] . As early as 1982, this alignment approach was complemented by de novo sequence assembly approaches and programs [2] .

The alignmentprimarily focuses on finding almost perfect correspondence between sequences of interest but also attempts to identify conserved sequence regions between very unrelated biological species and their DNA and RNA sequence material. The currently established developments include widely used algorithms such as the BLAST algorithm (basic local alignment search tool) [3] , BLAT [4, 5] , and the FASTA (sequence comparison) algorithm [6, 7] , which find "local or global similarity" between sequences based on the mentioned criteria. Both tools work with protein or DNA sequences and can translate between these types.

In addition to these core tools, there are many other alignment programs currently available, with a recent review focussing on short read alignment programs summarizing current developments in the fields of sequence alignment tools, sequencing technologies, and their core properties [8] . Musich et al. [9] provide a critical examination of the differences between alignment programs discussing various alignment and assembly algorithms in terms of their strengths and weaknesses, which can lead to slightly to significantly different results, necessitating a more detailed analysis of these situations. Finally, the handling of 'gaps' also influences the quality of alignment results and can vary between alignment programs [10, 11] .

The presented multi-alignment framework (MAF) takes these alignment tool differences into account and tries to leverage the necessary efforts to use several selected algorithms in parallel and on different scenarios repeatably. It is clear that all the possible complex data set conditions might not follow one conceptual scheme and therefore for a specific data situation a new comparison of alignment programs and their results is advised. In addition, programs change over time and there is no guarantee that the program behavior will remain constant. To this end, the MAF framework was created to remain flexible over time, adapt to different application scenarios, and be adjustable to the user's needs.

The MAF is based on the common Linux operating system and uses Bash command line based code to combine the alignment programs and further post- and preprocessing programs, all of which are command line based, into an integrated workflow. The three provided major Bash scripts : 30_se_mrna.sh, 30_pe_mrna.sh and 30_se_mir.sh are the three provided examples for a simple single end mRNA analysis, a simple paired end mRNA analysis and a complex small RNA analysis workflow.

The programs called from within the scripts are already well established in the scientific community. They take the fastq based data, perform quality control, trim adapters, optionally adjust the reads for other sequence features, optionally deduplicate based on read sequence similarities, align to genomic or transcriptomic references, and finally optionally deduplicate BAM file hosted reads by UMI barcode information. All these steps are well described below. The entire system is open source based.

The result folder structure created by the Bash scripts places each major workflow level in a separate folder, chronologically sorted by an increasing number along with a specific folder name.

In this context, the bioinformatics pipeline can be viewed as a ready-to-use application concerning each of the three provided Bash scripts together with the provided human references and the supplied program specific references. The three Bash scripts facilitate the alignment and quality control of microRNA and mRNA data sets of varying sizes and numbers depending on the available computing resources. In house experiments utilized fastq file sizes up to 10 GB and sample numbers up to 200 on a 10 year old server with 24 cores and 256 GB of main memory.

However, the modular structure of the scripts allows for flexibility to adapt or modify Bash code, as the code inherently contains all the necessary functions for writing your own application modules.

There has been some discussion about building a containerized solution like Docker or Singularity, or something similar. However, for maintaining the ability to use newer versions of the program or to switch program versions due to bugs e.g., after operating system updates, the containerized approach is not ideal for a complex framework. Despite potential license issues with the programs, the chance to resolve error scenarios is much higher if the user follows a stepwise installation procedure. In summary, the solution should be accessible to biologists who have some computer science knowledge but are not proven experts in advanced information technology solutions.

The simple Bash script setup, - which assigns a specific task to a specific script, was preferred over a universal Bash script with configuration files and a lot more script code due to the complexity of the latter approach.

The presented approach has been successfully applied in three published studies [12-14] and supports three ongoing studies. To facilitate the learning curve for less experienced scientists, some basic terms are explained in Table 1.

Table 1

Terms and abbreviations placed in context.

MAF	Multi-alignment framework.
DNA	Deoxyribonucleic acid - in a concrete way a sequence of bases coding for the functions of a biological cell, like a program code.
RNA	Ribonucleic acid - building a sequence which is very often a regulatory object and if being mRNA also translated into a protein.
mRNA	The messenger RNA class.
tRNA	Transfer ribonucleic acid - carrying amino acids to the ribosomes.
microRNA	A small RNA class (approximately 21 nucleotides/ bases long).
piRNA	Another small RNA class (approximately 30 bases long).

PCR	Polymerase chain reaction - for amplification of sequence fragments.
read	A sequence fragment resulting from a certain sequencing technology procedure - short reads normally 35 to 300 bases, long reads up to 40 K to 1 M bases.
contig	Contiguous sequences - a result of many overlapping reads forming a large sequence fragment without gaps.
UMI	Unique molecular identifier - a artificial sequence of known length introduced in a sequence read to normalize the PCR amplification bias.
fastq	An absolute basic and common text only file format for sequence raw data - so based on reads created by the sequencer technologies.
BAM	A binary read file format - programs are needed to read it.
SAM	A text based read file format requires much more space than BAM. Often, this file format is created first and then translated into the BAM fomat.
STAR, Bowtie2, BBMap	The three used alignment programs.
Salmon, Samtools	The programs used for RNA read quantification per genomic or transcriptomic sequence feature denote a defined sequence region.
quantification	Means that reads belonging to a certain transcriptomic feature are counted to estimate the expression rate of this feature in comparison to other features in the same sample or across different samples.
editor	A program to view and process text files specifically, either with a graphical interface like 'Gedit' or 'Pluma', or in the terminal environment with a purely text based editor like 'nano'. Note: it is possible to search inside nano. Other search possibilities on the command line include programs like 'grep'. For BAM files 'samtools view' is needed.

Implementation

Overview

The analysis framework MAF (Figure 1) presented here has the purpose of aligning transcriptomic read data (mRNA or microRNA) and quantifying these data in several different ways. It can be seen as a flexible workbench for sequence related tasks.

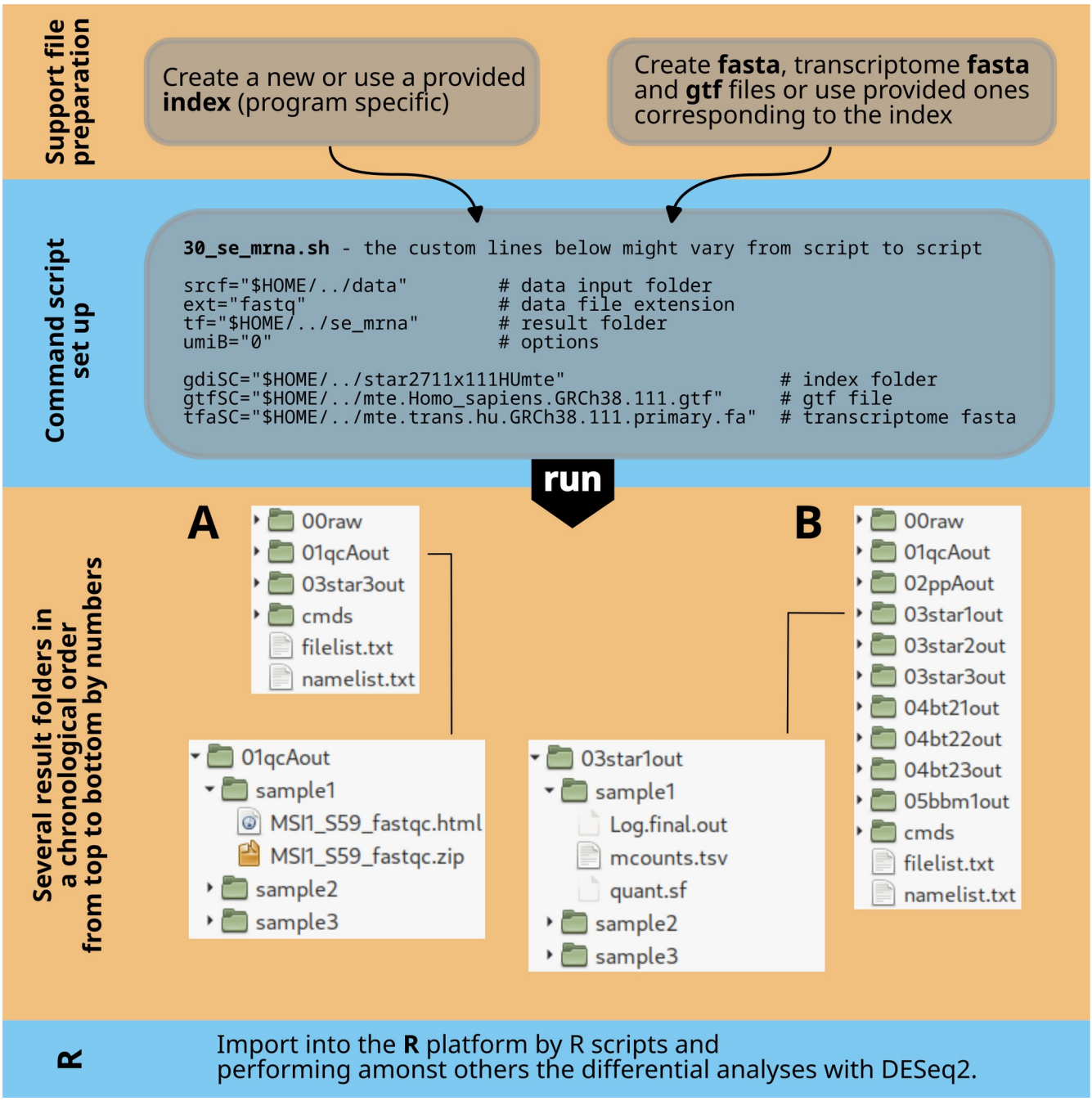


Figure 1. Overall workflow.
The processing architecture is used to estimate the impact of preprocessing tools and different

alignment tools on the quantification results. The first step (top) involves organizing or creating all the necessary files to apply the alignment programs. The second step focuses on the selected Bash script that combines all the different programs. Customization of path information is required in this step, along with adjusting program specific options in the program call lines. The third step provides insight into the data structure created by the Bash script. Result folders are numbered in processing order, with names indicating the tools used. (A) demonstrates this for scripts 30_se_mrna.sh and 30_pe_mrna.sh, while (B) does so 30_se_mir.sh. The fourth step involves analyzing the data in R.

Prerequisites

The computing platform needs to be a Linux operating system. All our examples are tested on a Debian 12 system but should not be limited to a specific Linux version of 2024 and above because of Bash and standard libraries. The program versions themselves might change slightly over time but this might only have some impact on the program parameters less on the overall design of the Bash script command file.

The presented example for transcriptomic 'single end' mRNA data (30_se_mrna.sh) is based on one aligner, a quality control, a UMI normalization option and can be seen as a simple warm up scenario to understand the framework and its application. Nevertheless, it is even useful in everyday situations.

The same setup with only one modified module inside is provided for transcriptomic 'paired end' mRNA data (30_pe_mrna.sh) and is fully functional in this form.

The most advanced example for transcriptomic microRNA or small RNA data (30_se_mir.sh, single end) consists of three aligners and seven pre- and post-processing tools, along with some additional utilities to generate a multiperspective view of the data alignment and the results of the various analyses. All three scripts can be found in the /bin folder of the unpacked MAF.zip file (Figure 2).

For all three approaches, minimal test data as well as suitable index and reference files are provided to reduce the time needed to get started. Many of the necessary files are available for download due to their large file size.

Within each command script, the parameter section at the top needs to be adjusted for some user specific run parameters. The script is ready to be executed after unpacking the supplementary file MAF.zip in the user's home folder. Additionally, all the large files need to be downloaded and unpacked in the appropriate subfolders of the MAF.zip folder structure (Figure 2), as descibed in the readme.txt files within those folders. All the relevant programs must be accessible and executable by the script commands. Detailed instructions for these initial procedures are provided at the top level of the MAF.zip folder structure in 'install.txt'. If the provided program versions on are not used, the 'readme.txt' file at the top level supplies download locations for all the programs.

Additionally, we provide specific functions to import the results into R [15] , a platform for mathematical computing, to analyze the quantification data (counts). The R scripts can be found in the folder /R_import_scripts. Additionally a folder /R_dedup_script provides a naive deduplication script for the framework if this option is set.

In the folders /data, /index, /reference, additional 'readme.txt' files explain how to fill the currently empty folders with certain files (Figure 2). This step is necessary to start with the provided resources, but if other resources are of interest those can be used too.

Linux libraries

Several essential computer libraries need to be installed to establish a functional system. The details are included in the supplementary file 'install.txt' in MAF.zip at section: # Libraries #. The following section is intended as an overview of the utilized programs.

The sequence tools

Additionally, the tools themselves need to be installed. For this purpose, and to simplify the complex work, the present versions of the tools are provided on the download page (see the 'Supplementary material & information' section at the end). The compressed tool files need to be placed in the /install folder of the unpacked MAF file. Installation procedures are described in the 'install.txt' file.

The following list briefly introduces the tools with a focus on their transcriptomic purpose.

Samtools **-** an established program family for processing high-throughput sequencing data. The program collection is based on three basic elements : Samtools, BCFtools and HTSlib. Here we are only using Samtools [16, 17] sub programs: index, view, sort, markdup and idxstats for quantification.

STAR **-** sequence aligner. The program is well described in publications [10, 11, 18] . This tool is used by the ENCODE project [19] and handles gapped situations very well.

Bowtie2 **-** sequence aligner. The recommended publications are [20, 21] . This program is widely used in the genome community, although it can also be applied to transcriptome analysis and can handle gaps in version 2.

BBTools **-** sequence aligner. The tool collection is only partly published [22] and more detailed information can be found via links in the main 'readme.txt' file. BBMap performs less perfect in the presented small RNA situation.

FastQC **-** fastq read file quality control. Before starting any sequence alignment, it is essential to perform quality control on the sample(s) to asses their homogeneity and specific characteristics. FastQC [23] is a good choice for this purpose and is also scriptable.

fastp tool **-** fastq file preprocessing, quality control, and deduplication. The fastp tool [24, 25] is specifically used for preprocessing purpose in the presented scripts: deletion of adapters, cut and move UMI codes to the 'qname' variable of the SAM format, and removing reads with a length smaller than 16 bases. In a second run, the tool trims the two CA bases at the start of the read and again removes reads with a length smaller than 16 bases.

nubeam-dedup tool **-** de-duplicates sequencing reads without a reference genome. This approach is an option if no UMI codes are used and no perfect reference is available. Nubeam-dedup [26] is a pure C++ tool, so it is fast.

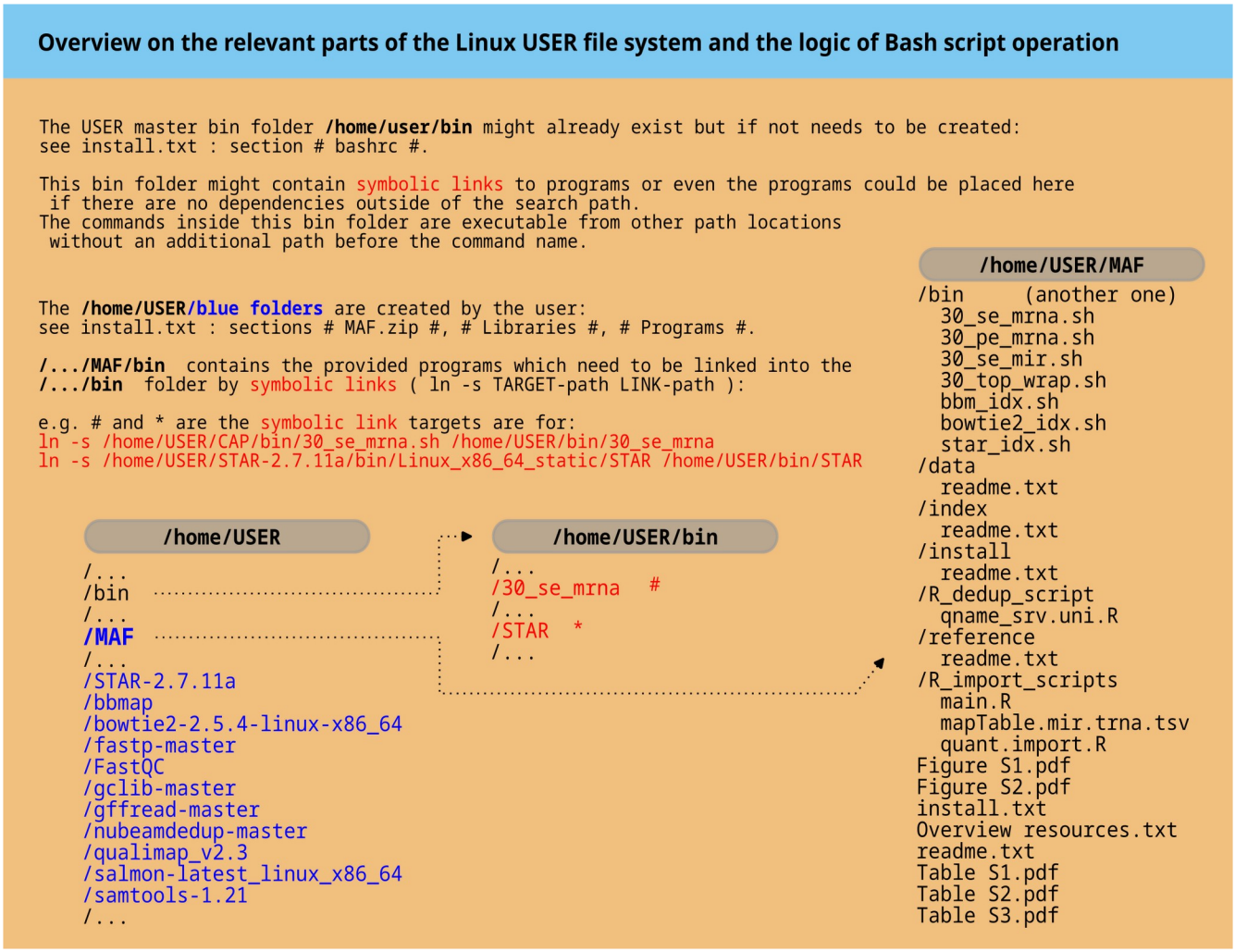


Figure 2. The master plan for folders.

The left part of the scheme describs system folders like the 'home/USER' path, which exists after a standard Linux installation is successfully completed. The name USER should be changed to the real user name. The blue color indicates the newly created folders after unpacking the MAF.zip file and installing all the programs. The 'home/USER/bin' path contains symbolic links to the Bash command scripts in MAF/bin and the installed program executables. The far right part shows the folder structure of the unpacked supplementary MAF.zip file, with additional information files like e.g. Figure S1.

umi_tools - improves quantification accuracy by UMI code based deduplication. The umi_tools [27] is a Python tool set with 6 major program features, but only the "dedup" part is used here. Since it is Python, we recommend installing it via pipx as a self executable program, as described in detail in the supplementary file 'install.txt'.

Salmon - transcript quantification. The Salmon tool [28] is a very fast quantification program. The tool is used here in alignment mode, which is recommended. This tool also has a single cell extension called Alevin. There is a similar tool from a different group named Kallisto [29] .

QualiMap - transcript quantification. Only the quantification feature is used in the context of this program [30] developed with biology in mind.

GffRead - not only performing a genome- to transcriptome fasta modification. GffRead together with GffCompare has a broad functional scope [31] . The tool is used here to create an adjusted fasta file necessary for quantification with the Salmon tool together with the transcriptomic BAM files of STAR (option: --quantMode TranscriptomeSAM).

Description of the main command scripts

The core element that combines all the different programs and data formats together is, as described above, a Bash script. More precisely, there are three different examples of such a type of command file named '30_se_mrna.sh', '30_pe_mrna.sh' and '30_se_mir.sh' (supplementary file MAF.zip, folder /bin). Additionally, a wrapper shell script '30_top_wrap.sh' is provided for combining several different tasks and running them from this top level file altogether, for example, over several days. Index creation scripts for all three alignment program specific indices are also provided here in case references other than the provided ones are to be used. Additional information can be found in the alignment programs' help files.

Obviously, this wrapper script, as well as the other Bash files, might need some customization to fit for the user's computing environment. The customization takes place primarily at the beginning of the scripts. If the lines that call the programs need to be adjusted for their parameters, those lines need to be identified and adjusted. The script names include numbers which represent the current version number of the scripts.

The first and second file are simple examples for performing single end and paired end mRNA analysis using the STAR alignment program and providing some postprocessing support for deduplication. Viewing the script files is done using a text editor, which are programs specifically designed for pure text based files like the Bash computer language or small fastq sequence files. Having an editor with a text coloring scheme for the Bash script language helps reading the commands more efficiently. An example inside a command terminal would be "nano [path/]filename" and in the graphical environment it might be Gedit, Pluma, or others.

Single end and paired end Bash scripts

Looking at the code for the single end alignment script '30_se_mrna.sh' with an editor, it starts with comment lines denoted by a hash character '#' before the comment text, indicating that they are not active commands. These lines typically provide helpful information for the user or briefly explain the purpose of the command line(s).

The 'parameters' section at the beginning of the script file is crucial for running the script. In a user specific computing environment, it may require some adjustments:

```
#### parameters here need to be adjusted for specific runs
srcf="$HOME/MAF/data"              # /absolute path to input fastq / fq folder
ext="fastq"                        # file extension
tf="$HOME/MAF/results/se_mrna"     # /absolute path to result folder
```

```
umiB="0"    # barcodeB: 0:no dedup, 1:R by umi, 2:samtools markdup, 3:umi_tools
...
```

The **srcf** parameter or variable is receiving a quoted path to the fastq data folder. "$HOME" is a system variable containing the user's home path, but it can also be written literally. **ext** is the variable containing the file extension of the data files without the dot, normally 'fastq' but sometimes also 'fq' or something else. The important thing is not the extension but that the file itself is in fastq format.

It is good practice to check the file format before using it by opening a terminal and executing the following command: **`head -n 10 filename`** to see if it is correct. In the case of a simple 'fasta' file format (its' own section below) it might not work without some adjustment to the script but in the case of fastq everything is fine. Compressed files with the extension .fastq.gz might also require extra effort for all programs to work with such a file type. Therefore, for the first experiments, stick with a pure unpacked fastq file (command: **`gunzip name.fastq.gz`**).

The **tf** parameter contains a path to a result folder. This folder may already exist but does not need to. In every call of the script, the folder is deleted and then newly created so be careful.

The **umiB** parameter has four options indicated by quoted numbers. '0' indicates no deduplication is done. '1' calls a simple R script to unify all different UMI codes, while '2' uses the Samtools markdup feature, and '4' (recommended) uses the well published umi_tools program to capture PCR amplification bias. This function is applied after the alignment and before the quantification starts. In the case of the R script, an R standard installation needs to be performed.

The **gdiSC**, **gtfSC** and **tfaSC** parameters provide three paths: the STAR aligner index folder path used, the appropriate gtf file corresponding to the file used in the index creation step of STAR, and the *GffRead* created transcriptome fasta file based on the fasta file used for the STAR index creation process. For custom files see MAF/bin/star_idx.sh.

At this point, everything is normally done, and the script can be executed. The 'echo' commands in the Bash script provide feedback on the processing status in the terminal window.

Howerver, there are further program sections described in the following paragraphs.

The terms 'module' and 'number' describe a Bash function. The function combines several Bash command lines, and these functions are specific to particular task. The order of the Bash function calls at the end of the script indicates the processing scheme. A full overview of all the modules of the command scripts is provided in Figure 3A for 30_se_mrna.sh and 30_pe_mrna.sh, and in Figure 3B for 30_se_mir.sh.

Module 1, internally named 'icreate', creates the result file folder or deletes (!) and recreates it if it is already exists. Additionally, all symbolic links to the raw data files in the source folder are created inside this first folder (00raw). Note that the two mRNA scripts, single end 30_se_mrna.sh and paired end 30_pe_mrna.sh, have some differences in their code to handle the situation of one fastq file in

single end versus two fastq files in the paired end situation (there may also be interleaved files, but they are not considered here).

The **Module 2** - FastQC - internal name 'qualcon', generates an html file and a corresponding zip file that can be opened in a web browser. The file provides a detailed insight into the raw data quality for each raw data file. Therefore, each raw data file has a quality control file pair in a separate folder named 'sample' followed by a number (the order of the sample names is preserved in filelist.txt).

The file system structure in this case is as follows: /result folder /01qcAout /sample1, where the file names are sample1_fastqc.html and sample1_fastqc.zip. This type of structure is repeated for every main module and task. The file namelist.txt is also used by the R import functions to create a data structure inside R with the original names. Therefore, major changes in the bash code probably may also require changes in the R code.

The next module is the deduplication **Module 3** termed 'barcodeB' as already described in the 'parameter' section above concerning the options. Every option has its own code element composed of a few to several lines of code.

As can be seen here and in the other Bash functions, the code lines trigger the creation of a command line file, in this case called 'sub1.sh', which is finally executed by 'cmds/sub1.sh'. The entire script generates several command line files, which are all collected in the /cmds folder and additionally executed through a command line in the script. If script problems exist, these files might provide some insight into the situation.

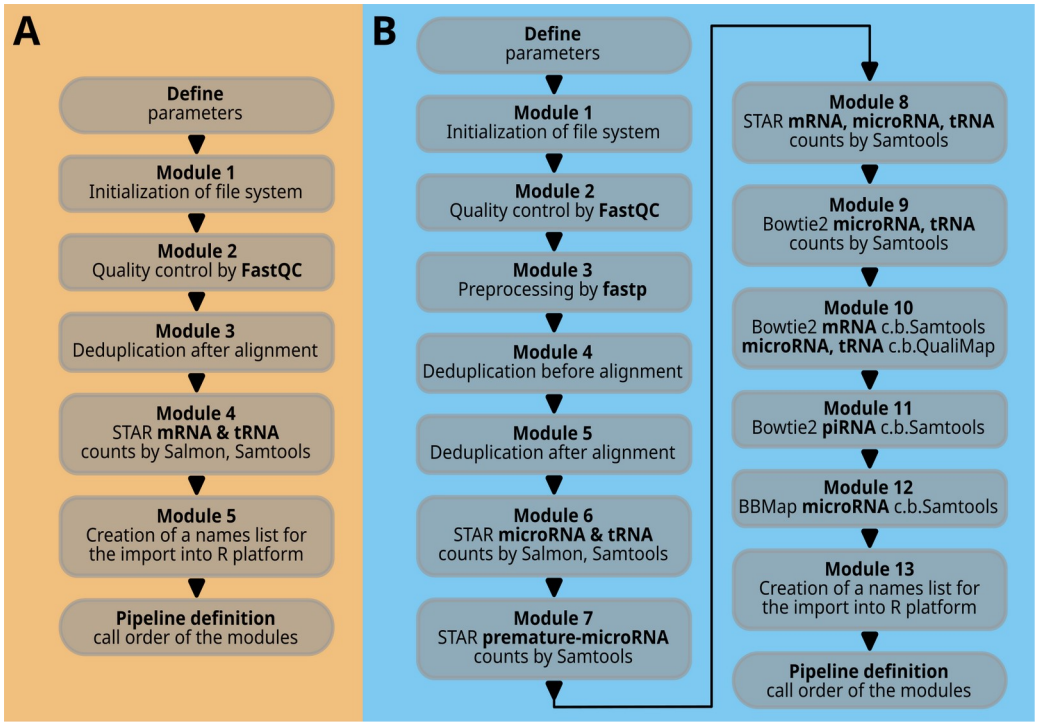


Figure 3. Code overview. In part (A), the modules of the single end or paired end workflow are listed. In part (B), the comprehensive microRNA workflow is presented, which is composed of many more modules. A more detailed description of all the modules is given in the text.

Module 4 - STAR alignment (named staroutC) combines several tasks. The STAR command line itself with all the parameters, some Samtools actions, **`samtools idxstats`** counting, Salmon counting, and

deleting some intermediary files. All these actions are collected in several command files (mapstC.sh, sc3C.sh, cu1c.sh, sc3C.sh, sal3.sh, cu1c.sh) which are stepwise executed. It should be noted that the STAR command line is slightly different for single- and paired end data.

An important aspect of the code is the usage of the xargs command. The ls command creates a sample file list piping it into the xargs command, which generates, with the template command line written after xargs, an output command line for every input element. This output command line is saved into a Bash script file. 'echo' in this context is the command used to create a real text line out of text and variables, which can be written to the file. This file is then executed. '>'denotes save and overwrite, while '>>' means save and append to the file.

Module 5 - named 'fnlA' - extracts the pure name list with the correct order from the filelist.txt statistic file. This list is necessary to translate back the 'sample1' nomenclature back into the real sample name. The cut and tr commands are used to extract a specific name element from the input text file.

The final section of the Bash command script executes all the individual Bash functions (modules) in a specific order. 'echo' feedback lines are included to indicate in the terminal which modules have been successfully executed. The 'date' command is used to estimate the program's runtime.

Small RNA Bash script

The script for microRNAs and tRNAs, or small RNAs in general (30_se_mir.sh), is shown in Figure 3B and is much more complicated. After an in depth inspection, it can be seen that the basic structure of the modules is very similar or identical to the already described modules above. Therefore not all modules are discussed here again instead some important code elements are selected for discussion.

The structure of the microRNA script starts here with a very long section of parameters needed to run this script. The parameters follow the order of the script: general elements like source path, etc., in the first paragraph, three STAR aligner paragraphs for the three STAR modules, three Bowtie2 paragraphs and one BBMap paragraph. All of these file or folder links need to be valid and carefully selected concerning the intended alignment task.

The following Module 1 and 2 are similar or identical to the above described files 30_se_mrna.sh and 30_pe_mrna.sh. A new insertion here is **Module 3** (named 'preprocess'). This module is uses the program fastp to remove adapter sequences, move the UMI identifier to the 'qname', trim the CA prefix, and remove reads smaller than 16 bases. This module is set up for single end reads. Additionally, the program also creates another set of quality control files in its folder, which can be viewed using a web browser (fp1- and fp2.html and .json).

Also new is **Module 4** ('seqcodeA'). This module is specialized for tasks that require deduplicating sequences based on pure sequence similarity, rather than barcode based methods. Module 5 (Figure 3B) correspond to Module 3 in Figure 3A, which is based on barcoding ideas.

Modules 6,7,8 ('staroutA', 'staroutB', 'staroutC') are all STAR aligner based modules. The first approach focuses on microRNAs and tRNAs, the second on premature microRNAs, and the third additionally on micro- and tRNAs, as well as mRNAs. The latter is sometimes useful to see whether

mRNA sequences share sequence similarities with small RNAs and therefore produce ambiguous counts.

Modules 9,10,11 ('bt2outA', 'bt2outB', 'bt2outC') are Bowtie2 based and provide a comparative third and fourth view on micro- and tRNAs, directly by Bowtie2 and indirectly by the QualiMap tool. Additionally, chromosomal (and consolidated contig) count numbers are created to get a general overview of the count distribution of transcriptome features across genomic master elements. The third approach specifically focuses on piRNAs, which form a second small RNA domain, and only very few elements show overlaps with microRNAs (possibly due to bugs in the database). These overlaps were identified by the blastn tool and the microRNAs took precedence before we created our custom references.

Finally **Module 12** ('bbmoutA') uses the BBMap aligner for the detection of microRNAs and is the fifth approach in this category.

The last two modules are very similar or identical to the script approaches in Figure 3A.

Formats - understand the role of fastq, fasta, GTF and transcriptome fasta files

The input file format is based on the 'fastq' format. Two examples are given below:

```
@SRR19991171.1 1 length=75
ATTCANATGATTTCTATGATGAATCAAACTAGCTCACTATGACCGACAGTGAAAATACATGAACACCTGAGATGG
+SRR19991171.1 1 length=75
???????????????????????????????????????????????????????????????????????????????
```

```
@VH00416:321:2222KKCNX:1:1101:28919:1000 1:N:0:CCTTTCAC+AAACAAGA
GGGAAATCAGGCCGACGCTGTCCCGCCCCACGAACTTCAGGCTCCTCTCTCTGGAGACAGGTCCATGCCTGT
+
III9IIIIIIIIIIIII9IIIIIIIIIIIIII9III9IIIII9IIIIIIIIIIIII9IIIIIIIIIIII-II
```

The two different examples show a typical block of 4 lines for one 'read'. The first line contains the unique identifier starting with a '@', which may or may not have split characters ':' and barcode sequences at the end. The second line comprises the sequence, the third line contains a '+' with or without an identifier (standard) and the fourth line contains the quality sequence. The Phred score used in this line is the probability of an incorrect base call, ranging from 1 to 0.0001, marked by symbols !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI.

The fastq file might be compressed as described already with an extension of '.gz', but other compression extensions are also possible. It is recommended to decompress the files before running the scripts. However, the script could be extended to handle gz compressed files, although not all tools might support this. Since many programs have their own file decompression handling, it is more general to start with uncompressed files.

The 'fasta' files may contain any type of sequence starting with the '>' character followed by a name and in the next line, the sequence. Some institution prefer a line break every 80 characters, as shown in the

example below, but there is no fixed standard. The sequence does not contain gaps. The next continuous sequence would start again with a name line, a line break, and the sequence:

```
>sequence name
ATTCANATGATTTCTATGATGAATCAAACTAGCTCACTATCTATGTATCTAACACAAGCAAGGGGTCGAAAAGATTGGGC
TCGAGAATGCCGCATAGGCAAATTATGAGCCTGCCGCTTCCCTATACCATCGATCTCACCCCTCGATCGGCCCAGATGGA
GACCGACAGTGAAAATACATGAACACCTGAGATGG
```

The human chromosomes have many lines until the next chromosome name appears. The nano editor in the terminal might be able to handle such large files, at least for viewing and searching, but the main memory of the computer being used might be the limiting factor.

The genome fasta file is necessary to generate the STAR, Bowtie2, and BBMap index data structures, which consist of several files in each case. The files star_idx.sh, bowtie2_idx.sh and bbm_idx.sh are provided to assist in creating new index files.

The STAR file also includes the call of the **gffread** tool to generate the transcriptome fasta file from the primary genome fasta file. This is necessary for using the Salmon quantifier tool. The **samtools idxstats** quantifier does not require such a reference. It is important to note that the Samtools approach counts but does not normalize count data according to paired end situations, unlike Salmon. This means that in strong overlapping paired end approaches the count numbers in Samtools compared to Salmon behave approximately 2:1.

The GTF file (sometimes a GFF file needs to be converted to a GTF file) describes features within the pure fasta sequence file. The main information includes the sequence start and end positions (column 4, 5) and additional annotation information. The chromosome name in column 1 must correspond with the genome fasta chromosome name (note: it could be written as '1' or 'chr1' or something else). Not all columns are shown below:

```
1 2                                  3           4     5     6 7 ...
1 transcribed_unprocessed_pseudogene gene        11869 14409 . + ...
1 processed_transcript               transcript  11869 14409 . + ...
```

As long as standard feature information is necessary standard GTF files can be used. In the described microRNA framework script, microRNA and tRNA information from the respective databases is inserted into the standard GTF files at the appropriate place and format, which requires a lot of customization procedures. This is a whole topic on its own. The same is also done for the piRNAs. This needs to be done because the standard Ensembl and other main databases are still missing a lot of those features.

Sometimes it is necessary to shrink the features down to the targets of interest to reduce the run time significantly. However, keep in mind that the whole genome view can also reveal signal crosstalk between different features or transcripts. This is one of the reasons why the microRNA script includes several different approaches and even whole transcriptome views.

An overview of all alignment procedures, parameters, and files is provided in supplementary Table S1, which is part of the supplementary file MAF.zip.

Results

As described earlier, the concept has been applied in some published studies and is also being used in ongoing studies. To complement the published information in a more detailed and methodological manner, this section focuses on the stringency of the algorithmic results.

The presented study is based on Ensemble version 111 with a microRNA and tRNA extension, and the sequence data is mentioned in the 'Availability of data and materials' section at the end.

The microRNA script was applied to human samples, which were divided into five conditions and two technical control samples that were not part of the conditions. The research questions addressed here are: a) to analyze the overall count distributions, microRNA distribution, and proportions to identify any problematic trends related to overall expression strength, and b) to understand how the final significant candidates establish their significance, which in several cases could lead to rejection upon examination. Both ideas provide insight into the complexity of the data and the challenges of identifying robust and sufficiently annotated candidates.

The application of the 30_se_mir.sh script generates multiple alignment results on small RNAs and other transcripts. These result structures are then imported into R using R scripts, preferably the ones provided in /MAF/R_import_scripts. Subsequently, two differential analyses are conducted using the DESeq2 R package, and R graphs are created to present the results. It is important to note that the DESeq2 procedures may slightly alter the raw data counts due to the variance smoothing approach ([32] and the Bioconductor page). So the DESeq2 counts of the differential analysis might need some comparison with the imported raw data counts.

In Figure 4, the raw data situation after import is depicted. It is evident that the majority of counts correspond to RNA transcripts that are not part of the microRNA family. The small RNA library preparation kit used filters for sequence fragments that are not greater than 100 bases. Therefore, there are likely many other RNA transcript families expressed, either as full molecules or fragments, which can be captured by the alignment procedures.

It is important for the differential analysis to determine if the total number of counts and the microRNA counts per sample are following an identical trend, as this would invalidate differential results. The trend in mean values per sample shows a very weak positive correlation, indicating that there is no strong correspondence. The median correlation, as expected, is higher but less informative regarding the feature properties of the distribution. The features themselves are much more heterogeneous, weakening the positive correlation. Examining the ratios reveals that the samples do not exhibit extreme heterogeneity within or between groups, making them suitable for a final differential analysis.

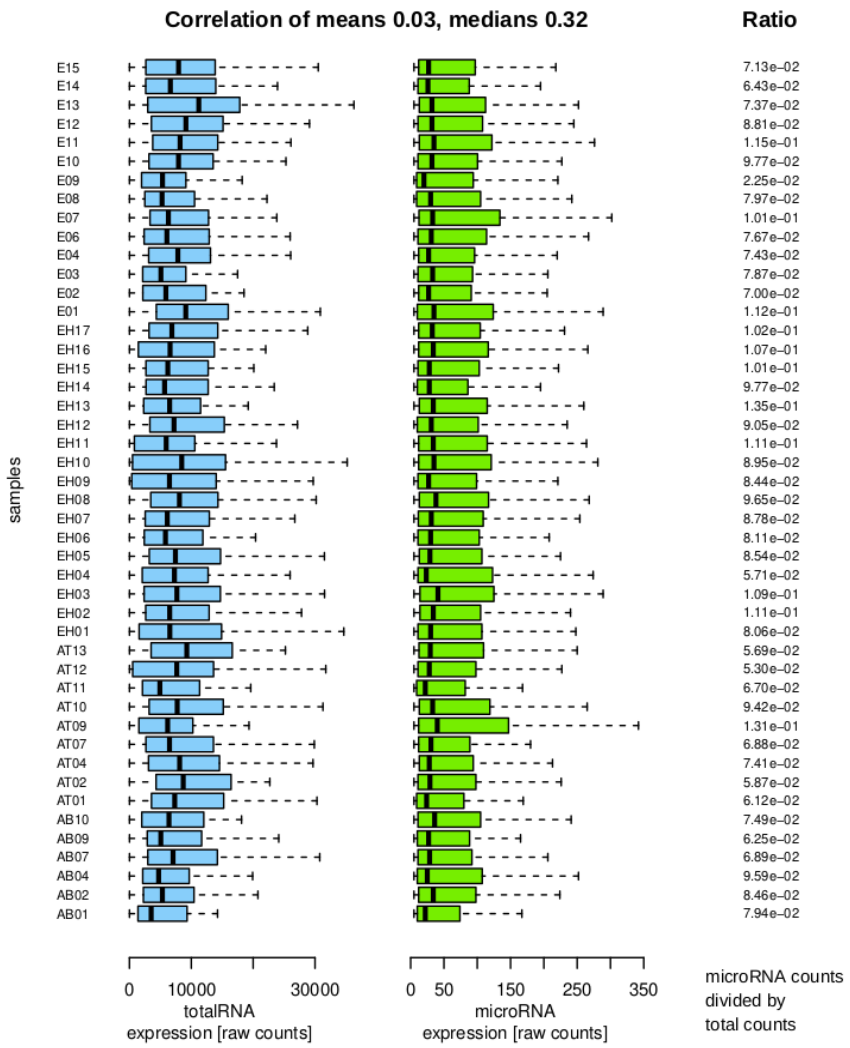


Figure 4. Sample distributions and properties

On the left, the sample acronyms are given. The blue boxplots represent the total RNA count numbers, while the green boxplots represent microRNA expression count numbers - as can be seen, there are much fewer counts in this fraction. On the right, the ratios between the total and the microRNA counts are given per sample. At the top, the correlation coefficient of the mean and median values of the sample distributions is calculated between the total RNA group (blue) and the microRNA group (green).

The differential analysis is mainly dependent on the group variance, the size of each group, and the absolute count numbers per feature. Performing statistics at that level is advised but not explicitly shown here. In the following graphs, this topic is taken up again from a slightly different perspective. Table 2 provides more insight into the final significant candidate numbers out of the pool of possible candidates, which is also a benchmark number that can be addressed by a sampling procedure at the input data level. It is noticeable here that the coherence between the STAR and Bowtie2 aligners is high, despite some existing differences. BBMap, in contrast, is not able to deal appropriately with the microRNAs in this data set (line five).

Figure 5 and supplementary Table S2 show how the established candidates overlap across different technical approaches and which alignment procedures lead to certain candidates. Even the two different quantification algorithms based on the same STAR approach lead to some differences in the results, particularly in weak cases concerning the associated p values. More significant differences can be observed between the STAR and Bowtie2 approaches, as well as the Bowtie2 approach with the

QualiMap based quantification. In these cases, significant candidates are disappaering or new ones turn up in the result lists.

Table 2

Number of significant results for the differential analyses with DESeq2.

A

microRNA	DE-1	DE-2	
of 1074	46	14	STAR mt index, Salmon quantification whole transcriptome (sts).
of 1076	46	12	STAR mt index, Samtools quantification idxstats (stm).
of 1359	22	11	Bowtie2 mt index, end-to-end, Samtools quantification (btq).
of 731	45	14	Bowtie2 whole index, Qualimap quantification, mt reference (btm).
of 2016	0	2	BBMap mt index, Samtools quantification (bbm).
of 994	49	17	STAR pmm index, Samtools quantification (pmm).

B

tRNA	DE-1	DE-2	
of 141	6	24	STAR mt index, Salmon quantification whole transcriptome (sts).
of 142	6	24	STAR mt index, Samtools quantification idxstats (stm).
of 257	35	15	Bowtie2 mt index, end-to-end, Samtools quantification (btq).
of 235	18	17	Bowtie2 whole index, Qualimap quantification, mt reference (btm).
of 256	8	8	BBMap mt index, Samtools quantification (bbm).

DE-1 and DE-2 denote the two differential DESeq2 analysis sets considered here, with multiple testing correction according to Benjamini-Hochberg. The 'of' refers to the input number of non negative expression features for the differential analysis. 'mt' stands for microRNA and tRNA, while 'whole index' refers to all possible expression features. 'pmm' denotes the pre-mature microRNA approach. Additionally, refer to the Venn diagrams in Figure 5 and Table S2 for a more detailed analysis of the five mature microRNA approaches.

The most distinct program is BBMap, but with this type of microRNA data, it is not really useful. In the first differential analysis, it can not find any microRNA, while in the second differential analysis, a prominent candidate is found, but only this one is confirmed, and a highly significant one is newly introduced. Taking a look at the details (supplementary Table S3), the counts are only created by BBMap. The other alignment programs and quantification procedures find exactly zero counts. The sequence of this BBMap specific microRNA is conspicuous concerning the sequence, which contains one A, three Gs, and again a stretch of 10 A bases in the beginning. This might point to some weakness of the BBMap algorithm to handle such types of sequences, which the other alignment programs might not have (poly A).

The premature sequences aligned with STAR shown in line number six of Table 2A indicate that the overlap of significant premature microRNAs with mature microRNAs using STAR or Bowtie 2 ranges from 24% to 33% for the first differential set and from 34% to 41% in the second differential set (see

supplementary Figure S1). The main reason for this is that the premature analysis represents a different biological state compared to mature microRNA forms, which are the final regulatory molecules. Additionally, there is may be a crosstalk between both forms, with fragments of the premature form contributing to the mature forms. Therefore, the alignment procedures may not efficiently distinguish between premature and mature forms. Nevertheless, the results of premature microRNA analysis substantially support the results of mature microRNA analysis in this case, even though regulatory conclusions may be weaker.

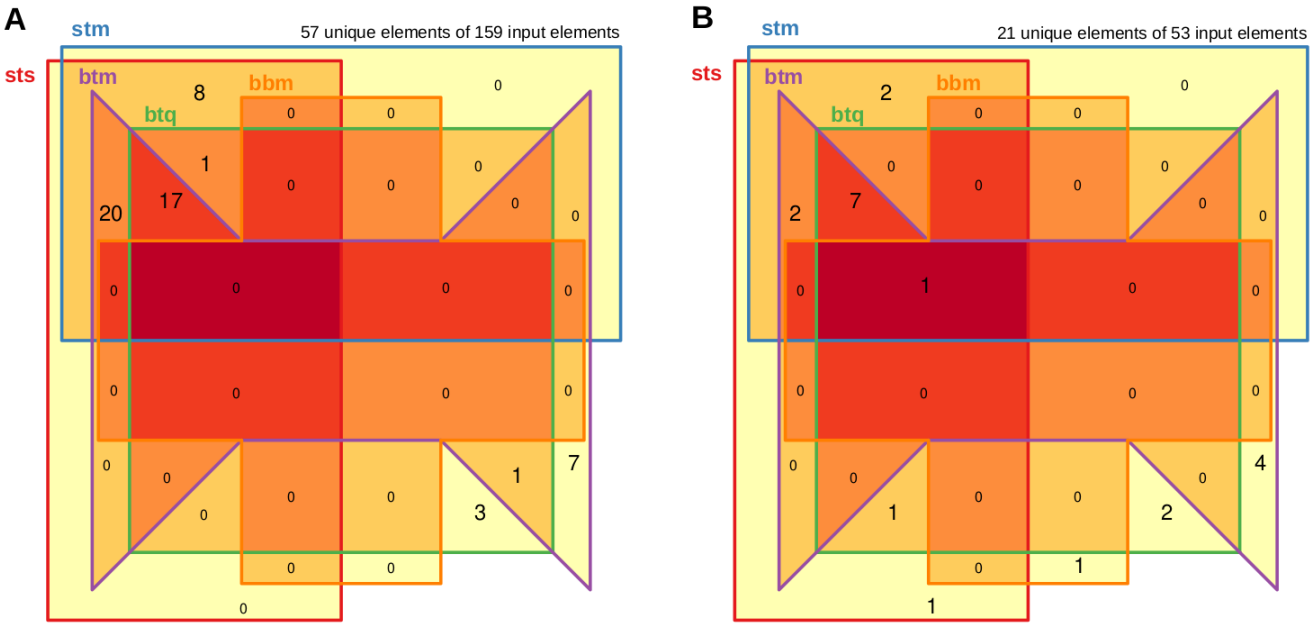


Figure 5. Overlap of significant candidates between the 5 algorithmic approaches.
Panel A shows differential analyses of the DE-1 set and panel B shows differential analyses of the DE-2 set. Each set is composed of the five mature microRNA approaches and shows their overlap between significant candidates. The darkest red color represents the overlap of all sets, while yellow indicates features unique to one group. The nomenclature of the five analyses, from 'sts' to 'bbm', is explained in Table 2 in the right column. The top text line denotes the total number of unique features compared to all features of the five groups, which form the basis for this Venn diagram. The unique features may be unevenly distributed among the groups.

In Table 2B, results with tRNAs can be seen. These RNAs are 3 to 4 times longer and have a completely different biological role compared to microRNAs. It is important to note that the trend of the number distribution in this table of differential candidates is remarkably different from the table with the microRNA results. This is a good indicator that we are not only mapping expression strength trends with the significant result sets but instead generating molecule class specific results. As a side aspect, it is interesting to see that the BBMap results are at least numerically better with this molecule class.

Finally, it is important to examine the count distribution per feature. Many highly significant microRNA candidates may appear impressive based on their p values, but are less convincing when

considering how the significance is influenced by the distribution of individual count values per group. The top candidates from the two differential analyses are presented in Figure 6.

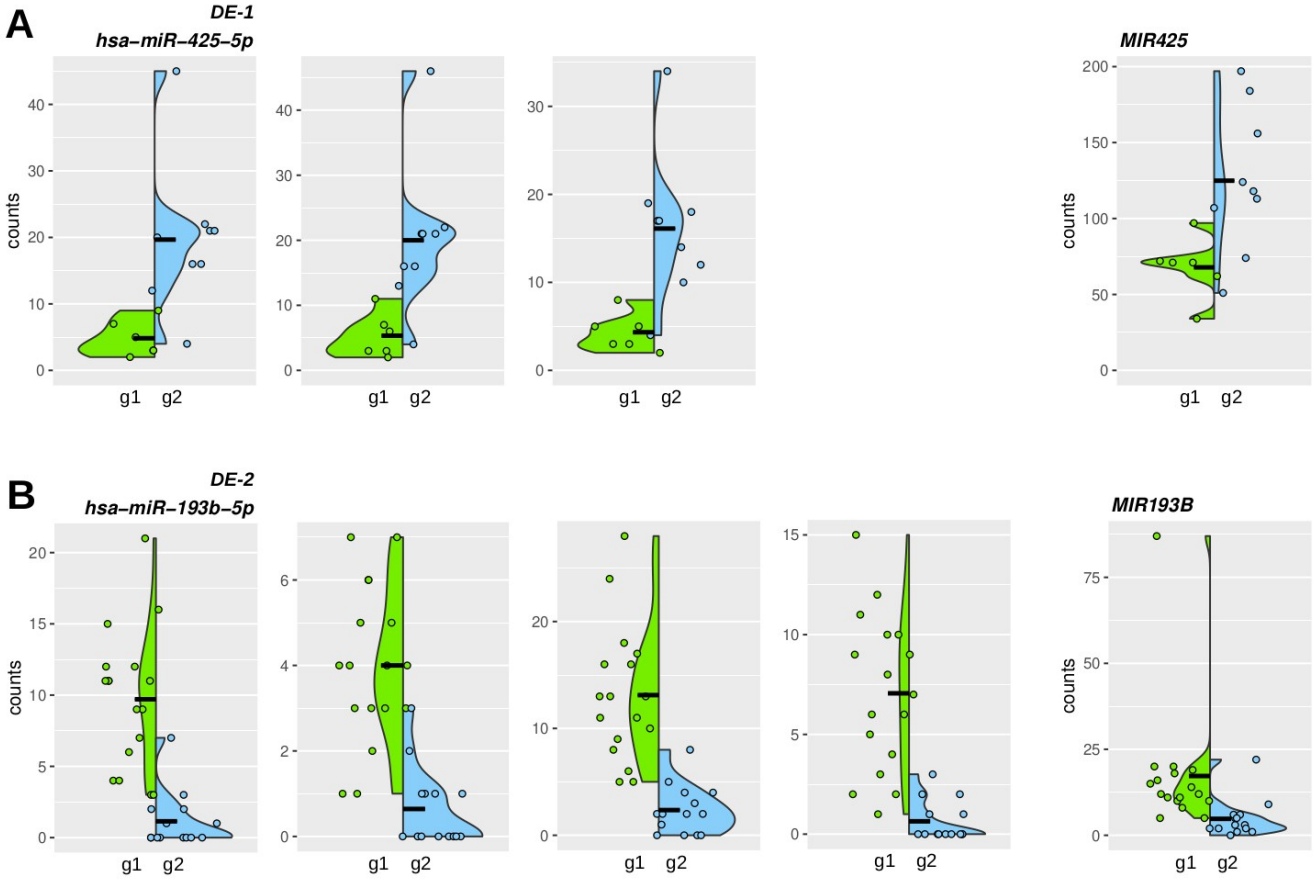


Figure 6. Group distributions of all the individual count values.
The count values in this graph are presented as a violin plot with all the individual count values denoted by small dots filled with the respective group color. Because many count values in a low count situation have the same value these values are randomly scattered horizontally - so these differences do not mean anything. Vertical differences obviously are y scale differences. 'g1' (green) and 'g2' (blue) depict the two groups, and 'DE-1' and 'DE-2' differentiate the two different sets of comparisons described already before. The left part presents the different observed microRNA count distributions, and on the right part, the premature microRNA results are given, identified by their different naming scheme denoting the premature state.

The differences between the algorithmic procedures are small, even in the premature forms denoted on the right side by the uppercase 'MIR' identifier. The count number scale of the premature forms is slightly to remarkably higher compared to the mature microRNA forms, which is expected. It is important to note that the higher count number does not really affect the outcome of the differential situation. To gain a broader insight into the wide variances possibly occuring behind those features *with* significant p values, further results are presented in supplementary Figure S2 for microRNAs, tRNAs, piRNAs and the premature microRNAs.

Of course, further properties of the final candidates need to be investigated, such as which ranges of transcriptomic features are actually covered by the reads. This is obviously necessary, but due to the additional programs and their introduction it goes beyond the scope of this discourse.

Conclusion

The multi-alignment framework MAF and the provided protocol allow researchers to better understand the complex influences of technology and software tools on their results, enabling them to select robust candidates. The time investment is reasonable as a significant amount of work has already been put into developing a basic framework, customized indices, and references, especially for the microRNA situation.

For microRNAs, the BBMap alignment program may be less useful compared to STAR and Bowtie2. Overall, STAR appears to be the most stable procedure. However, each procedure has its specialties, justifying the need for comparison with each dataset under investigation. The differential analysis clearly shows that significance alone is often not the best measure of relevance while count distributions add an additional criteria.

Furthermore, the MAF approach helps researchers develop more advanced bioinformatics knowledge. It is flexible, based on open source ideas, and allows researchers to focus on biology while exploring the strengths and limitations of computational methods from a multiperspective view. This approach also highlights the impact of computational methods on the interpretation of sequence experiments and their results.

Availability and requirements

Project name: MAF

Project home page: SOURCEFORGE or CCSR

Operating system(s): Linux

Programming language: Bash

Other requirements: Third party tools are described in *MAF.zip* in *readme.txt* and *install.txt* and are as a courtesy provided on the CCSR download location as well as all the different custom references.

License: GNU GPL

Any restrictions to use by non-academics: GNU GPL

Ethics approval and consent to participate

This method work is based on techniques in the process of next generation sequencing data alignment and -analysis and therefore is theoretical work. The used data in this study is publicly available. Furthermore the presented transcriptomic results cannot be tracked back to individuals.

Consent for publication

Not applicable

Availability of data and materials

Supplementary Figures, Tables and materials associated with this article can be found in the file MAF.zip in the supplementary information section of the full text article or on SOURCEFORGE. The MAF.zip needs to be unpacked by preserving its structure, for example, in the HOME folder. Due to file size reasons (84 GB) all the custom versions of the reference- and index files, the two test data files and the third party programs, belonging to this MAF release are available via the CCSR research website. They need to be placed and unpacked after the download in the respective template folders available in the MAF.zip folder structure. The used public data is available in GEO: GSE275002, GSE207522. The test data is based on one sample of GSE207522 (microRNA) and one sample of a so far unpublished data set for the mRNA data. All the mentioned elements as already described above can generally be exchanged according to the demands of the user but might require extra efforts by the user. So, the MAF.zip is, in a certain way, a functional starter kit that enables the user to realize his research plans by first starting with the provided plan based on Ensemble version 111 with a microRNA and tRNA extension.

Competing interests

The authors declare that they have no competing interests.

Funding

We acknowledge support from the Open Access Publication Fund of the University of Münster, Münster, Germany.

Authors' contributions

XZ: methodology, software, formal analysis, writing – original draft, writing – review & editing. EK: conceptualization, methodology, data selection, visualization, supplementary information, formal analysis, writing – original draft, writing – review & editing, supervision.

Acknowledgements

Not applicable

References

[1]M. S. Rosenberg, Sequence Alignment : Methods, Models, Concepts, and Strategies, (2009) doi:www.jstor.org/stable/10.1525/j.ctt1pps7t.

[2]E. W. Myers, A history of DNA sequence assembly, it Inf. Technol.. 58 (2016) 126-132 doi:10.1515/ITIT-2015-0047.

[3]C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. S. Papadopoulos, K. Bealer, T. L. Madden, BLAST+: architecture and applications, BMC Bioinform.. 10 (2009) 421 doi:10.1186/1471-2105-10-421.

[4]W. J. Kent, BLAT--the BLAST-like alignment tool., Genome research. 12 (2002) 656-664 doi:10.1101/gr.229202.

[5]Y. Wang, C. Nie, T. Zang, Y. Wang, Predicting circRNA-Disease Associations Based on circRNA Expression Similarity and Functional Similarity, Frontiers in Genetics. 10 (2019) 832 doi:10.3389/fgene.2019.00832.

[6]W. R. Pearson, D. J. Lipman, Improved tools for biological sequence comparison., Proceedings of the National Academy of Sciences. 85 (1988) 2444-2448 doi:10.1073/pnas.85.8.2444.
[7]W. R. Pearson, Selecting the Right Similarity-Scoring Matrix, Current Protocols in Bioinformatics. 43 (2013) 3.5.1-3.5.9 doi:10.1002/0471250953.bi0305s43.
[8]J. Kim, M. Ji, G. Yi, A Review on Sequence Alignment Algorithms for Short Reads Based on Next-Generation Sequencing, IEEE Access. 8 (2020) 189811-189822 doi:10.1109/ACCESS.2020.3031159.
[9]R. J. Musich, L. Cadle-Davidson, M. Osier, Comparison of Short-Read Sequence Aligners Indicates Strengths and Weaknesses for Biologists to Consider, (2021) doi:10.3389/fpls.2021.657240.
[10]A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, T. R. Gingeras, STAR: ultrafast universal RNA-seq aligner., Bioinformatics (Oxford, England). 29 (2013) 15-21 doi:10.1093/bioinformatics/bts635.
[11]A. Dobin, T. R. Gingeras, Mapping RNA-seq Reads with STAR., Current protocols in bioinformatics. 51 (2015) 11.14.1-11.14.19 doi:10.1002/0471250953.bi1114s51.
[12]Y. Tang et al., Bromocriptine inhibits proliferation in the endometrium from women with adenomyosis, (2023) doi:10.3389/fendo.2023.1026168.
[13]C. Frisendahl et al., miR-193b-5p and miR-374b-5p Are Aberrantly Expressed in Endometriosis and Suppress Endometrial Cell Migration In Vitro, (2024) doi:10.3390/biom14111400.
[14]K. A. Brücksken et al., Musashi inhibitor Ro 08-2750 attenuates triple-negative breast cancer cell proliferation and migration and acts as a novel chemo- and radiosensitizer., Biomedicine & pharmacotherapy = Biomedecine & pharmacotherapie. 186 (2025) 118002 doi:10.1016/j.biopha.2025.118002.
[15]{R Core Team}, R: A Language and Environment for Statistical Computing, (2021) doi:https://www.R-project.org/.
[16]H. Li et al., The Sequence Alignment/Map format and SAMtools., Bioinformatics (Oxford, England). 25 (2009) 2078-2079 doi:10.1093/bioinformatics/btp352.
[17]P. Danecek et al., Twelve years of SAMtools and BCFtools., GigaScience. 10 (2021) doi:10.1093/gigascience/giab008.
[18]Y. Du, Q. Huang, C. Arisdakessian, L. X. Garmire, Evaluation of STAR and Kallisto on Single Cell RNA-Seq Data Alignment., G3 (Bethesda, Md.). 10 (2020) 1775-1783 doi:10.1534/g3.120.401160.
[19]Y. Luo et al., New developments on the Encyclopedia of DNA Elements (ENCODE) data portal, Nucleic Acids Res.. 48 (2020) D882-D889 doi:10.1093/NAR/GKZ1062.
[20]B. Langmead, L. Salzberg Steven, Fast gapped-read alignment with Bowtie 2, Nature Methods. 9 (2012) 357-359 doi:10.1038/nmeth.1923.
[21]B. Langmead, C. Wilks, V. Antonescu, R. Charles, Scaling read aligners to hundreds of threads on general-purpose processors, Bioinform.. 35 (2019) 421-432 doi:10.1093/BIOINFORMATICS/BTY648.
[22]B. Bushnell, J. Rood, S. Esther, BBMerge – Accurate paired shotgun read merging via overlap., PLoS ONE. 12 (2017) e0185056 doi:10.1371/journal.pone.0185056.
[23]S. Andrews, F. Krueger, A. Segonds-Pichon, L. Biggins, C. Krueger, S. Wingett, FastQC, (2012)
[24]S. Chen, Y. Zhou, Y. Chen, J. Gu, fastp: an ultra-fast all-in-one FASTQ preprocessor, Bioinform.. 34 (2018) i884-i890 doi:10.1093/BIOINFORMATICS/BTY560.
[25]S. Chen, Ultrafast one-pass FASTQ data preprocessing, quality control, and deduplication using fastp, iMeta. 2 (2023) e107 doi:10.1002/imt2.107.
[26]H. Dai, Y. Guan, Nubeam-dedup: a fast and RAM-efficient tool to de-duplicate sequencing reads without mapping, Bioinform.. 36 (2020) 3254-3256 doi:10.1093/BIOINFORMATICS/BTAA112.
[27]T. Smith, A. Heger, I. Sudbery, UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy., Genome research. 27 (2017) 491-499 doi:10.1101/gr.209601.116.
[28]R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, C. Kingsford, Salmon provides fast and bias-aware quantification of transcript expression., Nature methods. 14 (2017) 417-419 doi:10.1038/nmeth.4197.
[29]N. L. Bray, H. Pimentel, P. Melsted, L. Pachter, Near-optimal probabilistic RNA-seq quantification., Nature biotechnology. 34 (2016) 525-527 doi:10.1038/nbt.3519.
[30]K. Okonechnikov, A. Conesa, F. García-Alcalde, Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data., Bioinformatics (Oxford, England). 32 (2016) 292-294 doi:10.1093/bioinformatics/btv566.
[31]G. Pertea, M. Pertea, GFF Utilities: GffRead and GffCompare, F1000Research. 9 (2020) 304 doi:10.12688/f1000research.23297.2.
[32]M. I. Love, W. Huber, S. Anders, Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2, (2014) doi:10.1186/s13059-014-0550-8.