Title

Multi-alignment framework - Consider and analyse genomic and transcriptomic alignment variability and read count distributions by comparing different techniques.

Authors
Xinwei Zhao, Eberhard Korsching*

Affiliations
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 presented multi-alignment framework (MAF) provides researchers with 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 based on Linux the major platform in Bioinformatics. If the Bash script language and Linux knowledge is not a hurdle development time could be saved by the user with the provided materials like the Bash script templates and the detailed installation information for the tools. The script structure enables the integration of all the processing steps and saves time if procedures need to be redone a lot of times with different parameter combinations or many different data sets. Because of the script base every user can modify the given templates to its own needs. The exemplary and comprehensive microRNA analysis exemplifies the value of looking into the details of alignment and quantification procedures and therefore gives insight into how the quality of the result candidates can be assessed and presumably false positives excluded.

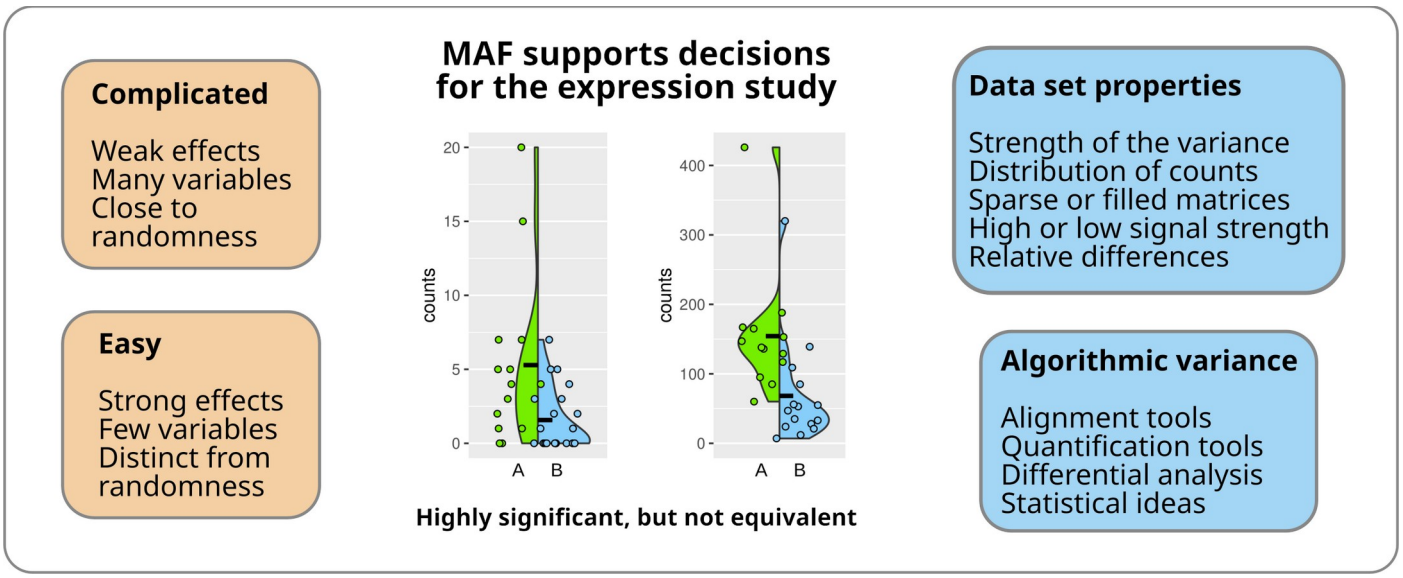
Conclusion

The three provided templates offer a structure to combine in a flexible way pre- and post processing steps, and are inherently extendable and completely redesignable to support many sequence analysis scenarios. So this approach is suitable for scientists which really want to look down to the basics of their alignment results and expect to be flexible in every respect of their results. The microRNA template shows a complex real world application backed by three publications.

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, while algorithms and concepts are constantly being adjusted [1] . As early as 1982, this alignment approach was complemented by de novo sequence assembly approaches and programs. [2] .

The alignment focuses primarily on finding merely perfect correspondence between sequences of interest but also tries to find vague conserved sequence regions between very unrelated biological species and their DNA and RNA sequence material. The currently established developments include the widely used BLAST algorithm (basic local alignment search tool) [3] or BLAT [4, 5] or the FASTA (sequence comparison) algorithm [6, 7] , which find "local or global similarity" between sequences implementing the before mentioned criteria. Both tools work on protein or DNA sequences and are able to translate between these types.

Beyond these core tools there are many other tools especially alignment programs currently available, and a recent review centering on short read alignment programs is summarizing current developments in the fields of sequence alignment tools, sequencing technologies and their core properties [8] . A critical examination of differences between alignment programs is given in Musich et al. [9] . The latter discusses different alignment and assembly algorithms in terms of their strengths and weaknesses and shows slightly to significantly different results, so that a more detailed treatment of these situations is necessary. Among other things, the handling of gaps is also something that 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 guaranty 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 which are called from within the scripts are already well establish by the science community. They are taking the fastq based data, doing quality control, trimming adapters or optional adjust the reads for other sequence features, optional deduplicate based on read sequence similarities, align to genomic or even transcriptomic references, and finally optional deduplicate bam file hosted reads by the umi barcode information. All these steps are well described below. So the whole system is open source based.

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

In this sense, the bioinformatics pipeline can be seen as a ready-to-use application concerning the three provided Bash scripts together with the given human references and the supplied program specific references. The three Bash scripts enable directly the alignment and quality control of mRNA and microRNA data sets in every size and number obviously depending on the computing resources. The in house experiments used 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 main memory.

However, the module structure of the scripts allows the 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 keeping the opportunity to use newer versions of the program, or to exchange program versions due to bugs e.g. after operating system updates, the containerized approach isn't ideal for a complex framework. Despite there might be license issues by the programs, the chance to solve 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.

Additionally, the simple Bash script setup - a certain task a certain script - was preferred over a universal Bash script with configuration files because of the complexity of the latter approach.

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

Table 1

Terms or 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 - needs 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 - which 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 specifically text files either with a graphical surface 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 are programs like 'grep'. Bam files need 'samtools view'.

Implementation

Overview

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

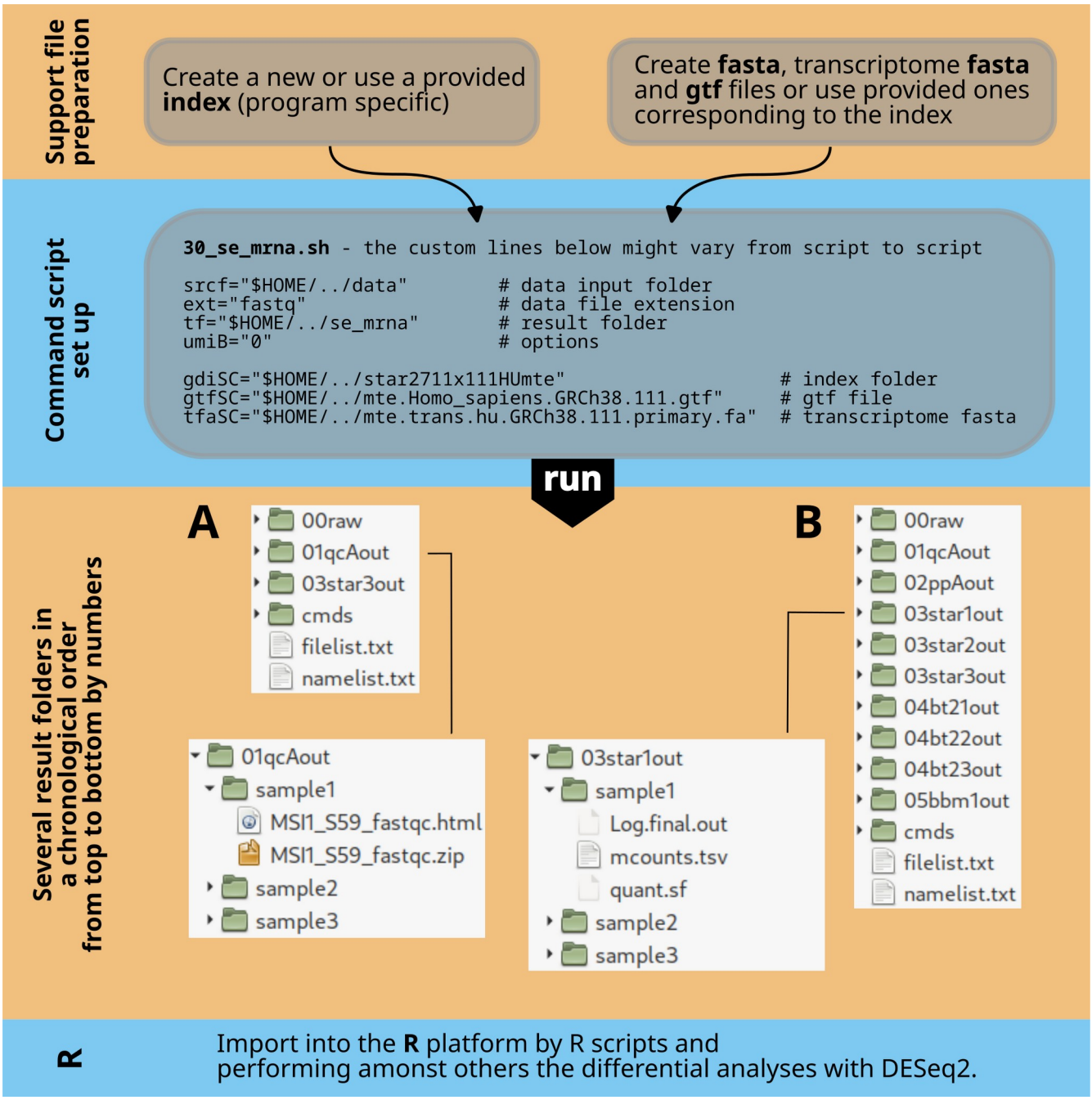


Figure 1. Overall workflow.
The processing architecture for estimating the impact of preprocessing tools and different alignment tools on the quantification results. The first step on top is mainly on organizing or creating all the necessary files to apply the alignment programs. The second step is focusing on the specific Bash script

which is joining all the different programs together. Here mainly path information to files need to be customized. Some program specific options need to be adjusted in the program call lines. The third step gives insight into the data structure created by the Bash script. The result folders are numbered in processing order and names are pointing to the tools behind them. (A) is showning this for script 30_se_mrna.sh and 30_pe_mrna.sh and (B) for 30_se_mir.sh. The analysis in R is the fourth step.

Prerequisites

The computing platform needs to be a Linux operating system. All our examples are tested on a Debian 12 system but should be not really limited to a specific Linux version of 2024 and above because of Bash and standard libraries. The program versions itself 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, an 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 in this form fully functional.

The most advanced example for transcriptomic microRNA or small RNA data (30_se_mir.sh, single end) is composed of three aligners and seven pre- and post-processing tools and some further utilities to generate a multiperspective view on the data alignment and finally on the results of the various analyses. All these 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 leverage the time to get started. Many of the necessary files are supplied as a download option because of the file size.

Inside of each command script the parameter section at the top needs to be adjusted for some user specific run parameters and is ready to be started if the supplementary file MAF.zip is unpacked inside the user's home folder and all the large files are downloaded and unpacked in the appropriate subfolders of the MAF.zip folder structure (Figure 2) descibed by readme.txt files inside those folders which are not filled. Obviously, all the relevant programs need to be accessible and executable by the script commands. A comprehensive startup guidance for this primary procedures is given at the top level of the MAF.zip folder structure ('install.txt'). If not our program versions on our site should be used the file 'readme.txt' on the top level supplies the download locations for all the programs.

Additionally we provide some 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 at the moment empty folders with certain files (Figure 2). This step is necessary for starting with the provided resources but if already 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 supplementary file install.txt in MAF at section: # Libraries #. The following section is intended as an overview on the utilized programs.

The sequence tools

Additionally the tools themselves need to be installed. For this purpose and to ease the complex work we have provided the present versions of the tools on the download page (see at the end section: 'Supplementary material & information'). 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.

In the following list the tools are shortly introduced with a focus on the 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 the quantification.

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

Bowtie2 - sequence aligner. The recommended publications are [20, 21] . This program is mainly adopted by the genome community despite also transcriptome applications are described - can handle gaps in version 2.

BBTools - sequence aligner. The tool collection is only partly published [22] and detailed information is available via links in the main readme.txt file - here BBMap is used and performs less perfect in the small RNA situation.

FastQC - fastq read file quality control. Before any sequence alignment is started some type of quality control of the sample(s) is an indispensable first step to see how homogeneous multiple samples behave and which specificities each sample own. FastQC [23] is in this respect a good choice out of the many other available tools and moreover it is scriptable.

fastp tool - fastq file preprocessing, quality control, and deduplication. In our pipeline the fastp tool [24, 25] is used only for preprocessing purpose : deletion of adapters, cut and move umi codes to the sam-format 'qname' variable and remove reads with length smaller than 16 bases. In a second run trimming of the two CA bases at the start of the read and again removing reads with length smaller than 16 bases.

nubeam-dedup tool - de-duplicate sequencing reads without reference genome. This approach is an option if no umi codes are used and even no perfect reference is available. Nubeam-dedup [26] as a pure C++ tool it is fast.

umi_tools - improve the quantification accuracy by umi code based deduplication. The umi_tools [27] are a Python tool set which has 6 major program features where we are using only the "dedup" part.

Because it is Python we recommend to install it via pipx as a self executable program described in detail in supplementary file install.txt.

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

QualiMap - transcript quantification. Only the quantification feature is used in the context of this straight program [29] developed with biology in mind.

GffRead - not only performing a genome- to transcriptome fasta modification. GffRead together with GffCompare have a broad functional scope [30] . The tool is here used to create an adjusted fasta file necessary for the quantification with the Salmon tool in the "quant mode" together with the transcriptomic bam files of STAR (option: --quantMode TranscriptomeSAM).

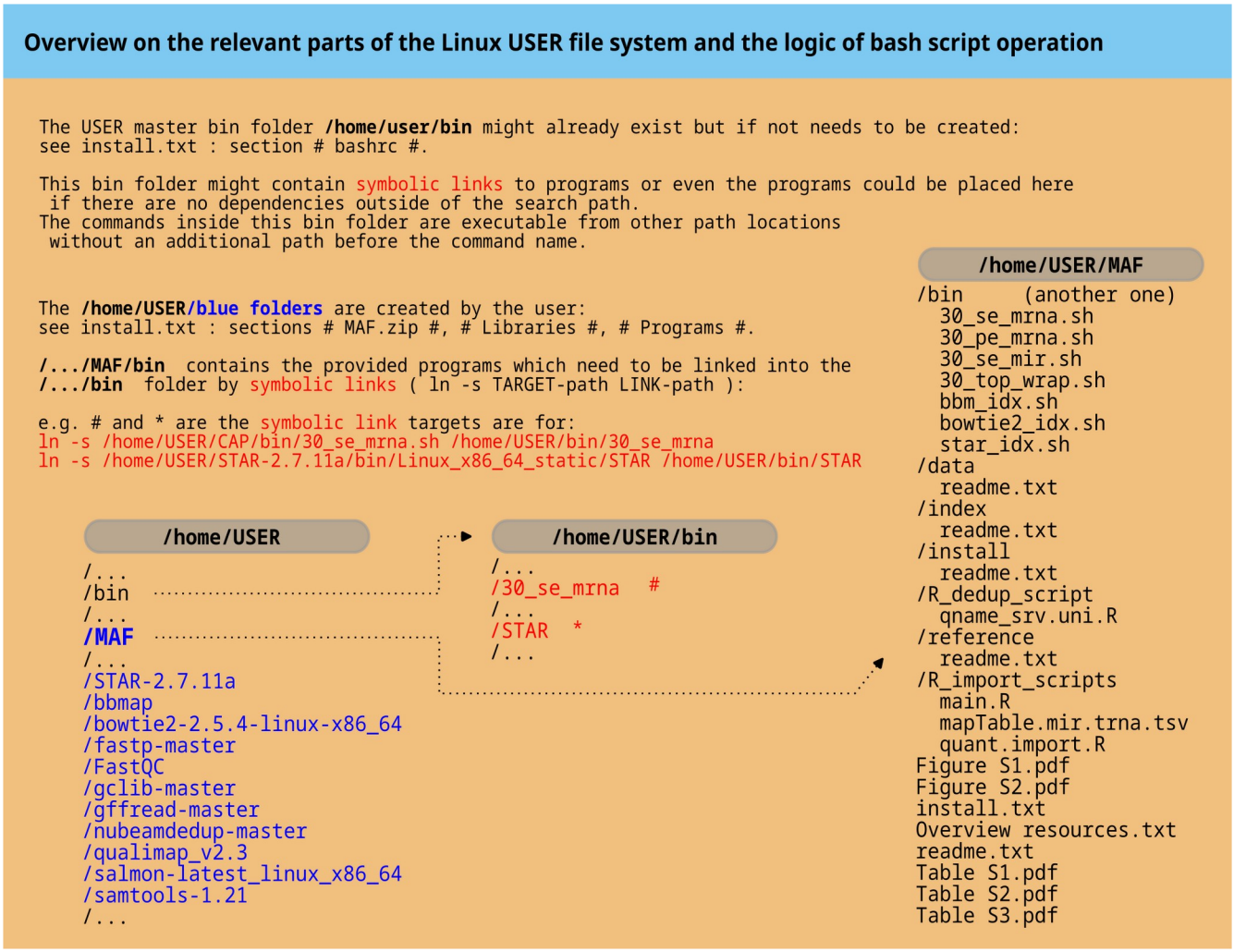


Figure 2. The master plan for folders.
The left part of the scheme is describing systems folders like the 'home/USER' path which exists if a standard Linux installation is successfully finished. The name USER needs to be changed to the real user name. Blue is indicating the new created folders by unpacking the MAF.zip file and after installing all the programs. The 'home/USER/bin' path is containing the symbolic links to the Bash command

scripts in MAF/bin as well as the symbolic links to the installed program executables. In the far right part the folder structure of the unpacked supplementary MAF.zip file is shown. Here also additional information files are placed (e.g. Figure S1).

Description of the main command scripts

The core element combining all the different programs and data formats together are as described above in our case three Bash scripts or to be more precise three different examples for such a type of command file. They are 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 if there is the need to combine several different tasks together and run them from this top level file all together over e.g. several days. Additionally index creation scripts for all the three alignment program specific indices are provided here for the case that other than the provided references are to be used. For this part additional information can be found in the alignment programs' help files.

Obviously this wrapper script but also the other Bash files might need some customization to fit for the user's computing environment. The customization takes place above all in the beginning of the scripts but if the lines which call the programs need to be adjusted for their parameters those lines need to be identified and adjusted. The script names include numbers which are the present version number of the scripts.

The first and second file are simple examples for doing a single end and a paired end mRNA analysis by using the STAR alignment program and offering some postprocessing support for deduplification. Viewing the script files is done by a so called 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 which has some text coloring scheme for the Bash script language is helping to read 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 now with an editor into the code for the single end alignment script '30_se_mrna.sh' the code is starting with some comment lines denoted by a hash character '#' preceding a comment text and therefore is not an active command. These text lines normally contain helpful information for the user or explain shortly the objective of the command line(s).

The 'parameter' section at the beginning of the script files are the most important lines to run the scripts. In a user specific computing environment it might need some adjustment:

```
#### 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 getting a quoted path to the fastq data folder. "$HOME" is a systems variable which is containing the users home path - but in any case you can also write it literally. **ext** is

the variable which is containing the file extension of the data files without the dot, normally 'fastq' but sometimes also 'fq' or something else. Important is not the extension but that the file itself is from a format type 'fastq'.

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 right. 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 need some extra efforts so that all programs are able to work with such a file type. Therefore, stick for the first experiments with a pure unpacked fastq file (command: **`gunzip name.fastq.gz`**).

The **tf** parameter is containing a path to a result folder. This folder might already exist but does not need to exist. 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' says no deduplification is done. '1' is calling a simple R script to unify all different umi codes while '2' is using the samtools markdup feature and '4' (recommended) is using the well published umi_tools program to capture the PCR amplification bias. This function is applied after the alignment and before the quantification starts. In the case of the R script obviously an R standard installation needs to be performed.

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

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

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

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

Module 1, internal name 'icreate', is creating the result file folder or is deleting (!) and re-creating it when it is already existing. Additionally, all the 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 in their code some differences to cope with the situation of one fastq file in single end versus two fastq files in the paired end situation (there might also be interleaved files - but they are not considered here).

The **Module 2** - FastQC - internal name 'qualcon', is generating a html file and a corresponding zip file which can be opened by a web browser. The file is giving a deep insight into the raw data quality for one raw data file. So every raw data file has a quality control file pair in a separate folder with the name 'sample' plus a number (the order of the sample names is preserved in filelist.txt).

The file system structure is in this case: /result folder /01qcAout /sample1 and here the file names sample1_fastqc.html and sample1_fastqc.zip. This type of structure is repeated for every main module and task. This structure and the result file namelist.txt is also used by the R import functions to generate a data structure inside R with the original names. So major changes in the bash code probably trigger also 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 few to several lines of code.

As can be seen here and in the other Bash functions that the code lines trigger the creation of a command line file in this case called 'sub1.sh' which at the end finally is executed by 'cmds/sub1.sh'. So the whole script is generating 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 give some insight into the situation.

Module 4 - STAR alignment (name: staroutC) - is combining several tasks together. The STAR command line itself with all the parameters, some samtools actions, samttools idxstats counting, Salmon counting, 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.

Important in the code is the usage of the **xargs** command. The ls command is creating a sample file list piping it into the xargs command which is generating with the template command line written after xargs for every input element an output command line which is saved into a Bash script file. 'echo' in this context is the command to create out of text and variables a real text line which can be written to the file. This file is finally executed. '>'denote save and overwrite while '>>' means save and append to the file.

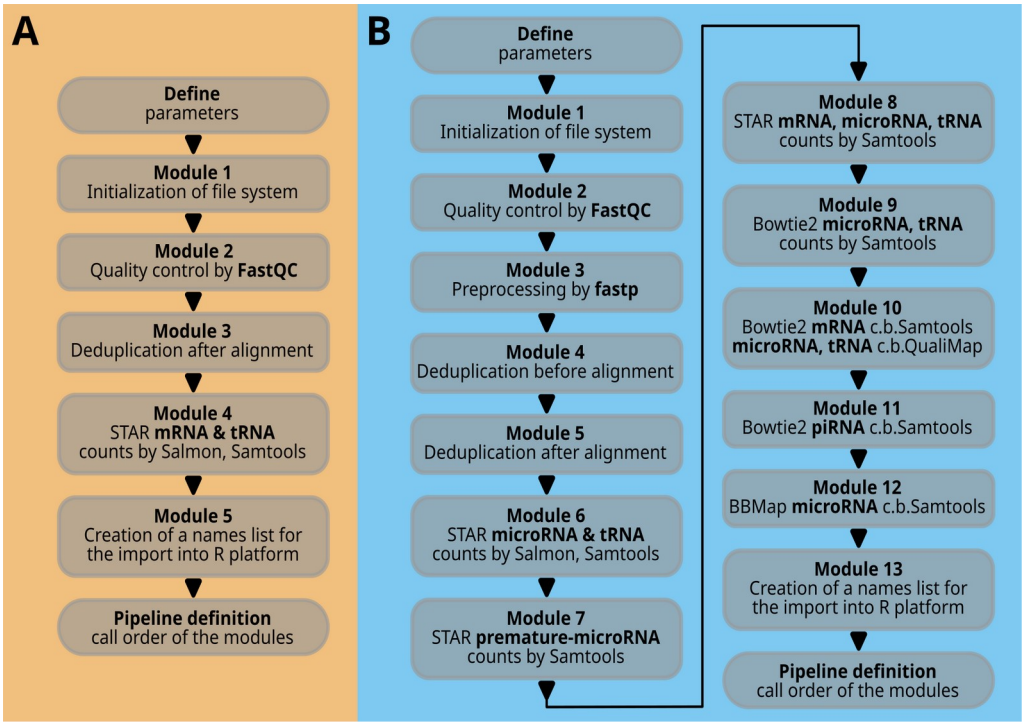


Figure 3. Code overview. In part (A) the modules of the single end or paired end workflow are listed. In part (B) the comprehensive microRNA workflow is presented which is composed of much more modules. A more detailed description of all the modules is given in the text.

Module 5 - named 'fnlA' - is extracting from the filelist.txt statistic file the pure name list with the right order which is necessary to translate back the 'sample1' nomenclature into the real sample name from the beginning. Here the cut and tr commands are used to extract a certain name element out of the input text file.

The last section in the Bash command script is executing all the single Bash functions, which we called modules, in a certain order, separated with some 'echo' feedback lines visible in the terminal to see which modules are successfully executed. The 'date' command is used to estimate the program 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. But 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 is starting here with a very long section of parameters needed to run this script. The parameters are following 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 to identical to above described files 30_se_mrna.sh and 30_pe_mrna.sh. A new insertion here is **Module 3** (name 'preprocess'). This module is using the program fastp to prune adapter sequences, to move umi identifier to the 'qname', to trim the CA prefix and to remove reads smaller than 16 bases. This module has a single end setup. Additionally, the program does also create another set of quality control files in its folder which can be viewed by a web browser (fp1- and fp2.html and .json).

Also new is **Module 4** ('seqcodeA'). This module is specialized on tasks which need to deduplicate sequences on a pure sequence similarity basis - so a non barcode based idea while Module 5 (Figure 3B) is corresponding with Module 3 in Figure 3A which is based on barcoding ideas.

Module 6,7,8 ('staroutA','staroutB','staroutC') are all STAR aligner based modules. The first approach is focusing on microRNAs and tRNAs, the second on premature microRNAs and the third additionally to micro- and tRNAs also on mRNAs. The latter is sometimes useful to see whether mRNA sequences share sequence similarities with small RNAs and therefore produce ambiguous counts.

Module 9,10,11 ('bt2outA','bt2outB','bt2outC') are Bowtie2 based and create 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 on the count distribution of transcriptome features across genomic master elements. The third approach is specifically looking on the piRNAs which form a second small RNA domain and only very few elements are showing overlaps with microRNAs (maybe bugs in the database - was identified by the blastn tool and solved towards the microRNAs before we created our custom references).

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

The last two modules are again 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
+
III9IIIIIIIIIIIII9IIIIIIIIIIIIII9III9IIIII9IIIIIIIIIIIII9IIIIIIIIIII-II
```

The two different examples show a typical block of 4 lines for one 'read'. First line the unique identifier starting with a '@' having or not having split characters ':' and barcode sequences at the end. The second line comprises the sequence, the third line a '+' with or without identifier (standard) and the fourth line the quality sequence (Phred score: 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 even other compression extensions are possible. Here the recommendation is to decompress the files before running the scripts. But the script could be extended so that also gz compressed files are possible but maybe not for all tools. Nevertheless, because many programs are coming together and each program might have it's own file decompression handling it is more general to start with uncompressed files.

The 'fasta' files might 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 by 80 characters like in the example shown 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 is coming. The nano editor in the terminal might be able to handle such big files at least for viewing and searching but the main memory of the used computer might be the limiting factor.

The genome fasta file is necessary to generate the STAR, Bowtie2 and BBMap index data structures. A folder with several files in all cases. The files star_idx.sh, bowtie2_idx.sh and bbm_idx.sh are provided to help in creating new index files.

The STAR file is also including the call of the **gffread** tool to generate the transcriptome fasta file out of the primary genome fasta file. This is necessary for using the Salmon quantifier tool. The Samtools idxstats quantifier is not using such a reference. Important to note: The Samtools approach is counting but not normalizing count data according to paired end situations in contrast to Salmon which does. 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 files need to be converted to a GTF file) is describing features inside the pure fasta sequence file. The main information is the sequence start and end position (column 4, 5) and some additional annotation information, at the first place (column 1) the chromosome name which needs to correspond with the genome fasta chromosome name (note: could be written '1' or 'chr1' or even 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 obviously microRNA and tRNA information from the respective databases is inserted into the standard GTF files at the appropriate place and format which needs a lot of customization procedures but this is a whole topic on its own. The same is also done for the piRNAs. This needs to be done because the standard Ensemble and other main databases are still missing a lot of those features.

Sometime it is also necessary to shrink the features down to the targets of interest to reduce the run time remarkably. But have in mind that also the whole genome view will open the eyes for signal crosstalk between different features or often between different transcripts. This is one of the reasons why the microRNA script contains several different approaches and even whole transcriptome views.

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

Results

As described in the beginning the concept was applied in some published studies and is used in still running studies. In order to complement the published information in a more detailed and methodological manner, this section focuses on the stringency of the algorithmic results. This study is based on Ensemble version 111 with a microRNA and tRNA extension and sequence data mentioned in 'Availability of data and materials' section at the end.

The microRNA script was applied on human samples splitting up into five conditions and two technical control samples not part of the conditions. The research questions addressed here are a) to analyze the overall count distributions, the microRNA distribution and the proportions, to see if there are any problematic trends concerning the overall expression strength, and b) how the final significant candidates build their significance which in several cases would lead to a rejection after the examination. Both ideas give insight into the complexity of data and the challenges to find robust and sufficiently annotated candidates.

The application of the 30_se_mir.sh script is generating several alignment results on small RNAs and on further transcripts. Next the result structures are imported into R by some R scripts - preferably the provided ones in /MAF/R_import_scripts. Finally two differential analyses with the DESeq2 R package are performed and R graphs are created presenting the results. It is important to note that the DESeq2 procedures does change the raw data counts slightly because of the DESeq2 variance smoothing approach ([31] and some guide on 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 situation after the data import is shown. It can be seen that the majority of counts goes into RNA transcripts which are not part of the microRNA family. The used small RNA library preparation kit is filtering for sequence fragments not being greater than 100 bases. So there are obviously much more other RNA transcript families expressed be it as full molecules or fragments which can be captured by the alignment procedures.

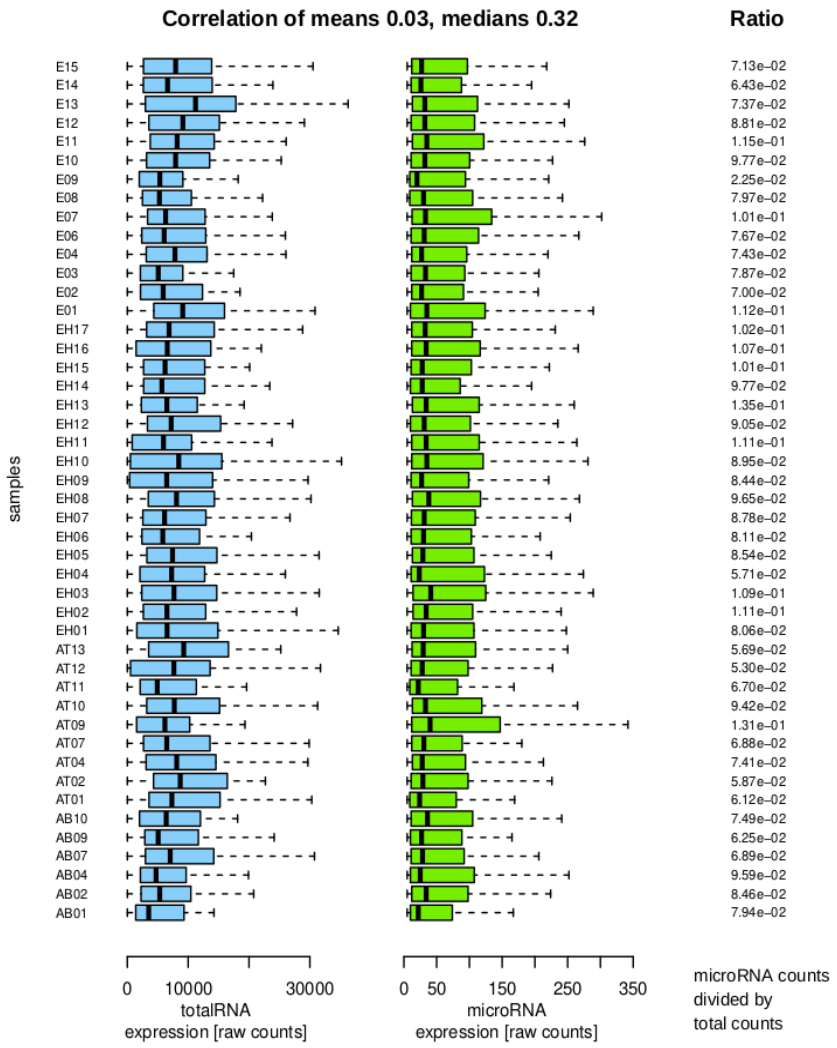


Figure 4. Sample distributions and properties
On the left the sample acronyms are given. The blue boxplots denote the total RNA count numbers while the green boxplots denote microRNA expression count numbers - as can be seen on the scale range much less counts are in this fraction. On the right the ratios between the total and the microRNA counts are given per sample. On top the correlation coefficient of the mean and the median values of the sample distributions is calculated between the total RNA group (blue) and the microRNA group (green).

Important for the differential analysis is if the total number of counts and the microRNA number of counts per sample are following an identical trend - which would void in a certain way differential results. The per sample mean value trend is showing a very weak positive correlation which is

indicating that there is no strong correspondence. The median correlation is as expected higher but less informative concerning the feature properties of the distribution. The features itself are much more heterogeneous and weakening the positive correlation. Looking on the ratios shows that the samples are not exposing an extreme heterogeneity in groups and between groups and therefore should be suited for a final differential analysis.

The differential analysis is mainly dependent on the group variance, the size of each group and the absolute count numbers per feature. Performing statistics on that level indeed is advised but not shown here explicitly. In the following graphs this topic is taken up again from a slightly different perspective. The Table 2 gives more insight into the final significant candidate numbers out of the pool of possible candidates which is also a benchmark number which can be addressed by a sampling procedure on the input data level. It is noticeable here that the coherence between the STAR and Bowti2 aligner is high despite some differences exist. BBMap in contrast is not able to deal in an appropriate way with the microRNAs in this data set (line five).

Table 2

Numbers 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 to 2 are denoting the two differential DESeq2 analyses considered here, with multiple testing correction according to Benjamini-Hochberg. The 'of' is the input number of not absolutely negative expression features for the differential analysis. 'mt' stands for microRNA and tRNA while 'whole index' means all possible expression features. 'pmm' denotes the pre-mature microRNA approach. See

additionally the Venn diagrams in Figure 5 and also Table S2 for a more detailed analysis between the five mature microRNA approaches.

The Figure 5 and supplementary Table S2 show how the established candidates overlap across the different technical approaches and which alignment procedures lead to certain candidates. Even the two different quantification algorithms based on the same STAR approach with leads to some differences in the results in the weak cases only concerning the associated p values. More relevant differences can be observed between the STAR and Bowtie2 approaches and the Bowtie2 approach with the QualiMap based quantification. There significant candidates go and new ones are getting in the result lists.

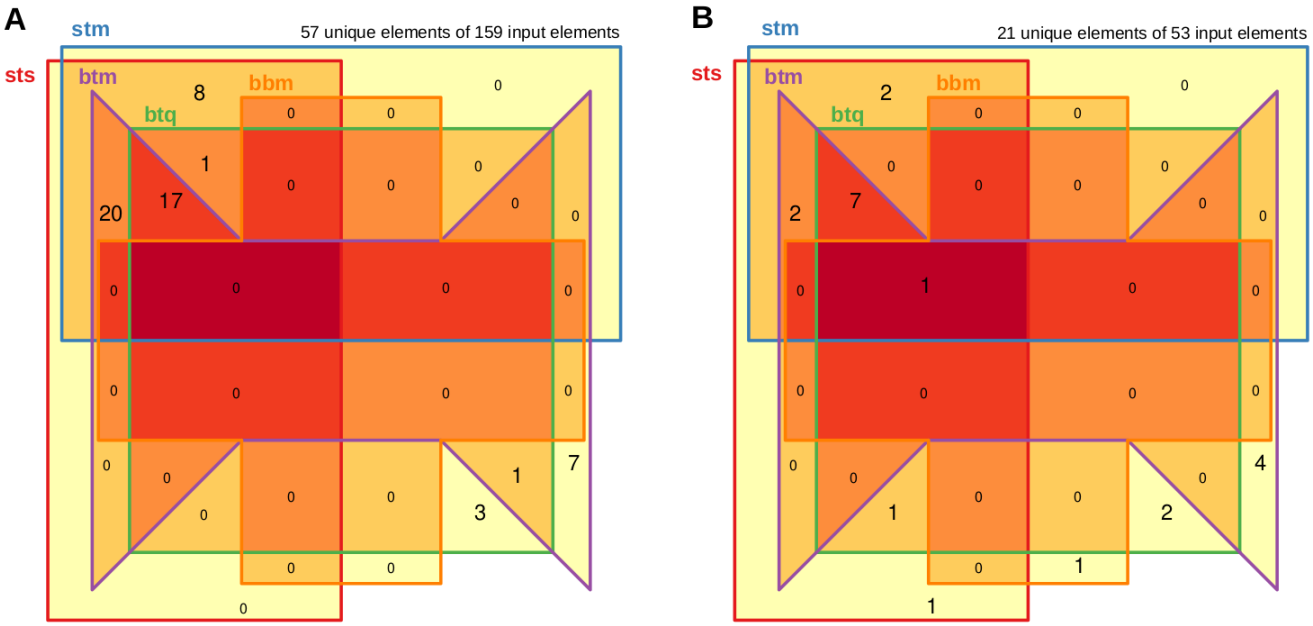


Figure 5. Overlap of significant candidates between the 5 algorithmic approaches.
Panel A is showing differential analyses DE-1 and panel B is showing differential analyses DE-2 focusing on the five mature microRNA approaches and their overlap between significant candidates. Darkest red color represents overlap of all sets yellow means unique for one group. The nomenclature of the five analyses 'sts' to 'bbm' is explained in Table 2 in the right column. The top text line is denoting the total number of unique features in comparison to all features of all five groups which are the basis for this Venn diagram. The unique features might be distributed unevenly between the groups.

The most distinct program is BBMap and with this type of small RNA data eventually not really useful. In the first differential analysis it can not find any small RNA while in the second differential analysis the most prominent one but only this one is confirmed and a highly significant one is newly introduced. Taking a look into 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 G and again a stretch of 10 A bases in the beginning. This might point to some weakness of BBMap algorithm to handle such type 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 show that the overlap of the significant premature microRNAs to the mature microRNAs with STAR or Bowtie 2 are between 24% to 33% for the first differential set and between 34% to 41% in the second differential set

(supplementary Figure S1). The main reason is that the premature analysis is biologically a different state as the mature microRNA forms which represent the final regulatory molecules. Additionally, there is possibly a crosstalk between both forms in that fragments of the premature form account for the mature forms. Therefore the alignment procedures might not be efficient to discriminate between premature and mature forms. Nevertheless, the premature microRNA results are supporting the mature microRNA results substantially in this case despite regulatory conclusions might be weaker.

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

Finally it is important to look on the count distribution per feature. A lot of highly significant microRNA candidates are shiny by their p values but less convincing in how the significance is created by the distribution of individual count values per group. The top candidates out of the two differential analyses are given in Figure 6.

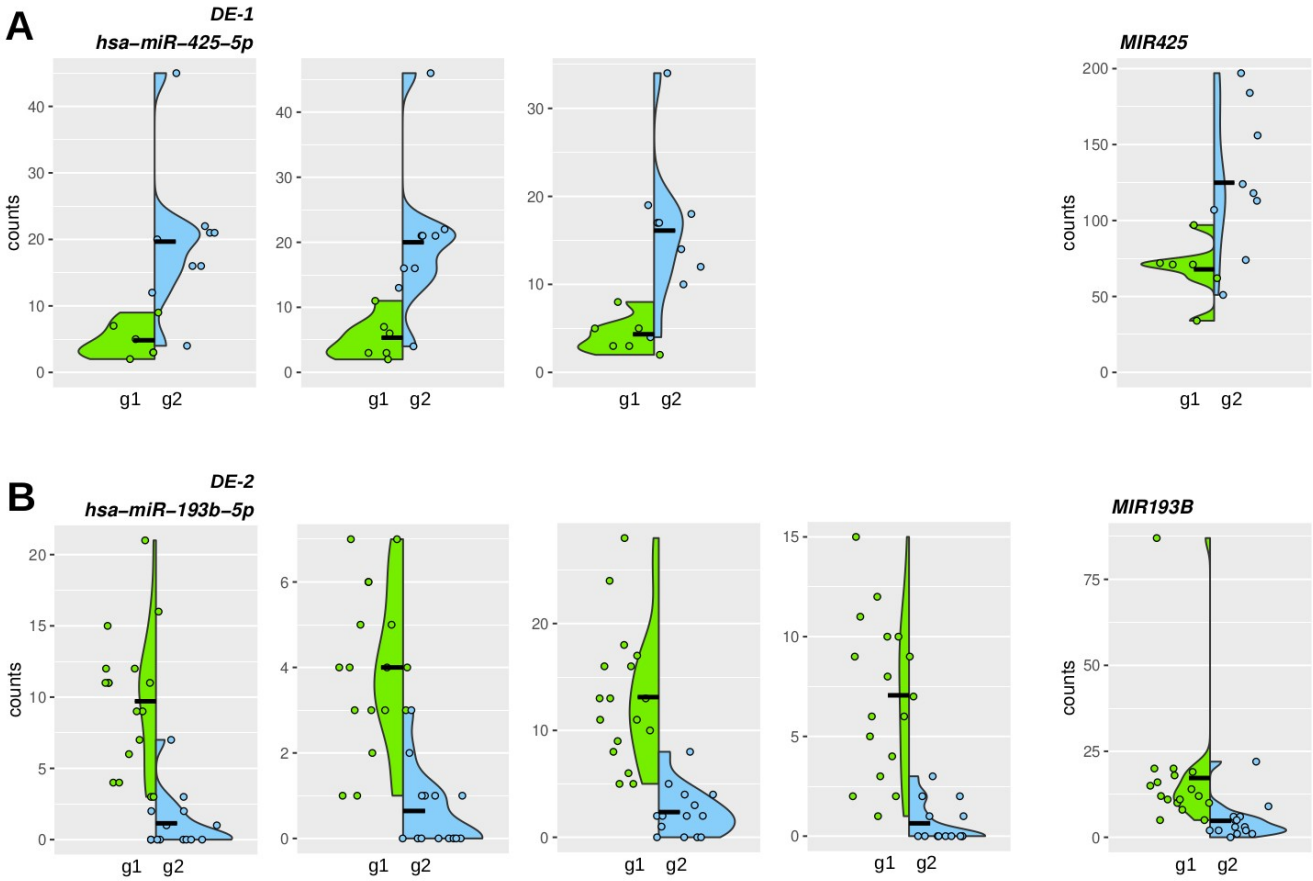


Figure 6. Group distributions of all the individual count values.
The count values in this graph are presented as a violin plot with all the individual count values denoted by small dots filled with the respective group color. Because many count values in a low count situation

have the same value these values are randomly scatter horizontally - so these differences does 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 to the premature forms denoted on the right side by uppercase 'MIR' identifier. The count number scale of the premature forms is as expected slightly to remarkably higher compared to the mature microRNA forms. Important to note that the higher count number does not really affect the outcome of the differential situation. To get a broader insight into the wide variances possibly happening behind those features *with* significant p values further results are presented in supplementary Figure S2 for microRNAs, tRNAs, piRNAs and the premature microRNAs.

Conclusion

The multi-alignment framework MAF and the given protocol allows researchers to understand in a better way the complex influences of technology and software tools on their results and allows to select finally robust candidates. The investment of time is reasonable because a lot of work is already invested in a basic framework, customized indices and references at least for the microRNA situation.

For microRNAs BBMap alignment program might be less useful than STAR and Bowtie2. Overall STAR seems to be most stable procedure. Nevetheless, all procedures are having their specialties justifying that comparison should be done for every dataset under investigation. The differential analysis is exposing in the count distributions clearly that significance is often not the best measure for relevance.

Furtheron the MAF approach is guiding the skills to acquire more advanced bioinformatics knowledge. It is flexible and based on open source ideas and enables researcher to focus on biology while by a multiperspective view explore the strength and limitations of computational methods and their impact on the interpretation of sequence experiments and their results.

Availability and requirements

Project name: MAF

Project home page: https://complex-systems.uni-muenster.de/sinfo.html

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 download location.

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. Due to file size reasons and as a courtesy all the custom versions of the reference- and index files, the two test data files and also the third party programs, belonging to this MAF release are available via https://complex-systems.uni-muenster.de/sinfo.html and need to be placed after the download in the respective template folders available in the MAF.zip folder structure after unpacking the MAF.zip file. 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 in general be exchanged according to the demands of the user but might need extra efforts by the user. So the MAF.zip is in a certain way a functional starter kit which enables the user to realize his own 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]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.

[30]G. Pertea, M. Pertea, GFF Utilities: GffRead and GffCompare, F1000Research. 9 (2020) 304 doi:10.12688/f1000research.23297.2.
[31]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.