# Challenges for the development of automated RNA-seq analyses pipelines

## Herausforderungen bei der Entwicklung automatisierter RNA-seq Analyse-Pipelines

**Matthieu Beukers**[1,2]
**Jens Allmer**[1,3]

1 Applied Bioinformatics, Bioscience, Wageningen University & Research, Wageningen, Netherlands

2 Department of Genetics, University of Groningen, University Medical Center Groningen, Groningen, Netherlands

3 Medical Informatics and Bioinformatics, Hochschule Ruhr West (University of Applied Sciences), Mülheim an der Ruhr, Germany

## Abstract

**Background:** Transcriptional changes are hallmarks of development and disease. RNA sequencing (RNA-seq) allows qualitative and quantitative RNA expression analysis. Raw RNA-seq data passes through a multi-step computational pipeline to derive meaning from such measurements. Often *ad hoc* scripts are used for such analyses. However, the use of workflow management systems (WFMS) should be encouraged in order to enhance result reproducibility, to establish best data analysis practices, and to share such data analysis workflows. In this work, we created RNA-seq data analysis workflows in three WFMS, namely Galaxy (free, open-source), KNIME (free, commercial, and partially open source), and CLC (commercial, closed source).
**Methods:** These tools were compared using a variety of criteria ranging from installation to workflow execution and sharing. Four different workflows (WFs) performing RNA-seq data analysis were successfully constructed in all three WFMS. In summary, Galaxy currently provides the most significant number of analysis tools for RNA-seq, while CLC offers the most intuitive visualization. KNIME lags behind in these two aspects but excels at other levels, such as machine learning.
**Results:** Since we already decided on the three WMFS, many of the criteria we suggest for WFMS evaluation do not apply to our situation and we focus on the WF creation here. While it was possible to construct RNA-seq analysis WFs with all three WFMS tools, the constructed WFs are different. These differences entailed disparate results, which were further sensitive to processing settings leading to different biological interpretations in the worst case. We further performed an in-depth analysis of challenges using the three WFMS and provide decision support for which WFMS to use in RNA-seq analysis. In short, RNA-seq is currently best performed using Galaxy, followed by CLC, and KNIME. The level of expertise with these WFMS should be taken into account during the WFMS selection. Finally, we share the WFs in the hope of reducing the use of *ad hoc* scripts and that sharing them will lead to the development of best practices for RNA-seq data analysis.

**Keywords:** RNA sequencing, RNA-seq, data analysis workflow, workflow management system

## Zusammenfassung

**Hintergrund:** Transkriptionelle Veränderungen sind Kennzeichen von Entwicklung und Krankheit. RNA-Sequenzierung (RNA-seq) ermöglicht die qualitative und quantitative Analyse der RNA-Expression. Rohdaten von RNA-seq durchlaufen typischerweise eine mehrstufige, computergestützte Pipeline, um aus solchen Messungen eine Bedeutung abzuleiten. Oft werden dafür Ad-hoc-Skripte verwendet. Allerdings sollte die Verwendung von Workflow-Management-Systemen (WFMS) gefördert werden, um die Reproduzierbarkeit von Ergebnissen zu verbessern, bewährte Datenanalyseverfahren zu etablieren und solche Workflows zur Datenanalyse zu teilen. In dieser Arbeit haben wir RNA-seq Datenanalyse-Workflows in drei WFMS erstellt, namentlich: Galaxy (kostenlos, Open Source), KNIME (kostenlos, kommerziell und teilweise Open Source) und CLC (kommerziell, Closed Source).

**Methoden:** Diese Werkzeuge wurden anhand einer Vielzahl von Kriterien verglichen, von der Installation bis zur Ausführung und Freigabe von Workflows. Vier verschiedene Workflows zur RNA-seq Datenanalyse wurden in allen drei WFMS erfolgreich erstellt. Zusammenfassend bietet Galaxy derzeit die größte Anzahl an Analysetools für RNA-seq, während CLC die intuitivste Visualisierung bietet. KNIME hinkt in diesen beiden Aspekten hinterher, glänzt jedoch auf anderen Ebenen, wie z.B. dem maschinellen Lernen.

**Ergebnisse:** Da wir uns bereits auf die drei WFMS festgelegt haben, sind viele der von uns vorgeschlagenen Kriterien für die Bewertung von WFMS in unserer Situation nicht relevant, und wir konzentrieren uns hier auf die Erstellung von Workflows. Obwohl es mit allen drei WFMS möglich war, RNA-seq Analyse-Workflows zu erstellen, sind die erstellten Workflows unterschiedlich. Diese Unterschiede führten zu unterschiedlichen Ergebnissen, die bei unterschiedlichen Verarbeitungseinstellungen in schlechtesten Fällen zu unterschiedlichen biologischen Interpretationen führten. Wir haben zudem eine eingehende Analyse der Herausforderungen mit den drei WFMS durchgeführt und Entscheidungsunterstützung für die Auswahl des richtigen WFMS für RNA-seq Analysen bereitgestellt. Kurz gesagt, wird RNA-seq derzeit am besten mit Galaxy durchgeführt, gefolgt von CLC und KNIME. Der Kenntnisstand mit diesen WFMS sollte bei der Auswahl berücksichtigt werden. Wir teilen die Workflows, die wir erstellt haben, in der Hoffnung, den Einsatz von Ad-hoc-Skripten zu reduzieren und die Entwicklung bewährter Verfahren für die RNA-seq Datenanalyse zu fördern.

**Schlüsselwörter:** RNA-Sequenzierung, RNA-seq, Datenanalyse-Workflow, Workflow-Management-System

## Introduction

Diseases can often be tied to the dysregulation of gene expression, which can be experimentally quantified using RNA sequencing (RNA-seq) [1]. RNA-seq also enables the detection of novel alternative splicing variants, which was not possible with hybridization technologies [1]. While RNA-seq experiments can have various goals and protocols [2], one of the most common is the quantification of gene expression levels among different conditions as, for example, done for tomato under different watering regimes [3]. Raw RNA-seq data needs to be processed computationally and for that purpose passes through a multi-step computational analysis pipeline using bioinformatics tools [2], [4]. Spjuth et al. previously interviewed several research groups about their experiences with workflow management during the SeqAhead hackathon and found that often *ad hoc* scripts are developed to automate the overall workflow [4]. Such idiosyncratic works may be hard to understand, reuse, and may not reproduce the results in different environments. A slight improvement over such scripts is the use of more structured make-files, for example, SnakeMake [5]. Alternatively, workflow management systems (WFMS) can be used to automate such multi-step analyses. Workflows (WFs) created using such WFMS are aimed to automate and document multi-step analyses. WFs are instrumental in assuring reproducibility of analyses and are generally used in two different ways [4]. First, they are used for routine analyses, allowing the reuse of the analysis pipeline [4]. Second, WFs are employed for more exploratory analyses where the visual programming capacities of WFMS are leveraged [4]. Many WFMS exist and are in use in bioinformatics such as Taverna [6], Galaxy [7], KNIME [8], CLC Genomics Workbench, Pegasus [9], Conveyor [10], and many more. In general, a WF transforms input data via various interconnected processing steps (often referred to as nodes) into output data. The FAIR guiding principles describe the sharing of data, algorithms, and WFs in an attempt to improve research reproducibility and transparency [11]. Data, computational tools, and WFs should, therefore, be FAIR: findable, accessible, interoperable, and reusable [11]. Much has been done for FAIR data such as Dataverse [12], OpenPHACTS [13], and Zenodo (http://zenodo.org/). For WF sharing, the common workflow language (CWL) [14], SHIWA [15], and the common tool descriptor [16] are examples with the aim of making WFs interoperable and reusable among WFMS. Together with WF repositories such as MyExperiment [17], the FAIR guidelines for data can thus be emulated for WFs, at least in theory.

A small list of WFMS was selected for comparison based on their status of commercialization and use within the bioinformatics community. Another criterion was that they needed to provide a graphical user interface (GUI) with WF visualization facility so that they can be used to communicate the data analysis among stakeholders. The availability of a GUI also simplifies the application and development of WFs by the non-specialist. The three selected WFMS are Galaxy, free and open-source, KNIME, partially free and partially open source with added commercial products, and CLC, commercial and closed source. KNIME was developed at the Konstanz University and released in the mid-2000s. It allows users to create WFs via a GUI with strong visual programming capabilities [8]. Several packages and extensions have been produced to create bioinformatics WFs in KNIME, such as SeqAn [18] and KNIME4NGS [19]. Galaxy is a server application that was first released in 2005 to enable biologists to perform computational analysis using a web interface [7]. The application was designed to increase access to computational analysis, allow the creation of automated multi-step analyses, and provide transparent analyses [7]. Galaxy offers the platform as a software package as well as a public server containing many tools that can be used [7]. CLC Genomics Workbench allows researchers to construct and perform various multi-step NGS data analyses, and its main aim is user-friendliness. CLC Genomics Workbench is offered as a standalone program for a workstation, and its functionality can be enhanced in organizations with CLC Genomics Server. CLC provides its own implementations of bioinformatics tools, although other tools can be integrated.

We believe these three WFMS give a succinct representation of the large amount of WFMS available, and judging by their amount of mentions in title and abstracts of manuscripts listed in PubMed, they are also quite popular. In order to compare the WFMS, we chose a common task in bioinformatics: RNA-seq analysis. Among other differences, we investigated WF creation, tool availability, and WF flexibility from a user perspective. Different types of RNA-seq analysis are possible, and as a practical example, four different WFs for each WFMS for the identification of differentially expressed (DE) genes were constructed following the general analysis steps described in Conesa et al. [2]. While we had an interest in creating WFs for the analysis of alternative splicing, this was not possible with all three WFMS tools without implementing new nodes, which is beyond the perspective we take here. KNIME is missing processing nodes for that type of analysis, but it can be overcome with implementing custom nodes, as we showed in a study involving altORF identification [20]. Apart from this particular problem, other challenges in the field are that for each WFMS the availability of analysis tools differs and that even if the same tool is available, the versions might vary. Therefore, it is impossible to create the exact same WFs in multiple WFMS out of the box. We identified this as one of the major challenges for cross-WFMS reproducibility. To still create the exact same WFs in different WFMS, the tools need to be added to the various WFMS, which can be quite challenging and is beyond what a casual user of WFs can be expected to do. Our main aim was to provide a recommendation of which WFMS tool to use for RNA-seq analysis. However, it was possible to develop RNA-seq WFs with all three WFMS. Therefore, a general recommendation was hard to provide, and potential users should consider the limitations and challenges that were determined for each of these tools in this study. We outline these limitations and the challenges we faced in the results section and provide decision support for a number of use cases in the conclusion. Additionally, we created a list of criteria to take into consideration when newly embarking on an RNA-seq endevour and looking for a WFMS. These criteria are in no particular order and have no weights attached to them so that the reader can evaluate them in light of their existing expertise. Finally, we demonstrate that variations among the WFs lead to differing results from the same input data. In the worst case this may lead to contrasting biological interpretations. Often divergent results can be attributed to differences in algorithm of the employed tools, but if some tools are commercial black boxes, interpretation becomes difficult. Thus, we hope that from the WFs we developed in three WFMS and which we share on GitHub [21], others can develop them further to reach a consensus for RNA-seq data analysis.

# Materials and methods

## Test data

Test data provided with a test WF from the KNIME4NGS package [22] was used in all three WFMS. The data consisted of eight paired-read datasets (forward and reverse reads in separate files), the chromosome 16 reference sequence, and a GTF annotation file (https://github.com/MatthieuBeukers/RNA-seqFlow/blob/master/testdata/Homo_sapiens.GRCh37.75.chr16.gtf). All data were from the human genome build GRCh37. A design table to be used in the differential expression analysis was also provided (see https://github.com/MatthieuBeukers/RNA-seqFlow/blob/master/testdata/dea_design_table.tsv). Additionally, a reference transcriptome consisting of all cDNA sequences for the human build GRCh37.75 was downloaded from the Ensembl website (see https://github.com/MatthieuBeukers/RNA-seqFlow/blob/master/testdata/chr16_reference_transcriptome.fa). This data was filtered to retain only chromosome 16 cDNA sequences. FASTA definition lines were modified to keep only the transcript identifiers. The downloaded cDNA data were subsequently used to create a transcript to gene translation table (see https://github.com/MatthieuBeukers/RNA-seqFlow/blob/master/testdata/chr16_t2gene.csv). We performed a quality check of the RNA-seq files using FASTQC. All data were of high quality, and none of the files needed to be discarded due to

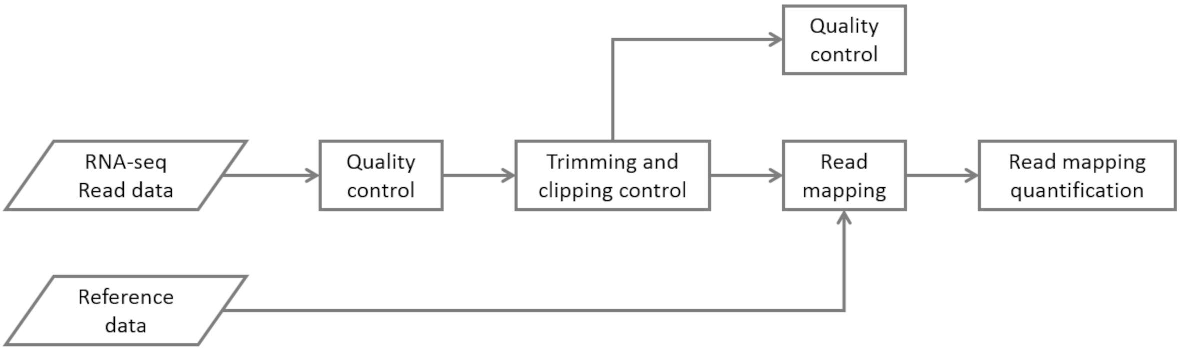

**Figure 1: General RNA-Seq WF design.**
WFs implemented in this work follow the general design based on the 'classical' RNA-seq pipeline described by Conesa et al. [2].

low quality (https://github.com/MatthieuBeukers/RNA-seqFlow/blob/master/testdata/QualityOVerview.pdf).

## Workflow management systems

KNIME 3.4.2 + with all free extensions (32bit) was installed in a 32bit Ubuntu (Ubuntu Mate 16.04) virtual machine running on Windows 7 Enterprise (64bit) in Virtualbox 5. Required programs and dependencies for the KNIME4NGS package were built from source (see Attachment 1).

Galaxy 18.01 was installed from an archive on a local cluster server and run once for configuration. Only one admin account was created after installation. All other settings were kept at their defaults.

CLC Genomics Workbench 11 was installed on a Windows 7 (64bit). A CLC license was obtained from the local CLC Server.

With this setup, we also stress tested the support of older operating systems with the selected WFMS. Please note that all three WFMS support modern operating systems.

## Workflow creation

Four different WFs were created in each WFMS. Two WFs were aimed at performing differential gene expression analysis (DEA) from read mappings to the transcriptome or genome, respectively. The other two WFs were aimed at mapping many datasets against either the genome or transcriptome, respectively, to test parallelization. Figure 1 displays the general design for RNA-seq analysis, which was implemented in the selected WFMS environments.

# Results

## Factors considered for WFMS recommendation

All three WFMS were tested on a single computer as a single user. Potential difficulties and strengths with multiple users and behavior in a cluster with multiple machines were, therefore, not assessed. Furthermore, capabilities and problems with parallelization, which can be of great importance in big data analysis, were not tested. Another aspect that was ignored is automated WF testing. Proper testing of WFs, whether connected tools work and obtained results are correct, is a vital aspect in WFs, as noted by Piras et al. [23]. They correctly suggest WFs should be tested rigorously like any software application and created the wft4galaxy application for testing Galaxy WFs [23]. KNIME, as well, offers several nodes to test WFs. In CLC, no specific tools were found for WF testing.

Many factors were compiled to support the recommendation for which WFMS to use under which conditions. Factors included are, for example, the availability of bioinformatics tools, options for WF branching, and managing input/output. Table 1 shows a selection of these factors, and Attachment 2 contains all (65) factors. Attachment 2 records all of the challenges we encountered while creating the RNA-seq WFs for this work and presents questions that need to be asked prior to creating WFs with the WFMS, such as whether the desired tools are available (Table 1, rows 2 and 3). Therefore, many of these questions could not be applied to this study since we chose the WFMS before compiling the list and we focus on the items that presented the largest problems during creation of the WFs.

## Workflow readability and flexibility

Regardless of the platform employed in this study, small WFs are easy to read and understand. However, with increasing complexity, WFs become more challenging to read in Galaxy and CLC. KNIME offers meta nodes, encapsulating smaller WFs, which improves readability. Galaxy can also create sub-WFs. However, there are a few caveats with them, such as that the tool settings cannot be changed globally. CLC has no option to encapsulate and reuse sub-WFs, and the WFs become harder to read with an increasing number of tools, inputs, and outputs (Figure 2). Readability decreases because each tool displays all possible in- and outputs. Larger versions of the

**Table 1: Factors considered during WMS comparison.**
Factors are coloured using a traffic light system with green generally indicating 'good'
(for the full version refer to Attachment 2).

| | KNIME | Galaxy | CLC Genomics Workbench |
|---|---|---|---|
| **Workflow flexibility** | • Two way branching with if switch<br>• Two way branching with if switch controlled by java code<br>• Three way branching with case switch | • Galaxy does not offer branching/decisions (is planned to be incorporated however) | • No options for branching or decision making |
| **Available mapping programs** | Bowtie, Bowtie2, BWA, Masai, RazerS, YaraMapper, Segemehl, Star | Bowtie, Bowtie2, TopHat, TopHat2, BWA, STAR, HISAT2, Segemehl, Mosaik2, rqrnastar | 'Map Reads to Reference' tool, 'RNA-Seq Analysis' tool |
| **Availability of DE Analysis programs** | DESeq, edgeR, limma | DESeq2, edgeR, limma, Cuffdiff | 'Differential Expression for RNA-Seq' tool |
| **Output production of files** | • KNIME4NGS saves output in same folder as input<br>• Other nodes offer different output location | • Output saved in History (can be saved in a new History)<br>• History is located in subfolders of Galaxy instance | • Output is saved in user chosen workbench directory<br>• Option to save results in subfolder for each dataset in batch mode |

image in Figure 2 and figures for the other WFMS can be found in the GitHub repository [21] in the sections for the individual WFMS. In summary, KNIME WFs support complex WFs more intuitively, while it can be challenging to work with larger WFs in the other two platforms.

Visual WFs can be instrumental for discussion among stakeholders, and the better they are presented, the less work is necessary to develop additional representations such as UML diagrams. KNIME excels in this task (Figure 3).

Galaxy WFs can be run from the command line/console, and KNIME WFs can be run headlessly in a similar manner. Galaxy also offers an API, thereby creating flexible interfacing with the WFMS.

Unique to KNIME is WF branching with *if* and *switch* nodes enabling the control of WF execution at runtime. Branching can be of interest when data-dependent choices need to be made in a WF (e.g. which read mapper (or read mapper settings) to use could be decided from the input data). While WF branching is not available in Galaxy and CLC, the former has been planning to add this option.

## Input and output

In KNIME input can be set by configuring various file or database nodes, by using WF variables, and quick forms. Similarly, Galaxy offers data selection from history before executing a WF. This option makes deploying a WF easier as there is a clear distinction between the actions and input of the WF. CLC offers two options for setting input when a WF is opened in the editor. Input can be set by configuring input nodes or before running the WF. Installed WFs limit users to select input available in the workbench.

In KNIME, there are special nodes that configure serialization to files or databases. For the RNA-seq analysis, third party nodes had to be used, which did not always adhere to this separation of concerns and saving of output and thus depended on these nodes. Some of these nodes write output in the same folder where the input data is located, while others allow users to select the output location during configuration. Galaxy and CLC enable users to select where to save the output. CLC also offers the option to only display the results instead of saving them. A challenge with the development of the WFs in this study was that tools integrated into the WFMS by third parties needed comprehensive testing. Such tools often did not adhere to the design philosophies of the WFMS.

## Tools and parameters

KNIME offers tools for most RNA-seq analysis steps except for transcriptome assembly, transcriptome read mapping quantification, and isoform discovery (this prevented alternative splicing analysis). Nodes that are not part of core packages can be obtained from the community nodes by adding additional software and by writing scripts. Galaxy offers tools for every step in the RNA-seq analysis, but they first need to be installed by a Galaxy administrator. Many tools are available in multiple versions, allowing WFs to depend on a particular one. CLC Genomics Workbench offers its own tools for many RNA-seq analysis steps, but other tools could be added, which was, however, not a focus of this study. We were not able to find clear descriptions of the algorithms used by CLC. Not providing such specifications is a pity since other commercial tools such as Matlab provide algorithms for the functionality they provide in their help section.

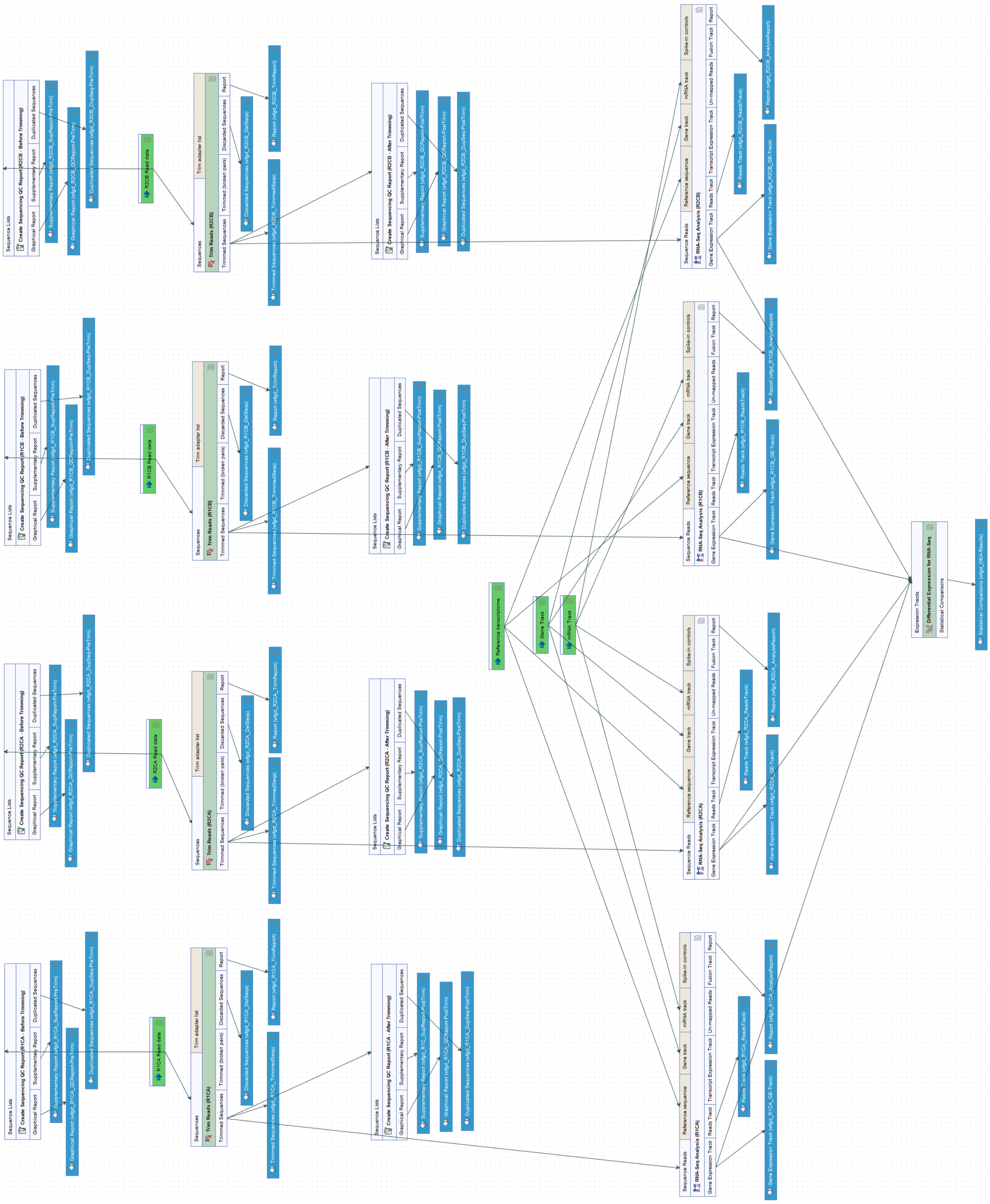

**Figure 2: KNIME transcriptome DEA WF.**

KNIME WF for the analysis of transcriptomics data to derive DE between two conditions based on read data mapped against a transcriptome. Processing nodes are placed in a loop structure to process data sets individually and to allow the automatic processing of arbitrary amounts of read files.
Larger figures are available in our GitHub repository [21] in the knime directory.

All tools in the three WFMS offer a variety of parameters to configure the algorithm. However, not all parameters that the standalone tool offers are always reflected in the nodes of the WFMS. Galaxy WFs need to be edited to change tool parameters. Galaxy also allows tool parameters to be changed via the API or when rerunning a specific WF step. CLC also allows parameters of tools to be changed when running the WF. Parameter learning in WFs might be interesting to allow for automatic optimization, but at present, this is only possible using KNIME. The major challenge in this section is that tools integrated into the WFMS by third parties may not make available all tool options or even misname them (the latter was not observed for RNA-seq analysis).

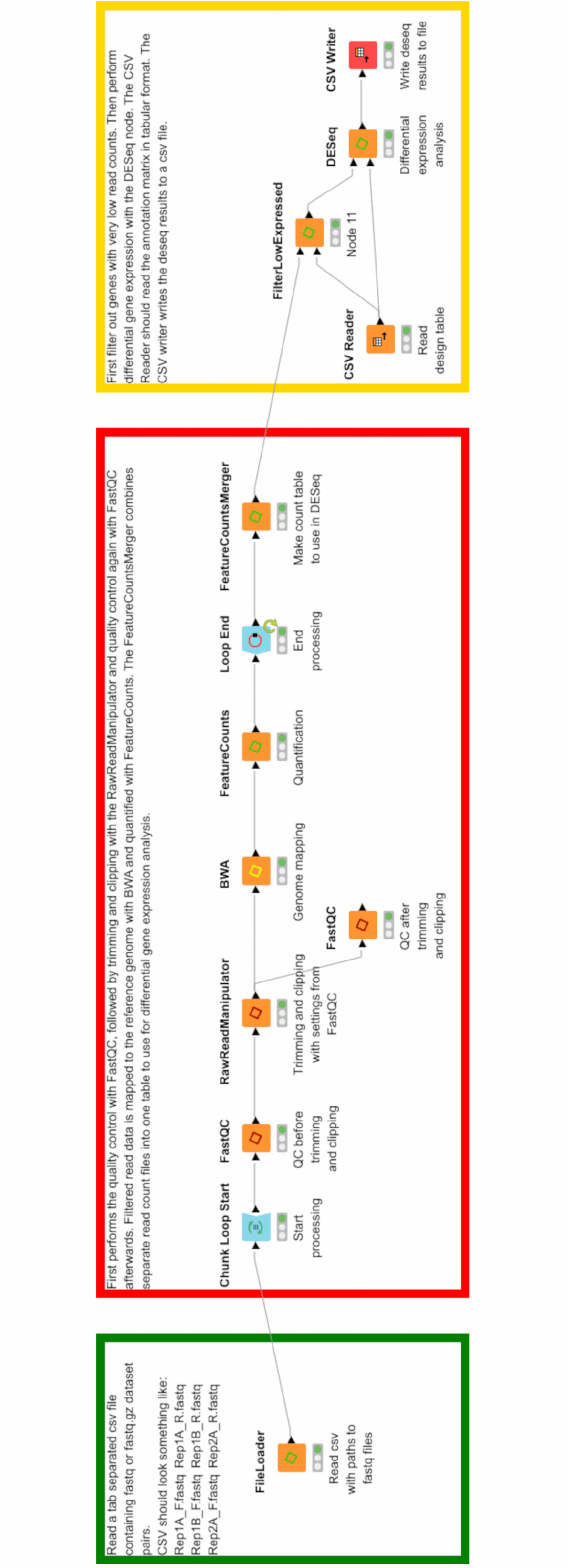
Read a tab separated csv file containing fastq or fastq.gz dataset pairs.
CSV should look something like:
Rep1A_F.fastq Rep1A_R.fastq
Rep1B_F.fastq Rep1B_R.fastq
Rep2A_F.fastq Rep2A_R.fastq
FileLoader
Read csv with paths to fastq files
First performs the quality control with FastQC, followed by trimming and clipping with the RawReadManipulator and quality control again with FastQC afterwards. Filtered read data is mapped to the reference genome with BWA and quantified with FeatureCounts. The FeatureCountsMerger combines separate read count files into one table to use for differential gene expression analysis.
Chunk Loop Start
Start processing
FastQC
QC before trimming and clipping
RawReadManipulator
Trimming and clipping with settings from FastQC
FastQC
QC after trimming and clipping
BWA
Genome mapping
FeatureCounts
Quantification
Loop End
End processing
FeatureCountsMerger
Make count table to use in DESeq
First filter out genes with very low read counts. Then perform differential gene expression with the DESeq node. The CSV Reader should read the annotation matrix in tabular format. The CSV writer writes the deseq results to a csv file.
CSV Reader
Read design table
FilterLowExpressed
Node 11
DESeq
Differential expression analysis
CSV Writer
Write deseq results to file

**Figure 3: Galaxy transcriptome DEA WF.**

Galaxy WF that performs DEA between two conditions based on read data mapped against a transcriptome. Each condition has its own, but conceptually identical, path to process data sets, resolving at the DEA step. Larger figures are available in our GitHub repository [21] in the galaxy directory.

## Result visualization

KNIME does not offer many specific nodes for the visualization of genomic data, although IGV [24] was used within KNIME in at least one study [25]. In general, plotting nodes could be used to visualize, for example, the results of the differential expression analysis. The R [26] integration in KNIME allows interaction with R. For example, plotting can be achieved or the integration of scripts from CRAN. R includes a wide variety of bioinformatics tools, but adding scripts (especially encapsulated ones) reduces WF comprehensibility and compromises reproducibility. Galaxy and CLC both offer many visualization options ranging from viewing plain text files and reports to plotting and inspecting genome mappings via genome browsers.

## Workflow sharing

Each WFMS allows users to share, import, and export WFs. Imported WFs provide warning messages if tools are missing or the versions differ. KNIME and CLC require their server applications to share WFs, whereas Galaxy users can select with which other users to share their WF. KNIME and Galaxy WFs can also be exported and imported using the WFMS. Projects like SHIWA [15] and CWL [14] aim to make WFs shareable among WFMS. CLC and KNIME do not have CWL support, but projects like KNIME2gUse [16] allow KNIME WFs to be executed by gUse [27]. Galaxy has CWL support in the alpha stage and is likely to continue the development and implementation of the proposed format. Galaxy WFs can, furthermore, be shared with different platforms through projects like Tavaxy [28], Closha [29], and Galaxy2gUse [16]. An alternative approach to sharing WFs is to encapsulate them into containers such as Docker or virtual machine images. While this can be useful, it does not touch on the reproducibility of WFs among WFMS.

# Workflow: Differential expression for transcriptome mapping

In each WFMS, several RNA-seq WFs were implemented while comparing them. One of these WFs is discussed in more detail below as an example to demonstrate differences between each WFMS in a specific WF case. The WF performs differential expression analysis (DEA) between two conditions with multiple replicates from read mapping to the transcriptome. The greatest challenge was that the same RNA-seq analysis tools were not available in the three platforms, which may lead to differing results among WFs.

## KNIME workflow

The WF (Figure 3) has been designed to allow for two or more replicates and multiple conditions. To process paired data sets individually, trimming, quality control (QC) before and after, mapping, and quantification were placed into a loop structure. Finally, count tables are combined, filtered, and used in differential expression analysis with a design table. Only samtools [30] idxstats was available for transcriptome read mapping quantification but could not be used, as the required bam index (.bai) file could not be created using any available KNIME node. A Java Snippet node, calling several samtools commands (view, sort, index, and idxstats), was, therefore, developed to quantify read mapping. Alternatively, the External Tool node could have been used to call, for example, Kallisto [31], but this type of manipulation did not fit the aim of this study. Also, there were no nodes available to combine count tables. Therefore, an R script node was used to perform this task. The challenges were missing nodes, which forced us to develop scripts, thereby reducing the comprehensibility and reproducibility of the WF – furthermore adding the possibility to introduce errors.

## Galaxy workflow

The Galaxy WF (Figure 4) consists of two paths, one path per replicate, with each path performing trimming and clipping, including QC before and after, mapping and quantification for one condition. Replicate data for each path was saved as a dataset collection in the History instead of as separate datasets. The differential expression tool then joins the two paths. Tool parameters were kept at default values. One factor, called ‘condition’, with ‘control’ and ‘treated’ as the two levels, was used for DEA. Initially, Sickle [32] was used for trimming but was later replaced with Trimmomatic [33] since Sickle did not support processing of dataset collections. As a side note, some tools may require R packages to be installed (e.g. DESeq2 [34]). The first approach to the WF was the usage of a sub-WF performing QC, trimming, mapping, and quantification. This sub-WF was used four times, and each sub-WF was connected to the DESeq2 tool. As noted before, sub-WFs did not forward the output to the connected tools and thus could not be used, which makes the overall WFs look a bit convoluted (Figure 4). This complication and the absence of loops makes scaling to more complex WFs a challenge.

## CLC workflow

The WF design (Figure 2) consists of four paths, each performing trimming and clipping with QC before and after, mapping and quantification for paired datasets. At the differential expression step these paths are joined. Most tool parameters were kept at their default value. During trimming, the option to remove reads deemed too short or long was set to default values. The type of DEA was set to ‘group against group’. Two designs made earlier, meant to be run in batch mode, could not be successfully implemented. The first aimed to make the WF similar to the one implemented in KNIME. The other aimed to make the WF similar to that implemented in Galaxy. Both implementations failed at the DEA step since

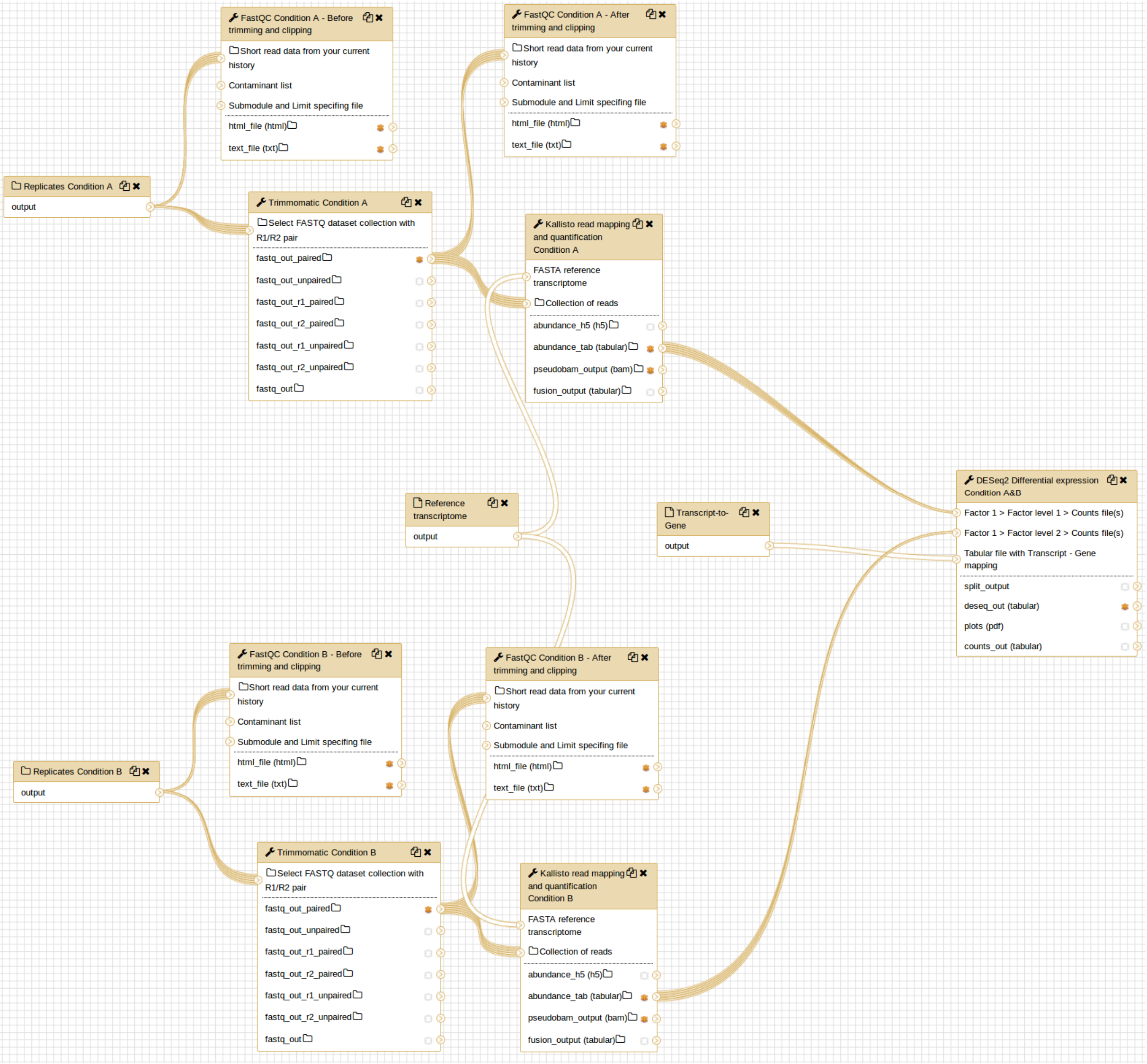
FastQC Condition A - Before trimming and clipping
Short read data from your current history
Contaminant list
Submodule and Limit specifing file
html_file (html)
text_file (txt)
FastQC Condition A - After trimming and clipping
Short read data from your current history
Contaminant list
Submodule and Limit specifing file
html_file (html)
text_file (txt)
Replicates Condition A
output
Trimmomatic Condition A
Select FASTQ dataset collection with R1/R2 pair
fastq_out_paired
fastq_out_unpaired
fastq_out_r1_paired
fastq_out_r2_paired
fastq_out_r1_unpaired
fastq_out_r2_unpaired
fastq_out
Kallisto read mapping and quantification Condition A
FASTA reference transcriptome
Collection of reads
abundance_h5 (h5)
abundance_tab (tabular)
pseudobam_output (bam)
fusion_output (tabular)
Reference transcriptome
output
Transcript-to-Gene
output
DESeq2 Differential expression Condition A&B
Factor 1 > Factor level 1 > Counts file(s)
Factor 1 > Factor level 2 > Counts file(s)
Tabular file with Transcript - Gene mapping
split_output
deseq_out (tabular)
plots (pdf)
counts_out (tabular)
FastQC Condition B - Before trimming and clipping
Short read data from your current history
Contaminant list
Submodule and Limit specifing file
html_file (html)
text_file (txt)
FastQC Condition B - After trimming and clipping
Short read data from your current history
Contaminant list
Submodule and Limit specifing file
html_file (html)
text_file (txt)
Replicates Condition B
output
Trimmomatic Condition B
Select FASTQ dataset collection with R1/R2 pair
fastq_out_paired
fastq_out_unpaired
fastq_out_r1_paired
fastq_out_r2_paired
fastq_out_r1_unpaired
fastq_out_r2_unpaired
fastq_out
Kallisto read mapping and quantification Condition B
FASTA reference transcriptome
Collection of reads
abundance_h5 (h5)
abundance_tab (tabular)
pseudobam_output (bam)
fusion_output (tabular)

**Figure 4: CLC transcriptome DEA WF.**
CLC WF performing DEA between two conditions with two replicates. Four identical paths each process one data. The paths are joined during the DEA step. Blue boxes represent output, not tools. Larger figures are available in our GitHub repository [21] in the clcbio directory.

the differential expression tool cannot be run in batch mode. Creating more complex WFs may be a challenge hard to overcome as the WFs would become visually convoluted (Figure 2).

# Discussion

## Transcript quantification

The three WFs were run using the same data so that their results should, in theory, be comparable (Figure 5). However, different tools were used for the analysis because the same tools were not available on all platforms. For example, CLC has only its own implementations, and adding custom nodes to CLC, KNIME, or Galaxy was beyond the scope of this study. However, this can be done by experts for a given WFMS so the expertise of the user is another important factor. For KNIME and Galaxy, it would potentially have been possible to use the same tools, but the tool versions were not clear for KNIME and some of the tools could not be connected to other nodes in one of the platforms. Therefore, different tools had to be used to construct the RNA-seq data analysis WFs. We show the results below without claim to any biological significance.

Instead, we intend to highlight that small differences among WFs lead to varying results. Furthermore, the data is not annotated with a ground truth, which makes it impossible to judge the success of the WFs. We believed it is essential to quantify the difference and, therefore, we tried to compare the results with the intent of making the reader aware of the consequences of choosing one WF over another. Our strategies for filtering the count data were based on the count distribution per WF. Filtering can be performed using count cutoffs, and to make the results comparable, the cutoffs were determined from the count distributions (Table 2). Applying no filtering re-

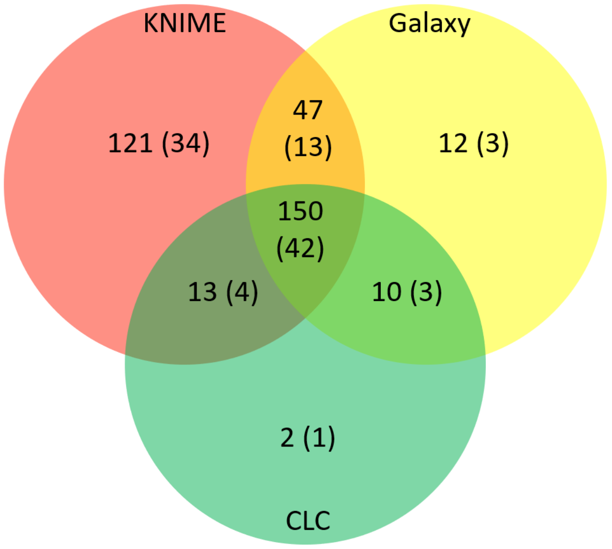

**Figure 5: Venn diagram of shared and unique transcripts among WFMS.**
Results shown are filtered at the 75th percentile (for Venn diagrams at other cutoffs see Figure S4 in Attachment 1).
Galaxy and CLC share most transcripts with at least one other WFMS while KNIME has many unique transcripts at this cutoff.

**Table 2: Shared and unique transcripts per WMS.**
Each row indicates results after applying filtering choosing different cutoffs.
Cutoff values were decided based on the count data distribution (see Attachment 1).

| | KNIME | Galaxy | CLC | KNIME, Galaxy | KNIME, CLC | Galaxy, CLC | All |
|---|---|---|---|---|---|---|---|
| **No filter** | 0 | 0 | 0 | 0 | 0 | 0 | 8,351 |
| **Filter >10th percentile** | 385 | 68 | 2 | 203 | 79 | 0 | 479 |
| **Filter >25th percentile** | 342 | 32 | 4 | 146 | 67 | 0 | 429 |
| **Filter >50th percentile** | 313 | 0 | 1 | 10 | 118 | 0 | 209 |
| **Filter >75th percentile** | 121 | 12 | 2 | 47 | 13 | 10 | 150 |
| **Filter >90th percentile** | 3 | 0 | 7 | 0 | 3 | 1 | 2 |

veals that all tools share all identified transcripts, whereas applying stringent filtering at the 90th percentile shows that only a few shared results remain among the different approaches (Table 2). The Venn diagram in Figure 5 shows the identified transcript distribution for filtering at the 75th percentile. The majority of the results are shared among at least two tools (~62%). The diagram corresponds to row five in Table 2. Venn diagrams for the other percentiles are available in Figure S4 in Attachment 1. Some of the results, such as zero values for no filter applied in Table 2, may not be intuitive at first glance. They result from the fact that all transcripts are shared among all tools, which leaves individual parts and intersections between any two tools empty. With progressive filtering, the transcripts may be differentially filtered for different WFs leading to largely different distributions with occasional increases for shares between tools. KNIME and Galaxy more consistently share transcripts that CLC did not identify. This higher similarity in results can be attributed to the usage of open source tools in Galaxy and KNIME. Open-source tools may be tested more openly, and algorithms may converge, whereas the black-box approach in CLC prevents such discussions.

Interestingly, when filtering at the 90th percentile, CLC retains more results than KNIME or Galaxy. No biological question was analyzed in this study, and no follow-up experimental analysis was planned. For such studies, it is, however, essential to choose a cutoff, which reduces the number of results to make them accessible in the wet-lab. The question is then which WF to use in which WFMS. We hope that sharing our WFs will spark discussion and testing so that best practices can be established in the future.

The greatest challenge is to develop the same WFs in all WFMS. Implementing a particular WF following some best practices holds the same challenge, and the availability of tools and their versions must be investigated before implementation. Another challenge is the difference in results found in this proof of principle. Here, we merely want to make the reader aware that when creating WFs with different tools (which may be unavoidable), different results entail, which may, in the worst case, lead to contradicting biological explanations. The filtering of the results does not alleviate the challenge, and it is prerogative to scrutinize results in the wet lab. In the future, a comprehensive evaluation of RNA-seq analysis tools would be beneficial.

## Other workflows

Apart from the WFs discussed in detail above, three other WFs have been constructed in each WFMS. These WFs follow the same steps as the WF described above and differ in the tools used at specific steps such as read mapping. One of the three WFs (named 'wf_genmap_dea') first maps and quantifies multiple paired read datasets for two conditions against a reference genome and subsequently performs DEA. Implementing this WF in Galaxy and CLC posed some minor challenges. In Galaxy, the FASTA definition line present in the count files had to be removed manually before DEA. In CLC, the same approach as for DEA from transcriptome mapping had to be implemented. The other two WFs are 'wf_genmap_multi' and 'wf_trmap_multi' map and quantify many paired read datasets against either a reference genome or transcriptome.

# Conclusions

RNA-seq analysis is a common task in data-driven biology and medicine. Our research concerned the development and comparison of RNA-seq WFs in KNIME, Galaxy, and CLC Genomics Workbench. In summary, we find that Galaxy can best be used for the development of RNA-seq WFs. This belief is due to several factors: 1) many bioinformatics tools and their versions are available, 2) Galaxy allows the sharing of WFs most seamlessly, for example, between Galaxy instances, CWL support, Tavaxy, Closha, and Galaxy2gUse, and 3) because the Galaxy API allows a lot of flexibility and possibilities for the use of the WFs. CLC Genomics Workbench, however, might be the best solution for exploratory data analysis. This suggestion is mainly due to its user-friendly interface, focus on bioinformatics and its strong visualization facilities. These two aspects allow researchers to quickly perform their primary analysis in a streamlined and consistent environment.

KNIME supported the scalability of analysis best due to the implemented loop structure. However, it was not selected for a recommendation for RNA-seq WF analysis since it is currently missing tools, and scripting nodes had to be used to circumvent the problem. Additionally, some of the available nodes had idiosyncratic approaches to file handling. However, unlike Galaxy and CLC, KNIME was not designed for bioinformatics analyses. KNIME's design philosophy allows nodes from different fields to be connected easily. Therefore, KNIME can be of great use for machine learning and data integration tasks in bioinformatics, for which it offers many nodes and functionalities. Also note that the development of new nodes in KNIME is not very involved, and the missing tools could be added.

One factor for the recommendations above that we cannot factor in is the familiarity of a user with any of the WFMS. The expertise of a research group with one WFMS should be weighed against our recommendation above. Another caveat is that by preselecting the three WFMS we could establish criteria that may be important to select WFMS but were not able to use them. Therefore, the suggestions above are not based on a scoring system (Attachment 2) but are based on the major challenges we experienced while developing the criteria and the WFs.

In conclusion, it was not possible to develop the exact same WFs in the three systems using the same computational tools. This problem makes cross-WF reproducibility a formidable challenge. Resulting differences in DEA for the same data show that the underlying tools in the WFMS need a comprehensive evaluation and comparison in the future. End-users need to become aware that performing RNA-seq analysis with a pipeline presents a choice that produces results that may differ or even contrast biological interpretation when compared to results from a slightly different analysis pipeline. In conclusion, while we were able to give a recommendation of which WFMS to use for RNA-seq analysis, we raised the question of how the pipeline should be constructed. The latter question needs to be answered via a comprehensive analysis of RNA-seq analysis tools and their impact on the results. We hope that making our WF available will be a step in that direction.

# Notes

## Acknowledgments

The authors thank Prof. Dr. Robert D. Hall for proofreading the manuscript.

## Competing interests

The authors declare that they have no competing interests.

# Attachments

Available from https://doi.org/10.3205/mibe000245

1. Attachment 1_mibe000245.pdf (950 KB)
   Appendix – supplementary figures and tables
2. Attachment 2_mibe000245.pdf (129 KB)
   Factors considered during WFMS comparison

# References

1. Costa-Silva J, Domingues D, Lopes FM. RNA-Seq differential expression analysis: An extended review and a software tool. PLoS One. 2017;12(12):e0190152. DOI: 10.1371/journal.pone.0190152
2. Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, Szcześniak MW, Gaffney DJ, Elo LL, Zhang X, Mortazavi A. A survey of best practices for RNA-seq data analysis. Genome Biol. 2016 Jan;17:13. DOI: 10.1186/s13059-016-0881-8
3. Albert E, Duboscq R, Latreille M, Santoni S, Beukers M, Bouchet JP, Bitton F, Gricourt J, Poncet C, Gautier V, Jiménez-Gómez JM, Rigaill G, Causse M. Allele-specific expression and genetic determinants of transcriptomic variations in response to mild water deficit in tomato. Plant J. 2018 Nov;96(3):635-50. DOI: 10.1111/tpj.14057
4. Spjuth O, Bongcam-Rudloff E, Hernández GC, Forer L, Giovacchini M, Guimera RV, Kallio A, Korpelainen E, Kańduła MM, Krachunov M, Kreil DP, Kulev O, Łabaj PP, Lampa S, Pireddu L, Schönherr S, Siretskiy A, Vassilev D. Experiences with workflows for automating data-intensive bioinformatics. Biol Direct. 2015 Aug;10:43. DOI: 10.1186/s13062-015-0071-8
5. Köster J, Rahmann S. Snakemake – a scalable bioinformatics workflow engine. Bioinformatics. 2012 Oct;28(19):2520-2. DOI: 10.1093/bioinformatics/bts480
6. Hull D, Wolstencroft K, Stevens R, Goble C, Pocock MR, Li P, Oinn T. Taverna: A tool for building and running workflows of services. Nucleic Acids Res. 2006 Jul;34(Web Server issue):W729-32. DOI: 10.1093/nar/gkl320
7. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Cech M, Chilton J, Clements D, Coraor N, Grüning BA, Guerler A, Hillman-Jackson J, Hiltemann S, Jalili V, Rasche H, Soranzo N, Goecks J, Taylor J, Nekrutenko A, Blankenberg D. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018 Jul;46(W1):W537-44. DOI: 10.1093/nar/gky379
8. Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Sieb C, Thiel K, Wiswedel, B. KNIME: The Konstanz Information Miner. In: Preisach C, Burkhardt H, Schmidt-Thime L, Decker R, editor. Data Analysis, Machine Learning and Applications. Berlin: Springer; 2008. p. 319-26.
9. Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling PJ, Mayani R, Chen W, da Silva RF, Livny M, Wengerc K. Pegasus, a workflow management system for science automation. Future Gener Comput Syst. 2015;46:17-35. DOI: 10.1016/j.future.2014.10.008
10. Linke B, Giegerich R, Goesmann A. Conveyor: a workflow engine for bioinformatic analyses. Bioinformatics. 2011 Apr;27(7):903-11. DOI: 10.1093/bioinformatics/btr040
11. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJ, Groth P, Goble C, Grethe JS, Heringa J, 't Hoen PA, Hooft R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone SA, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016 Mar;3:160018. DOI: 10.1038/sdata.2016.18
12. King G. An introduction to the dataverse network as an infrastructure for data sharing. Sociol Methods Res. 2007;36(2):173-99. DOI: 10.1177/0049124107306660
13. Williams AJ, Harland L, Groth P, Pettifer S, Chichester C, Willighagen EL, Evelo CT, Blomberg N, Ecker G, Goble C, Mons B. Open PHACTS: Semantic interoperability for drug discovery. Drug Discov Today. 2012 Nov;17(21-22):1188-98. DOI: 10.1016/j.drudis.2012.05.016
14. Amstutz P, Crusoe MR, Tijanic N, Chapman B, Chilton J, Heuer M, Kartashov A, Leehr D, Ménager H, Nedeljkovich M, Scales M, Soiland-Reyes S, Stojanovic L. Common Workflow Language, v1.0. Figshare; 2016. DOI: 10.6084/m9.figshare.3115156.v2
15. Terstyanszky G, Kukla T, Kiss T, Kacsuk P, Balasko A, Farkas,Z. Enabling scientific workflow sharing through coarse-grained interoperability. Future Gener Comput Syst. 2014 Jul;37:46-59. DOI: 10.1016/j.future.2014.02.016
16. de la Garza L, Veit J, Szolek A, Röttig M, Aiche S, Gesing S, Reinert K, Kohlbacher O. From the desktop to the grid: Scalable bioinformatics via workflow conversion. BMC Bioinformatics. 2016 Mar;17:127. DOI: 10.1186/s12859-016-0978-9
17. Goble CA, Bhagat J, Aleksejevs S, Cruickshank D, Michaelides D, Newman D, Borkum M, Bechhofer S, Roos M, Li P, De Roure D. myExperiment: A repository and social network for the sharing of bioinformatics workflows. Nucleic Acids Res. 2010 Jul;38(Web Server issue):W677-82. DOI: 10.1093/nar/gkq429
18. Döring A, Weese D, Rausch T, Reinert K. SeqAn an efficient, generic C++ library for sequence analysis. BMC Bioinformatics. 2008 Jan;9:11. DOI: 10.1186/1471-2105-9-11
19. Hastreiter M, Jeske T, Hoser J, Kluge M, Ahomaa K, Friedl MS, Kopetzky SJ, Quell JD, Mewes HW, Küffner R. KNIME4NGS: A comprehensive toolbox for next generation sequencing analysis. Bioinformatics. 2017 May;33(10):1565-7. DOI: 10.1093/bioinformatics/btx003
20. Has C, Lashin SA, Kochetov AV, Allmer J. PGMiner reloaded, fully automated proteogenomic annotation tool linking genomes to proteomes. J Integr Bioinform. 2016 Dec 18;13(4):293. DOI: 10.2390/biecoll-jib-2016-293
21. Beukers M, Allmer J. RNA-seqFlow. Available from: https://github.com/MatthieuBeukers/RNA-seqFlow
22. Beukers M, Allmer J. Testdata. In: RNA-seqFlow. Available from: https://github.com/MatthieuBeukers/RNA-seqFlow/tree/master/testdata
23. Piras ME, Pireddu L, Zanetti G. wft4galaxy: A workflow testing tool for galaxy. Bioinformatics. 2017 Dec;33(23):3805-7. DOI: 10.1093/bioinformatics/btx461
24. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinform. 2013 Mar;14(2):178-92. DOI: 10.1093/bib/bbs017
25. Has C, Allmer J. PGMiner: Complete proteogenomics workflow; from data acquisition to result visualization. Inf Sci (NY). 2017 Apr;384:126-34. DOI: 10.1016/j.ins.2016.08.005
26. R Core Team. R: A Language and environment for statistical computing. 2016.
27. Kacsuk P, Farkas Z, Kozlovszky M, Hermann G, Balasko A, Karoczkai K, Marton I. WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities. J Grid Computing. 2012;10:601-30. DOI: 10.1007/s10723-012-9240-5
28. Abouelhoda M, Issa SA, Ghanem M. Tavaxy: Integrating Taverna and Galaxy workflows with cloud computing support. BMC Bioinformatics. 2012 May;13:77. DOI: 10.1186/1471-2105-13-77
29. Ko G, Kim PG, Yoon J, Han G, Park SJ, Song W, Lee B. Closha: Bioinformatics workflow system for the analysis of massive sequencing data. BMC Bioinformatics. 2018 Feb;19(Suppl 1):43. DOI: 10.1186/s12859-018-2019-3

30. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R; 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009 Aug;25(16):2078-9. DOI: 10.1093/bioinformatics/btp352

31. Bray NL, Pimentel H, Melsted P, Pachter L. Near-optimal probabilistic RNA-seq quantification. Nat Biotechnol. 2016 May;34(5):525-7. DOI: 10.1038/nbt.3519

32. Joshi N, Fass J. Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files. 2011. Available from: https://github.com/najoshi/sickle

33. Bolger AM, Lohse M, Usadel B. Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics. 2014 Aug;30(15):2114-20. DOI: 10.1093/bioinformatics/btu170

34. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014;15(12):550. DOI: 10.1186/s13059-014-0550-8

**Corresponding author:**

Prof. Dr. Jens Allmer
Hochschule Ruhr West, University of Applied Sciences, Medical Informatics and Bioinformatics, 45479 Mülheim an der Ruhr, Germany
jens@allmer.de

***Please cite as***

*Beukers M, Allmer J. Challenges for the development of automated RNA-seq analyses pipelines. GMS Med Inform Biom Epidemiol. 2023;19:Doc06.*
*DOI: 10.3205/mibe000245, URN: urn:nbn:de:0183-mibe0002457*

***This article is freely available from***
*https://doi.org/10.3205/mibe000245*

***Published:*** *2023-07-04*