



RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis

Margaux Haering, Bianca Habermann

► To cite this version:

Margaux Haering, Bianca Habermann. RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis. F1000Research, 2021, 10, pp.654. 10.12688/f1000research.54533.2. hal-03451631

HAL Id: hal-03451631

<https://amu.hal.science/hal-03451631>

Submitted on 26 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Check for updates

SOFTWARE TOOL ARTICLE

REVISED RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis [version 2; peer review: 2 approved, 1 approved with reservations]

Margaux Haering, Bianca H Habermann 

Aix-Marseille University, CNRS, IBDM UMR 7288, The Turing Centre for Living systems, Marseille, 13009, France

V2 First published: 26 Jul 2021, 10:654
<https://doi.org/10.12688/f1000research.54533.1>
 Latest published: 12 Nov 2021, 10:654
<https://doi.org/10.12688/f1000research.54533.2>

Abstract

RNA sequencing (RNA-seq) is a widely adopted affordable method for large scale gene expression profiling. However, user-friendly and versatile tools for wet-lab biologists to analyse RNA-seq data beyond standard analyses such as differential expression, are rare. Especially, the analysis of time-series data is difficult for wet-lab biologists lacking advanced computational training. Furthermore, most meta-analysis tools are tailored for model organisms and not easily adaptable to other species.


With RNfuzzyApp, we provide a user-friendly, web-based R shiny app for differential expression analysis, as well as time-series analysis of RNA-seq data. RNfuzzyApp offers several methods for normalization and differential expression analysis of RNA-seq data, providing easy-to-use toolboxes, interactive plots and downloadable results. For time-series analysis, RNfuzzyApp presents the first web-based, fully automated pipeline for soft clustering with the Mfuzz R package, including methods to aid in cluster number selection, cluster overlap analysis, Mfuzz loop computations, as well as cluster enrichments. RNfuzzyApp is an intuitive, easy to use and interactive R shiny app for RNA-seq differential expression and time-series analysis, offering a rich selection of interactive plots, providing a quick overview of raw data and generating rapid analysis results. Furthermore, its assignment of orthologs, enrichment analysis, as well as ID conversion functions are accessible to non-model organisms.



Keywords

RNA-seq, data normalization, data visualization, differential expression analysis, time-series analysis, soft clustering, Mfuzz, R shiny

Open Peer Review

Reviewer Status   

	Invited Reviewers		
	1	2	3
version 2 (revision) 12 Nov 2021	 report		
	↑		
version 1 26 Jul 2021	 report	 report	 report

1. **Oliver Hahn** , Stanford University, Stanford, USA
2. **Rhonda Bacher** , University of Florida, Gainesville, USA
3. **Anita Grigoriadis**, King's College London, London, UK

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the **RPackage** gateway.

Corresponding author: Bianca H Habermann (bianca.habermann@univ-amu.fr)

Author roles: **Haering M:** Conceptualization, Data Curation, Formal Analysis, Methodology, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Habermann BH:** Conceptualization, Formal Analysis, Funding Acquisition, Project Administration, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported by the French National Research Agency with ANR grant ANR-18-CE45-0016-01 MITO-DYNAMICS awarded to BHH.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2021 Haering M and Habermann BH. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Haering M and Habermann BH. **RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis [version 2; peer review: 2 approved, 1 approved with reservations]** F1000Research 2021, **10**:654 <https://doi.org/10.12688/f1000research.54533.2>

First published: 26 Jul 2021, **10**:654 <https://doi.org/10.12688/f1000research.54533.1>

REVISED Amendments from Version 1

We have tried to clarify a few points that seemed unclear in the paper. Furthermore, we have created a detailed user manual, including detailed instructions how to install the software and how to run its different functions.

We now also provide test data files (which are used both, in the paper, as well as the manual) so that researchers can test our software and also verify the correctness of the format of their data.

We have meanwhile also updated RNfuzzyApp, and added new functionalities.

- 1) It is now possible to compare two datasets, even if more than 2 datasets were initially uploaded. For this, the Filter data function has been introduced.
- 2) Groups are now assigned automatically. The user has to follow the guideline on how to format the column names (condition1_replicate1, condition1_replicate2, etc.).
- 3) It is now possible to directly load DEG gene lists into the enrichment function of RNfuzzyApp, making the analysis more streamlined.

Any further responses from the reviewers can be found at the end of the article

Introduction

The development of next generation sequencing (NGS) methods has boosted the rapid generation of large datasets and RNA sequencing (RNA-seq) has become the standard for performing robust transcriptional profiling and thus quantifying gene expression in various contexts. Next to the comparison of two conditions, the generation of time-series RNA-seq data has become amenable and popular, allowing to monitor the gene expression dynamics over a process such as development, ageing or cancerogenesis. While web-based, user-friendly R shiny apps have become available recently for differential expression analysis and data visualization of RNA-seq data,^{1–7} the analysis of time-series data within R remains largely command-line based and therefore challenging for bench scientists without programming knowledge.

We here present RNfuzzyApp, a user-friendly, web-based R shiny app with an intuitive user interface for the full workflows of differential expression analysis, as well as time-series analysis of RNA-seq data. RNfuzzyApp provides an interface for easy and fast data normalization and differential analysis using several methods, a variety of interactive plots for a quick overview of data and results, and an easy-to-use interface for the complete pipeline of time-series expression analysis using the fuzzy clustering algorithm Mfuzz.⁸ In addition, RNfuzzyApp offers ID conversion, orthology assignment and enrichment analysis using gprofiler2.⁹ We show the usability of RNfuzzyApp on two examples: an RNA-seq dataset of the ageing limb muscle of mouse, as well as developmental time-series RNA-seq data of the *Drosophila melanogaster* leg.

Methods

Implementation

RNfuzzyApp was built in R (V.4.0.4) using the **Shiny framework**. The app currently depends on the following R packages: *shiny*, *shinydashboard*, *shinycssloaders*, *shinythemes*, *shinyWidgets*, *shinyBS*, *rmarkdown*, *plotly*, *dplyr*, *RColorBrewer*, *utils*, *tidyr*, *devtools*, *cluster*, *DESeq2*,¹⁰ *edgeR*,¹¹ *TCC*,¹² including *baySeq*,¹³ *heatmaply*,¹⁴ *gprofiler2*,⁹ *Mfuzz*,⁸ as well as the package *e1071*. As a basic feature and to allow users to upload any identifier for analysis, ID conversion is included, using the *gprofiler2* package.

Operation

RNfuzzyApp can be launched locally from any computer with R (version 4.0.4 or higher) installed and will run in any web-browser. As RNfuzzyApp auto-installs all required R-packages, there exist no additional software requirements. Installation instructions are also available. All interfaces and plots of RNfuzzyApp are highly interactive, allowing users to visualize data in real-time as well as to interact efficiently with the data and plots.

Workflows of RNfuzzyApp

The general workflow of RNfuzzyApp is shown in **Figure 1**. It can be divided into two independent parts: 1) a complete workflow for differential expression analysis of RNA-seq data (**Figure 1a**); and 2) a complete workflow for the clustering of RNA-seq using the soft clustering algorithm Mfuzz (**Figure 1b**).

Differential expression analysis workflow of RNfuzzyApp

The differential expression analysis workflow of RNfuzzyApp can be divided in three main parts: the upload and visualization of the raw data; the normalization of the data and the differential expression analysis; and finally, enrichment

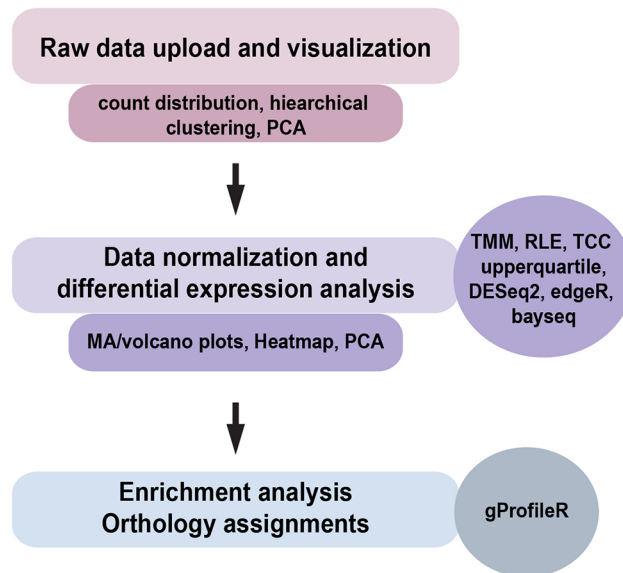
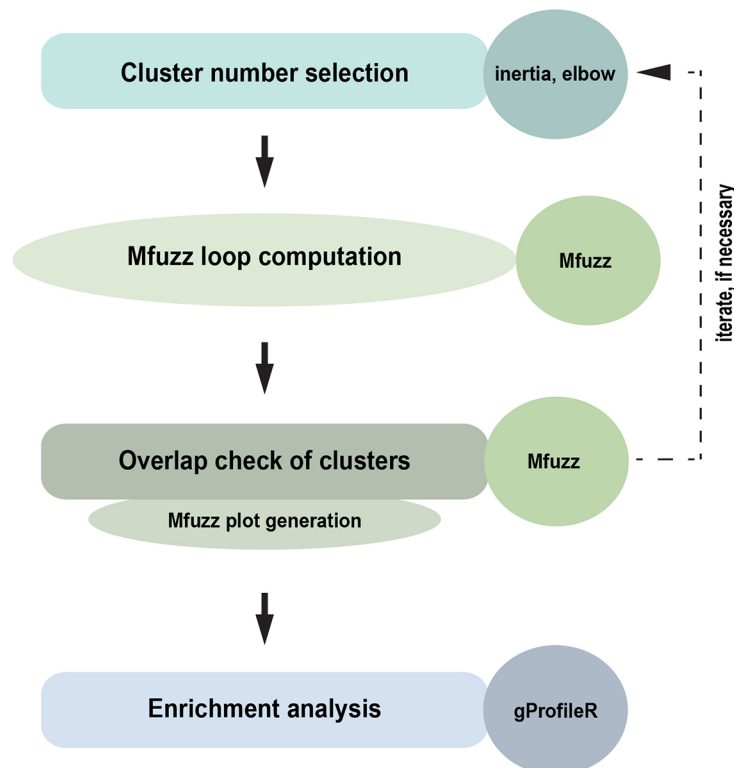
a Differential expression analysis workflow**b Mfuzz clustering workflow**

Figure 1. The two RNFuzzyApp analysis pipelines. (a) RNA-seq differential expression analysis workflow with the three main parts: data upload and visualization, data normalization and differential expression analysis, as well as enrichment analysis and the assignment of orthologous genes across species. The types of analyses are shown, as well as the various possible R programs provided for data analysis. (b) Mfuzz workflow for clustering of time-series RNA-seq data. The workflow includes the selection of cluster numbers, checking the overlap of Mfuzz clusters, loop calculations of Mfuzz, Mfuzz plot generation, as well as enrichment analysis of Mfuzz clusters.

analysis of results and orthology assignment (Figure 1a). In each part, several options exist for visualizing the data and thus getting a first-hand impression of the quality of the data, as well as the filters that are applied.

Data upload and visualization of raw gene expression data

Figure 2 shows the RNfuzzyApp start interface, featuring data upload, filtering, as well as raw data visualization possibilities. As a first step, raw read counts need to be uploaded to the app, in the form of a csv count matrix. Data can be filtered for raw read counts (Figure 2c), the resulting summary of the data are interactively updated in the Summary box of

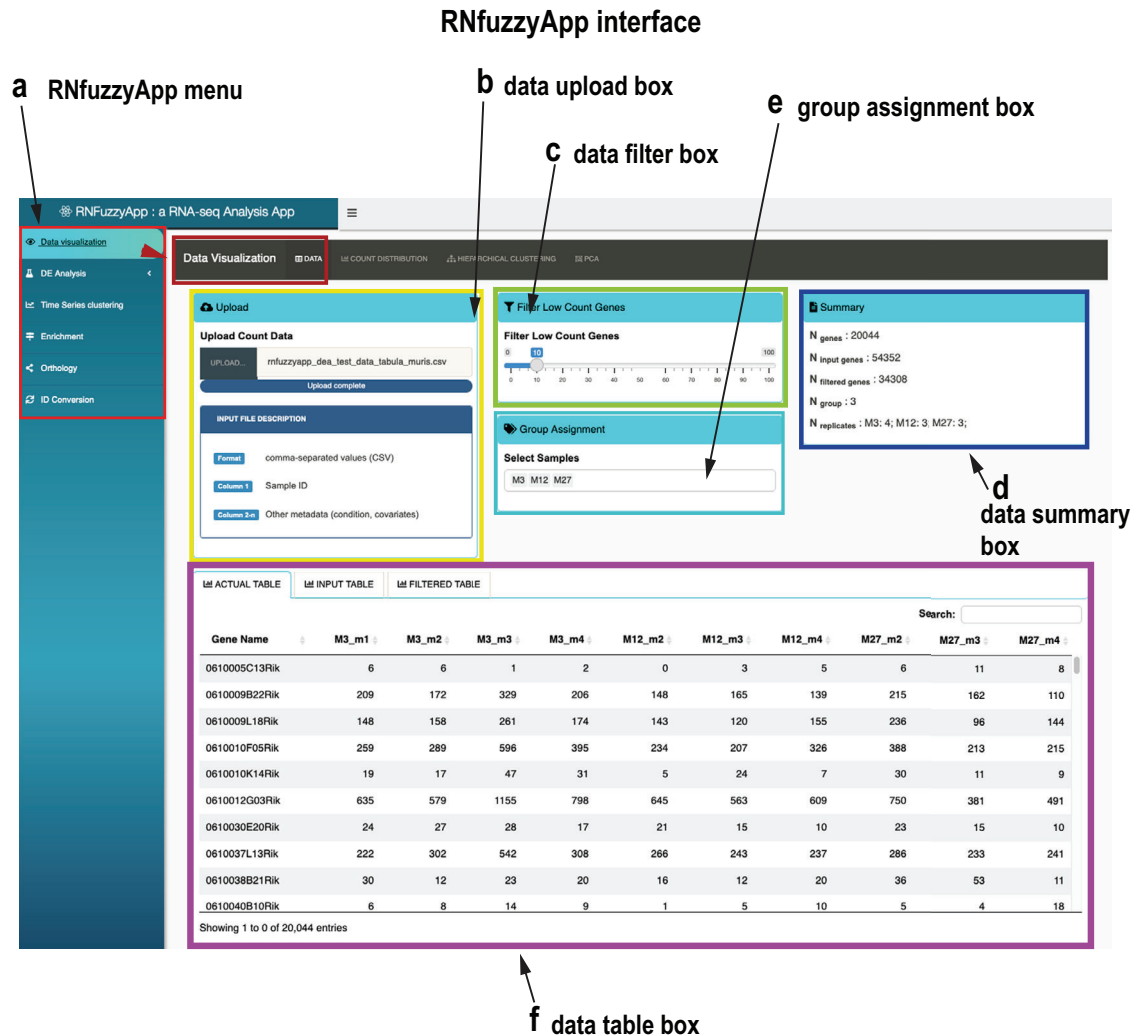


Figure 2. RNfuzzyApp start interface. (a) The RNfuzzyApp menu box is highlighted in red. This box is shown consistently over all interfaces and links to Data visualization, DE analysis, Data normalization and analysis, different visualization possibilities (MA plot, Volcano plot, Heatmap and PCA), Time series clustering (using the pipeline for Mfuzz soft clustering), Enrichment analysis, Orthology assignment, as well as ID-conversion. The main interface shown here belongs to Data visualization (highlighted in dark red). (b) Data upload box, for upload of user-provided data. (c) Data filter box, in which the user can choose to filter out genes with low read counts, (d) in which the identified groups are listed (e.g. wild-type and mutant or different time-points). Group assignment is automatic, so the name format of the samples has to follow a specific pattern. (e) Data summary box. The data shown in this box are updated and in case of filtering and are renewed on the fly. (f) Data table box. Three tables are provided: the actual table, including genes that were not discarded due to filtering; the input table, containing all data uploaded; finally the filtered table, containing genes that were removed due to filtering.

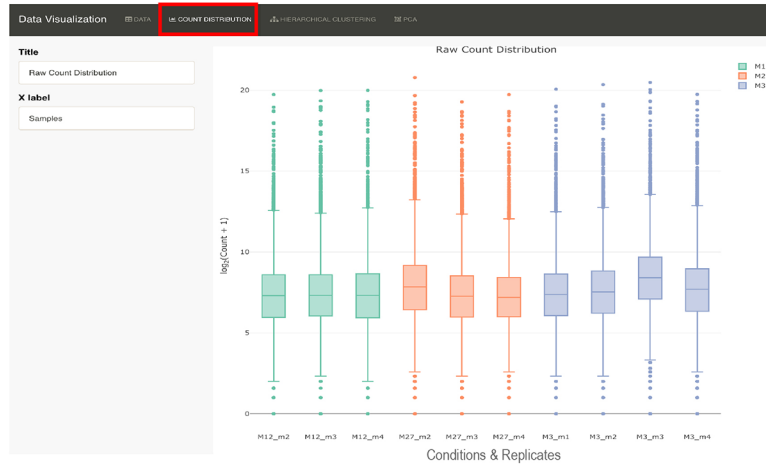
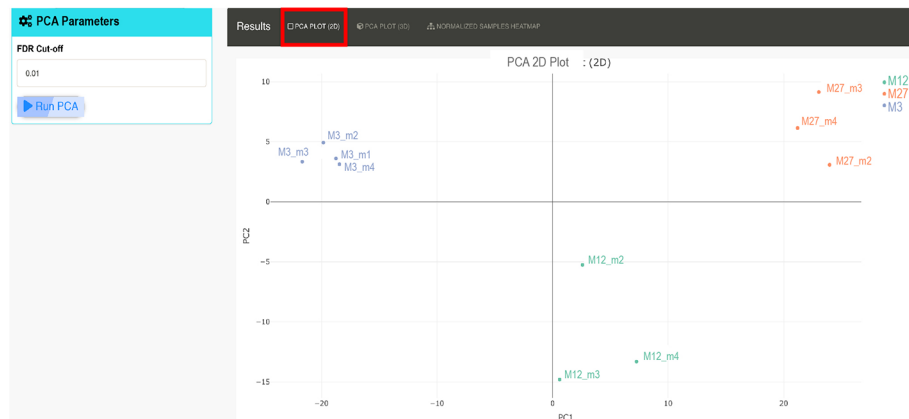
a**Count distribution plot of raw read counts****b****Hierarchical clustering of normalized read counts****c****PCA analysis of normalized read counts**

Figure 3. Data visualization plots offered in RNfuzzyApp. We used data from the Tabula muris senis project for demonstration purposes. (a) Count distribution plot of mice from 3 months, 12 months and 27 months. Only replicates from male mice were chosen. Data are grouped by condition. The title of the plot, as well as the X-axis label can be chosen by the user. Raw read counts were chosen for visualization. (b) Hierarchical clustering of normalized read counts. Spearman correlation was used for clustering. (c) PCA plot of normalized read counts. The PCA plot can be visualized in 2D or 3D. DESeq2 was used for normalization of data.

the interface (Figure 2d). Groups can be assigned directly in the interface (Figure 2e). Three tables are available for download: the actual table, containing only the genes that pass the filtering threshold; the original input table; as well as a table containing all genes filtered out due to low read counts (Figure 2f). Raw data can also be visualized (see top of the menu, Figure 2): the count distribution of raw read counts can be visualized (Figure 3a). Moreover, raw read counts can be used for hierarchical clustering, as well as a PCA analysis.

Data normalization and differential expression analysis

RNfuzzyApp offers several packages for data normalization, as well as for differential expression analysis. Normalization can be done using DESeq2; TMM (trimmed mean of M values), RLE (relative log expression) or upperquantile offered by edgeR; finally the TCC package providing TMM or DESeq2 normalization. As for raw read counts, the count distribution, a heatmap for clustering samples (Figure 3b), as well as PCA analysis with a 2D as well as 3D PCA plot (Figure 3c) is available for visualising normalized data. Differential expression analysis can be done using DESeq2, edgeR and bayseq. If more than two conditions are uploaded, normalization and initial differential expression analysis will be done over the entire data set. However, it is often useful to perform pairwise comparison of two conditions or time-points. For pairwise comparisons of two conditions or time-points of larger datasets, a Filter menu is provided. Data resulting from pairwise comparison can be visualized with MA and Volcano plots. All plots are interactive and the user can obtain detailed information about a gene hovering over the dots of the plots. All details on normalization and differential expression analysis can be found in the user manual of RNfuzzyApp.

Clustering of expression data using heatmaply

We wanted to provide a simple way of clustering gene expression data from a limited number of samples, e.g. from a short time-series. To this end, RNfuzzyApp offers a heatmap, generated by *heatmaply*. Figure 4a shows the heatmap of a time-course of 3 time points from the Tabula muris senis project,¹⁵ where the gene expression levels of the replicates per time point are clustered using hierarchical clustering. The user can choose the distance matrix and agglomeration method, as well as the FDR cut-off of genes to include in the clustering. Clustering is done using *hclust* and *cutree* to generate the coloured dendrogram on the heatmap. The genes contained in the different clusters indicated by the colours in the dendrogram are downloadable as a table for further analysis.

Enrichment analysis and orthology assignment.

For enrichment analysis of Gene Ontology (GO-) terms,¹⁶ pathways (Wikipathways,¹⁷ Reactome¹⁸ and KEGG¹⁹), Human Protein Atlas,²⁰ CORUM data on protein complexes,²¹ and TRANSFAC,²² the gprofiler2 package is included in RNfuzzyApp. Results are displayed as an image of overall enrichment, as well as a results table. The table together with a bar plot of the enriched term names sorted according to p-value are downloadable by the user (Figure 4b, bar plot of enrichments generated from downloaded table). Gprofiler2 is also used to find orthologs in another species of a user-provided list of genes. To this end, the user simply needs to upload a list of genes, and select the original and the target species.

Complete workflow for fuzzy clustering of time-series data

Fuzzy clustering of time-series expression data is a highly useful technique for analysing temporal data. The Mfuzz package from R was developed for soft clustering of temporal gene expression data.⁸ Starting from a count matrix, genes are clustered according to their expression profiles over time. As Mfuzz is a soft clustering algorithm, a gene can in theory be part of more than one cluster. Mfuzz, however, is not straightforward to use for non-experts. First, a number of clusters must be chosen prior to clustering. Second, repeated Mfuzz runs will result in slightly different cluster memberships of genes. A user is therefore well advised to repeat Mfuzz clustering several times to test the robustness of the clustering. The decision, which cluster number is suited for the data then often includes analysis of cluster overlaps, as well as enrichment analyses of clusters and comparative analysis between several chosen numbers of clusters. Several packages exist to help decide on cluster numbers and the entire workflow for a successful Mfuzz clustering can be programmed in R. However, for untrained bench scientists, this is not easily done. We therefore included the complete workflow of Mfuzz soft clustering of time-series expression data in RNfuzzyApp (Figure 1b): first, for choosing the right cluster number, we implemented the *inertia* (using the *hclust* and *dist* packages) and *elbow* (using the *e1071* package) methods. *Inertia* performs hierarchical clustering and plots the dendrogram, indicating the distance steps (height) against the number of clusters. Ideally, a cluster number is chosen when the drop in height gets minimal (Figure 5a). The *elbow* method looks at the total of the within-clusters sums of squares (WCSS) as a function of the number of clusters. The “elbow” shape is formed when WCSS is minimal. These two methods should converge to help choose the right number of clusters. After Mfuzz clustering has been performed, the overlap of clusters can be checked. To do so, the *overlap.plot* function from

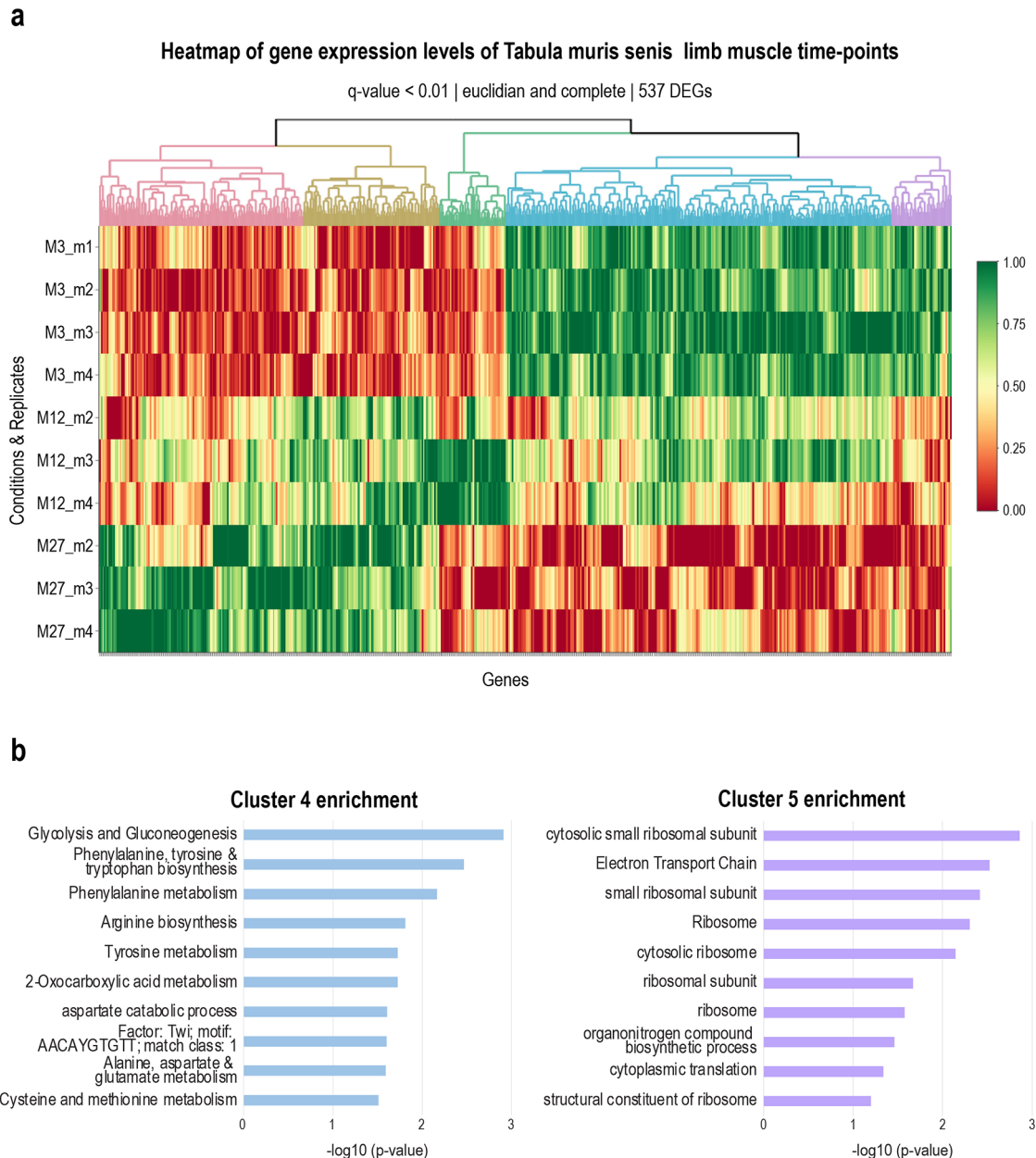
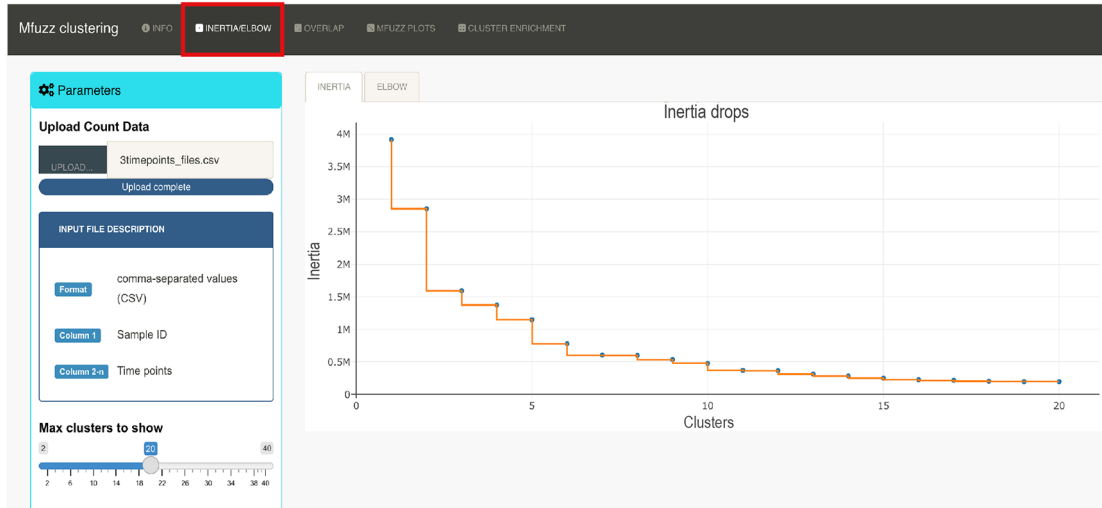


Figure 4. Heatmaply clustering of *Tabula muris senis* limb muscle data. (a) Heatmap of gene expression levels of the *Tabula muris senis* limb muscle data. Only significant genes were selected for plotting, with an FDR cutoff of 0.01, resulting in 624 DEGs. These DEGs could be clustered in 5 independent clusters, indicated by different colours in the dendrogram. (b) Enrichment analysis results for cluster 4 and cluster 5 of the limb muscle heat map. In cluster 4, processes related to energy and amino acid metabolism are enriched. Genes belonging to this process have a higher expression level in young versus old mice, suggesting more active metabolism in young muscle cells. In cluster 5, processes related to translation are enriched, whereby associated genes have higher expression levels in young muscle versus old muscle. The minus log10 of the p-value of the enriched process is plotted.

Mfuzz is used and results can be visualized (see Figure 5b). After choosing a suitable number of clusters, Mfuzz is run ten times in a loop to test the robustness of clustering results (Figure 1b). Membership lists of the ten Mfuzz clustering runs can be downloaded and checked for robustness. Plots are generated using the *mfuzz.plot* function and are also downloadable (Figure 6a). Should core clusters be unstable, this entire process can be repeated. Finally, enrichment can be done on Mfuzz cluster gene lists, using gprofiler2 (Figure 6b).

a

Selection of Mfuzz cluster numbers method: WSS



b

Control of Mfuzz cluster overlap

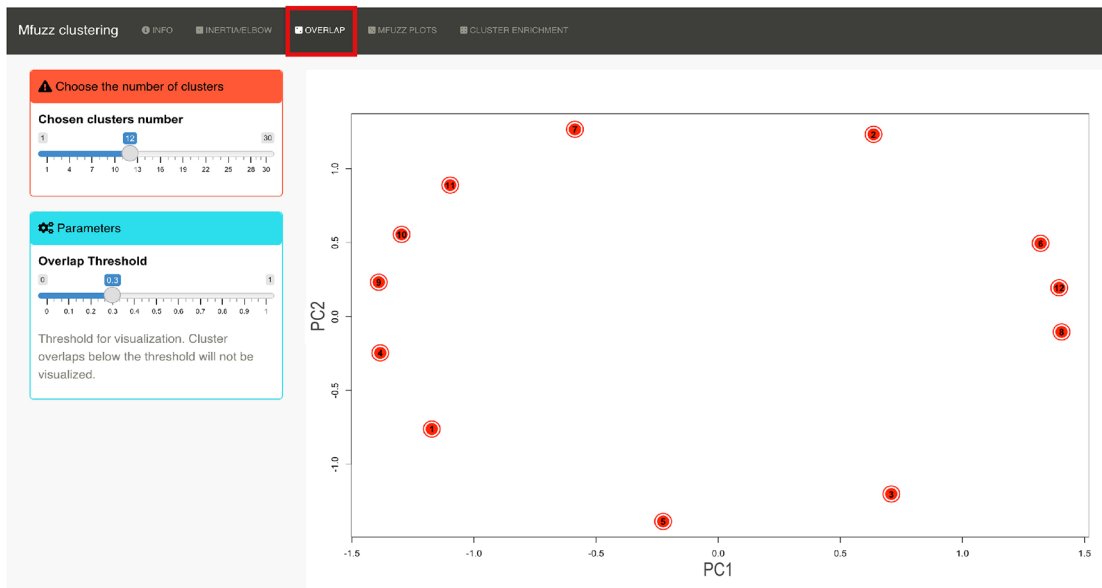


Figure 5. Pre-processing steps required for Mfuzz cluster analysis and cluster number selection. (a) Pre-clustering of data to select the cluster number for Mfuzz time-series clustering. The plot shows the *inertia* drops of the dendrogram. At 12 clusters, the inertia drop was minimal, suggesting that additional clusters would not provide better modelling of the data. (b) Control plot of Mfuzz cluster overlap. A PCA plot is performed with the selected 12 clusters, showing here that no overlap between clusters exists. Data from the developing *Drosophila* leg were chosen for demonstration purposes.

Data preparation

Tabula muris senis¹⁵ limb muscle raw read data (data accessible at NCBI GEO database,²³ accession GSE132040) were taken as is and read into RNfuzzyApp for data processing and differential expression analysis. Raw read data from developing leg muscle²⁴ (data accessible at NCBI GEO database, accession GSE143430) were first averaged over replicates before reading them into RNfuzzyApp, as Mfuzz does not accept replicates (available as Habermann, Bianca; Haering, Margaux (2021), [extended Datatables](#)).



Figure 6. Mfuzz soft clustering analysis. (a) Mfuzz clusters of *Drosophila* leg developmental RNA-seq data. Some similar patterns emerged, with expression profiles peaks early (30 h), mid-phase (50 h) or late (72 h APF (after puparium formation)). (b) Mfuzz clusters have been enriched using gprofiler2. In the parameters box, settings have to be chosen, such as the cluster number submitted for enrichment, the number of results to show, the organism, as well as the databases used for enrichment analysis. The plot on the right-hand side shows the enrichment of cluster 9. (c) Enrichment of the top 10 processes from cluster 10 and cluster 12. In cluster 10, processes related to RNA metabolism, as well as splicing are enriched, associated genes show a decrease in expression over time. In cluster 12, processes related to mitochondrial energy metabolism are enriched, with associated genes showing an increase in expression over time.

Results

RNA-seq analysis of *Tabula muris senis* bulk RNA-seq data on the ageing limb muscle

We used data from the ageing limb muscle from the *Tabula muris senis* project (GSE GSE132040¹⁵). We selected three time-points: 3 months, 12 months and 27 months. We only used samples from male mice. After data upload, we filtered for lowly expressed genes with less than 50 read counts. We then visualized raw read counts of all samples (Figure 3a). After normalization using DESeq2, we compared samples using hierarchical clustering (Figure 3b), which showed that replicates cluster together. We also performed PCA analysis and could confirm that samples from the same time-point cluster together (Figure 3c). We next subjected samples to differential expression analysis using DESeq2, comparing all time-points against each other (see *Extended data*: Tables 1a-c). We found 177 genes differentially regulated between 12 and 3 months, 873 genes differentially regulated between ages 27 and 3 months and 31 genes differentially expressed between ages 12 and 27 months when using an FDR of 0.01 and a log2FC of 10.51. Enrichment analysis of the lists of differentially expressed genes revealed terms related to translation in young versus adult mice, metabolic and extracellular organisational processes between young and aged mice, as well as between adult and aged mice (*Extended data*: Tables 1d-f).

We next used hierarchical clustering of genes to identify gene groups changing over time. After finding appropriate points to cut the dendrogram from hierarchical clustering using *heatmaply*, we found five different clusters with differing expression levels of genes (Figure 4a): two clusters with high expression levels in aged mice and low in young mice, one with high expression levels in adult mice, and two with high expression levels in young mice and low expression levels in aged mice. We subjected all clusters to enrichment analysis (*Extended data*: Tables 2a-e). For example, cluster 4 and 5, which both show high expression levels in young and low ones in aged mice, had terms related to translation as well as metabolism (energy, amino acids) highly enriched, suggesting more active translation, as well as metabolism in young muscle cells (Figure 4b).

Mfuzz soft clustering of a time-series of RNA-seq data from the developing *Drosophila* leg

We used RNA-seq data from a developmental time course of *Drosophila* leg for soft clustering using the Mfuzz pipeline included in RNfuzzyApp. We used normalized read counts from GEO²⁵ gene expression dataset GSE143430²⁴ and uploaded it to RNfuzzyApp. In brief, leg samples had been collected at three stages during pupal development (30, 50 and 72 h APF) and had been subjected to RNA-sequencing. We wanted to analyse the wild-type expression profiles of genes during these three developmental stages and to identify potentially enriched terms and pathways.

We first checked with the *inertia* method the ideal number of clusters (Figure 5a). We chose 12 clusters, as we found no significant change with cluster numbers higher than that. We next tested the overlap of clusters using the *overlap.plot* function of Mfuzz and found good separation of the 12 clusters (Figure 5b). We ran Mfuzz for clustering gene expression profiles and repeated this step 10 times. One of the resulting Mfuzz plots is shown in Figure 6a. We found expression profiles with high expression at 30 h gradually decreasing at 50 h and 70 h (clusters 9 and 10), high expression at 30 h and 50 h, which decreased at 70 h (clusters 1 and 4), expression peaks at 50 h (clusters 3 and 5), low expression levels at 50 h (clusters 2 and 7), low expression levels at 30 h, gradually increasing at 50 h and 70 h (clusters 8 and 12), as well as expression peaks at 70 h (cluster 6). We used all genes of each cluster for enrichment analysis using gprofiler2 within RNfuzzyApp (Figure 6b, *Extended data*: Table 3a-l). We found terms relevant for muscle development enriched in cluster 10 (high expression at 30 h, which gradually decreased at 50 h and 70 h), relating to mRNA metabolic processes and specifically, RNA splicing. RNA splicing has been shown to be essential for muscle cell type specification^{26,25} (Figure 6c). Cluster 12, which contained genes with increasing expression levels from 30 h to 70 h was enriched for terms related to mitochondrial function and energy production (Figure 6c). These results are in accordance with earlier observations of increasing electron transport chain components in flight muscle development.^{27,26}

Conclusions

We introduced RNfuzzyApp, an intuitive R shiny app for the complete and interactive workflows of RNA-seq, as well as time-course RNA-seq data analysis. RNfuzzyApp includes several algorithms for data normalization and differential expression analysis and offers the possibility for intuitive and interactive data visualization. All data tables and plots are downloadable by the user. While several R shiny apps exist for differential expression analysis, to the best of our knowledge, this is the first web-based, user-friendly R shiny interface for the complete workflow of time-series analysis using the soft clustering app Mfuzz, making RNfuzzyApp the first accessible tool for time-series analysis for wet-lab biologists. We demonstrated the usability of RNfuzzyApp with two examples of RNA-seq data, one from a mouse ageing study of the *Tabula muris senis* project, and one from the developing leg in *Drosophila melanogaster*.

We chose to offer several packages for normalization, as well as differential expression analysis. This allows the user to exploit several possible combinations of tools for differential expression analysis. Our choice for enrichment analysis in

this version of RNfuzzyApp fell on gprofiler2. There are many software tools available for enrichment analysis. Gprofiler2, however, is available also for non-standard model organisms. Therefore, our app can be used for organisms other than human, mouse, *Drosophila*, *C. elegans* or yeast. Moreover, gprofiler2 allows in addition ID conversion, as well as ortholog assignment and both these functions were made available in RNfuzzyApp. In future releases of RNfuzzyApp, we consider including more enrichment tools, providing a broader spectrum of data to include, such as EnrichR.²⁸

To conclude, RNfuzzyApp is an intuitive and easy to use R shiny app that was designed for experimental biologists to enable them to perform RNA-seq and time-series RNA-seq analysis without the need of coding to get a fast overview of their data, results and figures.

Data availability

Underlying data

Gene Expression Omnibus (GEO): Tabula Muris Senis: Bulk sequencing. Accession number GSE132040; <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE132040>.¹⁵

Gene Expression Omnibus (GEO): Muscle-type specific transcriptomic expression patterns in *Drosophila*. Accession number GSE143430; <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE143430>.²⁴

Extended data

Dryad: Extended data tables to Haering and Habermann, F1000Res, RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis. <https://doi.org/10.5061/dryad.8pk0p2nnd>.

This project contains the following extended data:

- Table 1a: Haering_etal_extendedDatatable_1a_Tabulamurissenis_3vs12m_DEA.txt: results of differential expression analysis (DEA) of Tabula muris senis project (GSE132040), limb muscle, 3 vs 12 months.
- Table 1b: Haering_etal_extendedDatatable_1b_Tabulamurissenis_3vs27m_DEA.txt: results of DEA of Tabula muris senis project (GSE132040), limb muscle, 3 vs 27 months.
- Table 1c: Haering_etal_extendedDatatable_1c_Tabulamurissenis_12vs27m_DEA.txt: results of DEA of Tabula muris senis project (GSE132040), limb muscle, 12 vs 27 months.
- Table 1d: Haering_etal_extendedDatatable_1d_Tabulamurissenis_3vs12m_gprofiler.txt: gprofiler results of Tabula muris senis project (GSE132040), DEA, limb muscle, 3 vs 12 months.
- Table 1e: Haering_etal_extendedDatatable_1e_Tabulamurissenis_3vs27m_gprofiler.txt: gprofiler results of Tabula muris senis project (GSE132040), DEA, limb muscle, 3 vs 12 months.
- Table 1f: Haering_etal_extendedDatatable_1f_Tabulamurissenis_12vs27m_gprofiler.txt: gprofiler results of Tabula muris senis project (GSE132040), DEA, limb muscle, 3 vs 12 months.
- Table 2a: Haering_etal_extendedDatatable_2a_Tabulamurissenis_cluster1_gprofiler.txt: gprofiler results of hierarchical clustering of Tabula muris senis project (GSE132040), limb muscle, cluster 1.
- Table 2b: Haering_etal_extendedDatatable_2b_Tabulamurissenis_cluster2_gprofiler.txt: gprofiler results of hierarchical clustering of Tabula muris senis project (GSE132040), limb muscle, cluster 2.
- Table 2c: Haering_etal_extendedDatatable_2c_Tabulamurissenis_cluster3_gprofiler.txt: gprofiler results of hierarchical clustering of Tabula muris senis project (GSE132040), limb muscle, cluster 3.
- Table 2d: Haering_etal_extendedDatatable_2d_Tabulamurissenis_cluster4_gprofiler.txt: gprofiler results of hierarchical clustering of Tabula muris senis project (GSE132040), limb muscle, cluster 4.
- Table 2e: Haering_etal_extendedDatatable_2e_Tabulamurissenis_cluster5_gprofiler.txt: gprofiler results of hierarchical clustering of Tabula muris senis project (GSE132040), limb muscle, cluster 5.

- Table 3a: Haering_etal_extendedDatatable_3a_DmLeg_cluster1_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 1.
- Table 3b: Haering_etal_extendedDatatable_3b_DmLeg_cluster2_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 2.
- Table 3c: Haering_etal_extendedDatatable_3c_DmLeg_cluster3_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 3.
- Table 3d: Haering_etal_extendedDatatable_3d_DmLeg_cluster4_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 4.
- Table 3e: Haering_etal_extendedDatatable_3e_DmLeg_cluster5_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 5.
- Table 3f: Haering_etal_extendedDatatable_3f_DmLeg_cluster6_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 6.
- Table 3g: Haering_etal_extendedDatatable_3g_DmLeg_cluster7_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 7.
- Table 3h: Haering_etal_extendedDatatable_3h_DmLeg_cluster8_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 8.
- Table 3i: Haering_etal_extendedDatatable_3i_DmLeg_cluster9_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 9.
- Table 3j: Haering_etal_extendedDatatable_3j_DmLeg_cluster10_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 10.
- Table 3k: Haering_etal_extendedDatatable_3k_DmLeg_cluster11_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 11.
- Table 3l: Haering_etal_extendedDatatable_3l_DmLeg_cluster12_gpofiler.txt: gprofiler results of mfuzz clustering of Drosophila leg dataset (GSE143430), cluster 12.
- Table 4: Haering_etal_extendedData_DmdevLeg_GSE143430_mean.txt: mean normalized read counts from GSE143430 to be uploaded for Mfuzz clustering.

Data are available under the terms of the [Creative Commons Zero “No rights reserved” data waiver](#) (CC0 1.0 Public domain dedication).

Software availability

Software available from: https://gitlab.com/habermann_lab/rna-seq-analysis-app.

Zenodo: 10.5281/zenodo.5084275 (https://zenodo.org/record/5084275#.YO_e_y0iuiik).

Source code available from: https://gitlab.com/habermann_lab/rna-seq-analysis-app.

Archived source code as at time of publication: Zenodo: RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis. <https://doi.org/10.5281/zenodo.5084275>.²⁹

License: GNU public license 3.

Author contributions

MA and BHH conceived this project. MA was solely responsible for code implementation, software development and testing. MA and BHH performed data analyses. MA and BHH wrote this manuscript.

Acknowledgements

We want to thank Fanny Chazal, Cedric Maurange for critical input and acting as test user of RNfuzzyApp and Fabio Marchiano for helpful discussions. We thank the CNRS, Aix-Marseille University, as well as the IBDM for supporting this work.

References

1. Su W, Sun J, Shimizu K, *et al.*: **TCC-GUI: a Shiny-based application for differential expression analysis of RNA-Seq count data.** *BMC Res Notes.* 2019; **12**: 133–6.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Guo W, *et al.*: **3D RNA-seq: a powerful and flexible tool for rapid and accurate differential expression and alternative splicing analysis of RNA-seq data for biologists.** *RNA Biol.* 2020: 1–14.
[PubMed Abstract](#) | [Publisher Full Text](#)
3. Zhang C, *et al.*: **iSeq: Web-Based RNA-seq Data Analysis and Visualization.** *Methods Mol Biol.* 2018; **1754**: 167–81.
[PubMed Abstract](#) | [Publisher Full Text](#)
4. Gao B, *et al.*: **Quickomics: exploring omics data in an intuitive, interactive and informative manner.** *Bioinformatics.* 2021;
[PubMed Abstract](#) | [Publisher Full Text](#)
5. Sundararajan Z, *et al.*: **Shiny-Seq: advanced guided transcriptome analysis.** *BMC Res Notes.* 2019; **12**: 432–5.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. Gadepalli VS, Ozer HG, Yilmaz AS, *et al.*: **BISR-RNAseq: an efficient and scalable RNAseq analysis workflow with interactive report generation.** *BMC Bioinformatics.* 2019; **20**: 670–7.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
7. Marini F, Linke J, Binder H: **ideal: an R/Bioconductor package for interactive differential expression analysis.** *BMC Bioinformatics.* 2020; **21**: 565–16.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Kumar L, Futschik E, Mfuzz M: **Mfuzz: a software package for soft clustering of microarray data.** *Bioinformatics.* 2007; **2**: 5–7.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
9. Kolberg L, Raudvere U, Kuzmin I, *et al.*: **gprofiler2 – an R package for gene list functional enrichment analysis and namespace conversion toolset g:Profiler.** *F1000Res.* 2020; **9**: 709.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
10. Love MI, Huber W, Anders S: **Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.** 2014; **15**: 550–21.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
11. Robinson MD, McCarthy DJ, Smyth GK: **edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.** *Bioinformatics.* 2010; **26**: 139–40.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
12. Sun J, Nishiyama T, Shimizu K, *et al.*: **TCC: an R package for comparing tag count data with robust normalization strategies.** *BMC Bioinformatics.* 2013; **14**: 219–14.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
13. Hardcastle TJ, Kelly KA: **baySeq: empirical Bayesian methods for identifying differential expression in sequence count data.** *BMC Bioinformatics.* 2010; **11**: 422–14.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
14. Galili T, O'Callaghan A, Sidi J, *et al.*: **heatmappy: an R package for creating interactive cluster heatmaps for online publishing.** *Bioinformatics.* 2018; **34**: 1600–2.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
15. Schaum N, *et al.*: **Ageing hallmarks exhibit organ-specific temporal signatures.** *Nature.* 2020; **583**: 596–602.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
16. Ashburner M, *et al.*: **Gene ontology: tool for the unification of biology.** *The Gene Ontology Consortium. Nat. Genet.* 2000; **25**: 25–9.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
17. Martens M, *et al.*: **WikiPathways: connecting communities.** *Nucleic Acids Res.* 2021; **49**: D613–21.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
18. Jassal B, *et al.*: **The reactome pathway knowledgebase.** *Nucleic Acids Res.* 2020; **48**: D498–D503.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
19. Kanehisa M, Furumichi M, Tanabe M, *et al.*: **KEGG: new perspectives on genomes, pathways, diseases and drugs.** *Nucleic Acids Res.* 2017; **45**: D353–61.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
20. Thul PJ, Lindskog C: **The human protein atlas: A spatial map of the human proteome.** *Protein Sci.* 2018; **27**: 233–44.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
21. Giurgiu M, *et al.*: **CORUM: the comprehensive resource of mammalian protein complexes-2019.** *Nucleic Acids Res.* 2019; **47**: D559–63.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
22. Wingender E, Dietze P, Karas H, *et al.*: **TRANSFAC: a database on transcription factors and their DNA binding sites.** *Nucleic Acids Res.* 1996; **24**: 238–41.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
23. Edgar R, Domrachev M, Lash AE: **Gene Expression Omnibus: NCBI gene expression and hybridization array data repository.** *Nucleic Acids Res.* 2002; **30**: 207–10.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
24. Kao S-Y, Nikonova E, Ravichandran K, *et al.*: **Dissection of Drosophila melanogaster Flight Muscles for Omics Approaches.** *J Vis Exp.* 2019: e60309.
[PubMed Abstract](#) | [Publisher Full Text](#)
25. Clough E, Barrett T: **The Gene Expression Omnibus Database.** *Methods Mol Biol.* 2016; **1418**: 93–110.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
26. Spletter ML, *et al.*: **The RNA-binding protein Arrest (Bruno) regulates alternative splicing to enable myofibril maturation in Drosophila flight muscle.** *EMBO Rep.* 2015; **16**: 178–91.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
27. Spletter ML, *et al.*: **A transcriptomics resource reveals a transcriptional transition during ordered sarcomere morphogenesis in flight muscle.** *Elife.* 2018; **7**: 1361.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
28. Kulshov MV, *et al.*: **Enrichr: a comprehensive gene set enrichment analysis web server 2016 update.** *Nucleic Acids Res.* 2016; **44**: W90–7.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
29. Margaux Hearing M, Habermann BH: **RNfuzzyApp: an R shiny RNA-seq data analysis app for visualisation, differential expression analysis, time-series clustering and enrichment analysis.** *Zenodo.*
[Publisher Full Text](#)

Open Peer Review

Current Peer Review Status:   

Version 2

Reviewer Report 15 November 2021

<https://doi.org/10.5256/f1000research.79151.r100047>

© 2021 Hahn O. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Oliver Hahn 

Department of Neurology and Neurological Sciences, Stanford University, Stanford, CA, USA

The authors are commended for addressing all my concerns. I support the indexing of the manuscript!

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, Systems Biology, Epigenetics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 25 August 2021

<https://doi.org/10.5256/f1000research.58026.r90441>

© 2021 Grigoriadis A. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Anita Grigoriadis

Cancer Bioinformatics, Cancer Centre at Guy's Hospital, King's College London, London, UK

This is an excellent paper presenting an R-shiny to analyze RNA-seq data for non-computational scientists. It provides clear instruction on how to proceed, gives several options to analyze the data, and even includes a mfuzz clustering method, to work with time-series data. I can foresee

that it will be used frequently.

Haering and Habermann have clearly stated why the software is developed. RNAseq analyses has become a standard analytical approach. This software will give non-computational scientists to explore RNAseq data on several levels.

The description of the software is sound and can be reproduced by others.

The graphical display and the results provide detailed information, so that the analysis and the data can be interpreted accurately.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: cancer bioinformatics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 16 August 2021

<https://doi.org/10.5256/f1000research.58026.r90440>

© 2021 Bacher R. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Rhonda Bacher 

Department of Biostatistics, University of Florida, Gainesville, FL, USA

This manuscript presents an R shiny application called RNfuzzyApp. The app is designed for ease-of-use for visualizing and analyzing RNA-seq datasets. The app is very nicely designed and has the potential to be very useful to scientists. However, a number of clarifications are needed.

1. What is orthology assignment mean? This should be defined more clearly in the manuscript
2. It is not clear how to install or run the app. This took me a few tries to figure out. I finally just downloaded the folder from gitlab and then did: `> library(shiny)` `> runApp("~/Downloads/rna-seq-analysis-app-master-App/App/")`. Instructions should be put on the main page of the Gitlab landing page if that is where users will be initially directed.
3. A simple example csv the users/reviewers can download to test the app would be useful. This would also help users understand the format that the app is expecting. I tried to upload the csv from the GEO and this message appeared: "Maximum upload size exceeded".
4. Improve error handling, I tried uploading data with genes on the rows and the app crashed saying it had duplicate row names. The app also crashed or would freeze when trying to upload a dataset that was a 5.5MB csv (8 samples). Are there limitations to the upload size?

I highly suggest finding someone to try the app who was not involved in the development to 'try to break it' to identify issues like these.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: genomics, RNA-seq, single-cell RNA-seq, time-series RNA-seq, package development, R shiny

I confirm that I have read this submission and believe that I have an appropriate level of

expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 04 Nov 2021

Bianca H Habermann, Aix-Marseille University, CNRS, IBDM UMR 7288, The Turing Centre for Living systems, Marseille, France

We would like to thank Dr. Bacher for reviewing our paper and their very useful comments, which helped us to improve our App, as well as the paper. We have tried to address all points. Indeed, they were greatly overlapping with the comments from Dr. Hahn and show that our user instructions were really done very poorly. We have tried to improve them greatly by creating a very detailed user menu that is available from our gitlab account.

Concerns raised:

1. What is orthology assignment mean? This should be defined more clearly in the manuscript

Our response:

Orthology assignment means that we can find the orthologs of another species using the gprofiler package. We have now tried to clarify in the text what we mean and provide a detailed user manual on how to use this function.

2. It is not clear how to install or run the app. This took me a few tries to figure out. I finally just downloaded the folder from gitlab and then did: > library(shiny) > runApp("~/Downloads/rna-seq-analysis-app-master-App/App/"). Instructions should be put on the main page of the Gitlab landing page if that is where users will be initially directed.

Our response:

We now provide on the gitlab readme page the instructions on how to install the app. We also tried to remove any installation error that we encountered on machines with a very 'basic' R environment. We also provide a detailed user manual, which again includes installation instructions and detailed instructions on how to use the app. The user manual is available from the RNfuzzyApp gitlab page.

3. A simple example csv the users/reviewers can download to test the app would be useful. This would also help users understand the format that the app is expecting. I tried to upload the csv from the GEO and this message appeared: "Maximum upload size exceeded".

Our response:

We now provide test files for both functions of RNfuzzyApp, differential expression analysis, as well as Mfuzz clustering, on our gitlab account (in App/test_files).

The "maximum upload size exceeded error" could be due to size limits within R shiny, which we have tried to circumvent. The size limit of R shiny can be changed with the following command: `options(shiny.maxRequestSize = n*1024^2)`. We have changed this to 100MB, which should be sufficient for most datasets. In case of larger datasets, this would have to be set individually by the user.

4. Improve error handling, I tried uploading data with genes on the rows and the app crashed saying it had duplicate row names. The app also crashed or would freeze when trying to upload a dataset that was a 5.5MB csv (8 samples). Are there limitations to the upload size?

Our response:

See our response above: it seems that R shiny has a rather low size limit for data upload, which can be changed with the command: `options(shiny.maxRequestSize = n*1024^2)`. We have set it to 100MB, which should suffice for most datasets.

5. I highly suggest finding someone to try the app who was not involved in the development to 'try to break it' to identify issues like these.

Our response:

We have tried this now and hopefully have found all obvious bugs of RNfuzzyApp.

Competing Interests: No competing interests were disclosed.

Reviewer Report 11 August 2021

<https://doi.org/10.5256/f1000research.58026.r90442>

© 2021 Hahn O. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Oliver Hahn 

Department of Neurology and Neurological Sciences, Stanford University, Stanford, CA, USA

RNA-sequencing (RNA-seq) is a widely used technology across a range of biomedical fields. While the method itself - that is, the data generation step - is relatively easy to execute by wet lab researchers or can be outsourced into genomic core facilities, the analysis of the resulting data remains a challenge without substantial coding experience. This is particularly the case for experimental designs involving more than two conditions (i.e. treatment vs. control) as it is the case for time course data or dose-response paradigms. In this study, the authors present a new, graphical user interface (GUI) software package, RNAfuzzy App, with the aim to provide a single environment for standard quality control, normalization, differential expression, clustering and functional enrichment operations.

The authors re-analyzed two time course datasets from mouse and fruitfly to demonstrate the basic steps and workflows that can be executed with RNAfuzzy App. The package is distributed as a single R shiny app, that the authors design with the intent for easy installation and setup. The demonstrated workflows encompass known steps in RNA-seq analysis including filtering, quality control (via PCA/hierarchical clustering), normalization with several state-of-the-art methods and differential expression involving the accepted software packages Deseq2 and edgeR. The authors highlight the implementation of a soft-clustering workflow involving iterative runs of the Mfuzz

algorithm (for analysis of time course trajectories), as well as diagnostic plots that are aiding the user towards correct selection of adjustable workflow parameters. Finally, the authors use the clustering results as input for a functional enrichment workflow, that includes reference data from a range of pathway and transcription factor databases.

The authors have pursued the development of a highly relevant software package that provides a very comprehensive set of functions and tools. Current RNA-seq data is generated at impressive scale yet remains frequently under-analyzed due to limited coding experience of the experimenters that generate the data. The implementation of elaborative clustering methods and diagnostic plots to aide the user in deciding for the right settings is particularly commendable. Similarly relevant is the implemented functional enrichment workflow, that is well-suited within this package, instead of just finishing the analysis with a set of gene lists that needs to be interpreted elsewhere. The workflows are based on accepted practices and are in agreement with the field's 'standard' analyses. This is important and commendable as several other tools provide misleading workflows that experimental biologists are unaware of.

However, the software package is currently challenging to install, several functions throw error messages or cause a crash of the app altogether. There is - at least as far as I can tell - very limited documentation (none regarding installation) on how to prepare input data or operate individual workflows. I provide specific details below. I consider myself proficient in bioinformatics in general and RNA-seq data analysis in R in particular, and I struggled and ultimately failed to operate this app.

While I do acknowledge that the authors are not primarily responsible for the individual challenges the user may face - there are always unforeseeable issues - it needs to be noted that a significant emphasis of the manuscript is on the 'user-friendly' (mentioned at least six times in the manuscript) features of RNfuzzy. It also makes the software, which is the heart of this manuscript, difficult to review. There is no link to an already-running 'reviewer web version' or a built-in function that loads a tutorial dataset (like the ones used in the manuscript). I have tried to load both the published datasets and a bulk RNA-seq dataset of my own but could not progress beyond loading and quality control due to frequent errors or crashes. That was probably due to some mis-formatting of the input data but without clear guidance on how to prepare the count matrix, or at least an already-formatted dataset, I do not see how untrained experimental biologists will be able to efficiently interact with this app.

If the authors can improve on these aspects and after I have been able to verify the functionality of the software (at least with data provided by the authors or within a 'reviewer version'), I would support the indexing of the manuscript.

In addition, there are multiple points regarding the analysis and statistical methods that would be required to be addressed as well.

Please find my detailed comments below:

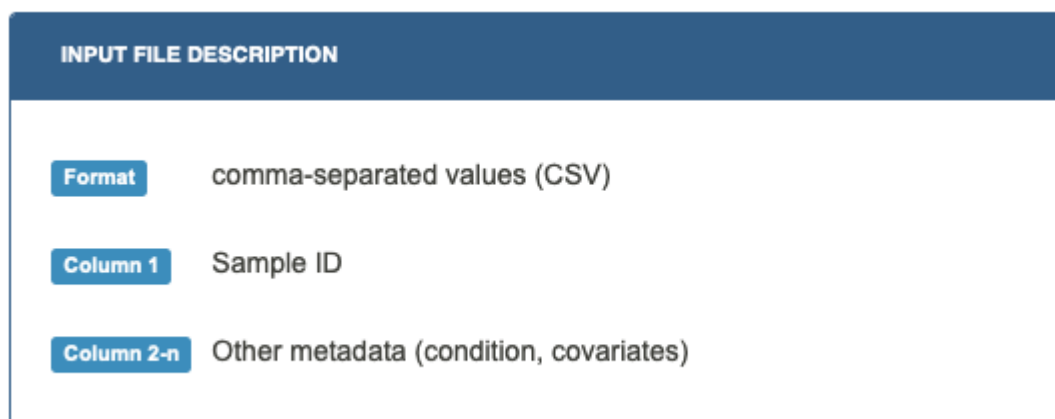
Major concerns:

1. Currently there is no guidance on how to install and launch the app (at least to my knowledge). The authors state on page 3 that 'Installation instructions are also available.' This is not correct - the adjacent github provides no instructions on how to install

the app or how to launch it. Most users do not know how to start a shiny app in their own desktop environment. In addition, when launched, the app throws an error message as the devtools package is not properly loaded and so the 'install_github' function is missing. Only when opening R, loading devtools and then launching the app, the setup (including auto-install) is working.

2. There is very limited information on how to prepare the input count matrix. The authors state that a csv matrix is required but there is no information on how genes or sample names should be provided. I assume as row- and column-names, respectively, but that is not clear. Also, should the genes receive an extra header (i.e. a column name that says 'genes' or something like that)? Should genes be provided as symbols, ensmebl ids...?

Neither the github or the app provide much information herein. E.g. the 'input file description' within the app states:



INPUT FILE DESCRIPTION	
Format	comma-separated values (CSV)
Column 1	Sample ID
Column 2-n	Other metadata (condition, covariates)

How are all sample IDs provided in just one column? How should other covariates be supplied here? I could not find information on how to use covariates in the provided analysis workflows, neither in the manuscript nor within the app.

If the input data is not provided correctly, I assume that many analyses will not work or just crash. So this input step is critical. To prevent failures here, the authors could provide a) a detailed description on the input format and/or b) the option to load a template/tutorial dataset.

3. While I was able to load a count matrix into app and run e.g. the hierarchical clustering, all other functions caused errors (e.g. Count distribution, PCA) or a complete crash (normalization, differential expression using DESeq2). I assume this to be due to some problems with the input data, but I cannot verify that as no example/tutorial dataset was provided.

Minor concerns:

1. The authors allow the filtering of low count genes. Could the authors provide references on the legitimacy of this? Packages like Deseq2 require an unaltered count matrix to properly build up e.g. independent filtering, dispersion estimation, etc. and are therefore warranting

against any prior filtering or data manipulations.

2. Is the PCA plot calculated on the whole count matrix, or only the filtered one? Is the data somewhat transformed (rlog, varianceStabilizing transformation?) before running the PCA? The PCA function is currently running very slowly and may benefit from some sort of feature selection (e.g. top 10,000 expressed genes).
3. Why do the authors limit their time course analysis to just three timepoints of the published tabula muris datasets? The dataset contains up to 10 timepoints and it may be more convincing to demonstrate the usefulness of RNfuzzy on as many datapoints as possible.
4. The enrichment analysis is currently only allowing 'Only annotated genes' or 'All known genes' as statistical domain scope (i.e. the 'background set'). It is common within the RNA-seq field to rather use a set of 'expressed genes' to compensate for biases in e.g. tissue-specific expression. DEGs from neuronal tissue will inevitably be enriched for neuronal functions when the whole genome is used as background. To prevent wet lab researches to be deceived by such results, a user-provided expressed set (e.g. normalized counts of 1+) or one that is directly loaded from the input matrix, would be beneficial.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, Systems Biology, Epigenetics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 04 Nov 2021

Bianca H Habermann, Aix-Marseille University, CNRS, IBDM UMR 7288, The Turing Centre for Living systems, Marseille, France

We thank Dr. Hahn for their very critical and helpful comments. Indeed, we failed to provide a thorough user manual to help inexperienced users with the installation procedure, as well as the workflow itself. As Dr. Hahn correctly points out - this software should be user-friendly and not challenge the user with problems, already at the installation step. We hope that with the help of Dr. Hahn's comments, we could significantly improve the user-friendliness of RNfuzzyApp and would like to thank him here for his valuable input.

In general, we do not provide a running web-server. Our idea was to provide an app that can be locally installed, also because of limitations in server availability on our side. However, we have now tried to significantly improve the installation instructions, as well as corrected several issues of RNfuzzyApp during the installation procedure, which arose due to the fact that many of us make daily use of some of the tools required during automatic installation of RNfuzzyApp. We also realized, when installing it on a 'brand-new' machine that has not seen many of the required packages before that quite a few of them need to be installed in addition. We have now added all required packages to the install procedure and thoroughly tested it. We therefore hope that the software installs smoothly. However, it should be noted that depending on the users' environment, still some errors could occur due to missing pre-installed packages that should normally be present within the R environment. The best advice is to read the error messages during installation within R, which should list the packages that still need installing.

Our responses to the **major** and **minor** concerns of Dr. Hahn:

Major concerns:

1. Currently there is no guidance on how to install and launch the app (at least to my knowledge). The authors state on page 3 that 'Installation instructions are also available.' This is not correct - the adjacent github provides no instructions on how to install the app or how to launch it. Most users do not know how to start a shiny app in their own desktop environment. In addition, when launched, the app throws an error message as the devtools package is not properly loaded and so the 'install_github' function is missing. Only when opening R, loading devtools and then launching the app, the setup (including auto-install) is working.

Our response:

We now provide detailed installation instructions in the user manual that can be found in RNfuzzyApp git repository, including how to install shiny itself (should it be missing), how to call the App and what other errors might occur, e.g. when a Mac is used. We have also tried to remove any error coming from missing packages, so installation should now work in most R environments.

2. There is very limited information on how to prepare the input count matrix. The authors state that a csv matrix is required but there is no information on how genes or sample names should be provided. I assume as row- and column-names, respectively, but that is not clear. Also, should the genes receive an extra header (i.e. a column name that says 'genes' or something like that)? Should genes be provided as symbols, ensembl ids...?

Neither the github or the app provide much information herein. E.g. the 'input file description' within the app states:

How are all sample IDs provided in just one column? How should other covariates be supplied here? I could not find information on how to use covariates in the provided analysis workflows, neither in the manuscript nor within the app.

If the input data is not provided correctly, I assume that many analyses will not work or just crash. So this input step is critical. To prevent failures here, the authors could provide a) a detailed description on the input format and/or b) the option to load a template/tutorial dataset

Our response:

a) We now provide all sample data with the App, in a folder called test_files that can be found at the RNfuzzyApp gitlab site and is downloaded when the software is downloaded.

b) Covariates was a misleading term, in fact, we only allow read counts (from any type of experiment). We have removed it from the Input File Description.

c) The genes can be provided as any valid identifier, e.g. Symbols, ENSEMBL IDs etc.

3. While I was able to load a count matrix into app and run e.g. the hierarchical clustering, all other functions caused errors (e.g. Count distribution, PCA) or a complete crash (normalization, differential expression using DESeq2). I assume this to be due to some problems with the input data, but I cannot verify that as no example/tutorial dataset was provided.

Our response:

While we cannot tell why RNfuzzyApp crashed, we have now tried the app on several of our machines and were not able to get it crashing during normalization or differential expression analysis. We hope that with the provided input sample data, this error will be solved. We also now show in detail in the user manual how to load the data, normalize it, perform differential expression analysis and visualize the results.

It should be noted at this point that, when uploading samples from three different conditions or time-points, normalization and initial differential expression analysis will be done over all samples. In order to compare only two conditions, and hence profit from the MA and Volcano plot visualizations, we have now introduced the 'Filter data' menu, where the user can choose which conditions (or time-points) to compare.

We have also simplified the sample assignment. The app now automatically recognizes replicates from a given condition, provided that the header of the columns of the respective data are formatted correctly (condition1_rep1, condition1_rep2, ...).

Minor concerns:

1. The authors allow the filtering of low count genes. Could the authors provide references on the legitimacy of this? Packages like Deseq2 require an unaltered count matrix to properly build up e.g. independent filtering, dispersion estimation, etc. and are therefore warranting against any prior filtering or data manipulations.

Our response:

In fact, in the DESeq2 vignette, filtering is suggested (see the DESeq2 vignette, which states (<http://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#pre-filtering>):

'Pre-filtering

While it is not necessary to pre-filter low count genes before running the DESeq2 functions, there are two reasons which make pre-filtering useful: by removing rows in which there are very few reads, we reduce the memory size of the dds data object, and we increase the speed of the transformation and testing functions within DESeq2. Here we perform a minimal pre-filtering to keep only rows that have at least 10 reads total. Note that more strict filtering to increase power is automatically applied via independent filtering on the mean of normalized counts within the results function.'

However, the user can choose whether they want to do low read count filtering or not.

2. *Is the PCA plot calculated on the whole count matrix, or only the filtered one? Is the data somewhat transformed (rlog, varianceStabilizing transformation?) before running the PCA? The PCA function is currently running very slowly and may benefit from some sort of feature selection (e.g. top 10,000 expressed genes).*

Our response:

a) PCA analysis of raw read counts is done using all the genes.

b) PCA analysis of normalized data can be done once differential expression analysis has been performed. In this case, the top genes, as well as the FDR-cutoff can be chosen by the user.

In both cases, the data are log transformed using the log1p() function.

3. *Why do the authors limit their time course analysis to just three timepoints of the published tabula muris datasets? The dataset contains up to 10 timepoints and it may be more convincing to demonstrate the usefulness of RNfuzzy on as many datapoints as possible.*

Our response:

We have limited our analysis here on only very few time-points for easier understanding of the tool and also to provide small test data files. Indeed, Mfuzz clustering will especially be useful to analyse larger time-series. However, this will inevitably lead to large test data files, as well as large figures. As an example, for our dataset on flight muscle development published in (doi: [10.7554/eLife.34058](https://doi.org/10.7554/eLife.34058)), we have created 40 clusters from 8 time-points using Mfuzz. Using only three time-points makes the demonstration of the software simply easier.

We would also like to note that many researchers will only have two conditions to compare using the differential expression analysis function, which is equally useful in RNfuzzyApp.

4. *The enrichment analysis is currently only allowing 'Only annotated genes' or 'All known genes' as statistical domain scope (i.e. the 'background set'). It is common within the RNA-seq field to rather use a set of 'expressed genes' to compensate for biases in e.g. tissue-specific expression. DEGs from neuronal tissue will inevitably be enriched for neuronal functions when the whole genome is used as background. To prevent wet lab researchers from being deceived by such*

results, a user-provided expressed set (e.g. normalized counts of 1+) or one that is directly loaded from the input matrix, would be beneficial.

Our response:

While we agree in principle with the idea of taking only the expressed genes as a background list, we would prefer to currently not make this possibility available. We would like to note that many enrichment tools do not provide this functionality and either annotated genes or all known genes from the organism's genome are used as background lists. I do also think that when comparing DEGs from two conditions of a neuronal tissue (e.g. WT and knock-out), or two time-points from the same tissue, not only neuronal-specific terms will appear, so I do not think that the user will be deceived by the results. In our example of the mouse muscle of mice from 3, 12 and 27 months, for instance, muscle terms are not strongly enriched between the different time-points.

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research