tappAS installation manual

Authors: Pedro Salguero & Ángeles Arzalluz-Luque

Preliminary note

This document contains all the necessary steps to successfully install tappAS, including:

- A **checkpoint** containing a series of steps so that users can make sure that the application runs correctly.

- A **troubleshooting section** at the end, containing info on known problems that users can encounter during installation. If you encounter an **error during installation**, visit this section.

Resources/links

We first recommend going through tappAS **README** file on the GitHub repository (https://github.com/ConesaLab/tappAS/blob/master/README.md), which contains a general description of the application and all relevant links.

- tappAS GitHub repository (source code): https://github.com/ConesaLab/tappAS

- tappAS GitHub Releases section (application download): https://github.com/ConesaLab/tappAS/releases

- tappAS website (brief documentation and app description): https://app.tappas.org/

We also recommend reading the **tappAS paper**, recently published in Genome Biology:

de la Fuente, L., Arzalluz-Luque, Á., Tardáguila, M. et al. tappAS: a comprehensive computational framework for the analysis of the functional impact of differential splicing. Genome Biol 21, 119 (2020). https://doi.org/10.1186/s13059-020-02028-w

Application Overview and Requirements

tappAS is a Graphical User Interface (GUI) application written in Java. It uses SQLite for its database management and R, along with some of its statistical packages, for data analysis. The software is designed to be run locally but, like most applications dealing with large data sets, the more computational power and memory available, the faster the application will run.

Listed below are the general requirements for running tappAS:

Computer Hardware

Minimum of 4 cores (multi-core CPU or single core CPUs).
Minimum of 8 GBs of RAM.
Minimum of 20 GBs of available, unused, disk space.

***NOTE:** tappAS will run under insufficient resources, but be aware that the application may become sluggish or unusable when dealing with large datasets.

Operating System

Linux Ubuntu (recent version)
Mac OS X (recent version)
Windows (versions 7, 8 or 10)

***NOTE:** other Linux/Unix desktop environments may work but HAVE NOT BEEN TESTED.

Java client

Java (Oracle JRE or JDK)* **version 8 update 40 or later** – versions 9 and later HAVE NOT BEEN TESTED or shown to fail running tappAS.

***NOTE** for Ubuntu/Linux users: tappAS does **NOT** work with OpenJDK.

Software

R-script version 3.6.1 or later.

R packages

When opening the application, if any of the required R packages are missing, a pop-up message will show asking whether they should be automatically installed. If click OK, tappAS will install them internally and let you know when the process is finished.

For first-time users, however, there may numerous packages to be installed and the process could be very time-consuming and/or fail.

To avoid this, we recommend to automatically installing all R package dependencies by entering the following in the R terminal:

```
source("http://app.tappAS.org/resources/downloads/tappAS_packages.R")
```

tappAS main dependencies for analysis and graphics are:

DEXSeq - recent version (Bioconductor)
maSigPro - recent version (Bioconductor)
edgeR - recent version (Bioconductor)
NOISeq - recent version (Bioconductor)
goseq - recent version (Bioconductor)
mdgsa - recent version (Bioconductor)
ggplot2 - recent version (CRAN)
MASS - recent version (CRAN)
plyr - recent version (CRAN)
VennDiagram - recent version (CRAN)
Devtools - recent version (CRAN)
ggrepel - recent version (CRAN)
cowplot - recent version (CRAN)
tidyverse - recent version (CRAN)
GOglm - version 0.4.2. or later (GitHub)

Package installation notes

- Make sure to check to see if you already have some of the software and packages installed before trying to install them.
- Also, if you have already installed R on your computer, you may already have some of the required packages.
- Most statistical R packages are installed from https://www.bioconductor.org/
- If you are using Ubuntu, it is a lot simpler to use the Ubuntu Software Center application or Advanced Packaging Tool (APT) command line tool to install R and RStudio.
- GOglm is currently not available via Bioconductor. In case that installation via the installation script fails, you can enter the following in the R console:

```
install.packages("devtools")
library(devtools)
install_github("gu-mi/GOglm")
```

Application Installation

1. Install **Java 8**.

2. Install **R, Rscript** and all required packages (see above).

3. Finally, to install and run **tappAS**, download the application file's, **tappas.jar**, from

the Releases section of tappAS GitHub repo into your computer:

 https://github.com/ConesaLab/tappAS/releases

Please make sure to download the application's LATEST VERSION.

Installation Checkpoint

To make sure that your tappAS installation completed successfully, follow the steps below:

1. **Run** the application by double clicking on the **tappas.jar** file. If nothing happens, open a command terminal, **cd** into the uncompressed tappAS folder location, and run the following command:

   ```
   java -jar tappas.jar
   ```

   In order for this to work, Java must be installed and associated with JAR files. To check whether you have correctly installed Java, run the following command:

   ```
   java –version
   ```

   For which the output should be similar to this:

   ```
   java version "1.8.0_212"
   Java(TM) SE Runtime Environment (build 1.8.0_212-b10)
   Java HotSpot(TM) 64-Bit Server VM (build 25.212-b10, mixed mode)
   ```

   NOTE: For some Linux distributions, we have found that Open JDK remains the default Java environment even after Oracle Java installation. This will show up when running the `-version` command. In this case, you'll need to **switch your JRE from OpenJDK to Oracle Java 8**, and put Java 8 into your PATH.

2. The first time you run the application or a new version, you will be asked to **select a location** for the application's data folder, **tappASWorkspace**.

   This is where the application will store all the application data files, including your data analysis results. A common place would be your home directory.

3. Once you have selected the path to the tappASWorkspace and managed to open tappAS, you will see this screen:
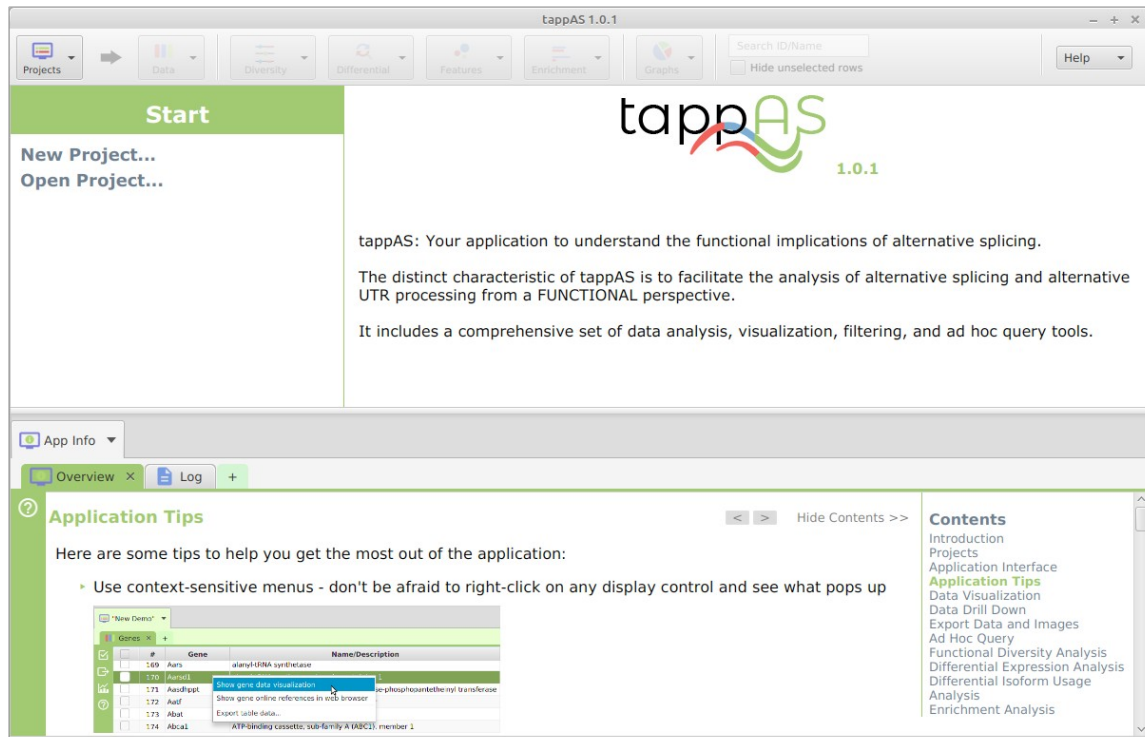
4. Click on the "New project…" option. The following window will pop-up:

5. Insert the following information:

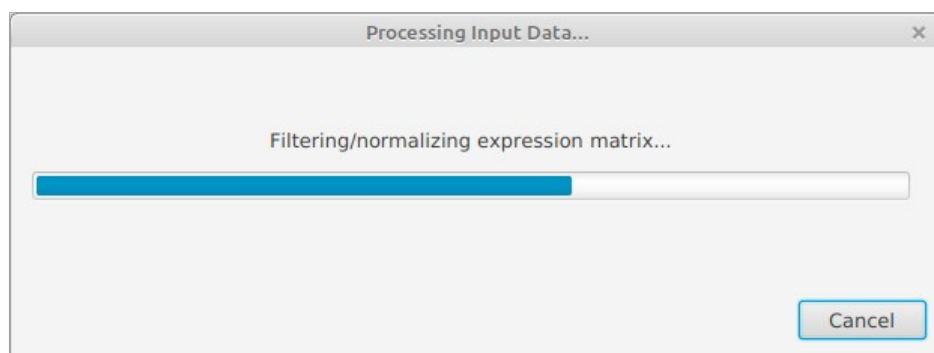- **Project Information**. Name: Demo

- **Annotation Data.** Species: *Mus musculus.*

6. Under "Use application's annotation feature file", select the "Mus musculus Demo 1" option and leave the rest of the options as default:
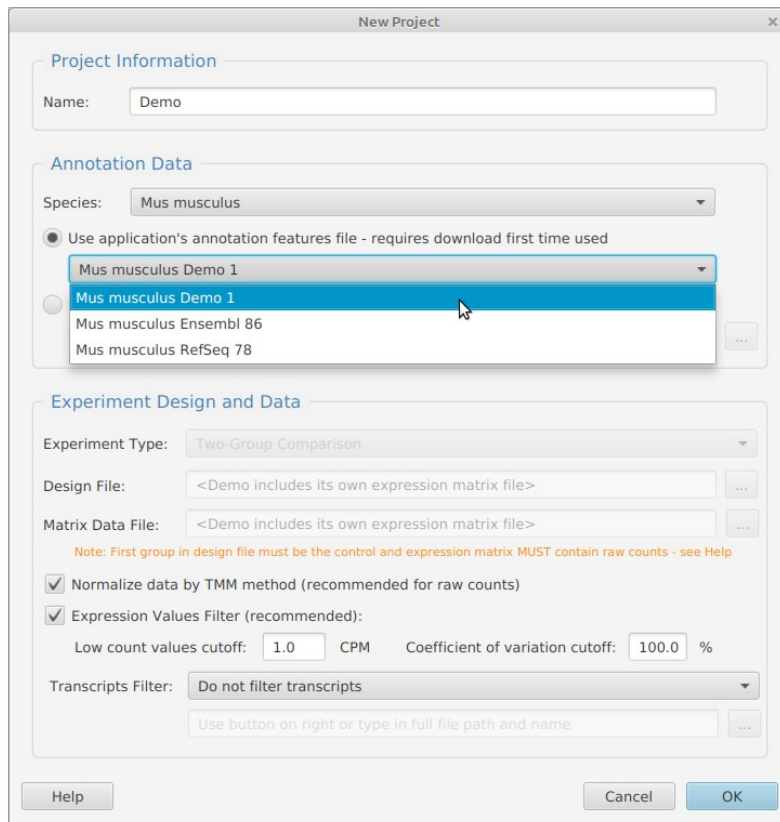
7. Click "OK" to start creating the project. The following window should appear, and start showing different steps of project creation and their progress:

8. After this step has completed, and if no error messages have appeared, you will have succesfully created a Demo project in tappAS, and see the following window, which corresponds to tappAS GUI:
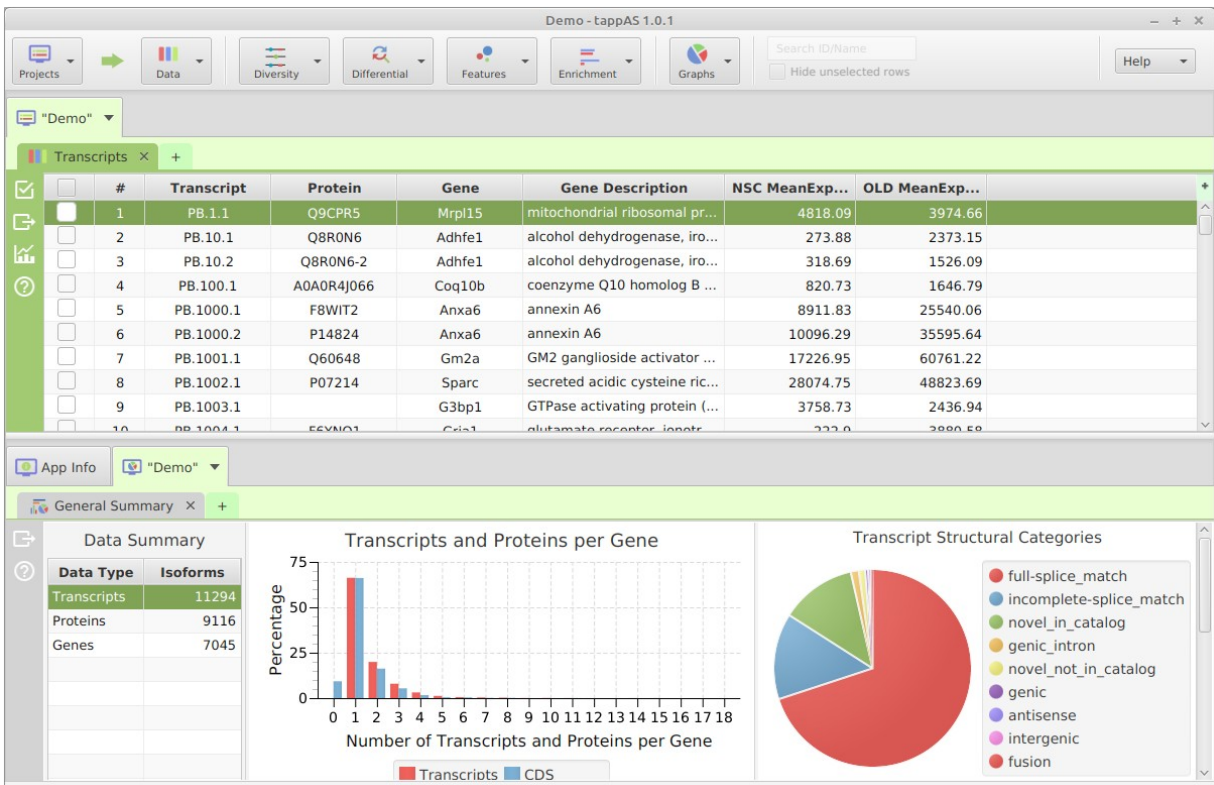
9. If you have managed to create a Demo project, tappAS is ready to be used with your own data. Alternatively, **you can use the demo data set to learn your way through the application.**

If you encountered an error, see the **troubleshooting section** below.

Installation Notes & troubleshooting

- If you are using a Windows OS, tappAS will ask you to select your RScript.exe* path. It usually can be found at following pathway:

  ```
  C:\Program Files\R\R-version\bin\RScript.exe
  ```

  *If you have not installed R, you can download it at: https://www.r-project.org/.

- On a Windows OS, you may get the following message:

  WARNING: Could not open/create prefs root node Software\JavaSoft\Prefs at root 0x80000002. Windows RegCreateKeyEx(...) returned error code 5.

  This is a known Java bug, and you can easily search for the solution online using the message. A working solution is posted at https://stackoverflow.com/questions/16428098/groovy-shell-warning-could-not-open-create-prefs-root-node. It requires the creation of an entry in the Windows registry using the Windows Registry Editor, RegEdit.

- On Linux and Mac, you may encounter the following error when running `java -jar tappas.jar`:

  Error: Could not find or load main class tappas.Tappas

  This error has been previously reported by users, and we believe this could be due to your OS not being able to use Java 8 to execute the tappas.jar file, or to switch from OpenJDK to Oracle Java. One way around this would be to download Java 8 and run the JAR file as follows:

  Linux:

  ```
  wget -O jre-8u251-linux-x64.tar.gz
  https://javadl.oracle.com/webapps/download/AutoDL?
  BundleId=242050_3d5a2bb8f8d4428bbe94aed7ec7ae784

  tar xvf jre-8u251-linux-x64.tar.gz

  ./jre1.8.0_251/bin/java -jar tappAS.jar
  ```

Mac:

```
wget https://javadl.oracle.com/webapps/download/GetFile/1.8.0_251-
b08/3d5a2bb8f8d4428bbe94aed7ec7ae784/unix-i586/jre-8u251-macosx-x64.tar.gz

tar xvf jre-8u251-macosx-x64.tar.gz

./jre1.8.0_251.jre/Contents/Home/bin/java -jar tappas.jar
```

This avoids installing Java entirely, and has been found to help when running into the error with Ubuntu Java packages. This solution was provided by Sam Hunter.

- On Mac, you may run into issues if you already have another version of Java installed. Typically, the `java` command may run the pre-existing version of Java by default (for instance, Java 14), instead of Java 8. You can find a solution for running multiple Java versions on Mac on the following threads:

  - To change the default Java version in your Mac OS X, see:
    https://stackoverflow.com/questions/21964709/how-to-set-or-change-the-default-java-jdk-version-on-os-x

  - To set aliases for the different Java versions in your Mac OS X, see:
    https://stackoverflow.com/questions/26252591/mac-os-x-and-multiple-java-versions

  - To detect the installed version of the Java JRE in Mac OS X, see:
    https://docs.oracle.com/javase/8/docs/technotes/guides/install/mac_jre.html#A1096856