Differential Gene Expression Analyses with R

Ariel Rodríguez
Zoological Institute, University of Veterinary Medicine Hannover
Bünteweg 17, 30559, Hannover
email: ariel-rodriguez@tiho-hannover.de

2022-03-24

Introduction

In this tutorial we will learn to conduct a gene expression analysis to identify statistically significant differences in expression between sample groups. The practice will use the data generated during a previous study of our lab. In this study, we compared the expression estimates obtained from skin RNA extracts of 15 individuals of 3 different color morphs (red, green and blue) of the Strawberry Poison Frog, *Oophaga pumilio* found in Bocas del Toro archipelago in Panama (Rodríguez, Mundy, Ibáñez, & Pröhl, 2020). The expression quantification was obtained from the alignment of illumina RNASeq reads (SRA bioproject number: PRJNA610154) against the *Oophaga pumilio* reference transcriptome (TSA accession number: GIKS00000000) with Kallisto software. Kallisto provides a fast and statistically sound quantification of gene expression by pseudoaligning the reads against a reference transcriptome while accounting for alignment uncertainty using a bootstrap procedure (Nicolas L Bray & Pachter, 2016).

Differential gene expression analysis with R

The statistical tests of differential expression will be conducted with the R package 3D RNA-Seq for R (R Core Team, 2021), which needs to be installed in your computer together with Rtools.

3D RNA-seq integrates state-of-the-art differential expression analysis tools and adopts best practice for RNA-seq analysis. The program is easy-to-use, flexible and yet powerful enough to identify differential gene/transcript expression, differential alternative splicing and differential transcript usage (Guo et al., 2021). It is available within Bioconductor and we start by installing some necessary package dependencies:

```
## Install dependencies
cran.package.list <- c("shiny","shinydashboard","rhandsontable","shinyFiles",
                       "shinyjs","shinyBS","shinyhelper","shinyWidgets",
                       "magrittr","DT","plotly","ggplot2","eulerr",
                       "gridExtra","grid","fastcluster","rmarkdown","base64enc",
                       "ggrepel","zoo","gtools")
for(i in cran.package.list){
   if(!(i %in% rownames(installed.packages()))){
     message('Installing package: ',i)
     install.packages(i,dependencies = T)
   } else next
}

###---> Install packages from Bioconductor
bioconductor.package.list <- c('tximport','edgeR','limma','RUVSeq',
```

```
                                'ComplexHeatmap','rhdf5')
for(i in bioconductor.package.list){
  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
  if(!(i %in% rownames(installed.packages()))){
    message('Installing package: ',i)
    BiocManager::install(i,dependencies = T)
  } else next
}
```

Once all dependencies have been properly installed, we can easily install 3DRNASeq itself with this code:

```
## use devtools R package to install ThreeDRNAseq from Github
###---> If devtools is not installed, please install
if(!requireNamespace("devtools", quietly = TRUE))
  install.packages('devtools',dependencies = TRUE)

###---> Install ThreeDRNAseq
if(!requireNamespace("ThreeDRNAseq", quietly = TRUE))
  devtools::install_github('wyguo/ThreeDRNAseq')
```

Importing Kallisto quantification results and sample metadata into R.

For this analysis we will need the following input files: the results of the Kallisto quantification (compressed into a zip-file), a .csv table with the transcripts to gene mapping, and a .csv table with the experimental design of the study. The table of the experimental design is as follows:

Table 1: Experimental design table

sample	color
AL1	red
AL2	red
AL3	red
AL4	red
AL5	red
PO1	green
PO2	green
PO3	green
PO4	green
PO5	green
AG1	blue
AG2	blue
AG3	blue
AG4	blue
AG5	blue

The gene-to-transcripts mapping table contains the results of the annotation of the 140,684 transcripts expressed in the skin of *Oophaga pumilio*. The first 12 lines of this table look like this:

Table 2: Transcripts-to-gene map table

transcript	gene
asmbl_100001	tefm_1
asmbl_100002	tefm_1
asmbl_100003	tefm_1
asmbl_100004	tefm_1
asmbl_100007	atad5
asmbl_100008	atad5
asmbl_100010	atad5
asmbl_100012	atad5
asmbl_100013	atad5
asmbl_100036	pcpi_1
asmbl_100041	spint1_2
asmbl_100046	col6a3

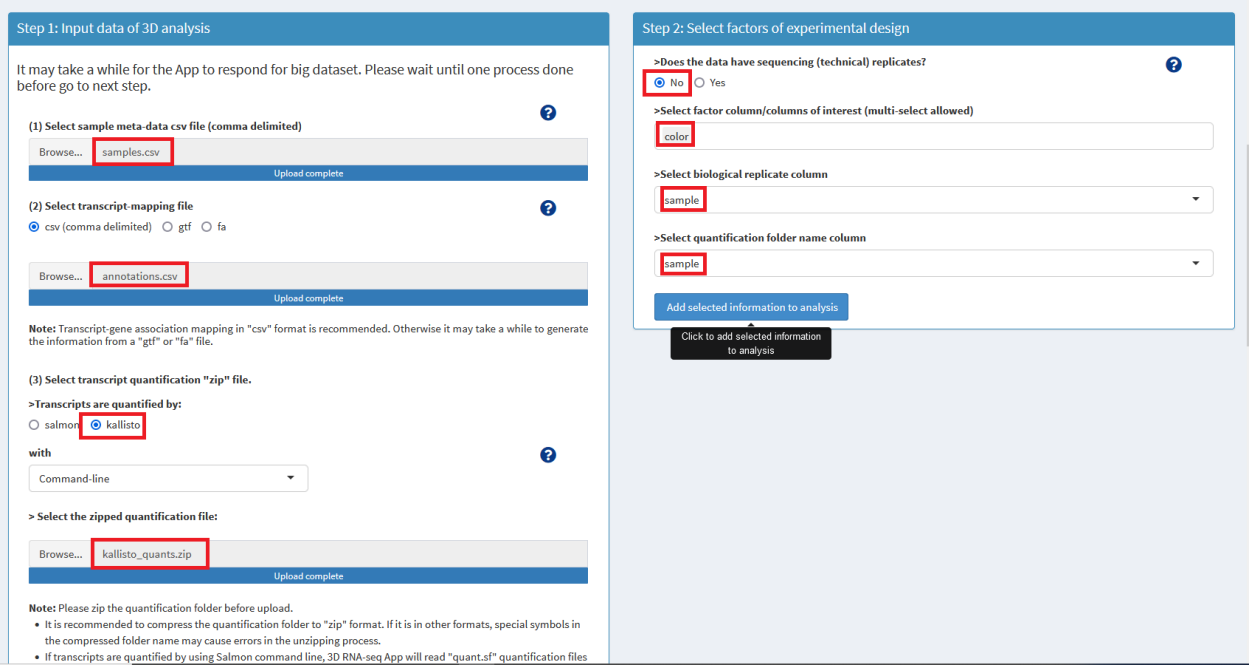
You can download the three required input files from this *Google Drive link*. Once you have these files in your computer, you can start a 3D RNA-Seq session.

The following code chunk will open an interactive interface to select the desired working directory where all the output will be redirected to. Subsequently the 3DRNASeq app will pup-up on your internet explorer. Follow the instructions therein to import the data and configure the analyses:

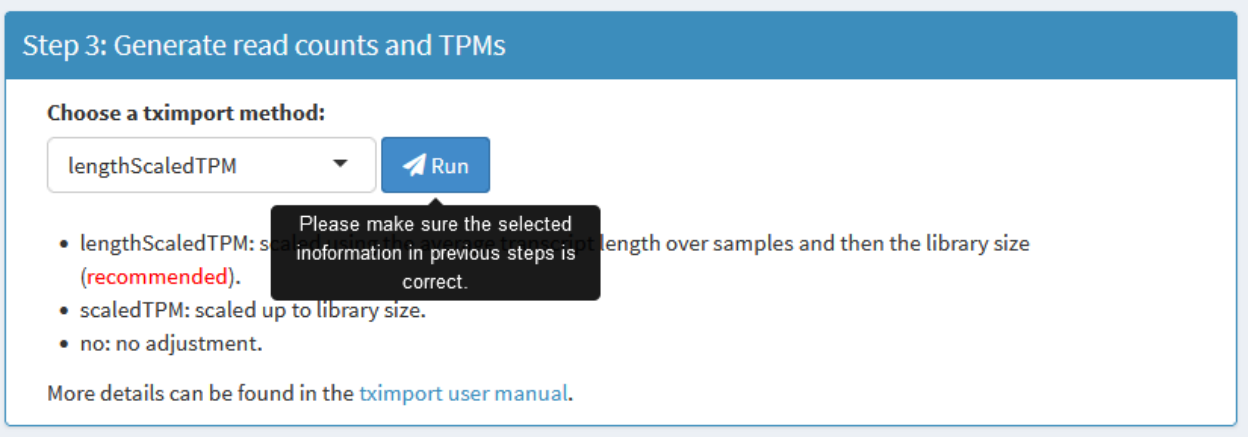
```
library(tcltk)
dir<-tk_choose.dir(getwd(), "Choose the working folder")
setwd(dir)
library(ThreeDRNAseq)
run3DApp()
```

Data import

To import all the necessary data into 3D RNA-Seq, navigate to the input page in the app and select the corresponding entries of steps 1 & 2, as illustrated in this figure.

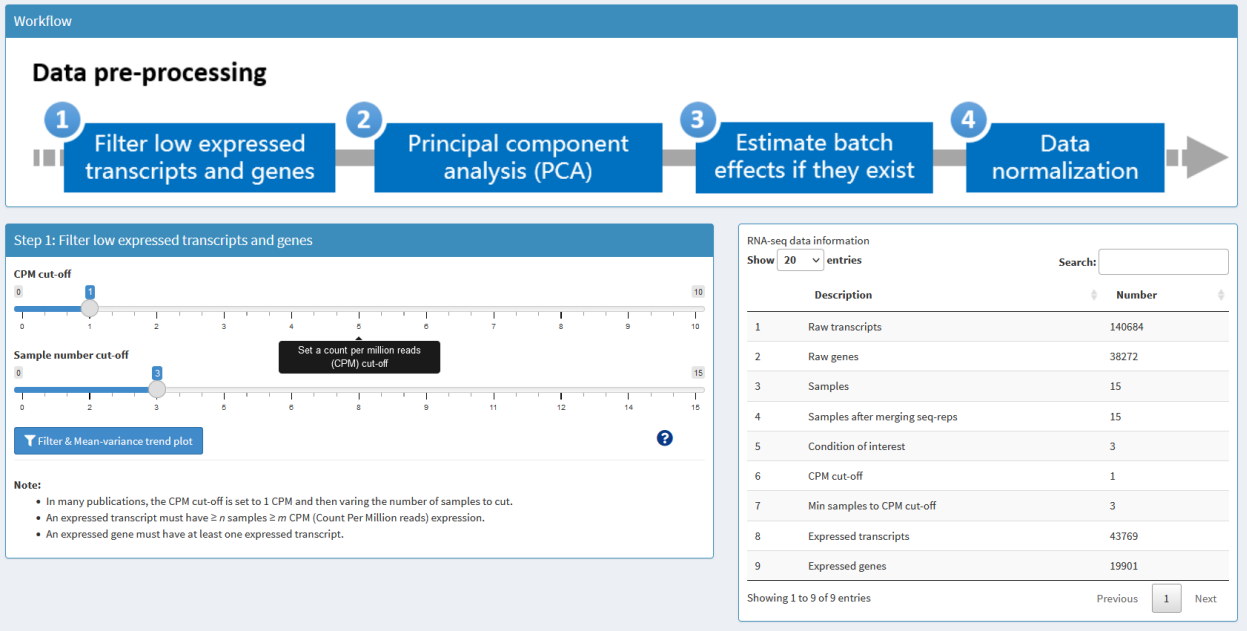


The next step scales and summarizes the transcript-level and gene-level abundance estimates from Kallisto while controlling for differences in transcript length. This is achieved by using the tximport lengthScaledTPM option, which first multiplies TPM by feature length and then scales up to library size.



Data pre-processing

Once all the required input steps are completed, we can move on to the data pre-processing step. The goal of this phase is to normalize the expression values and to filter out lowly expressed transcripts. The Step 1 allows you to select the filtering settings in terms of Counts per Million reads (CPM) and number of samples covered. In our case we have 5 samples per color morph and we can define lowly-expressed transcripts as those that show >= 1 CPM in >= 3 samples (which restrict the analysis to transcripts expressed in > 50% of samples of a given group).



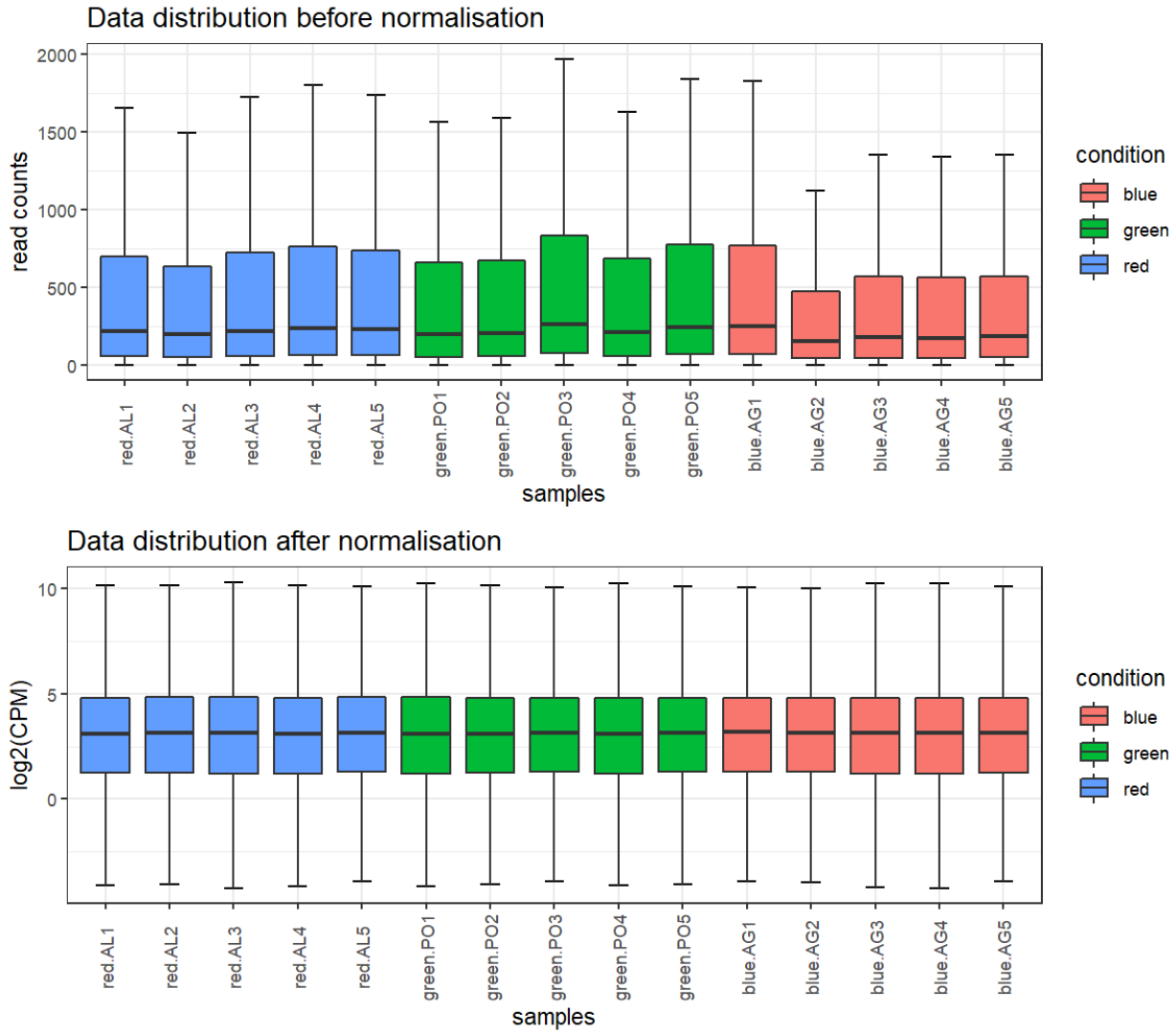
These lowly-expressed transcripts will then be filtered out and the summary statistics along with plots of the variance trend of included transcripts will be displayed. The effects of the filtering can also be visualized in a bar plot of the number of transcripts per gene. In our case, the lowly-expressed transcripts are predominantly single-isoform transcripts.

It is also very useful to plot the results of a principal components analysis of the expression values per sample. In this case the samples are clustered into color morphs which indicates that the variation in expression among color morphs is larger than that observed within a given color morph.

The next step, batch effects correction is only needed when samples from different Illumina runs or from different labs are to be combined in the same study. This is not our case and we can skip this step.

Finnaly, the last step in this part is the normalization of expression values. Normalized expression values are necessary to control for known effects in RNA sequencing like differences in coverage and gene/transcript length. The goal here is to make gene expressions directly comparable within and across samples. TMM is a between-sample normalization method ideal for our case as was designed for studies comparing samples where RNA expression values are expected to be very different among the samples (different tissues, genotypes,

populations). After the unwanted sequencing bias are controlled for when this normalization is applied, the variation within and among samples is significantly reduced.

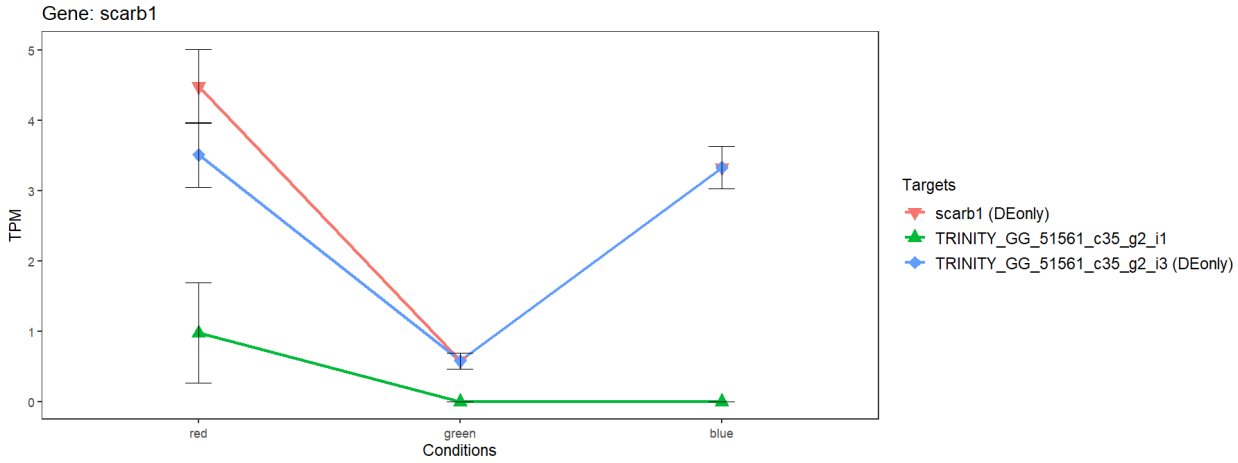


3D analysis

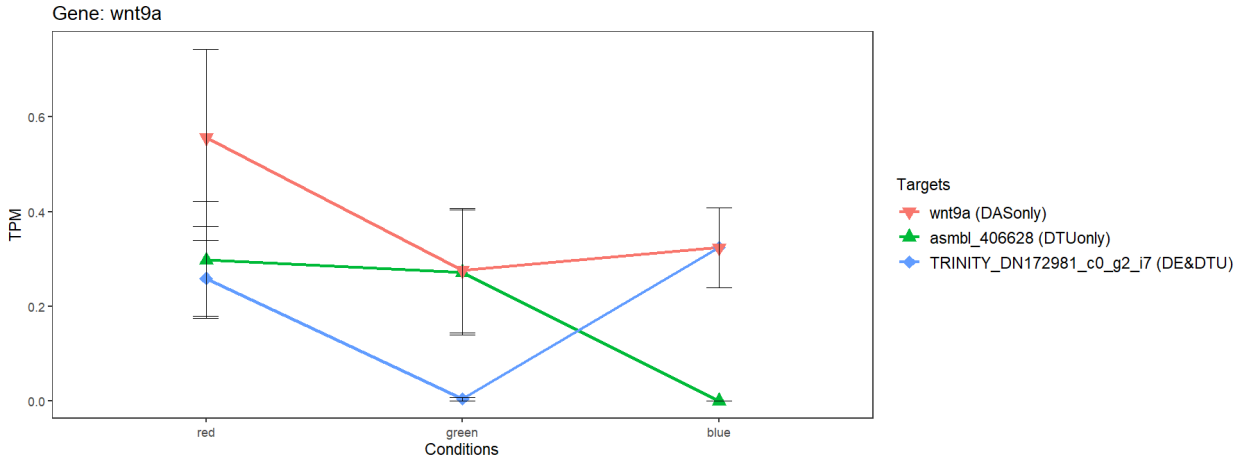
3D stands for three levels of differential expression test: gene/transcript expression differences (DE), alternative splicing differences (DAS) and transcript usage differences (DTU). To conduct these test it is first necessary to speciffy the experimental design. The example in this practice is very simple as we are only testing the effect of one categorical factor ("color"). To configure the tests, on the Step 2 dialog window set the three possible contrasts between color phynotypes of *O. pumilio* in Bocas del Toro archipelago and click on "generate contrast groups" below.

On the Step 3 select "limma" "voom" procedures, an F-test of the DAS values, a Benjamini-Hochberg procedure for error rates correction, an adjusted p-value of 0.05 and a log2 fold change of 1 and a delta PS of 0.1. Once all options have been properly set, run the statistical tests.

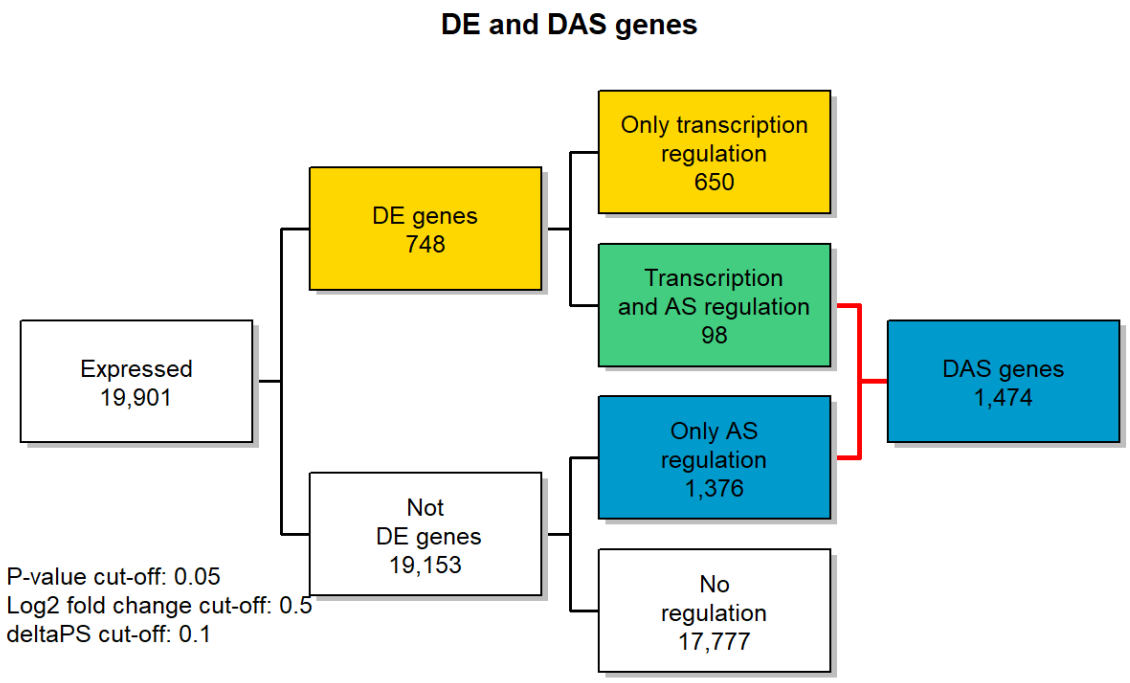
The results will contain tables and plots of each of the three levels of analysis that can be conveniently explored in the app. An example of DE-only gene is the scarb1 gene which is part of the carotenoid synthesis pathway and is down-regulated in green frogs in this case.



A more complex example is the wnt9a gene, which intervenes in the melanin synthesis pathway. In this cases the gene-level analysis identifies this gene as DAS while there is DTU for one of the transcripts and DE and DTU for the other transcript.



Continue to explore the diversity of plots, the tables and use this information to identify the relative roles of each level of differential expression in relation to the color phenotypes. The number of genes / transcripts in each differentially expressed category are nicely summarized in the provided plots.



References

Guo, W., Tzioutziou, N. A., Stephen, G., Milne, I., Calixto, C. P., Waugh, R., ... Zhang, R. (2021). 3D RNA-seq: A powerful and flexible tool for rapid and accurate differential expression and alternative splicing analysis of RNA-seq data for biologists. Retrieved from https://pubmed.ncbi.nlm.nih.gov/33345702/

Nicolas L Bray, P. M., Harold Pimentel, & Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. Retrieved from http://www.nature.com/nbt/journal/v34/n5/full/nbt.3519.html

R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org

Rodríguez, A., Mundy, N. I., Ibáñez, R., & Pröhl, H. (2020). Being red, blue and green: The genetic basis of coloration differences in the strawberry poison frog (oophaga pumilio). Retrieved from https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-020-6719-5