An EU-Canada joint infrastructure
for next-generation multi-Study Heart research

Deliverable D4.4

Bioinformatics Toolbox

Reference	D4.4_euCanSHare_UKE_30112020
Lead Beneficiary	UKE
Author(s)	Prof. Dr. Tanja Zeller, Dr. Andrej Spiess, Dr. Anna Lena Engels
Dissemination level	Public
Type	Websites, patents filing, etc.
Official Delivery Date	30.11.2020
Date of validation by the WP Leader	30.11.2020
Date of validation by the Coordinator	30.11.2020
Signature of the Coordinator	

euCanSHare is funded by the European Union's H2020 Framework under Grant Agreement 825903.

Version Log

Issue Date	Version	Involved	Comments
20/11/2020	1	Andrej Spiess	First draft
23/11/2020	2	Katharina Heil, Karim Lekadir	First review
30/11/2020	3	Andrej Spiess, Tanja Zeller	Included GUI figures of RNAseq toolbox
30/11/2020		Katharina Heil, Karim Lekadir	Final updates and finishing of the deliverable
30/11/2020			Final version

Executive Summary

Meaning and purpose of this deliverable is to demonstrate the applicability of a bioinformatical tool (part of a larger toolbox) that can either analyse external data through an upload mechanism or offer the automatic analysis of internal server-housed data.

For this initial case, we selected the analysis of RNA sequencing (RNAseq) data, the *de facto* standard of today's gene expression measurement, as it is widely applied in the scientific community.

We have programmed a tool that (as it currently stands) can analyse differential gene expression between two groups, based on a provided "raw count" RNAseq matrix and three additional files containing gene annotation data, group definitions and covariates. All data is automatically matched and a subsequent extensive analyses of the data is conducted, including visualizations of expression levels, variance structure analysis by decomposition (PCA), variance contribution analysis, hierarchical clustering of top differential transcripts, profile plots, and diagnostic plots (MA plot, Volcano plot). During analysis, the obtained data to generate these exported plots is also automatically exported and named accordingly. The differential gene expression is calculated by covariate-adjusted linear models with multiple testing-corrected *p*-values. Finally, a large result matrix is generated, with the original count matrix augmented with annotations, gene names and the complete statistical data and sorted ascendingly by the corrected *p*-value, so that the most differential transcripts reside on the top of the data.

In future, it is envisaged that the user selects RNAseq data deposited alongside clinical variables and defines the desired grouping of the samples, which then is sufficient to create a complete analysis output as described above.

Table of Contents

Acronyms

RNAseq: RNA sequencing

GWAS: Genome-wide association study

PCA: Principal Component Analysis

PVCA: Principal Variance Component Analysis

R: The R language for statistical computing, www.r-project.org

QC: Quality control

FDR: False discovery rate

VRE: Virtual Research Environment

GUI: Graphical User Interface

1 The (artificial) RNAseq data

For this initial analysis case, we generated artificial RNAseq data with defined properties, in order to provide a quality control throughout the pipeline. Specifically, we generated RNAseq counts for 100 samples and 151,298 transcripts ID's of the current Ensemble Genes 101 database (https://www.ensembl.org/biomart/martview/). In a first step, a vector *RV* of size 151298 containing data from an exponential distribution with rate 0.0001 was generated, which mimics the distribution of transcripts within cells, i.e. many transcripts with low abundance and less transcripts with moderate to high abundance. Since RNAseq data often contains 0-cells (sparsity), the data was also zero-inflated with many zero counts (Figure 1).

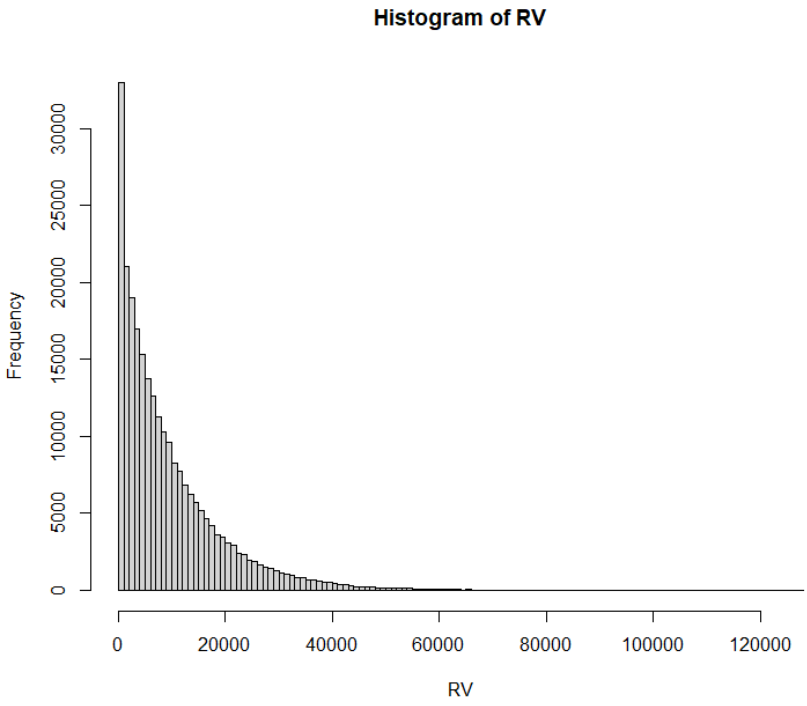

Figure 1: The underlying distribution of the artificial RNAseq data.

Finally, for each of the exponentially distributed random deviates RV_i, 100 further random numbers from a normal distribution with mean 0 and a standard deviation of 10% were generated and added, $RV_i + \mathrm{N}(0, 0.1 * RV_i)$, delivering exponentially distributed data that is normally distributed throughout the samples.

In a last step, a defined number of artificial differential transcripts was spiked into this matrix, in detail 800 transcripts in Group 1 (Samples 1-50) and 200 transcripts in Group 2 (Samples 51-100), where these 1,000 transcripts were 2-fold upregulated (Figure 2). This paradigm was chosen as to provide a means of **following these 1,000 transcripts through the analysis pipeline and offer an internal QC for the complete procedure**.

In a last step, artificial covariates for the linear model were generated, consisting of the two covariates sex (binomial distribution) and an age group of four classes (binned normal distribution).

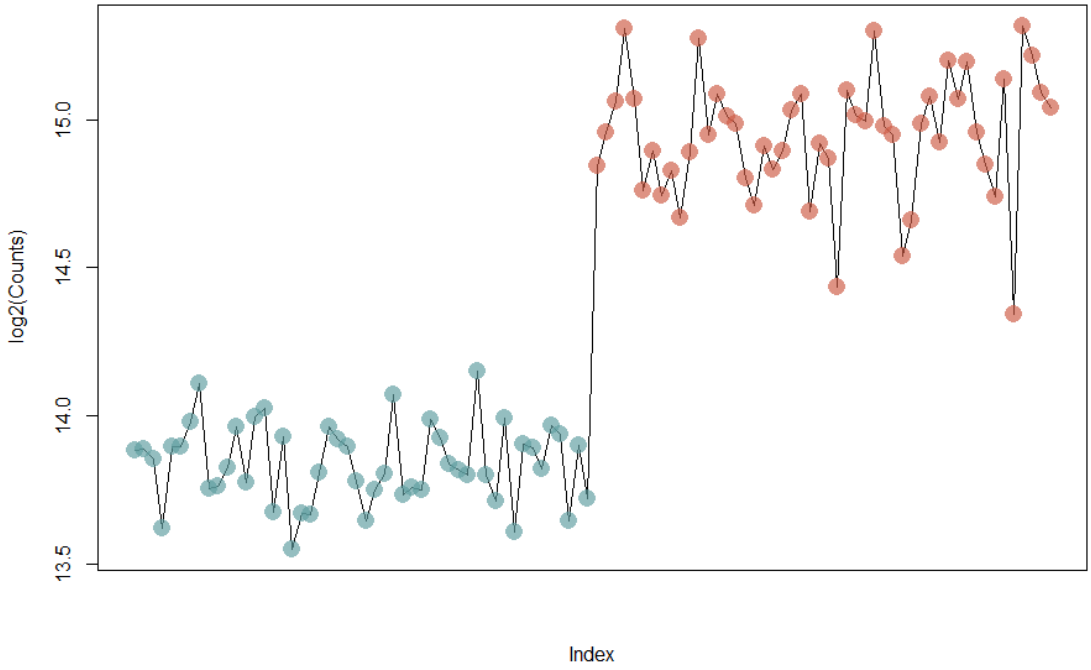

Figure 2: An artificially generated differential transcript with 2-fold upregulation in Group 2.

2 Upload, analysis and download structure

The user either supplies the RNAseq count matrix or (in the future) it will be fetched from the server as companion data to the clinical variables. In addition, further files containing the dichotomous group definition (Sample1 => 0, Sample 2 => 0, Sample 3 => 1, etc.) and the covariates need to be supplied. The data is then ported through a *Python* skeleton and analysed within a server-sided *R* environment. The generated plot file (.pdf) and tabular result file (.xlsx) is automatically generated and stored in a newly created and project-specific folder, downloadable by the user (Figure 3).

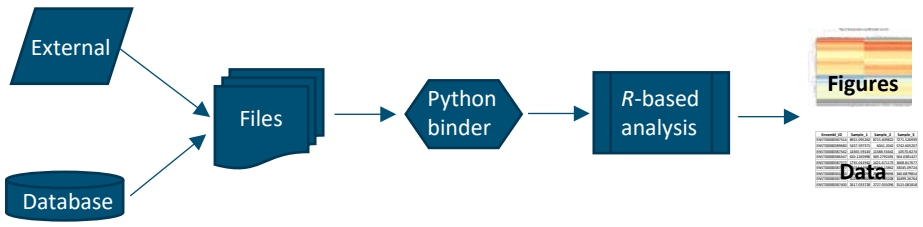

Figure 3: Upload, analysis and download procedure.

3 The analysis pipeline

For this initial case, our synthetic RNAseq dataset containing 1,000 artificial differential transcripts was supplemented with group definitions and covariates and then imported into the *R* environment. Within the analysis pipeline, the following steps were conducted, each of these delivering figures and tabular output as compiled below (Table 1).

Table 1: Analysis steps and their corresponding output.

Step	Analysis	Figure	Table
1	Import of data		
2	Background removal		
3	Distribution of Counts	Figure 01	
4	PCA of all transcripts	Figure 02	Sheet 01
5	PVCA analysis	Figure 03	Sheet 02
6	Top 2000 filtering		
7	PCA of top 2000	Figure 04	Sheet 03
8	Heatmap of top 200	Figure 05	
9	Profile plot of top 10	Figure 06	
10	Adjusted linear model		
11	MA plot	Figure 07	
12	Volcano plot	Figure 08	
13	Final Result Matrix		Sheet 04

During the analysis, the *R* script goes through all 13 steps and generates one .pdf (Figures) and one .xlsx (statistical analyses) output file. Essentially, the user is supplied with a large data file containing the original raw count matrix, but supplemented with complete gene annotation (Ensembl ID, Gene Description, Refseq ID, Gene Symbol, Gene Type) and the statistical results of the covariate-adjusted linear model analysis, including the corresponding variance value, raw *p*-value, adjusted *p*-value (FDR), and differential expression ratio. The output is ascendingly sorted by adjusted *p*-value, so that **the user finds the most differential transcripts – based on the prior group definition – directly on top.**

In the end, the pipeline generates 8 Figures in the .pdf and 4 data sheets in the .xlsx file. In both of these, we can observe that the pipeline has fully unveiled and recovered the set of 1,000 transcripts that was artificially increased in both groups (800 in Group 1, 200 in Group 2): The two groups (in green and red) are clearly separable by PCA (Figure 4, “Figure 2”) and hierarchical clustering (Figure 4, “Figure 4”), display a differential profile (Figure 4, “Figure 6”) and are fully reconstituted from the remaining set of non-differential transcripts (red dots in Figure 4, “Figure 8”). Furthermore, the ratio estimates obtained from the linear model at the top of the exported result matrix indicate, with ratios ~2 and ~0.5, that we have successfully extracted these from the pool of 151,298 transcripts (Figure 5, right side).

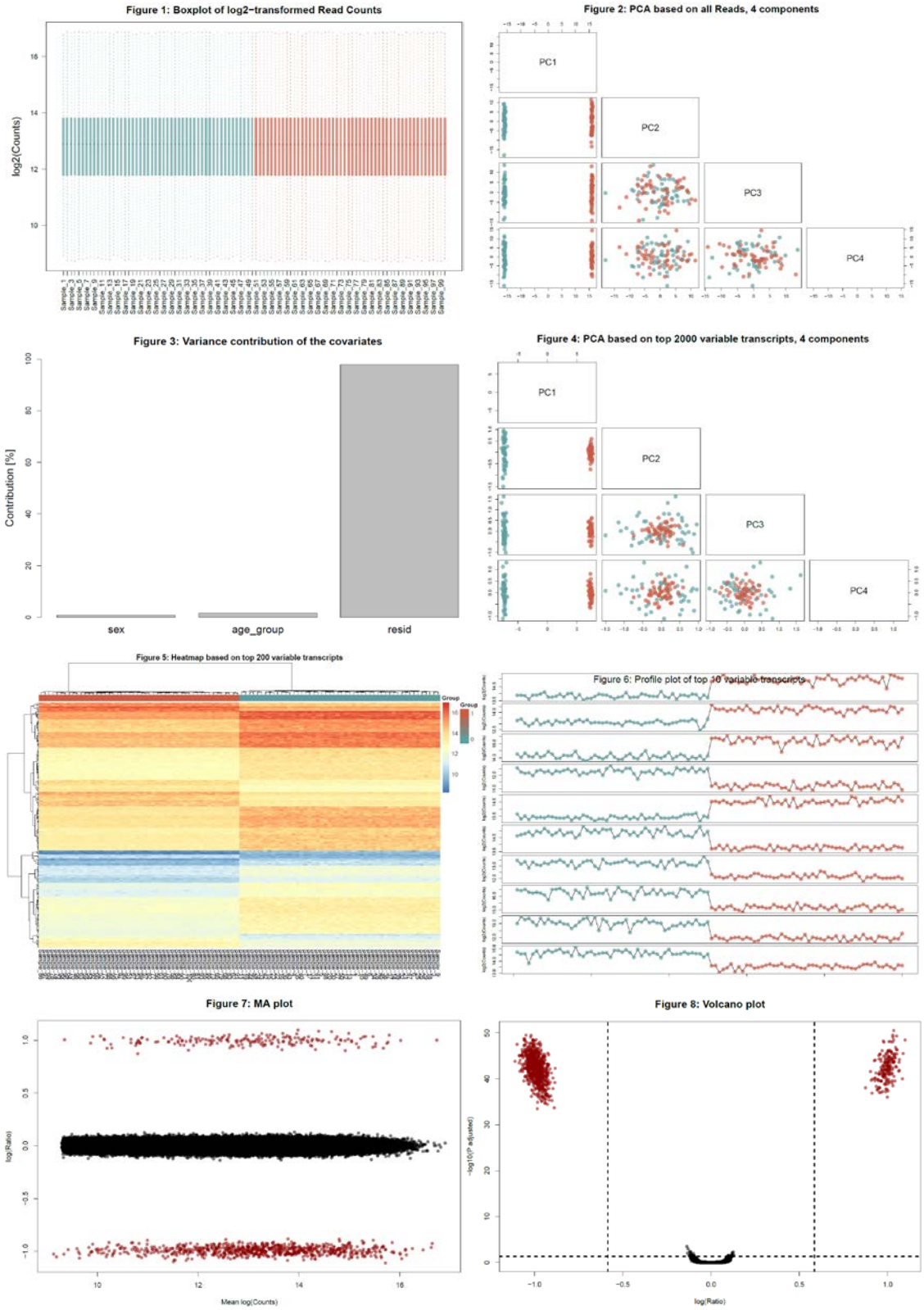

Figure 4: The eight figures as generated by the RNAseq analysis pipeline.

Annotation							Statistics						
Transcript	Gene_des	Chr	RefSeq	Transcript	Gene_typ	Strand	Transcript	Gene_Syn	log(Estimate)	SE	t	P	Var	Padj	Ratio
ENST0000	NADH:ubi	20		NDUFAF5-	protein_c	1	933	NDUFAF5	1.034524549	0.029321	35.28226	2.16E-56	0.298028	2.94E-51	2.048438
ENST0000	long inter	10		LINC01517	lncRNA	1	1968	LINC01517	1.0288423	0.0302	34.06783	4.73E-55	0.28802	3E-50	2.040386
ENST0000	LDL recep	CHR_HG1362_PATCH		LRP6-214	protein_c	-1	10252	LRP6	-1.04826794	0.030887	-33.9386	6.6E-55	0.29044	3E-50	0.483548
ENST0000	complem	CHR_HSCHR6_MHC_		C4A-216	protein_c	1	974	C4A	-1.019745155	0.030164	-33.8072	9.29E-55	0.283103	3.16E-50	0.493203
ENST0000	G protein	CHR_HSCHR6_MHC_		GNL1-228	protein_c	-1	774	GNL1	1.050673728	0.031292	33.57644	1.69E-54	0.2958	4.61E-50	2.071497
ENST0000	carnosine	18		CNDP1-20	protein_c	1	1511	CNDP1	-1.02172667	0.03075	-33.2267	4.24E-54	0.28725	8.25E-50	0.492527
ENST0000	mucin 17,	7		MUC17-20	protein_c	1	403	MUC17	1.029158923	0.030933	33.27032	3.78E-54	0.286104	8.25E-50	2.040834
ENST0000	protein ph	14		PPP2R3C-	protein_c	-1	492	PPP2R3C	-1.060034236	0.032249	-32.8703	1.09E-53	0.299964	1.85E-49	0.479621
ENST0000	novel tran	6		AL590428.	lncRNA	-1	1973	AL590428.	-1.054772989	0.032184	-32.7728	1.41E-53	0.313582	2.14E-49	0.481373
ENST0000	trans-2,3-	19		TECR-213	protein_c	1	508	TECR	-0.990502318	0.030367	-32.6173	2.14E-53	0.268959	2.91E-49	0.503303
ENST0000	novel tran	4		AC104806.	lncRNA	-1	409	AC104806.	1.034162066	0.031776	32.54501	2.59E-53	0.290576	3.21E-49	2.047924
ENST0000	LHX1 dive	CHR_HSCHR17_7_CT		LHX1-DT-2	lncRNA	-1	1462	LHX1-DT	-1.008295654	0.031017	-32.5078	2.87E-53	0.273047	3.25E-49	0.497133
ENST0000	novel tran	17		AC127521.	lncRNA	-1	575	AC127521.	-1.009185952	0.031095	-32.4547	3.31E-53	0.290499	3.34E-49	0.496827
ENST0000	PRELI dom	20		PRELID3B-	protein_c	-1	612	PRELID3B	1.098386893	0.033859	32.44033	3.44E-53	0.333946	3.34E-49	2.141152
ENST0000	centroson	20		CEP250-20	protein_c	1	868	CEP250	-1.057453398	0.03265	-32.3878	3.96E-53	0.297663	3.59E-49	0.480479
ENST0000	MON1 hor	16		MON1B-2	protein_c	1	877	MON1B	-1.008236611	0.031158	-32.3594	4.27E-53	0.278312	3.64E-49	0.497154
ENST0000	long inter	3		LINC02037	lncRNA	1	935	LINC02037	-1.006613471	0.031186	-32.278	5.32E-53	0.279064	4.26E-49	0.497713
ENST0000	RNA bindi	8		RBPMS-21	protein_c	1	494	RBPMS	-0.998714166	0.031019	-32.1968	6.63E-53	0.278234	5.02E-49	0.500446
ENST0000	armadillo	19		ARMC6-20	protein_c	1	563	ARMC6	-1.05279531	0.032745	-32.1518	7.49E-53	0.309756	5.37E-49	0.482033
ENST0000	novel tran	1		AC239809.	lncRNA	-1	1023	AC239809.	0.975830742	0.030391	32.10905	8.41E-53	0.256706	5.73E-49	1.966773
ENST0000	calcium bi	11		CABP4-20	protein_c	1	1426	CABP4	1.016428461	0.031772	31.99095	1.16E-52	0.287956	7.51E-49	2.022905
ENST0000	novel tran	7		AC005165.	lncRNA	-1	658	AC005165.	1.007254037	0.031502	31.97391	1.21E-52	0.275755	7.51E-49	2.010082
ENST0000	SPOC dom	1		SPOCD1-2	protein_c	-1	563	SPOCD1	1.027429482	0.032235	31.87334	1.6E-52	0.293432	9.45E-49	2.038389
ENST0000	novel tran	12		AC008127.	lncRNA	1	499	AC008127.	-0.970908594	0.030533	-31.799	1.96E-52	0.261771	1.05E-48	0.510185

Figure 5: The annotated and top differentially sorted result matrix.

4 Implementation

The RNAseq toolbox is part of the VRE Toolbox homepage at https://vre.eucanshare.bsc.es/vre/tools/RNAseq/input.php?op=0 (Figure 6) and offers a GUI in which to upload the four mandatory files (expression matrix, group definitions, annotations, covariates) needed for the analysis (Figure 7).

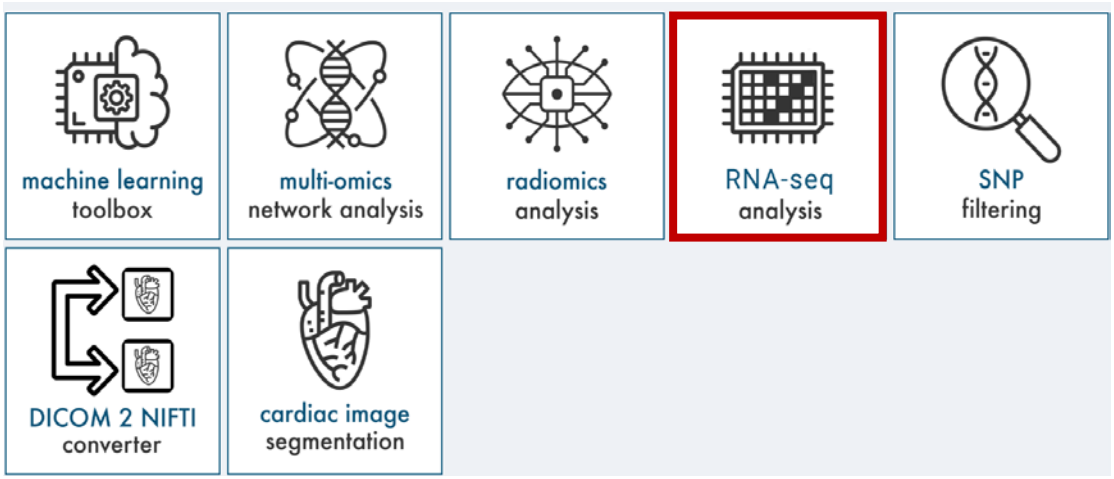

Figure 6: The RNAseq toolbox as implemented on the VRE toolbox homepage.

After successful upload, the figures and statistical analyses are started after clicking a "Compute" button, and then it takes approx. 5-10 minutes to create the two files (pdf, xlsx) that are deposited in a result folder for the scientist to download.

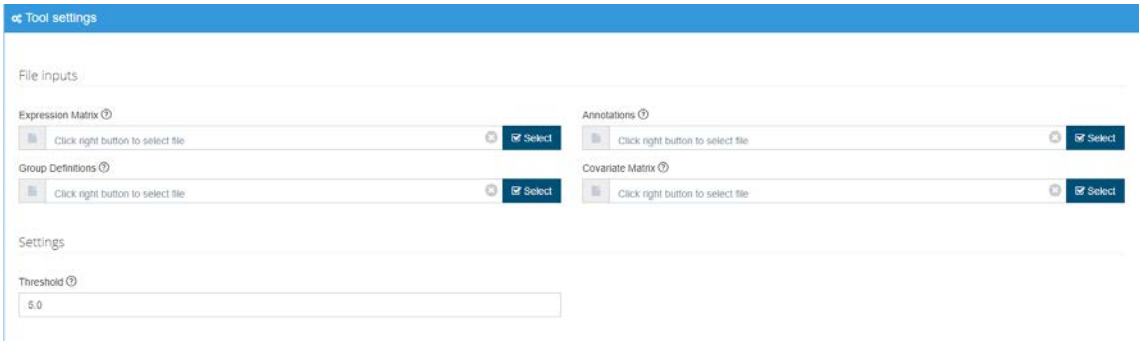

Figure 7: The GUI for uploading the four mandatory files.

5 Requirements

The implementation requires, as depicted in Figure 3, the following infrastructure:

1) A raw RNAseq count matrix, either in a server-sided database, e.g. Opal, or alternatively for client upload.
2) A GUI for either selecting the database data or uploading data.
3) An implementation of *R* version 4.0.3 and the packages “openxlsx”, “lme4” and “pheatmap” on the server.
4) The *R* script Skript-analyze.R on the server.
5) A binding of the GUI and server to *R*, by Python codelets.
6) Automatically created, project-specific directories on the server, for the output files to download.

6 Outlook

The tool, as described here, offers an initial inspection on how a more complex bioinformatics toolbox might be represented. In principle, these should be implemented in a way that either the user can upload his own data, or alternatively fetch similar data from the server. To date, the “SNP filtering” tool implemented in the euCanSHare platform VRE is the adequate template to this approach, with upload buttons for the corresponding data and a “Compute” button to start the analysis. The underlying *R* scripts are relatively simple to establish, however the Python-binding and calculation/export *via* the server-sided *R* environment constitutes the most complex part.

Once this kind of pipeline is established, similar tools can be implemented in a fairly quick manner. To our opinion, the development approach should be demand-driven, that is, be guided by the molecular data that will be supplied together with the clinical cohorts. If, for instance, this will be mainly genetic data, then establishing a more sophisticated tool for filtering and analysing variant-calling files might be feasible.

The main advantage of these tools is that they provide a simple interface for the bioinformatical “layman”, such as clinicians or wet-lab biologists. In the approach developed here, we tested the performance by using artificial data with defined properties, so that we have *a priori* knowledge on how the results need to appear, which in turn i) tests the proper performance of the pipeline and ii) ensures that the implemented method is not of a black box-type.