R coding task

Here is the description of tasks, required in the R coding task. This will test the usage of basic R commands, ability to edit character strings, the basic usage of DESeq2 etc. One step with DESeq2 will require appr. 2 Gb of memory. Also the same step will take 30 – 45 minutes (not quite sure on this)

I would like you to A) write a run script, with little commenting, that will perform the required steps and B) a very small report, where you answer the couple of questions and add the output that I requested. Alternatively, you can use R Markdown.

This is first time I am using this task to evaluate the workers. Don't get angry, if it is too hard, too easy or not clear!

Step 1: Read the datasets to R

You should have the data file where you have the read data. This has different biological samples in columns and different genes in rows. Two first columns contain the names for the genes. Rest of the columns contain the read values. Read this file to R.

You should also have a table that gives information on what types of biological samples are in question. It includes various classifications for samples (patient information). This is used when we compare sample groups. Read this file also to R. Both files are tab-delimited. Here each row represents one sample. Different columns represent various sample classifications, like control gut, gut from risk group, carcinoma samples etc.

Step 2: Select relevant data columns and data rows

Two first column have non-numeric information. Exclude them from the further analysis. Use, however, the first excluded column from the input data as row names in your data matrix

Four last rows are control data rows in the count data matrix. Exclude also them from the further analysis. Check the first two excluded columns for these rows. What were these rows called?

(BTW: This data was created with HTSEQ program. It generates output with these control rows.)

Step 3: Clean the column names from data matrix

Column names in the data table are too long, so we have to shorten them. In addition, the shorter versions will be used in the next step to reorder the sample type table. This step will be easier if we shorten these names.

Current names look like this:

```
...
[83] "MMR42_L_2_S52_L001_R1_001"      "MMR42_LN_2_S51_L001_R1_001"
[85] "MMR43_N_1_S53_L001_R1_001"      "MMR48_N_1_S54_L001_R1_001"
[87] "MMR612_S55_L001_R1_001"         "MMR658_S56_L001_R1_001"
...
```

You have to remove the area highlighted with yellow.

Step 4: Reorder the sample types table

Unfortunately the two tables, read count data matrix and sample type table, are not ordered in same way. We have to reorder the rows in the sample type table so that they match the columns in the read count table.

Generate the ordering vector for sample type table that can reorder it to match the count table. How would you check the result?

Step 5: Load data to DESeq2 object

Load the count data and the sample type information to DESeq2 object. Use only columns from 4 to 8 and column 15 from the sample type table (order of columns from left to right). Remember the reordering. Use the command that loads dataset as a matrix to DESeq2. You have to also select a design (classification of samples to studied groups). Use the column that was called “Sample type 3” in the original sample type table.

What has happened to that column name inside the R? Is this a wanted behavior?

Step 6: Relevel the design

DESeq2 selects the reference sample group by alphabetical order. This is also called reference level for factor. Use the command relevel, and set the sample group 'CTRL_gut_CTRL' here as reference group.

Step 7: Analyze Differential Expression of Genes

One main task of RNA-seq data analysis is to look for genes that show different activity between samples. This is called Differential Expression analysis. Do the actual calculation using DESeq2. This step took appr. 2 Gb of memory on our unix server. It took over 30 minutes. I did not use any parallelization.

I did this on server as my data was originally there. No other reason for that choice.

Step 8: Extract results from DESeq2 object

Extract results from the output object of DESeq2 to a variable called res. Take a quick look on the results, by typing res to console. What sample groups were compared here?

Step 9: Extract the overall view of the results

Convert the generated output, res, to data frame. Use summary function on the data frame, generated in the earlier command. Add the output to your report.

Did you notice errors here? There might be as I wrote this quickly!