Next Generation Sequencing Technologies: Data Analysis and Applications

Hands-on 2 Setting up the system

Dr. Riddhiman Dhar, Department of Biotechnology

Indian Institute of Technology Kharagpur

Good day, everyone. Welcome to the course on Next Generation Sequencing Technologies, Data analysis, and Applications. In the last class, we started our hands-on, and we focused on installing R and setting up the system. So, we will continue that in this class. So, we will set up the system in R. I will show you how we can get this data and how you can start loading packages in R for all sorts of analyses that will come afterward.

Just to remind you, the goal is to perform a differential gene expression analysis and identify genes that show differences in expression between two different conditions. So, I have also shown you the data set, where we have 16 samples and raw count data for about 6800 genes. And in the last class, we loaded that data in R, and we were analyzing or looking at some of the statistics and how to access some of the elements in the data, ok? So, we will continue that.

We will go into much more advanced analysis now, and we will also see how to load packages. So, this is the agenda for this class. Now, let us move on to the R part, okay? So, we have the R installed, right? So, here it is.

We have the data; we have loaded the data and this is the data we have, ok? So, head data, this is sorry, head tab, right? So, the name of the variable is tab. So, we have the data here, and I have shown you how we can access the data. Now, with R, we can do a lot of very interesting things.

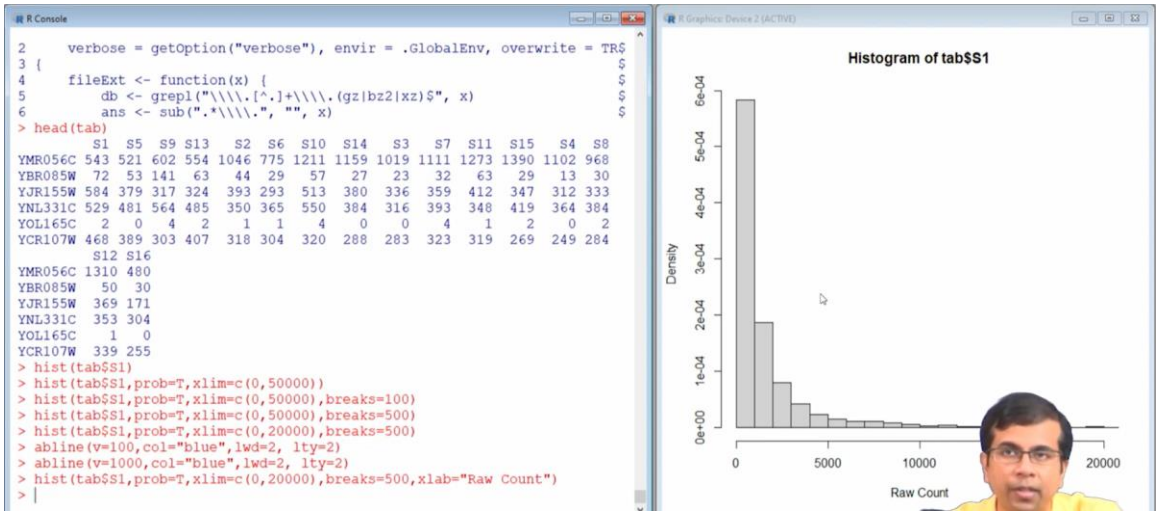
We can actually also look at the data, plot the data, and do statistical analysis, right? We can do a lot of things, ok? So, one of the first things, we can just play around a little bit so that you get a feel for the data and how to handle the data, ok? Of course, I also encourage you to play around, right? Only then will you learn a lot more about R, ok, not just by reading the instructions or manuals because sometimes those might be very difficult to

understand unless you try yourself on this console, ok.

So, what I will do is simply generate, for example, here we can probably generate something like a histogram, ok, by accessing this histogram tab dollar s1, ok. So, I showed you that this is a very convenient way of accessing the column data in R, right? So, let us say we want to create a histogram, and we will see, right, what is the distribution of this read count in the sample s1, ok? So, for that, we are doing this histogram tab, dollar s1, right? And as you can see from the histogram, it looks a bit weird, right, because we have this frequency here, and then most of the values are within 50000, and then you have probably very few values around 50000, right?

So, what you can do is change this instead of a frequency histogram and make it a density, which probably equals true because for each sample, the number of counts or the frequency would be different. So, we can make it dense, so we can probably say it is probably true. We can also change the x-axis limit, right? So, instead of this 0 to really big limit, we can simply say x lim equals to c 0 to c. So, we can say that it is up to $50,000, right?

So, we want to see up to 50,000 because the rest of them would be outliers. Now, this is a problem, right? It is a histogram, just like you see just one bar here, right? Again, we can solve this issue by using this command called "breaks. So, this will decide, right, this will actually design these breaks in such a way that, right, we can see these bars, right, of different heights, ok. And similarly, we can increase this, so maybe we can make it 500, and now you see probably a bit nicer, right? If you look at it, we can also change the limit further, and you can see this.



This is what the distribution looks like, ok? So, this is something that you can do very easily, and one thing you will probably realize as you start using R is that it is very flexible and you can actually add a lot of things to your data. So, one of the things maybe we want to see, let us say we want to draw a line at, let us say, 100, right, so where the count is 100, right, and we want to see, like, what fraction of these genes actually have read count more than 100, ok? So, for that what we want to do here is add a vertical line at 100 here, ok? So, in R, we can do that very easily.

Line, since this is a vertical line, we can say V equals 100, right, and we can then also give a color to that line so that we can clearly see this line, right. We can also mention line width, so if we want a thicker line, we will increase this width and also line type. So, lty is line type, so maybe we wanted a dotted line, right, or a dashed line, ok? So, again, you can change that with this lti command, and you can see here, right? This is a line that appears around here, ok? This is a blue line.

Maybe we can change it to 1000, so this would be more visible, ok, and you can see there is another line, ok. So, this is something that is really possible in R, right? You can add these vertical lines and the horizontal lines using these commands after you have drawn a plot. Maybe let us say you want to change the labels of this histogram, right? So, how do you change these labels? Something like xlab, right, and xlab is the, we will say, raw count, right.

This is the raw count. The X label is raw count, the Y label is fine for us, and the main title is also fine. So, you can see now that the raw label has changed to raw count, ok? Instead of this something else, it is now raw count, ok? So, this flexibility is there, and we will actually explore this a bit more when we actually go for the visualization part, right?

So, we have this result visualization part for DEseq analysis and we will explore that when we go there, ok? So, another thing we can do is maybe we want to see how good these counts are, right? How good is this correlation between counts between these, let us say, S 1 and S 5 samples? Maybe we can see that visually, ok? So, I will tell you, right? This information will come later.

So, S 1 and S 5 are technical replicates of the same sample, ok? So, they come from the same condition. These are simply technical replicates, ok? Can we see how good this correlation is between the gene counts and the raw counts that we have for the genes?

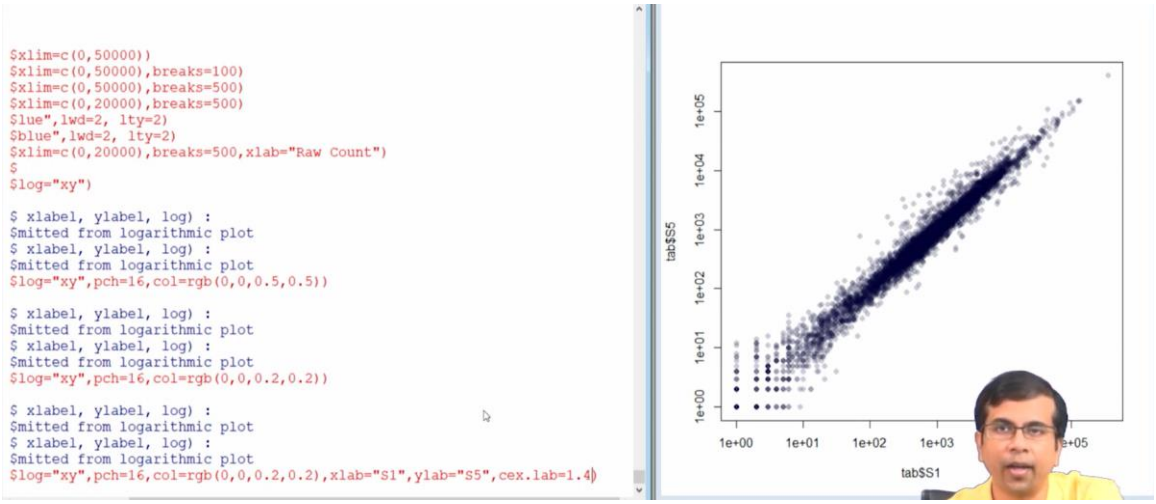
You can do that, right? You can simply say plot; we are getting the column data now, right, and we can access these two columns of data. So, what we are doing is simply saying plot x and y. So, x is this column data from the S 1 sample, and y is the column data from the S 5 sample, ok? And you can see, right, this is how it looks, ok? Maybe this is not clear, ok, because there is one extreme count, right, a very high count for one gene here that you can see, and the rest of them are concentrated here.

All these 6800 genes are concentrated in this region. So, that is why they are not really visible. So, what you can do is plot this on a log scale, right, and in R, you can do that very easily by saying that we are dealing with positive numbers only, so R = 0. So, we can simply say log x, y. So, it will transform both the x and y axes into log scales, ok?

And this is something you see now, right? So, you see, there is a pretty good correlation, ok? And one of the things you can probably see here is that it says 273 x values are omitted because they are less than or equal to 0, because they are probably 0, so there is no count. So, anyway, we are not really that interested where we have these 0 counts and the rest of

them are here, ok? Now, what can you do also? So, one thing you probably notice from this analysis immediately is that in the reads or genes that show a very low count, there is a lot more variability between these technical replicates.

You can see this part is quite diffuse and the difference between two replicates is large, ok? This is what we always observe in RNA-Seq data, right? So, the reads or the genes that show a low read count or are lowly expressed show more variability, ok? This is expected, but then there are some genes where we see some variability, ok? And this is something that is part of our experiment; we have this technical variation that we will have to deal with, ok?



So, what can you do? We can probably now also see that this part looks very dense, right? So, maybe we can change it a little bit. We can also change the colors, right? So, there are all these options that you can add here after we have said plot, right? So, you can say we want these filled circles, not open circles; we want to change the colors, ok? Now, instead of using this blue-red color in R, you can also use this RGB scale.

And you can specify which color you want, ok? Let us say we want a blue color, but a very light blue color, ok? So, the first number is here, so in here you will have four numbers, ok? When you say RGB, the first number is for the red scale, ok? We are saying 0 because we do not want red.

The second one is for green, and the third one is for blue; we want blue points. So, we are

using this if you want pink or some combination of red, green, and blue. You just mention these numbers and different fractions, and you will get these colors. And the last number is for transparency, ok? So, we want this only on these points to be 50 percent transparent because we want to understand the density of these points inside. Otherwise, they become fully black in here, ok?

So, all these things we can do, and we can simply write this. You can see, right, this has changed now, ok? The color of these types has changed, and this color has also changed. Maybe we want even lighter colors, right?

So, we do not want this darker color. So, we change the color code from 0.5 to reduce it to 0.2; we also increase the transparency, and this becomes a bit darker, ok? So, you can see this now getting a bit more transparent, but there are so many points around here that still have a very dark region, ok? Now, maybe we want to see this correlation better, right?

So, maybe we can draw and y equals the x line, ok, and maybe we can add a dotted line to this plot, ok. So, before we go there, maybe we want to change the x-axis levels, right? So, you can say x lab is S 1, right, y lab is S 5, right, and y level. We can change the size of these levels; we can increase them by cx dot lab 1.

4. So, any number above 1 would increase the size; any number below 1 would decrease the size, ok? And by the way, you can do the same thing for the axis as well as for the points that are in the plot, ok? So, we will not touch the axis and the points at this point at this moment; we will do that later on, ok? So, as you can see, S 1 and S 5 levels have appeared now, right, in the x axis level and the y axis level, and now we want to add this line y equals to x, ok? So, that will actually tell us how good the correlation is, ok?

So, for doing that up line, a equal 0, right, and b equals 1. So, of these two numbers, the first number is the intercept, ok? So, we want this line to pass through 0, ok? So, we are fitting this y equals to b x plus a, right, or a plus b x, right.

So, a is the intercept and b is the slope, ok? So, b equals 1, which mean the slope is 1, right? And then we can say the line width, right, line type 2. So, this would give us a dotted line, and we can perhaps say the color would be red here. We want a red color because we will get a good contrast, perhaps. And you can see that this y equals the x line here, right? And you can see most of these points are falling around this or very close to this y equals to x line, except this group of points for these genes that show a low read count, ok?

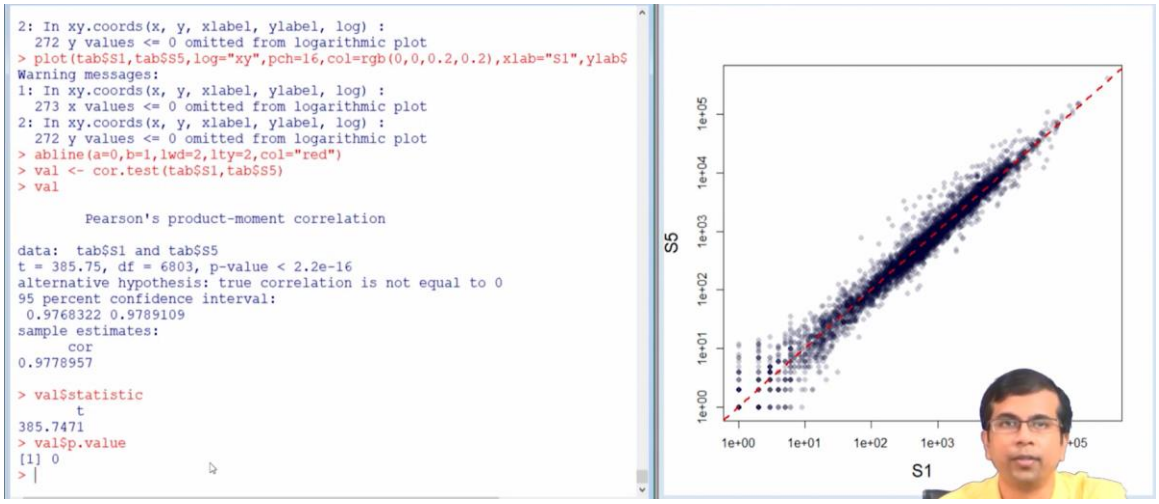
So, this is again expected. We have talked about this because there are a lot more fluctuations that will be observed in this data. Now, this is all good; we have generated a very nice plot. So, one of the things we want to do now is actually look at the correlation. Can we do this correlation calculation? And it turns out that you can in R with the command that we use. So, let us say value equals to, right, so we are assigning this value here correlation dot test, ok?

So, this is the command in R, ok? And we just need to give the names of the points in the data sets for which we want to calculate this correlation, ok? So, it is very simple. We call the test tab color S1, tab color S5, right. These two data sets, or these two columns, we want to calculate the correlation, ok? And this correlation is now generated and stored in this variable value, ok?

And we can then type value, and it says, "Okay, Pearson's product-moment correlation. So, it is Pearson's correlation. And then the p-value is less than a certain threshold.

You can see this is 2.2 into 10 to the power minus 16. This is actually the limit. And the correlation value here is 0.98, ok? So, this is a very good correlation between two replicates, right?

And you can also get the exact p-value. You can simply write well dollar p dot value, ok? So, again, like the data set for this variable, you can also have this dollar and many of these values you can simply find out by typing this using this tab after you type the dollar. So, the tab dollar statistic will give you the statistic that has been generated here.



You can see this, right? So, 85.1. Similarly, the dollar p dot value, right, will give you the actual p value. So, it is here. So, it is a very, very low p value, right? So, it is not 0, of course. It has an extremely low p-value because these two data sets are very highly correlated, ok?

So, we can change this correlation format again, right? And you can say method equals So, instead of Pearson, let us say we want this Spearman, ok? So, it is giving me an error because I have typed in capital S, which should be in small. So, it will tell you that in case there is an error in your command, it will tell you, ok.

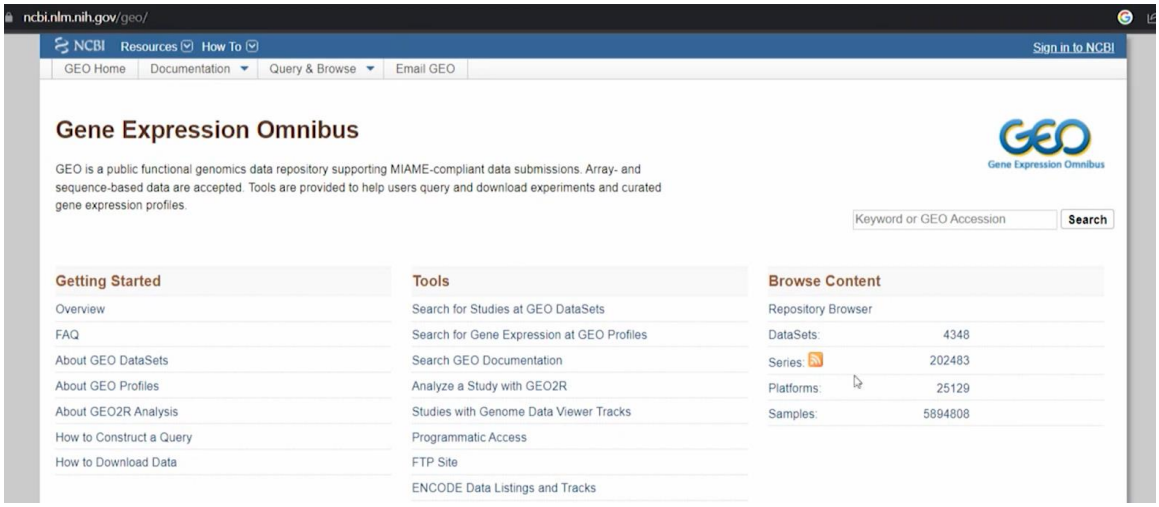
So, the argument should be one of Pearson, Kendall, or Spearman. Pearson is the default correlation it calculates. So, if you want the Spearman correlation, this is the rank correlation, right? It will give you some value. It also gives you a warning, right?

So, because you probably have these ties in the data, right? So, again it will not give you the exact one, but it is ok, right? Again, this p-value can be obtained.

You can simply write value. It is 0.983. So, it is slightly different from what you got for Pearson, ok? So, for this data, it does not really matter whether you are taking Pearson correlation or Spearman correlation, but for some data sets and also depending on the situation, you might want to choose the right correlation that you will use. So, one of the things I wanted to show you now is that we have seen that we can now plot these data sets,

look at correlation, do very nice things, generate very nice plots, etcetera using R. So, what I will do now is actually show you where you can get these data sets, right? So, if you have your own data, you can generate this raw count data using this mapping and then counting, right?

But if you do not have your own data, if you want to do this hands-on, you can, of course, get access to some of the data sets that are freely available. And these data sets are available in NCBI-GEO, right? I have shown you before, I think, but now I will show you where you get these data sets, ok? So, what I will do is go into this series, right? So, if you search NCBI-GEO, you will come here, and once you go into the series part, you can see a lot of data sets that are coming in, and you can see these very recent data sets that have just been deposited here, ok?



We can take any data set; there are a huge number of data points, but you can see this expression profiling by high-throughput sequencing; maybe I will zoom in a bit, right? You can see the different studies that are coming in; you have expression profiling by high-throughput sequencing, etcetera, right? And we can go on, right, and we can download. You can download anything you like, any data or any data that is relevant for your work, right, from this data set. And let us say, I will click on this one. Let us say this is a human data set, ok?

What happened? I do not know. So, maybe I will click here. Oh, yeah, sorry. So, I will, because when I click on that, it actually goes only to the human data sets, I think. So, I will

just click on this GSE number. So, every study will have a unique GSE number, and once you click on that, you will get a lot more detail about the data. So, you have this title, you have the organism, expression profiling by high-throughput sequencing, right?

And then, of course, you have the sample types, right? So, you have wild types, probably three replicates, and then three knockouts. Of course, we need to look at the paper, etcetera, and see the experimental design carefully to see what is actually happening. And here you can find this GSE 23552 gene expression dot xls dot z, right. So, this is actually, I think, raw count data or maybe normalized count. This will actually describe whether this is raw count or normalized count data, but some sort of count data that is given in an xls file is okay.

We can also look at other data. So, where can we actually get this text file? Right, and it will mention whether it is a raw count. So, again, this is another study, and you can now see here that this is where all these files would be provided. This one is normalized count data, right? So, this is how you can see this FPKM normalization has been done in this region, right? So, this is why it says FPKM expression genes dot text, right?

So, this is normalized count data we know right away, ok? So, depending on the type of analysis that you want to do, right, what else suggests you probably should try to download raw count data because we will go through the differential expression analysis? And when you want to go through the differential expression analysis, we will use the DEseq2 platform, which only takes raw count data. It does its own normalization, etcetera, and it will do its own normalization before it does this differential expression analysis. Here is another study again, and you can see this FPKM dot txt dot gz.

So, this is FPKM normalized, ok? So, this is something you can look at, and carefully choose the right data for your analysis. Of course, you do not want to choose really big data, but a small data set that you can run in a reasonable amount of time. So, hopefully, this is clear. Now, we can actually go back to R again, and we can see how we can install some of these packages.

So, this is all fine. Now, we want to use some packages, right? So, as we have mentioned, we want to use this DEseq2 package, right, and for plotting or for different types of tests, we want to install certain packages, ok? So, if you go back to this R home page, right on the left side, you will see these package options, ok? And for these package options, you can now go into the table of available packages that are available for this system for this version, etcetera. You can see very new packages that are coming in, ok? And these are the names of the packages that are available, with and very short description on the title of the package. Right, what it does, ok?

And you can find a huge number of packages, and only some of them are required for our purpose, right? But then again, we can install them through the terminal, ok? You do not have to go through this, right, and find out, right, which package is doing what, right. We can, if we know which package we want, simply install it through the R console. So, one thing I will do—maybe what I will try now—is look for a package that you can install, ok?

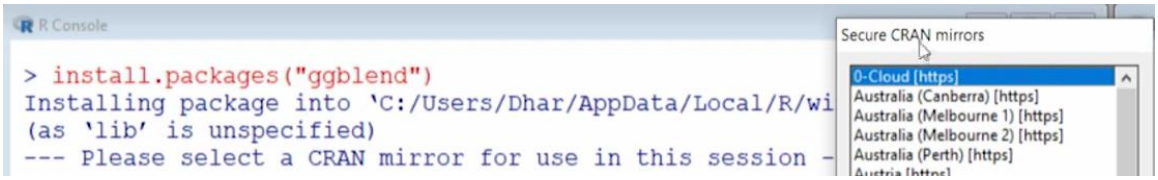
So, let us see, we will need a package something like, let us say, this ggplot, right? So, here it will again, once you click on that package, you will come to a page like this, and it will have a description of what this package does, right, and which version of R you can use for this, right. So, this is mentioned here, and again, you have some more details about the authors, right, and also the reference manual here, ok? So, this is the most important document. Once you click on that, you will have a PDF manual that will describe the functions that are available within that package, what these functions do, and some examples.

So, this kind of information would be available for all packages. Now, let us say if we want to install this package, right? So, let us say ggplot, ok? We want to install this package here, ok? So, to do that, we can simply go to the R console here, and we will clear this window.

So, that will be easier for you to see. Here, you simply write install dot packages ggplot,

ok? Be careful with the spelling as well as the capital letters. So, you can simply write this to install dot packages in ggplot. You can choose any mirror. We will just choose the default one, the zero cloud, and it should install this in a moment, ok?

So, you will see something like this which will just install it. So, here, you see that this is installed. In some cases, it might install some of the dependencies, etcetera, and some of the required packages along with this package, ok? This process is exactly similar. If you are working in Linux, you can simply type install dot packages ggblend, ok? So, now if you want to use this package, what you have to do is load it by saying library ggblend, ok?



So, now it is saying, ok, this package was built under R version 4.3.1, but I have 4.3.0. So, it might give you some warning messages, but that is ok. So, you can also use certain packages that we will use, right? For example, ggplot2, ok. So, whatever package you want to use, you simply type. If it is installed, you can simply type library, then package name, and this will be loaded, and you can now use these commands from this package.

If it is not installed, it will tell you, ok, you cannot find this package, the name of the package, right, and then you can say install dot packages and the name of the package, ok. So, this works for most of the R packages, right? But there are other ones as well, ok? So, when we are discussing the theory etcetera, about bias correction, different methods, etcetera, right, we talked about that there are some Bioconductor packages, ok, and in R Bioconductor, ok. So, this is actually part of R that contains all the packages that are utilized for bioinformatics, ok? So, you will see a lot of these tools for transcriptomic data analysis are present in Bioconductor.

So, the first one is DEseq2, ok? So, we will use this DEseq2. You can search here at the top. You can see this search button on top here, right, and we can simply search DEseq2, right, and this will come, ok? So, the first option you can see is conductor DEseq2, and you can see the package name, right? So, you have differential gene expression analysis based

on the negative binomial distribution, right? It gives you the author names, the paper, the reference, etcetera.

Now, the installation of this Bioconductor package is slightly different. You will see, right? First, you have to use these commands. It tells you what the commands are.

Now, this is under version 4.3, ok? This is the latest version, right? You can simply copy this and type it on your R console, ok? So, I am simply running that, ok, and this is already present in my case, bio conductor 3.17, and then you write bio c manager install DEseq2, ok. So, the first command, this first part of this command, is installing Bio C manager which will manage the installation of all the packages from Bio Conductor in the future.

```
> if (!require("BiocManager", quietly = TRUE))
+     install.packages("BiocManager")
Bioconductor version 3.17 (BiocManager 1.30.21), R 4.3.0 (2023-04-21
  ucrt)
> BiocManager::install("DESeq2")
'getOption("repos")' replaces Bioconductor standard repositories, see
'help("repositories", package = "BiocManager")' for details.
Replacement repositories:
    CRAN: https://cloud.r-project.org
Bioconductor version 3.17 (BiocManager 1.30.21), R 4.3.0 (2023-04-21
  ucrt)
Installing package(s) 'DESeq2'
```

So, once you have installed this, you do not have to install it again, ok? So, let us say we say DEseq2, ok. We want to install this package, DEseq2. So, it will take some time and will tell me, ok, that this is already installed because it is, right? So, I have already installed this. So, it should find this installation and it might just say, ok, this package is already installed, ok, depending on the internet connection, ok, ok.

So, I said, Let us know. I do not want anything, ok? So, maybe it has found some updated version of DEseq2. Now, we can load this DEseq2, right, and see whether it loads here, ok? So, it should come here, ok, and in a moment, it should load, ok. Some of the things might take a bit of time because they are quite heavy, ok? So, as you can see, it is loading.

When you type this library DEseq2, it is loading other packages that are required for this package. So, similarly, you can see a lot of commands, right, but as long as you are not

getting any errors, right, you are fine, ok? We can also try another package, right? So, that we will use is, by the way, I remember these packages that we need. So, I am just installing them, but as we go on, if you see some packages are missing, we can install them one after another, ok?

So, again, it will ask for updates. So, I must say no because it might take a long time, and it is saying to me here that this package is not installed, right? Because this is already present, ok? The version that you and I are to install is already present. So, I do not have to install again. So, it is just giving me this message, ok? And similarly, we can load this, right, library RUVseq; sorry spelling; again, you have to be careful about this, and it will load this package here.

We can use the commands inside this package for all our analysis, ok? So, to summarize, we have now seen, right, where we can get these datasets, right, from this NCBI GO database, right, and we can try these different sorts of things in R, right. So, that is what I will encourage. If you have your own dataset, that is great, and then you can actually load it in R once you have obtained the count data, and you can load it in R, and you can then start doing all sorts of analysis that we are doing. I have also shown you how to install packages, so any package that you need for your analysis, you can use this install.

packages command, and it should be installed very quickly, and you can load this using this library command, ok? And finally, there are packages that are utilized for bioinformatics that are available in the Bioconductor. Again, this installation command is slightly different, but again, you can install it very easily from the R console. So, that is it for this class. So, in the next class, we will start with the preliminary analysis of the data and follow the steps one after the other. Thank you very much.