Next Generation Sequencing Technologies: Data Analysis and Applications

Transcript Abundance Quantification

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 few classes, we have talked about transcriptome read mapping, and then we talked about transcriptome assembly. We started discussing the quantification of transcript abundance, and this is a very important step. In this class, we will be continuing the discussion on this transcript abundance quantification. So, here are the concepts that we will be covering in this class. So, we will talk about the read data to count data process, we will talk about some tools, and we will also have a look at how to do this ourselves with a small hands-on approach, and then we will talk about alignment-free approaches for transcriptome quantification.

So, many of the tools that we have discussed rely on alignment maps or SAM or BAM files for this quantification, and some of the tools that we will discuss also rely on these alignment maps, but there are tools that can actually do this without doing any alignment, and they are actually significantly faster than alignment-based tools. So, we will discuss those as well. Here are the keywords: raw count data and model-based estimation. So, here is the data processing pipeline again for RNA sequencing data.

We have talked about mapping; we have also talked about assembly in very brief, and we have come to the quantification step. Now that we have discussed some of the tools that can do assembly, they can also do quantification. For example, we have talked about string ties and cufflinks. So, they can also do quantification after the assembly process. In this class, we will be talking about some of the tools that are actually dedicated for quantification purposes because this is a very important process for any downstream analysis.

So, you want this accurate, very accurate quantification of the transcript abundance for any

inference later on, ok? So, if you make mistakes here, those will lead to erroneous inferences, ok? So, this is a very important step. So, this is something we will discuss today: quantification. So, there are several tools and cuff links we have already discussed, but then you have something called htseq-count, feature counts are the same, etcetera, and we will discuss those in this class, ok?

So, htseq-count actually simply counts the number of reads that map to each gene transcript or feature, right? So, this is the simplest of all the tools. It takes the alignment map and then the feature file, and it will simply count how many reads map to each gene transcript or feature. So, these are the input files: right, alignment map, exam format, and you can give feature annotation, which is a GFF file or GTF file, and then output what you will get is a table with counts for all features, ok? So, this is a very simple process that simply counts.

And what are the challenges here? Of course, this is a very simple process. So, one is, if you have read that map to multiple features or multiple genes, ok. So, this is something that again, we go back to the mapping to the repetitive sequences, right? So, if you have reads that map to multiple features and genes, how do you count those reads? Do you count them for both genes or both features or do you just discard them? So, what are you going to do? And this is something that is a very difficult question to answer, right? And we cannot answer this by simply looking at a simple counting process.

Of course, there are more sophisticated models that can deal with these kinds of reads that actually map to multiple features or multiple genes. There is another tool called feature counts which is very similar to the htseq-count. So, again, it provides gene-level counts and will not provide any counts for isoforms, right? Again, as we have seen, doing this for isoforms is very difficult, especially from short-read data. So, again, here the input files are the alignment map, the SAM BAM file, and then you have the feature annotation file, which is the GFF file.

And here is the package that you can use, download, and use, or you can get it from R in

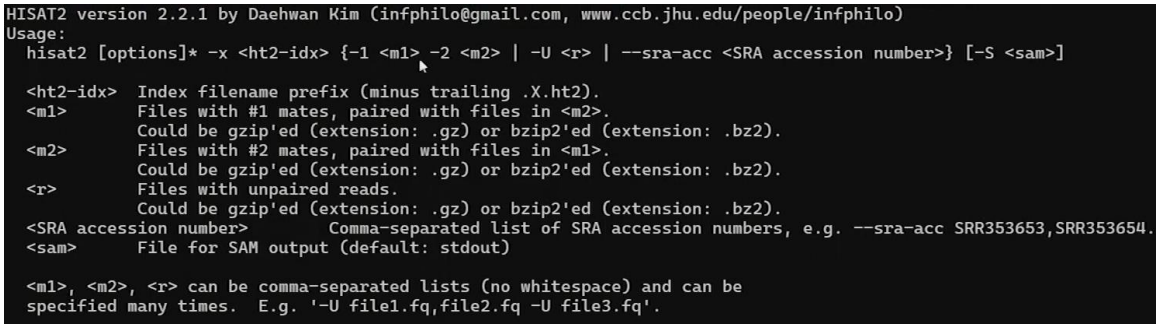
Bioconductor and you can use it, right? So, again, if you are not familiar with R, you can simply go with the subject package and download and use that package. So, what we will what we are going to do is we will have a hands-on now. We will just look at this raw RNA-Seq data, and we will get to the raw count data using this very simple step, ok? We will not talk about the QC and pre-processing because you already know this: right, first QC, etcetera.

If you need to do trimming, etcetera, you can do that. What we are going to do is take an example of this raw RNA-Seq data. We will map using HISAT2, of course, only a small part of it, and then we will try to quantify with an HT-Seq count, ok? It is a very simple step that I can show you right here right now, ok? So, let us now go to the terminal, ok, and we will open the terminal, ok, and here we are and what we need are these two tools.

So, one is the HSAT 2, right, and another one we need is the HT-Seq count, ok? So, here is the HISAT 2 tool. This is the link where you can find this, ok, and here is the link for HISAT 2, where you can find the tool, and again you can check the manual here, ok. Here is the manual, right, and you can find a lot of details, ok, and how to download, install, etc.; in addition, how to actually run the HISAT 2, what are the options, etc. So, what I have done is that I have actually installed this tool already on my system, so I can simply use it.

So, as you can see here, we have this HISAT 2 folder. I have this HISAT 2 folder here that I actually have. So, this is the program that I downloaded. This is the compressed file. Then I extracted, and then I actually added this to the path.

So, I can simply call HISAT 2, and I can get all the options, etcetera. So, this is what I am going to do, right? I am just going to run this HISAT 2 with minus-minus help. So, it will tell me what the options are, etcetera, ok? So, again, what you will see is that HISAT 2 is very similar to Bowtie 2 because, again, it uses this BWT algorithm, right, the Burrows-Wheeler transform.



So, the tool has also been kept very similar to yours. So, it is actually very easy to use if you know this bowtie, right? So, we have HISAT 2, and we have minus x. Again, we have to provide the index. Then we have the minus 1 M 1 minus 2 M 2 minus U R, right?

So, we are going to run this right here now, ok? So, all these options are there. So, you can simply run this with a dataset. Now, what I have done is downloaded one dataset here from NCBI GEO, ok? And I will show you that later during the hands-on, when we have a full hands-on demonstration of how to download these datasets.

And from this, you have created only a part of that file, right? I have just created a small file from that because that file is quite big; it will take a long time to run or to map, ok? So, if you want to do this in a short period of time, we will just use this part file to do one part dot first, ok? So, let us run this now.

HISAT 2 minus x, right. Now, we need the indexes, ok? Now, what is good about this is that HISAT 2 can build these indexes, but you can also download indexes for some genomes. So, for example, you can check here; you have specific species here; you already have these indexes built, ok? So, you have this homo sapiens, etcetera. So, I have downloaded this index, which is for Saccharomyces cerevisiae because the data that I have downloaded is for this species, Saccharomyces cerevisiae.

And you can see different types of indexes, right? So, you have genome SNPs, right? So, you have these single-layer polymorphisms with the genome that has been added, the genome transcriptome, right? So, you have the genome, SNPs, transcriptome, etcetera, right? And similarly, for different species, you will see these different types of indexes,

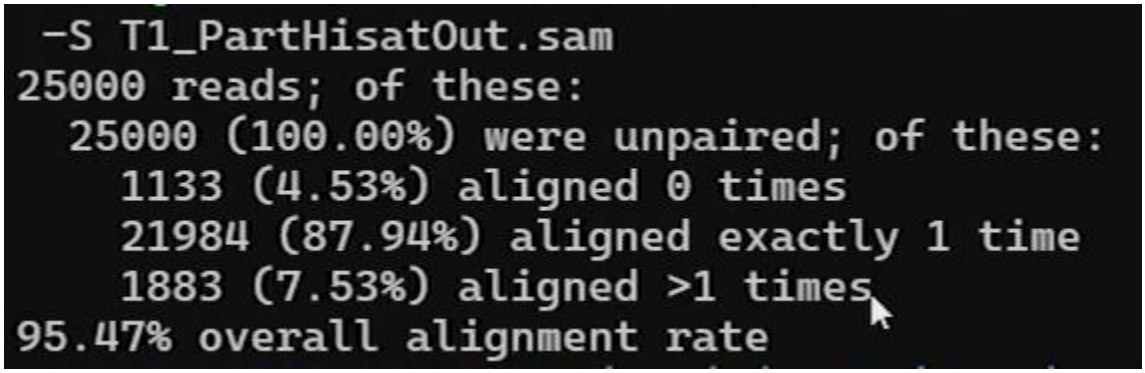
ok?

And for Saccharomyces cerevisiae, you have the genome as well as the genome transcriptome. So, I have downloaded this genome transcriptome. If you just click on this, ok. So, it will download, but it will take a lot of time.

So, I will just stop. I have actually downloaded this already, and it is present here. So, if you will check, this is the R64 underscore trans, this is the sequence, and this is the index file that is present in this folder, ok? So, we will use this index, ok? So, we just need to give the base name, right? We do not need to give all those dots 1 dot ht 2 dot 2 dot ht 2, right?

This is very similar to both I and II; remember, we just use the base name for this purpose, ok? So, the next step is that we have to give minus u because we are working with unpaired data. We do not have paired-end data, right? And this is the t1_part dot fastq. This is the fast point that we want to analyze, and then we have minuses, right?

We can simply say, right, part is beyond dot sam, right? Again, this program gives output in the SAM format, right? So, we will have to again mention that the minus s part is is without dot sam, ok? So, part means this is only part of the full data we are working with. So, that is why I gave this name part, ok?



Maybe we can just say t1 part, right? Because this is from the t1 part file, alright? So, let us run this and see, ok? So, what is the output? We have 25000 reads, ok? These are unpaired reads. Then it gives very similar statistics, like both bowties, ok?

You can see this, right? So, some reads did not align, some reads aligned exactly one time and some reads—about 7.5 percent—aligned multiple times. Now, this is where this part is actually interesting, right? This is what we will do with this, ok. If they are aligning to multiple genes, whether you count for both genes or simply discard them from your analysis, ok.

So, this is where more sophisticated tools are actually important, right? So, which we will discuss a bit later, right? What are these tools that we can use to read data that aligns to multiple positions in the genome or in the transcriptome, and how do you utilize those to actually estimate transcript abundance? Because these are discarded, discarding them means we are underestimating maybe the abundance of some transcripts, and counting them for multiple genes or transcripts might lead to overestimation for some abundance of some transcripts, ok? So, this is a very important problem, right?

It is not just for genomes; it is not like just genome sequencing. Here, the count—the number of reads mapping to each feature—is very important. So, the overall alignment rate is 95.

5 percent. So, this is a good alignment rate. So, we can now go ahead with the quantification. Okay, alright. So, let us see now if we can do the htseq-count, right? So, let us see if we can, ok? So, I have also opened this.

So, here is the function htsec count, right? So, how do you actually use this function? So, again, here you have all the details of this, right? You can find a lot of these options, and you have to install a program called htsec. You can see this, right, this program and again, the prerequisites and installation; these are all given, ok, and you can check them, and you can see that how do you install all this? So, if you are using Python, you can use this pip install htsec, or if you are using Linux, in Ubuntu, you can simply say sudo apt-get install.

We have talked about this in the Linux installation process, right? We can do that,

right? Simply put, we can use this sudo apt-get install, and then, of course, in Windows, you can also have different download packages, etcetera. So, let us see if we can run this, right, htsec count.

Here are the options. Again, we will not use any option. So, what it needs is the alignment file, the SAM file that we generated, and the feature file, right, the GFF file. So, we have downloaded the GFF file. So, let us see, right, the GFF file is here, GFF GTF. You can see this, right, inside this folder, and I have simply just copied this GCF64 genomic new dot GTF.

This is the file that you can simply use, ok? So, let us see if we can run this htseq-count because I have already installed it, right? So, what it needs is the alignment file, which is the SAM file, ok, and this is the SAM file that we generated, ok, and we need the GFF file, or GTF file. So, I am using this dot GTF and let us see, ok? So, it should run, and it will generate statistics, ok, and you can see these statistics, ok. So, what I am going to do is, I am going to save this into another file, ok, and in it inside a text file, ok.

So, you can see, right, that we are not getting the output in the terminal, right?

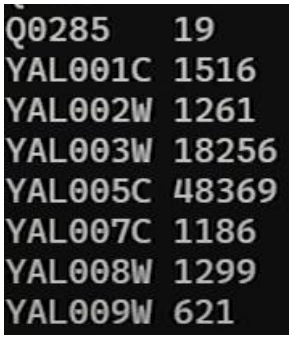
```
rdhar@LAPTOP-3K4C9VBI:/mnt/c/Users/Dhar/Desktop/NGS_Data_Analysis_HandsOn1/RNAseq_data_analysis$ htseq-count T1_PartHisatOut.sam GCF_000146045.2
_R64_genomic_New.gtf > Count_T1partOut.txt
```

I can, and what I can do is simply run it again so that you can see clearly, ok, and it gives you, gives out some statistics, etcetera, right, and what you want to do is, we want to check this count file, right, that we generated, ok, because this will contain the data, ok, and this is where you can see the raw count now, ok. So, the first column you see is the gene names, ok, and the second column is the raw count data, ok. What it simply says, ok, there are, for example, in this gene, right, you have 9 reads mapping to this gene. For this read, you have, for this gene, 32 reads mapping to this gene, right?

Similarly, you can see this, right? For many genes, you have this kind of statistic, ok? Now, you might say that for many genes, we see 0, right? That is expected because we are working with only partial data, ok? We are not going to working with the full data, right? If you work with full data, we can see counts for all genes, which is almost okay. So, what

I have actually done is processed the full data with HISAT 2 before and I have created this file, which is actually present here.

You can see this HISAT map out here, ok, and this is the processing of the full FASTQ file here, ok. It took some time—right, a few hours. So, that is why I have to run in the background, and then I have generated this sample so that I can show you what the full output would look like, ok, and then I have processed this with the htseq-count, and you can see this raw count HISAT map out dot text, right? So, let us have a look at that raw count file because this is after processing the full data, ok, not just the part data, and you can now see that for most genes you have some counts, and for some genes the counts are very high, right? For example, you can see this gene here, right? You have 48,000 reads mapping to this gene, ok, but still, you have some genes that are probably not expressed because you see these 0 counts, right? And now, because you have so much data, you can be sure, ok? These genes are probably not expressed or have very low expression, but what you see for most genes, you have expression, right?



You can see some read counts, and this is the raw read count, ok, and this is the way; this is a very important step for all subsequent data analysis, ok. So, hopefully this is clear how we actually simply generate this kind of raw count data from the mapping, ok, using a tool like htseq-count, ok. So, what we will do is now talk about some of the more sophisticated tools that can actually deal with these multiple mappings, etcetera. So, let us go back to the presentation and let us continue our discussion on some of the other tools that are available and that can deal with this multiple-read mapping, ok?

So, one such tool is called the RSEM, ok? So, this tool in its full form is RNA sequencing by expectation maximization, ok, and so this tool does not require any reference sequence.

It can assemble transcripts as well as quantify transcripts, ok? So, we have discussed many such tools before. Now, there are several steps here, right? So, one is the generation of reference transcripts from the read data, right, and then the reads are aligned against the reference, ok, and based on this alignment, you have the estimation of transcript abundance, ok.

So, what is unique about this tool is that it generates something called a maximum likelihood estimate of abundance along with confidence intervals. So, it kind of gives you, ok, what is the 95 percent confidence interval for each feature, right? So, it is not just a single number for counting. You also have a confidence interval, right? So, if that confidence interval, if that value is like if the variance is very high, right, then you probably cannot trust the data that you are getting, ok?

The second unique feature is that it utilizes something called the expectation maximization algorithm. Again, we are not going into details about this. So, it is again related to the maximum likelihood estimate, but it goes through two steps. One is the expectation, or E-step, and then you have the M-step, ok?

So, it performs maximum likelihood estimation. So, then it actually does this in an iterative process, ok? So, you have E-step, you have M-step, then again E-step until you converge, ok? Again, we are not going into the details of this, but again, you are encouraged to actually look into these algorithms if you are also interested in the algorithm part. So, what we get after this tool is that we get gene and isoform level estimates of abundance. This tool can give us an estimate of isoform abundance because it utilizes these maximum likelihood estimates, right, using these expectation maximization algorithms. So, if you look into the algorithm, you will see that there are variables that cannot be directly measured or quantified, but they can be predicted from the data that is there, and this EM algorithm actually tries to optimize the parameter values so that it feeds the data.

So, that is the idea behind the maximum likelihood estimate, right? Given the data, what is the most likely value of the parameters or the most likely value of these abundances in this

case? So, this is how it can actually generate these estimates of abundance, and you can also see it can handle these multiple mappings, ok? So, now what we are going to do is discuss some of the alignment-free methods that are out there, ok? So, what we have discussed so far is that most of these methods require some sort of alignment, right, against the reference transcript or a reference genome for quantification. Now, this mapping step, this alignment step, is the slowest step in the full process, right?

So, if you look at the pipeline, this is a very slow process because you have to align this huge number of reads—millions and billions of reads, right? So, that is why these very efficient algorithms were designed, like the BWT, etcetera, and also the HISAT2. Now, if you can completely skip this alignment step and if we can still get to the abundance, right, directly from the read data somehow, ok, and this is what these alignment-free methods actually do, ok. And there are several methods that we will discuss very briefly again without going into the full statistics, the algorithms, etcetera.

So, these tools are called callisto, salmon, and shellfish. So, these are the tools we will discuss. There are other tools out there as well. So, let us discuss the first tool, Callisto, ok? So, it quantifies the abundance of transcripts, right? This is one of the tasks that this tool does, but it does so based on something called pseudo-alignment.

There is no real alignment. So, that is why it is called the pseudo-alignment or alignment-free approach, ok, and the advantage is that this is much faster than the alignment-based approaches, ok. So, you do not have to do this alignment or mapping. So, it is actually much faster. In addition, it is less sensitive to sequencing errors. So, because you are not aligning exactly, you do not have to worry about these mismatches in the read data, etcetera.

So, this is actually less sensitive to sequencing errors than you can get, ok? So, what is this pseudo-alignment approach? So, the pseudo-alignment approach is that it actually tries to identify transcripts, from which a read could have come, So, given a read, we want to identify which transcript this read could come from, right? That will serve the purpose of

quantifying abundance, ok? And again, it is based on a deep brain graph approach for the transcriptome. We are not going into the exact algorithm and details, right, but this deep brain graph approach we have mentioned also in the transcriptome assembly, and it will come again when you talk about a genome assembly, ok?

So, if you are interested in the full algorithm and the details, here is the reference and also the tool you can download from this link, which is given here. So, feel free to look into the paper and also the tool and see the options that are there, ok? Then, we have another method, which is called shellfish. So, again, this is an alignment-free method, and it can also help in isoform quantification, ok?

So, as I mentioned, this is an alignment-free method. So, what it does is actually utilize something called an indexing of transcripts, ok? So, it is a very similar idea to what we have talked about earlier: seed, seeding, etcetera. It creates indexes of certain length, right, k-mars of certain k value, ok. And then it looks for these k-mars, right, from the read data.

So, it is kind of like a hash-based method. So, it searches for these indexes from the read data against the reference transcript, ok? And from that, it can quantify the transcript abundance. And similar, it can actually generate what the k-mers are from isoforms, different isoforms, and it can find whether these k-mers are present in the read data, etcetera. So, this method can also account for different biases in RNA-seq data, ok?

So, we will talk about these biases in much more detail in the next class. And you will see that this model and this method can actually model those biases when processing RNA-sequencing data. The other tool that is very popular is SALMON, ok? So, again, we will not go into the statistics. I will just very briefly mention what it does.

So, it actually uses something called the dual-phase statistical inference procedure. And it can also take into account RNA-sequencing biases, for example, sequence-specific biases or GC-content biases. Don't worry about these biases which we will discuss in much more detail. So, there are two phases in this algorithm, as you can see, right? So, this is the first

mapping model that actually estimates expression levels and model parameters.

So, it is a model, and it is a model-based estimation of the abundance. And then, after the estimation, it kind of refines these expression estimates, ok? So, using these two steps, ok. So, here is a reference for SALMON, and you can have a look into the original paper. You also have the tool available for download from this link, and you can, again, explore and learn a lot more about these methods. So, here are the references that we have used for this class. So, what we have discussed is how we actually go from raw read data to raw count data, right?

So, this is the quantification step in RNA-sequencing, and this is a very important step, right? So, we have now the abundance of transcripts, quantified in terms of numbers, as you have seen. So, we have done the hands-on and we have seen, right, we have these abundance values generated from the map data, ok? And we have talked about some of the tools that can do this very simply, right?

We have this htseq-count and the feature counts. We have not used feature counts, but we have used htseq-count in our analysis, and we have generated the raw count data. And this raw count data serves as a proxy for the expression level. Of course, you cannot take this count data immediately as the expression level because there are a lot of biases in the data that we need to account for before we can actually compare expression levels. So, we will talk about these biases in the next class, ok? And you will understand, right, how we actually go from the raw count data to the actual expression level, right?

So, that is the idea, right? We want to measure the expression level of genes from the read data that we have. So, we have proceeded a bit, right? We have now got the raw count data, and we can generate it. And from this one, we now want to go to the actual expression levels of these genes, right?

So, we need to account for a lot of biases along the way. We have also talked about all the algorithms that can account for these multiple mappings of reads. So, this is a problem with

simple tools, right, to HTseq-count, or the other tool that we have just discussed is that they cannot decide, right, whether this comes from a specific isoform or it is just counted as a gene, right? So, in those cases, this is something that can be a problem, right, where the isoform expression is of interest in your experiment, right? So, this is something that some of the tools can deal with. For example, the RSEM, which we have discussed, can provide gene and isoform level abundance estimates.

It utilizes something called the expectation maximization algorithm. Again, without going into a lot of details, it fits a model and tries to generate this maximum likelihood estimate from the data. There are other tools, and that kind of tool does very similar work. For example, we have something called ISOEM that can also give you isoform-level abundance estimates. We then talked about some of the alignment-free approaches, right?

So, for transcript quantification, we talked specifically about three tools; just mention them. So, one is Callisto, the other is Selfish, and the third is Salmon, right? And these tools do not require any alignment, right? So, they can go from the read data to the quantification part, right?

So, there is no mapping or assembly step. So, this is actually jumping through the mapping part, right? So, again, this kind of reduces the time requirement because mapping is the most time-consuming process. If you imagine, you have to map—let us say billions of reads. The other advantage is that you do not have to worry about sequencing errors too much because these are less sensitive to those types of errors. And these methods are substantially faster than mapping approaches.

So, that is why they are quite popular now. So, you can have, you can look at, you can look into those references, and you can see, right, there are time comparisons between these different tools in those papers. And you can see that these algorithms, Callisto, Selfish, and Salmon, are significantly faster than all those tools that we have discussed that rely on alignment. And some of these models can also incorporate models for biases, right? So, we have a lot of biases in RNA sequencing data, and some of these tools, these alignment-free

methods, can also take those biases into account when quantifying this transcript abundance. So, we will talk about these biases in much more detail, as well as the methods to deal with them, in the next class. Thank you.