Next Generation Sequencing Technologies: Data Analysis and Applications

Data formats

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. So far, we have discussed the next-generation sequencing technology method, and then we discussed some of the key terms that are important for the course. We will be using those terms throughout the course. We have introduced coverage and quality scope. And finally, we talked a little bit about experimental design.

So, if you are planning to do next-generation sequencing experiments yourself, you will know what are the parameters, what are the aspects that you need to look at. In today's class, we will be focusing on data formats. So, you have designed your experiment, and you have sent your sample for sequencing. You get the data back from the sequencer. So, how would you interpret the data? How would the data look?

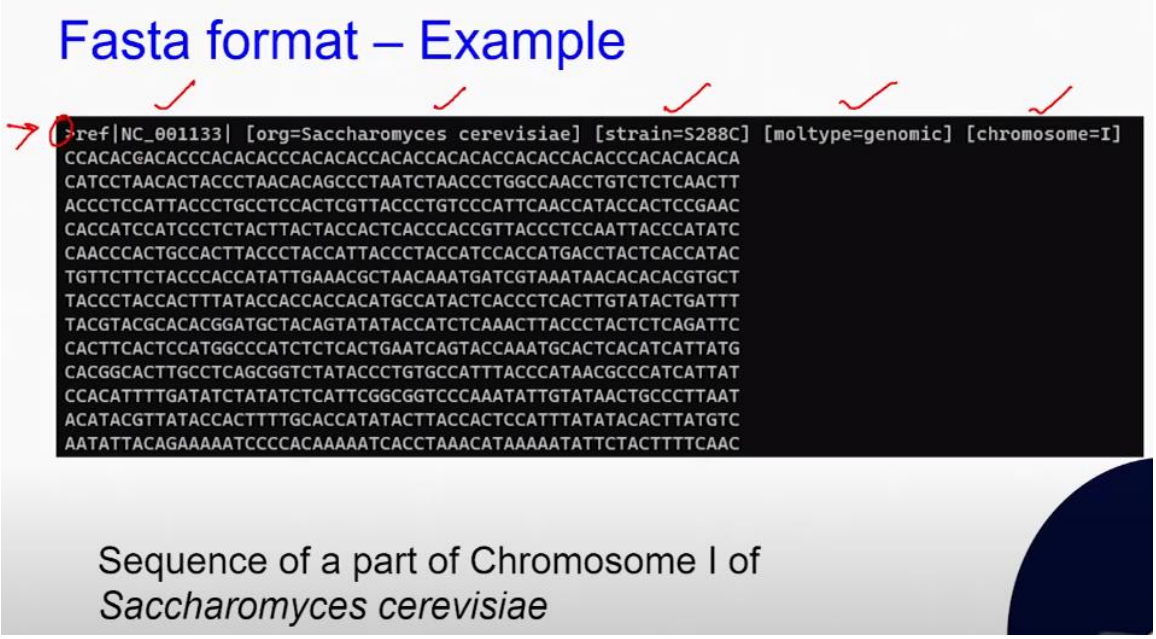
So, this is something that we will be discussing today. And specifically, we will be talking about two data formats in this class. One is the FASTA format, and the second is the FASTQ format. So, these are the two formats that we will be discussing first, and then there are other data formats that we will be discussing in the next class. So, these are the keywords that we will come across during today's class: FASTA, quality score, and FASTQ OK.

So, to give you an overview of the full process, So, here is what you have done right. Up to this point, hopefully, you have decided on the sequencer, the platform, and the read length that you want to use for your experiment, and you get the read data from the sequencer. And this data can come in different formats, depending on the platform that you have chosen. Now, once you have this right, the first step will be the data quality check, and this is something we have briefly mentioned.

Then there will be different sorts of problems right they separate out into two branches. One is the read mapping against the reference sequence if you have a reference sequence already. And this is something that we do if you are interested in looking at single-length polymorphisms, indels, or structural variations or if you want to do transcriptome analysis. We are looking at gene expression patterns or gene expression changes between different samples, and also for epigenomic studies if you are looking at DNA modifications genome-wide. We will look into all these different types of analysis step by step in subsequent classes. On the other hand, we have the read assembly problem, where we are building a new reference genome.

So, this will require different sets of tools and different sets of algorithms, and again, we will be discussing this part towards the end of this course. So, here we are now, and step by step, we will go along this flow chart. As you will see, as we progress, we will come down along this program. So, today we will talk about these data formats, as I mentioned. So, the first one is in FASTA format, ok? So, many of you might be familiar with the FASTA format. This is a standard sequence data format.

So, we use it for storing DNA data as well as protein data probably have seen, but I want to just make sure that everyone is on the same page. So, in FASTA format, the header's first line starts with the greater than sign OK. So, this is something you will notice, and we will see an example in the next slide. This greater than sign is followed by a sequence ID which is usually a maximum of 25 characters. So, it could be anything that you want to put in their sequence that will identify your sequence, and so, these first two parts are mandatory, right? These two things would have to be there.



The next parts right here are the additional information about strains, chromosomes, cell lines, etcetera, or if you want to put something like gene information, etcetera, those could be there, ok? So, these are optional parts, right? This part is optional, and in some cases, you will find this information, but you will always have this part right in this header line with the greater than sign followed by the sequence ID, ok? So, just to give you an example, this is the sequence of part of chromosome 1 of this PC, Saccharomyces cerevisiae. As you can see, this is the header line here, right? You have the greater than sign. Yeah, you can see this greater than sign, and this is the sequence ID, right? Then you have some optional information, right? You have this organism named Saccharomyces cerevisiae. This is the strain molecule type. This is genomic DNA. Right? We are looking at the genomic DNA sequence. Also, you have a chromosome equal to 1. So, chromosome 1 is part of chromosome 1, okay?

So, this is the header line, and then you have the sequence information, right? So, from the actual sequence information, you can see this CCAC, etcetera, etcetera. So, this should be your first format; you have probably seen this before. So, Sanger sequencing data, it turns out. So, get two files, ok?

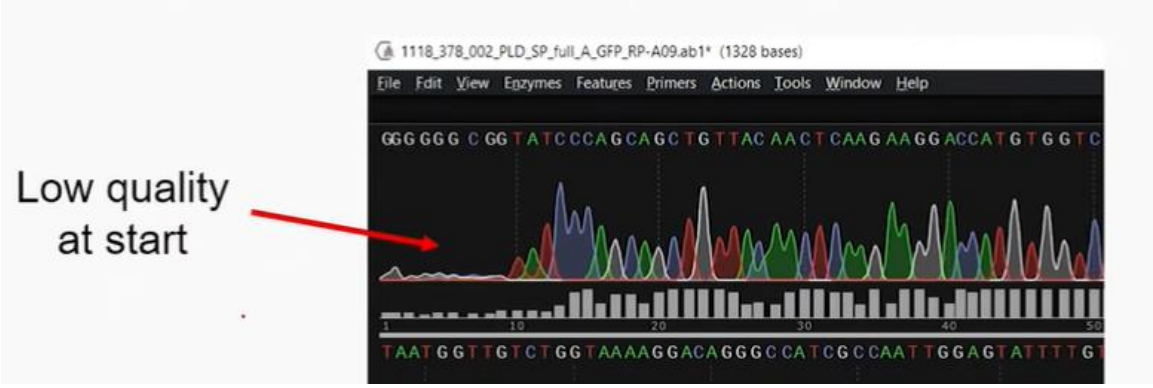
So, one is called a trace file. For example, you will see something like sequence 1 dot ab1 right here as an example. So, these are called ab1 files, and whenever you do Sanger sequencing, you will get these ab1 files. How do they look? We will see in a moment. So, it is important to actually see what Sanger sequencing data looks like, and then we will compare it against what you know. So, compare with the NGS datasets, right? So, this is something you get in Sanger sequencing: the trace file in addition will get the sequence in the first format, ok?

So, we just talked about the first format. So, we will get the sequence 1 dot first, or sequence 1 dot FSA, or something like that, ok? If you open this ab1 file right now, you will see something like this. So, this is called the chromatogram data, where you have this sequencing data.

So, ok. So, what you see here is So, you have like 3 rows of information and 3 rows of sequence here in this diagram. So, in the first row, this is the sequence ok. This is the sequence in the first row, ok, and for each space, you have a peak ok, and these peaks are also color-different. So, if you go back to the Sanger sequencing principle and how it works, we have these different DD entities, which are labeled differently. So, these levels are kind of shown in different colours here, right?

So, T is in red, right? C is in green, A is in blue, A is in green, and G is in white. So, there are 4 different colours you see here, ok? And for each of these bases, you have these corresponding signal peaks. So, for some, it is very high; this peak height is high. So, the signal intensity is high, whereas for some, the signal intensity is slightly lower.

So, but the ah, but this signal ah, peaks are well resolved, right? So, you can call these bases one by one. What you will notice is that in this first part, you said the peaks are almost non-existent. You do not see these peaks right from the first 10 or 12 bases; these peaks are not there, ok? Although there are bases called OK, you can probably guess that these bases will have a low-quality score because the signal intensity is lower and the quality score would probably reflect the signal-to-noise ratio, right?



So, let us look at the quality score, right? So, you can actually, ah, also look at this quality score of these sequence for each base, and you will have a quality score. We have discussed right here how quality scores are assigned. So, here is the quality score, right? This base, for example, represents the quality score, and the height of this column actually represents

the quality score. So, the higher the column, the better the quality. So, as we mentioned in the beginning, the quality score is low.

So, you see, this column height is very, very low. So, you have low quality. So, this is something that is common in Sanger sequencing, and well, this is a well-known issue. As previously stated, this is a low-quality value at this time. In addition, if you sequence, let's say, 1 kb of DNA per 1000 bases per DNA, you will start to see quality degradation at the end as well.

So, here you can see the number. So, we have already sequenced about 1000 bases here; you see this number 1000 here. So, we have already sequenced 1000 bases here, and as we go along right now the quality starts to drop and then starts to fluctuate. So, before this 1000 base position, the quality was more or less ok. Now that the quality starts to fluctuate a bit, you can see this here in this row, and then it goes down further right here, and you can also see the peaks right there, and these are not so well resolved anymore, ok? There are some overlaps, ah, between the peaks, and that is probably the reason why this quality score goes down because you are again measuring the signal-to-noise ratio.

So, if there is a lot of noise compared to the signal, you will get a lower quality score. So, these are the two well-known issues we know. So, the question is: why are the quality scores low in the beginning? So, one is in the beginning part. If you remember the Sanger sequencing principle, we have this DDNTP right and we are generating fragments of all sizes. Now, for very small fragments, the electrophoretic medium that we are using in capillary electrophoresis is not able to resolve those fragments very well.

These are extremely small. So, you can imagine these are like a DNA agarose gel or a protein gel, right? You have some resolution power. Now, this electrophoretic medium right now cannot resolve very small fragments very well, ok? This is one reason. The other reason, of course, is that you have this DDNTP addition, right? These also have some ah; they also have a higher molecular weight compared to normal DNA, and they will shift the small DNA fragments much more than they do for the bigger fragments, ok? So, this will

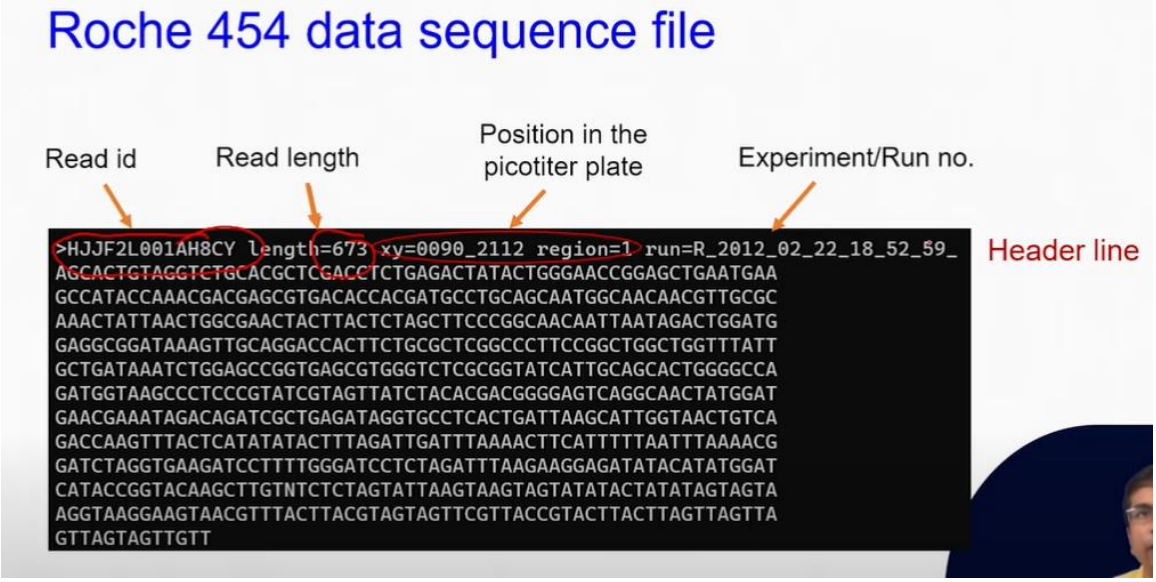
kind of change the expected positions of these small DNA fragments in the electrophoretic medium.

So, these two factors contribute to this low-quality score. So, ah, detection does not work that well. What about the low-quality score at the end? So, this is something we have seen again: resolution power comes into play, right? So, this electrophoretic medium has some optimal range, right?

So, probably around 50 to 1 kB, right? So, 50 base pairs to 1 kB length, right? Beyond that, the resolution is not very good, ok? So, again, what happens is that these fragments tend to now start to overlap with each other. So, they tend to overlap with each other, which means you get this overlap in the signal peaks, as we have discussed.

Now, we are zooming into that, and you can probably notice some of these overlaps here, right? For example, here you can see the signal is for this T right in red, but there is also background noise in grey, which is for G right. So, this background noise is almost equal in magnitude to the signal. So, that is why the quality score, if you notice, is substantially lower here right? So, this is kind of the situation you will probably see in many places now ok?

The quality score is going down because you have this signal-to-noise ratio going down. And probably you can see here, for example, or maybe here also, right? You can see and notice these events. If you look at this chromatogram data carefully, you will see this lower signal-to-noise ratio. So, that kind of gives you an introduction to the sequencing data format for Sanger sequencing. So, now, we can move into the Roche 454 data format, right? So, this introduction would be helpful, as we will see for the 454 data format because the sequence data that you get from 454 is stored in fasta format.



So, in 454 you get one standard format file. So, which is which will see like sequence 1 dot s s f s f f or star dot s f f? So, star is the name that you can give any name that you want, and then this is equivalent to the ab1 file that we have just seen, ok? So, we have seen that this information about sequencing is stored in the ab1 file, similarly, in Roche 454 this sff file, it actually stores this information about sequence information right what actually happened during the sequencing process. You can read this file, and you can generate two different files. Okay, this is something that is done by the program, right? When y you get the output from 454 data, you will get these two different files.

You will get the sff file along with this sequence file, which is the sequence 1 dot fna, for example, and you will also get a quality score file, which is the sequence 1 dot qual ok. So, you will typically see these two files. Now, the sequence file is actually stored in fasta. So, this means we have learned how the first data would look, the first one would look right. So, here is the header. For example, we have this first sequence here.

So, this is the header, and you have the sequence ID right here. This is the sequence identifier and then you have some extra fields right here. These are optional fields that are given in the 454 run. Then you have the sequence, right? The actual DNA sequence is here, and then you have a second header line, ok? So, this is how the data is stored: you have the

first header line, you have the sequence, then you have another header line, the next sequence, and this goes on. So, if you are getting 100000 reads from 454 you will have these 100000 sequences one after another, ok? So, this will be quite a big file, ok? You can imagine how much data will be there.

So, how do you open that file? How do you look at that file without crashing the system? We will talk about that when we discuss the hands-on tutorial, ok? Now, what about these other fields that you have there? So, these are the fields that we see. So, as I said, this sequence ID actually refers to the read ID. So, this is the read number that we are getting; we have the read length right.

So, here are 673 bases, then x and y is the positions in the picotiter plate, and x and y are the regions. This gives you the position in the picotiter plate. If you remember these picotiter plates where you have these wells, each well contains one bead, and on the bead are these DNA fragments, ok? So, this gives you the position on the picotiter plate, okay? And then, followed by that, you have this run number or experiment number right? So, probably you can see the date and that is probably the time when this experiment was run, and this kind of information is there, ok?

So, then you have So, once you have the sequence file, you also have the quality score. So, in SANGER, you get the quality score in the ab1 file, and since you are looking at one sequence, you can probably manually go and check the quality score for each of them, and maybe you are handling maybe 8–10 sequences at a time in parallel, and that is manageable. In 454 you are dealing with 100,000 reads, or 100,000 sequences at once. So, you need to put the quality score in some file where you can process it by writing a program, or some software tools can process this quality ore. So, this quality score is stored in a separate file, as I mentioned, the dot-qual file.

And this stores the quality scores in fasta format and in the same order as in the sequence data. So, if you have read xyz in the sequence file right, the order in the qual file would be the same xyz, ok? So, if you just use a program, you can map this very easily back and

forth, you can read the sequence, and you can get the quality ore. So, this is how the quality score files will look, like the header line is here, like this is again in the first month of a month, and you can see the header lines for the same tools and read data,

right? So, this is the read id here, then you have the length, etcetera. The same information is there, exactly the same information.

But the difference is that now you see the numbers OK instead of this ATGC, you see the numbers right, for example, 21, 21, etcetera. So, each number is a quality score for a base in the sequence data. So, the first number is the quality score for the first base in the sequence data, the second number is for the second one, and so on. So, in this way, you have this quality score for all of them, okay?

And then again, you follow the first format. So, you have mentioned all the quality scores for one read, then you have the header line, and then you have the quality score for the next read, ok? So, this is how the whole thing is done, okay? So, this is just to highlight that this is a quality score of one base. Now, what are the drawbacks of the 454 data format? So, it looks great, right? You can now look at the sequence data, read the quality score, and check the quality score of the basis, etcetera.

You can write your own program, or you can probably use some software tools that will read the quality score as well as the sequence data and give you an idea of how the sequencing ns. So, this is important because the quality score is important, as I have shown

you in the flowchart, because the next step is the quality check, or quality control. So, for that, we need this quality score in there, okay? We cannot just look at the sequence data and say, "Okay, this is of great quality, right? We need to have the quality scores alongside it. So, what is the drawback? Right now, we have the quality score in 454 data points.

So, what is the drawback? Why do we not use this FASTA file and quality file separately and then have the data? So, one thing you probably would notice is that the quality file actually takes up more storage space, ok, more than twice the space of the FASTA sequence file. If you can, if you go back, this sequence file is ATGC; there is no space in between them, just one later at the time you are storing the data. Here in the quality score file, you have the numbers right. You have these two-digit numbers: 12 usually, the quality score is between 0 and 40, and maybe it is 12 or 15 most of the time. If it is less than 10, it is actually very low quality, and it is probably not good. So, most of the time, it should be above 20, 15, or 20 at least.

So, you have these two-digit numbers for every single base in the sequence, right? So, you are generating these 2 digits for every base that you have, and on top of that, you have the space between these numbers. So, that takes up a lot of storage space on the hard drive, right? So, if you are generating a lot of this sequencing data, you will probably run out of space very quickly if you have this kind of system right. You have this quality score, and these numbers are stored by spaces right, differentiated by spaces right.

So, this is something that is not very efficient, right? The second problem is that finding quality scores for individual bases is computationally time-consuming. Why? So, if you can, I can, you can think of right in this case, right you have these read 1 and you have the quality score, you can write a programme that will say ok read 1 find this read and look at the quality score of these bases. Now, this is ok for maybe the first few bases, or maybe the first 100, 200, or 1000 basis reads right. What about the reads that come at the end of this file? So, if there is a read that is like, let us say in the 10000 position, right? So, once you are reading this file in the other file, you will also first read all the 9999 reads, and only then will you reach that 10000.

So, it is a very inefficient process; it will take a lot of time because every time you are reading one read sequence, you have to read like the quality score you have to traverse through that file. So, this traversing through the quality file is actually very time-consuming, ok? And this is especially hard for the reads that appear in the middle or at the end of this sequence file. For the first few reads (1000, 2000, 5000 maybe you will not notice the time taken, ok, but as you go along, as you reach towards the end of this file, getting a quality score for every base in the read will take much longer than it used to take right for the first or second read in the file, ok.

So, this is something that is not very efficient, ok? So, here is an example, right? So, for example, you see here that this is the sequence file. Okay, this is the first format where you have the read data, and probably you have reached about 99 percent. So, here is the thing towards the end of that file, right? So, here it is, actually, in the Unix terminal, right? So, it shows you the percentage of files that you have already covered.

So, it is that we have covered only 99 percent of the file, where this is the line number that we are at right, and this is about 300,000, right 335,000, ok. So, this is the sequence; imagine we are reading here somewhere, ok? In the other quality file, we also have to read these 335,000 lines before we reach the quality score value. So, you can imagine that this is going to take a lot of time, ok?

So, how can we address these drawbacks? So, this is something that we saw in 454: that we have this quality score and the sequence data separately, and this is really, really inefficient. So, how do we address these drawbacks? We cannot just throw away the quality; we need that for quality control in the next step. So, there could be some solutions, but the probably best solution is: why not put the quality score along with the sequence data in the same file? And if you can find some way to put the quality score along with the sequence data very close by, that will serve a purpose, ok? Because then you do not have to read two different files; you do not have to traverse through all those lines.

So, the moment you read the sequence data, you are also reading the quality file ok? So, which means you require a new data format, right? It is not going to happen with fasta format that you are going to need a new data format, and that is something that is required. So, what you need is a data format where the quality scores are stored in a more efficient manner. So, one is that you are storing these digits, right? So, two digits for every base, then you have the spaces between these numbers; without spaces, you cannot distinguish which quality score is for which base.

So, again, we probably can do better, right? We can have more efficiency there. So, if you can do that right, this will require less storage space on the computer or on the server. So, this is something that is probably desired, right? So, we need a data format where the quality scores can be stored in a more efficient manner and also where the data quality scores are stored alongside the sequence data. So, these two requirements, if we can fulfil them, will be of great help to us.

And it turns out that there is this data format that has been defined now where you can do this. So, it is actually called first queue format ok? The name comes from the first format with quality scores. So, you have this queue, which means quality. It still resembles the first format to some extent, but it also contains the quality score in the same file. In addition, you have this sequence and the quality score in one place in the file.

In addition, you have the quality score encoding right. So, thus, these quality scores are not represented by numbers anymore, like inefficient two-digit numbers with no spaces. So, you have a more efficient way of doing this through some sort of encoding, ok? So, just to give you an example, So, this fast format is used by the Illumina platform, right?

So, all Illumina data comes in fastq format, okay? And just to give you an example, this is the FastQ data format, right? So, you will see that this looks fast, but not exactly fasta, okay? So, one of the things you probably notice again is that you have these header lines. So, this is your header line. What is this information that you have in this header line? We will discuss it in the next class. There is a lot of information here, and what you will

probably notice is that this header line starts with this on the right side.

So, this is a difference with the fasta file. Then in the next line, you have the actual DNA sequence. Right here, you see CA, CC, TTA, etcetera. This is the actual DNA sequence. Then you have a plus sign. Okay, this is actually a spacing right between the sequence and the next line. So, what is the next line? So, this next line actually contains the quality score OK, and this is actually an encoded quality score OK. You do not see the numbers anymore; it is not 21 or 25 or 35, right? You see some letters and some symbols, right? So, there is an encoding that is used, and this makes it much more efficient in terms of storage space.

You do not have any more space. So, it takes exactly the same amount of space as the sequence itself. So, this is something that we will see, and we will discuss again how this encoding is done in the next class, plus the spacing between the sequence data and the quality score. So, this is something you probably now see: this is a much more efficient way of storing this next-generation sequencing data, right? You have the sequence along with the quality score right in the same place.

So, when you are reading this file, for every read with this identifier, So, here you have this identifier, right? For every read, you have the sequence data along with the quality score right one after another, and if you are using a program or if you are writing a program yourself, you can just go through this very quickly. Also, the data is stored in a very efficient manner because it is not taking up too much space. And this kind of repetition is right: you have this one, then you have this next read. Here is the next read, and for each read, you have the sequence information and the quality score. So, what you will see is that

most of these files come in this sequence-dot FASTQ format. So, the name will be some sequence name, some name dot FASTQ, and most of the time they are also stored with this dot GZ compression.

So, this is compression in UNIX, and this also again reduces the file size. So, the final file name would be sequence dot FASTQ dot CZ, right? So, this will be useful when we actually do the hands-on tutorial, ok? So, these are the references that we have used in this class, and to conclude, Sanger sequencing data is stored in FASTA format. So, this is something we have seen and sequenced data from the 454 platform; they come in FASTA files, and quality scores are provided in qual files. And separate sequence and quality files are inefficient for computation processing. It is probably discussed why this is so.

And this drawback is addressed by the FASTQ format, which combines sequence data and quality scores in a single file. In the next class, we will be talking about the FASTQ format in much more detail about what kind of information it stores, and we will be looking at actual sequencing data. Thank you.