Next Generation Sequencing Technologies: Data Analysis and Applications

Calling SNP, SNV and Indels

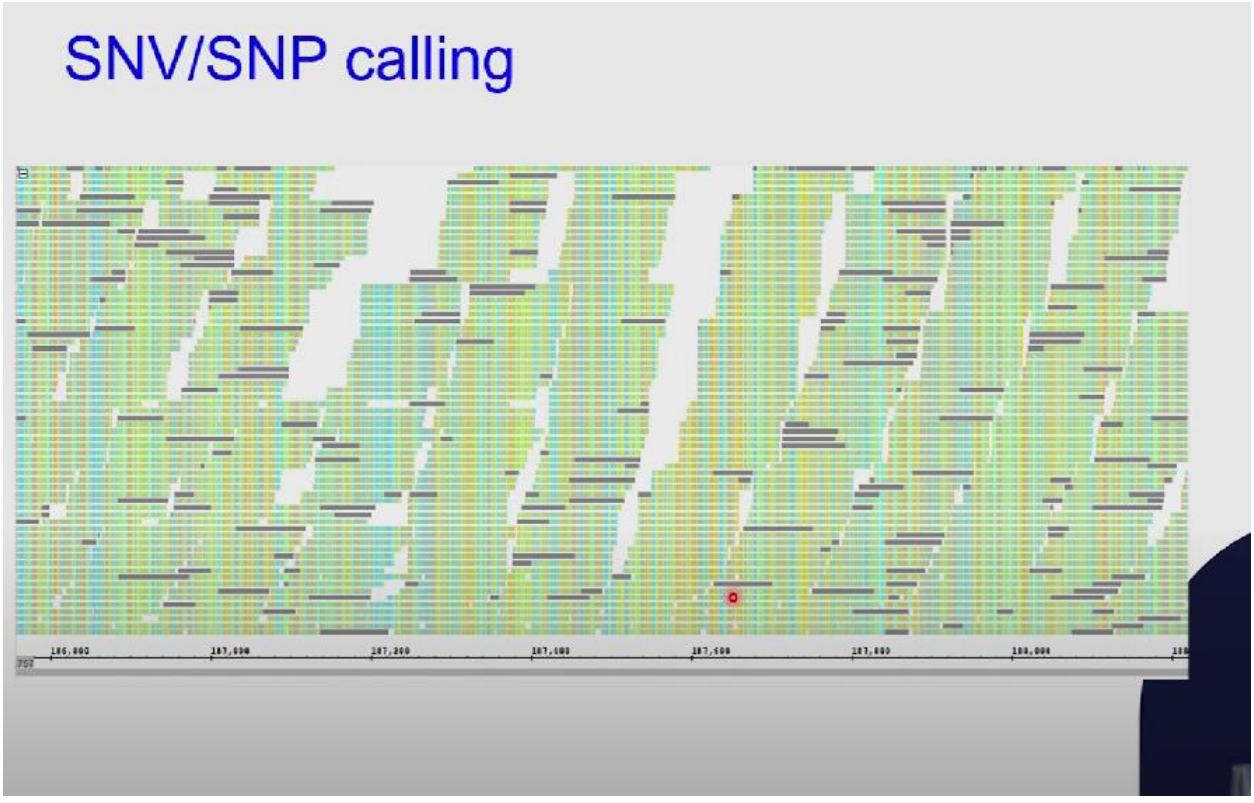
Dr. Riddhiman Dhar, **Department of Biotechnology**

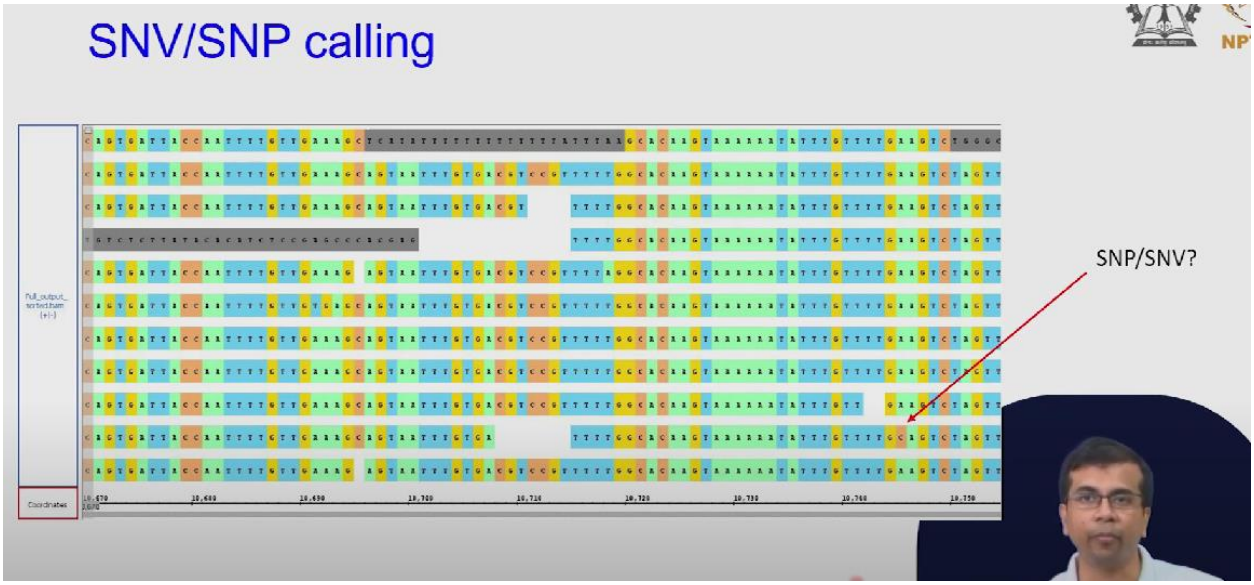
Indian Institute of Technology, Kharagpur

Good day, everyone. Welcome to the course on Next Generation Sequencing Technologies, Data Analysis, and Applications. In the last class, we have introduced different variants, especially SNPs and SNVs. And in this class, we will talk about how we call these SNPs and SNVs and the small Indels. So, we will talk about the steps, and later on, we will actually do these steps one by one. And so you will learn how to do them yourselves with real data sets, ok? So, this is what we will be covering today. So, we will talk. We will be talking about SNVs, SNPs, and Intels and how to call them, ok, from the map data that we have already generated. These are the keywords that will come across SNPs, SNVs, Intels, and Qualities. The quality score is actually important when you are describing or identifying these SNVs and SNPs.

So, now we can talk about this SNV-SNP call, right. So, we will start with this image of the SAM file, right? So, this is something we generated when we were doing the mapping, and we discussed what these fields mean; hopefully you remember, right? And one of the parts that is very important is this MDZ part, right? So, the MDZ part gives you what the mismatches are that are present in the read compared to the reference sequence, ok? So, I have highlighted three reads, for which you see these mismatches, right? So, you have this MDZ 12A something here, and you also have this

other MDZ 40A 84, right? So, you can see these mismatches, right? So, now the question is: do these mismatches mean there are SNPs or SNVs present? So, this is something we will now tackle, right? We will talk about how we actually go to the variants and identify those SNPs starting from this SAM file, ok? So, are these SNPs or SNPs? So, keep that question in mind, and we will see that this will be answered at the end of this class, ok? So, what are the steps for SNPs and recall? So, here is an image of the reference genome against which we have aligned these reads, ok? So, we have done this using an integrated genome browser, which we will again use later on in the hands-on. o, for each sequence, you can see this, right? Each fragment is a read that has been mapped against the reference genome and the reference genome coordinates are given at the bottom; you can see them, right? And you have these reads; certain regions are in gray because they are not aligned; this is like the soft clip regions that you see here, ok?





Now, we can zoom in a little bit, right, and we can see the matches or the mismatches that are present in this read compared to the reference sequence. I have highlighted one such mismatch here. You can see most of the reads show A in this position, but this read shows C, right, and the question is whether this is SNP or SNV, right? So, this is a very nice way of visualizing the mapping results, and then you can probably ask whether this is SNP or SNV, right? So, the answer is no, this is not because there are a lot of factors that can impact this calling process, ok, and we will talk about these factors that can affect, and then we will talk about how we can overcome these issues and then actually identify the real SNP or SNV, right, and not the sequencing artifacts or the errors that are present, ok. So, here is a short summary. Right, what we have is that we have the reference genome and we can have short reads and long reads. Right, back to the genome and certain positions, you will see these red marks. Right, they are showing mismatches. We will not call them SNV or SNP yet. Now, some of these mismatches would be sequencing errors, right? The other mismatches could come from duplicates, right? So, we can falsely call SNV or SNP if this is present in PCR duplicates. For example, if both copies of the duplicate show the same variant, we might falsely call the duplicates SNV or SNP. So, these are some things that we have to keep in mind when we are doing this analysis. So, we will now formalize these steps, ok?

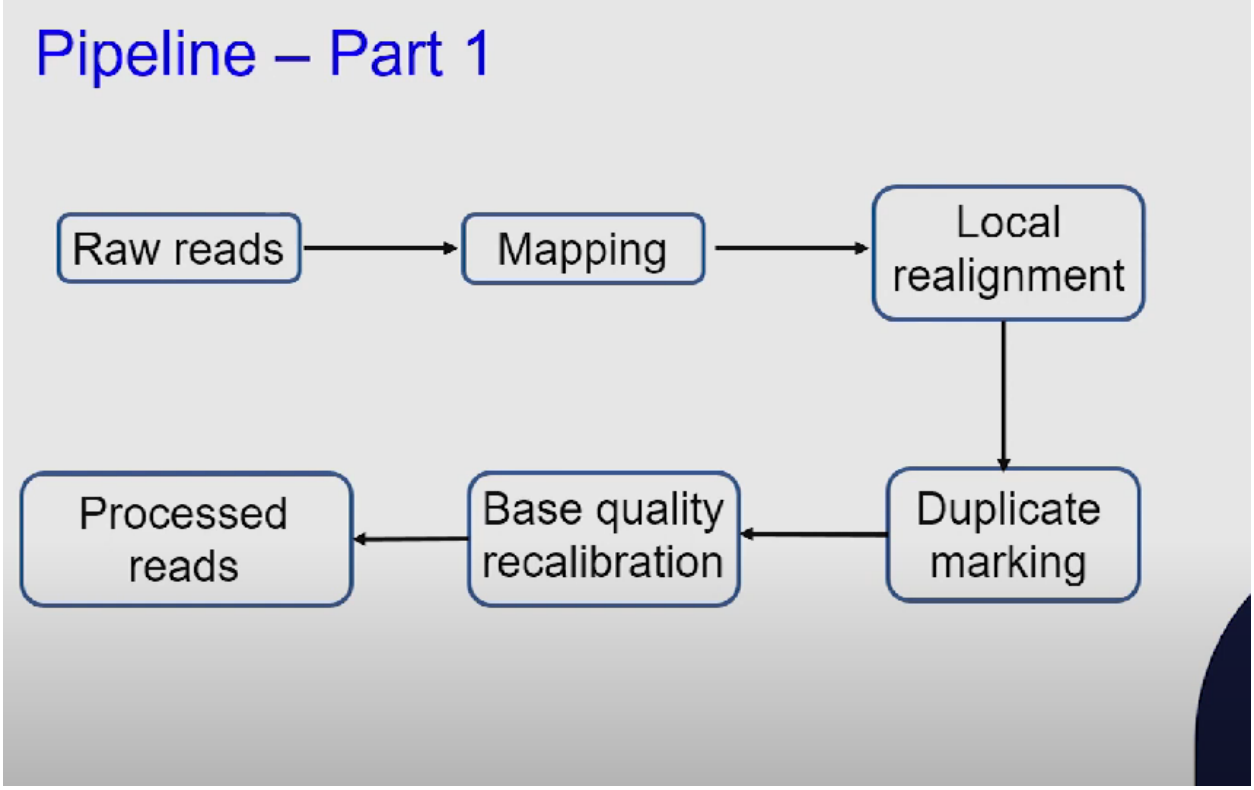
So, what are the challenges? So, as I mentioned, you can have sequencing errors, which are present in all sorts of data. Even if you think about the Illumina platform, even with very high accuracy, even if you have 99.

9 percent accuracy, you have 0.1 percent error, and that becomes quite a large number if you consider we are sequencing, let us say 30 billion or 50 billion bases, right? So, that 0.1 percent still becomes a really large number, which means we have a lot of sequencing errors present in the data, and this is something we would have to be careful about when you are analyzing this thing. The second problem we have talked about is the duplicate reads I just mentioned. So, if you have duplicate reads and they show certain variants, you will see multiple copies of that variant in the data, and you might think, OK, this is present in multiple reads. So, this must be a true variant, ok, and this is something we again would have to deal with when we analyze the data. So, as I have just mentioned, right, imagine these duplicates. These are the two duplicate ones here you can see I have highlighted, and both these duplicates show this mismatch, ok. This shown by red color here red box. So, you might think, OK, this is probably a SNP or SNV, right? So, this is something that again, is not the case, right? This is merely because we have duplicate reads, and they are showing that both of them show certain errors in the data that lead to this mismatch. Then you also have something called erroneous alignment. So, when you are doing this mapping, the mapping tool is not perfect, and it sometimes makes mistakes. If the alignment is disturbed when you have indels present or if you are in a low-complexity region, So, we have talked about what these low-

complexity regions are. These are regions that contain only certain bases that may be repeated many times and do not contain a complex combination of these different bases, ok? So, the alignment suffers, and as a result, you have these false calls, and again, this is something we need to be careful about when you are analyzing this variance. So, as I have just mentioned, low-complexity regions these are the regions in the genome that lack complex patterns of basis and are simple base repeats. So, you can imagine, for example, the homopolymeric regions, right? So, we have one single base repeated multiple times, and that can cause problems for the alignment, and that can also lead to several false calls, ok? And then finally, we have coverage non-uniformity. So, what do we mean by coverage non-uniformity? It is that not all regions of the genome have the same coverage. There is a lot of bias, and this bias comes from, for example, the amplification bias, where certain sequences are amplified much more efficiently during the amplification step compared to other sequences. So, you probably remember these amplification steps that are required for the preparation of sequencing. For example, in Illumina, we have bridge amplification, and in 454 we also have emulsion PCR. So, many of these methods require this amplification, which in turn can lead to biased representation of certain regions of the genome in the library, which can lead to this coverage non-uniformity.

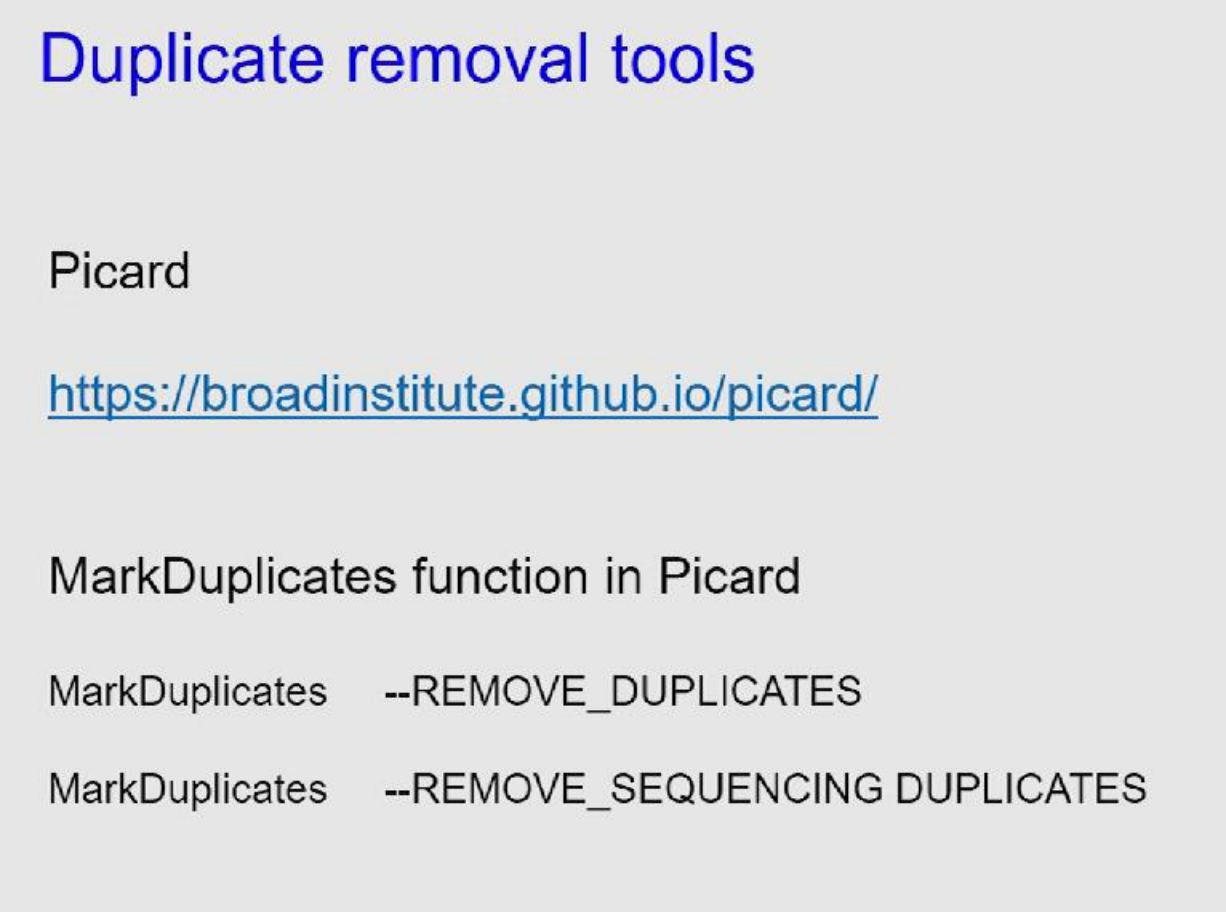
So, what it means is that if you have non-uniformity in coverage, you might not be able to call variance in certain regions of the genome because you do not have enough coverage, or if you see differences in coverage between the reference genome and the target genome, you might not be able to call variance properly. So, this is something that you would have to keep in mind when we are facing the data. So, the general approach we will talk about, and then we will go into the specific steps that we can follow that can help us reduce these errors in variant calling, So, the approach is that we identify variances that are present in multiple reads because these are high-confidence variances that are present in multiple reads. And again, this variance should appear in reads in both the forward and reverse strands. So, there will be reads that map to the forward strand and reads that map to the reverse strand, and these variants should appear in both types of reads, right? There should not be any strand-specific bias, ok? This is very important, and the variant should show a quality score above a certain threshold, right? We can set this threshold. So, for example, you can choose Q20 or Q30 for a score of 20 or 30.

If you want to be strict, we put 30; otherwise, we put 20, right? Again, this can be varied, and it is up to you how stringent you want to be. So, let us now talk about this pipeline. So, this is part 1. There are multiple parts, and the first part, of course, that we have already talked about is that we take raw reads and do the mapping. So, once we have generated the mapping, we do something called local realignment. This is again very important, as we have seen, or as we have just discussed, because in certain regions where you have indels present or these regions are low-complexity regions, unless you do local realignment using other tools, you will have false variant calls, and we want to minimize that. So, we will do something called local realignment. You will see in our hands that we will actually do the realignment. Then the next step is the duplicate marking, ok? As we just discussed, duplicates can make this analysis quite complex, right? We can lead to false positive calls, right? So, we need to actually identify duplicates and be careful when we are analyzing the data, ok? We need to do the local realignment before duplicate marking because the duplicate marking process relies on the positions of the reads, right? So, it will identify reads as duplicates, which actually start at the same position and end in the same position, right? So, the position information is very important for duplicate marking.

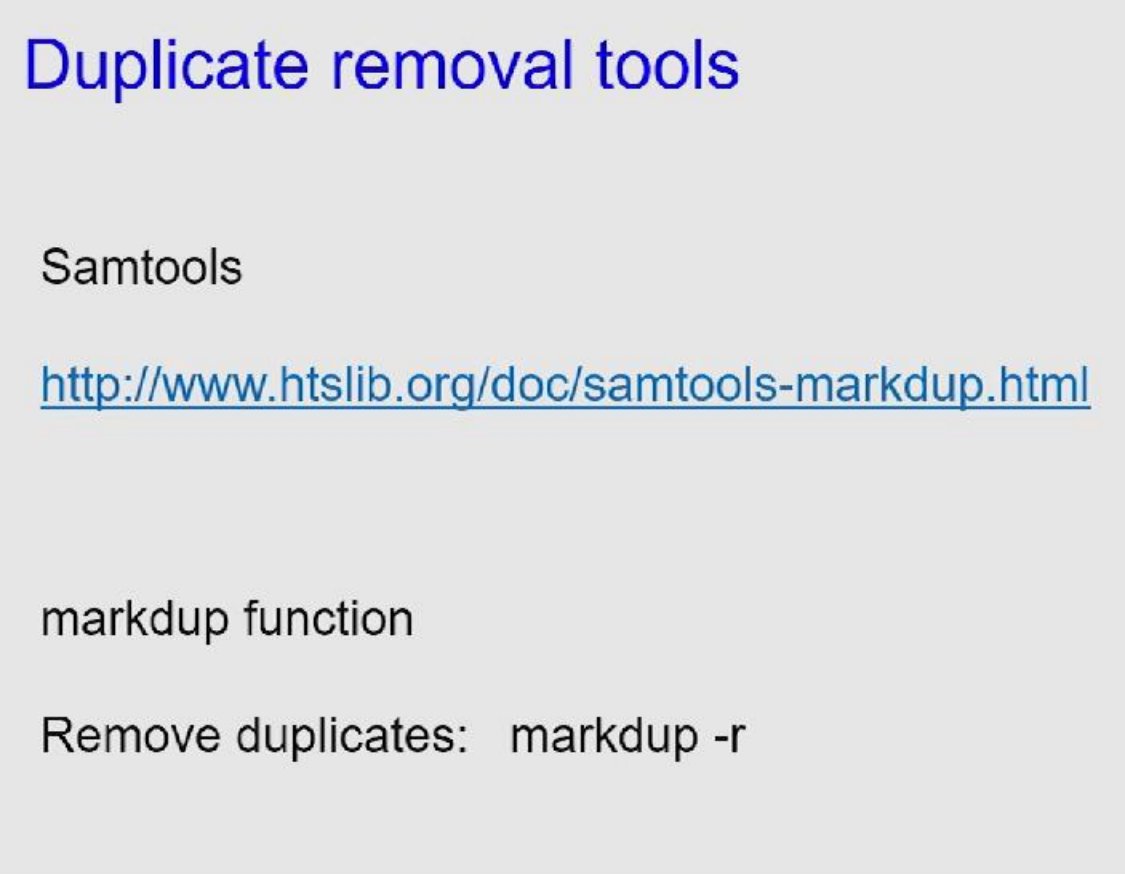


Then we have something called base quality recalibration which we will talk about in a moment, and then finally, you get process rate. So, if you want a more detailed description of this idea and this pipeline, you can read the reference that I have just linked, which is present at the end. So, as I mentioned, local realignment helps correct misalignments near indels and low-complexity regions. So, that is why we want to make sure that the alignment that we have is correct. So, we use certain other tools to do this realignment, ok? Not the mapping program that we use to do the realignment, but we have other tools to do this realignment process. So, in the realignment, what we do is just take the reads that are present in certain positions but present in around, say, a certain variant, and then we just realign them properly, ok. So, what we have seen is that it reduces false positive variant calls for both SNPs and indles, ok. So, there are some examples, and you can read some references where you see this kind of analysis. The next step we have mentioned is the removal of duplicates or duplicate marking.

So, as I have already mentioned and as you have seen in the schematic, one figure that I have shown you is that it reduces erroneous base calls due to duplicates and also helps in improving the variant frequency calculation. So, for example, if you have a duplicate that is showing a true variant of course, but it is repeated multiple times, So, you are calculating the frequency of that variant in a population. This presence of this duplicate reads what you will do; it will change the frequency calculation, right? So, we need to count these duplicates as just one read, ok? Otherwise, we might overestimate the frequency of a variant in the population. So, there are different tools we have mentioned before. So, I just mentioned it again because we will actually use this for the hands-on.



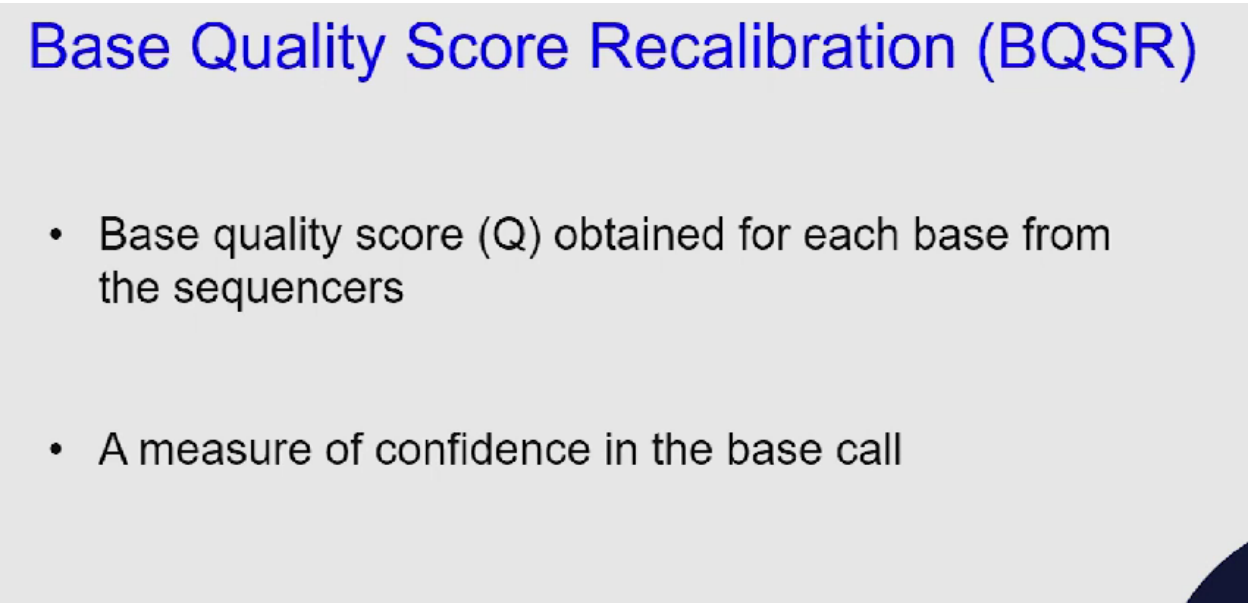
So, one is Picard, and they have this mark duplicates function and now you are familiar with these functions on the command line in Linux. So, we have marked duplicates; we can use minus minus to remove duplicates, or we can also use minus minus to remove sequencing duplicates.

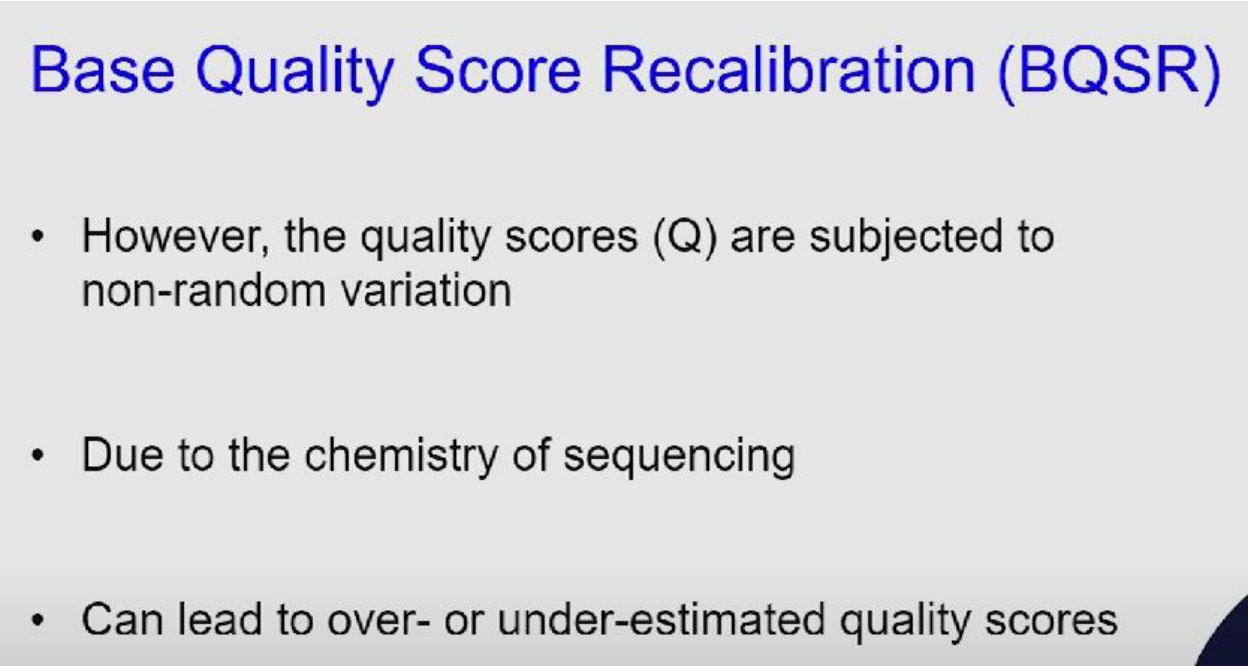


We also have SAM tools. This is the tool that we will use for our hands-on, and it has a mark duplicates function. If we have to remove duplicates, we can use mark duplicates minus R. So, coming to the base quality score recalibration, ok. So, what happens is that the base quality score of each base is recalibrated, ok? So, you might ask what the need is for such recalibration, ok? So, as you probably remember, the base quality score, or Q score, is obtained for each base from the sequencer.

So, it measures the confidence, right, in the base call. It is generated from the signal-to-noise ratio, ok. But what happens is that these Q scores are subjected to non-random variation, ok because these are generated by a specific sequencing process or the chemistry, right? There are certain biases that are present, and these can lead to non-random variation, which means we are overestimating or underestimating the quality of the basis. So, that is why we tend to do this base

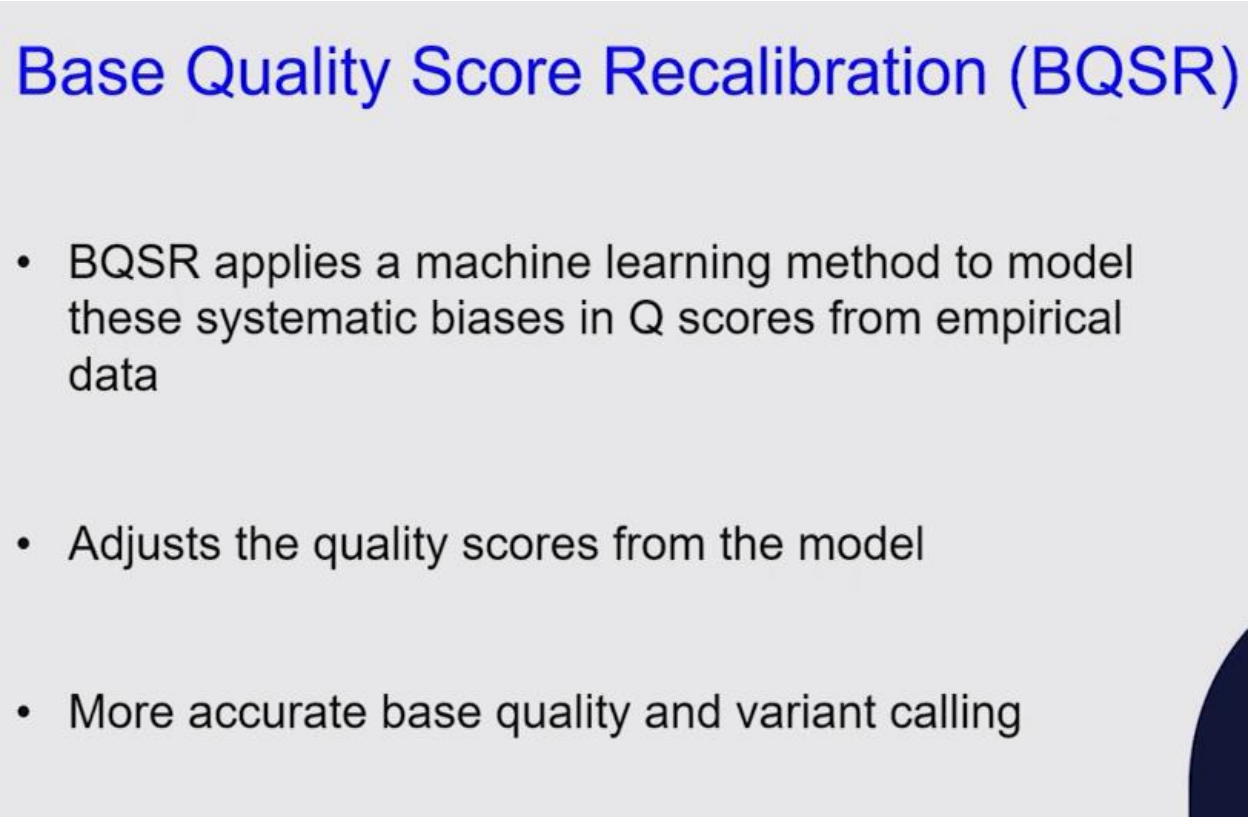
quality score recalibration that we will see in certain pipelines.

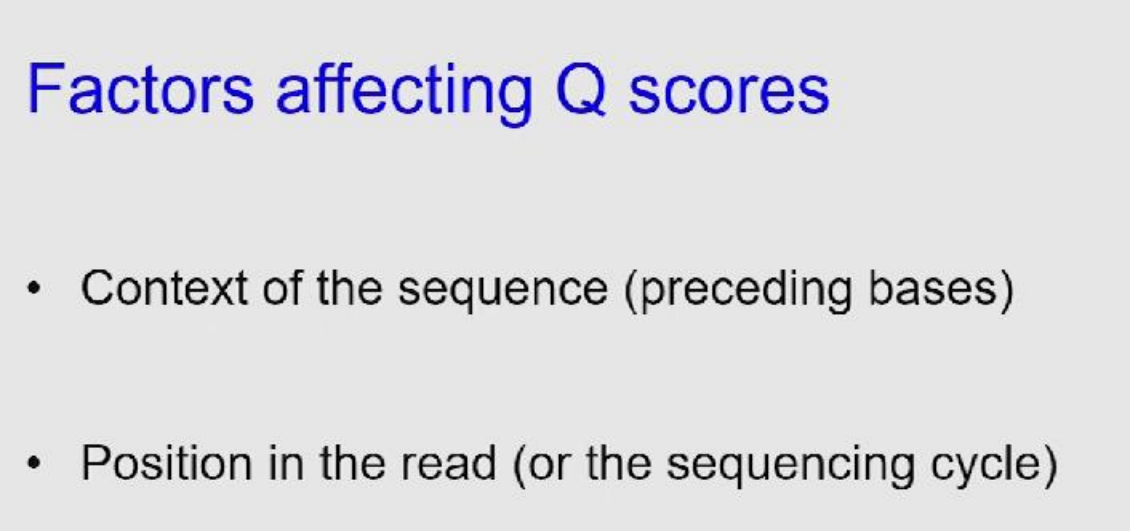




So, what BQSR does is apply a machine learning model to actually identify or model these systematic biases in Q scores from experimental data. So, it can identify the systematic biases that are arising from different sequencing platforms, and then it applies this model to recalibrate or adjust the quality scores of the basis, ok. So, one thing we need to keep in mind is that when it is adjusting the quality score, it will not change the base, ok. So, for example, if you have an A base and its quality score was 20, it might readjust the quality score to let us say 18, but it will not say,

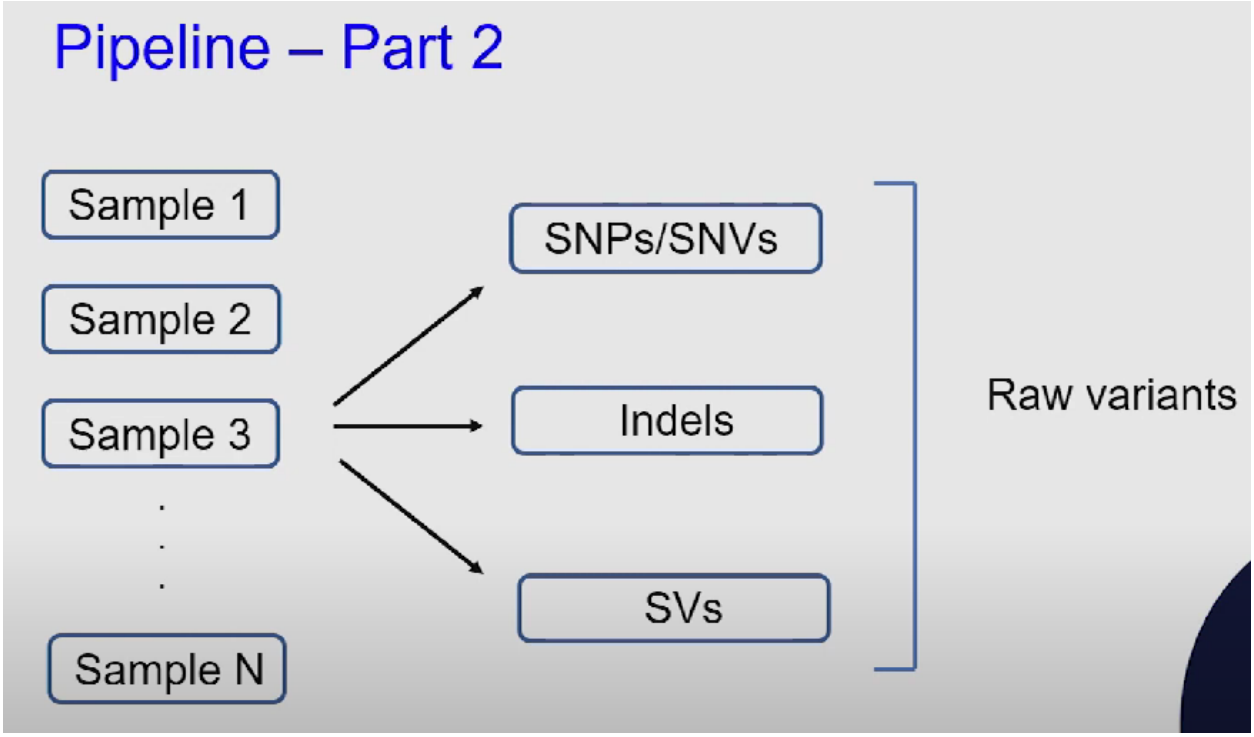
OK, A is very low quality, but it was actually T in the original sequence; it cannot do that, ok. It is not designed for that, and what happens is that if you do this recalibration, it actually leads to more accurate base quality and more accurate variant calling.



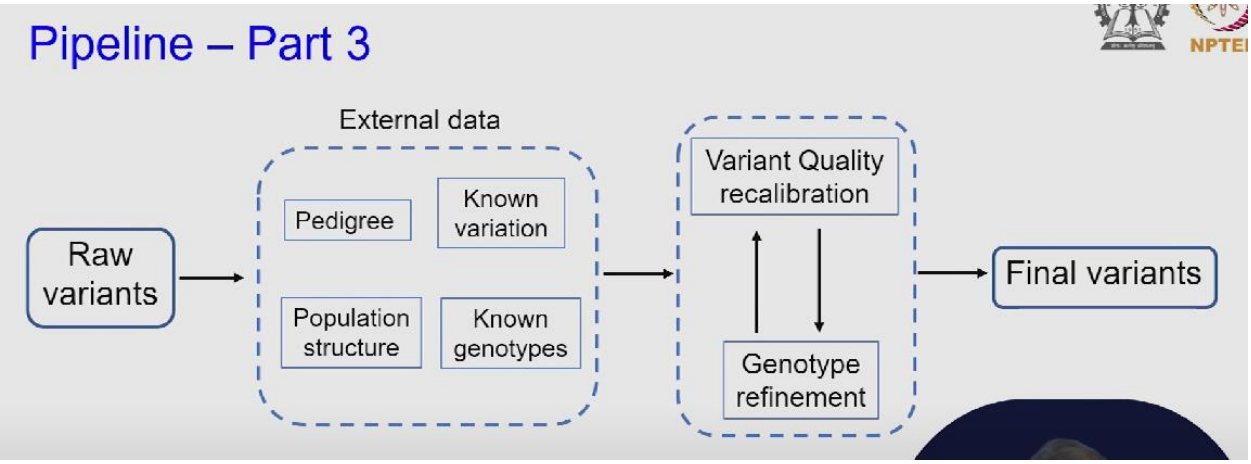


Again, the goal of all these processes is to actually do a more accurate analysis and not identify false positive variants. So, what are the factors that affect Q scores, mostly those that have been

studied? So, one is the context of the sequence, for example, the predicting bases, right? What are the bases that have been observed beforehand and also position in the read or the sequencing cycle? So, this is something we also learned earlier, right? For example, in Illumina, the quality score starts to drop towards the end. There are different reasons we have talked about, ok? So, the position in the read is actually a very important factor, and this is not surprising. So, here is a reference in which you can read about all these methods and the pipelines, the different roles of these different steps, etcetera, in much more detail. Part 2 of the pipeline, ok.

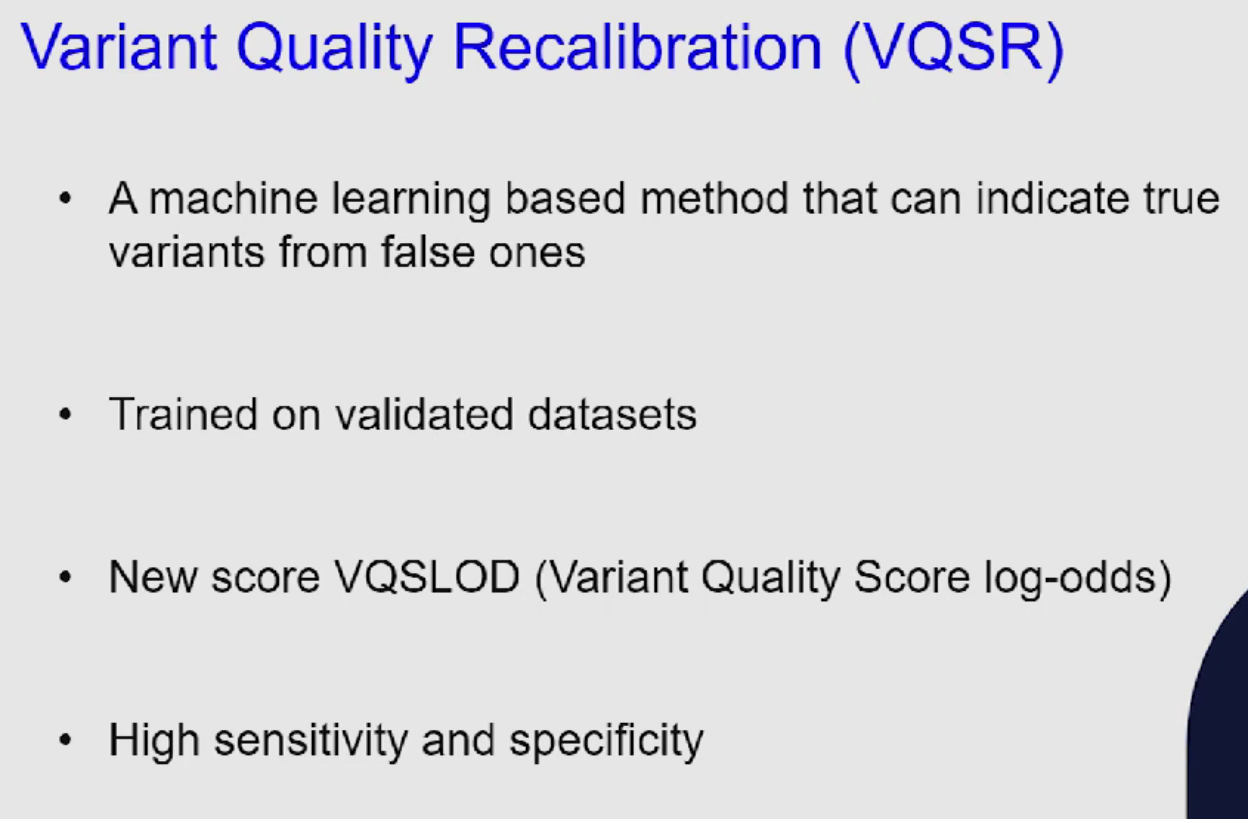


So, we now have multiple samples if you are doing a comparative analysis, and we are trying to identify these SNPs, or small individuals, right? We can do that now once we have these processed reads, right? So, from that, we can now identify these different types of variants, and these are the raw variants, ok? Now, we can go further for certain datasets, where we can kind of check whether these variants that we are seeing are expected or whether they have been found earlier in other datasets. So, if some of the variants have been found earlier, they are likely to be true variants. So, this is what we can do in the next step. Again, I am linking to the reference, which actually discusses this in much more detail.

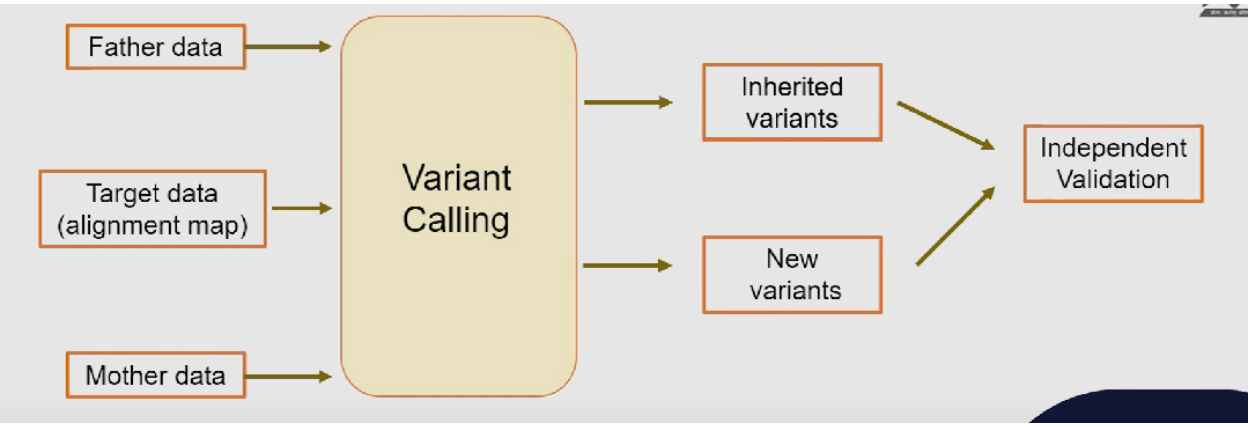


So, we have raw variants, and we can take some external data where we have an idea of this known variation or known general types, and compare against this external dataset, ok? And we do something called variant quality recalibration, which is also associated with genotype refinement. In essence, we are kind of identifying which of these variants we have identified are already present in those sets of results, right? So, these are high-confidence variants, and finally, we get this final variant list, ok? So, a brief word about variant quality recalibration, or VQSR.

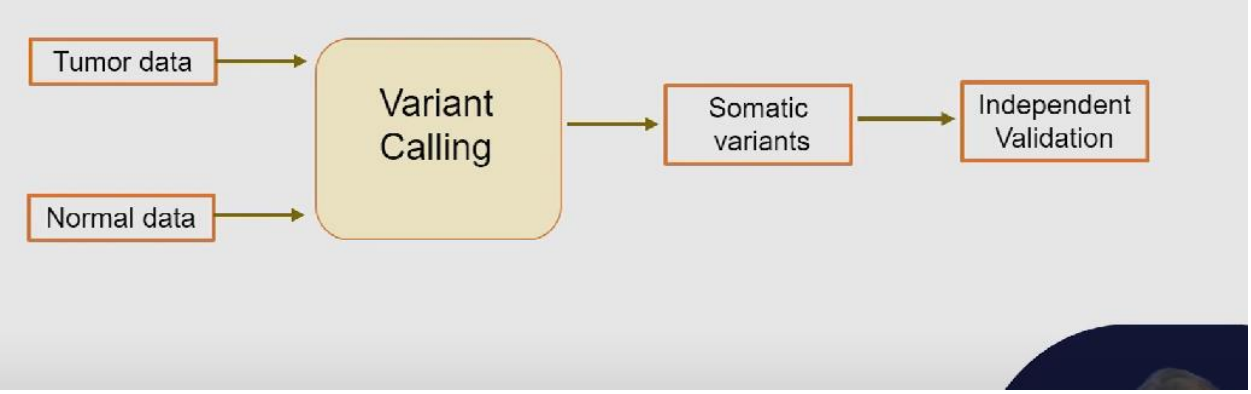
So, what it does is again apply a machine learning-based method that can actually identify the true variants and separate them from the false ones. Do not confuse this with VQSR, which is the base quality score; this is a different recalibration, ok? So, what happens is that this model is trained on validated datasets, and it can identify certain features of those variants again without going into too much detail. It can identify certain features of these variants in these validated datasets, and from those features, it can define a new score, something called VQSLOD. It is a variant quality score log odds, and it can generate this variant list or identify variants with very high sensitivity and specificity, ok.



So, once we have this training done on validated datasets, we can then apply it to the target genome or target dataset to identify these variants, which will give us this VQSLOD score, ok. Now, we will talk about some of the pipelines for different types of variants that you can identify. So, one is the germline variants; this is actually very important, right? So, what we do in the germline variant is take the sample data or target data. Right now, we have one individual data, which is usually the alignment map SAM or BAM, and we also take similar data from the parents. So, we have further data and mother data, and we do the variant calling following all the guidelines and all the steps that we have just discussed.



So, for all these three samples, we do the variant calling, and we can identify these variants that are common across these individuals, right? So, some of the variants that are common are the inherited variants, and we can also identify some of the new variants, right, that are present in our sample data or the individual data but not present in the parents, ok? And of course, once we have identified them, we want to validate them through some independent validation methods just to be sure that these are actually true variants, and again, this kind of information is very important for revising the pipeline. So, once we have this validation done and we see, ok, we are identifying mostly true positive ones, then we can be confident about our variant calling pipeline, right? So, this is again a trial-and-error process, right? So, these independent validation experiments actually help in validating the pipeline that we use, ok? So, we will also talk about how we call the somatic variants in tumor cells. So, as we have mentioned, the somatic variants are present in tumor cells, and again, we can identify, for example, the driver mutations by studying these somatic variants. So, what we do here is take tumor data and normal data, and again, following the steps that we discussed, we can do the variant calling for each of these datasets, starting from the alignment map, and we can specifically identify the somatic variants, right in the tumor data, which are present in tumor cells and are not present in normal, right? So, this will be the somatic variants; this is a comparative analysis, and again, we have some sort of independent validation experiments for this pipeline, ok?



Again, we can have other types of ideas and other types of steps that we can introduce to make it more robust, but this is kind of the general approach that we can take, ok? So, what we will talk about now is something called variant filtering approaches. So, there have been a lot of these variant filtering approaches that have been developed again. The goal is to identify the true variants

and reduce the number of false positives. So, when I say false positive, these are not actual variants that are present in the sample, but they are called because of the errors in the data or artifacts in the data, right? They are kind of generating these false results, and we want to minimize the number of false positives.

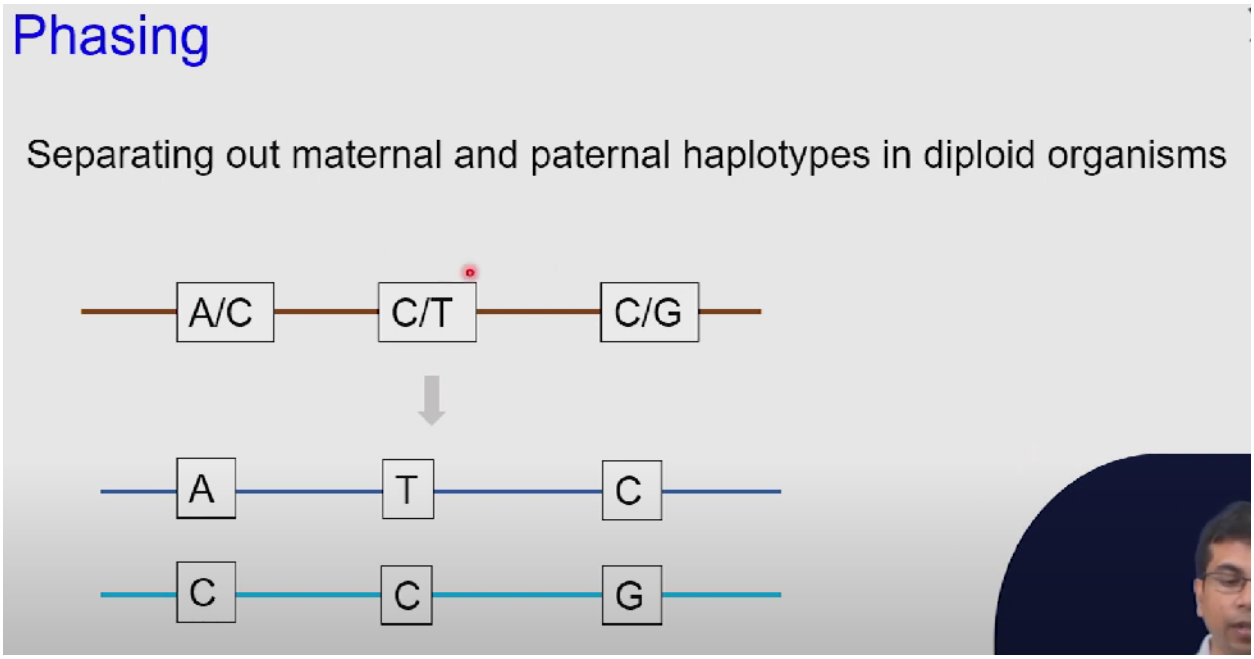
So, we will talk about some of the variant filtering approaches that have been discussed in the literature. So, one is called the low-complexity filter. So, what happens here is that we filter out variants that are present in low-complexity regions, ok? So, this is kind of being conservative, as we have discussed low-complexity regions and alignment issues. So, which means we can get a lot of erroneous variants called SNPs as well as small indels. So, maybe it is better sometimes to just filter out variants in low-complexity regions, ok? We do not even consider variants that are present in these regions, ok? Something that we discussed and that you can probably relate to is the variants present in the homopolymeric stretches in the 454 data, right, and also similarly in the ion torrent data, right. So, we have a lot of errors right in those regions, and perhaps we need to be very careful, or maybe we can filter out variants in those regions altogether. The second type of filter that you can use is something called an allele valence filter.

So, here what happens is that we filter sites where the number of target reads is too low, ok? So, this makes sense, right? If your coverage is actually quite low for a region where there is a variant call, then probably you have low confidence in that call, right? So, because this call is not supported by many readers, So, this is something you need to be careful about again, and again, you need to set what is actually too low, right? What is the number? That means your coverage is too low, or the reads number is too low. So, usually, it is set to at least 3 or 5. You need to have at least 5 coverages, and a certain number of reads should show this variant. The third one is the double-strand filter, right? So, what it means is that we filter out variants if the number of target reads on the forward strand or on the reverse strand is below a threshold, ok? So, again, what we have mentioned, right? So, in the general approach, when we are identifying a variant, we want this variant to be present in the reads that are mapping to the forward strand as well as in the reads that are mapping to the reverse strand, right? Now, if the number of reads mapping to the forward strand or reverse strand is too low, what it means may be that the variant is mostly called by reads mapping to just one of the strands, ok? And this is something that is not desirable; maybe there is

some sort of artifact, etcetera; there might be some sort of bias. So, which might actually lead to a false positive cause, ok? So, this is something we want to avoid. So, that is why we can deploy this double-strand filter. Coming to the next one, this has been called the Fisher strand filter, ok. So, what we do here is remove sites where the numbers of reference and target reads are highly correlated with the strands of the reads, ok. So, what it means is that, for example, the number of reads that you get in the reference is mostly on the positive strand, and the number of reads that you get in the target is mostly on the reverse strand. So, that kind of bias might be there, and this is something we want to avoid again, right? So, this is again, as we have mentioned, we want variance to be called from read mapping to post strands, ok? And we want to avoid any strand bias in the data. And finally, of course, we have quality filtering, right? So, we want to filter variance with a low quality score, right? So, if the base that is being called, especially in the case of SNP or SNV, the variant base that has been called by the sequencer, is of low quality, then it is it is probably or there is a high chance that this is a sequencing error, right? So, again, we want to be really sure that this is a true positive call, and for that reason, we will filter out variances that show a low quality score. Again, you can set this filter; you can decide on the threshold, whether it is 10, 20, 30, etcetera. This is something that is again up to you; if you want to be strict, you set it at 30 or so. Another thing you have to keep in mind, right? So, because we are calling this from multiple strands, you can probably set this threshold a bit low because if multiple strands are showing, multiple reads are showing the same variant, right? So, then, it is slightly likely that this is probably a true variant. Again, if not all reads or if not all the quality scores are low, there is something that you need to think about, especially with the quality score, ok? So, one of the things that we want to discuss again in the case of deployed genomes is that you can find heterozygous variants and homozygous variants, ok? So, heterozygous variants are variants that are present in only one copy of the chromosome, and then you have homozygous variants that are present in both copies of the chromosome. Again, this will affect your analysis, right? So, if you have a heterozygous variant, you might see that only 50 percent of the reads will have that variant, and this is something that you also have to keep in mind.

There is one final point that we will discuss: something called phasing. This is again related to deployed genomes, just like us. So, if you are working with human genome data and you are identifying variants, sometimes you would like to do this phasing, ok? We will see some of the

variant data, and you will see the phasing being done. So, what phasing means is that it means you have to separate out maternal and paternal haplotypes, ok? So, what are these haplotypes, right? As I just mentioned, in zygotes you have one copy of the chromosome from the mother and one copy of the chromosome from the father, but then you have to figure out, right when you are sequencing you will get all sorts of variants there and you want to separate out, ok, which variants come from which parent, ok, whether some of the variants come from the mother, and which of these variants come from the mother.



This is something that you have to separate out, right? So, when you are sequencing a deployed organism like a human, right, if you have to have this human genome data and for certain regions you have this situation, right? So, for a position, you see AC; right for this position, you have these two variants: AC; for the other one, you have CT; for the other one, you have CG; right for this variant, Now, the question is whether A occurs with C or T, right? So, the first A here is in the first position, whether it occurs with C in the second position or T in the second position, and whether it occurs with C or G, right? In other words what is the combination of these variants? Right in these two copies of the chromosome, ok.

And this is the problem, right? This is what you do when you are doing the phasing that you are separating out these combinations, right? So, what we see in one combination is ATC, and the other one is CCG, right? So, from this combination here, we are separating out these two

combinations. So, this process is called phasing, ok? So, here are the references that we have used for this class, and to summarize, we have seen that identification of SNVs and SNPs is a multi-step analysis. There are different processes that we go through, and the goal of each of these is to reduce the number of false positive calls and maximize the number of true positive calls. And here are some challenges that we talked about, right? So, we have sequencing errors, and we have low-complexity regions and quality scores, right? So, each of these factors will contribute to false positive calls. So, again, you have to be very careful and mindful of each of these factors when doing the variant calling analysis. And of course, we have talked about recalibration, realignment, and filtering approaches that actually counter these biases, and they can deliver high sensitivity and high specificity variance. Thank you.