CONSENSUS & VARIANT CALLING PRACTICAL

Introduction

In this session, you will learn about how to generate a consensus and look at minority variants within viral NGS datasets. For the latter, essentially, nucleotide frequency variation is determined across a region or regions of interest, and these variants are interpreted with downstream algorithms. Some tools are able to report linkage, also known as 'phasing', that is, to what degree mutations are found on the same reads (and implicitly viral genome molecules). This is clearly limited not only by depth of coverage, but also by the length of the sequenced library molecules. This is usually ~300-500bp for Illumina sequencing of viral genomes but can be much longer when looking at bacterial and eukaryotic genomes using e.g. mate-pair analysis techniques.

Variant calling starts with reads being mapped to a reference sequence, generating an alignment file, almost universally in SAM or BAM format (BAM files are simply SAM files that have been compressed to save disk space). More information about how alignment information is tabulated and stored in SAM format is found on the **samtools** github page:

https://samtools.github.io/hts-specs/SAMv1.pdf

The practical is based around looking at a variant/consensus calling tool and their outputs on two data files derived from the same FASTQ reads. Students should discover that the choice of reference for mapping is of critical importance, together with the limitations of minority variant calling when confidence in the data is low.

Command prompt

Throughout these practical notes, lines to be typed into the Unix shell are prefixed with `GCV2025:~$` in green. In the practical itself, '**GCV2025**' will be extended by text specific to the training shell, possibly including the current directory. This will not affect the running of the practical.

Results files

If time is running low, and completing all the analysis is looking unlikely, then there are pre-made results files available. In the practical directory, there is a zipped directory called **results.zip**. Within these are outputs generated by running the commands on the data for that section. You are encourage to run the commands as much as possible however!

To access the pre-run data, whilst in the directory containing the **zip** file, run the following command:

```
GCV2025:~$unzip results.zip
```

All the outputs of the tools used in this practical will be extracted into the directory

Variant calling with QuasiBAM

For the first exercise, we will take a pair of SAM file-reference FASTA pairs and use **QuasiBAM** to explore differences in outputs caused by different references. The SAM files were prepared by firstly generating FASTQs from an amplicon covering part of the HIV pol domain important for drug susceptibility testing, running **trimmomatic** to clean the low-quality ends of reads, and then mapping them to two references using **bwa mem**:

1 HXB2
A subtype B virus dating from 1983. For three decades or so, co-ordinates of nucleotides and coding domains, roots of phylogenetic trees, and many laboratory experiments have all used HXB2 as the reference strain and it continues to serve this function.

2 de_novo_contig
This has been generated by *de novo* assembly of the trimmed FASTQs after a **normalise** step (see *de novo* practical for details of this process)

The following tools will be used:

1 **samtools**
A widely-used package of Linux tools for manipulating SAM and BAM files[1].
Li H, *et al*. Bioinformatics (2009) 25(16):2078-9

2 **QuasiBAM**
A **C++** script, written by Richard Myers at PHE for the express purpose of analysing variant frequencies in viral genomes.
Penedos A, *et al*. PLoS One (2015) 10(11): e0143081

3 **qb_reanalyse**
A **perl** script (also written by Richard Myers) that parses the **QuasiBAM** nucleotide frequency table output, producing a new FASTA based upon user-defined depth, mixture and reporting threshold parameters[2]. It saves re-running **QuasiBAM**.

[1] See http://www.htslib.org/doc/samtools.html for an online manual outlining all the options.
[2] On the training shell, an '*alias*' has been written into the `.bashrc` file (located in the root directory) such that whenever `qb_reanalyse` is seen in a command line, the shell replaces it with `perl /home/training/Programs/qb_reanalyse.pl`, the long-hand command and script location.

A: Using `samtools` to convert SAM to BAM

The practical starts in **`/DFB/variant_calling`**, which contains two SAM files:

`HXB2_pol.sam`
`contig_pol.sam`

two reference files:

`HXB2_ref.fas`
`contig_ref.fas`

and two FASTQ files:

`FASTQ_R1.fastq`
`FASTQ_R2.fastq`

The first step here is to convert each SAM file (large tabular text file that is readily viewable in e.g. *Excel* or *Calc* – try using **`less`** to look inside) into a BAM file (a SAM file that has been through binary compression to save space – not readily viewable, using **`less`** here is unhelpful). This is achieved using **`samtools view`** – a function within the versatile and extremely useful **`samtools`** package – that both converts and/or filters SAM and BAM files[3].

```
GCV2025:~$ samtools view -Sbh HXB2_pol.sam > HXB2_pol.bam
```

Use **`ll`** to see how the BAM files are much smaller than the SAM file. Try reversing the process to see how all the information is retained (note the absence of **`Sb`** input/output flags in this instance):

```
GCV2025:~$ samtools view -h HXB2_pol.bam > HXB2_pol_clone.sam
```

We can use **`diff`** to compare the two files. However, there will be no output at all, as both files are exactly the same!

```
GCV2025:~$ diff HXB2_pol.sam HXB2_pol_clone.sam
```

Hence, to store a mapping result, BAM files are much more space-efficient than SAMs. Even the unmapped read data is retained, so the FASTQ files used to generate the SAM file can be discarded!

This can be demonstrated by using another of **`samtools`**' functions – **`fastq`**. This tool generates FASTQ files from BAM files:

```
GCV2025:~$ samtools fastq -1 test1.fq -2 test2.fq HXB2_pol.bam
```

Try **`diff`** again to look for differences[4]:

```
GCV2025:~$ diff -s FASTQ_R1.fq test1.fq
```

[3] The set of flags applied here are **`-S`**, **`-b`**, and **`-h`** (collated into **`-Sbh`** for convenience):
`-S` declares the *input* to be in SAM format.
`-b` dictates that the *output* should be in BAM format – it can be omitted if the output file has the **`.bam`** suffix, i.e. **`samtools`** detects what is needed.
`-h` flag tells **`samtools view`** that the header information is to be retained, which is essential for many downstream BAM-using applications, including **QuasiBAM**.
[4] The **`-s`** flag tells **`diff`** to report when the two files are the same rather than outputting nothing.

B: Using QuasiBAM – nucleotide frequency tables

QuasiBAM takes as input a BAM file and the single reference sequence used to generate it[5]. In order to obtain in-frame amino acid variant information, the reference sequence can be appended with sub-sequences corresponding to coding regions. These subsequences must be exactly contained within the larger reference genome, i.e. not derived from an alternative source. All sequences must be provided in FASTA format.

Here, the subsequences of the protease and RT fragments have been provided – this can be seen using **`less HXB2_ref.fas`**.

Run **QuasiBAM** on the first BAM-FASTA pair:

```
GCV2025:~$ quasi_bam HXB2_pol.bam HXB2_ref.fas
```

There are 3 output files per **QuasiBAM** run:

`<name>.txt` A nucleotide frequency table with one row per nucleotide position, and columns including depth, **`A`**/**`C`**/**`G`**/**`T`** frequencies, gaps, inserts and more. This can be visualised using **`less`**[6], but may be easier to look at in a spreadsheet application like *Excel* or *Calc*.

```
GCV2025:~$ less -S HXB2_pol.txt
```

`<name.fas>` A FASTA file generated from the nucleotide frequency table containing a sequence for each subsequence (if present) or the reference only (if not). There are two key parameters dictating how these are derived:

1. Depth
   Only positions where the depth of coverage exceeds a specified value (default = 100) are reported. Positions with lower depths are reported as Ns.

2. Consensus frequency
   At each position, the output IUPAC-IUB nucleotide code[7] reflects all bases whose frequency exceeds a specified percentage (default = 20). This is the threshold for minor variant reporting.

   By default, the FASTA headers of the outputs will take the following format. If there are no subsequences, the first field will contain the reference name:

   `<subsequence name>_<cons>_<depth>_<reference name>`

`<name>.err` A small table generated to investigate 'STEP' errors[8]. This file can be ignored for the purposes of this practical exercise.

[5] Unlike some variant callers, **QuasiBAM** explicitly requires the BAM file to have been generated with only a single reference sequence – mapping to multi-reference FASTAs requires that the SAM/BAM files be split prior to variant calling analysis.
[6] The **`-S`** flag chops off the ends of lines that extend beyond a single screen width (as opposed to wrapping the text onto the next screen line). Watch out for effects like that seen at line 31!
[7] See https://www.gendx.com/SBTengine/Help_220/hs310.htm for more information on base codes.
[8] See the Variant Calling presentation for more information about these errors.

Questions

1. Have a look at the FASTA output (e.g. using **`less HXB2_pol.fas`**) and identify the two subsequences. Are they complete?

2. Are all bases present?

3. How many are missing at the 5’ end?

4. Examine the nucleotide frequency table **`HXB2_pol.txt`**. The third column gives the depths. Can these values be correlated with the bases identified in the previous question?

5. If you have access to *Calc* or *Excel* can the missing positions in RT be correlated with data in the nucleotide frequency table?

C: Re-analysing the **QuasiBAM** output

The problem with the initial **QuasiBAM** output is that there are regions where the depth is below 100 – the FASTA file has Ns at these loci. To generate a FASTA file with a lower depth, we can use a **perl** script[9].

By entering the following command, the usage is displayed:

```
GCV2025:~$ qb_reanalyse
```

The values in square brackets at the end of each line are the defaults, i.e. *the value that is used if no other value is specified*.

Looking at the usage, we need to provide an input file (**-i**). For this exercise, we also need to enter the new depth using **-d**. It is useful to supply both a name for the new FASTA file (**-o**) and a FASTA sequence header (**-s**). Try these:

```
GCV2025:~$ qb_reanalyse -i HXB2_pol.txt -d 30 -o HXB2_d30.fas -s HXB2_d30
```

The script is super quick as it is only reading the tabular **HXB2_pol.txt** file rather than the BAM file.

[9] **perl** is a low-level language similar in many ways to **awk**.

Questions

1. Look in **`HXB2_d30.fas`**, has the the complete sequence has been recovered by changing the minimum reportable depth to 30?

2. Have a look at the top of the **`HXB2_pol.txt`** file. How might you rewrite the **`qb_reanalyse`** code to make sure that the first five bases of protease ('PR') were reported?

3. What is the default *consensus* frequency for **`qb_reanalyse`**?

4. Think about whether you might have any concerns about using this frequency at depths below 100?

5. What about at lower frequencies, e.g. 10%, 5% or 2%?

6. Type **`quasi_bam`** into your shell. Can depth and consensus frequency be controlled at this stage?

D: The importance of the reference sequence when using mapping data

Repeat the initial analysis on the second BAM/FASTA filepair, replacing **`HXB2`** with **`contig`** in the filenames.

Questions

1. Take a look at the FASTA file with the default depth of 100. How is it different to the one using HXB2 as a reference?

2. Why might there be Ns at the beginning and ends of the reference sequences in both FASTAs (hint: think about where the data came from)?

E: Minority variants (interpretation)

Now let's look at minority variants.

Firstly, using the **qb_reanalyse** tool, generate two new consensus files, each with a depth threshold of 30 and a consensus threshold of 2 (i.e. we are going to look at all nucleotides present at ≥2%)

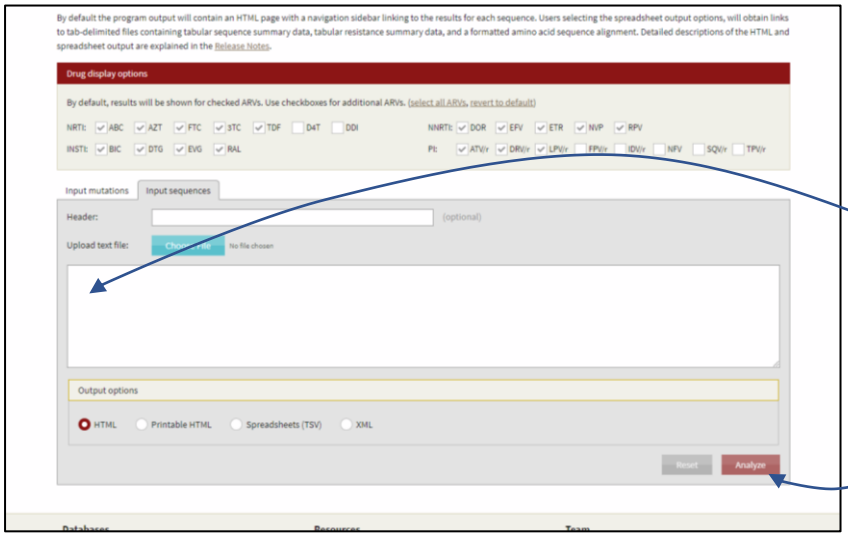
```
GCV2025:~$ qb_reanalyse -i HXB2_pol.txt -d 30 -c 2 -o HXB2_2%.fas -s HXB2_2%
```

Repeat the above command, replacing **HXB2** with **contig** in the filenames and header.

Open the files and submit the *RT sequences* to the online Stanford HIV drug resistance interpretation tool, found at the following URL:

https://hivdb.stanford.edu/hivdb/by-sequences/

Paste the sequences into the large text box, and click Analyze.



Notice the differences in the reports – specifically, the drug resistance interpretations and the quality information.

Questions

1. What are the key differences in output between the two sequences?

F: Minority variants (a closer look at the data)

Now let's look more closely at the **`HXB2/contig_pol.txt`** files. It is useful to be able to view the data in a more user-friendly format. If possible, open them both in *Excel* or *Calc*, whichever is available on your terminal.

Start by looking at the amino acid positions in RT where there were resistance mutations reported in Stanford:

65	This is a key position for multiple NRTI resistance, particularly for tenofovir and 3TC, two drugs used in many first line therapeutic regimens. Under- or over-reporting this mutation would have a significant impact on patient treatment.
77	Mutations at 77 are usually found as part of a multi-resistant array of mutations centred around Q151M. Found individually, they are of uncertain significance.
103	Lysine (K) is the wild-type, and usually it is N that is seen as at this position in viruses resistant to first-generation NNRTIs such as nevirapine and efavirenz.
184	Methionine (M) usually passes through an intermediate isoleucine (I) intermediate *en route* to valine (V) when HIV acquires resistance to 3TC and FTC. This reflects the mutational bias of retroviral reverse transcriptases.

Questions

1. Is there any trace of K65R in the *HXB2* mapping report at the amino acid or nucleotide level?

2. What is its frequency in the *contig* mapping report?

3. Could this be significant?

4. What about F77L in **`contig_pol.txt`**?

5. Why was this not seen in the Stanford report?

6. What is happening in the '**`G`**' column (column F) at position 103 in **`contig_pol.txt`**?

7. Why might this not be reflected in the **`AA`** and **`codon`** columns?

8. Imagine if it were established that in cases where M184I is present at 3% or above, it is considered prejudicial to treatment with 3TC and FTC. Is this sample resistant?

Now look at some other positions and try to see what is happening between the nucleotide and amino acid columns (3-8 vs. 19-21).

Questions (cont)

9. What is happening at positions 39 and 44 of RT? Why might **`HXB2_pol.txt`** be reporting gaps here?

10. What are the amino acid mixes at position 11 of PR and position 93 of RT?

11. What difficulties might be encountered when trying to establish the premise of Question 8 in the previous set?

12. Where did the Leucines seen at just over 1% at RT position 25 in the **`AA`** and **`codon`** columns (#20 & #21) of both **`.txt`** files come from, considering that there is no **`T`** reported at nucleotide position 74 (hint: look at the **`-f`** flag of **QuasiBAM**)?

MORE ABOUT THE QuasiBAM NUCLEOTIDE FREQUENCY TABLE

The meaning of each column is given in the following table:

#	Column header	Meaning
1	**Pos**	Position (within subsequence, if present)
2	**RefN**	Reference nucleotide at position **Pos**
3	**Depth**	Depth of coverage at that *nucleotide* locus
4	**A**	Frequency of A
5	**C**	Frequency of C
6	**G**	Frequency of G
7	**T**	Frequency of T
8	**Gap**	Frequency of gap (i.e. deletions)
9	**Ins**	Frequency of insertions
10	**I Desc**	Insert description & frequency
11	**Cons**	Consensus nucleotide at position **Pos**
12	**qA**	Average quality score of all As at this position
13	**qC**	Average quality score of all Cs at this position
14	**qG**	Average quality score of all Gs at this position
15	**qT**	Average quality score of all Ts at this position
16	**Apos**	Amino acid position within subsequence (if present)
17	**RCod**	Reference codon triplet starting at position **Pos**
18	**RAA**	Reference amino acid starting at position **Pos**
19	**AA Dep**	Depth of coverage of the *codon* starting at position **Pos**
20	**AA**	Frequency distribution of amino acids starting at position **Pos**
21	**Cod**	Frquency distribution of codon triplets starting at position **Pos**

It is from the data in columns 4-8 that column 11 is derived. From columns 3-10 are derived the FASTA outputs, with insertions and deletions being treated according to the depth and consensus parameters together with the flag for mixed indel handling (**-mg**).

Importantly, the **AA Dep** parameter in column 19 is used to calculate the frequencies in columns 20 and 21. Here, the quality scores of each nucleotide are paramount in explaining the difference in depths between columns 3 and 19. Essentially, the depth reported in column 3 is that of nucleotides passing a quality threshold (default=30, this can be adjusted with the **-m** flag) – i.e. if the run quality was poor, the number of *reads* that cover that position may be considerably higher. In this situation, the (hidden) frequency of low-quality bases will influence the number of triplets in which all bases pass the quality threshold.

For example, for a given codon triplet, if the ratio of high-quality bases to low-quality reads is high, e.g. 9:1 as in

Table 1 (below), then assuming the quality bases are independently distributed amongst all reads, there will be:

900 * 0.9 * 0.9 = 729 triplets with high quality bases at all positions.

However, if there are a large number of hidden low-quality bases as the 50% seen in

Table 2, although the nucleotide depths are identical, the number of high-quality triplets will be:

900 * 0.5 * 0.5 = 225.

In **QuasiBAM** outputs, the 900 high-quality bases is the number in the **`Depth`** column, and the number of high-quality triplets (e.g. 729 or 225) is the **`AA Dep`** column figure.

Table 1:

Codon position	High-quality bases	Low-quality bases
1	900	100
2	900	100
3	900	100

Table 2

Codon position	High-quality reads	Low-quality reads
1	900	900
2	900	900
3	900	900

FURTHER READING

Lefterova, M. I., Suarez, C. J., Banaei, N. & Pinsky, B. A. Next-Generation Sequencing for Infectious Disease Diagnosis and Management: A Report of the Association for Molecular Pathology. J. Mol. Diag. 17, 623-634 (2015).

Macalalad, A. et al. Highly sensitive and specific detection of rare variants in mixed viral populations from massively parallel sequence data. PLoS Comput. Biol. 8, e1002417 (2012).

McCrone, J. T. & Lauring, A. S. Measurements of Intrahost Viral Diversity Are Extremely Sensitive to Systematic Errors in Variant Calling. J. Virol. 90, 6884–6895 (2016).

Orton, R. J. et al. Distinguishing low frequency mutations from RT-PCR and sequence errors in viral deep sequencing data. BMC Genomics 16, 229 (2015).

Schlaberg, R. et al. Validation of metagenomic next-generation sequencing tests for universal pathogen detection. Arch. Pathol. Lab. Med. 141, 776–786 (2017).

Verbist, B. M. P. et al. VirVarSeq: a low frequency Virus Variant detection pipeline for Illumina Sequencing using adaptive base-calling accuracy filtering. Bioinformatics 31, 94-101 (2014).

Yang, X., Charlebois, P., Macalalad, A., Henn, M. R. & Zody, M. C. V-Phaser 2: variant inference for viral populations. BMC Genomics 14, 674 (2013).

ANSWERS TO QUESTIONS

Please note that the exact numbers may vary slightly owing to changes in software versions and online BLAST databases between the setting of the questions and the course itself.

B: Using QuasiBAM – nucleotide frequency tables

1. No
2. No
3. 27
4. Positions with depths <100 are set to N
5. Positions with depths <100 are set to N

C: Re-analysing the QuasiBAM output

1. Not quite – 5 bases are still Ns at the 5' end of protease, and 52 bases are still missing from the 3' end of RT.
2. Set the **`-d`** flag to 17 or lower.
3. 20
4. How reliably do the variant frequency percentages reflect 'true' frequencies in the original material? At higher depths, the number of sampled molecules is high, mitigating stochastic variation.
5. 5% of depth 20 would represent a single read having the variant.
6. Yes, **`-d`** and **`-c`** are flags here too.

D: The importance of the reference sequence when using mapping data

1. Fewer Ns, particularly in the middle of the sequence.
2. Because the PCR product is not much larger than the reference sequence. During Nextera tagmentation, library fragments are generated by cleaving the target DNA in two places and ligating barcode adapters to both ends. For a terminal base to be included in a fragment, one of the tagmenting cleavages must be even more terminal. Thus chance dictates that fragments are more likely to contain internal sequences, and up to 50 bases at either end of a dsDNA molecule can be unsequenced.

E: Minority variants (interpretation)

1. One more unusual PR mutation and 5 more unusual RT mutations in the *HXB2* output, when compared to the *contig* output. Resistance mutations F77FL seen in the *HXB2* but not in the *contig*, and K65KR seen in the *contig* output but not *HXB2*.

F: Minority variants (a closer look at the data)

1. No (rows 493-495)
2. 4.4335%
3. Yes, it may indicate low-level resistance to a number of drugs (abacavir and tenofovir especially).
4. Present at 1.24611%
5. The threshold was set at 2%
6. There are ~1% of reads with G at codon position 2 and the same at codon position 3.
7. The quality of the three codon position bases within the reads containing these Gs may be below the reporting threshold. As nucleotides, their quality may pass, but as codons, they may not. There is a default reporting frequency of 1% and it wouldn't take many poor quality codons to reduce their frequency below that limit.
8. Yes, but only if the contig data are consulted. If the incorrect reference is used (HXB2), then it would be declared as susceptible.
9. Notice the corresponding insertions of equal size at nucleotide positions 121 and 136. The reference mapper has struggled to align the reads correctly as HXB2 is too genetically distant from the sample virus.
10. PR11 is 96.2% V, 3.8% I, and RT93 is 100% G. Do not trust the *HXB2* outputs!
11. There are inevitably going to be large variations in variant frequency outputs between assays, owing to the laboratory methods (PCR vs. metagenomics, vs. Sequence Capture) and equally importantly the bioinformatic pipelines and reference sequences. Converging on a precise figure that would apply in all circumstances will be effectively unattainable. Only if a universal method was used could a cut-off be investigated.
12. As per the answer to Q7, there is a reporting threshold of 1%. The frequencies of C at codon position 2 (row 374) in both the *HXB2* and *contig* datasets are only just above 99%, suggesting there might be a second nucleotide present at >0.9%. If these are in reads of relatively high base quality, then their associated codon frequencies may exceed the 1% threshold. See the section "More about the QuasiBAM nucleotide frequency table" for more info.