**aDNA Data Analysis I: recovering mitochondrial genomes from NGS shotgun data**

**Scenario:** You have received NGS shotgun data for three skeletons from Sant Llàtzer, a medieval leprosarium in Barcelona: **UF104**, **UF703**, **UF801**. The genetic diversity of *M. leprae* strains across medieval Europe and within leprosaria was high. You would like to know if diverse *M. leprae* strains were maintained locally through community transmission, or if the pathogens were being introduced by people traveling from outside of Western Europe.

**To see if people were traveling from afar to Sant Llàtzer, you are going to process and analyze NGS data to determine the mitochondrial haplogroups of individuals buried within the leprosarium**.

Figure – DNA molecules from extraction to indexed libraries

**Part 1: Raw sequencing data and anatomy of a .fastq file**

DNA libraries are mixed together during sequencing. Sequencing centres generally send you demultiplexed .fastq files. ***Demultiplexing*** is the process of separating the sequences into their own unique sample files. If a DNA library is demultiplexed into one .fastq file or two .fastq files depends on the sequencing chemistry. In **single-end (SE)** sequencing, the DNA fragment is sequenced from only the forward direction, generating one read (**Fig. 1**). During **paired-end (PE)** sequencing, the DNA fragment is sequenced from both directions, resulting in two reads per fragment (**Fig. 2**).

A picture containing diagram

Description automatically generated

A picture containing diagram

Description automatically generated

**Figure 1.** SE sequencing

**Figure 2.** PE sequencing

These libraries were PE sequenced, so each library is represented by two **.fastq** files, R1 and R2. **Fastq** files, like many genomic file types, have a standardized format.

Each sequenced read is described by four rows:

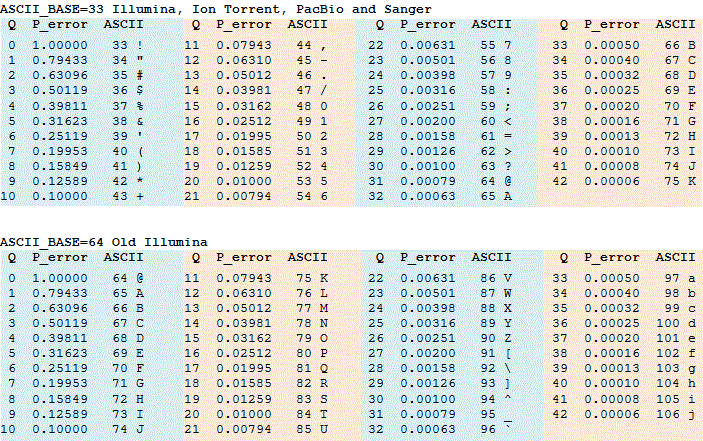
@ a header beginning with “@” with a unique library identifier, information about the sequencing run, such as lane and cluster coordinates.  
  
The base calls of the sequence  
  
+ a spacer line beginning with a “+”  
  
The quality score for each base call that represents the probability that the sequencing machine called the base by error (i.e., how strong and clear was the fluorescent signal?)

**Fastq** files are very large; they contain millions to billions of sequencing reads. Opening a text file this large can be difficult for many computers. Therefore, I have prepared two truncated pairs of **.fastq** files for you to investigate. Open the **1.fastq\_files** folder. Inside, open **UF703\_head\_R1.fastq** and **UF703\_head\_R2.fastq** in NotePad++ or a text editor of your choice (BBedit is a good one).

* What is the header for the forward read **@SRR15512701.3**? What does the information in the header indicate about this read?
* What is the ID of the corresponding reverse read?

It is critical that read pairs are kept in the same order in their respective files. If either a forward or reverse read mate is missing, the **.fastq** file is corrupt, and most tools will not be able to process it.

* Do **UF703\_head\_R1.fastq** and **UF703\_head\_R2.fastq** have the same number of reads?
* **UF703\_tail\_R1.fastq** and **UF703\_tail\_R2.fastq** are the last reads of the their respective .fastq files. How many reads were demultiplexed to library UF703?

**Fastq format** encodes base quality scores in ASCII format. This means the probability that the base was called in error can be represented by a single character. This is important because it means that every sequenced base corresponds to one quality score, so there are the same number of characters in the base call and quality score lines. A Q value ≥ 30 (**Fig. 3**) is generally accepted as a low enough probability of error. A good rule of thumb is if your quality scores look like they’re cursing you ([!@\*$?@?’./$/$@/=+&#](mailto:!@*$?@?’./$/$@/=+&)!!!), they are because your data are so bad.

**Figure 3.** Table of Q values, error probabilities, and ASCII characters.

* What is the reverse sequence for read **@SRR15512705.3698378**?
* Is it a good quality read? Why or why not?
* Identify a high quality read. What is the read’s ID? Is it a forward or reverse read?

**Part 2: Quality control of sequencing data using fastqc**

Clearly you cannot inspect every read individually! *Fastqc* is a tool that everyone in the field uses. It assesses .fastq files and reports ten quality metrics. Some of these metrics are more informative than others, and we want to keep track of them as we process the genomic data in the *mapping report*.

Open **Sant\_Llàtzer\_mapping\_report.xlsx**. Some of the fields are empty! You will fill them in based on the *Fastqc* reports and the files introduced in the rest of the practical.

You will find a *Fastqc* report for each set of forward (R1) and reverse (R2) reads for your three samples. **Open the .fastqc** and do the following.

* Use the *Fastqc* reports to fill in the missing cells in the ***Total reads or read pairs*** column in the **Sant\_Llàtzer\_mapping\_report.xlsx**. Remember, your data were PE sequenced.

*Fastqc* gives you some idea of how your data look by having a Check mark, Wingdings font, character code 252 decimal., !, or X.

* Look at the forward and reverse read reports for UF104. Look at the per base sequence qualities for both sets of read. What can you tell about the reads? Are the base quality distributions different?
* Choose a metric marked with a red X for any sample and interpret what information it conveys about the sequencing data.

Diagram

Description automatically generated

* Look at *Overrepresented sequences* and *Adapter content* sections for sample **UF104**. Looking at the figure above, why do you think adapter sequences are overrepresented in the sequencing data? *Remember that ancient DNA fragments are short and that sequencing occurs over a set number of base calling cycles*.

As you can imagine, artificial adapter sequences attached to biologically real sequences affect how the sequences are aligned to a reference genome. We need to get rid of them before proceeding with analysis. We would also like to get rid of bases with low quality scores, because they can erroneously affect results.

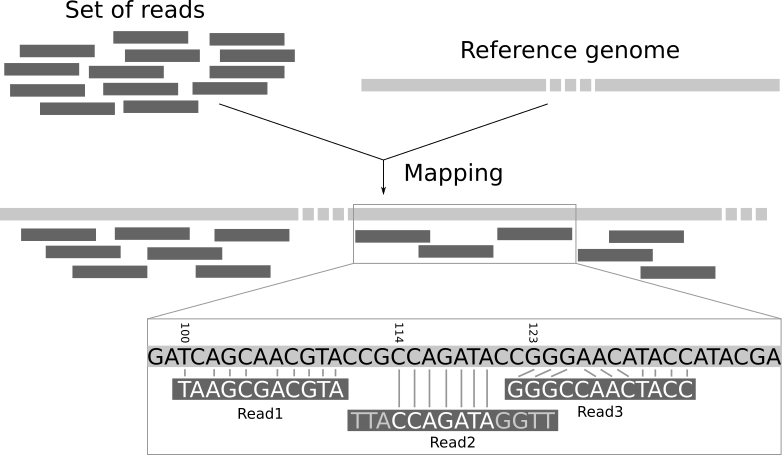
Another consideration with PE sequenced aDNA is that there is usually substantial overlap in the forward and reverse reads. The best way to deal with this is to collapse the read pairs into one sequence. I trimmed and merged these data using a tool called *Adapterremoval2*, and I re-ran *fastqc*.

Open the folder *Trimmed\_and\_merged\_fastqc\_reports* and the *Fastqc* reports inside.

* Why is there only one sample per *Fastqc* report now?
* How do the quality metrics of the trimmed and merged reads compare to the fastqc reports of the unmodified .fastq files?
* Why is per base and sequence (GC) content flagged for all quality filtered samples?
* Why is the sequence length distribution flagged for all quality filtered samples?
* Using the fastqc reports, fill in the *Total retained reads or read pairs after trimming (and merging for ancient)* and *proportion kept after trimming (and merging)* columns in the mapping report.
* The proportion of reads kept after trimming and merging is an important metric to considering when assessing sequencing read and run quality. Think of some reasons why.

**Part 3: Alignment of quality filtered sequencing reads**

Now that you have performed quality control measures on the .fastq data, the next step in the processing pipeline is to align the reads in the .fastq files to a reference genome. This process is called *mapping* or *aligning*. It is a critical step in your data analysis pipeline, and depending on how strictly or loosely you perform alignments, you can get false negative or false positive results. Below is a diagram of the mapping process.



The reference genome is in a one-dimensional file format called *fasta*. It is “one-dimensional” because it does not hold any higher-level information about quality, heterozygosity, read support, etc. A reference genome is a consensus sequence. It can be one long sequence or broken up into sections called contigs. These samples have been mapped to the **revised Cambridge reference sequence**, the primary mitochondrial reference genome.

Open the folder **3.sam\_files** and open **rCRS.fasta** in NotePad++.

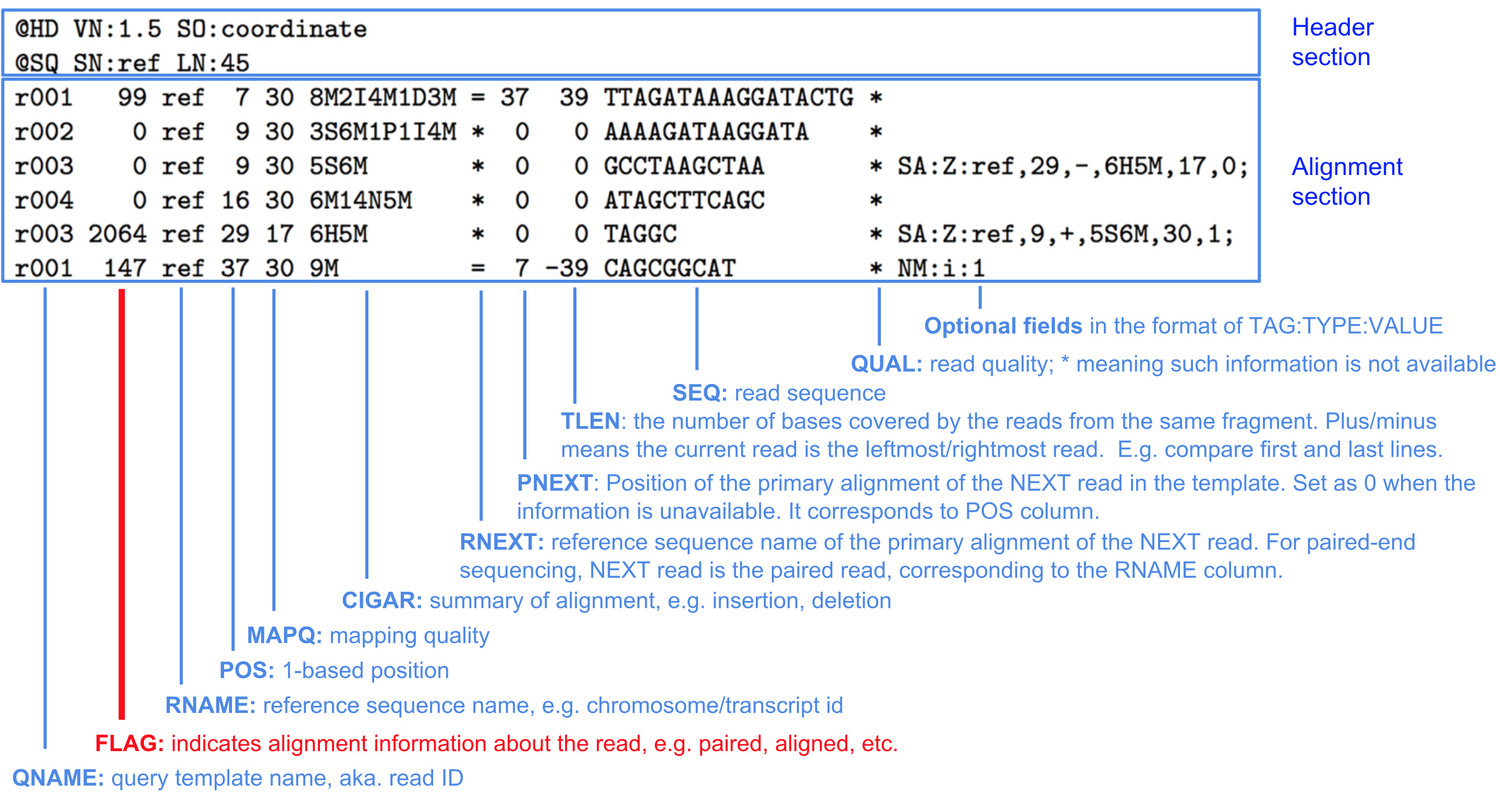
* Based on what you see, what do you think fasta format is?

The output of alignment/mapping is a .**sam/.bam** file. .sam and .bam have the same structure and convey the exact same information, but .bam files are the compressed version of .sam files. They are not human readable, but they are computer readable. Analysis of .bam files is more computationally efficient that .sam files. What you see when you open a .bam file:

A picture containing text

Description automatically generated

**.sam/.bam** files have a mandatory header section. The header usually contains the sample name, the name of the reference genome, and the tool and commands used to generate the alignments. Each row corresponds to a sequencing read, the QNAME. There are 11 mandatory fields, including the QNAME and many optional fields. Below is diagram of a .sam file.



Important fields are the **FLAG**, **MAPQ**, and **CIGAR**. The **FLAG** field gives bit-coded information about the read, for example whether it is mapped to the reference, whether it mapped but its pair did not (not applicable with merged reads), and whether it was mapped to more than one place in the reference. The **MAPQ** field provides an overall quality score for how well the read mapped. The **CIGAR** string is a coded explanation of the alignment, which was used to calculate the mapping quality.



For example, the CIGAR string of the above alignment is 3M7N4M. There are three bases mapping following by a 7 bp gap followed by 4 mapped bases. The simplest CIGAR strings produce the highest mapping quality scores.

I have provided you with filtered .sam files. Using the -f 4 FLAG, I filtered out all of the unmapped reads. Using the -q 25 MAPQ, I filtered out all the reads with a mapping quality score < 25. I have also run a tool that removes duplicate reads, which do not add information.

Open **UF703\_f4\_q25\_sortc\_markdup.sam.** Do you recognize the read IDs from the .fastq files?

* Find read M\_SRR15512701.615723. What is the mapping quality? What is the CIGAR score?
* Overall, do these CIGAR strings look like high quality alignments? Why?

By now you’re aware that aDNA has characteristic damage that result from miscoding lesions. These miscoding lesions are important for authenticating aDNA, but also they can seem like SNPs/SNVs in the sequenced reads. One way to deal with this is by using a tool called MapDamage to rescale the base quality scores for C -> T and G -> A transitions. Rescaled bases have low quality scores so that they *will not* be called as variants by a variant caller. MapDamage takes a .bam as input and outputs a rescaled .bam file.

Open **UF703\_f4\_q25\_sortc\_markdup.rescaled.sam** and compare with **UF703\_f4\_q25\_sortc\_markdup.sam.**

* Look at line 20 of the rescaled .sam. Have any of the base quality scores for that read been rescaled?
* Look at line 237 of the rescaled .sam. Have any of the base quality scores for that read been rescaled?

**Part 4: Using qualimap to determine reference coverage and depth**

Qualimap is a common tool used to assess genome coverage and depth from .bam files. I have run the tool for you, and you have the outputs: a .pdf report and a .txt file for each sample.

* Using the .pdf reports fill out the following columns in the **Sant Llátzer mapping report**: Q25 mapped reads after duplicate removal, Avg length of mapped Q25 reads, Mean Cov, and SD Cov
* Choose one of the figures in the Qualimap .pdf report. Explain what this figure/calculation is conveying about your mapped reads. If you are not sure how to interpret the figure, see the Qualimap manual: <https://hpc.nih.gov/docs/QualimapManual.pdf> Looking up obscure details about tool reporting is 40% of bioinformatics, so it’s worth practicing!

Now that you have read through the .pdf reports, open the Qualimap .txt file. Much but not all the information from the .pdf is reported here as well. Text file outputs are useful because you can write code that will loop through .txt files and pull out the information you need. Most geneticists automate their mapping reports so that they do not have to manually pull out each piece of information.

* Complete the next two columns in your **Sant Llátzer mapping report:** **%ref covered at >=1x** and **%ref covered at >=5x** using the Qualimap .txt files.

Some of the metrics you need to assess your data are not generated by a tool, so you need to generate them yourself. **Cluster factor**, also known as **library complexity**, is informative about the number of unique fragments in your library. It tells you on average how many times a read was sequenced. For example, cluster values over 2 indicate that on average, every fragment has been sequenced twice, so the library is probably exhausted of DNA from this organism. Depending on how deeply you sequence and if you’ve done targeted enrichment, cluster factors can be huge, anywhere from 20 to 30.

* Calculate cluster factor for each sample using the formula in the cell and the values in the columns you’ve already filled out.

Endogenous DNA frequency can be used in tandem with cluster factor to decide whether to sequence your library more deeply or determine if how efficient your capture enrichment was.

* Calculate the frequency of endogenous DNA (mtDNA in this case) for each sample using the formula in the cell and the values in the columns you’ve already filled out.

**Part 5: MapDamage plots**

There are several tools that produce iconic aDNA “damage plots,” but here we are using the tool MapDamage because it also rescales base qualities, which other damage profiling tools do not. Open **5.MapDamage\_outputs**.

Open the **fragment misincorporation plots** for each sample. The four plots at the top indicate base (A, C, T, and G) frequencies within the read (within the grey square) and outside of the read. **What are these plots illustrating?** Hint: depurination.

The bottom two plots are the most common damage plots that you may have seen before. The red line charts C to T substitutions, and the blue line charts G to A substitutions.

* From the bottom two plots, estimate the values for the last four columns in your **Sant Llátzer mapping report**.
* Based on these plots, do all of the samples have authentic aDNA?

**Part 6: Inspecting variants**

Finally, what everyone is really interested in: the genetic variation! Variant calling is the process of identifying genetic variants in your sample based on the sequence alignments. Point mutation SNPs/SNVs are the most common variants in aDNA. This is because it can be hard to confidently identify structural variants due to incomplete genome recovery.

There are many variant callers (freebayes, GATK Haplotypecaller, BCFtools, VarScan, and GATK UnifiedGenotyper, just to name a few). Some of them are better for working with aDNA.

Like the other files we have looked at today, VCF has a standardized format starting with a header with general information about the version, variant calling tool, and a list of abbreviations used in variant reporting and what they mean. Each row corresponds to a variant and there are several columns of mandatory information for each variant:

Text

Description automatically generated

Instead of using the VCF files to identify variants in our samples, we are going to manually investigate the .bam files in **Integrated Genome Browser (IGV).** Our genetic data are low coverage, so we need to manually inspect SNPs. We will also use this as an opportunity to see how different decontamination methods modified our .bam files.

Open IGV and load reference genome by click “Genomes” -> “Load genome from file” and select the file rCRS.fa. The reference genome is now loaded.

Now go to “File” -> “Load from file” -> select all three bams from the same sample (.bam, .rescale.bam, and .pmds3filter.bam). You should three tracks in IGV, one from each .bam. It should look like this:

A picture containing graphical user interface

Description automatically generated

The hypervariable regions of mtDNA are useful for analysing lineage diversity, because they are the most polymorphic regions of the mitochondrial genome. There are two hypervariable regions in mtDNA: **HVRI (positions 16024–16383**) and **HVRII (positions 57–372).**

For each sample, manually scan the hypervariable regions using the arrows at the top. It will be easier for you to zoom in all the way. Search the hypervariable regions for evidence of true variants and mark the ones you identify in the table below.

The way to mark polymorphisms is **reference base + position # + variant base**

For example, **A374C** would be an A -> C transversion at position 347.

While you are scanning the sites, see if you can tell the difference between the three .bam files. Remember, .bam hasn’t been filtered for contamination, .rescaled.bam has had its bases rescaled to low qualities, and .pmds3filter.bam has been filtered to remove reads that are likely contaminants.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UF104** | | **UF703** | | **UF801** | |
| HVI  16024–16383 | HVII  57–372 | HVI  16024–16383 | HVII  57–372 | HVI  16024–16383 | HVII  57–372 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

After sorting through both hypervariable regions, check your variants against the known list of mtDNA polymorphisms at <https://www.phylotree.org/tree/>

* Can you assign a haplogroup to any of the individuals?