MHC Genomics Analysis Project - Task 3 Report

Pairwise Alignment and Allele Typing

Mahla Entezari*

Bioinformatics Assignment 1

January 6, 2026

1 Introduction

The goal of this task is to perform MHC-focused allele typing by comparing MHC gene sequences extracted from each sample to a database of known HLA alleles. Specifically, this task requires (i) extracting MHC genes from sample-level tables, (ii) converting the extracted sequences into FASTA format, and (iii) performing pairwise sequence alignment against reference allele sequences to identify the best-matching allele for each gene based on alignment score and sequence similarity.

2 Materials and Methods

2.1 Input Data

The following inputs were used:

- Per-sample gene tables from Task 1 in CSV format: `project/csv/MOT*.task1.csv`
- Per-sample MHC FASTA sequences produced by Task 3 Step 2: `project/task3/step2_mhc_fasta/*.mhc_step2.fa`
- Reference HLA allele nucleotide sequences downloaded from the IPD-IMGT/HLA database mirror (ANHIG/IMGTHLA GitHub), locus-specific FASTA files combined into a single reference file:

  `project/ref/hla_alleles.fa`

2.2 Step 1: Extract MHC Genes from Sample Tables

In Step 1, each sample CSV (`MOT*.task1.csv`) was filtered to retain only rows corresponding to MHC-related genes. The output per sample was written as:

- `project/task3/step1_mhc_tables/MOTxxxx.mhc_step1.csv`

A practical issue encountered at this stage was that some CSV fields (particularly the sequence column) can exceed Python's default CSV field size limit. This was addressed by increasing the parser field limit in the Step 1 implementation to ensure robust reading of long sequences.

2.3 Step 2: Convert MHC Tables to FASTA

In Step 2, each `*.mhc_step1.csv` file was converted into FASTA format. Each FASTA record used a header containing the gene name, the MHC class label, and genomic coordinates, for example:

```
>HLA-A|class=I|chr6:...
ACGT...
```

Outputs were written to:

- `project/task3/step2_mhc_fasta/MOTxxxx.mhc_step2.fa`

A sanity check was performed by verifying that the number of FASTA records (count of `>`) matches the number of MHC rows in Step 1 logs for each sample.

2.4 Step 3: Pairwise Alignment and Allele Typing

2.4.1 Reference Allele Database

Reference HLA allele sequences were assembled from locus-specific IPD-IMGT/HLA nucleotide FASTA files (`A_nuc.fasta`, `B_nuc.fasta`, `C_nuc.fasta`, `DRB1_nuc.fasta`, ...) downloaded via HTTPS and concatenated into:

- `project/ref/hla_alleles.fa`

*mahlaentezari.sbu@gmail.com

Because IMGT/HLA FASTA headers may not begin with the allele name directly (often containing internal IDs such as `HLA00001` followed by the allele designation), reference header parsing was implemented using a robust pattern search over the entire header line to reliably extract locus and allele (`A*01:01:01:01`, `DRB1*15:01:01`).

2.4.2 Alignment Strategy

For each sample gene sequence, pairwise global alignment was performed against reference allele sequences corresponding to the same gene (locus). A Needleman–Wunsch global alignment (dynamic programming) was used with a simple scoring scheme:

- Match score: +2
- Mismatch score: -1
- Gap penalty: -2

The best allele was selected using:

1. Highest alignment score
2. Tie-breaker: higher percent identity

For genes without a corresponding allele set in the reference database (many class III genes), the output fields were set to empty/zero values.

2.4.3 Output Format

The allele typing output was written in the required tabular format (matching the provided example) with columns:

`sample_id,gene_name,mhc_class,best_allele,read_support,mean_alignment_score,percent_identity`

Here, `read_support` represents the number of sequence records supporting a gene in the sample FASTA (typically 1 per gene in this workflow). The `mean_alignment_score` and `percent_identity` summarize the best-match alignment(s) for that gene.

Outputs were generated under:

- `project/task3/step3_alignment/MOTxxxx.mhc_allele_typing.real.csv`

3 Results

3.1 Pipeline Completion Across All Samples

This task was executed successfully for all 20 samples. The following outputs were generated consistently:

- 20 Step 1 MHC tables: `project/task3/step1_mhc_tables/*.mhc_step1.csv`
- 20 Step 2 FASTA files: `project/task3/step2_mhc_fasta/*.mhc_step2.fa`
- Step 3 allele typing outputs per sample: `project/task3/step3_alignment/*.real.csv`

3.2 Allele Typing Output Characteristics

As expected, HLA class I and class II genes (`HLA-A`, `HLA-B`, `HLA-C`, `HLA-DRB1`, `HLA-DQB1`, ...) produce non-empty best allele matches. In contrast, many class III genes (complement-related genes) do not have allele entries in the HLA reference set, and therefore appear with empty `best_allele` and zero scores/identity.

The final per-sample CSV outputs therefore contain a mixture of typed HLA genes and non-typed (non-HLA) MHC-region genes, reflecting the scope of the reference allele database.

4 Discussion

Demonstrated an end-to-end MHC allele typing workflow driven by pairwise sequence alignment, two implementation details were essential for correctness:

- **Handling long sequence fields in CSV:** Without raising the CSV field size limit, Step 1 can fail when sequences exceed the default limit.
- **Robust IMGT/HLA header parsing:** IMGT/HLA FASTA headers often contain internal identifiers; extracting allele names requires searching the full header for locus*allele patterns.

5 Conclusion

In this task, MHC genes were extracted from per-sample gene tables, converted to FASTA, and typed against a curated reference allele database using pairwise global alignment. The pipeline generated consistent outputs across all samples and produced allele calls for HLA genes based on alignment score and percent identity.