UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA

Implementation of a data analysis pipeline for the genetic characterization of non-seasonal influenza A WGS samples in the context of laboratory surveillance of viral outbreaks

João Luís Gomes Pereira

Mestrado em Bioinformática e Biologia Computacional

Trabalho de Projeto orientado por:
Doutor Daniel Vieira Noro e Silva Sobral
Professor Doutor Francisco José Moreira Couto

2025

Agradecimentos

Aos meus pais, pelo apoio moral e financeiro nesta pequena grande aventura que foram estes dois últimos anos.

Ao meu orientador Daniel Sobral pela paciência e tranquilidade com que orientou a pilha de nervos que eu sou.

Ao Vitor Borges por ser o quality assurance mais implacável, mas também a pessoa que mais rápidamente nos faz ver a luz ao fundo de um túnel.

À equipa de bioinformática do DDI do INSA pela oportunidade de trabalhar com pessoas que nos fazem perceber que os heróis não usam capas. Em especial ao João Santos por toda a ajuda que me deu no decorrer deste trabalho.

Ao meu co-orientador Professor Francisco Couto que me orientou e me deu coragem do início até ao fim do mestrado.

Aos laboratórios de referência:

O laboratório de referência da gripe e vírus respiratórios do INSA coordenado pela Doutora Raquel Guiomar pelo apoio, disponibilidade e amostras facultadas.

Ao INIAV-IP e em especial à Doutora Margarida Henriques por me ter facultado as sequências dos surtos de HPAI portugueses que foram instrumentais para o proof-of-concept deste projeto

Aos meus amigos de sempre, Nuno, Ritas, Maria, Tiago, Joana, Sara, Pedro e Mariana que me acompanharam sempre nesta jornada.

Aos meus novos amigos Paulo, Simona, Miguel, Filipa e Joana. Quem eu conheci no mestrado e não me deixaram de acompanhar.

À Inês Diniz pelas horas intermináveis em lojas de aquariofilia, jantares no sushi e conversas e conselhos que vou levar comigo.

À Joana Soares, quatro anos depois e uma missão cumprida!

Aos meus patrões no decorrer deste mestrado, Bruno Vasques e Inês Leal, que sem o vosso apoio no ajuste da minha atividade profissional eu não teria conseguido fazer este mestrado tão bem.

Aos meus colegas de clínica Joana, Fábia, Inês, Helena, Tatiane, Dienifer, Ana, Rita e Patrícia. Vocês são as maiores!

Às minhas irmãs, cunhados e sobrinhos por serem a família que aturaram estes anos mais chatos.

Abstract

Background: Influenza A viruses (IAV) are rapidly evolving pathogens with high zoonotic and pandemic potential. Their segmented genome allows antigenic drift and reassortment, key drivers of adaptation and cross-species transmission. The ongoing H5Nx panzootic underscores the need for timely genomic surveillance to detect adaptive mutations, reassortments, and antiviral resistance. Existing frameworks such as the INSaFLU-TELEVIR platform, work well for seasonal strains but face challenges with non-seasonal IAV due to reference selection, representation bias, and database redundancies.

Objectives: This project aimed to develop an automated pipeline for the genetic characterization of non-seasonal IAV whole-genome sequencing (WGS) samples. Goals were: (1) accurate identification of genomic segments, subtypes/genotypes, host origins and closely-related reference sequences per segment; (2) characterization of mutations of biological relevance, including host adaptation and antiviral resistance; (3) integration of results into user-friendly and machine-readable outputs.

Methods: The pipeline, named *AFluID* (Automatic Influenza Identification pipeline), was implemented in Python and combined clustering (cd-hit), similarity search (BLAST), clade assignment (Nextclade), and mutation screening (FluMut). It was validated on curated datasets from NCBI, GISAID, EQA panels, and Portuguese outbreak samples resulting from an INSA-INIAV-IP cooperation.

Results: *AFluID* rapidly identified IAV segments and subtypes across datasets, while its multi-feature design further streamlined the identification of closely-related references (and detection of reassortment events), along with clade classification, host/geographic inference, and identification of mutations potentially linked to adaptation, virulence, and resistance. Proof-of-concept analysis of outbreak samples confirmed applicability in real surveillance scenarios.

Conclusions: *AFluID* addresses major limitations of current pipelines by offering an automated, scalable, and reproducible framework tailored for non-seasonal IAV. Although reassortment detection requires further refinement, the pipeline strengthens laboratory surveillance capacity and represents a step toward integration with global frameworks such as INSaFLU.

Keywords: Influenza A virus, genomic surveillance, sequence classification, mutation analysis, AFluID

Resumo

A gripe causada pelo vírus *Influenza A* (IAV) constitui uma ameaça persistente e multifacetada à saúde pública mundial, consequência direta da sua notável capacidade de evolução genética e adaptação a diferentes hospedeiros. O genoma segmentado destes vírus, composto por oito segmentos de RNA de sentido negativo, possibilita dois mecanismos complementares de variabilidade: a *deriva antigénica*, resultante da acumulação progressiva de mutações pontuais, e a *recombinação de segmentos* entre estirpes co-infetantes, responsável pelo surgimento súbito de variantes com perfis antigénicos e biológicos profundamente distintos. Estes processos são centrais para a dinâmica evolutiva do IAV e explicam a sua capacidade de escapar à imunidade pré-existente, adaptando-se rapidamente a novos hospedeiros e ecossistemas. Consequentemente, o vírus mantém-se uma das principais ameaças zoonóticas globais, sendo capaz de originar pandemias devastadoras quando determinadas combinações genéticas aumentam a transmissibilidade entre mamíferos. O atual panzootismo de estirpes H5Nx, com impacto documentado em aves selvagens, aves domésticas, suínos e várias espécies de mamíferos, incluindo casos esporádicos em humanos (Peacock et al. [2024], Alvarez et al. [2025]), demonstra a urgência de uma vigilância genómica reforçada, abrangente e adaptável a um cenário de diversidade viral sem precedentes.

Embora plataformas consolidadas como o INSaFLU-TELEVIR (Borges et al. [2018], Santos et al. [2024]) tenham revolucionado a vigilância genómica de vírus respiratórios, introduzindo sistemas reprodutíveis e padronizados, persistem desafios quando aplicadas à análise de IAV não sazonais. Estes desafios derivam essencialmente de três fatores: a dependência de referências adequadas, a subrepresentação de genótipos aviários nas bases de dados públicas e a redundância massiva de sequências altamente semelhantes, que compromete simultaneamente a eficiência computacional e a precisão da classificação. O impacto cumulativo destas limitações manifesta-se na dificuldade em caracterizar de forma rápida e rigorosa amostras emergentes e em incorporar novas estirpes nos fluxos de vigilância laboratorial. A Organização Mundial de Saúde (2023) e o ECDC/EFSA (2025) têm salientado a importância de plataformas integradas e interoperáveis que minimizem redundâncias, melhorem a curadoria de metadados e permitam a deteção precoce de combinações genéticas potencialmente críticas para a saúde pública.

Neste contexto, o presente trabalho teve como objetivo o desenvolvimento do **AFluID** (*Automatic Influenza Identification pipeline*), um sistema automatizado e modular concebido para a caracterização genética de amostras de IAV obtidas por sequenciação de genoma completo (*Whole Genome Sequencing* - WGS). O **AFluID** foi projetado de modo a integrar num único fluxo reprodutível as principais etapas analíticas da vigilância genómica: identificação e validação de segmentos, tipificação genética, análise filogenética e epidemiológica, pesquisa de referências mais próximas e deteção de mutações de relevân-

cia biológica. A filosofia de desenvolvimento assentou na flexibilidade, escalabilidade e transparência metodológica, garantindo compatibilidade com os fluxos laboratoriais já existentes. Os objetivos específicos incluíram: (1) identificar automaticamente segmentos genómicos, subtipos/genótipos e potenciais origens do hospedeiro em sequências de consenso ou contigs; (2) caracterizar mutações com relevância biológica e epidemiológica, incluindo marcadores de adaptação ao hospedeiro, virulência e resistência antiviral; e (3) integrar os resultados em relatórios legíveis por humanos e em formatos tabulares legíveis por máquina, assegurando interoperabilidade entre plataformas laboratoriais e repositórios internacionais.

O **AFluID** foi desenvolvido integralmente em Python e integra ferramentas consolidadas da bioinformática moderna. A ferramenta *cd-hit* (Li and Godzik [2006]) foi utilizado para reduzir redundâncias e mitigar enviesamentos de representação nas bases de dados, agrupando sequências em *clusters* de homologia e selecionando representantes não redundantes que preservam a diversidade informativa. BLAST (Camacho et al. [2009]) foi incorporada como método complementar de classificação e confirmação, atuando tanto na validação de resultados ambíguos como na curadoria manual de metadados, assegurando maior precisão na atribuição de segmentos e subtipos. A ferramenta *Nextclade* (Aksamentov et al. [2021]) permitiu a atribuição de clados filogenéticos e a normalização dos critérios de qualidade das sequências, enquanto a ferramenta *FluMut* (Giussani and Sartori [2024]) foi integrada para a deteção e anotação de mutações reportadas na literatura, com impacto reconhecido em adaptação, virulência e resistência a antivirais. O pipeline foi validado com múltiplos conjuntos de dados: sequências de IAV do NCBI, subconjuntos enriquecidos de estirpes aviárias da GISAID (Shu and McCauley [2017], Khare et al. [2021], Elbe and Buckland-Merrett [2017]), painéis de Avaliação Externa da Qualidade (EQA) fornecidos pelo INSA e amostras reais de surtos ocorridos em Portugal no âmbito da colaboração INSA-INIAV-IP.

Resultados e Discussão: Os resultados obtidos demonstraram que o **AFluID** identifica com elevada consistência segmentos e genótipos de IAV, mesmo em amostras de baixa cobertura, fragmentação acentuada ou qualidade desigual. A aplicação da *cd-hit* reduziu substancialmente a redundância de sequências, acelerando o processamento sem perda de exatidão, sobretudo em bases de dados extensas como o NCBI e a GISAID. Esta otimização contribuiu para uma representação mais equilibrada de genótipos aviários, tradicionalmente sub-representados. A agregação automática de metadados por *cluster* permitiu associar cada grupo de sequências a hospedeiros, regiões e períodos de isolamento, acrescentando um valor epidemiológico direto, em conformidade com as recomendações da WHO (2023).

A análise de mutações revelou que a *FluMut* detetou de forma robusta alterações associadas à adaptação a mamíferos, à resistência a antivirais e ao aumento da virulência (Alvarez et al. [2025]; Holmes et al. [2021], Mohapatra et al. [2023]). Entre as mutações mais relevantes observaram-se substituições nos genes PB2 e PA, associadas à replicação eficiente em células de mamíferos, e alterações no gene HA, relacionadas com modificações na ligação ao recetor e na estabilidade do virião. Estas observações são particularmente úteis para a avaliação rápida do risco de transmissão interespecífica e para a priorização de medidas laboratoriais de biossegurança e vigilância.

Além da análise de mutações, o pipeline demonstrou capacidade preliminar para identificar padrões compatíveis com *recombinação de segmentos*, fenómeno central na evolução do IAV e frequentemente associado ao aparecimento de estirpes pandémicas (Centers for Disease Control and Prevention [2024a]). Apesar de esta funcionalidade se encontrar ainda em fase exploratória, os resultados obtidos indicam o

potencial do **AFluID** como ferramenta de deteção precoce de combinações genéticas emergentes e como apoio à monitorização integrada de eventos de diversidade genética.

A prova de conceito com amostras reais da parceria INSA-INIAV confirmou a aplicabilidade prática e a flexibilidade do sistema. O **AFluID** demonstrou compatibilidade plena com diferentes métodos de montagem, nomeadamente o utilizado pelo INSaFLU (baseado em *Snippy*) e o IRMA, assegurando reprodutibilidade entre referências canónicas e referências obtidas automaticamente. O sistema mostrou ainda desempenho consistente em contextos laboratoriais com recursos variáveis, mantendo estabilidade e fiabilidade na análise de dados provenientes de múltiplas origens. A exportação dos resultados em formatos estruturados e legíveis por humanos simplificou a integração dos outputs com relatórios técnicos e plataformas de vigilância, reduzindo significativamente o tempo necessário à interpretação e partilha dos dados.

Problemas encontrados e mitigações: Durante o desenvolvimento e validação do **AFluID**, foram identificados diversos desafios técnicos e metodológicos. A *qualidade desigual dos metadados* das bases de dados públicas, particularmente do NCBI, conduziu a ambiguidades na anotação de segmentos e hospedeiros. Este problema foi mitigado através de curadoria manual, harmonização de nomenclaturas e aplicação de filtros de qualidade. Observaram-se igualmente *clusters ambíguos* contendo sequências de segmentos distintos (por exemplo, HA com H1 e H3), cuja resolução exigiu redefinição de critérios de homologia mínima e validação cruzada com BLAST. Finalmente, as *limitações intrínsecas dos algoritmos*, nomeadamente a sensibilidade da *cd-hit* ao comprimento das sequências, não foram completamente eliminadas, mas foram propostas três estratégias mitigadoras: (1) definição de *superclusters* baseados em homologia e contexto epidemiológico; (2) exclusão de clusters pouco representativos e remoção de *outliers* de tamanho; e (3) substituição gradual da *cd-hit* por BLAST em etapas críticas, equilibrando precisão com desempenho computacional. Estas medidas aumentaram a consistência interna dos resultados e prepararam o pipeline para futuras expansões.

Conclusões: O **AFluID** constitui um avanço substancial na vigilância genómica da gripe, ao disponibilizar um quadro automatizado, escalável e reprodutível, especificamente adaptado à diversidade genética dos IAV não sazonais. O sistema alia eficiência computacional a flexibilidade modular, permitindo a identificação rápida de segmentos, subtipos e mutações relevantes, mesmo em dados incompletos ou de qualidade variável. Embora a deteção de *recombinação de segmentos* permaneça em desenvolvimento, o **AFluID** demonstra potencial evidente para reforçar a capacidade de resposta a estirpes emergentes e para facilitar a interoperabilidade entre laboratórios e plataformas de vigilância. A aplicação prática confirmou a sua robustez, comprovando a utilidade do sistema em cenários laboratoriais reais. O pipeline representa, assim, um contributo concreto para a modernização e padronização da vigilância genómica, em consonância com as orientações internacionais que apelam a sistemas abertos e reprodutíveis. No futuro, prevê-se a expansão dos princípios trabalhados no **AFluID** com sistemas dedicados à deteção automatizada usando algoritmos de inteligência artificial de *recombinação de segmentos* em vírus segmentados, e à integração direta com o INSaFLU-TELEVIR, fortalecendo a capacidade global de resposta a ameaças zoonóticas e pandémicas emergentes.

Palavras-chave: Influenza A; vigilância genómica; classificação de sequências; análise de mutações; AFluID.

Contents

List of Figures

List of Tables

Acronyms

IAV	Influenza A Viruses
IBV	Influenza B Viruses
ICV	Influenza C Viruses
IDV	Influenza D Viruses
HA	Haemagglutinin
HEF	Haemagglutinin-esterase-fusion protein
NA	Neuraminidase
MP	Matrix Proteins
NP	Nucleoprotein
NS	Non-structural protein(s)
NEP	Nuclear export protein
PB2	Polymerase Basic 2
PB1	Polymerase Basic 1
PB1-F2	Polymerase Basic 1 - Frame 2 protein
PA	Polymerase Acidic
PA-X	Polymerase acidic X protein
SA	Sialic Acids
RdRp	RNA-dependent RNA polymerase
RNP	Ribonucleoprotein(s)
WHO	World Health Organization
WOAH	World Organization for Animal Health
CDC	Center for Disease Control and Prevention
ECDC	European Centre for Disease Control and Prevention
EFSA	European Food Safety Agency
INSA	Instituto Nacional de Saúde Doutor Ricardo Jorge
INIAV	Instituto Nacional de Investigação Agrícola e Veterinária
IP	Instituto Público
CLI	Command line interface
IO	Input(s)-Output(s)
ONT	Oxford Nanopore Technologies
WSL	Windows Linux subsystem

Chapter 1

Introduction

1.1 Motivation

Influenza A viruses (IAV) are rapidly evolving pathogens that have pandemic potential, as confirmed in recent history (Liang [2023]).

These viruses have a wide wildlife reservoir and their segmented genome allows not only for frequent genetic drift but also genetic shifts that have the potential to create new and virulent or contagious subtypes that can jump species barriers and cause severe and/or widespread disease.

The previous Influenza A pandemics were traced to genome re-assortment and species spillover events namely avian to mammal on the 1918 H1N1, 1957 H2N2, 1968 H3N2, 2004 H5N1 and the 2009 H1N1pdm pandemics (Liang [2023]).

Currently, since 2021 a new H5N1 isolate (clade 2.3.4.4b), is causing widespread disease in animals worldwide, with several outbreaks detected in areas as remote as Antarctica and species not traditionally associated with H5N1 infection such as cattle (Peacock et al. [2024]).

Current spillover and genetic re-assortment events have been frequent in the latest circulating subtypes of H5Nx IAV, and thus screening of isolate genetic segments and their likely origin as well as the most frequent mutations of biological interest is paramount.

1.2 State of the art

1.2.1 Influenza A virus

Influenza virus, aetiological agents of the disease commonly known as the flu are viruses belonging to 4 genera of the *Orthomyxoviridae* family, *Alphainfluenzavirus* (Influenza A virus - IAV), *Betainfluenzavirus* (Influenza B virus - IBV), *Gammainfluenzavirus* (Influenza C virus - ICV) and *Deltainfluenzavirus* (Influenza D virus - IDV)(Liu et al. [2020]).

Influenza, or the flu, is an acute, highly contagious, respiratory disease caused by the aforementioned viruses in which only IAV has the greatest zoonotic potential for it infects a broad range of species, both mammal and avian, having shorebirds and waterfowl as their main wildlife reservoir and the only one known to have caused pandemic crisis (Merriam-Webster [2025]).

All Influenza viruses are segmented, negative sense, single-stranded RNA (ssRNA) viruses. IAV and IBV have 8 genomic segments whereas ICV and IDV have 7 genomic segments, in the latter viruses the main difference is the existence of only one surface protein, the Haemagglutinine-esterase-fusion protein (HEF) instead of the Haemagglutinin (HA) and Neuraminidase (NA) proteins found in IAV and IBV (Liu et al. [2020]).

1.2.1.1 Genome characterization

As mentioned before IAV is a segmented negative sense ssRNA virus comprising 8 segments, that encode 12 proteins, numbered by size as follows:

1. Polymerase Basic 2 (PB2) - A component of the RNA-dependent RNA polymerase (RdRp) complex, that also inhibits JAK/STAT signaling, inhibiting host immune response.

2. Polymerase Basic 1 (PB1) - A component of the RdRp complex that also degrades the host cell's mitochondrial antiviral signaling protein. This segment also encodes the protein PB1-F2 (Polymerase Basic 1 Frame 2) that does not participate in viral replication, only interfering with the host's immune response.

3. Polymerase Acid (PA) - A component of the RdRp complex, more active in acidic pH. This segment also encodes the PA-X protein that inhibits host immune responses such as interferon and cytokine production.

4. Haemagglutinin (HA) - Part of the viral envelope, this protein binds to sialic acid receptors on the surface of host cells, initiating the infection process through receptor binding and triggering cell entry by clathrin-mediated endocytosis or other endocytic routes.

5. Nucleoprotein (NP) - The nucleoprotein associates with the viral RNA to form a ribonucleoprotein (RNP). The RNP binds to the RdRp complex and will bind to the host cell's importin-α transporting the viral RNA to the nucleus. In later stages will also bind to the newly-formed RNA particles to create the nascent virions' cores.

6. Neuraminidase (NA) - The neuraminidase is also a part of the viral envelope. It is responsible for two functions: enabling newly formed virions to escape cells and facilitating viral penetration through mucus layers.

7. M (MP) - This segment encodes 2 proteins: Matrix protein 1 (M1), responsible for forming the viral capsid, that supports the structure of the viral envelope and also assist the function of the NEP

protein, and Matrix protein 2 (M2), that forms a proton channel in the viral envelope, activated upon adhesion to a host cell promoting the uncoating of the viral particle, exposing its contents to the host cell's cytoplasm.

8. NS - This segment encodes 2 proteins: NS1 (non-structural protein 1) that counteracts the host's immune system and inhibits interferon production and NEP (Nuclear export protein formerly known as NS2) that cooperates with M1 to mediate the export of viral RNA copies from nucleus into cytoplasm in the late stage of viral replication.

IAV is also classified into subtypes (or genotypes) encoded by the HA and NA segments into a HxNy nomenclature. To date there were reported 16 different types of HA and 9 types of NA that to some extent can be host specific.

1.2.1.2 Ecology and evolution

As previously mentioned IAV has an ample range of hosts and through the course of its known evolution has been expanding its ecological breadth both in hosts as in geographic locations.

The virus in all its subtypes has been isolated in multiple avian and mammalian species, a far cry from the suspected original primary hosts: waterfowl (ducks, geese and swans).

IAV has also been isolated in all continents, including Antarctica, using migratory birds as an efficient carrier (Alvarez et al. [2025]).

The fast evolution and adaptability of IAV is supported by genetic drift and genetic shift.

Genetic drift refer to point mutations in the proteins encoded by the genomic segments that are responsible for viral adaptations, namely host adaptation, antiviral resistance and immune system evasion, whose importance in the context of viral surveillance cannot be understated.

As a segmented virus, IAV can also undergo reassortment, or genetic shift. Reassortment occurs when there is exchange of segments between two different IAV when the two aforementioned viruses co-infect a cell. This phenomenon allows for extremely fast evolution of viruses and has been correlated with flu pandemics over the course of the previous century (Alvarez et al. [2025],Peacock et al. [2024]).

1.2.1.3 Mechanism of infection

To understand host adaptation and its relation with viral evolution it is important to know how IAV infects and replicates.

1.2.1.3.1 Infection and cell invasion

IAV initiates infection by coming into contact with the host's mucosal surfaces, typically through aerosolized particles from respiratory secretions or faeces. At this interface, the virus encounters the mucus layer, that poses as the host's first line of defense. The viral neuraminidase (NA) is believed to play a critical role here by removing decoy receptors such as sialic acids present on mucins, cilia, and

the glycocalyx, thereby facilitating the binding of hemagglutinin (HA) to the actual cellular receptors (Matrosovich et al. [2004]).

The binding of HA to sialic acids on the host cell surface is vital in determining host specificity and adaptation. Avian-adapted IAV strains predominantly bind to α2,3-linked sialic acids, which are common in the avian respiratory epithelium. In contrast, mammalian-adapted strains bind preferentially to α2,6-linked sialic acids, which are prevalent in the upper respiratory tracts of mammals. Swine are unique in having both types of sialic acids throughout their respiratory tracts, making them especially susceptible to infections from both avian and mammalian strains and positioning them as potential intermediaries for viral reassortment and cross-species transmission (Matrosovich et al. [2004],Suttie et al. [2019],Alvarez et al. [2025]).

This difference in receptor specificity significantly influences viral pathogenesis, transmission, and host adaptation. In mammals, viruses that bind to α2,6-linked sialic acids tend to infect the upper respiratory tract, facilitating more efficient transmission through aerosolized droplets and causing generally milder disease. In contrast, viruses that bind to α2,3-linked sialic acids are more likely to infect the lower respiratory tract or conjunctiva. These infections are typically more severe, harder to transmit, and more likely to be eliminated through negative selection due to reduced host-to-host spread (Alvarez et al. [2025],Peacock et al. [2024]).

Although NA is also hypothesized to assist HA in facilitating cell invasion, this function has not yet been experimentally verified (Matrosovich et al. [2004]).

1.2.1.3.2 Uncoating and replication

After entry into the host cell, IAV relies on host-mediated processes for successful replication. The viral M2 ion channel acidifies the interior of the virion, triggering the release of viral ribonucleoproteins (vRNPs) into the cytoplasm. These vRNPs carry nuclear localization signals that hijack the host's importin-α/β system to enter the nucleus. Importin specificity is closely linked to host adaptation, as different IAV strains can vary in their compatibility with host importins, influencing species tropism (Dou et al. [2018]).

Inside the nucleus, replication begins with the synthesis of complementary RNA (cRNA) from vRNA templates, a process that does not require primers but depends on precise nucleotide pairing and efficient use of host-derived rNTPs. The viral polymerase then uses cRNA to generate new vRNA strands. The efficiency and fidelity of these steps can be influenced by host cell conditions and polymerase mutations, some of which are associated with adaptation to new hosts or resistance to antiviral drugs targeting polymerase function (Dou et al. [2018],Holmes et al. [2021],Mohapatra et al. [2023]).

Transcription of viral mRNA is primed through a host-dependent mechanism known as cap snatching, in which the viral polymerase steals 5′ caps from host mRNAs. This process is highly adapted to the host transcriptional machinery and relies on interaction with host RNA polymerase II. Mutations in polymerase subunits (PB2, PA) that enhance cap snatching efficiency or alter host specificity are determining factors in cross-species transmission and adaptation (Dou et al. [2018]). Additionally, changes in cap-binding or

endonuclease domains have been linked to reduced susceptibility to polymerase inhibitors (Holmes et al. [2021],Mohapatra et al. [2023]) .

1.2.1.3.3 Protein Biosynthesis and Transport

Influenza A virus (IAV) depends entirely on the host's cellular machinery for protein synthesis. After export of viral mRNAs from the nucleus, cytosolic ribosomes synthesize internal proteins, while ER-bound ribosomes produce envelope proteins like HA, NA, and M2. guided by signal sequences and transmembrane domains. These features, particularly in NA, can regulate membrane insertion and protein levels, playing a role in host adaptation and evasion of antiviral responses (Dou et al. [2018]).

Newly synthesized NP and polymerase proteins are transported back into the nucleus, where they assist in viral RNA replication and transcription. NS1 is among the earliest expressed proteins and acts to suppress host interferon responses, enhancing viral replication and possibly aiding viral mRNA export. NEP and M1 coordinate the export of vRNPs from the nucleus via the CRM1 pathway. (Dou et al. [2018]).

HA and NA undergo complex post-translational modifications, including glycosylation, folding, and oligomerization. An important step in HA processing is its cleavage: it is initially synthesized as an inactive precursor, HA0, which must be cleaved into HA1 and HA2 to become fusion-competent. This cleavage is a major determinant of virulence. In highly pathogenic avian strains, HA possesses a multibasic cleavage site that can be recognized and processed by the ubiquitous protease furin, enabling systemic infection and high virulence. In contrast, human and low-pathogenic avian strains have a monobasic cleavage site, which is cleaved by more restricted proteases such as TMPRSS2 and HAT, limiting viral activation to the respiratory tract (Dou et al. [2018]).

1.2.1.3.4 Assembly and Exocytosis

Viral assembly takes place at lipid rafts: cholesterol and sphingolipid-rich domains of the apical plasma membrane where components like HA and NA are directed via modifications in their transmembrane or cytoplasmic regions. M1 is thought to interact with the cytoplasmic tails of HA and NA or with the lipid environment itself, while M2 localizes at the edges of these rafts. vRNPs are recruited to budding sites through M1 interactions (Dou et al. [2018]).

The formation and release of virus particles require precise control over membrane curvature and scission. This is achieved through a combination of protein crowding (especially HA and NA), membrane-bending activities of M1 and M2, lipid composition, and possibly cytoskeletal support. M1 oligomerization at the membrane helps bend it into the virion shape, while M2 contributes to final detachment through its amphiphilic α-helix, inducing negative curvature at the bud neck (Dou et al. [2018]).

NA plays a central role in the release and transmission of IAV. It cleaves SA residues from both the host cell surface and the viral envelope, preventing newly formed virions from re-binding via HA. This enzymatic function is essential for viral egress and spread. While NA operates as a monomer at the catalytic level, its tetrameric structure and calcium binding increase efficiency. Differences in NA

activity and SA-binding preferences influences transmission routes and tissue tropism, directly affecting host adaptation (Dou et al. [2018]).

Interestingly, some NA variants display poor catalytic activity but retain SA-binding capacity, hinting at possible compensatory functions in receptor interaction or immune evasion. This functional variability in NA also has implications for antiviral resistance, especially against neuraminidase inhibitors, where structural flexibility may enable the virus to escape drug targeting while maintaining sufficient receptor-cleaving activity (Dou et al. [2018]).

1.2.1.4 Adaptative mutations

Understanding the IAV life cycle from infection to exocytosis is paramount to gain understanding of how different point mutations' loci promote different adaptations (e.g. HA mutations promoting binding to different SA; PB2 mutations increasing catalytic activity in mammalian or avian cells).

IAV mutations are myriad and authors such as Suttie et al. [2019] describe in thorough detail mutations with biological impact. Despite the extensive work characterizing mutations and their effects there are some important issues to note concerning analysis of mutations.

Firstly there is the issue of references. Reference sequences are of paramount importance for mutations can only be detected and described as variations of a reference, and unlike model organisms with well-established references like humans, mice or even SARS-CoV2, influenza presents several challenges. For example Suttie et al. [2019] use multiple references in their work. For internal proteins they map mutations against A/ goose/ Guangdong/ 1/ 1996(H5N1), whereas for Haemagglutinin they use A/ Aichi/ 2/ 1968(H3N2) and A/ Vietnam/ 1203/ 2004(H5N1) to describe H3 and H5 numbering of mutations. For NA they use N2 numbering based on A/ Tokyo/ 3/ 1967(H3N2). Different institutions, depending on their research contexts, may use different references, which can lead to different mutation profiles. Therefore, it is essential that reference sequences be uniform through mutation identification studies so that the data can be as consistent and replicable as possible.

Secondly, amongst the haemagglutinin variants there is the issue of numeration. Burke and Smith [2014] in their work on proposing a system of HA numbering schemes describe several challenges, namely the variation in signal peptides' cleavage sites, where most HA subtypes are cleaved on an aspartate residue 3 residues from a conserved cysteine residue, H3, H5 and H14 subtypes are cleaved at a glutamine residue, resulting in larger N-terminal regions; Subtype-specific and Clade-specific indels that disrupt straightforward residue comparison even within the same subtypes; and Inconsistencies in structural data where bioinformatic data can start with mature or immature sequences that alter the number of residues within each sequence.

Finally, mutation effects are described within certain proteins in certain contexts, such as particular subtypes and clades. Extrapolating mutation effects between different subtypes and even clades may not yield the expected biological implications, where experimental or clinical data is unavailable. In the context of the current H5Nx panzootic and its pandemic potential, a ECDC/EFSA consensus in 2025

(Alvarez et al. [2025]) defined the main mutations of interest that signal avian to mammal adaptations.

When IAV genomic vigilance is performed, a secondary, albeit important set of adaptations to be mindful of are those that confer resistance to antiviral drugs. Antiviral drugs act by inhibiting NA activity (Oseltamivir, Zanamivir), blocking polymerase binding sites (Baloxavir Marboxil, Favipiravir) or inhibiting M2 ion channels (Amantidine), and resistance to these drugs severely worsens the prognosis of patients hospitalised with severe disease (Holmes et al. [2021]). A list of the mutations coding host adaptations as well as antiviral resistances are summarised in Tables A.1 and A.2, presented in Appendix A.

1.2.1.5 Epidemiology

1.2.1.5.1 Seasonal Influenza vesus Non-seasonal Influenza

In order to have a full understanding of the relevance of Influenza surveillance and public health risks one must first understand the differences between seasonal and non-seasonal Influenza. Seasonal Influenza refers to the endemically circulating IAV and IBV subtypes to which the human population has been exposed and has some degree of immunological memory. The normal subtypes of IAV associated with seasonal infections in humans are H3N2 and H1N1, while the main seasonal subtypes of IBV are Yamagata and Victoria (Centers for Disease Control and Prevention [2024b]). These infections tend to be clinically mild and their prevalence increases during autumn and winter months, being almost completely absent during spring and summer months. Surveillance of these subtypes normally includes incidence and vaccine adequacy, as antigenic drift is quite likely and vaccines for one season can be of limited use in following seasons.

However, non-seasonal or emerging Influenza comprises all subtypes of IAV to which the population has had no previous exposure and thus no immunological memory. Although these viruses are named as non-seasonal, they appear to follow the seasonal trends seen in endemic IAV. These viruses normally arise from zoonotic infections that can have variable fitness levels in the human population. These infections tend to be moderate to severe; however, case fatality rates are thought to be overestimated due to lack of investigation and under-reporting of mild cases. Viruses with higher fitness for human-to-human transmission are also known to cause pandemics. The main subtypes described for these non-seasonal infections are H1N1, H1N2, H5N1, H5N6, H7N9 and H9N2. (Centers for Disease Control and Prevention [2024a])

1.2.1.5.2 Previous IAV pandemics

As mentioned above, when an IAV subtype crosses the species barrier with fitness higher than normal, allowing a new host infection and transmission to another host of the same species, conditions are met for a pandemic event (National Academies of Sciences, Engineering, and Medicine [2023]). Since the past century there have been four Influenza A pandemics:

- 1918 H1N1 “Spanish Flu”

- 1957 H2N2 “Asian Flu”
- 1968 H3N2 “Hong-Kong Flu”
- 2009 H1N1pdm09 “Swine Flu”

All of the aforementioned pandemic events have been linked to avian-human or avian-swine-human reassortment events (fig) (Alvarez et al. [2025],Peacock et al. [2024],Liang [2023]). The 1918 H1N1 pandemic was by far the pandemic that had the worst societal consequences, with death estimates ranging from 17 to 50 million people.

Pandemic strains, when adapted to the human population, integrate endemic seasonal influenza. Today, the H1N1 subtype originating from the H1N1pdm09 pandemic is still circulating in human and swine populations endemically.

1.2.1.5.3 Ongoing 2020 H5 Influenza Panzootic

Starting in autumn 2020 there was a rise through reassortment of a new clade of IAV H5N1, clade 2.3.4.4b, detected after infection of wild and domestic birds across central Asia and eastern Europe. It took one year for the virus to be isolated in North America, and by 2023 the virus was isolated on all continents with the exception of Australia. (Centers for Disease Control and Prevention (CDC) [2024],Peacock et al. [2024])

To date this panzootic has caused several wildlife mortality events, namely in wild birds and seals, and avian-to-mammal spillover events mainly in mink (Spain), dairy cattle (USA) (Peacock et al. [2024]) and cats (Poland) (Rabalski et al. [2023]). The dairy cattle cases are particularly noteworthy as bovines are not particularly common hosts of IAV, particularly H5 subtypes. Moreover, the main symptom was mastitis, an infection of the mammary gland. Milk from these cows transmitted the virus, proven by dead exposed cats at the dairy farm as well as reported conjunctivitis in farm workers. (Peacock et al. [2024])

Thus far, this outbreak has not caused widespread cases in humans, with confirmed reports of 80 cases with 7 fatalities, resulting in an estimated case fatality rate of 8%. Most cases were described in the US (Alvarez et al. [2025],Pan American Health Organization (PAHO) and World Health Organization (WHO) [2025]).

There were also 42 reported cases of H5N6 infections between November 2021 and October 2024, 41 reported in China and one in Laos. All the cases were linked to occupational exposure to poultry. Case-fatality rate was 41%. The HA clades responsible ranged from 2.3.4.4a to 2.3.4.4h (Alvarez et al. [2025],Sengkeopraseuth et al. [2022],Chen et al. [2022]).

The first cases in Portugal of H5N1 were detected from November 2021 up to March 2022 in 13 outbreaks in captive birds both domestic, or hobby flocks, and poultry and 6 identifications in wild bird flocks. On August 2022 there was a recurrence event reported in Évora, and by November 2023 there was another detection in Setúbal (World Organisation for Animal Health (WOAH) [2025]).

In July 2024 a mass outbreak of avian cases happened with wildlife detections in Faro, Aveiro, Coimbra and Viana do Castelo accompanied by a Poultry outbreak in Faro. There was a wildlife recurrence event in Faro in December of the same year (World Organisation for Animal Health (WOAH) [2025]).

By January 2025 there was another wildlife detection in Leiria and the largest outbreak to date in poultry farms in Sintra, Lisbon. To date there were no human cases reported (World Organisation for Animal Health (WOAH) [2025]).

Taking into consideration the nature of exposures, the lack of evidence of transmission between humans and the low number of human cases, the CDC classifies the risk of a H5N1 pandemic as moderate (U.S. Centers for Disease Control and Prevention [2025]).

1.2.2 Genomic Surveillance: The example of INSA - the INSaFLU-TELEVIR platform

Genomic surveillance refers to the use of genomic information for the epidemiological monitoring of pathogens. By enabling finer differentiation between strains of the same agent, genomic data allow for more precise characterization of outbreak dynamics and a deeper understanding of pathogen evolution.

While the potential of genomic surveillance is considerable, it also presents significant challenges, particularly regarding laboratory infrastructure, technical capacity, and coordination across surveillance systems. Key difficulties include ensuring adequate resources, preventing duplication of efforts, and achieving effective data harmonization. At the same time, these areas offer strategic opportunities for improvement, especially when tailored to the specific needs of each country. Developing a well-resourced national genomic surveillance strategy or action plan enables countries to set clear goals and priorities that both align with global objectives and address local realities (World Health Organization Global Genomic Surveillance, Research for Health (RFH) Team [2023]).

To support countries in overcoming these challenges, the World Health Organization (WHO) launched a 10-year global strategy in March 2022 aimed at strengthening genomic surveillance for pathogens with epidemic and pandemic potential. This strategy seeks to enhance the quality and timeliness of public health responses by scaling up surveillance capabilities at all levels—from local to global. A central focus is on building national capacity and promoting harmonized, pathogen-agnostic approaches that facilitate collaboration and data sharing across borders (World Health Organization Global Genomic Surveillance, Research for Health (RFH) Team [2023]). Figure (Figure 1.1) represents the main steps in genomic surveillance. The work in this project will cover mostly step 2d.

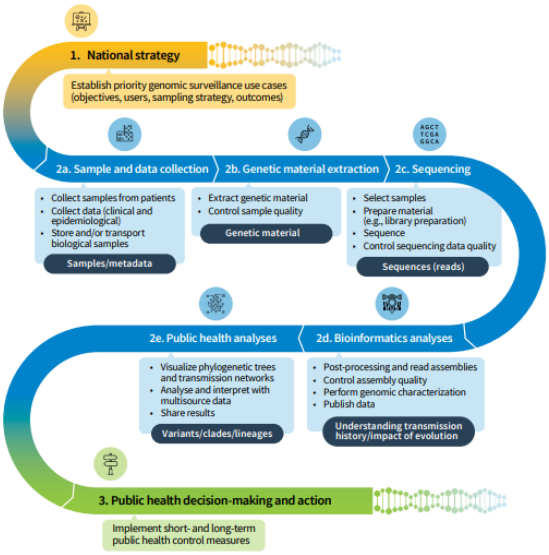

Figure 1.1: Genomic surveillance steps, from World Health Organization Global Genomic Surveillance, Research for Health (RFH) Team [2023]

One example of a surveillance framework is the one developed and employed by the Instituto Nacional de Saúde Doutor Ricardo Jorge (INSA) named INSaFLU-TELEVIR. INSaFLU-TELEVIR is an automated, open-source web-based bioinformatics suite initially developed only as INSaFLU for Influenza A and B genomic surveillance (Borges et al. [2018]), that provides a start to finish framework for genomic surveillance of Influenza virus.

On later updates, the platform evolved to be able to cope with other pathogens, such as Covid-19 (SARS-CoV-2), respiratory syncytial virus (RSV) or Dengue virus. The major upgrade to the INSaFLU was the integration of the TELEVIR platform used for viral metagenomics (Santos et al. [2024]). As the focus of this work lies on the surveillance routine, INSaFLU will be the relevant part of the compound framework, with Influenza as the main case study.

1.2.2.1 Sample collection to pathogen identification

At INSA every sample obtained from suspect clinical cases of respiratory infection is processed in the Reference Laboratory for Respiratory Viruses. After detection, the sample will be sequenced using high-throughput sequencing systems, either short-read (Illumina) or long-read (Oxford Nanopore). After obtaining the output reads (fastq.gz file format) files from the sequencer, the files will be uploaded into INSaFLU along with associated metadata.

The fastq will then be processed, namely quality control will be made using FastQC and read trimming will be performed using Trimmomatic. Afterwards the cleaned reads will be used to perform a draft

assembly with SPAdes. The draft assembly will then be screened with ABRIcate against two databases:

- "influenza_typing": for identification of the influenza A and B types, as well as all currently defined influenza A subtypes (18 hemagglutinin subtypes and 11 neuraminidase sub-types) and the two influenza B lineages (Yamagata and Victoria);
- "influenza_assign_segments2contigs" for automatic assignment of assembled contigs/nodes to each corresponding viral segment and a closely related reference influenza virus.

ABRIcate (Seemann [2023]) is a command-line bioinformatics tool used for screening contigs (typically from draft genome assemblies) for the presence of antimicrobial resistance (AMR) genes, virulence genes, plasmid replicons, or other gene sets using curated databases. It resorts to a BLAST algorithm to query sequences against the aforementioned curated databases. While the "influenza_typing" is highly robust for type/subtype identification (Borges et al. [2018]), the "influenza_assign_segments2contigs" uses a relatively limited of references that does not capture the diversity of non-seasonal IAV subvariants. This hindrance could be overcome with larger databases, however, with larger databases come longer query times that can lower performance.

1.2.2.2 Sequence identification frameworks

The BLAST algorithm is of central importance to bioinformatics in general and to this work in particular, therefore a more in-depth explanation of this algorithm is crucial.

BLAST stands for Basic Local Alignment Search Tool (Camacho et al. [2009]) and this program searches sequence databases for sequences similar to a query sequence. It is used for all biological sequences (proteins and nucleotide) and is based on an algorithm that takes a sequence and splits it into words or *w-mers*. These *w-mers* are substrings of size w, that take the default sizes of w=3 for proteins and w=11 for nucleotides. The *w-mers* are then aligned against database sequences and act as seeds for alignment elongation (Kellis et al. [2025]).

The elongation of a certain alignment is scored using substitution matrices, and continues until the alignment score falls below a certain threshold due to mismatch and gap penalties. After this step, the E-value is calculated, measuring the statistical significance of each alignment, and the alignments are sorted from more to less significant or more to less identical according to user input.

BLAST-based tools are quite accurate and reliable, but can be computationally taxing, particularly when handling several queries on very large databases.

Influenza viruses are among the most sequenced viruses in public databases, on NCBI virus (National Center for Biotechnology Information (NCBI) [1988]), by September 2024, for IAV alone there were 1117106 sequences available for download. In order to be able to capture the full knowledge-base of this virus, and to use such a wealth of data and metadata the computational effort required would be quite high. Naturally, most of these sequences would be of seasonal subtypes and one could even infer that many of these sequences would be quite homologous to one another, therefore two issues can arise from using a

full database: high redundancy can increase query times and computational loads both in processing as in storage and bias towards certain subtypes of IAV can lower the identification accuracy for rarer subtype samples such as non-seasonal avian ones. Enriching the databases with non-seasonal samples poses a different challenge. Other databases that retain this data such as GISAID (Khare et al. [2021]; Elbe and Buckland-Merrett [2017]; Shu and McCauley [2017]) are not open and thus, data obtained from them is not distributable without permission. This makes any tool whose accuracy is reliant on non-open data not completely distributable, even with open-source code.

Considering the drawbacks of redundancy and representation bias, a strategy involving homology clustering needs to be considered. Cd-hit (Li and Godzik [2006]) is a sequence clustering tool initially developed for protein clustering but also adapted for nucleic acid sequences (-est mode). This tool uses a short word filtering algorithm as the cornerstone of measuring sequence similarity. This algorithm divides a sequence into short substrings or *k-mers*, and does a pairwise comparison of two sequences' substrings.

The clustering process happens using a greedy incremental clustering algorithm that sorts sequences by size and the largest one will be the representative, or centroid of the first cluster. Then, pairwise comparisons are made against the representative, if the query sequence is within the identity threshold it will be clustered with the representative, otherwise it will become the representative of a different cluster. This process is repeated until the last sequence is clustered. Should the comparison between two sequences with short word filtering be impossible, an alignment step is performed.

When comparing two sequence datasets (the 2d mode) the query dataset sequences are compared with a subject dataset by sorting the datasets from longest to shortest sequences and applying the same algorithm stated on the previous paragraph, without an alignment confirmation step.

The main advantage of this tool is its time-efficiency when compared to BLAST, particularly when handling large datasets. This increased efficiency is achieved by two processes: firstly, clustering will remove the redundancies within a database, reducing the number of sequences to query; secondly, substring comparison is faster than aligning seeds and expanding the sequences into alignments. However, for this particular work, cd-hit has two major drawbacks. One is the inability to discriminate concatenated sequences; therefore, the identification will be made on the basis of the largest element of the concatenated sequence. The second drawback is the ability to cluster low-complexity sequences that are typically filtered by tools like BLAST, potentially leading to inaccurate assignments. As such, sequence length thresholds need to be enforced prior to the cd-hit analysis.

1.2.2.3 Assembly and Reference Selection

The next module of INSaFLU takes as input the treated reads and uses an adapted version of snippy (Seemann [2025]), a tool that comprises samtools (Li et al. [2009]), Burrows-Wheeler Aligner (BWA) (Li and Durbin [2009]), SnpEff (Cingolani et al. [2012]) and vcftools (Danecek et al. [2011]), with the objectives of generating consensus sequences, and variant calling and annotation.

Snippy uses a reference based approach to consensus creation. Consensus sequences are generated by alignments, with user specified quality criterion, of reads against a reference. References can be obtained through the INSaFLU's reference database or provided directly by the user manually.

In regards to non-seasonal IAV sequences that have a larger array of subtypes, frequent reassortment and more variability, even within subtypes, reference selection is a challenge. To respond to this challenge the Iterative Refinement Meta-Assembler or IRMA(Shepard et al. [2016]) was developed.

Recently IRMA, that also plays an important part on the CDC's surveillance protocol (Shepard et al. [2016]), was also added to the platform to generate consensus sequences on highly variable viruses such as Influenza. IRMA matches reads to a consensus template within the internal IRMA database, but it also does "on-the-fly" iterative reference editing, increasing horizontal and vertical coverage, identifying low-frequency variants, and generating plurality consensus without the need for further reference sequence additions. Therefore, IRMA should be theoretically superior to the INSaFLU-Snippy approach when using less ideal references in highly variable viruses.

One should note that even with the use of IRMA, quality references are still necessary for bioinformatics analysis after consensus generations, as will be seen in section 1.2.2.4

1.2.2.4 Post-Assembly Analysis

After consensus generation, INSaFLU offers coverage analysis, alignment with MAUVE and MAFFT, phylogenetic trees, clade analysis with NextClade,inference of mutations and their potential consequences and recently integrated H5N1-specific mutation analysis with FluMut.

Nextclade was developed by the Nextstrain team as a set of functionalities to streamline initial analysis of viral sequences, initially focusing on SARS-CoV2 but now expanded to a multitude of virus families (Aksamentov et al. [2021]).

Nextclade provides an integrated platform for clade assignment, mutation identification, quality control, and sequence alignment. It compares input viral genome sequences against a curated reference, calling nucleotide and amino acid mutations, and detecting insertions, deletions, and other relevant genomic features. Sequences are then placed onto a phylogenetic tree and assigned to clades based on predefined mutation profiles, enabling real-time monitoring of viral evolution and variant tracking. In addition to its web interface, which facilitates user-friendly, client-side analyses, Nextclade also offers a command-line tool for batch processing and integration into automated pipelines, making it a versatile choice for genomic surveillance workflows.

Performing a complete phylogenetic analysis of one sample is quite a long and taxing process envolving reference sequences, tree creation and analysis of sample positioning within the tree. NextClade in a fast and streamlined way positions samples within a phylogenetic clade allowing for fast lineage characterization without the need for tree construction.

FluMut is a python-based application developed by the Istituto Zooprofilattico Sperimentale delle Venezie with the aim of reporting mutations that confer different host adaptation profiles, antiviral resis-

tance and virulence in H5N1 IAV (Giussani and Sartori [2024]). This program sorts sequences in their segments, by their fasta headers using regular expressions, then uses biopython functions to do the translation of nucleotide sequences and AlignIO to perform pairwise alignments to references present in their database. The FluMut database (version 6.3) is built in SQL and integrated within the main program using sqlite. FluMut outputs either an excel spreadsheet or machine-readable .tsv files comprising a mutation matrix for reported positions of interest, a tabular file containing every marker of interest found for each sequence and a table of bibliography associated with each marker found.

This tool tries to automate the traditional human curated Single Nucleotide Variant (SNV) calling methods. These methods require alignment of sample sequences against known quality references to be able to detect the nucleotide differences.

Despite being quite easy to use and computationally fast, FluMut has important drawbacks such as the lack of validation for other HxNy subtypes, the incompleteness of the database, and the unspecified numbering scheme that makes the validation for other subtypes more challenging. FluMut also lacks the output of point mutations that are not reported in the literature for in-depth analysis.

1.3 Problem

Reference-based consensus generation in INSaFLU depends on a good reference. Since IAV are segmented viruses, a reference sequence is needed for every genomic segment. This problem is compounded by the fact that segmented viruses can reassort which makes finding the best combination of segment references a challenging task. This is particularly critical for H5N1 avian influenza due to its variability and recombination. Rapidly finding a good H5N1 reference for consensus generation is currently a gap in INSaFLU that this project aims to fill.

Moreover, after the generation of a good consensus, there is the need for a rapid characterization in terms of host, zoonotic potential and resistance to antivirals. INSaFLU already provides some functionality to these ends but there is still much need of an integrated solution that uses in an efficient way all the information available in resources such as NCBI. Such an automated solution should also deal with the problem of representation bias, stemming from the under-representation of non-seasonal IAV compared to seasonal IAV; and the problem of sequence redundancy. These problems hinder performance when dealing with large databases.

To help solve this problem this project suggests a homology cluster-based approach that has two main advantages: reduction of database redundancies (also reducing the size of the database), and possibly a concomitant reduction of representation bias. Assuming the most studied subtypes have less variability amongst samples, other subtypes will be enriched within the databases.

Aggregation of sequence metadata inside a cluster unit will offer another advantage, allowing for a more robust identification and characterization of similar sequences. Sequences clustered within the same cluster have likelihoods of belonging to certain subtypes, certain host taxonomic groups and certain geographic regions that if well delimited may offer the user a faster way to characterise each sample.

Finally, there is the problem of sequence feature detection and annotation, such as mutation analysis and reassortment inference. This work aims only to select and sort validated sample subtypes to the appropriate validated tools, like NextClade and FluMut as a default behaviour. Unfortunately, lack of validation of CLI tools for all subtypes, absence of reference sequence uniformization and differences of nomenclature make this problem an impossible one to tackle fully on this particular project.

1.4 Objectives

As stated above, this work proposes to fill some gaps in existing systems and solve some current issues in the industry:

1. The first goal of this project is the development of an automated system that accurately identifies in an IAV sequencing sample (consensus/contig), that should comprise up to eight sequences, each sequence's segment, close representative sequences and possible host origin as well as the sample's subtype/genotype.

2. The second objective of this work is to characterise, for every identified segment, the mutations of biological interest if possible. This system also needs to be automated and integrated with the previous system.

3. The final goal is integration of the data obtained from the previous steps into a user-readable output in addition to tabular machine-readable outputs.

Chapter 2

Materials and Methods

The pipeline created for this project was named AFluID (Automatic Influenza Identification pipeline).

To implement the data analysis pipeline, data from multiple sources was obtained and curated, and a variety of bioinformatics tools were considered.

There were also decisions about programming languages, libraries, and programming logic that impacted the development of the AFluID pipeline. These will also be discussed in more detail in the following sections.

2.1 Data

In order to enable robust identification of IAV sequences, a comprehensive knowledge-base was needed, as well as good validation datasets.

As AFluID annotates IAV sequences and IAV sequence clusters with sequence metadata, correct identification of segment and genotype is paramount for posterior bioinformatics analysis.

2.1.1 Influenza A dataset

2.1.1.1 Sequence dataset

A total of 748096 sequences were downloaded from the NCBI virus database on the 18th of September 2024, comprising complete assemblies of Influenza A Virus. This dataset was used for BLAST database and cd-hit-est cluster database creation.

2.1.1.2 Sequence metadata

In order to annotate reports and do an inventory on training dataset sequences a metadata table was also downloaded from the NCBI virus database on the 18th of September 2024 comprising 1117106 entries with the following information:

- Accession number
- Organism name
- Genbank refseq
- Assembly
- Release date
- Isolate
- Species
- Length
- Nucleotide completeness
- Genotype
- Segment
- Country
- Host
- Collection date

The metadata was then filtered to include only the entries for the 748096 sequences present within the sequence database. Clustering the sequence database using cd-hit yielded a cluster database with 59876 clusters that would be annotated with the sequence metadata using aggregate dictionaries representing each segment, subtypes and host ranges for each cluster. All of these annotations are performed automatically and are made possible because of data connections between each database. The data connection diagram in Figure 2.1 illustrates the relationships between sequences and their respective metadata.

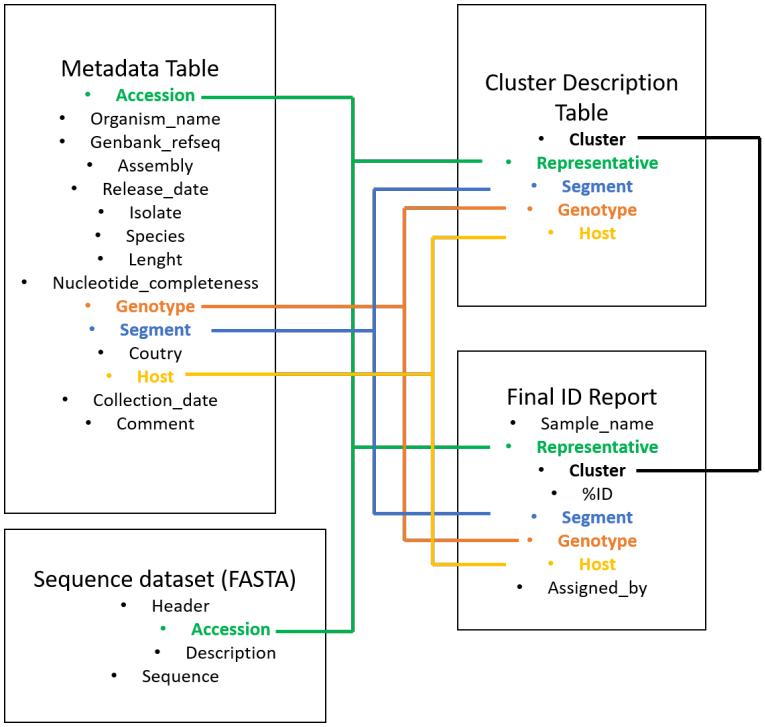

Figure 2.1: Data connection diagram within the pipeline

2.1.1.3 Manual curation and Metadata updates

It was very clear from the first test-runs of the pipeline that incomplete and incorrect metadata would be a major hindrance to the system's performance. As seen in Section 2.1 the output of the pipeline is heavily dependant on the correctness and completeness of the metadata. The presence of incomplete host, genotype and segment fields was addressed through a staging of different approaches.

1. Text mining using string manipulation and regular expressions of the "Organism name" field to obtain genotype and host information;

2. Text mining using substring matching and regular expressions of the fasta headers to obtain segment information;

3. When the previous approaches failed to complete the fields BLAST was performed to identify segments and genotypes closest to the query sequences.

Completion of the information for all of the 748096 sequences that created the cd-hit cluster and BLAST databases was possible, however upon the definition of clusters some problems have arisen such as segment-ambiguous clusters (a cluster comprised of sequences from two different genomic segments)

or genotype-ambiguous haemagglutinin and neuraminidase segments (example: a haemagglutinin segment with both H3 and H1 genotypes, or a neuraminidase cluster with both N1 and N2 genotypes). The aforementioned data had to be manually curated. Formatting errors, such as extra commas in some lines of the CSV file were identified and corrected. Incorrect annotations on NCBI and low-quality assemblies were also addressed when found, through online BLAST and manual inspection of every suspicious sequence. In the end, two full assemblies of 8 sequences (segments 1-8) were removed as well as 23 sequences of NS that were clustered within a MP cluster.

2.1.2 Validation datasets

2.1.2.1 NCBI randomly selected IAV sequences

To perform initial debugging, fine-tuning and validation of pipeline routines a subset of 719 unique sequences randomly selected from NCBI virus database was obtained using the same queries than the ones used to obtain the main dataset of sequences. This dataset also includes sequences that don't originate from full assemblies, therefore will not be present within the original database.

2.1.2.2 Influenza B, C and Ebolavirus datasets

In order to test the pipeline's power to discriminate between different Influenza viruses and between more closely related RNA viruses, three datasets were obtained on 29th of October 2024 from NCBI virus using the queries "Influenza B virus, taxid:11520", "Influenza C virus, taxid:11552" and "Ebolavirus, taxid:186536". For the influenza virus sequences 100 random sequences of influenza C and 300 random sequences of influenza B were downloaded. As for Ebolavirus only 14 sequences were downloaded when filtered by sequence length from 850 to 2200 base pairs.

2.1.2.3 Avian Influenza External Quality Assessment datasets

As an official reference institute for respiratory disease surveillance, INSA participates in regular external quality assessments (EQAs). Two datasets of 80 and 138 sequences from Avian Influenza EQAs were provided to validate the pipeline against surveillance samples already characterized. The drafted contigs resulting from the assembly process were also provided to ascertain the pipeline's identification power in unrefined NGS data.

2.1.2.4 GISAID datasets

To validate the pipeline using datasets from different databases and to incorporate greater variability across genotypes, we used a subset of 8457 Influenza A sequences enriched for avian samples, obtained from GISAID.

From the previous dataset it was apparent that environmental samples were recognized with low identity (<90%), therefore a dataset with 23184 environmental sequences was downloaded from GISAID.

2.1.2.5 Portuguese Avian Influenza Outbreak Samples

INSA collaborated with INIAV-IP in the first detections of H5N1 IAV since 2021. Taking adavantage from this close collaboration, 28 samples of avian influenza NGS data were provided (covering different hosts) for AFluiD testing and validation. These included the NGS reads (fastq.gz) and the respective curated consensus sequences, originally obtained using INSAFLU after manual reference selection and consensus curation. The references of these sequences, found in GISAID will also be presented in appendix table A.7. This particular data set aims not only to test AFluID's precision and flexibility in dealing with real-world surveillance data, but also the quality of reference gathering to perform consensus curation. This dataset also aims to compare consensus generated by IRMA (version 1.3.0), a non-reference based assembly tool with the ones generated using reference based methods.

2.2 Bioinformatics tools

AFluID uses several bioinformatic tools to identify and annotation of IAV NGS data.

2.2.1 cd-hit

Cd-hit (version 4.7) Li and Godzik [2006] was used though the CLI application. To run in a CLI the following command was used to create the clustering file with one dataset:

```
cd-hit-est -i SEQUENCE_FILE -o OUTPUT_PATH -c ID_THRESHOLD (0.99) -M
    MEMORY_ALLOCATION (5000)
```

The next command was used to compare two different datasets (database vs sample):

```
cd-hit-est-2d -i CLUSTER_DB -i2 SAMPLE_FILE -o OUTPUT_PATH -c ID_THRESHOLD
    (0.99)
```

Additional commands accepted for the cd-hit call are made available in Table A.3 of appendix A.

2.2.2 Basic Local Alignment Search Tool - BLAST

BLAST (Camacho et al. [2009]) was used as the Web application for final confirmations and curation, and the CLI application (version 2.16.0+) supports the pipeline workflow.

In the context of this work, BLAST databases were created using the following command:

```
makeblastdb -in INPUT_FILE_PATH -dbtype nucl -out OUTPUT_DB_PATH
```

As for the comparisons between query files and the BLAST databases the command used was the following:

```
blastn -db BLAST_DB_PATH -query SAMPLE_PATH -out OUTPUT_FILE_PATH -outfmt
    OUTFMT (6) -num_threads 4 -max_target_seqs 7
```

Additional commands accepted for the blastn call are made available in Table A.4 of appendix A.

As the BLAST search would output the seven best matches per each query sequence, the matches were filtered by the best E-value and then Identity at thresholds configured within the pipeline.

2.2.3 FluMut

FluMut (version 0.6.3) (Giussani and Sartori [2024]) was used with automatic updates of the FluMut database enabled by default and only for H5Nx sample typing enabled by default. For the main part of this work a previous step of conforming sequence headers to respect the flumut input regular expression was implemented after the main identification component of the pipeline.

The CLI command used to run flumut was as follows:

```
flumut -i SAMPLE_PATH -m SAMPLE_markers.tsv -M SAMPLE_mutations.tsv -l
    SAMPLE_literature.tsv -n "(.+)\|(.+)"
```

Additional commands accepted for the flumut call are made available in Table A.5 of appendix A.

2.2.4 Nextclade

Nextclade (Aksamentov et al. [2021]), in the context of this work, was used for the assignment of the haemagglutinin clade, for the H1, H3, and H5 subtypes.

The command used in this work to run Nextclade is the following, where ”BUILD” corresponds to the phylogeny build used to map the sequences, ”SAMPLE_H-GENOTYPE.fasta” corresponds to the H-GENOTYPE haemagglutinin sequences to analyse and ”SAMPLE_H-GENOTYPE_nextclade.tsv” corresponds to the output tabular file.

```
nextclade run -d BUILD --output-tsv SAMPLE_H-GENOTYPE_nextclade.tsv
    SAMPLE_H_GENOTYPE.fasta
```

Additional commands accepted for the Nextclade call are made available in Table A.6 of appendix A.

2.2.5 Other bioinformatics tools - MAFFT and MEGA

During evaluation of assembly workflows there was the need to perform alignments and calculate pairwise distances between consensus sequences, for that end and not for integration within AFluID, MAFFT (version 7.526) (Katoh and Standley [2013]) and MEGA (version 12.0) (Kumar et al. [2024]) were used.

2.3 Development and workflow

2.3.1 Environment and Dependencies

Dependencies and their specific versions are paramount for functionality and reproducibility. Python libraries, for example, are subject to frequent updates, but sometimes asynchronously between different libraries, which can cause temporary or permanent loss of support between libraries. For example, Python 3.12 lost support for the NumPy 1.26.4 longdouble type; Pandas 3.0 will experience changes in programming logic, namely, when slicing and reassigning values within dataframes.

The algorithms of bioinformatic tools, namely BLAST, suffer minor changes from version to version that can affect search results. Therefore, it is vital that the algorithm version be fixed to ensure reproducibility.

As such, a decision was made to use a conda environment with a YAML configuration file to create a similar environment every time, guaranteeing that every tool and dependency is adequately installed and isolating the pipeline from other system dependencies that could create conflicts.

2.3.2 Paths

In order to organise the pipeline's workflow in regards to data, metadata, samples, run files, reports, and references, the following directory tree was adopted (Figure 2.2).

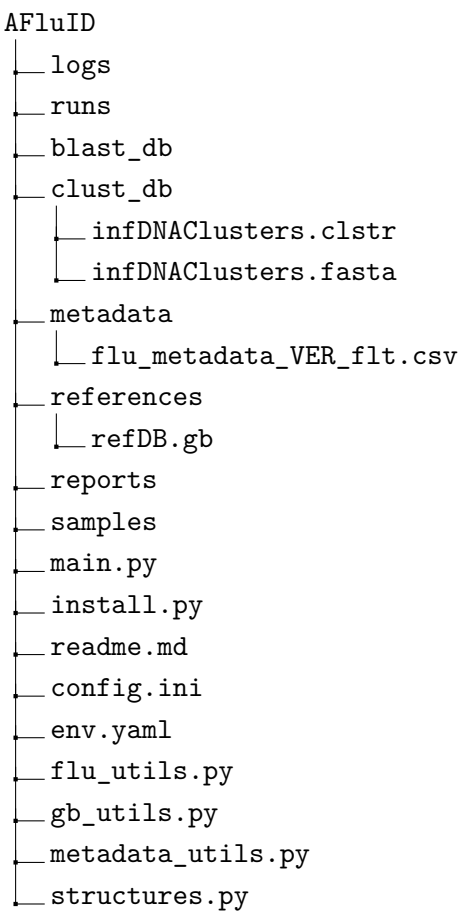

Figure 2.2: Pipeline directory tree displaying directories and some of the most relevant files.

2.3.3 Pipeline Steps

The pipeline comprises the following steps:

1. Check paths, modules and libraries;
2. Preprocess the NGS data file in order to remove illegal characters such as non-IUPAC characters in sequences and metacharacters;
3. Remove sequences whose length falls outside certain thresholds (recommended or user-specified);
4. Remove long FASTA headers, remapping them after the analysis is complete.
5. Perform cd-hit-est-2d analysis against a local cluster database within user-defined identity thresholds (this work was performed using 99% identity);
6. Parse the cd-hit outputs to find assigned and unassigned samples;
7. Run a BLAST search against the cluster representatives, parsing the BLAST output and output the best assignment hit down to 90% identity;
8. Run a BLAST search against the full sequence database of any unassigned sample on the previous step;
9. Generate an identification report, reporting all sequences identified, their closest reference, matching identity, segment, subtype, host, geographic location and assignment method;
10. Parse the identification report in order to redirect each sequence to further bioinformatics analysis (clade assignments, mutation analysis, reference gathering)

2.3.4 Python modules and libraries

Python is an interpreted, dynamic-typed, high-level language applicable to different programming paradigms such as functional programming or object-oriented programming, renowned for its simplicity and versatility. As such, Python version 3.10.14 was chosen as the main language to develop this pipeline (Python Software Foundation [2025]). Base Python is enhanced by several libraries and modules that offer new classes and methods, expanding the language's functionalities. Several modules were used for this work.

2.3.4.1 os, glob, re, copy, subprocess, json and sys modules

The following modules are part of the core python installation and were instrumental for the functioning of this pipeline.

The os module in Python allows users to access features that depend on the operating system. This lets programmers interact with the system in a way that works on different platforms. It includes functions for handling file and directory paths, managing the file system, and getting or setting environment variables. With the os module, users can create or delete directories, list directory contents, and manage file paths across various systems using os.path. It also offers tools to run system commands and access information about processes, making it a useful resource for scripting and automation at the system level (Python Software Foundation [2025]).

Although the os module provides a rather complete range of functionalities, sometimes during the processing of files and directories, some flexibility was required, and hence there was the association of the glob module. The glob module in Python helps with Unix-style pathname pattern expansion. It allows users to search for files and directories that fit specific patterns. The module includes functions like glob() and iglob() that return lists or iterators of pathnames matching a wildcard pattern. For example, the user can use *.txt to find all text files or data/*.csv to locate CSV files in a directory. This module is especially helpful for batch file operations, like reading multiple files from a directory, without having to hardcode filenames. It supports common wildcards such as *, ?, and [], making file matching quick and easy in scripts.

The sys module in Python gives access to variables and functions that interact directly with the Python interpreter. It allows developers to change the runtime environment. For example, they can read command-line arguments using sys.argv, exit the program with sys.exit(), or access the standard input, output, and error streams (sys.stdin, sys.stdout, sys.stderr). It also includes details about the Python version, platform, and the module search path (sys.path). This makes it important for tasks like debugging, configuring the environment, and building command-line interfaces. The sys module is essential for writing flexible and portable Python programs.

When running CLI commands namely when stdout capture into an object is required the Subprocess module is preferred to the os.system() function. The subprocess module in Python lets users start new processes, connect to their input, output, and error streams, and get their return codes. It offers a useful way to run and manage external commands and scripts from within the user's Python code. Functions like subprocess.run(), subprocess.Popen(), and subprocess.call() are commonly used to execute shell commands, capture output, and interact with the process in real time. The module supports features such as timeouts, input piping, and environment setup. This makes the subprocess module important for automating tasks that involve system commands or integrating external tools into Python applications.

During this work, it was evident that merely using string comparison would not suffice for the capture of more intricate and variable patterns. Therefore the re module was used. The re module in Python helps users work with regular expressions. Using a String search tool, users can search, match, and manipulate strings in a powerful and flexible way. Users can define patterns using a specific syntax and use functions like re.search(), re.match(), re.findall(), and re.sub() to find, extract, or replace parts of strings based on those patterns. Regular expressions are particularly helpful for tasks such as input validation, data parsing, and text processing. The module also lets users compile regular expressions with re.compile() for better

performance and reusability.

Due to the ample use of complex and nested mutable data structures the copy module had to be used to allow for duplication of these structures. The copy module in Python offers functions for duplicating objects, allowing for shallow and deep copying. A shallow copy, made with copy.copy(), creates a new object but includes references to the original objects inside it, which means that nested objects are not copied. On the other hand, a deep copy, made with copy.deepcopy(), copies all objects recursively, creating completely independent copies of the object and everything inside it.

This work also required manipulation of JavaScript Object Notation (JSON) data, therefore the json module was used to handle the I/O of these particular files. The json module in Python offers tools for handling JSON (JavaScript Object Notation) data for it enables easy conversion between Python objects and JSON strings. One can use functions like json.dumps() to convert Python data into JSON or json.loads() helps turn JSON strings back into Python objects. The module can also read from and write to files with json.dump() and json.load(). It works with standard data types like dictionaries, lists, strings, numbers, and booleans. This module is commonly used for web APIs, configuration files, and data storage in a format that is both user-readable and portable across different programming languages.

2.3.4.2 argparse and configparser modules

The following modules, that are also a part of the python core installation, allow for change and customization of pipeline behaviour, both for default runs as in individual runs. (Python Software Foundation [2025])

The configparser module enables parsing of .ini configuration files and conversion of their values into a python dictionary. This .ini file will set the path variables as well as the pipeline's default behaviour.

The argparse module comes as a more complete solution compared to the argument vectors of the sys module (sys.argv()). The module allows for the input of named arguments upon the pipeline's call that will dictate the mode of execution as well as override defaults stated within the configuration file. It also generates an automatic help output when an input error occurs or when prompted by the user.

The combination of these two strategies gives the user the option to change defaults or alter behaviour of the pipeline at will, or even customise defaults for the individual's needs.

2.3.4.3 NumPy

During the development of this work, the need to parse and handle tabular data efficiently led to a requirement of use of the Pandas and NumPy libraries.

NumPy ("Numerical Python") is the fundamental package for scientific computing in Python. It introduces the ndarray, a powerful multidimensional array object that enables fast, efficient operations on large datasets. Arrays in NumPy are homogeneous: all elements share the same data type and occupy contiguous memory, allowing highly optimised performance through C-backed calculations rather than

Python-level loops. Because of this design, operations like slicing, computing with mathematical functions, linear algebra, statistics, and Fourier transforms can be executed quickly and with minimal syntax overhead (Harris et al. [2020]; NumPy Developers [2025]).

Beyond raw arrays, NumPy offers an extensive library of tools, including universal functions (ufuncs) for elementwise operations that support broadcasting, type casting, and output control. It also features routines for shape manipulation, sorting, I/O, random simulations, and linear algebra, making it an all-in-one hub for data processing.

2.3.4.4 Pandas

As stated previously the wealth of tabular files as inputs and outputs required the use of the NumPy and Pandas libraries, excellent for handling these files.

Pandas is a powerful open-source Python library designed for data manipulation and analysis, building directly on the strengths of NumPy. At its core are two fundamental data structures: the Series, a one-dimensional labeled array capable of holding diverse data types (integers, floats, strings, Python objects, etc.), and the DataFrame, a two-dimensional, size-mutable, potentially heterogeneous table with labeled axes. These structures support intelligent label-based alignment, automatic handling of missing data (e.g., NaN), and seamless integration with NumPy operations—making them ideal for tasks like slicing, aggregation, summarization, and time-series analysis. (McKinney [2010]; NumFOCUS Inc. [2025])

Beyond its data containers, pandas offers a comprehensive suite of tools covering I/O, reshaping, grouping, time-series functionality, and more—all designed for efficiency and developer productivity. The DataFrame API provides flexible operations such as database-style joins (merge), reshaping and pivot tables, and statistical aggregations (groupby, describe).

2.3.4.5 Biopython

This work is centered around a large volume of biological data. Therefore, integration of Biopython for its capability for handling this data in its multiple formats was essential, particularly for the later stages of the pipeline's implementation.

Biopython is a comprehensive, open-source toolkit for computational molecular biology, developed by an international community of contributors. It provides Python-based abstractions for biological sequences, such as the Seq and SeqRecord classes, enabling content-aware operations like slicing, transcription, translation, and motif analysis. Beyond raw sequence manipulation, Biopython offers parsers for a wide range of common bioinformatics file formats, including BLAST, FASTA, GenBank, ClustalW, and PubMed records, as well as streamlined support for accessing online repositories like NCBI and ExPASy via modules such as Bio.SeqIO and Bio.Entrez. This makes it ideal for workflows that integrate sequence parsing, database access, and format conversion. (Cock et al. [2009, 2025])

2.3.4.6 Custom modules

Whenever custom functionalities needed to be created that were not within the aforementioned libraries and modules, they were created and stored within the flu_utils.py (general sequence, metadata, binary storage and parsing functionalities), gb_utils.py (genbank and entrez-related functionalities), metadata_utils.py (classes to handle tabular formats) and structures.py (lookup and template data structures).

2.3.5 Pipeline Installation

The pipeline installation is performed by running the install.py script with the following command:

```
python3 install.py config.ini
```

Repository cloning, environment creation, and sequence data and metadata download instructions are available on the AFluID GitHub repository. It is also advisable to review the config.ini file to set up the Entrez email and the conda environment path. The user can make additional customisations of paths and filenames, though care is advised as breakdown of the pipeline is possible with incorrect setup. The installation process will create paths and databases necessary for the main pipeline.

2.3.6 Pipeline Arguments

Through argparse, it is possible to modify the pipeline's default behaviour bypassing some of the config.ini parameters. The pipeline runs in two main modes: the contig mode and the consensus mode. The contig mode, by default, runs the identification part of the pipeline and fetches references for every assigned sequence. The consensus mode runs the identification part of the pipeline and by default, redirects H5Nx identified sequences through FluMut and H1, H3 and H5 sequences through NextClade using the appropriate builds.

The arguments also enable or disable single sample mode. In this mode that corresponds to one sample per fasta file, the pipeline gather subtyping and sequence length information along with the identification. Single sample is the default behaviour for the consensus mode, and is disabled for the default contig mode runs. Single sample mode should be disabled for multi-sample fasta files, where length capturing and multi-sample subtyping informations are not currently supported.

The user may also be able to change minimum and maximum sequence length thresholds without changing the config.ini using the -ml and -Ml arguments.

It is possible to force FluMut and reference gathering on sequences where there are not enabled by default as well as disable FluMutDB updates. It's also possible to run the pipeline with other config.ini files.

The simplest pipeline call (consensus mode, default config.ini, single sample mode without any additional forces) is done using the following command.

```
python3 main.py -f SAMPLE_FASTA.fasta -m consensus
```

For more command information the user can use the -h command to call the argument documentation.

2.3.7 Pipeline Workflow

As stated before, the pipeline works in two modes, consensus and contig, but as of the moment of writing the modes only differ in the later stages of the workflow. The workflow of the pipeline is schematised in Figure 2.3.

The pipeline begins by checking paths and metadata files, loading run parameters and overriding the config.ini ones with the optional arguments should they exist. A master dictionary will also be loaded, and will store sequence, assignment and error information.

Then it will parse the sample file and creating two dictionaries: a sequence dictionary that will store a coded header with Seq_X where X is the number of sequence in the file, and a mappings dictionary that will store the relation between generated headers and true headers. These dictionaries will only store sequences whose length is within the permitted range and all metacharacters and non-IUPAC sequence characters will be removed.

The sequence dictionary will then generate a .fasta file that will serve as input to cd-hit. The cd-hit .clstr file will then be parsed to retrieve the assignments that will in turn generate the clustering report.

The clustering report will then be parsed, not only storing the assignments and unassigned samples in the master dictionary, but also generating a to_blast.fasta file with the unassigned sequences and their headers. This file will then be subjected to BLAST against the cluster representative database. The outputs of BLAST will be parsed into a dictionary and all of the unassigned sequences, or sequences assigned with less than 90% identity, will be stored into a to_reblast.fasta file and subjected to a second round of BLAST, this time against the full sequence database.

After all of the search tool runs, the assigned samples will generate an annotated identification report, remapped for the true headers of each sequence, that will in turn be parsed by a redirector function. This redirector function will sort every sequence for appropriate secondary bioinformatics analysis, for example, H5Nx subtyped sequences to FluMut, H3, H5 and H1 HA sequences to NextClade, contig sequences for reference gathering.

Regarding FluMut, unless forced by the user, it will analyse the consensus sequences identified as originating from a H5Nx sample. This will require generation of a .fasta file with the sequences to analyse and the conformation of the sequence headers to the FluMut regular expression to enable segment identification. This will output three .tsv reports, one for mutations (the mutation matrix by locus), one for markers (as the markers from identified mutations within the FluMutDB), and one for literature (indicating the sources of each marker effect description).

As for NextClade a clade assignemnt file will be outputted for every subtype of HA present in the .fasta file. Each subtype of HA sequences will in turn generate their own .fasta file that will be processed by NextClade with the appropriate builds. The phylogenetic builds used were chosen to be as broad as possible.

Finally, for reference gathering sequences will be mapped against their representative reference accession numbers. The genbank reference database will then be opened and a lookup table generated. Should the representative accession numbers be present at the lookup table, those references will be automatically gathered from the file. Any missing reference will be obtained from NCBI via Entrez. The references will be compiled into one .gb file and one .fasta file per sample analysed. This functionality by default also concatenates every missing reference obtained to the genbank reference database file for faster reference gathering posteriorly

2.3.8 Final Report implementation

After all the tabular outputs are generated, at the time of writing, a HTML final report was implemented for single-sample runs. This report unifies all data from the tabular inputs into a unified file , that allows for roll-up and drill-down of host and geographic metadata.

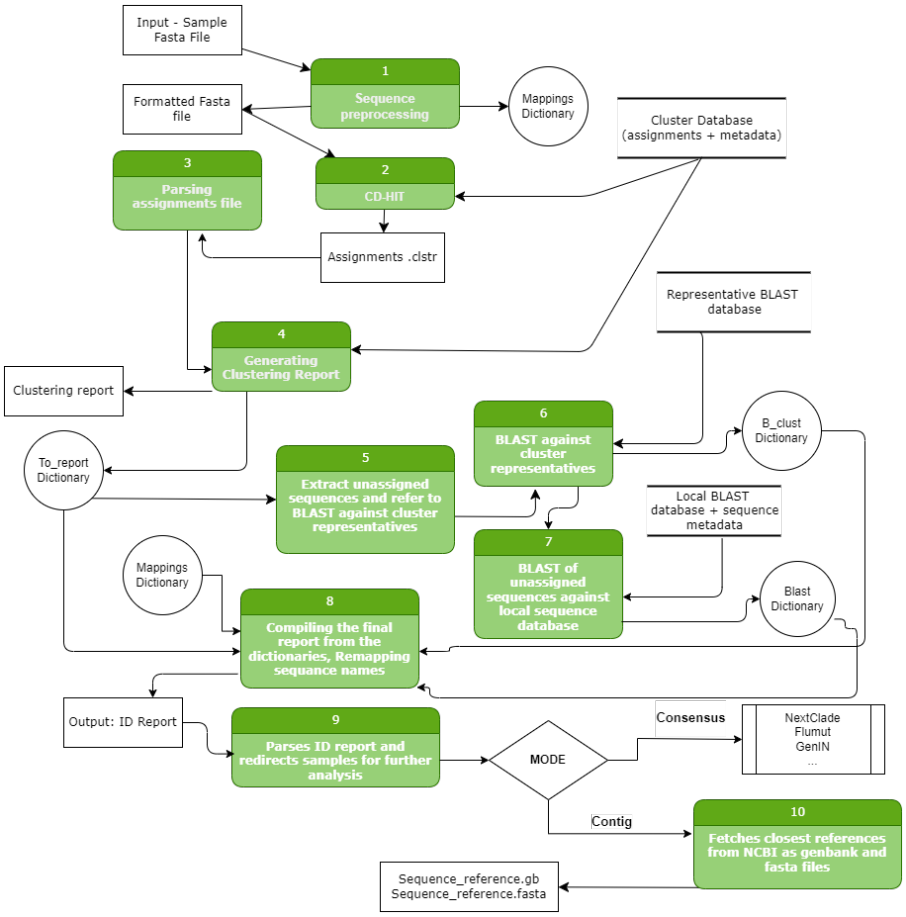

Figure 2.3: AFluID workflow diagram. The white boxes represent I/O files, the white circles represent python objects and the green boxes represent pipeline steps present in the main.py main() function.

2.4 Data and code availability

The latest version, including manual curation, of the sequences and sequence metadata used for this work is available in Zenodo at the address https://zenodo.org/records/15358183, and the code and documentation is available in a GitHub repository at https://github.com/jopereira88/AFluID. (Pereira [2025, 2024])

Chapter 3

Results

3.1 The pipeline - AFluID

AFluID was created as a pipeline that has two main functions: To identify IAV sequences and to direct these sequences post-identification to other bioinformatics tools performing clade assignment, mutation analysis and reference selection and retrieval. The following sections will review the pipeline's contract, I/O, development history and the rationale behind the decisions made until the release of this pipeline.

3.1.1 AFluID contract

AFluID must, from one file of processed NGS data (contigs or consensus):

- Conform and clean sequence data;
- Perform clustering of every sequence;
- Perform local alignment searches against cluster representatives of every unclustered sequence;
- Perform local alignment searches against the full sequence database on every unclustered sample on the previous steps;
- Compile the results of the aforementioned analysis in a final identification report;
- Redirect all identified sequences to appropriate bioinformatics analysis.
- Allows customizable thresholds and pipeline behaviour through configuration files or arguments.

In regards to the use and interpretation of the outputted data, input data quality and correct pipeline behaviour, the responsibility falls on the User.

3.1.2 Pipeline Inputs-Outputs(I/O)

As seen in Figure 2.3, AFluID accepts as input a FASTA format file and will output, depending on the set pipeline behaviour: an Identification Report tabular file, a Flags JSON file, a Nextclade clade assignment tabular file for every HA (H1, H3 and H5) sequence in file, Flumut Markers, Mutation and Literature tabular files and Reference GenBank and FASTA files (these two files are outputted to the references path).

Regarding the pipeline installation, the User will be required to provide a sequence database as a FASTA file along with sequence metadata as a CSV file delimited by semi-colons (;), a reference GenBank file (that can be empty) and a valid config.ini file. More documentation on installation and setup can be found at the AFluID repository mentioned in section 2.4 (Pereira [2024]).

3.1.3 Development iterations

The development of the pipeline was made in two iterations. In the first iteration this pipeline was conceived in Python for the different modules and Bash to run and manage each tool and module's I/O.

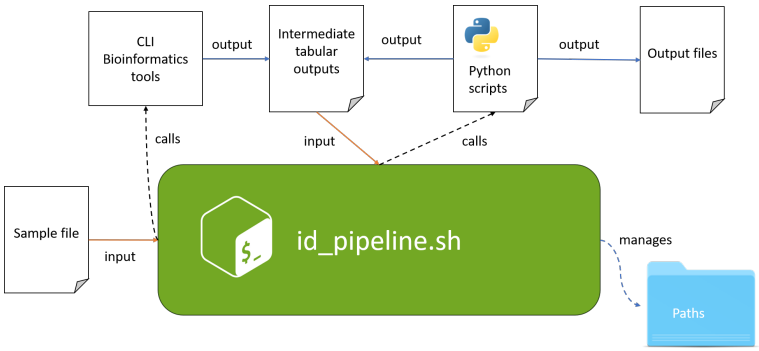

Figure 3.1: Workflow diagram of the first iteration of the AFluID pipeline
Inputs in orange, outputs in blue, calls represent a check and an input. Managing of paths includes creation, reads, writes and removal.

A backbone in Bash created some issues, mainly with compatibility, being incompatible with the Microsoft ecosystem due to command incompatibility between Powershell or Windows Command Prompt and Unix-based shells present in Linux OS Distributions and MacOS, and due to path nomenclature.

The pipeline's modular Python components were useful during prototyping, as they simplified debugging when exceptions or unexpected outputs occurred. However, having to deal with output as files in higher-throughput systems would increase space complexity.

Therefore, for a second iteration, in an effort to make the pipeline more concise and OS path agnostic, the code was completely refactored in Python, forgoing the Bash components.

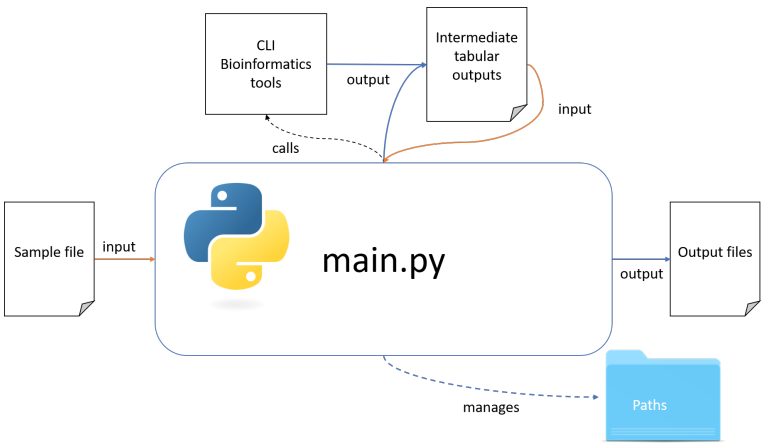

Figure 3.2: Workflow diagram of the second iteration of the AFluID pipeline.
Inputs in orange, outputs in blue, calls represent a check and an input. Managing of paths includes creation, reads, writes and removal.

3.1.4 Algorithm staging approach

AFluID was developed for rapid characterization of influenza A sample sequences with information that is of particular importance for genomic surveillance. As IAV is a source of potential zoonotic infections that can have pandemic potential it was considered that for this implementation subtypes and hosts were the crucial information needed to be presented to the end user.

As processing speed is a priority, this pipeline was developed with algorithm optimization in mind. Cd-hit uses a faster search algorithm than BLAST, therefore it was selected as the first step of the identification part of the system. The stringent 99% identity threshold for cd-hit identification was selected to match the criterion used for the generation of the cluster database. Also, cd-hit compares sequences without alignments, having less sensitivity for low-complexity sequences. If samples are unassigned by cd-hit, the pipeline defaults to the more reliable, but more costly BLAST algorithm to identify these.

For the same number of sequences using cd-hit is more than 10 times faster than using BLAST, on one processing thread, against the full sequence database that originated the clusters (23.33 seconds vs 413.85 seconds - on an Intel® Core™ i7-8750H 2.2GHz, with 8 Gb of RAM and running Ubuntu 22.04.5 LTS WSL on Microsoft® Windows™ 10). So in order to reach a compromise to increase speed without losing accuracy an intermediate step of BLAST against the cluster centroid database was devised. For this step the pipeline was also designed in a way so that only the unassigned sequences were subjected to BLAST, thus reducing the number of query sequences.

At this stage, it was observed that samples with less than 90% identity often failed to receive reliable assignments. In order to also avoid assigning matches that were too distant from their cluster centroids. Therefore, we set a minimum identity threshold of 90%. Samples falling below this threshold, or remaining unassigned, were subsequently subjected to BLAST against the full IAV sequence database. This

approach allows AFluID to maintain strong performance while preserving high accuracy.

The lower threshold was, to date, not enforced but is recommended at 85% identity backed by validations done and reported in the following sections 3.4.1 and 3.4.3)

3.2 Pretreatment of data

During development certain situations along the development and testing enforced the need to employ gate-keeping mechanisms that treat, conform and filter sequences prior to their input into the identification bioinformatics tools.

When dealing with drafted contigs and sequences with low coverage and complexity, sequence length thresholds were implemented to remove the smaller sequences that are often of no identification value and can even cause errors in identification (see section 3.4.1). The thresholds were first estimated using non-parametric limits due to the inability to normalise the sequence length data (see section 3.3.1), however when dealing with drafted contig data the defaults were expanded to capture shorter sequences that could be of identification value.

Also, along the development sequence header length as well as metacharacters such as the new line character('\n') or non-IUPAC nucleotide characters in sequences created several problems, in some cases even raising exceptions. Header conformation was then employed with a mapping dictionary to remap the original headers to the reports and non-IUPAC filters were added to remove these characters from sequences.

3.2.1 Choice of tools

The choice of bioinformatics tools that complement this pipeline (FluMut and NextClade) was made considering their use in INSaFLU, as NextClade, or for being the only validated tool for non-seasonal IAV, such is the case of FluMut and NextClade.

To deal with the limitations of tool validation but retaining some degree of user flexibility, a system of default configurations and optional arguments was integrated within the pipeline along with a redirector function that parses the identification report and by default only sends for NextClade or FluMut samples that meet the tools' validation criteria. Should the user want to perform mutation analysis of other subtypes with FluMut, this can be forced with the *-ff flumut* argument.

3.3 Datasets description

3.3.1 Sequences and Sequence Metadata

The sequence database that generated the BLAST and cluster databases comprises 748057 sequences annotated with information downloaded from NCBI virus and curated used the approaches described in

chapter 2. Upon generation and annotation of clusters, further analysis were performed that expanded the available fields adding ”Cluster” and ”Mutations” to the metadata.

Table 3.1 describes the data present for the sequence metadata and table 3.2 describes summary statistics for numeric and date-time variables, where applicable.

Table 3.1: Variable description of Sequence Metadata Variables

Variable	Type of variable	Example	Description
ACCESSION	CATEGORICAL	NC_026431.1	NCBI Acession Number
ORGANISM_ NAME	CATEGORICAL	Influenza A virus (A/California/07/ 2009(H1N1))	Organism name with strain
GENBANK_ REFSEQ	CATEGORICAL	RefSeq	From GenBank or RefSeq databases
ASSEMBLY	CATEGORICAL	GCF_001343785.1	Assembly Identifier
RELEASE_DATE	DATE-TIME	2015-02-23	Sequence release date
ISOLATE	CATEGORICAL	NaN	Isolate name or identifier
SPECIES	CATEGORICAL	Alphainfluenzavirus influenzae	Virus species
LENGTH	NUMERICAL	982	Sequence length
NUC_ COMPLETENESS	CATEGORICAL	complete	Nucleotide completeness (complete or partial)
GENOTYPE	CATEGORICAL	H1N1	Influenza subtype
SEGMENT	CATEGORICAL	7	Segment number (1 to 8)
COUNTRY	CATEGORICAL	USA	Isolate country
HOST	CATEGORICAL	Homo sapiens	Host species
COLLECTION_ DATE	DATE-TIME	2009-04-09	Sample collection date
MUTATIONS	CATEGORICAL	['31N']	Mutations of interest according to [Alvarez et al., 2025], as reported by FluMut
CLUSTER	CATEGORICAL	Cluster_47926	Cluster assignment after CD-Hit run

Table 3.2: Descriptive statistics of numeric and date-time variables

Variable	Type of variable	Min	Max	Mean	Std deviation	Median	1st Quantile	3rd Quantile
RELEASE_DATE	DATETIME	1982-06-09	2024-09-20	2018-05-29 21:54:45	N/A	2019-02-27	2015-03-06	2024-09-20
COLLECTION_DATE	DATETIME	1902-01-01	2024-08-27	2015-10-23 02:32:39	N/A	2017-07-21	2012-01-01	2022-02-11
LENGTH	NUMERIC	100	2867	1660	535	1565	1027	2233

With regards to sequence length attempts were made to define length cut-offs that could in turn elicit quality control flags particularly when working with poor quality samples. Mean and Standard deviations were calculated (Table 3.3) and length histograms were plotted for each segment to assess normality (Figures 3.3 and 3.4).

Table 3.3: Mean sequence lengths and standard deviations

Segment	**Mean Length**	**Length Standard Deviation**
PB2	2291.451498	121.988187
PB1	2290.628143	120.477240
PA	2178.265191	109.198921
HA	1719.393851	74.856051
NP	1521.954090	60.057030
NA	1422.786611	46.078498
MP	994.996300	26.820149
NS	856.274947	26.543735

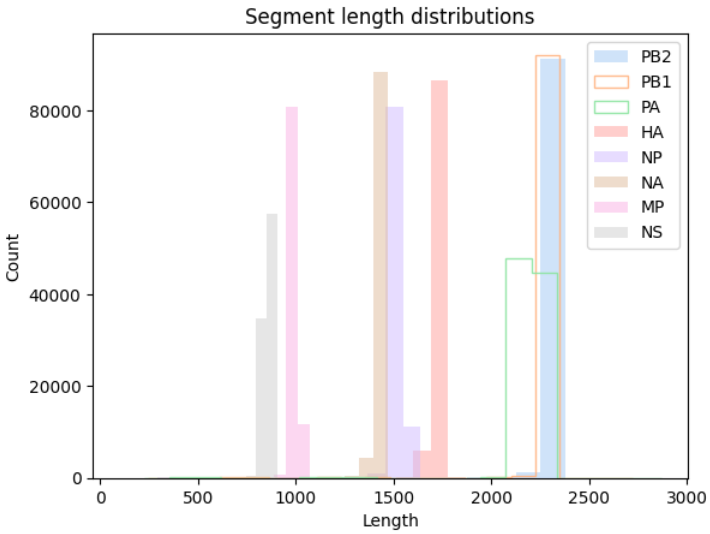

Figure 3.3: Histogram of sequence length by segment

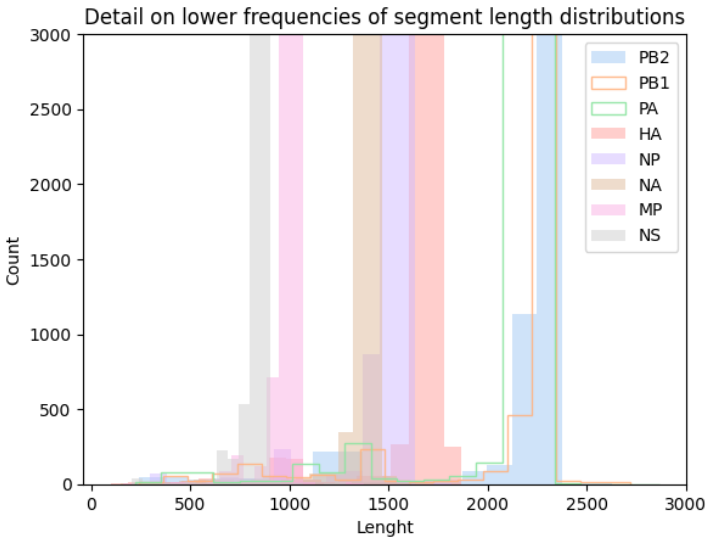

Figure 3.4: Rescaling of Figure 3.3 to allow for the viewing of the left-skewedness inapparent with a wider scale

It is apparent from figures 3.3 and 3.4 that the length distributions of sequence lengths is left-skewed. This asymmetry was still present when the lengths were converted to a logarithmic scale, therefore a non-parametric approach using the median and interquartile range was used to define what values could be considered outliers. These results are displayed in table 3.4

Table 3.4: Non-parametric thresholds of sequence length using the median subtracted and added to 1.5 times the interquartile range (IQR)

Segment	**Median - 1.5 IQR**	**Median + 1.5 IQR**
PB2	2226	2370
PB1	2211	2379
PA	2293.5	2605.5
HA	1639.5	1803.5
NP	1431	1607
NA	1363.5	1487.5
MP	952	1032
NS	797.5	905.5

As a next step, representation biases were ascertained by calculating and plotting the proportions of different hosts and subtypes. For the hosts, as taxa are hierarchical, roll-up functions were designed to collapse the taxa to the Order or Class levels. Plots showing the proportions of subtypes and host orders are presented on figures 3.5 and 3.6

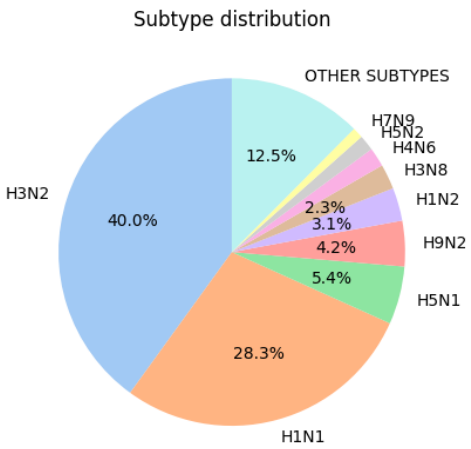

Figure 3.5: Subtype representation in the sequence database, other subtypes represent individually less than 2.5% and are, for example H9N2, H3N7 or H10N8

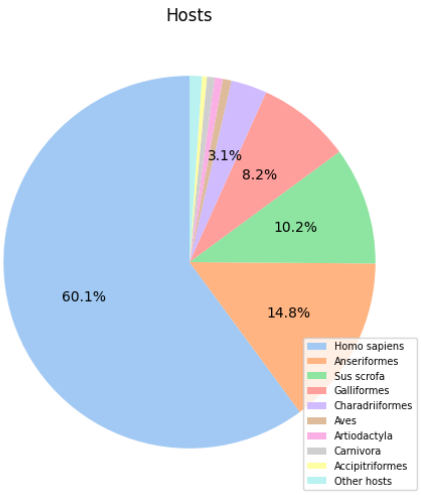

Figure 3.6: Host representation by Taxa, only human and swine samples are represented by species, the rest are represented by order or class

There is a great over-representation of H3N2 and H1N1,the main subtypes implicated in human seasonal endemic flu, therefore is of no surprise that the over-represented hosts within the dataset are humans (*Homo sapiens*), comprising 60.1% of all samples and swine (*Sus scrofa*) representing slightly more than 10 % of the dataset as a species (Figure 3.6).

It's also noteworthy that ducks, geese and swans (order *Anseriformes*) and seaguls and other shorebirds (order *Charadriiformes*) comprise 14.8 and 3.1% of all samples, respectively, considering they are the natural reservoir for non-seasonal IAV. A final note for the over-representation of Landfowl (such as chickens and turkeys - order *Galliformes*), that amount 8.2% of all samples, possibly due to the staggering economic impact IAV, particularly high pathogenic variants, have for these species.

This data thus reveals a dataset bias towards the seasonal influenza subtypes and hosts, that can pose difficulties with identification performance of non-seasonal subtypes.

3.3.2 Clustering and Cluster Annotations

The analytic steps done before were also performed for the clustered dataset. Homology clustering groups very similar sequences into cluster units, reducing the number of very similar sequences and enriching rarer, more variable ones. Clustering should prove very advantageous in two ways:

- Reduction of dataset redundancies decreasing representation biases and increasing computational efficiency;
- Grouping sequence annotations, clustering should yield more robust indications on similar sequences' subtypes and backgrounds (ecological and geographical).

The first step to prove these premises was to determine what were the 59876 clusters' centroid's subtypes (Figure 3.7).

From the plot in figure 3.7, it's clear a reduction of bias towards the H3N2 subtype, creating a more homogeneous dataset. On figure 3.8 it's also apparent the reduction of bias towards human-originated sequences. Now the large percentage of waterfowl sequences is very apparent. This data transformation reduced bias and enriched less frequent subtype sequences.

Another feature that is quite apparent is the different number of clusters per segment, a testament of how homology clustering enriches variability, seen by the increased number of HA and NA clusters compared to the non-antigenic segments (Figure 3.9).

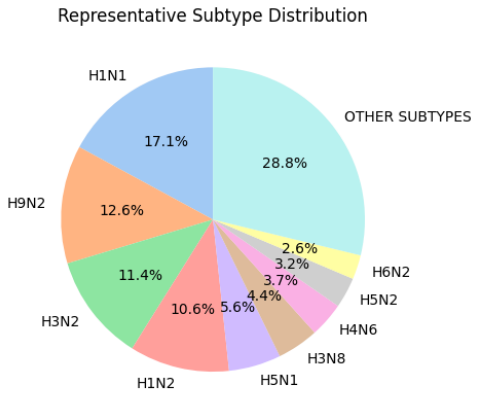

Figure 3.7: Subtype representation in the clustered database, other subtypes represent individually less than 2.5% .

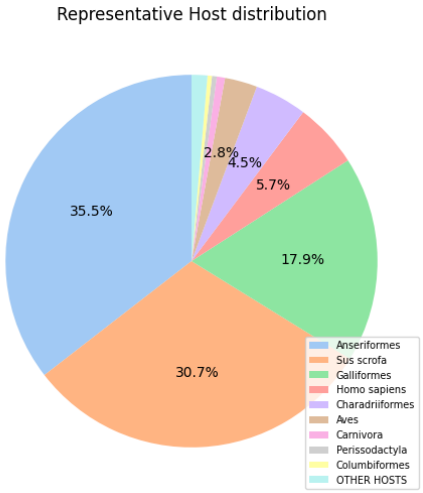

Figure 3.8: Host representation by Taxa, only human and swine samples are represented by species, the rest are represented by order or class

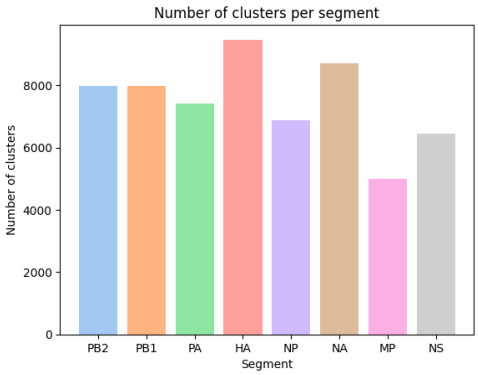

Figure 3.9: Number of clusters by segment.

Two more striking cluster particularities that are quite useful for genomic surveillance arose after analysing the data: Clusters are more commonly either avian or mammalian (Figure 3.10), mixed clusters being rare, and clusters are more frequently confined to a single continent rather than being widespread (Figure 3.11).

The plots represented in figures 3.10 and 3.11, show exactly this as pure avian and mammalian clusters represent 98.3% of all clusters, and 96.5% of clusters are confined to one continent and 2.2% are confined to two continents.

These results show that clusters are quite confined to taxonomic class and geographic landmass, giving them high discriminatory value for spillover and importation events.

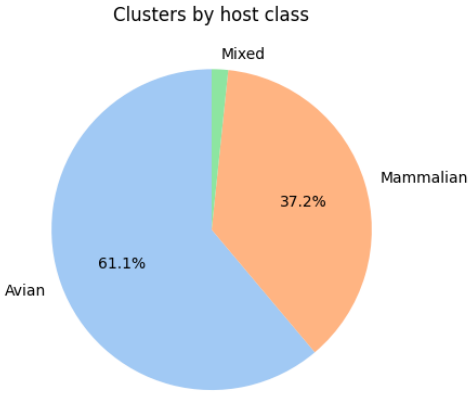

Figure 3.10: Clusters by taxanomic class content.

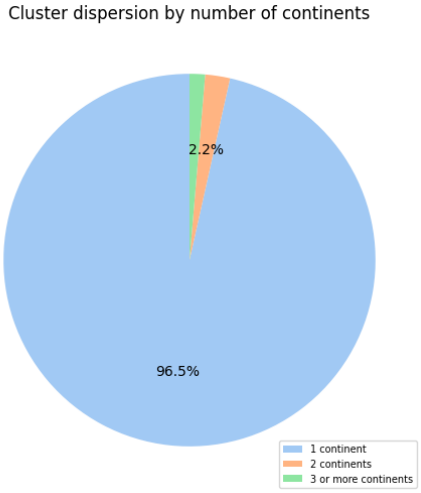

Figure 3.11: Cluster span in continents.

3.4 Validations

3.4.1 Testing workflow stability, computational efficiency, discriminatory power and some edge cases - NCBI IAV sequence dataset, NCBI IBV, ICV and Ebolavirus datasets

This first set of validation experiments aimed to test and fine-tune the identification component of AFluID's workflow, and to test identification accuracy and its limits.

To start debugging and validating the pipeline, a training dataset of 719 unique randomly selected sequences was obtained from NCBI. After the conclusion of the identification runs (i.e. without secondary bioinformatics analysis and size or identity thresholds) all of these sequences were identified. However, six of them were misidentified as valid sequences. Though vaccine sequences were assigned as antigenic proteins their identities were in two cases far below 85% (75%). Interference RNA (RNAi) molecules were clustered with cd-hit making clear the need to filter sequences by size as cd-hit could not discriminate low-complexity sequences.

Overall the pipeline managed to correctly identify, against sequence annotations, 99.17% of segments and 93.18% of subtypes.

The next part of these experiments was aimed at determining if other influenza samples were recognized as IAV by the pipeline. Thus, 300 IBV sequences and 100 ICV sequences were obtained from NCBI. Of these datasets there was positive identification of 13.3% and 4% of sequences, respectively, all for the PB1 segment albeit with less the 85% identity.

These experiments were crucial to begin determining the lower identity cut-off of the pipeline. The experiment in section 3.4.3 concluded the determination of the lower recommended cut-off of 85% identity.

Finally to determine if other RNA viruses could be identified as IAV, 14 sequences from Ebolavirus were also retrieved from NCBI and, as expected, no identifications were made.

3.4.2 Testing non-seasonal IAV enriched datasets - GISAID dataset

Due to the relative lack of sequences from other subtypes of IAV other than H1N1, H3N2, H1N2, H5N1 and H5N6, 8457 sequences were downloaded from GISAID. This dataset yielded 100% of segment identification and 72.85% of correct subtype identification, 100% in determining segments (HA and NA). The interesting result of this experiment was the correct identification of environmental sample with less than 90% identity. These identifications were the basis of the experiment described in the next section.

3.4.3 Testing for edge cases in low-quality sequences - GISAID environmental dataset

In order to test the performance of the pipeline when analysing environmental samples and recognising the lower identity cut-off for correct identification of the identification component of the pipeline, 22698 environmental IAV sequences were downloaded from GISAID. Here performance faltered due to

the low quality of the sequences. The correct identification of segment was 99.996% whereas correct subtyping was 78.28%, 98.95% in HA and NA segments. The lowest identity was 82% and was the only hit lower than 85%, therefore 85% will be adopted as the lowest cut-off value for identification.

3.4.4 Testing discriminatory power with curated datasets and detecting mutations - EQA datasets

The next experiment aimed to ascertain the performance of the pipeline against two well curated datasets from an external quality assessment. These datasets had both seasonal and non-seasonal samples, and comprised 80 and 138 sequences. These datasets also had reported mutation profiles that included some of the ones reported by Alvarez et al. [2025].

As far as identification performance is concerned both datasets had 100% identification rate for segment and had 86.25% and 81.88% correct identification of subtype respectively. When considering only HA and NA segments the identification accuracy rose to 100%.

Regarding mutation detections, a comparison was made between mutations of interest that were defined by Alvarez et al. [2025] present in the manual curation made by the EQA's authors against what AFluID was able to detect. In this particular analysis AFluID was able to detect all of the mutations of interest described in the EQA documentation. However, more mutations considered of interest by Alvarez et al. [2025] were found by AFluID, that were not reported in the documentation. Table 3.5 displays the results of this experiment.

Table 3.5: Results of mutation detection for the EQA datasets

EQA	Mutations of interest reported and found	Mutations of interest reported and not found	Mutations of interest found but not reported
EQA-1	26	0	28
EQA-2	10	0	23

Using the draft contigs from the EQA-2 dataset, we carried out the first experiment to assess identification performance on this type of data. Draft contigs play a key role in the INSaFLU identification pipeline, since their correct classification provides the references needed to build consensus sequences.

In this experiment, 11 sample files containing draft contigs were analysed without applying sequence length thresholds. Together, these files comprised 9027 sequences, of which 4464 were successfully identified. After parsing the determining segments, 10 sample subtypes were detected, while the eleventh sample could not be subtyped due to the absence of HA and NA sequences.

Overall, the results were positive. The sequences that were rejected were excluded mainly because of short length or low complexity. Importantly, the analysis demonstrated that AFluID is capable of handling draft contig data effectively, supporting its potential as a valuable addition to the reference-gathering process.

3.4.5 Detecting reassortments - Human seasonal reassortant IAV dataset

For this small experiment a reassortant seasonal IAV sequence with a predetermined reassortment profile was provided to be analysed through AFluID. The sequences were made available through GISAID with the acession number EPI_ISL_304183. Table 3.6 displays an abridged output of AFluID, highlighting the reassortment profile.

Table 3.6: Abridged output of AFluID highlighting in light orange the reassortant segments

SAMPLE_NAME	CLUSTER	%ID	SEGMENT	GENOTYPE	HOST
>1 | PB2 | A/Netherlands/10407/2018 | A / H1N2	>Cluster 35	99.1	1:7767	H3N2:99.0, H1N1:1.0	Homo sapiens:100.0
>2 | PB1 | A/Netherlands/10407/2018 | A / H1N2	>Cluster 67	99.1	2:6655	H3N2:100.0	Homo sapiens:100.0
>3 | PA | A/Netherlands/10407/2018 | A / H1N2	>Cluster 15328	99.2	3:4114	H3N2:100.0	Homo sapiens:100.0
>4 | HA | A/Netherlands/10407/2018 | A / H1N2	>Cluster 22866	99.4	4:1888	H1N1:100.0	Homo sapiens:98.5, Sus scrofa:1.4, Acinonyx jubatus:0.1
>5 | NP | A/Netherlands/10407/2018 | A / H1N2	>Cluster 32212	99.1	5:2137	H3N2:100.0	Homo sapiens:99.9, Sus scrofa:0.1
>6 | NA | A/Netherlands/10407/2018 | A / H1N2	>Cluster 38751	99.2	6:3404	H3N2:100.0	Homo sapiens:99.9, Sus scrofa:0.1
>7 | MP | A/Netherlands/10407/2018 | A / H1N2	>Cluster 48032	99.4	7:15312	H3N2:100.0	Homo sapiens:99.9, Sus scrofa:0.1
>8 | NS | A/Netherlands/10407/2018 | A / H1N2	>Cluster 52990	99.1	8:2058	H1N1:98.9, H3N2:1.0	Homo sapiens:97.5, Sus scrofa:2.2, Mustela lutreola:0.1, Meleagris gallopavo:0.1

3.5 Proof-of-concept experiment: the INSA-INIAV avian influenza dataset

This set of final experiments was the real proof-of-concept, regarding accuracy and application in a real-life setting. As mentioned in Chapter 2, 28 samples collected from Portuguese outbreaks, identified by the INSA-INIAV partnership and kindly provided for this experiment after being sequenced and then assembled by INSaFLU were processed by AFluID in four different runs:

- INSaFLU generated draft contigs were first analysed in **contig mode** to automatically gather references and rebuild consensus using the INSaFLU platform.

- Consensus sequences generated with INSaFLU reference library were analysed in **consensus mode** for identification, typing, clade assignment and mutation profiling. These sequences correspond to the INSaFLU curations using the laboratory's standard practices.
- Consensus generated using IRMA, were analysed also in **consensus mode**. These consensus were fully generated from the reads using IRMA's reference database an its iterative assembly refinement algorithm.
- Consensus generated using AFluID obtained references in the **contig mode** run were also analysed in **consensus mode**. These consensus were generated using INSaFLU but with reference sequences provided through AFluID's reference gathering capabilities.

After the four aforementioned runs results were interpreted and compared between assembly methods. The results will be presented in the following sections.

3.5.1 Sequence and sample characterization

All of the 28 samples were identified when considering all assembly methodologies as H5N1 coming from mainly an avian background. The clade assignment revealed that all samples belonged to the 2.3.4.4b clade.

There were some examples of potential reassortment that warrant future investigation, namely sample 19 where PA, NP and NS were closely related to references sequences with a H13 background, and for NP and NS there was also close relation to sequences coming from a H16 background.

3.5.2 Assembly method comparison

For this experiment 3 methods of assembly were compared. Two INSaFLU reference-based assemblies, one with INSaFLU references and one with AFluID references, as well as a non reference-based assembly by IRMA. There were two noteworthy results from this experiment.

Firstly IRMA managed to build consensus of all the segments every time. INSaFLU assemblies are more constrained by reference quality and vertical and horizontal coverages and were unable to generate consensus for all segments sometimes. This hindered characterization of samples generated with INSaFLU because of missing determinant segments.

Secondly there were differences with cluster assignments for similar sequences coming from different assembly methods. Although the differences in cluster number did not translate in differences in sequence epidemiological background, the causes for different cluster assignments for the same sequences had to be investigated. Therefore, alignments per segment per sample were made using MAFFT (Katoh and Standley [2013]), and calculation of pairwise distances using number of differences was performed using MEGA (Kumar et al. [2024]). Between the reference based approaches in roughly 278214 nucleotides only 3 SNP's were seen in three HA sequences between methods, reduces the nucleotide differences to

0.0011%. Upon greater exploration it was noted that some assemblies retained the amplicon fragments in the ends of the sequences increasing their length. Cd-hit being length-sensitive placed similar sequences in different clusters.

3.5.3 Mutation analysis

As for the mutation analysis, every mutation of interest detected in one segment was detected for all assembly methods should the segment be present, therefore if the same segment was assembled in all assembly methods, the three sequences shared the same mutation profile. The mutations found and their biological effects are presented on table 3.7

Table 3.7: Results of mutation detection for the INSA-INIAV dataset

Mutations found	Samples where mutations were detected	Mutation biological effect
PB1-F2:66S	All	Increased virulence, replication efficiency and antiviral response in mice
HA-1:156A	All but sample 15	Increased virus binding to α2,6/ increased transmission in guinea pigs
PB2:292V	9; 28	Observed in H3N8 samples from humans and birds / increased polymerase activity in a mammalian cell line / increased virulence in mice
NP:52N	21	Significant mutation observed also in H5N1 / Role in Butyrophilin Subfamily 3 Member A3 (BTN3A3) evasion

On table 3.7 one can note that almost all samples have at least one mutation that allows for immune evasion in the mammalian host and one mutation that allows for better binding to the α2,6-SA. Although more mutations act synergistically to increase the risk for mammalian infection, these parallel findings warrant future investigation.

Chapter 4

Discussion

Considering the results presented in the previous chapter, some considerations have to be made.

4.1 Data sources and system sustainability

This work highlighted the importance of accurate curation and annotation of sequence data in classification systems. The principle of GIGO (*garbage in-garbage out*) was especially relevant when handling low quality sequences. Careful curation of the sequence database and associated metadata, together with the definition of thresholds, were essential to ensure reproducibility and to allow direct comparisons between AFluID and current industry standards.

The curation process was extensive and even revealed annotation errors in public repositories such as NCBI and GISAID. These problems directly affect the sustainability of the platform, since every automatically retrieved sequence must undergo validation tests with cd-hit and BLAST to verify whether the annotated segments and subtypes are correct. This curated dataset comes as a robust starting point for this, with ability to correctly type and subtype assemblies should the need arise.

When considering a full update routine further difficulties appear. These include:

- incompleteness of metadata, with differences between manual retrieval from online platforms and programmatic access via CLI applications or Python APIs,
- the need to implement database update check mechanisms,
- the requirement to perform clustering and BLAST database regeneration each time new data are added.

Some annotations that are not directly available through API calls can still be extracted from other API endpoints or from GenBank files. However, this mining procedure requires several steps and becomes increasingly complex as the number of sequences grows. Together with the necessity of reclustering and BLAST database regeneration, this means that updates cannot occur continuously. Instead, they must be

scheduled infrequently, ideally during periods of lower computational load. The use of a *sensu stricto* relational database optimised for transactions may improve this process, although the main bottlenecks remain the applications and API calls themselves.

With regard to the timing of updates, once the above issues are solved, they should follow seasonal epidemiological trends. Updates would therefore be performed once before each expected influenza season. Among the challenges described, the most manageable is the addition of update check mechanisms such as version hashes, release date checks and ID blacklisting of rejected sequences.

Another limitation arises from the inclusion of non-public data, such as GISAID sequences. Since GISAID does not broadly distribute open-sourced and standardised APIs or CLI binaries for data retrieval, any integration of these sequences requires extra processing to conform the data and metadata to AFluID's structure. Furthermore, if GISAID sequences are incorporated, AFluID or its databases cannot be distributed unconditionally as standalone tools.

Even without directly embedding GISAID sequences in the AFluID database, data from this source were essential for validation and for defining pipeline thresholds. For this reason, a list of the GISAID sequence contributions is provided in the AFluID repository to acknowledge their vital role in the development of the pipeline (Pereira [2024]).

4.2 Bioinformatics tools

Each bioinformatic tool comes with advantages and limitations. Some limitations, such as the cd-hit low specificity when dealing with oligonucleotides, or the slow throughput of BLAST when dealing with large sequence databases were surpassed by using *a priori* sequence processing, staging the use of different tools to increase time efficiency and by default redirection of specific subtype sequences to the validated tools (example: only H5Nx samples were run on FluMut, only H1, 3 and 5 HA sequences were processed through NextClade).

During the INSA-INIAV dataset experiment it was noted that some sequences were assigned to different clusters. Although the clusters had the same epidemiological background, differences in assignments for sequences that once aligned were basically identical reveals the main drawback for using cd-hit as a classification tool: length-sensitivity. Although local alignments are performed during the cluster definition stage, classification of sequences against another database is done through k-mer comparison. Even enabling options that use local alignment does not completely curb this issue. With that in mind some mitigation steps are considered going forward:

- Removal of clusters that contain only one or a few sequences, or remove the centroids and redefine the clusters and or removal of sequences that fall within more stringent quality control parameters such as sequence length or amount of degenerate nucleotides or N's;
- Considering the definition of superclusters that group clusters within a high threshold of identity within their centroids (e.g 97% using local alignment, with high coverage such as more than 95%)

that share the same epidemiological background;

- Considering forgoing cd-hit as a classification tool and using blast against the cluster centroids exclusively.

Circling back to pipeline design and validation, the reference gathering and selection pipeline performed quite well, generating consensus that were not distant or quite close to the ones generated by IRMA or the standard of INSaFLU practice. However, with regards to complete consensus generation IRMA outperformed reference based methods and will be a great addition to the current INSaFLU workflow.

Other issues such as detection of less reported mutations by lack of database information or by unsuitability of reference sequences were not solved. To attempt a broader solution to this issue other solutions were studied such as a FluMut-like approach with standardised and open references. In the end the third party tools were preferred due to their validation in limited scenarios. Although FluMut does not perform poorly in non-H5Nx scenarios, the unavailability of reference data within the documentation and the constant updates to the FluMutDB cast a shadow of doubt on the results of non-H5N1 samples. The previous issue was mitigated using only the mutations of interest described by Alvarez et al. [2025].

4.3 Performance and future improvements

Identification performance was good when comparing to annotation and manual inspection when needed, therefore AFluID can be relied upon as a standalone tool for identification of IAV samples and as a primary step for further bioinformatics analysis, as far as consensus sequences are concerned.

In the case of drafted contigs, some threshold calibration might be needed to be able to isolate all segment sequences. While running the pipeline it was noted that sometimes in order to identify all segments within a drafted contig file, the minimum length threshold had to be reduced, which in turn increased the number of sequences for segments already identified.

As a solution to this issue, an iterative refinement algorithm will be put in place that stores each segment's sequence within a data structure, by sequence length and identity. Should any segment be missing, the pipeline would run the identification portion again with lower minimum length thresholds until all segments are found or a minimum size limit be reached. The diagram in figure 4.1 illustrates this algorithm.

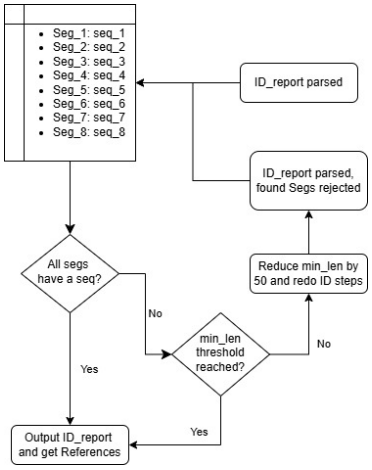

Figure 4.1: Iterative segment search within AFluID.
The internal storage block on the upper left corner represents a data structure, in this case a Hashmap or Dictionary, that stores sequence information for each sequence. The selection by size and Identity will be made during the parsing step using pandas data frame filters.

When dealing with non-IAV samples one must be mindful of the PB1 identification pitfalls, this is why the 85% minimum identity threshold should be respected. However, in order for AFluID to be able to detect and differentiate other influenza types, a database expansion with metadata reconditioning could be implemented to expand the pipeline's capabilities.

Currently the pipeline does not support automatic genotype constellation analysis with genin2, a tool developed by the same team that developed FluMut. This tool gives a characterisation of H5 clade 2.3.4.4b sequences within their representative genotype constellation nomenclature. Genotype constellation in segmented viruses refers to the specific combination of gene segments in association with their particular ancestral lineage. Analysis of genotype constellations can inform about reassortment profiles, genotype turnover and emergence and virulence and host adaptations (Youk et al. [2023]). Genin2 comes with the severe drawback of being validated only for clade 2.3.4.4b samples, making its use somewhat restricted. However, after NextClade clade assignment, in single-sample mode only this integration could be implemented with relative ease.

Parallels can be drawn between genotype constellations and the assignment of clusters by AFluID, since although the clusters are not obtained through phylogenetic algorithms, they group sequences with similar properties and origins. One could say that an AFluID analysis that identified only sequences against clusters would give us a "constellation" of clusters, one that is less limited in scope than the genotype constellations obtained with Genin2. These "constellations" were able to predict reassortments, however when considering broader reassortment detections, further studies are warranted to be able to infer reassortments from closely related clusters, namely when subtypes and hosts are concerned. Creating statistical measures of co-occurrence or reassortment likelihood would be a great addition to the

pipeline in the future.

Chapter 5

Conclusion

5.1 Conclusion

This work created several outputs and yielded six main conclusions:

- The generated AFluID pipeline is able to accurately identify segments and infer subtype and host background in IAV sequences from generated consensus, even low-quality ones. The accuracy of identification is enough that only depending on AFluID alone, sequences can be automatically directed to mutation analysis and clade assignment, fulfilling the first and part of the second objectives.
- In regards to sequences obtained from draft contig generation, AFluID is able to characterise the subtype and automatically gather closely related references, considerably hastening consensus generation with almost perfect similarity compared to INSaFLU's standard workflow.
- With secondary remarks to automation AFluID also automatically redirects sequences to the appropriate post-identification tools (Nextclade and FluMut) that provide additional characterization of clinical samples. In FluMut's particular case the filter for mutations of interest will also be applied automatically, streamlining the detections of mutations of interest on clinical samples.
- In regards to the detection of mutations of biological interest, upon filtering for the mutations of interest within the consensus agreed on by the EFSA and the ECDC (Alvarez et al. [2025]), there is accurate identification of the ones reported when that data is available, such as in the EQA datasets, and identification of others that were not reported, even in non H5N1 backgrounds. Nevertheless caution must always be employed when inferring biological implications from bioinformatics analysis.
- Clustering makes for an advantageous and fast way to characterise samples in their subtype and ecological background, as it reduces redundancy and representation bias, and their composition is

fairly homogeneous both in Taxa as in Geographic origin, aiding in genomic surveillance work. However, results from the INSA-INIAV dataset experiment were vital in determining that clusters, without their background, should not be used as an epidemiological identifier until further refinement of the cluster database is performed.

- Although reassortment inference may be performed using AFluID, more studies are required to be able to perform said inference with added certainty.
- As for the unified final report, it was implemented for single-sample mode as data analysis permitted mining of all outputs into more conformed media. For batched sequences the implementation will have to be a different one as AFluID does not output the same wealth of data with the same conformation for the batch mode.

As outputs from this work were created dataset and metadata publications in Zenodo, a GitHub repository and a poster presentation at the fifth virus bioinformatics meeting (ViBioM) hosted in Lisbon on May 13th 2025.

5.2 Future work

Improvements on AFluID are planned, as stated on 4. The most pressing ones being the unified user-friendly reports both for single and multi-sample modes, and the refinement of the contig mode to allow for the choice of only one drafted contig sequence per segment, facilitating consensus generation and single-sample analysis on drafted contig files.

This work was an exploratory work that will be expanded for more species of segmented viruses within a larger structure. Future plans also centre on developing systems relying in machine and deep learning frameworks to detect reassortments on segmented viruses.

References

Aksamentov, I., Roemer, C., Hodcroft, E., and Neher, R. (2021). Nextclade: Clade assignment, mutation calling and quality control for viral genomes. *Journal of Open Source Software*, 6:3773. VI, 13, 22

Alvarez, J., Boklund, A., Dippel, S., Dórea, F., Figuerola, J., Herskin, M. S., Michel, V., Ángel Miranda Chueca, M., Nannoni, E., Nielsen, S. S., et al. (2025). Preparedness, prevention and control related to zoonotic avian influenza. *EFSA Journal*, 23. V, VI, XV, 3, 4, 7, 8, 40, 51, 57, 61, 70, 71, 72, 73, 74

Borges, V., Pinheiro, M., Pechirra, P., Guiomar, R., and Gomes, J. P. (2018). Insaflu: An automated open web-based bioinformatics suite "from-reads" for influenza whole-genome-sequencing-based surveillance. *Genome Medicine*, 10:46. V, 10, 11

Burke, D. F. and Smith, D. J. (2014). A recommended numbering scheme for influenza a ha subtypes. *PLOS ONE*, 9:e112302. 6

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2009). Blast+: Architecture and applications. *BMC Bioinformatics*, 10:421. VI, 11, 21

Centers for Disease Control and Prevention (2024a). Influenza – cdc yellow book 2024: Health information for international travel. https://www.cdc.gov/yellow-book/hcp/travel-associated-infections-diseases/influenza.html. Accessed: 2025-07-29. VI, 7

Centers for Disease Control and Prevention (2024b). Types of influenza viruses. https://www.cdc.gov/flu/about/viruses-types.html. Accessed: 2025-07-29. 7

Centers for Disease Control and Prevention (CDC) (2024). Technical report: June 2024 highly pathogenic avian influenza a(h5n1) viruses. https://www.cdc.gov/bird-flu/php/technical-report/h5n1-06052024.html. [Online; accessed 2025-07-23]. 8

Chen, J., Xu, L., Liu, T., Xie, S., Li, K., Li, X., Zhang, M., Wu, Y., Wang, X., Wang, J., et al. (2022). Novel reassortant avian influenza a(h5n6) virus, china, 2021. *Emerging Infectious Diseases*, 28:1703–1707. 8

Cingolani, P., Platts, A., Coon, M., Nguyen, T., Wang, L., Land, S., Lu, X., and Ruden, D. (2012). A program for annotating and predicting the effects of single nucleotide polymorphisms, snpeff: Snps in the genome of *drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly*, 6(2):80–92. 12

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and De Hoon, M. J. L. (2009). Biopython: Freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423. 28

Cock, P. J. A. et al. (2025). Biopython documentation. `https://biopython.org/docs/latest/`. [Online; accessed 2025-07-23]. 28

Danecek, P., Auton, A., Abecasis, G., Albers, C., Banks, E., DePristo, M., Handsaker, R., Lunter, G., Marth, G., Sherry, S., McVean, G., Durbin, R., and 1000 Genomes Project Analysis Group (2011). The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158. 12

Dou, D., Revol, R., Östbye, H., Wang, H., and Daniels, R. (2018). Influenza a virus cell entry, replication, virion assembly and movement. *Frontiers in Immunology*, 9:1581. 4, 5, 6

Elbe, S. and Buckland-Merrett, G. (2017). Data, disease and diplomacy: Gisaid's innovative contribution to global health. *Global Challenges*, 1:33–46. VI, 12

Giussani, E. and Sartori, A. (2024). Flutmut. VI, 14, 22

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., et al. (2020). Array programming with numpy. *Nature*, 585:357–362. 28

Holmes, E. C., Hurt, A. C., Dobbie, Z., Clinch, B., Oxford, J. S., and Piedra, P. A. (2021). Understanding the impact of resistance to influenza antivirals. *Clinical Microbiology Reviews*, 34. VI, XV, 4, 5, 7, 74

Katoh, K. and Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772–780. 22, 53

Kellis, M. et al. (2025). *3.5 The BLAST Algorithm (Basic Local Alignment Search Tool)*, page 3.5.1–3.5.3. LibreTexts. 11

Khare, S., Gurry, C., Freitas, L., Schultz, M. B., Bach, G., Diallo, A., Akite, N., Ho, J., Lee, R. T. C., Yeo, W., et al. (2021). Gisaid's role in pandemic response. *China CDC Weekly*, 3:1049–1051. VI, 12

Kumar, S., Stecher, G., Suleski, M., Sanderford, M., Sharma, S., and Tamura, K. (2024). MEGA12: Molecular Evolutionary Genetics Analysis version 12 for adaptive and green computing. *Molecular Biology and Evolution*, 41(1):1–9. 22, 53

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760. 12

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and Subgroup, . G. P. D. P. (2009). The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079. 12

Li, W. and Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22:1658–1659. VI, 12, 21

Liang, Y. (2023). Pathogenicity and virulence of influenza. *Virulence*, 14:2223057. 1, 8

Liu, R., Sheng, Z., Huang, C., Wang, D., and Li, F. (2020). Influenza d virus. *Current Opinion in Virology*, 44:154–161. 1, 2

Matrosovich, M. N., Matrosovich, T. Y., Gray, T., Roberts, N. A., and Klenk, H.-D. (2004). Neuraminidase is important for the initiation of influenza virus infection in human airway epithelium. *Journal of Virology*, 78:12665–12667. 4

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 51–56. 28

Merriam-Webster (2025). Influenza. merriam-webster.com dictionary. https://www.merriam-webster.com/dictionary/influenza. [Online; accessed 2025-07-23]. 2

Mohapatra, S., Rath, D. K., Prusty, B., Pati, S. K., Parai, D., Shukla, S., Rana, S. K., Bishi, D. K., Rout, A. K., Joshi, H. K., et al. (2023). Evolution of influenza viruses—drug resistance, treatment options and emerging therapeutics. *International Journal of Molecular Sciences*, 24:12244. VI, XV, 4, 5, 74

National Academies of Sciences, Engineering, and Medicine (2023). What is the difference between seasonal flu and pandemic flu? https://www.nationalacademies.org/based-on-science/what-is-the-difference-between-seasonal-flu-and-pandemic-flu. Accessed: 2025-07-29. 7

National Center for Biotechnology Information (NCBI) (1988). National center for biotechnology information (ncbi) [internet]. [Online; accessed 2025-07-23]. 11

NumFOCUS Inc. (2025). Pandas user guide. https://pandas.pydata.org/docs/user_guide/index.htm. [Online; accessed 2025-07-23]. 28

NumPy Developers (2025). Numpy: Absolute beginner's guide. https://numpy.org/devdocs//user/absolute_beginners.html. [Online; accessed 2025-07-23]. 28

Pan American Health Organization (PAHO) and World Health Organization (WHO) (2025). Timeline – influenza a(h5n1) americas region. https://www.paho.org/en/timeline-influenza-ah5n1-americas-region. [Online; accessed 2025-07-23]. 8

Peacock, T., Moncla, L., Dudas, G., VanInsberghe, D., Sukhova, K., Lloyd-Smith, J. O., Worobey, M., Lowen, A. C., and Nelson, M. I. (2024). The global h5n1 influenza panzootic in mammals. *Nature*. V, 1, 3, 4, 8

Pereira, J. (2024). Afluid: Pipeline for influenza a virus segment and genotype identification through homology clustering and local alignment. GitHub repository. [Online; accessed 2025-07-23]. 33, 36, 56

Pereira, J. (2025). Influenza a sequence metadata with manual corrections and annotations (version 4.5). Zenodo. [Online; accessed 2025-07-23]. 33

Python Software Foundation (2025). Python 3.13.5 documentation. https://docs.python.org/3/. [Online; accessed 2025-07-23]. 25, 26, 27

Rabalski, L., Milewska, A., Pohlmann, A., Gackowska, K., Lepionka, T., Szczepaniak, K., Swiatalska, A., Sieminska, I., Arent, Z., Beer, M., et al. (2023). Emergence and potential transmission route of avian influenza a (h5n1) virus in domestic cats in poland, june 2023. *Eurosurveillance*, 28:2300390. 8

Santos, J. D., Sobral, D., Pinheiro, M., Isidro, J., Bogaardt, C., Pinto, M., Eusébio, R., Santos, A., Mamede, R., Horton, D. L., Gomes, J. P., and Borges, V. (2024). Insaflu-televir: An open web-based bioinformatics suite for viral metagenomic detection and routine genomic surveillance. *Genome Medicine*, 16:73. V, 10

Seemann, T. (2023). Abricate: Mass screening of contigs for antimicrobial and virulence genes. https://github.com/tseemann/abricate. [Online; accessed 2025-07-23]. 11

Seemann, T. (2025). Snippy: rapid haploid variant calling and core genome alignment. https://github.com/tseemann/snippy. 12

Sengkeopraseuth, B., Co, K. C., Leuangvilay, P., Mott, J. A., Khomgsamphanh, B., Somoulay, V., et al. (2022). First human infection of avian influenza a(h5n6) virus reported in lao people's democratic republic, february–march 2021. *Influenza and Other Respiratory Viruses*, 16:181–185. 8

Shepard, S. S., Meno, S., Bahl, J., Wilson, M. M., Barnes, J., and Neuhaus, E. (2016). Viral deep sequencing needs an adaptive approach: Irma, the iterative refinement meta-assembler. *BMC Genomics*, 17:708. 13

Shu, Y. and McCauley, J. (2017). Gisaid: Global initiative on sharing all influenza data – from vision to reality. [Online; accessed 2025-07-23]. VI, 12

Suttie, A., Deng, Y. M., Greenhill, A. R., Dussart, P., Horwood, P. F., and Karlsson, E. A. (2019). Inventory of molecular markers affecting biological characteristics of avian influenza a viruses. *Virus Genes*, 55:739–768. 4, 6

U.S. Centers for Disease Control and Prevention (2025). Influenza risk assessment tool (irat) — virus summaries. https://www.cdc.gov/pandemic-flu/php/monitoring/irat-virus-summaries.html. [Accessed: 2025-07-01]. 9

World Health Organization Global Genomic Surveillance, Research for Health (RFH) Team (2023). Considerations for developing a national genomic surveillance strategy or action plan for pathogens with pandemic and epidemic potential. Technical document, World Health Organization, Geneva. [Online; accessed 2025-07-23]. V, VI, XIII, 9, 10

World Organisation for Animal Health (WOAH) (2025). Avian influenza. [Accessed: 2025-07-09]. 8, 9

Youk, S., Torchetti, M. K., Lantz, K., Lenoch, J. B., Killian, M. L., Leyson, C., Bevins, S. N., Dilione, K., Ip, H. S., Stallknecht, D. E., Poulson, R. L., Suarez, D. L., Swayne, D. E., and Pantin-Jackwood, M. J. (2023). H5n1 highly pathogenic avian influenza clade 2.3.4.4b in wild and domestic birds: Introductions into the united states and reassortments, december 2021–april 2022. *Virology*, 587:109860. 58

Appendix A

Extra Information

This chapter presents auxiliary information about the work. The following tables describe all the mutations of interest considered for this work.

Table A.1: Mutations of Interest from [Alvarez et al., 2025]

Locus H5n	Locus H3n	Effect	Trait
HA:156A	HA:160A	Increased virus binding to α2,6/ increased transmission in guinea pigs	Increases mammalian specificity of virus attachment to receptor (receptor preference)
HA:156V	HA:160V	Observed in seal samples / decreased virulence in mice and increased affinity for the human-type receptor	Increases mammalian specificity of virus attachment to receptor (receptor preference)
HA:186D/221D	HA:190D/225D	Switch in the receptor specificity from avian-type to human-type receptor	Increases mammalian specificity of virus attachment to receptor (receptor preference)
HA:186V	HA:190V	Enhances binding affinity to mammalian cells and replication in mammalian cells / enhanced replication in mice	Increases mammalian specificity of virus attachment to receptor (receptor preference)
HA:221D	HA:225D	Increased virus binding to α2,6	Increases mammalian specificity of virus attachment to receptor (receptor preference)
HA:222L	HA:226L	Increased virus binding to α2,6 / transmitted via aerosol among guinea pigs/ enhanced replication in mammalian cells and ferrets / enhanced contact transmission in ferrets / Loss of binding to α2,3 / Increased acid and thermal stability	Increases mammalian specificity of virus attachment to receptor (receptor preference) / Increases HA stability in mammal's environment
HA:224S	HA:228S	Increased binding to α2,6 / increased viral replication in mammalian cells and virulence in mice / observed in human isolate / increase in the ability of the virus to infect mammals / decreased virus binding to α2,3	Increases mammalian specificity of virus attachment to receptor (receptor preference)
NA:399R	N/A	Mutation of 2SBS that are detrimental to the cleavage of sialosides linked to fetuin or transferrin / observed in seals	Disruption of the second sialic acid binding site (2SBS) in neuraminidase
NA:432E	N/A	Decreased cleavage of fetuin-containing α2,3-linked sialic acids (SIAs) but not that of monovalent substrates or of transferrin containing only α2,6-linked SIAs	Disruption of the second sialic acid binding site (2SBS) in neuraminidase

Table A.1: Mutations of Interest from [Alvarez et al., 2025] (continued)

Locus H5n	Locus H3n	Effect	Trait
PA:356R	N/A	Human host marker / Increase polymerase activity and enhanced replication in a mammalian cell line / increased virulence in mice	Increased activity of the viral polymerases in mammalian hosts
PA:552S	N/A	Enhanced viral RNA-dependent RNA polymerase (vRdRp) activity and viral replication in vitro	Increased activity of the viral polymerases in mammalian hosts
PA:85I	N/A	Enables guanine-rich sequence binding factor (GRSF1) to enhance the cytosolic accumulation and translation of a subset of viral mRNAs / enhanced replication in human A549 cells	Increased activity of the viral polymerases in mammalian hosts
PA:97I	N/A	Increased polymerase activity in mammalian cell line and enhanced replication and virulence in mice / increased polymerase activity in mammalian cells	Increased activity of the viral polymerases in mammalian hosts
PB1-F2:66S	N/A	Increased virulence, replication efficiency and antiviral response in mice	Increased activity of the viral polymerases in mammalian hosts
PB2:271A	N/A	Increase polymerase activity in the presence of avian acidic nuclear phosphoprotein 32 (ANP32) proteins / increase mortality and airborne transmission in ferrets / increased polymerase activity in avian and mammalian cell line	Increased activity of the viral polymerases in mammalian hosts
PB2:292V	N/A	Observed in H3N8 samples from humans and birds / increased polymerase activity in a mammalian cell line / increased virulence in mice	Increased activity of the viral polymerases in mammalian hosts
PB2:526R	N/A	Increased polymerase activity in mammalian cell line and virulence in mice	Increased activity of the viral polymerases in mammalian hosts
PB2:588I	N/A	Enhances 2009 H1N1 pandemic influenza virus virulence by increasing viral replication and exacerbating PB2 inhibition of beta-interferon expression	Increased activity of the viral polymerases in mammalian hosts

Table A.1: Mutations of Interest from [Alvarez et al., 2025] (continued)

Locus H5n	Locus H3n	Effect	Trait
PB2:588V	N/A	Increased polymerase activity and replication in mammalian and avian cell lines / increased virulence in mice	Increased activity of the viral polymerases in mammalian hosts
PB2:591K	N/A	Increased polymerase activity in mammalian and avian cell line / increased replication in a mammalian cell line / increased virulence in mice / observed in human isolates	Increased activity of the viral polymerases in mammalian hosts
PB2:591R	N/A	Enhanced mammalian ANP32 adaptation (non-biased)	Increased activity of the viral polymerases in mammalian hosts
PB2:627K	N/A	Observed in human isolate / increased polymerase activity and replication in mammalian cell line / increased virulence in mice and ferrets / contributes to airborne transmission of influenza A viruses (IAVs) in ferrets and contact transmission in guinea pigs / mammalian ANP32-specific adaptation (ANP32B biased)	Increased activity of the viral polymerases in mammalian hosts
PB2:627V	N/A	Observed in human isolate / mammalian ANP32-specific adaptation (ANP32B biased) / observed in human samples in Shenzhen / increased polymerase activity and replication in mammalian cell lines / increased virulence in mice	Increased activity of the viral polymerases in mammalian hosts
PB2:631L	N/A	Most dominant mutation in mouse-adapted virus that strongly upregulated viral polymerase activity and played a critical role in the enhancement of virus replication and disease severity in mice / observed in dairy cattle USA 2024	Increased activity of the viral polymerases in mammalian hosts

Table A.1: Mutations of Interest from [Alvarez et al., 2025] (continued)

Locus H5n	Locus H3n	Effect	Trait
PB2:701N	N/A	Increased polymerase activity / enhanced replication efficiency / increased virulence and contact transmission in guinea pigs / increased polymerase activity in mammalian cell line / increased viral replication and virulence in mice / mammalian ANP32-specific adaptation (non-biased) / observed in seals	Increased activity of the viral polymerases in mammalian hosts
PB2:702N	N/A	Observed in human isolate / increased polymerase activity in human and avian cells	Increased activity of the viral polymerases in mammalian hosts
M1:95K	N/A	Resistant to TRIM21 / increases replication and pathogenicity in mice	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:100I	N/A	Resistance against Myxovirus resistance protein A (MxA)	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:100V	N/A	Resistance against Myxovirus resistance protein A (MxA)	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:283P	N/A	Resistance against Myxovirus resistance protein A (MxA)	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:313V	N/A	Role in Butyrophilin Subfamily 3 Member A3 (BTN3A3) evasion	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:313Y	N/A	Role in Butyrophilin Subfamily 3 Member A3 (BTN3A3) evasion	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:52H	N/A	Role in Butyrophilin Subfamily 3 Member A3 (BTN3A3) evasion	Evasion of innate immunity and counteraction of mammalian restriction factors
NP:52N	N/A	Significant mutation observed also in H5N1 / Role in Butyrophilin Subfamily 3 Member A3 (BTN3A3) evasion	Evasion of innate immunity and counteraction of mammalian restriction factors

Table A.1: Mutations of Interest from [Alvarez et al., 2025] (continued)

Locus H5n	Locus H3n	Effect	Trait
PA:336M	N/A	Significantly enhanced pathogenicity in a mouse model	Increased virulence in mammals or miscelaneous
PB2:199S	N/A	Increased virulence in mice	Increased virulence in mammals or miscelaneous
PB2:9N	N/A	Observed in domestic poultry, Uganda / increased virulence in mice	Increased virulence in mammals or miscelaneous
HA:208T	HA:212T	Increased the viral replication in avian and mammalian cells / enhance viral replication in mice	Increased virulence in mammals or miscelaneous
PA:186S	N/A	Enables GRSF1 to enhance the cytosolic accumulation and translation of a subset of viral mRNAs	Increased virulence in mammals or miscelaneous
PB2:740N	N/A	Increase polymerase activity in the presence of avian ANP32 proteins, in human and avian cells	Increased virulence in mammals or miscelaneous

Table A.2: Main drug resistance mutations from [Holmes et al., 2021] and [Mohapatra et al., 2023]

Locus H5n	Effect
NA:275Y	Resistance to antivirals: Oseltamivir and Peramivir
NA:223R	Resistance to antivirals: Zanamivir
NA:292K	Resistance to antivirals: Zanamivir
NA:136K	Resistance to antivirals: Laninamivir
PA:38T	Resistance to antivirals: Baloxavir Marboxil
PA:38M	Resistance to antivirals: Baloxavir Marboxil
PA:38F	Resistance to antivirals: Baloxavir Marboxil
PB1:229R	Resistance to antivirals: Favipiravir
M2:31N	Resistance to antivirals: Resistance to adamantanes
M2:27A	Resistance to antivirals: Resistance to adamantanes

The next tables describe extra arguments the user can use with the CLI application calls.
The following table (Table A.3) will display a summary of the main arguments cd-hit-est can accept.

Table A.3: Overview of commonly used `cd-hit-est` arguments

Argument	**Description**
`-i`	Input nucleotide FASTA file.
`-o`	Output file for clustered sequences.
`-c`	Sequence identity threshold (e.g., 0.9 for 90% identity).
`-n`	Word length (must match the identity threshold; e.g., 10 for 0.9).
`-d`	Description length in output (default is 20).
`-M`	Memory limit in megabytes (0 means unlimited).
`-T`	Number of threads to use.
`-G`	Alignment mode: 0 = global, 1 = local (default is 1).
`-A`	Alignment coverage for the shorter sequence (default is 0).
`-aL`	Alignment coverage for the longer sequence (default is 0).
`-aS`	Alignment coverage for the shorter sequence (default is 0.0).
`-g`	Use of global sequence identity (1 = more accurate but slower).
`-b`	Bandwidth of alignment (default is 20).
`-G`	Use of greedy algorithm (0 = default, 1 = greedy).
`--sc`	Score file for scoring matrix (optional).
`-sf`	Sort order: 0 = none, 1 = increasing length, 2 = decreasing length.
`-bak`	Keep backup of existing output files.
`-i2`	Input FASTA file for dataset 2 (cd-hit-est-2d only).
`--help`	Display help message and exit.

The following table (Table A.4) will summarise the main arguments that the blastn command can accept.

Table A.4: Overview of commonly used `blastn` arguments

Argument	**Description**
`-query`	Input FASTA file containing query sequences.
`-db`	BLAST nucleotide database to search against.
`-out`	Output file for BLAST results.
`-outfmt`	Output format (e.g., 0 = pairwise, 6 = tabular, 7 = XML-like tabular, 15 = JSON).
`-evalue`	Expectation value threshold for saving hits (default is 10).
`-word_size`	Word size for initial match (default: 11 for blastn).
`-gapopen`	Cost to open a gap (default: -1 means use scoring matrix default).
`-gapextend`	Cost to extend a gap.
`-penalty`	Penalty for a mismatch (negative integer).
`-reward`	Reward for a match (positive integer).
`-perc_identity`	Minimum percent identity for matches (e.g., 90).
`-max_target_seqs`	Maximum number of aligned sequences to keep (default is 500).
`-num_threads`	Number of threads to use for the search.
`-dust`	Filter for low-complexity regions in queries (yes/no or specific parameters).
`-soft_masking`	Use soft masking (masking during extension only).
`-task`	BLAST task: `blastn`, `blastn-short`, `dc-megablast`, etc.
`--help`	Show usage information and exit.

The following table (Table A.5) summarises the main call arguments for flumut.

Table A.5: Overview of `flumut` command-line arguments

Argument	**Description**
`-x`, `--excel-output`	Output results in an Excel file (.xlsm or .xlsx).
`-m`, `--markers-output`	Output a text file listing identified markers.
`-M`, `--mutations-output`	Output a text file listing identified mutations.
`-l`, `--literature-output`	Output a text file listing literature references.
`-r`, `--relaxed`	Report markers where at least one mutation is found (default reports only markers with all mutations).
`-n`, `--name-regex`	Specify a custom regular expression to parse FASTA headers.
`-D`, `--db-file`	Use a custom markers database file instead of the default.
`--update`	Update the internal markers database to the latest version.
`--skip-unmatch-names`	Skip sequences with headers that do not match the expected pattern.
`--skip-unknown-segments`	Skip sequences with segment names not present in the database.
`-v`, `--version`	Display the current version of FluMut.
`-V`, `--all-versions`	Display the versions of both FluMut and the database.

The following table (Table A.6) summarises the commands available for the CLI application call.

Table A.6: Summary of main `nextclade` CLI commands and options.

Command/Option	Description
`nextclade run`	Main command to run analysis on input sequences. Produces mutations, clade assignments, QC scores, etc.
`--input-fasta`	Specifies the input FASTA file containing viral genome sequences.
`--input-dataset`	Specifies the path or URL to the reference dataset (e.g., for SARS-CoV-2).
`--output-all`	Directory to store all output files (JSON, TSV, aligned sequences, etc.).
`--output-tsv`	Output summary results as a TSV table.
`--output-tree`	Outputs a Newick-format tree with sequences placed on the reference phylogeny.
`--output-json`	Outputs detailed analysis results in JSON format.
`--output-fasta`	Exports aligned nucleotide sequences in FASTA format.
`--output-insertions`	Outputs information about insertions as a separate TSV file.
`--output-translations`	Outputs aligned amino acid sequences for annotated genes.
`--jobs` or `-j`	Sets the number of threads for parallel processing.
`--verbose`	Enables detailed logging of the analysis process.
`--help`	Displays help message with all available options.

The next table present the contributions of GISAID-obtained sequences for the INSA-INIAV-IP IAV dataset (Table A.7).

Table A.7: GISAID sequence acknowledgments — essential fields

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392823|4B/GAAP_HA, EPI4392734|4B/GAAP_HA EPI4392822|4B/GAAP_NP, EPI4392733|4B/GAAP_NP EPI4392732|4B/GAAP_NA EPI4392731|4B/GAAP_MP EPI4392730|4B/GAAP_NS

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392742|5/GAAP_PB2 EPI4392741|5/GAAP_PB1 EPI4392740|5/GAAP_PA EPI4392739|5/GAAP_HA EPI4392738|5/GAAP_NP EPI4392737|5/GAAP_NA EPI4392736|5/GAAP_MP EPI4392735|5/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392787|17/GAAP_PB2 EPI4392786|17/GAAP_PB1 EPI4392785|17/GAAP_PA EPI4392784|17/GAAP_HA EPI4392783|17/GAAP_NP EPI4392782|17/GAAP_NA EPI4392781|17/GAAP_MP EPI4392780|17/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392771|15/GAAP_PB2 EPI4392770|15/GAAP_PB1 EPI4392769|15/GAAP_PA EPI4392768|15/GAAP_HA EPI4392767|15/GAAP_NP EPI4392766|15/GAAP_NA EPI4392765|15/GAAP_MP EPI4392764|15/GAAP_NS

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida, Henriques; Teresa, Fagulha; Fernanda, Ramos; Margarida, Duarte; Raquel, Guiomar; Aryse, Melo; Inês, Costa; Patricia, Conde; Nuno, Verdasca; Camila, Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Saúde Dr. Ricardo Jorge	EPI1950484|A/ turkey/ Lisbon/ 1/ 2021 EPI1950485|A/ turkey/ Lisbon/ 1/ 2021 EPI1950486|A/ turkey/ Lisbon/ 1/ 2021 EPI1950487|A/ turkey/ Lisbon/ 1/ 2021 EPI1950488|A/ turkey/ Lisbon/ 1/ 2021 EPI1950489|A/ turkey/ Lisbon/ 1/ 2021 EPI1950490|A/ turkey/ Lisbon/ 1/ 2021 EPI1950491|A/ turkey/ Lisbon/ 1/ 2021
Henriques, A.M.; Fagulha, T.; Ramos, F.; Duarte, A.; Duarte, M.D.; Melo, A.; Guiomar, R.	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	National Institute for Health Dr. Ricardo Jorge	EPI4386770|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386771|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386772|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386773|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386774|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386805|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386853|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021 EPI4386888|A/ turkey/ Portugal/ 39240/ 3/ GAAP/ 2021

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392803|19/GAAP_PB2 EPI4392802|19/GAAP_PB1 EPI4392801|19/GAAP_PA EPI4392800|19/GAAP_HA EPI4392799|19/GAAP_NP EPI4392798|19/GAAP_NA EPI4392797|19/GAAP_MP EPI4392796|19/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392840|9/GAAP_PB2 EPI4392839|9/GAAP_PB1 EPI4392838|9/GAAP_PA EPI4392837|9/GAAP_HA EPI4392836|9/GAAP_NP EPI4392835|9/GAAP_NA EPI4392834|9/GAAP_MP EPI4392833|9/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392846|21/GAAP_PA EPI4392845|21/GAAP_HA EPI4392844|21/GAAP_NP EPI4392843|21/GAAP_NA EPI4392842|21/GAAP_MP EPI4392841|21/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392851|22/GAAP_HA EPI4392850|22/GAAP_NP EPI4392849|22/GAAP_NA EPI4392848|22/GAAP_MP EPI4392847|22/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392858|23/GAAP_PB2 EPI4392857|23/GAAP_PA EPI4392856|23/GAAP_HA EPI4392855|23/GAAP_NP EPI4392854|23/GAAP_NA EPI4392853|23/GAAP_MP EPI4392852|23/GAAP_NS

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392866|25/GAAP_PB2 EPI4392865|25/GAAP_PB1 EPI4392864|25/GAAP_PA EPI4392863|25/GAAP_HA EPI4392862|25/GAAP_NP EPI4392861|25/GAAP_NA EPI4392860|25/GAAP_MP EPI4392859|25/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392874|12/GAAP_PB2 EPI4392873|12/GAAP_PB1 EPI4392872|12/GAAP_PA EPI4392871|12/GAAP_HA EPI4392870|12/GAAP_NP EPI4392869|12/GAAP_NA EPI4392868|12/GAAP_MP EPI4392867|12/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392878|14/GAAP_NP EPI4392877|14/GAAP_NA EPI4392876|14/GAAP_MP EPI4392875|14/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4390946|3B/GAAP_HA EPI4390945|3B/GAAP_NP EPI4390944|3B/GAAP_NA EPI4390943|3B/GAAP_MP EPI4390942|3B/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4390954|3C/GAAP_PB2 EPI4390953|3C/GAAP_PB1 EPI4390952|3C/GAAP_PA EPI4390951|3C/GAAP_HA EPI4390950|3C/GAAP_NP EPI4390949|3C/GAAP_NA EPI4390948|3C/GAAP_MP EPI4390947|3C/GAAP_NS

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4390959|4/GAAP_HA EPI4390958|4/GAAP_NP EPI4390957|4/GAAP_NA EPI4390956|4/GAAP_MP EPI4390955|4/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392827|8/GAAP_HA, EPI4392747|8/GAAP_HA EPI4392826|8/GAAP_NP, EPI4392746|8/GAAP_NP EPI4392745|8/GAAP_NA EPI4392744|8/GAAP_MP EPI4392743|8/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392831|10/GAAP_HA, EPI4392752|10/GAAP_HA EPI4392830|10/GAAP_NP, EPI4392751|10/GAAP_NP EPI4392750|10/GAAP_NA EPI4392749|10/GAAP_MP EPI4392748|10/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392832|11/GAAP_NP, EPI4392755|11/GAAP_NP EPI4392754|11/GAAP_MP EPI4392753|11/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392763|13/GAAP_PB2 EPI4392762|13/GAAP_PB1 EPI4392761|13/GAAP_PA EPI4392760|13/GAAP_HA EPI4392759|13/GAAP_NP EPI4392758|13/GAAP_NA EPI4392757|13/GAAP_MP EPI4392756|13/GAAP_NS

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392779|16/GAAP_PB2 EPI4392778|16/GAAP_PB1 EPI4392777|16/GAAP_PA EPI4392776|16/GAAP_HA EPI4392775|16/GAAP_NP EPI4392774|16/GAAP_NA EPI4392773|16/GAAP_MP EPI4392772|16/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392795|18/GAAP_PB2 EPI4392794|18/GAAP_PB1 EPI4392793|18/GAAP_PA EPI4392792|18/GAAP_HA EPI4392791|18/GAAP_NP EPI4392790|18/GAAP_NA EPI4392789|18/GAAP_MP EPI4392788|18/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392811|24/GAAP_PB2 EPI4392810|24/GAAP_PB1 EPI4392809|24/GAAP_PA EPI4392808|24/GAAP_HA EPI4392807|24/GAAP_NP EPI4392806|24/GAAP_NA EPI4392805|24/GAAP_MP EPI4392804|24/GAAP_NS
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392825|7B/GAAP_HA, EPI4392816|7B/GAAP_HA EPI4392824|7B/GAAP_NP, EPI4392815|7B/GAAP_NP EPI4392814|7B/GAAP_NA EPI4392813|7B/GAAP_MP EPI4392812|7B/GAAP_NS

Continued on next page

Authors	Originating Laboratory	Submitting Laboratory	GISAID Accession ID(s)
Ana Margarida Henriques	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	Instituto Nacional de Investigacao Agraria e Veterinaria (INIAV)	EPI4392829|7E/GAAP_HA, EPI4392821|7E/GAAP_HA EPI4392828|7E/GAAP_NP, EPI4392820|7E/GAAP_NP EPI4392819|7E/GAAP_NA EPI4392818|7E/GAAP_MP EPI4392817|7E/GAAP_NS
Ana Margarida, Henriques; Teresa, Fagulha; Ana Duarte; Fernanda, Ramos; Margarida, Duarte; Raquel, Guiomar; Aryse, Melo;	Instituto Nacional de Investigação Agrária e Veterinária, I. P.	National Institute for Health Dr. Ricardo Jorge	EPI4390937|A/ turkey/ Portugal/ 38582/ 2/ GAAP/ 2021 EPI4390938|A/ turkey/ Portugal/ 38582/ 2/ GAAP/ 2021 EPI4390939|A/ turkey/ Portugal/ 38582/ 2/ GAAP/ 2021 EPI4390940|A/ turkey/ Portugal/ 38582/ 2/ GAAP/ 2021 EPI4390941|A/ turkey/ Portugal/ 38582/ 2/ GAAP/ 2021