UNIVERSIDAD DE LA REPÚBLICA URUGUAY

PEDECIBA MEC-UDELAR

TESIS DE MAESTRÍA EN BIOINFORMÁTICA

Herramientas para la vigilancia epidemiológica del virus de Influenza tipo A mediante aprendizaje automático

---

Autor: PÉREZ, Ramiro Andres

DIRECTORES DE TESIS: Flavio Pazos, Alvaro Fajardo

MONTEVIDEO, URUGUAY

Diciembre 2025

Agradecimientos

Muchas gracias a todas las instituciones que hicieron posible este trabajo: la Universidad de la República, PEDECIBA, el Institut Pasteur de Montevideo y el MGAP, en particular a DILAVE.

A Flavio, por su paciencia, confianza y constante impulso. En nuestras charlas siempre encontré un consejo acertado y la motivación constante.

A Álvaro, por su paciencia y confianza, y por enriquecer este trabajo a kilómetros de distancia.

A mis amigos y familia, por estar en todo momento.

A personas como **PG**, que me dieron oportunidades cuando más lo necesitaba.

Índice General

1. Resumen..... 8
2. Introducción.....9
2.1. Virus de influenza tipo A..... 9
2.2. Epidemiología y situación actual..... 10
2.3. Diagnóstico de Influenza.....12
2.4. Aplicaciones de Machine Learning.....13
2.4.1. Secuencias de ADN.....15
2.4.2. Secuencias de ARN..... 16
2.4.3. Secuencias de AA..... 16
2.5. Antecedentes en la clasificación de hospederos de influenza.....17
2.6. Desafíos en la identificación de hospederos..... 19
3. Objetivos..... 21
2.3. Objetivo General..... 21
2.4. Objetivos específicos.....21
4. Metodología general..... 22
4.1. Esquema general de trabajo.....22
4.2. Herramientas Bioinformáticas.....25
4.3. Datasets utilizados en el trabajo..... 25
4.4. Extracción de Características.....29
4.4.1. PSSM_AC (Position-Specific Scoring Matrix - Autocorrelation)..... 30
4.4.2. AAC (Amino Acid Composition).....32
4.4.3. DPC (Dipeptide Composition).....32
4.4.4. GDPC-Composición de Dipéptidos Agrupados..... 33
4.4.5. Autocorrelación de Moran.....33
4.4.6. PAAC (Pseudo Amino Acid Composition).....34
4.4.7. APAAC (Amphiphilic Pseudo Amino Acid Composition).....36
4.5. Algoritmos de aprendizaje.....37
4.5.1. Modelos no supervisados..... 37
4.5.2. Modelos supervisados.....38
4.6. Métricas de evaluación.....41
4.7. Implementación de modelos.....42
4.7.1. Problema de Clasificación Multiclase.....42
4.7.2. Modelo Uno-vs-Resto.....42
4.8. Limitaciones.....44

5. Capítulo 1 45
Modelos de clasificación de hospederos e identificación de regiones funcionales relevantes en la hemaglutinina (HA) 45
5.1. Resumen 45
5.2. Introducción 46
5.3. Objetivos 46
5.3.1. Objetivo General 46
5.4. Metodología 47
5.4.1. Conjunto de entrenamiento 47
5.4.2. Entrenamiento y ajuste de hiperparametros 48
5.4.3. Evaluación de modelos 49
5.4.4. Interpretación de características 49
5.5. Resultados 51
5.5.1. Exploración inicial de datos 51
5.5.1.1. PCA 51
5.5.1.2. t-SNE 55
5.5.2. Clustering 56
5.5.3. Desempeño de los modelos supervisados 60
5.5.4. Importancia de las características 66
5.5.5. Secuencias ambiguas 69
5.6. Discusión 70
5.6.1. PCA, t-SNE y Clustering 70
5.6.2. Modelos Supervisados 71
5.6.3. Importancia de características 75
5.7. Conclusiones 76
6. Capítulo 2 77
Clasificación del subtipo de HA y patogenicidad 77
6.1. Resumen 77
6.2. Introducción 78
6.3. Objetivos 79
6.3.1. Objetivo general 79
6.3.2. Objetivo específicos 79
6.4. Metodología 79
6.4.1. Conjunto de entrenamiento y Test 80
6.4.2. Entrenamiento y ajuste de hiperparámetros 80
6.4.3. Selección de modelo y descriptor 81
6.4.4. Patogenicidad 81
6.5. Resultados 82
6.5.1. Modelos supervisados 82

6.5.2. Detección de variantes de alta o baja patogenicidad..... 84
6.5.3. Plataforma interactiva..... 86
6.6. Discusión..... 86
6.7. Conclusiones..... 88
7. Conclusiones finales..... 88
8. Bibliografía..... 89
9. Anexos..... 102
9.1. Anexo I..... 102
9.2. Anexo II..... 104
9.3. Anexo III..... 107
9.4. Anexo IV..... 109
9.5. Anexo V..... 114

Lista de abreviaciones

A-a-I	Adenosia-a-inosina
AA	Aprendizaje automático
AAC	Amino Acid Composition (composición aminoacídica)
AC	Autocorrelación
ADN	Ácido desoxirribonucleico
APAAC	Amphiphilic Pseudo Amino Acid Composition (pseudo composición anfifílica)
AUC-PR	Área bajo la curva de Precisión-Recobrado (Precision-Recall)
BLAST	Basic Local Alignment Search Tool
BLASTp	BLAST de proteínas
CNN	Red neuronal convolucional
DL	Deep Learning
DPC	Di-peptide Composition (composición dipeptídica)
DT	Árbol de decisión
FAO	Food and Agriculture Organization of the United Nations
F1	F1-score (medida F1)
FASTA	Formato FASTA
GDPC	Grouped Di-peptide Composition (composición dipeptídica agrupada)
GMM	Modelo de mezclas gaussianas
HA	Hemaglutinina
HPAI	Highly Pathogenic Avian Influenza (influenza aviar altamente patógena)
IA	Inteligencia Artificial
KNN	k vecinos más cercanos
LGBM	Light Gradient Boosting Machine

LPAI	Low Pathogenic Avian Influenza (influenza aviar de baja patogenicidad)
MCC	Coeficiente de correlación de Matthews
ML	Machine Learning
NA	Neuraminidasa
NCBI	National Center for Biotechnology Information
OIE	World Organisation for Animal Health (nombre anterior)
OFFLU	Red OFFLU de influenza animal (FAO/WOAH)
PAAC	Pseudo Amino Acid Composition (pseudo composición aminoacídica)
PCA	Análisis de componentes principales
PFM	Position Frequency Matrix (matriz de frecuencias por posición)
PR	Precision–Recall (precisión–sensibilidad)
PSI-BLAST	Position-Specific Iterative BLAST
PSSM	Position-Specific Scoring Matrix
PSSM_AC	Position-Specific Scoring Matrix Autocorrelation
RBPs	RNA-binding proteins (proteínas de unión a ARN)
RF	Random Forest
RT-PCR	Reacción en cadena de la polimerasa con transcriptasa reversa
SVM	Support Vector Machine
t-SNE	t-distributed Stochastic Neighbor Embedding
VIA	Virus de influenza tipo A
WOAH	World Organisation for Animal Health (ex OIE)
XGBoost	Xtreme Gradient Boosting

1. Resumen

Este trabajo aborda la implementación de modelos de aprendizaje automático para la clasificación del virus Influenza A (VIA) a partir de secuencias de su proteína hemaglutinina (HA), con tres objetivos centrales: identificar el hospedero de origen, predecir el subtipo de HA y detectar automáticamente la patogenicidad viral (HPAI o LPAI). Esta información es clave para la vigilancia genómica y la toma de decisiones sanitarias rápidas ante brotes del virus.

En una primera etapa, se desarrollaron y evaluaron modelos supervisados y no supervisados para la clasificación del hospedero (aves, humanos, cerdos), utilizando distintos descriptores de secuencia. Se aplicaron técnicas de reducción de dimensionalidad (PCA, t-SNE) y clustering (KMeans, GMM), que permitieron observar agrupamientos coherentes con los subtipos, especialmente usando los descriptores DPC y Moran. Para desarrollar el clasificador de hospedero se entrenaron modelos supervisados (KNN, SVM, Random Forest, XGBoost), siendo KNN-DPC el modelo más robusto y generalizable, especialmente en la clasificación de secuencias parciales. El análisis de importancia de características reveló regiones funcionales específicas de HA asociadas diferencialmente a cada hospedero, como HR2, epítopo Ca2 y dominio transmembrana en aves; epítopo Sb en humanos; y sitios de clivaje y reconocimiento de ácido siálico en cerdos.

En la segunda etapa, se abordó la clasificación del subtipo de HA (entre 16 subtipos) y la detección de motivos de alta o baja patogenicidad. Para ello, se desarrollaron modelos supervisados utilizando nuevamente el descriptor DPC. El algoritmo SVM mostró el mejor desempeño general, manteniendo altos valores de F1-macro y AUC-PR incluso sobre secuencias parciales no contenidas en el entrenamiento. Adicionalmente, se diseñó un script en Python que permite identificar automáticamente la presencia de motivos multibásicos en el sitio de clivaje HA1/HA2, asociados a alta patogenicidad en subtipos H5 y H7. Este script fue validado con secuencias de la epizootia ocurrida en Uruguay (2023), clasificando correctamente todas las variantes como HPAI del clado 2.3.4.

Finalmente, los modelos entrenados y el script de patogenicidad se integraron en una plataforma interactiva desarrollada con Streamlit, capaz de procesar secuencias de HA y predecir en tiempo real el hospedero, subtipo y nivel de patogenicidad, constituyendo una herramienta ágil para la vigilancia molecular.

En conjunto, estos capítulos demuestran que el uso de descriptores proteicos adecuados, como DPC, combinado con modelos robustos de aprendizaje automático, como SVM y KNN, permite extraer información biológicamente relevante a partir de secuencias virales, con aplicaciones directas en la vigilancia genómica del virus Influenza A. Además, los análisis funcionales aportan evidencia sobre regiones moleculares vinculadas al tropismo y la adaptación interespecie, reforzando el valor de estos enfoques en virología y salud pública.

2. Introducción

2.1. Virus de influenza tipo A

El virus de influenza tipo A (VIA), un miembro de la familia *Orthomyxoviridae* y la única especie del género *Alphainfluenzavirus*, es un importante patógeno que afecta tanto mamíferos como aves. Es un virus altamente versátil, con una notable capacidad de mutación y adaptación que le permite infectar a diversas especies. La gripe, enfermedad infecciosa generada por este virus, tiene una gran importancia a nivel de salud animal y humana. Dentro de la población humana tiene alta prevalencia y se estima que esta tiende a subestimarse. En comparación con las epidemias de influenza estacionales, las pandemias por influenza son menos frecuentes, pero tienden a dejar millones de muertes (Long et al., 2019).

Los virus de la influenza se clasifican en cuatro tipos en base a su ribonucleoproteína interna: A, B, C y D. El virus de la influenza D no causa enfermedad en humanos. El virus de la influenza C es únicamente infeccioso en humanos; sin embargo, es poco probable que dé lugar a epidemias a gran escala. Por estos motivos las vacunas estacionales contra la influenza no incluyen cepas de influenza C ni D. Las epidemias estacionales son causadas principalmente por los virus de influenza A y B. El virus de la influenza B es infeccioso solo para humanos, mientras que el virus de la influenza A es infeccioso tanto para humanos como para animales, y puede ocasionar epidemias globales (por ejemplo, pandemias)(Cox et al., 2004).

Existen dos glicoproteínas de la envoltura del virus que distinguen a los subtipos del VIA: la hemaglutinina (HA) y la neuraminidasa (NA). Hasta la fecha se han identificado 18 subtipos de HA (H1-H18) y 11 subtipos de NA (N1-N11) (Lazniewski et al., 2018). Estas dos proteínas, se encuentran en la envoltura viral y están involucradas en los primeros pasos de la entrada del virus a la célula. A su vez son muy inmunogénicas, provocando que el organismo active su sistema inmunológico para evitar la infección del virus. Los sitios antigénicos de estas proteínas, que son reconocidos por el sistema inmune, tienen la particularidad de poder cambiar, debido a dos características de este virus, su alta tasa de mutación y la capacidad de reordenamiento de sus segmentos. Cuando el cambio en estos sitios antigénicos impide su reconocimiento por el sistema

inmune, se denomina deriva antigénica (*antigenic drift*), lo que puede dar lugar a variantes responsables de epidemias estacionales (Cox *et al*., 2004). Asimismo, la co-infección de un hospedador con dos VIA diferentes puede dar lugar al reordenamiento de sus segmentos genómicos, fenómeno conocido como cambio antigénico mayor (*antigenic shift*). Este proceso puede originar la aparición de nuevos subtipos con potencial pandémico (Brockwell-Staats et al., 2009).

2.2. Epidemiología y situación actual

El VIA puede infectar a diversos hospedadores, incluidos humanos, aves y cerdos. Las aves constituyen un importante reservorio natural para el virus (Long et al., 2019; Gorman et al., 1990). Además, se considera que los cerdos son un hospedador intermedio entre humanos y aves (Brown, 2001).

La transmisión de los virus de influenza puede darse de animal a humano (zoonosis) así como de animal a animal (enzoótica) (Long et al., 2019). Las infecciones zoonóticas pueden ser transmisiones de fin de línea o pueden conducir a una pandemia en la población humana, si acumulan suficientes mutaciones adaptativas que permitan la transmisión del virus entre personas, que luego circula de forma regular como una influenza estacional (Taubenberger y Kash, 2010). Dada esta capacidad del virus para cruzar fronteras entre especies, el origen de cada brote es difícil de determinar. Por otro lado, el proceso de mutación adaptativa y acumulación de cambios necesita tiempo (Long et al., 2019). La identificación temprana del hospedador viral original puede ayudar a prevenir o controlar efectivamente la propagación de un brote viral.

Históricamente, varias pandemias han sido consecuencia de cambios antigénicos extremos, frente a los cuales la población humana no tenía inmunidad preexistente, facilitando su expansión global. Como resultado del reordenamiento génico entre virus animales (porcina y aviar) y virus humanos, han surgido cuatro grandes pandemias de influenza desde 1900 (ver Figura 1) : la gripe española H1N1 (1918–1919), la gripe asiática H2N2 (1957–1958), la gripe de Hong Kong (1968–1969) y la pandemia de gripe de 2009 (2009–2010) (Taubenberger y Kash, 2010).

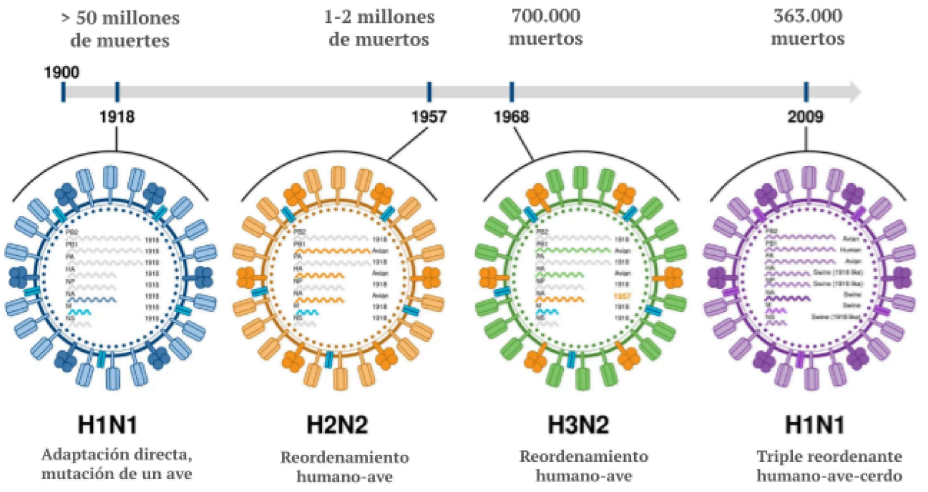


Figura 1. Representación de las variantes de los virus de influenza tipo A causantes de las cuatro mayores pandemias. Imagen adaptada de Harrington et al., 2021.

Los VIA se clasifican según su patogenicidad en dos categorías: altamente patógenos (HPAI) y de baja patogenicidad (LPAI). Se determina identificando, en el sitio de clivaje HA1/HA2 de la HA, ciertos motivos de aminoácidos monobásicos (en el caso de LPAI) o polibásicos (HPAI). Los HPAI pueden causar una mortalidad en la población infectada de hasta el 100%, y se han asociado con los subtipos H5 y H7, aunque no todos los virus H5 y H7 son HPAI. Los demás virus se consideran de baja patogenicidad y suelen causar una enfermedad respiratoria más leve (D.J. Alexander 2007).

En 1996 ocurrió un cambio significativo en la epidemiología del VIA con la aparición de este linaje H5 altamente patogénico. Inicialmente circulaba solo en granjas domésticas, luego este linaje se volvió enzoótico en aves silvestres y evolucionó en distintos clados genéticos y antigénicos (Xu et al., 1999). En el año 2020, surgió el clado 2.3.4.4b H5N1 y comenzó a propagarse a muchas partes del mundo, incluyendo África, Asia, Europa, América del Norte y del Sur. Este clado ha cambiado la epidemiología de la HPAI, causando mortalidad masiva en poblaciones de aves silvestres y de corral (Gilbertson y Subbarao 2023). Es notable que también ha causado infecciones en diversos mamíferos marinos y terrestres en una gran variedad de especies (ver Figura 2).

Esta situación es de preocupación ya que sugiere que el virus estaría desarrollando adaptaciones genéticas que promueven un mayor *fitness* en hospederos mamíferos, incrementando el riesgo de transmisión zoonótica (Agüero et al. 2023; Puryear et al. 2023; Pardo-Roa et al. 2023).

Durante 2020, los virus de influenza aviar altamente patógena (HPAI) A(H5N1) del clado 2.3.4.4b surgieron a partir de virus A(H5Nx) que circulaban previamente y se

diseminaron predominantemente a través de aves migratorias hacia muchas regiones de África, Asia y Europa. La epizootia ha provocado un número sin precedentes de muertes en aves silvestres y ha causado brotes en aves de corral domésticas. A finales de 2021, estos virus llegaron a Norteamérica y posteriormente a Sudamérica en el otoño de 2022 (World Health Organization, 2022).

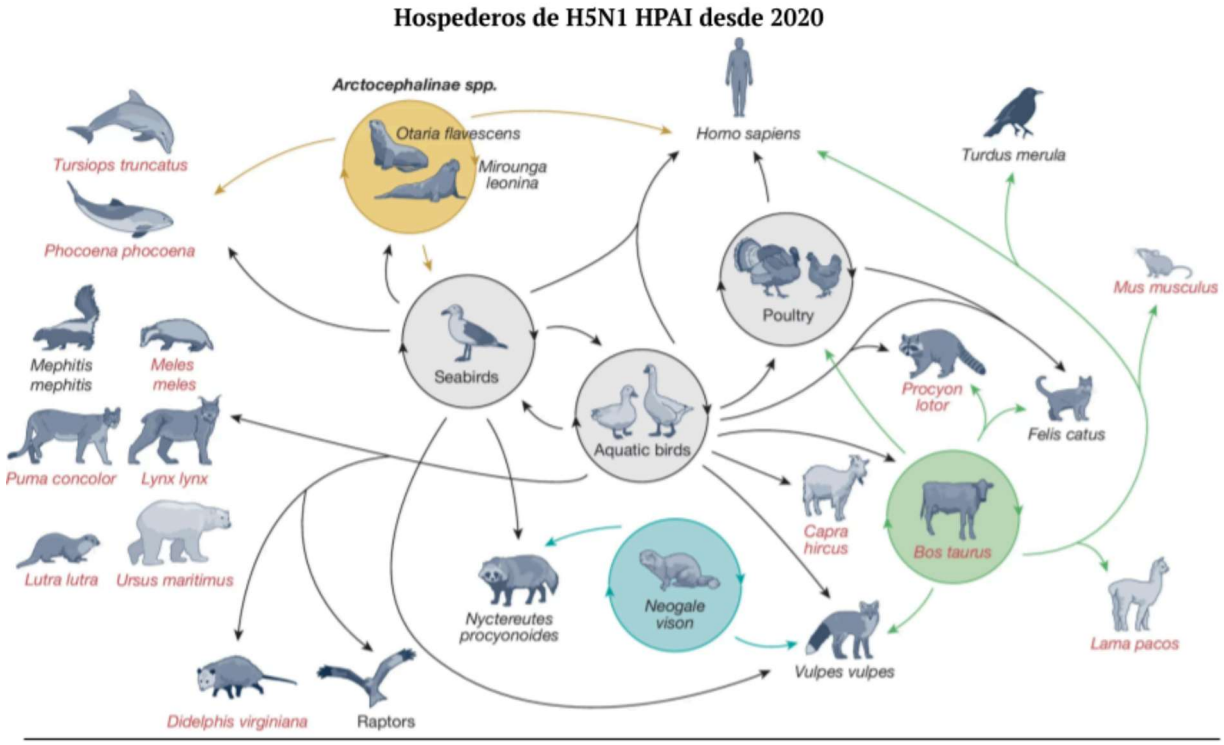


Figura 2. Las flechas indican eventos de derrame (*spillover*) hacia otras especies hospedadoras. Las flechas circulares indican transmisión sostenida de H5N1 en esa especie hospedadora. Los nuevos hospedadores mamíferos de H5N1 con transmisión sostenida se resaltan en amarillo (mamíferos marinos de Sudamérica), verde (ganado lechero de EE. UU.) y azul (visón europeo), con flechas de los mismos colores que muestran derrames desde esos brotes mamíferos hacia especies adicionales, posiblemente a través de intermediarios no muestreados. Los animales etiquetados en rojo son especies hospedadoras en las que se detectó IAV por primera vez durante este brote (basado en datos de secuencias genéticas, no en serología). Extraida de Peacock et al., 2024.

En Uruguay, el primer caso del clado 2.3.4.4b de gripe aviar altamente patógena H5N1 fue diagnosticado en febrero de 2023 en cisnes de cuello negro (*Cygnus melancoryphus*) en una laguna costera atlántica que recibe miles de aves migratorias y alberga una diversidad de especies de aves acuáticas (Marandino et al. 2023). El impacto de este clado se intensificó en marzo de 2024 cuando emergió en vacas lecheras en Texas, USA el genotipo B3.13 , un reordenante con variantes de aves de Norteamérica (Lowen et al., 2025; Oguzie et al., 2024). Este evento tuvo como consecuencia reciente la infección en humanos, ya que durante un brote en el estado de Colorado se infectaron trabajadores de una granja avícola (Vaidyanathan et al., 2024).

Este escenario destaca la urgencia de la necesidad de coordinar respuestas basadas en el abordaje de “Una Salud”, reconociendo el vínculo entre el ambiente, el humano y los animales (Koopmans et al., 2024). Es fundamental implementar un sistema de vigilancia epidemiológica robusto, que permita el monitoreo de diversas variables así como herramientas de interpretabilidad como son los modelos de aprendizaje automático para entender la evolución y circulación de los virus, a fin de reducir el riesgo de transmisión a humanos y respaldar la toma de decisiones en salud pública.

2.3. Diagnóstico de Influenza

Históricamente se han utilizado diferentes técnicas serológicas, como ELISA, ensayos de inhibición de la hemaglutinación, cultivos celulares, que han sido efectivas en el diagnóstico y la subtipificación de cepas de VIA así como otros virus (Kennedy, 2005). Más recientemente se han incorporado al diagnóstico técnicas moleculares (como RT-PCR o qRT-PCR), así como estrategias de secuenciación, que han mejorado significativamente la velocidad, sensibilidad y precisión en el diagnóstico. En los últimos años la reducción del costo de estas tecnologías y el mejoramiento en su versatilidad han permitido un aumento significativo de datos genómicos.

Si bien ha habido mejora en la eficiencia de los métodos de diagnóstico, por otro lado se han visto muy limitados en cuanto a velocidad, escalamiento y en algunos casos especificidad. Procesos de diagnósticos lentos en un escenario de pandemia pueden significar la diferencia entre contener o no transmisión viral. Del mismo modo, los tratamientos ineficientes pueden dar lugar a tratamientos prolongados, presionando la infraestructura del sistema de salud e incrementando la morbilidad.

2.4. Aplicaciones de Machine Learning

Para enfrentar estos desafíos, la intersección de la epidemiología y la virología clínica con abordajes basados en Inteligencia Artificial (IA) en general y en Aprendizaje Automático (*Machine Learning*, ML) en particular, es prometedora. Estos abordajes permiten desarrollar herramientas capaces de manejar grandes volúmenes de datos, reconocer patrones complejos y aprender de ellos, ofreciendo soluciones rápidas y precisas.

La IA busca simular procesos que implican inteligencia humana mediante máquinas. Cuando esto implica algoritmos que pueden aprender y mejorar a partir de

datos, se habla de ML. El aprendizaje profundo (*Deep Learning*, DL), un subcampo del ML, utiliza redes neuronales con muchas capas (de ahí el término "profundo") para analizar diversos aspectos de los datos. En el contexto de la virología clínica y epidemiología, estas herramientas pueden aprovecharse para una amplia variedad de aplicaciones, que van desde el diagnóstico hasta la predicción de epidemias (Padhi et al., 2023).

Una de las grandes ventajas de integrar la IA a la virología es su incomparable fuerza en el reconocimiento de patrones. Los métodos de diagnóstico tradicionales a menudo dependen de la observación minuciosa y análisis por parte de los técnicos especializados. Si bien el análisis de expertos es invaluable, también tiene sus limitaciones, incluyendo la susceptibilidad al error y la imposibilidad de procesar grandes volúmenes de datos.

Es aquí donde algoritmos de ML y de DL, entran en juego. Si existen datos suficientes, los algoritmos de ML pueden ser entrenados para reconocer marcadores o características de los datos que permitan identificar presencia o ausencia de infecciones virales u otra información relevante del agente patógeno en general. Pueden procesar rápidamente grandes volúmenes de datos de distinta naturaleza, como secuencias de nucleótidos en genomas virales o antígenos específicos en muestras de sangre, para identificar patrones que sean indicativos de una cepa viral determinada o de una etapa específica de una infección u otros datos epidemiológicos relevantes. Esto no sólo acelera el proceso diagnóstico, sino que también aumenta su precisión, aporta información para tomar decisiones y minimiza el riesgo de resultados erróneos.

Consideremos el enorme desafío que conlleva la vigilancia epidemiológica de VIA. Tradicionalmente, esta tarea se basa en gran medida en los casos reportados (que suelen estar subrepresentados), una medida reactiva que provoca retrasos en las estrategias de respuesta. En este sentido el poder de la IA para abordar proactivamente este problema fue puesto de manifiesto por un estudio pionero (Ginsberg et al., 2009), que al combinar datos de atención sanitaria con búsquedas en Google, pudo predecir las tendencias de la influenza con una precisión notable. Esta metodología no solo es rápida, sino que también opera en tiempo real, lo que permite respuestas inmediatas en salud pública, un privilegio que los métodos tradicionales rara vez ofrecen .

Los enfoques epidemiológicos tradicionales han dependido de la recolección manual de datos y de análisis estadísticos que consumen mucho tiempo, lo que a menudo provoca demoras en la detección y respuesta ante amenazas emergentes. La IA está preparada para revolucionar este escenario al incorporar automatización, escalabilidad y

un poder computacional sin precedentes en la vigilancia de la salud pública y la predicción de epidemias desde varios ángulos. Un aspecto importante a tener en consideración es que estos modelos de ML son tan buenos como los datos con los que fueron entrenados. Si los datos de entrenamiento están sesgados las predicciones de estos algoritmos también lo estarán (Ao et al., 2022).

La continuidad y la rapidez con la que se desarrollan las tecnologías de secuenciación en la era post-genómica ha producido un incremento exponencial de datos de secuencias. Con este continuo crecimiento los investigadores esperan comprender el significado biológico en estos datos por medio de la minería y análisis con métodos computacionales adecuados. La clasificación de estas secuencias biológicas (Lambert et al., 2018), incluye muchas líneas de investigación, identificar sus funciones y modificaciones es una de ellas, involucrando DNA, RNA y secuencias de aminoácidos.

El costo de esta tecnología y el mejoramiento en su versatilidad han permitido un aumento significativo de datos genómicos. Explorar estos datos y obtener información valiosa a partir de ellos es un desafío y es en esta área que los algoritmos de aprendizaje automático (AA) han adquirido relevancia, al poder manejar grandes volúmenes de datos complejos y abordar varios desafíos, como la identificación de patrones encubiertos, la asignación de clases y la generación de modelos predictivos, entre otros ((Borkenhagen L.K et al., 2021).

Las técnicas de aprendizaje automático se han aplicado ampliamente en la investigación de las disciplinas ómicas. Estas se dividen en dos grandes categorías: supervisadas y no supervisadas. En el aprendizaje supervisado, el objetivo principal es predecir una etiqueta (clasificación) o un valor (regresión) para cada observación utilizando un conjunto de ejemplos previamente etiquetados. Estos abordajes requieren de datos con etiquetas conocidas (por ejemplo, asignación de una característica funcional o categoría biológica) que puedan emplearse para predecir etiquetas en datos con etiqueta desconocida. Por otro lado, los métodos no supervisados extraen patrones y estructuras inherentes en los datos sin necesidad de etiquetas. El aprendizaje no supervisado incluye técnicas como el clustering o el análisis de componentes principales, que buscan identificar patrones subyacentes en los conjuntos de datos.

El desarrollo del ML, minería de datos y tecnologías asociadas en el campo de la ciencias computacionales ha promovido la investigación en el área de análisis de datos de secuencias biológicas. Métodos basados en ML para predecir o analizar las funciones biológicas se han vuelto muy populares en los últimos años gracias a su eficiencia. Los

algoritmos de ML tradicionales más utilizados en este campo incluyen Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), Regresión Logística (LR), Árbol de decisión (DT), Light Gradient Boosting Machine (LGBM) y Extreme Gradient boosting (XGBoost). El aprendizaje profundo también ha visto un uso creciente en la investigación sobre clasificación de secuencias biológicas (Lv et al., 2019; Cui et al., 2021).

2.4.1. Secuencias de ADN

La clasificación de secuencias de ADN es necesaria para comprender la función biológica de estas moléculas. Los métodos de clasificación de secuencias desarrollados han crecido con la proliferación de datos de secuencias de ADN (He et al., 2019, Lyu et al., 2020). Los métodos de clasificación basados en secuencias de ADN y los datos relacionados se dividen en dos categorías según la longitud de la secuencia: métodos de predicción generales (secuencias de longitud desigual) y métodos de predicción de sitios de modificación (secuencias de igual longitud). Los métodos de clasificación que trabajan con secuencias de diferente largo incluyen ejemplos en los que se busca identificar potenciadores de ADN, promotores, nucleosomas, o que buscan asignar funciones biológicas a cada secuencia de ADN. Los métodos de clasificación que trabajan con secuencias de igual largo incluyen ejemplos en los que se busca identificar modificaciones como N6-metiladenina (6mA), N4-metilcitosina (4mC) y 5-metilcitosina (5mC).

Por ejemplo, un predictor de dos capas desarrollado recientemente, iEnhancer-XG (Cai *et al.*, 2021), integra una variedad de características derivadas de la secuencia y utiliza aprendizaje integrado para clasificar potenciadores de ADN. iEnhancer-XG utilizar cinco métodos de extracción de características, incluidos PSSM, perfil k-spectrum, k-tuple con desajuste (*mismatch k-tuple*), PseDNC y perfil de subsecuencia, y combina la entrada de aprendizaje integrado de cinco vectores de características de salida separados, utilizando XGBoost como clasificador. El conjunto de datos de entrenamiento contiene 742 potenciadores fuertes, 742 potenciadores débiles y 1.484 no potenciadores. Se construyó un conjunto de prueba independiente (100 potenciadores fuertes, 100 potenciadores débiles y 200 no potenciadores para verificar el rendimiento del modelo). La precisión del clasificador evaluada sobre los conjuntos de prueba independiente fue de 81,10% y 75,75%, y los obtenidos por el clasificador de segunda capa fueron de 66,74% y 63,50%.

2.4.2. Secuencias de ARN

El ARN participa en las actividades vitales y desempeña funciones biológicas específicas en las células. La clasificación de secuencias de ARN juega un papel crucial para entender sus funciones biológicas y sus mecanismos de modificación. Existen varios trabajos que han utilizado ARN como input de estos modelos (Liu & Chen, 2020; Li *et al*., 2021).

La clasificación de secuencias de ARN se divide en dos categorías: (a) secuencias de largo desigual (utilizadas para la clasificación de ARNm eucariota, ARN no codificante, pre-microARN, ARN circular (circARN), ARN de transferencia (ARNt), ARN que interactúa con Piwi, ARN largo no codificante (lncARN) y ARN guía sencillo) y (b) secuencias de igual longitud (utilizadas para la clasificación de sitios de modificación postranscripcional de ARN m6A, N1-metiladenosina (m1A), pseudouridina (Ψ), adenosina-a-inosina (A-a-I), 5-metilcitosina (m5C), N2-metilguanosina (m2G), 2′-O-metilación (Nm), 5-hidroximetilcitosina (5hmC), dihidrouridina (D), N6,2′-O-dimetiladenosina (m6Am), N7-metilguanosina (m7G), 5-metiluridina (m5U) e inosina (I)). Un ejemplo de clasificación de secuencias de ARN de longitud desigual es SubLocEP (Li *et al*., 2021), un predictor desarrollado con ML para asignar localización subcelular de ARNm eucariota. Este método utiliza aprendizaje automático integrado para clasificar ARNm entre citoplasma, retículo endoplasmático, región extracelular, mitocondrias y núcleo. El conjunto de datos de entrenamiento incluye: citoplasma: 5.310; retículo endoplasmático: 1.185; región extracelular: 710; mitocondrias: 350; y núcleo: 4.855. El conjunto de prueba independiente contiene 3 conjuntos de datos, a saber, D1 (citoplasma: 1.066; retículo endoplasmático: 241; región extracelular: 145; mitocondrias: 71; y núcleo: 976), D2 (citoplasma: 91 y núcleo: 148) y D3 (retículo endoplasmático: 131 y núcleo: 131). La exactitud utilizando este método fue baja, de 0,659 ± 0,006. Las exactitudes de los 3 conjuntos de prueba independientes también fueron bajas: 0,601, 0,506 y 0,37.

2.4.3. Secuencias de AA

Las proteínas desempeñan papeles importantes en las células de los organismos, incluyendo la catálisis de reacciones, el reconocimiento, la regulación, la señalización celular, el transporte a través de membranas y el aporte de estructura (Bonetta & Valentino, 2020). A medida que la brecha entre el número de proteínas que se descubren y su caracterización funcional crece (especialmente debido a limitaciones experimentales), la predicción confiable de la función proteica por medios computacionales se ha vuelto

crucial (Niu et al., 2021). Los problemas comunes de clasificación de proteínas consideran secuencias de longitud igual o desigual.

Ejemplos de tareas de clasificación en las que se trabaja con secuencias de longitud desigual incluyen la identificación de proteínas de unión a ADN o ARN (DBPs/RBPs), proteínas secretadas, cancerlectinas, proteínas virales (incluidos viriones y fagos), enzimas líticas de pared celular, proteínas termofílicas, componentes del complejo mayor de histocompatibilidad (MHC), proteínas antioxidantes, proteínas bioluminiscentes, proteínas de transporte de electrones y sistemas de secreción tipo III (Niu et al., 2021).

La clasificación de secuencias proteicas de igual longitud está relacionada con la predicción de sitios de modificación postraduccional (PTM) de proteínas. Las PTM de proteínas implican la modificación o adición de grupos químicos que ocurren sobre residuos de aminoácidos. Un ejemplo en clasificación de proteínas utilizando ML que ha sido de interés en los últimos tiempos, tanto por la cantidad de datos que se han generado como los beneficios en la salud pública que conlleva el buen uso de los mismos, es la utilización de proteínas virales como input que permitan evidenciar ciertas características de los casos o brotes, como son la fuente de infección, patogenicidad, información sobre la evolución del patógeno, información epidemiológica e inclusive resistencia a drogas antivirales (Borkenhagen et al., 2021).

Hay dos estudios donde se codificaron numéricamente secuencias de nucleótidos similares a las que provocaron la pandemia de H1N1 de 2009. En estos estudios luego de reducir la dimensionalidad de las secuencias mediante análisis de componentes principales (PCA) se obtuvieron predicciones con árboles de decisión y redes neuronales. En otro caso (Shaltout e*t al.*, en 2015), se analizó la resistencia al inhibidor del canal iónico M2, adamantano. Los mejores resultados se obtuvieron con el árbol de decisión entrenado con secuencias M, que produjo una exactitud en prueba de 0,982, una sensibilidad de 0,980, una especificidad de 0,986 y una precisión de 0,973. El segundo estudio analizó la resistencia al inhibidor de la NA, oseltamivir (Shaltout e*t al.*, en 2016). En contraste con los hallazgos del estudio anterior, encontró que una red neuronal entrenada con secuencias de NA fue el predictor más exitoso, con una exactitud sobre la muestra de evaluación de 0,983, una sensibilidad de 0,980, una especificidad de 0,985 y una precisión de 0,98.

2.5.Antecedentes en la clasificación de hospederos de influenza

La proteína HA está involucrada en el primer contacto con la célula diana, interactuando con la membrana de la célula, permitiendo el reconocimiento de proteínas específicas en cada hospedero (residuos de ácido siálico) y la internalización del virión. Existen varios marcadores en esta proteína viral que no están descritos claramente y que también están involucrados en el tropismo por el hospedero (Sriwilaijaroen N et al., 2012, Lazniewski et al., 2018). En los últimos años, se ha investigado una amplia gama de métodos de extracción de características y modelos de ML, empleando conjuntos de datos, tanto de secuencias de nucleótidos como de aminoácidos. Estos estudios abordan de manera generalizada el desafío de clasificación con el objetivo de desarrollar una estrategia efectiva para predecir el hospedero original, genotipo de virus y potencial zoonótico (Borkenhagen L.K et al., 2021).

En el trabajo de Xu *et al*. (2022) se introduce una red neuronal *transformer* de 5-gramas, con excelente desempeño de clasificación y que resultó eficaz para predecir el origen de secuencias virales. Este enfoque aprovechó el poder de los *transformers* para analizar datos secuenciales, particularmente apto para manejar patrones complejos en secuencias virales.

Attaluri et al. (2010) y Shaltout et al. (2015) desarrollaron modelos que produjeron predicciones de hospederos con una exactitud sobre la muestra de evaluación igual o superior a 0.98. Sin embargo, sus modelos consideraron sólo ciertos subtipos de HA (H1, H3 y H5), lo que limita su utilidad. Por otro lado, Xu et al. (2017) y Kwon et al. (2020) lograron una exactitud ligeramente menor (0.96) con modelos que consideran todos los subtipos.

En tres trabajos se implementaron modelos de predicción de hospedero (aves, humanos o cerdos) basados únicamente en la secuencia de HA (Yin et al., 2018; El Hefnawi & Sherif, 2014). Las mejores exactitudes se alcanzaron por el árbol de decisión desarrollado por El Hefnawi y Sherif, con valores entre 0.912 y 1.0 sobre la muestra de evaluación, dependiendo del hospedero y subtipo. Yin y colaboradores desarrollaron un modelo similar basado en Random Forest, que luego fue extendido a todas las proteínas del VIA para estimar la probabilidad de que la secuencia surgiera de un evento de reordenamiento. La exactitud de la predicción de hospedero para cada proteína varió entre 0.865 y 0.965 sobre la muestra de evaluación, y el modelo identificó correctamente el 86% de los reordenantes conocidos.

Otros tres estudios se enfocaron en distinguir entre aves y humanos (Allen et al., 2009; Eng et al., 2014; King et al., 2010), el mejor rendimiento alcanzado por Eng et al. usando un modelo de Random Forest entrenado con propiedades de aminoácidos, el modelo tuvo una exactitud de 0.9983, una sensibilidad de 0.998, y una especificidad de 1.00 sobre la muestra de evaluación, con una AUC de 0.998 y un MCC de 0.997. Hay que tener en cuenta que, aunque los modelos de King et al. (2010) obtuvieron métricas más bajas, fueron capaces de identificar cambios en la secuencia de aminoácidos y en características fisicoquímicas examinando tendencias de conectividad de aminoácidos entre hospederos aviares y humanos. La conectividad de aminoácidos se identificó mediante comparaciones estadísticas de la frecuencia de co-ocurrencia de pares de aminoácidos. Encontraron que, si bien los virus de influenza en humanos tienden a tener mayores tasas de mutación que los aviares, en general, las redes de conectividad de aminoácidos en aves resultaron más diversas. Esto puede explicarse por la mayor diversidad de VIA esperada a ese nivel taxonómico (aves) en comparación con una sola especie (humano).

Attaluri y colaboradores (Attaluri et al., 2009) analizaron la discriminación entre tropismo humano y porcino, evalaundo el origen porcino propuesto para el virus pandémico H1N1 de 2009. En el estudio se entrenaron clasificadores utilizando el modelo Support Vector Machines (SVM) y árboles de decisión para distinguir virus aislados de humanos y de cerdos. Los modelos fueron usados para clasificar secuencias pandémicas de 2009 y la mayoría fueron clasificadas como porcinas, apoyando un origen porcino para dicho virus pandémico.

Aguas y Ferguson (Aguas & Ferguson, 2013) examinaron hospederos más allá de humanos, aves y cerdos, buscando identificar marcadores de adaptaciones específicas a humanos, aves, cerdos, equinos y caninos. Las secuencias de aminoácidos de PB2, con regiones conservadas eliminadas, fueron convertidas en una matriz, y se utilizó Random Forest para identificar patrones basados en las cinco etiquetas de hospedero. No se proporcionaron métricas de predicción, pero los autores identificaron 23 posiciones de secuencia como importantes para generar predicciones.

En los últimos años, se ha investigado también una amplia gama de métodos de extracción de características y modelos de ML, empleando conjuntos de datos tanto de secuencias de nucleótidos como de aminoácidos (Yin et al., 2018). Estos estudios abordan de manera generalizada el desafío de clasificación con el objetivo de desarrollar una estrategia efectiva para predecir el hospedero original, genotipo de virus y potencial zoonótico (Borkenhagen L.K et al., 2021).

En suma, existen varios antecedentes de modelos de ML entrenados para clasificar secuencias virales según su hospedero de origen. Sin embargo el campo aún enfrenta desafíos, que se detallan en la siguiente sección.

2.6.Desafíos en la identificación de hospederos.

Anteriormente se mencionaron las dificultades de los métodos tradicionales para la identificación de hospederos basados en técnicas de laboratorio como el uso de ensayos de inhibición de la hemaglutinación (HI) para subtipificar virus, el aislamiento viral y la PCR. Incluso los métodos más recientes que emplean tecnología de secuenciación presentan limitaciones, ya sea en inspecciones manuales de secuencias, en análisis mediante herramientas de alineamiento como BLAST, o en estudios filogenéticos. Si bien la asignación de subtipo a una nueva secuencia de VIA se realiza con herramientas de alineamiento como BLAST, este método no resulta confiable cuando existe baja homología de secuencia (Borkenhagen et al., 2021). Todos estos métodos son laboriosos, necesitan de técnicos especializados y consumen mucho tiempo.

Como también ya repasamos, con el fin de agilizar la toma de decisiones se han utilizado diversos algoritmos de aprendizaje automático para predecir los hospedadores virales. Entre los modelos más conocidos tenemos los k-vecinos más cercanos (KNNs) (Sherif et al., 2017), random forest (RFs) (Sherif et al., 2017), redes neuronales artificiales (ANNs) (Attaluri et al., 2010) y árboles de decisión (DTs) (Kargarfard et al., 2016). Sin embargo, la mayoría de estos estudios previos seleccionaron manualmente conjuntos de datos balanceados (Sherif et al., 2017; Attaluri et al., 2009), utilizaron conjuntos de datos pequeños (Attaluri et al., 2010), codificaron la secuencia como matrices dispersas (Attaluri et al., 2010; Mock et al., 2021) o incorporaron procedimientos de extracción de características a partir de varias proteínas virales (Yin et al., 2018).

Hasta el momento, no se ha establecido un flujo de trabajo integral que permita, de manera sistemática, la clasificación del hospedero, la identificación del subtipo de HA y la determinación de la patogenicidad del VIA asociada a distintos clados, todo lo cual evidencia la necesidad de seguir abordando la problemática mediante enfoques de ML. Es por ello que esta tesis pretende responder a esta necesidad mediante un enfoque integrado que combina herramientas de ML con interpretación biológica, priorizando no sólo la precisión de las predicciones, sino también la comprensión de señales moleculares que subyacen a la adaptación viral y la patogenicidad.

3. Objetivos

2.3. Objetivo General

- Implementar, evaluar y comparar modelos de aprendizaje automático para clasificar VIA según hospedero y subtipo de HA a partir de representaciones de secuencias de la proteína HA.

2.4. Objetivos específicos

- Generar una base de datos de secuencias del VIA provenientes de hospederos pertenecientes a tres grupos taxonómicos, humanos, aves y cerdos. La base de datos será procesada de manera que los modelos puedan extraer características relevantes de las secuencias correspondientes a las distintas clases taxonómicas.
- Implementar un modelo o conjunto de modelos de clasificación capaces de identificar los hospederos de muestras de virus tomando como entrada representaciones de las secuencias aminoacídicas de las proteínas de HA.
- Evaluar el desempeño de los modelos con un conjunto de secuencias correspondientes al periodo de la última epizootia de Influenza Aviar en Uruguay y la región, que en el año 2023 afectó varias especies de aves, mamíferos terrestres y marinos.
- Implementar un modelo o conjunto de modelos de clasificación que puedan identificar con precisión entre diferentes subtipos de HA tomando como entrada representaciones de las secuencias aminoacídicas de las proteínas de HA.
- Desarrollar una herramienta a nivel usuario no experto que permita de forma integrada utilizar los modelos desarrollados.

4. Metodología general

4.1. Esquema general de trabajo

Para abordar de manera ordenada los objetivos del proyecto, se propone el siguiente flujo de trabajo (Figura 3):

- **Base de datos**
  Obtención a partir de la plataforma NCBI/Virus datos de secuencias aminoacídicas de HA (multifasta) y base de datos descriptiva (metadata) de los hospederos humano, cerdo y aves. Se procedió a consolidar en una base de datos integral que luego se curó y segmentó generando dataset de entrenamiento, validación y test.
- **Generación de descriptores**
  Partiendo de la base de datos integral de secuencias aminoacídicas de HA, se calcularán los siguientes siete descriptores de secuencia utilizando el programa ProtFeat:

  - **AAC (Amino Acid Composition**): vector de 20 valores que indica la frecuencia relativa de cada aminoácido en la secuencia; resume composición global sin considerar el orden.

  - **PAAC (Pseudo Amino Acid Composition)**: composición "pseudo" que combina frecuencias de aminoácidos con factores de correlación que resumen el orden (dependencias a distancias λ) usando propiedades fisicoquímicas; busca representar composición y secuencia en un solo vector.

  - **DPC (Di-peptide Composition):** vector de 400 valores (20x20) con la frecuencia de cada dipéptido consecutivo; captura orden local.

  - **GDPC (Grouped Di-peptide Composition):** variante del DPC donde los aminoácidos se agrupan por propiedades (p. ej., hidrofóbicos, polares, cargados, etc.) y se calcula la frecuencia de dipéptidos entre grupos; reduce dimensionalidad y enfatiza propiedades

fisicoquímicas.

- **APAAC (Amphiphilic Pseudo-Amino Acid Composition):** extensión de PAAC que incorpora tendencias anfifílicas (relacionadas a hidrofobicidad e hidrofilicidad) y correlaciones a distintas distancias (λ); combina composición y orden.

- **Autocorrelación de Moran:** descriptor que mide la correlación (tipo "autocorrelación espacial") de una propiedad fisicoquímica a lo largo de la secuencia para distintos retardos (lag); captura patrones periódicos/gradientes del atributo.

- **PSSM AC (Position-Specific Scoring Matrix Autocovariance):** a partir de una PSSM (obtenida con PSI-BLAST), calcula autocovarianzas entre puntuaciones separadas por distintos lags; incorpora información evolutiva y dependencias de orden.

- **Aprendizaje no supervisado**
  Utilizando cada uno de los descriptores, se someterá a la base de datos etiquetada a técnicas de reducción de dimensionalidad (PCA y t-SNE) para visualizar su estructura en espacios de dos y tres dimensiones. Estos algoritmos realizan combinaciones lineales (PCA) y no lineales (t-SNE) de los datos originales y como resultados se obtienen componentes (dos o tres generalmente) que permiten evidenciar patrones de nuestros datos en un espacio menor. Se proyectarán las componentes en el espacio, evaluará la varianza conservada y se aplicarán algoritmos de clustering, evaluando la pureza de los clusters obtenidos, así como su cohesión y separación. También se evaluará cómo están distribuídos entre los clusters los distintos hospederos de orígen de las secuencias y los distintos subtipos de HA a los que pertenecen.

- **Selección de descriptores**
  A partir del análisis de los resultados del clustering y las correspondientes métricas de separación, se determinará qué descriptor o combinación de descriptores ofrece el mejor poder discriminativo entre hospederos y subtipos de HA.

- **Aprendizaje supervisado**
  Los descriptores seleccionados en el paso anterior se utilizarán como variables predictivas para entrenar modelos de aprendizaje supervisado. Los modelos serán

entrenados y validados para clasificar secuencias de HA virales según su hospedero de origen y el subtipo de HA al que pertenecen.

- **Evaluación de los modelos**
  Los modelos serán luego evaluados con conjuntos de datos no utilizados durante el entrenamiento y la validación, detallados en la siguiente sección.

- **Selección de modelos**
  Elección final de los modelos con el mejor desempeño para la clasificación de subtipos virales y hospederos, utilizando bases de datos de secuencias parciales y secuencias completas.

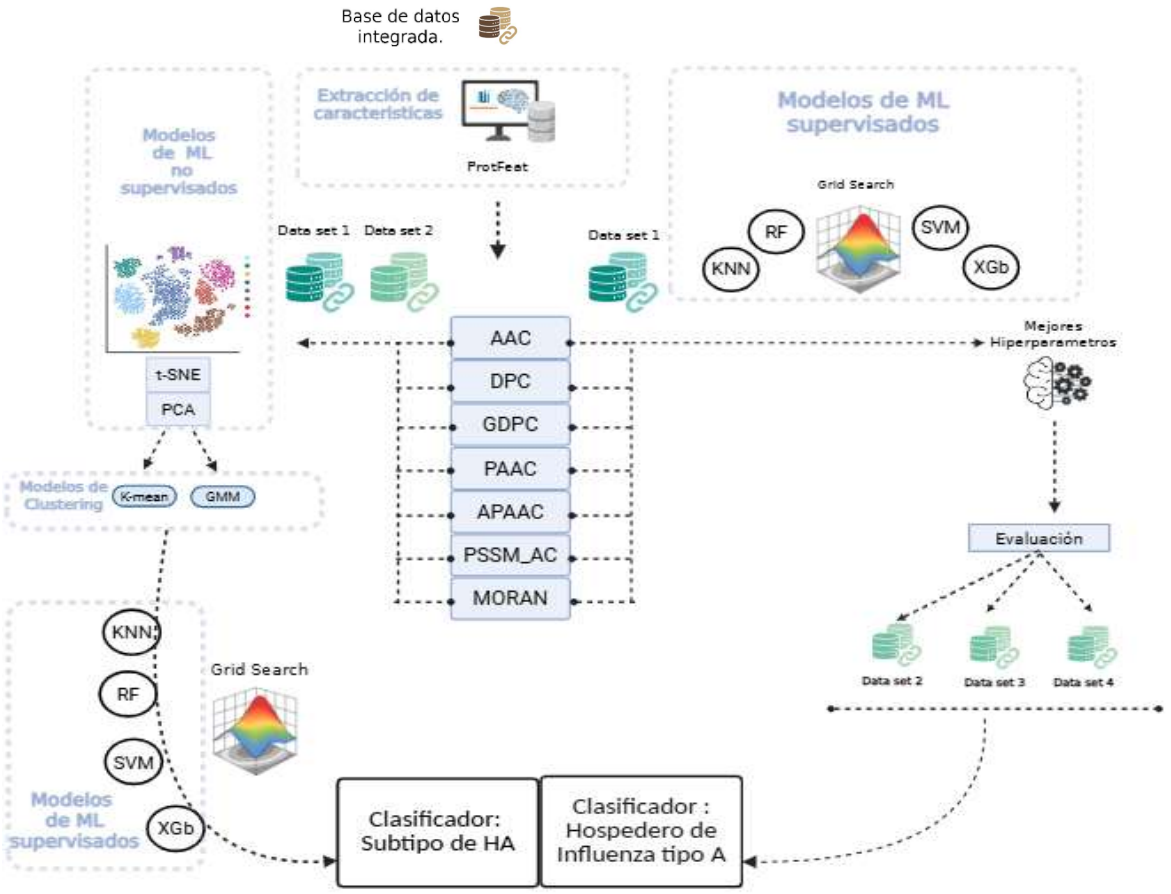


Figura 3. Esquema general del flujo de trabajo. A partir de la base de datos integrada se extrajeron los descriptores mediante ProtFeat, los cuales fueron evaluados con modelos de aprendizaje automático (AA) para generar un clasificador de hospedero y subtipo de HA.

4.2.Herramientas Bioinformáticas

Los análisis bioinformáticos que se realizaron en el presente trabajo se llevaron a cabo utilizando Google Colab (Google, 2023), una plataforma en línea basada en Júpiter Notebooks. El código se implementó en Python (Van Rossum & Drake, 2009), con el apoyo de bibliotecas especializadas como pandas (McKinney, 2010) para el manejo de datos, numpy para operaciones numéricas y scikit-learn, keras para la implementación de algoritmos de machine learning, matplotlib y seaborn para interpretación de resultados y gráficos. Por otro lado se utilizó un paquete de Python ProtFeat (Goszari, 2024) para computar los diferentes descriptores de proteína utilizados en el presente trabajo.

Con el fin de implementar el uso práctico de los modelos entrenados en el presente trabajo se plantea desarrollar una plataforma interactiva a nivel de usuario utilizando *Streamlit* (Streamlit Inc., 2024) . La plataforma permitirá utilizar secuencias aminoacídicas de la proteína HA del VIA como *input* y a través de integrar las funciones necesarias para ejecutar los modelos y *script* asociados al presente trabajo permitirá visualizar los resultados de los mismos.

4.3.Datasets utilizados en el trabajo

Las bases de datos utilizadas en este trabajo fueron obtenidas de la plataforma NCBI Virus, desarrollada por el Centro Nacional para la Información Biotecnológica (NCBI). Esta plataforma es un recurso integrador diseñado para facilitar la recuperación, visualización y análisis de una colección curada de secuencias virales y grandes conjuntos de datos. Los datos se obtuvieron de la colección curada de secuencias virales del NCBI, cuyo objetivo es aumentar la usabilidad de los datos archivados en GenBank y otros repositorios (National Center for Biotechnology Information [NCBI], 2025).

En este trabajo, se obtuvieron dos tipos de datos para cada hospedero. La base de datos descriptiva (en formato .csv), es un archivo que contiene metadatos de cada secuencia, identificadas mediante su número de acceso e información detallada asociada a cada secuencia. Este archivo proporciona datos como el año de recolección de la muestra, el tipo de muestra, subtipo de HA, especie de la cual se tomó la muestra y otras características asociadas. Por otro lado, a partir de la misma selección se obtuvieron archivos multifasta de proteínas, que almacenan las secuencias correspondientes a cada hospedero.

Las bases de datos fueron obtenidas inicialmente mediante la aplicación de filtros disponibles en la interfaz del sitio web de NCBI, seleccionando el tipo de virus, el hospedero y el segmento de interés.

Para el presente estudio, se seleccionaron los siguientes criterios:

- **Virus/Taxonomía:** *Alphainfluenzavirus* (TaxID: 19791).
- **Hospederos:**
  - *Homo sapiens* (human) – TaxID: 9606
  - *Birds* (birds) – TaxID: 8782
  - *Suidae* (pigs) – TaxID: 9821
- **Has Protein:** Hemagglutinin.

Luego de generada la búsqueda se obtuvieron por separado ambos archivos para cada uno de los hospederos, obteniendo un total de tres archivos multifasta de proteínas y tres archivos descriptivos. Luego de obtener la base de datos descriptiva y el archivo multifasta para cada hospedero, se procedió a unificarlas, asociando cada secuencia de proteína a las características asociadas a su id.

Inicialmente se procedió eliminar de la base de datos todas aquellas entradas que contenían letras "X " (aminoácido desconocido), "B"(ambiguo D/N) o "Z" (ambiguo E/Q) en las secuencias de proteínas, paso requerido para extracción de descriptores ya que estas no codifican para aminoácidos específicos. Luego se eliminaron las incidencias con datos ambiguos, incompletos o nulos acerca del subtipo, preservando sólo aquellas que contenían información completa del subtipo de HA (Ej:HxNx o Hx). Se eliminaron también las secuencias duplicadas y por último se concatenaron las tres bases de datos en una base de datos integrada, en la que el hospedero pasó a ser una etiqueta asociada a cada entrada. La base de datos integrada fue procesada utilizando ProtFeat (Goszari, 2024) y se calcularon siete descriptores diferentes a partir de cada secuencia. Finalmente, la base de datos se dividió en cinco conjuntos, uno de entrenamiento y otros cuatro para la evaluación, tal como se ilustra en la Figura 4. Los conjuntos de datos generados en este paso fueron:

- Dataset 1, que contiene sólo secuencias completas.
- Dataset 2, que contiene secuencias parciales contenidas en el dataset 1(Sp).
- Dataset 3, que contiene secuencias parciales no contenidas en el dataset 1 (Sp_nc).
- Dataset 4, que contiene secuencias completas colectadas del periodo 2020-2024 (20_24).
- Multi-etiqueta, secuencias que fueron encontradas en más de un hospedero.

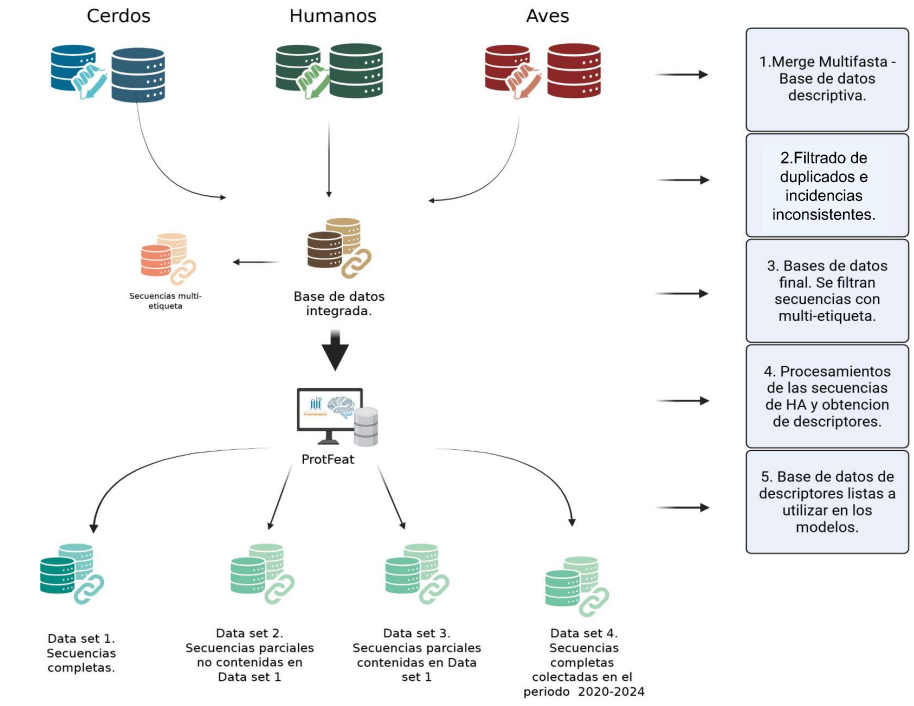


Figura 4. Diagrama de flujo para generar las bases de datos de entrenamiento y evaluación.

La tabla 1 muestra el número de secuencias en cada dataset, y resume sus características y propósitos. En todos los casos existe un desbalance hacia alguna de las clases, que fue considerado en el entrenamiento y evaluación de los modelos, ver Figura 5.

Tabla 1. Incidencias por conjunto de datos

Data set	Incidencias	Secuencias completas	NR[1]	Multi-etiqueta	X,B,Z filtrados	Propósito[2]
Original	172.975	-	-	-	-	Procesamiento
1(Completas)	35.810	✔	✔	-	✔	Entrenamiento/ Validación/Test
2 (Sp)[3]	12.374	-	✔	-	✔	Evaluación
3 (Sp_nc)[4]	7269	-	✔	-	✔	Evaluación
4 (20-24)[5]	7842	✔	✔	-	✔	Evaluación
Multi- etiqueta	169	-	✔	✔	✔	Evaluación

1: NR: No redundante.
2: Propósito del data set: entrenamiento, validación y testeo.
3: Secuencias parciales
4: Secuencias parciales no contenidas
5: Secuencia correspondientes al periodo 2020-2024

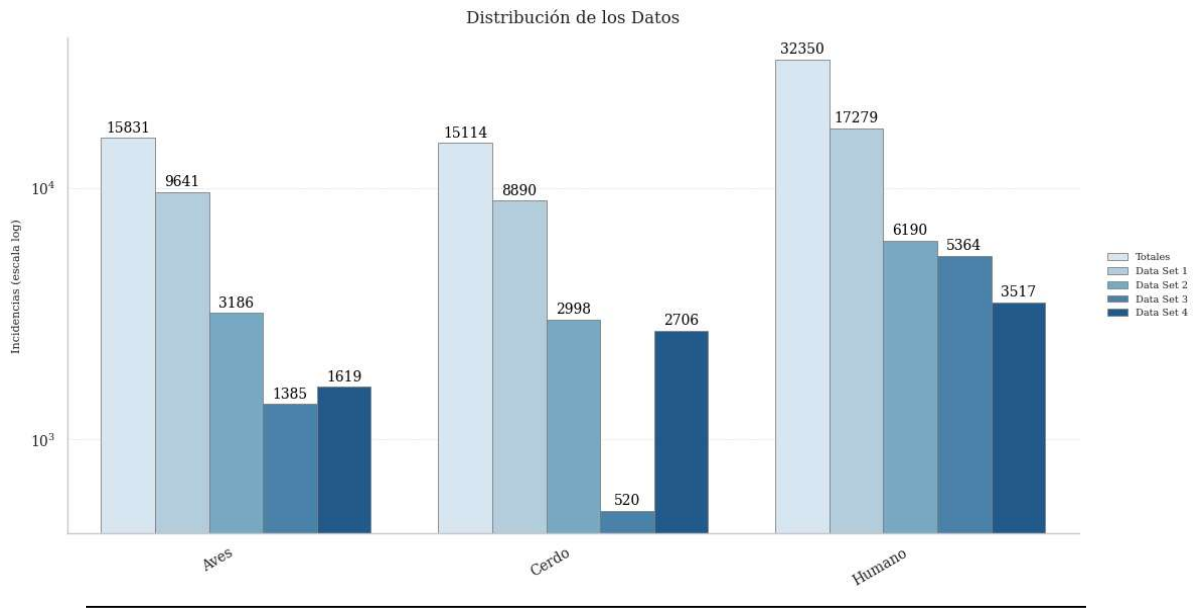


Figura 5. Distribución de los datos según la clase de hospedero y el dataset: (1) Secuencias completas, (2) Secuencias parciales contenidas, (3) Secuencias parciales no contenidas y (4) Secuencias correspondientes al período 2020–2024.

Las secuencias parciales se evaluaron en dos conjuntos de datos: aquellas contenidas dentro de las secuencias completas (Sp) y aquellas que no estaban contenidas (Sp_nc). Estas últimas presentan mayor proporción de secuencias de menor longitud, como se muestra en la Figura 6 .

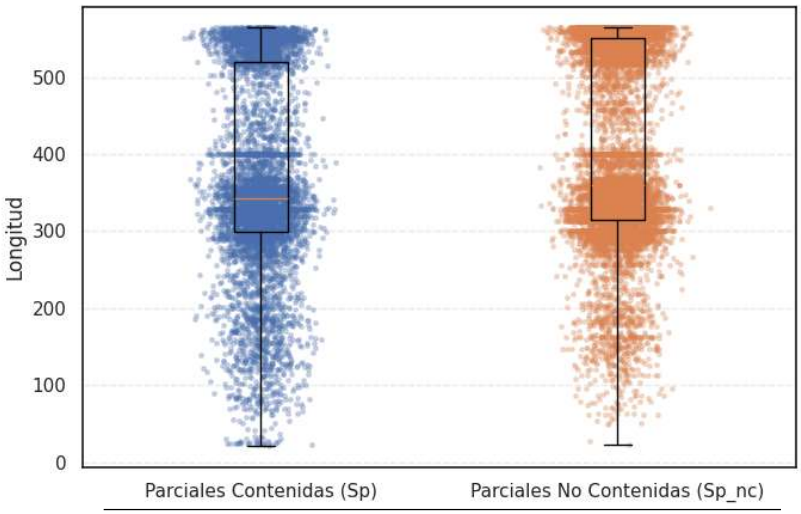


Figura 6. Distribución de las longitudes de secuencia correspondientes al conjunto de datos de secuencias parciales contenidas en las secuencias completas (Sp) y secuencias parciales no contenidas (Sp_nc).

4.4. Extracción de Características.

Para obtener las representaciones de las secuencias proteicas se utilizó el paquete de Python ProtFeat (Goszari, 2024), que utiliza las herramientas basadas en Python, POSSUM (Chen et al., 2018) e iFeature (Wang et al., 2017). ProtFeat incluye un total de 39 métodos distintos de extracción de características de proteínas (descriptores de proteínas). POSSUM (Generador de características basado en Matriz de Puntuación Posición-Específica para aprendizaje automático) es un kit de herramientas versátil con un servidor web en línea que puede generar 21 tipos de descriptores de características basados en PSSM. iFeature es otro kit de herramientas versátil basado en Python para generar diversos esquemas de representación numérica de características para secuencias de proteínas y péptidos. iFeature es capaz de calcular y extraer un espectro completo de 18 esquemas principales de codificación de secuencias, que abarcan 53 tipos diferentes de descriptores de características. En la Figura 7 se representa un esquema de la matriz de características utilizada para el análisis. Cada fila corresponde a una secuencia de HA (incidencia) y cada columna a una de las *m* componentes generadas, donde cada componente representa un valor asociado a una característica del descriptor seleccionado.

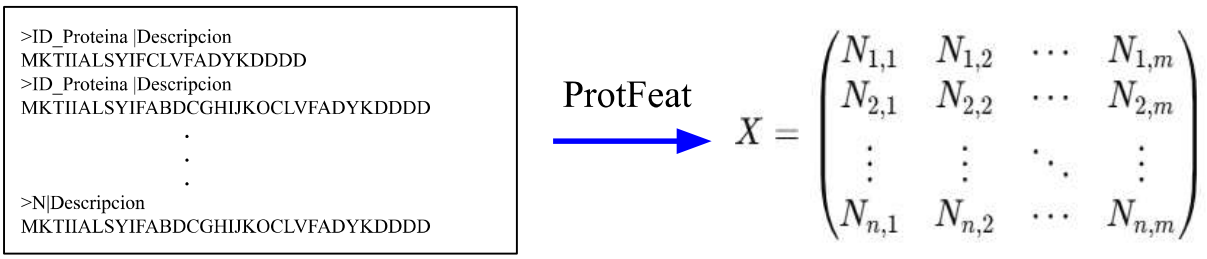


Figura 7. Ejemplo de matriz de característica *X*. En el panel de la izquierda se representa un archivo multifasta de proteína.

Para cada incidencia, se utilizó un archivo FASTA que contiene una secuencia de aminoácidos de la proteína HA. La matriz de características resultante tiene dimensiones *nxm*, donde *n* representa el número de secuencias en la base de datos final (un valor constante en todos los casos) y *m* es el número de características extraídas de cada secuencia, el cual varía dependiendo del descriptor utilizado. Las dimensiones de la matriz no dependen de la longitud de la proteína.

Los descriptores seleccionados fueron: autocovarianza de la matriz de puntuación por posición (PSSM AC), composición aminoacídica (AAC), pseudo composición aminoacídica (PAAC), composición peptídica (DPC), GDPC (composición peptídica), coeficiente de correlación Moran y pseudo composición aminoacídica anfifílica (APAAC). En conjunto, estos descriptores permiten que los algoritmos de ML detecten patrones moleculares distintivos entre secuencias virales de diferentes subtipos de HA y hospederos. A continuación se explica brevemente cada uno de ellos.

4.4.1. PSSM_AC (Position-Specific Scoring Matrix - Autocorrelation)

El PSSM_AC se refiere a un descriptor que combina el PSSM (Altschul et al., 1997) con una técnica adicional de autocorrelación (AC, por sus siglas en inglés), que busca capturar patrones de dependencia espacial a lo largo de la secuencia de aminoácidos (Zheng,2010). Este parámetro evalúa las correlaciones entre las posiciones dentro de la secuencia para obtener una representación más completa de la estructura y características funcionales de la proteína. El cálculo de autocorrelación es útil porque puede capturar información sobre la disposición de los aminoácidos en relación con sus vecinos cercanos o distantes. La matriz PSSM es generada a partir de la técnica de búsqueda PSI-BLAST. Esta es una variante del programa BLAST (Basic Local Alignment Search Tool), el cual es un método de búsqueda de similitud de secuencias donde una secuencia de proteína o nucleótido consultada se compara con secuencias de nucleótidos o proteínas en una base de datos para identificar regiones de alineamiento local y reportar aquellos alineamientos que superan un umbral de puntuación determinado. El método PSI-BLAST (Position-Specific Iterative BLAST) es una técnica de búsqueda de perfiles de secuencias de proteínas que se basa en los alineamientos generados en una ejecución del programa BLASTp (Altschul et al., 1990). La primera iteración de una búsqueda con PSI-BLAST es idéntica a una ejecución de BLASTp. Luego, se genera un alineamiento múltiple con los pares de secuencias con mayor puntuación de la búsqueda de BLASTp, siempre que estén por encima de un umbral preestablecido de puntuación o e-value. A partir de este alineamiento múltiple, se calcula un perfil o una matriz de puntuación específica por posición (PSSM, Position-Specific Scoring Matrix).

La PSSM captura el patrón de conservación en el alineamiento y lo almacena en una matriz de puntuaciones para cada posición en el alineamiento: las posiciones altamente conservadas reciben puntuaciones altas, mientras que las posiciones débilmente conservadas reciben puntuaciones cercanas a cero. Este perfil se usa en lugar de la matriz de sustitución original para realizar una nueva búsqueda en la base de datos y detectar secuencias que coincidan con el patrón de conservación especificado por la PSSM.

Las nuevas secuencias detectadas en esta segunda ronda de búsqueda, que superan el umbral de puntuación (e-value) especificado, se agregan nuevamente al alineamiento y el perfil se refina para otra ronda de búsqueda. Este proceso se repite iterativamente hasta que se alcanza el número deseado de iteraciones o hasta la convergencia, es decir, el estado en el que ya no se detectan nuevas secuencias por encima del umbral definido. El proceso iterativo de generación de perfiles hace que PSI-BLAST sea mucho más eficiente en la detección de similitudes de secuencias distantes en comparación con una consulta única en BLASTp, ya que combina la información de conservación de un conjunto de secuencias relacionadas en una sola matriz de puntuación (Figura 8).

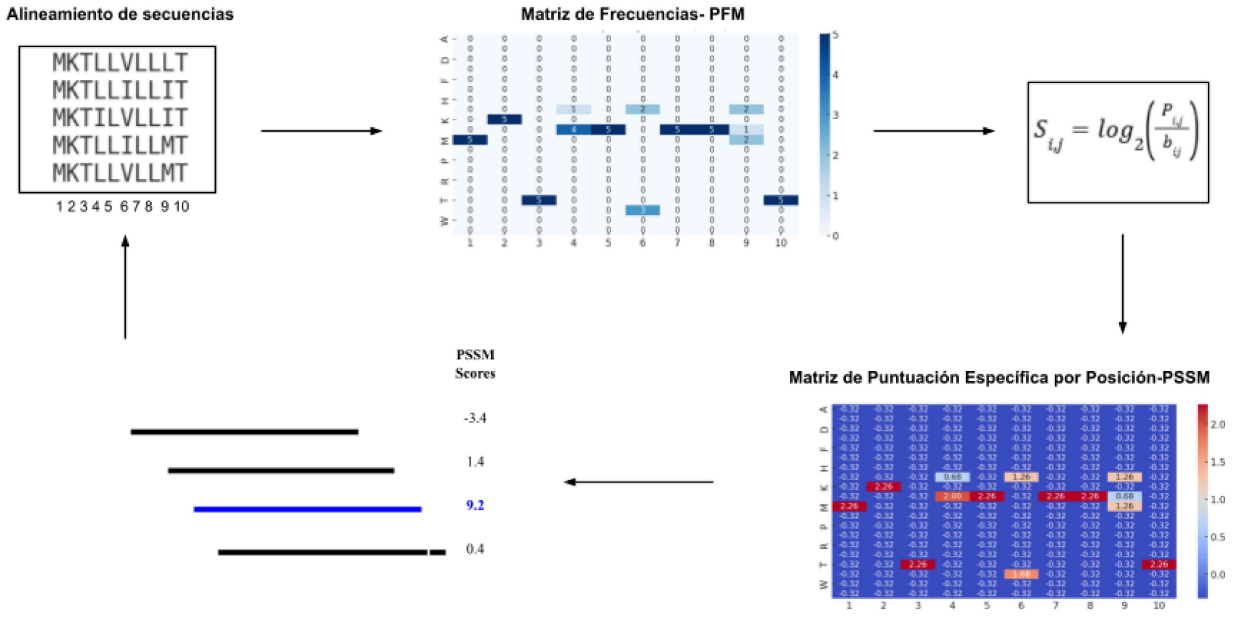


Figura 8. Esquema general para generar la matriz PSSM. Inicialmente se cuantifica los aminoácidos en cada posición del alineamiento generando la matriz de frecuencia (PFM), luego se escala este valor a su logaritmo a través de la función sigmoide. Estos valores conforman la matriz, que luego se utilizarán para ajustar la misma en cada iteración.

El componente *AC* mide la correlación de la misma propiedad entre dos residuos separados por una dis}

tancia *lg* a lo largo de la secuencia y puede ser calculada de la siguiente manera:

$$AC(i, lg) = \sum_{j=1}^{L-lg} (S_{i,j} - \overline{S}_i)(S_{i,j+lg} - \overline{S}_i)(L - lg) \quad , (1)$$

donde i es uno de los residuos, L es el largo de la secuencia de la proteína, S_{ij} es el score de PSSM del aminoácido i en la posición j ,$\overline{S}$ es el promedio del aminoácido i a lo largo de toda la secuencia.

$$\overline{S_i} = \sum_{j=1}^{L} \frac{S_{i,j}}{L}, \quad (2)$$

El número de la variable *AC* puede ser calculado como 20 x *LG*, donde *LG* es el máximo de lg (lg=1,2...,*LG*). Utilizando los parámetros default, el valor de *LG* es 10. Por lo cual las dimensiones de la matriz de características para PSSM AC es de 200.

4.4.2. AAC (Amino Acid Composition)

La Codificación por Composición de Aminoácidos (Bhasin y Raghava, 2004) calcula la frecuencia de cada tipo de aminoácido en una secuencia de proteína o péptido. Las frecuencias de los 20 aminoácidos naturales (es decir, "ACDEFGHIKLMNPQRSTVWY") pueden calcularse como se muestra en la Ecuación 3.

$$f(t) = \frac{N(t)}{N}, \quad t \in \{A, C, D, ..., Y\} \quad ,(3)$$

donde *N(t)* es el número de aminoácidos del tipo *t*, mientras que *N* es la longitud de la secuencia de proteína o péptido.

4.4.3. DPC (Dipeptide Composition)

El descriptor DPC calcula la frecuencia de dipéptidos (pares de aminoácidos adyacentes) en una secuencia, proporcionando información sobre el contexto local y las preferencias de dipéptidos entre residuos. Al considerar el orden secuencial, este descriptor identifica preferencias estéricas o funcionales en regiones específicas, como sitios activos de enzimas. Su simplicidad y baja dimensionalidad lo hacen adecuado para análisis de grandes conjuntos de datos, como estudios de proteómica comparativa o screening de mutaciones asociadas a enfermedades. La dimensión del vector resultante es 400, correspondiente a todos los dipéptidos que es posible formar con 20 aminoácidos (Saravanan and Gautham, 2015) .

$$D(r,s) = \frac{N_{rs}}{N-1}, \qquad r,s \in \{A, C, D, \ldots Y\} \quad ,(4)$$

En la Ecuación 4 se representa la ecuación para extraer las características del descriptor a partir de la secuencia de aminoácidos. N_{rs} son los aminoácidos en el dipéptido, siendo N el largo de la cadena y rs los aminoácidos que conforman el dipéptido.

4.4.4. GDPC-Composición de Dipéptidos Agrupados

El GDPC es otro descriptor basado en la frecuencia de aparición de dipéptidos pero agrupando los aminoácidos en 5 clases con características similares (Che J, 2022). Los cinco grupos son los siguientes : G1 (Alifáticos / hidrofóbicos): A, V, L, I, M ; G2 (Aromáticos): F, Y, W ; G3 (Positivos): K, R, H ; G4 (Negativos): D, E; G5 (Polares sin carga): S, T, N, Q. C, G, P.

$$f(r,s)=\frac{N_{rs}}{N-1}, \quad r,s\in\{g1,g2,g3,g4,g5\} \quad ,(5)$$

La Ecuación 5 corresponde a la frecuencia de dipéptidos que se agrupan por sus propiedades físico químicas, generando un vector de dimensión 25 para cada secuencia, resultado de las 5 x 5 combinaciones de los grupos.

4.4.5. Autocorrelación de Moran

Este descriptor mide la autocorrelación espacial de propiedades fisicoquímicas (ej. índice de refracción, polaridad) a lo largo de la secuencia proteica, utilizando un parámetro de distancia (lag). Al cuantificar patrones repetitivos o periódicos, es útil para identificar dominios estructurales como repeticiones en tándem o regiones transmembrana. Su aplicación también incluye la predicción de estructura secundaria y la clasificación de proteínas según su localización subcelular. Los descriptores de autocorrelación se definen en función de la distribución de las propiedades de los aminoácidos a lo largo de la secuencia (Feng y Zhang, 2000; Horne, 1988; Sokal y Thomson, 2006). Las propiedades fisicoquímicas de los aminoácidos utilizadas en este estudio corresponden a distintos tipos de índices de aminoácidos, los cuales se obtienen de la base de datos AAindex (Kawashima et al., 2008). Se emplean ocho índices específicos: 'CIDH920105' (Hidrofobicidad promedio normalizada), 'BHAR880101' (indice de flexibilidad promedio), 'CHAM820101' (Polarizabilidad), 'CHAM820102' (Energía libre de solución en agua (kcal/mol)), 'CHOC760101' (Área accesible al solvente en tripéptidos),

'BIGC670101' (Volumen del residuo), 'CHAM810101' (Parámetro estérico) y 'DAYM780201' (Mutabilidad relativa) (Xiao et al., 2015).

$$\sigma = \sqrt{\frac{1}{20}\sum_{r=1}^{20}(P_r - \overline{P})^2} \quad ,(6) \qquad \overline{P} = \frac{\sum_{r=1}^{20} P_r}{20} \quad ,(7) \qquad P_r = \frac{P_r - \overline{P}}{\sigma} \quad ,(8)$$

Todos los índices de los aminoácidos son centralizados y estandarizados luego del cálculo, siguiendo la Ecuación 8, donde $\overline{P}$ es el promedio de las propiedades de los 20 aminoácidos y σ es su desviación estándar.

$$I(d) = \frac{\frac{1}{N-d}\sum_{i=1}^{N-d}(P_i - \overline{P}')(P_{i+d} - \overline{P}')}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \overline{P}')^2}, \qquad d = 1, 2, 3..., nlag \quad , (9)$$

$$\overline{P}' = \frac{\sum_{i=1}^{N} P_i}{N} \quad , (10)$$

Los descriptores de autocorrelación de Moran (Feng y Zhang, 2000; Lin y Pan, 2001) se calculan con la Ecuación 9, donde *d* es el retraso (*lag*) de la autocorrelación, *nlag* es el valor máximo del retraso (valor predeterminado: 30), P_i y P_{i+d} son las propiedades de los aminoácidos en las posiciones *i* e *i+d*, respectivamente. $\overline{P}$ es el promedio de la propiedad considerada a lo largo de toda la secuencia de longitud *N* y se calcula con la Ecuación 10.

4.4.6. PAAC (Pseudo Amino Acid Composition)

El descriptor PAAC extiende la composición de aminoácidos tradicional al incorporar correlaciones secuenciales entre residuos, capturando así patrones no lineales en la disposición de aminoácidos. Este método integra parámetros hidrofóbicos, hidrofílicos y la masa de cadena lateral para modelar interacciones locales, lo que permite una representación más informativa que la AAC estándar. Estas Propiedades $H_{1o}(i)$ (Hidrofobicidad original), $H_{2o}(i)$(Hidrofilicidad original) y $M_o(i)$ (Masa de la cadena lateral) son estandarizadas como se muestra en la Ecuación 11. Cada una de las propiedades son convertidas de la misma manera para luego incorporarlas a la correlación.

$$H(i) = \frac{Ho(i) - \frac{1}{20}\sum_{i=1}^{20} Ho(i)}{\sqrt{\frac{\sum_{i=1}^{20}\left[Ho(i) - \frac{1}{20}\sum_{i=1}^{20} Ho(i)\right]^2}{20}}}, \quad (11)$$

Es obtenida a partir del promedio de cada propiedad como se indica en la Ecuación 12. Un ejemplo ilustrativo se representa en la Figura 9.

$$\Theta(R_i, R_j) = \frac{1}{3}\left\{\left[H_1(R_i) - H_1(R_j)\right]^2 + \left[H_2(R_i) - H_2(R_j)\right]^2 + \left[M(R_i) - M(R_j)\right]^2\right\}, \quad (12)$$

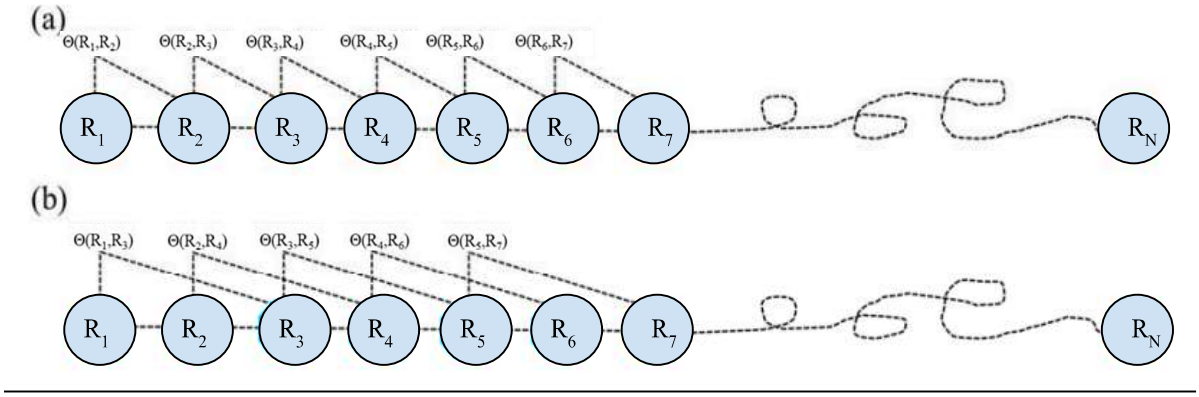


Figura 9. Esquema que muestra en (a) el modo de correlación de orden de secuencia de primer nivel, (b) el de segundo nivel a lo largo de una secuencia de proteína.(a) refleja el modo de acoplamiento entre todos los residuos más cercanos, (b) muestra el acoplamiento entre los residuos con un espacio de uno de distancia.Esta figura es una adaptación de (Chou, 2001).

Para un set de *n* propiedades de aminoácidos puede definirse como se muestra en la Ecuación 13, generando un set de descriptores. El parámetro λ (λ< N), debe ser elegido. Finalmente el descriptor PAAC tiene dimensiones 20+λ. Las Ecuaciones 14 y 15 corresponden a los dos componentes del descriptor. Frecuencia aminoacídica y su componente de correlación respectivamente. El factor de peso *w* se estableció con el valor de 0,05 por defecto (Chou, 2001) y corresponde al efecto del orden de la secuencia.

$$\theta_1 = \frac{1}{N-1}\sum_{i=1}^{N-1}\Theta(R_i, R_{i+1})$$
$$\theta_2 = \frac{1}{N-2}\sum_{i=1}^{N-2}\Theta(R_i, R_{i+2})$$
$$\theta_3 = \frac{1}{N-3}\sum_{i=1}^{N-3}\Theta(R_i, R_{i+3}) \quad (13)$$
$$\cdots$$
$$\theta_\lambda = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda}\Theta(R_i, R_{i+\lambda})$$

$$X_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j}, \quad (1 < c < 20) \quad (14)$$

$$X_c = \frac{w\theta_{c-20}}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{\lambda}\theta_j}, \quad (21 < c < 20+\lambda) \quad (15)$$

4.4.7. APAAC (Amphiphilic Pseudo Amino Acid Composition)

La Composición Pseudo-Aminoácida Anfifílica (APAAC) fue propuesta en (Chou, 2001; Chou, 2005). El APAAC combina los principios del PAAC con información anfifílica, integrando la hidrofobicidad y la hidrofilicidad de los aminoácidos. Este enfoque es esencial para estudiar proteínas con interacciones de membrana, como canales iónicos o receptores acoplados a proteínas. Permite identificar regiones anfipáticas críticas para la formación de micelas o la unión a lípidos, siendo clave en proyectos de bioingeniería de proteínas de membrana o en el desarrollo de fármacos dirigidos a compartimentos celulares específicos. La definición de este conjunto de características es similar a los descriptores PAAC. Usando *H1(i)* y *H2(j)* como se definieron previamente, las funciones de correlación de hidrofobicidad e hidrofilicidad se definen en la Ecuación 16.

$$H_{i,j}^{1} = H_1(i)H_1(j)$$
$$H_{i,j}^{2} = H_2(i)H_2(j) \quad , (16)$$

La secuencia del orden de los factores se resume en la Ecuación 17. Luego, se define el conjunto de descriptores proporcionados en la ecuación 18 y 19, *w* es el factor de ponderación.

$$\tau_1 = \frac{1}{N-1}\sum_{i=1}^{N-1} H_{i,i+1}^{1}$$
$$\tau_2 = \frac{1}{N-1}\sum_{i=1}^{N-1} H_{i,i+1}^{2}$$
$$\tau_3 = \frac{1}{N-2}\sum_{i=1}^{N-2} H_{i,i+2}^{1} \quad ,(17)$$
$$\tau_4 = \frac{1}{N-2}\sum_{i=1}^{N-2} H_{i,i+2}^{2}$$
$$\ldots$$
$$\tau_{2\lambda-1} = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} H_{i,i+\lambda}^{1}$$
$$\tau_{2\lambda} = \frac{1}{N-\lambda}\sum_{i=1}^{N-\lambda} H_{i,i+\lambda}^{2}$$

$$P_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{2\lambda} \tau_j}, \quad (1 < c < 20) \quad ,(18)$$

$$P_c = \frac{\omega\tau_u}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{2\lambda} \tau_j}, \quad (21 < u < 20 + 2\lambda) \quad ,(19)$$

El factor w se establece por defecto con un valor igual a 0.5 y el factor *lag* (λ) = 30, según lo descrito en el trabajo de Chou (Chou, 2001). La dimensión del vector es de 80, con 20 dimensiones que representan la secuencia ponderada y otras 60 (2 x λ) que corresponden a los valores de autocorrelación. En la Figura 10 se representa un esquema del acoplamiento de ambos factores.

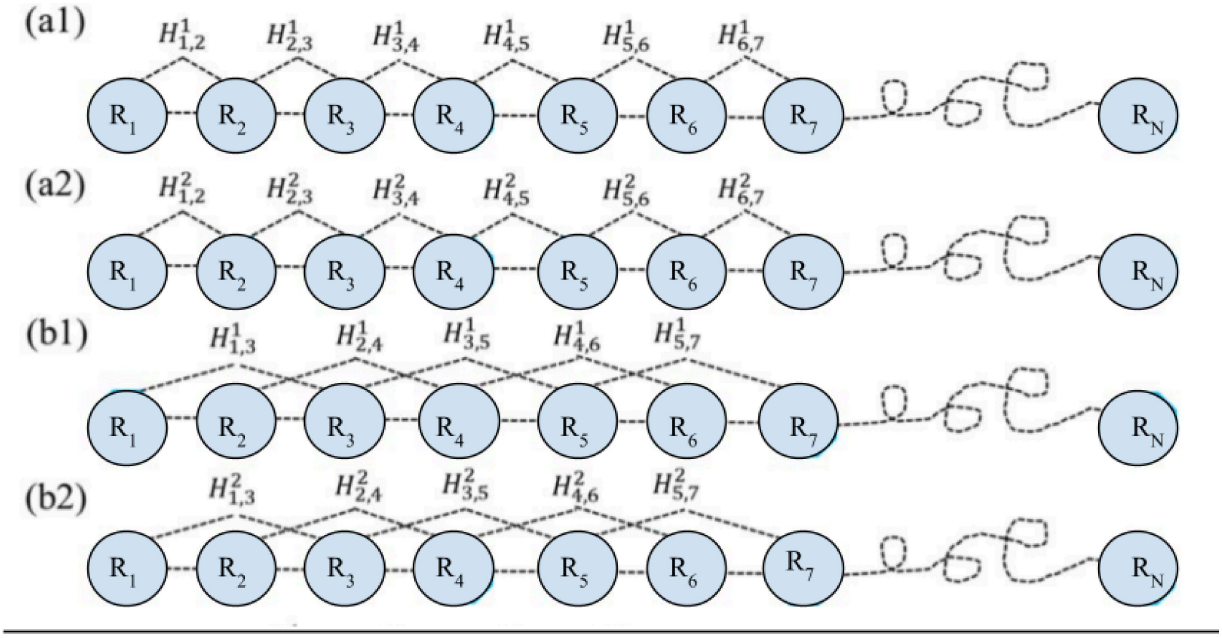


Figura 10. Diagrama esquemático que muestra el modo de acoplamiento del orden de la secuencia en diferentes rangos a lo largo de una secuencia proteica mediante una función de correlación de hidrofobicidad/hidrofilicidad: (a1/a2) primer rango, (b1/b2) segundo rango. Aquí, *H(1)i,j* y *H(2)i,j* se definen según la ecuación mencionada anteriormente. Los paneles (a1/a2) reflejan el modo de acoplamiento entre los residuos más cercanos, los paneles (b1/b2) muestran el acoplamiento entre los residuos adyacentes más uno. Esta figura ha sido adaptada de Chou (2005) con fines ilustrativos.

4.5.Algoritmos de aprendizaje

4.5.1. Modelos no supervisados

El análisis de componentes principales (PCA, por sus siglas en inglés) y la incrustación estocástica de vecinos distribuida en t (t-SNE, por sus siglas en inglés) son métodos de aprendizaje no supervisado ampliamente utilizados para reducción de dimensionalidad y visualización exploratoria de datos de alta dimensión, como ocurre cuando se representan secuencias biológicas mediante cientos o miles de características (p. ej., composición aminoacídica, dipéptidos, descriptores fisicoquímicos o perfiles evolutivos). Estas técnicas permiten proyectar los datos a dos o tres dimensiones para facilitar la inspección visual de posibles patrones, gradientes, subpoblaciones y

observaciones atípicas, sin requerir etiquetas durante la transformación (Pedregosa et al., 2011). El análisis de componentes principales es un método lineal que busca un nuevo sistema de ejes (componentes) que sean ortogonales entre sí y estén ordenados de manera que el primer componente capture la mayor proporción posible de la variabilidad del conjunto de datos, el segundo capture la mayor parte de la variabilidad restante, y así sucesivamente (Jolliffe, 2002). En la práctica, PCA es útil para resumir información reduciendo el número de variables, mitigar redundancia entre descriptores correlacionados y generar representaciones compactas que conservan la estructura global dominante del conjunto. Además, es relativamente interpretable: puede examinarse qué variables contribuyen más a cada componente (cargas), lo que ayuda a identificar qué propiedades (por ejemplo, composición o atributos fisicoquímicos) explican la mayor parte de la variación observada (Jolliffe, 2002). Por ello, PCA suele emplearse tanto como herramienta de exploración previa como paso de preprocesamiento para métodos posteriores.

En contraste, t-SNE es un método no lineal diseñado principalmente para visualizar datos de alta dimensión preservando la estructura local. Conceptualmente, t-SNE convierte las distancias entre puntos en el espacio original en probabilidades de vecindad y busca una proyección en dos o tres dimensiones cuya distribución de vecindades sea lo más similar posible, optimizando una función objetivo basada en divergencia (van der Maaten & Hinton, 2008). Esta estrategia hace que t-SNE sea especialmente eficaz para revelar agrupamientos locales (por ejemplo, subconjuntos de secuencias con características similares) aun cuando las relaciones no puedan capturarse bien con métodos lineales. Sin embargo, su interpretación requiere cautela: la optimización es no convexa, por lo que diferentes inicializaciones o configuraciones pueden producir mapas distintos, y las distancias globales entre grupos (por ejemplo, “qué tan lejos” están dos clústeres) no necesariamente reflejan distancias reales en el espacio original (van der Maaten & Hinton, 2008).

4.5.2. Modelos supervisados

El avance del aprendizaje automático ha proporcionado un enfoque eficaz para la resolución de problemas biológicos (Basith et al., 2022; Charoenkwan et al., 2021; Hasan et al., 2022; Jeon et al., 2022). El uso de estas técnicas para identificar proteínas en función de sus características de secuencia ha demostrado ser un método rápido y

ampliamente aplicado en diversos estudios (Yang et al., 2022; Yuan et al., 2022; Zhang et al., 2022). La construcción de modelos adecuados es fundamental para lograr predicciones precisas y robustas. En este trabajo, se utilizaron cuatro algoritmos de aprendizaje automático supervisado —K-Nearest Neighbor (KNN) (Cover & Hart, 1967)), Random Forest (RF) (Breiman, 2001) , Support Vector Machine (SVM) (Cortes & Vapnik, 1995) y XGBoost (Pedregosa et al., 2011). En todos los casos se utilizarán las implementaciones disponibles en la biblioteca SciKit-Learn.

KNN es un algoritmo de aprendizaje automático simple pero eficaz, basado en la medición de la distancia entre los datos. En su versión más simple, el algoritmo k-NN considera exactamente un solo vecino más cercano, que es el punto de datos de entrenamiento más próximo al punto para el cual queremos hacer una predicción. La predicción es, por lo tanto, simplemente el valor de salida conocido de ese punto de entrenamiento. En lugar de considerar solo el vecino más cercano, también podemos considerar un número arbitrario, k, de vecinos. De ahí proviene el nombre del algoritmo de k-vecinos más cercanos (k-nearest neighbors, KNN). Cuando se consideran varios vecinos, utilizamos votación para asignar una etiqueta. Esto significa que, para cada punto de prueba, contamos cuántos vecinos pertenecen a la clase 0 y cuántos pertenecen a la clase 1. Luego asignamos la clase que sea más frecuente; en otras palabras, la clase mayoritaria entre los k -vecinos más cercanos (Müller & Guido, 2016; Cover & Hart, 1967).

El modelo Random Forest es esencialmente una colección de árboles de decisión, donde cada árbol es ligeramente diferente de los demás. Es una forma de sobrellevar el probable sobreajuste que generan los árboles de decisión individuales. La idea detrás de los múltiples árboles generados en este algoritmo es que cada árbol puede hacer un trabajo relativamente bueno al predecir, pero probablemente se sobreajuste a una parte específica de los datos. Si construimos muchos árboles, todos los cuales funcionan bien pero se sobreajustan de maneras diferentes, podemos reducir el sobreajuste promediando sus resultados. Esta reducción, al mismo tiempo que se mantiene el poder predictivo de los árboles, puede demostrarse mediante matemáticas rigurosas. El resultado de la predicción se determina mediante el voto o el promedio de múltiples árboles de decisión (Müller & Guido, 2016;Breiman, 2001).

El principio básico de SVM es separar dos clases de datos de entrenamiento definiendo un hiperplano maximizando la distancia entre ambas clases. Durante el entrenamiento, SVM aprende qué tan importante es cada punto de datos de entrenamiento para representar la frontera de decisión entre las dos clases. Normalmente, sólo un subconjunto de los puntos de entrenamiento es relevante para definir esa

frontera: aquellos que se encuentran en el límite entre las clases. Estos puntos se denominan vectores de soporte (support vectors) y son los que dan nombre al algoritmo de máquinas de vectores de soporte. Para realizar una predicción sobre un nuevo punto, se mide la distancia entre ese punto y cada uno de los vectores de soporte. La decisión de clasificación se toma en función de esas distancias y de la importancia de los vectores de soporte aprendida durante el entrenamiento. Este algoritmo también utiliza *Kernels*, que permite construir modelos más complejos, que no se definen únicamente por hiperplanos en el espacio de entrada. Existen dos formas de mapear los datos a un espacio de mayor dimensión (kernels) que se usan comúnmente con las máquinas de vectores de soporte; el kernel polinomial, que calcula todos los posibles polinomios hasta cierto grado a partir de las características originales y el de base radial (RBF), también conocido como núcleo gaussiano. Este considera todos los polinomios posibles de todos los grados, pero la importancia de las características disminuye a medida que el grado aumenta (Müller & Guido, 2016;Cortes & Vapnik, 1995).

XGBoost es un algoritmo de gradient boosting optimizado, un método de ensamble que combina múltiples árboles de decisión para crear un modelo más potente. Se caracteriza por su alta eficiencia computacional, capacidad para manejar datos con valores faltantes y mecanismos integrados para evitar el sobreajuste, como la regularización L1 y L2. A pesar de que su nombre incluye la palabra “regresión”, estos modelos pueden utilizarse tanto para regresión como para clasificación. A diferencia del enfoque de Random Forest, el XGboost construye los árboles de forma secuencial (en serie), donde cada árbol intenta corregir los errores del anterior. Por defecto, no hay aleatorización en los árboles potenciados por gradiente; en cambio, se utiliza una poda previa fuerte (*pre-pruning*). Los árboles utilizados en este método suelen ser muy poco profundos, con una profundidad de entre uno y cinco niveles, lo que hace que el modelo sea más pequeño en memoria y que las predicciones sean más rápidas. La idea principal detrás del gradient boosting es combinar muchos modelos simples (conocidos en este contexto como *weak learners*), como árboles poco profundos. Cada árbol sólo puede realizar buenas predicciones sobre una parte de los datos, por lo que se van agregando más árboles de forma iterativa para mejorar el rendimiento global del modelo (Müller & Guido, 2016).

Para la implementación de los modelos de clasificación se emplearon las siguientes librerías y clases:

- K-Nearest Neighbors (KNN): se utilizó el clasificador KNeighborsClassifier de la librería *scikit-learn (scikit-learn, 2020a)*.

- Support Vector Machines (SVM): se implementó mediante la clase SVC de *scikit-learn (scikit-learn, 2020b)*.

- Random Forest: se empleó el clasificador RandomForestClassifier de *scikit-learn (scikit-learn, 2020d)*.

- Gradient Boosting Decision Trees (GBDT): se implementó mediante la clase XGBClassifier del paquete *XGBoost (XGBoost Developers, 2020)*.

4.6. Métricas de evaluación

Una vez entrenados, todos los modelos fueron evaluados con una o más muestras de evaluación, compuestas por secuencias etiquetadas que no fueron vistas por los modelos durante el entrenamiento. A partir de la matriz de confusión, que contabiliza los distintos tipos de errores que el modelo comete al clasificar casos nuevos, se definen múltiples métricas de evaluación. En el presente trabajo las métricas que se utilizaron fueron: *F1 score Macro* (ecuación 20), *Balanced Accuracy* (ecuación 21)*, Matthews's correlation coefficient (MCC)* (ecuación 23). También se utilizó la métrica del área bajo la curva *Preccision-Recall* (AUC-PR, ecuación 22), la cual es recomendada cuando las clases están desbalanceadas (Branco *et al*., 2016). En las ecuaciones mencionadas anteriormente el número de clases se representa con *c*, TP (Verdadero positivo) y TN (Verdadero Negativo) representan el número de datos correctamente clasificados; FP (Falso Positivo) es el número de datos negativos clasificados erróneamente como positivos; FN (Falso Negativo) es el número de datos positivos clasificados erróneamente como negativos.

Para la clasificación multiclase, se aplica la estrategia *one-vs-all* para calcular el F-score de cada clase.

$$F1 - Score_C = 2\,\frac{Precision_C + Recall_C}{Precision_C . Recall_C} \qquad F1\,macro = \frac{1}{k}\sum_{c=1}^{k} F1c \quad (20)$$

$$Balanced\ Accuracy = \frac{1}{k}\sum_{c=1}^{k}\frac{TPc}{TPc+FNc} \quad (21)$$

$$Precision_C = \frac{TPc}{TPc+FPc}, \quad Recall_C = \frac{TPc}{TPc+FNc} \quad (22)$$

$$MCC = \frac{TPc.TNc - FPc \cdot FNc}{\sqrt{(TPc+FPc)(TPc+FNc)(TNc+FPc)(TNc+FNc)}} \quad (23)$$

4.7.Implementación de modelos.

4.7.1. Problema de Clasificación Multiclase.

La clasificación multiclase es el problema de clasificar instancias entre varias clases posibles. El objetivo es aprender un predictor $h:X \rightarrow Y$ donde Y es un conjunto finito de categorías. Algunas aplicaciones incluyen, por ejemplo, la categorización de documentos según su tema (donde X es el conjunto de documentos e Y es el conjunto de temas posibles) o la identificación de objetos en una imagen determinada (donde X es el conjunto de imágenes e Y es el conjunto de objetos posibles).

La importancia del problema de aprendizaje multiclase ha impulsado el desarrollo de diversas estrategias para abordarlo. Un enfoque directo para abordar la clasificación multiclase consiste en reducirla a una serie de problemas de clasificación binaria. En el caso de la predicción de los diferentes hospederos del virus de influenza, este procedimiento se implementa de forma práctica mediante la codificación de clases previamente descrita. En el presente trabajo se empleó la estrategia *Uno-vs-Resto* (*One-vs-Rest*) para calcular las métricas mencionadas para cada clase y, posteriormente, obtener el promedio correspondiente que representa el rendimiento global del modelo.

4.7.2. Modelo *Uno-vs-Resto*.

El enfoque más simple para abordar problemas de predicción multiclase es mediante la reducción a clasificación binaria. Recordemos que en la predicción multiclase

queremos aprender una función $h: X \rightarrow Y$. Sin pérdida de generalidad, podemos denotar $Y=\{1,\ldots,k\}$.

En el método *Uno-vs-Resto* se entrenan k clasificadores binarios, cada uno de los cuales distingue una clase de las demás. Es decir, dado un conjunto de entrenamiento $S=\{(x1,y1),\ldots,(xm,ym)\}$, donde cada y_i pertenece a Y, se construyen k conjuntos de entrenamiento binarios $S1,\ldots,Sk$, donde:

$$S_i = (x_1, (-1)^{[y_1 \neq i]}) \ldots (x_m, (-1)^{[y_m \neq i]}) \quad (24)$$

Luego, dado el conjunto de clasificadores $h_1,\ldots,h_k$, se construye el predictor multiclase utilizando la regla:

$$h(x) \in \arg\max_{i \in [k]} h_i(x) \quad (25)$$

Es decir, la clase predicha será aquella cuyo clasificador binario devuelve la probabilidad más alta de pertenencia a la clase positiva. Cuando más de un clasificador binario asigna probabilidad igual a 1, debemos decidir de alguna manera a qué clase asignar esa instancia. Por ejemplo, podríamos resolver los empates eligiendo arbitrariamente el índice mínimo en $arg\ max_i\ h_i(x)$. Un enfoque mejor puede aplicarse siempre que cada h_i contenga información adicional que pueda interpretarse como la confianza en la predicción $y=i$.

A continuación, se presenta un pseudocódigo del enfoque Uno-vs-Resto utilizado por los modelos.

Uno- vs - Resto

input:

Set de entrenamiento $S = (x_1, y_1), \ldots, (x_m, y_m)$

Algoritmo para clasificación Binaria A

for i en Y

sea $S_i = (x_1, (-1)^{1[y_1 \neq i]}), \ldots, (x_m, (-1)^{1[y_1 \neq i]})$

sea $h_i = A(S_i)$

output:

la hipótesis multiclase definida por $h(x) \in argmax_{i \in Y} h_i(x)$

4.8. Limitaciones

En este trabajo se identificaron tres limitaciones principales asociadas a la construcción y preprocesamiento de la base de datos. En primer lugar, varios descriptores utilizados requieren secuencias compuestas exclusivamente por los 20 aminoácidos estándar; en consecuencia, la presencia de caracteres ambiguos o no estándar (por ejemplo, X, B o Z) impide el cálculo de dichas características y obliga a excluir una proporción relevante de secuencias, lo que reduce el tamaño muestral y puede introducir sesgos en la representatividad del conjunto de datos. En segundo lugar, la base de datos empleada no corresponde a la principal fuente de referencia utilizada globalmente para la vigilancia de influenza; en particular, repositorios especializados como GISAID suelen ofrecer secuencias más completas y mejor curadas, junto con metadatos epidemiológicos de mayor calidad, lo que podría mejorar la cobertura y consistencia del análisis.

Finalmente, no se aplicó un procedimiento sistemático para eliminar secuencias altamente similares o redundantes, lo que puede aumentar la dependencia entre observaciones y favorecer estimaciones optimistas del desempeño durante la evaluación, especialmente cuando existen secuencias muy cercanas entre los conjuntos de entrenamiento y prueba.

5. Capítulo 1.

Modelos de clasificación de hospederos e identificación de regiones funcionales relevantes en la hemaglutinina (HA)

5.1. Resumen

Este capítulo describe el desarrollo y evaluación de modelos de aprendizaje automático para clasificar secuencias de hemaglutinina (HA) del virus Influenza A según su hospedero de origen (aves, humanos o cerdos). El estudio integra métodos de reducción de dimensionalidad (PCA, t-SNE), algoritmos de agrupamiento (K-means, GMM) y modelos supervisados (KNN, SVM, Random Forest, Gradient Boosting) aplicados sobre múltiples descriptores de secuencia (AAC, DPC, Moran, PAAC, PSSM AC, entre otros), con el fin de identificar patrones discriminativos entre subtipos virales y hospederos.

Inicialmente se exploraron estructuras latentes en los datos mediante técnicas no supervisadas. Los descriptores DPC y Moran demostraron una clara capacidad para formar agrupamientos consistentes por subtipo, superando a otros descriptores en pureza de clúster. Posteriormente, se entrenaron clasificadores utilizando validación cruzada y ajuste de hiperparámetros, siendo evaluados con distintos conjuntos independientes (secuencias completas, parciales y recientes). Los modelos entrenados con descriptores DPC y Moran presentaron los mejores desempeños en métricas como F1-macro, MCC y AUC-PR, especialmente con el clasificador KNN.

El análisis de importancia de características permitió identificar dipéptidos relevantes para la clasificación, los cuales fueron mapeados a regiones funcionales de la proteína HA. Se encontraron asociaciones estadísticamente significativas entre ciertos dipéptidos y regiones como HR2, el epítopo Ca2 y el dominio transmembrana para aves; el epítopo Sb para humanos; y el sitio de clivaje HA1/HA2 y el sitio de reconocimiento de ácido siálico para cerdos. Estas regiones contienen señales moleculares asociadas con la especificidad del hospedero, lo cual refuerza la relevancia funcional de los patrones detectados por los modelos.

Los modelos también mostraron robustez al clasificar secuencias ambiguas (aisladas de más de un hospedero), asignándolas consistentemente a uno de los hospederos documentados. Además, la comparación del desempeño sobre secuencias completas y parciales evidenció la capacidad de generalización de ciertos modelos y descriptores, siendo DPC-KNN el más robusto.

En conjunto, estos resultados demuestran que el aprendizaje automático puede no solo clasificar con alta precisión secuencias virales según su hospedero, sino también revelar regiones funcionales críticas para la adaptación del virus. Este enfoque proporciona herramientas valiosas para la vigilancia molecular y el entendimiento de la evolución del virus Influenza A.

5.2.Introducción

La primera fase de este trabajo se centró en la exploración de algoritmos clásicos de aprendizaje automático no supervisado y supervisado aplicados a la clasificación de proteínas de HA. El objetivo es desarrollar un sistema que procese datos provenientes de estas secuencias, con modelos entrenados y ajustados a niveles adecuados, que permita generar clasificaciones confiables acerca del hospedero origen y evaluar su posible implicancia biológica.

5.3.Objetivos

5.3.1. Objetivo General

Implementar un clasificador de secuencias de HA del virus de Influenza tipo A según su hospedero de orígen.

Objetivos específicos

- Recolectar secuencias de la proteína HA y su correspondiente metadata, asegurando la calidad de la información para la posterior extracción de matrices de características.

- Aplicar técnicas de reducción de dimensionalidad (*PCA*, *t-SNE*) y métodos de agrupamiento (clustering) para identificar patrones y relaciones entre secuencias correspondientes a distintos subtipos de HA y hospederos.
- Evaluar distintos descriptores de secuencia y seleccionar uno para entrenar un clasificador del subtipo de HA
- Evaluar los modelos de aprendizaje automático entrenados utilizando varios conjuntos de datos de evaluación independientes.

- Interpretar los resultados obtenidos en términos de variabilidad funcional y regiones relevantes a nivel de secuencia.

5.4. Metodología

El proceso inicia con la base de datos integrada, conformada por secuencias completas y parciales de la proteína HA del virus Influenza A de tres hospederos: aves, humanos y cerdos. A partir de estas secuencias, se extrajeron matrices de características utilizando el software ProtFeat, incluyendo los descriptores AAC,PSSM AC, PAAC, DPC, GDPC, Moran y APAAC, ampliamente empleados en la literatura para representar propiedades fisicoquímicas, patrones espaciales y evolutivos de proteínas. Para la exploración inicial con PCA y t-SNE y para entrenar los modelos de aprendizaje supervisado se tomaron datos del Data set 1 (Secuencias completas) . Luego de la reducción de dimensionalidad se realizaron dos análisis de clustering utilizando los algoritmos KMeans y GMM. La base de datos integrada se subdividió en las bases de datos previamente mencionadas en los métodos generales para así obtener los sets de Entrenamiento/Validación y de Evaluación, para entrenar los algoritmos de aprendizaje supervisado, procediendo de la misma manera con todos los descriptores.

5.4.1. Conjunto de entrenamiento.

Se aplicó la técnica Hold-Out sobre el data set 1 (secuencias completas) dividiendo el conjunto de datos en entrenamiento (90%) y evaluación (10%), manteniendo la proporción de clases mediante estratificación (*train_test_split*, scikit-learn) utilizando el parámetro *stratify=y ,* que asegura la proporción de clases. Esta partición aseguró la disponibilidad de un conjunto independiente para la evaluación final.

5.4.2. Entrenamiento y ajuste de hiperparametros

Los métodos de aprendizaje supervisado explorados en este trabajo son:: k-Vecinos más cercanos (KNN), Support Vector Machines (SVM), Árboles de decisión, Random Forest y Gradient Boosting Trees. En la etapa de entrenamiento, se implementó una validación cruzada estratificada 5-*fold* (*StratifiedKFold*, scikit-learn), con el objetivo de obtener estimaciones robustas del desempeño durante el ajuste de hiperparámetros, (el procedimiento se esquematiza en la Figura 11).

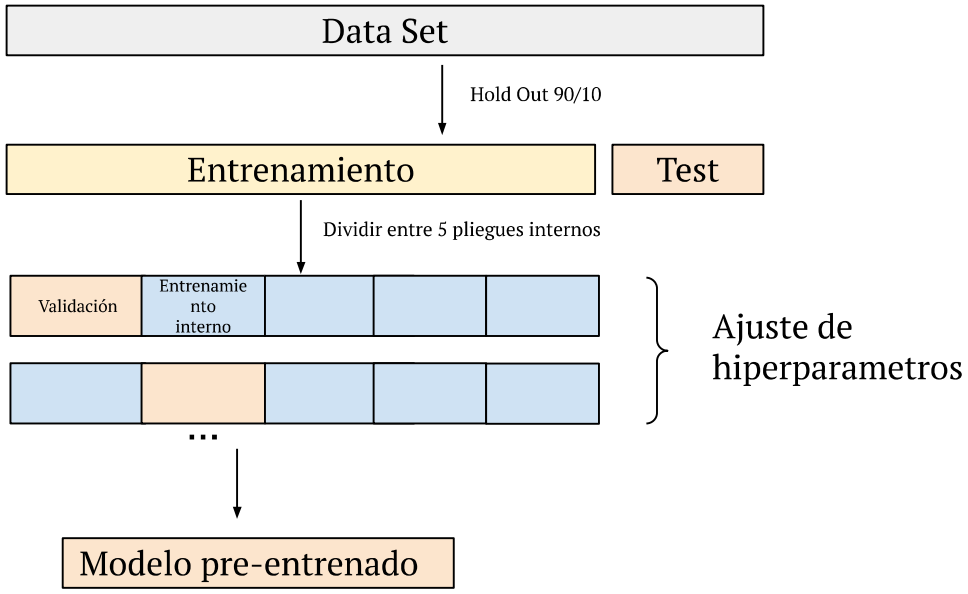


Figura 11. Ejemplo de validación cruzada anidada (k_externo = 5 y k_interno = 5): se utilizaron los modelos entrenados durante la validación cruzada anidada (es decir, modelos preentrenados) para predecir sobre datos no vistos.

Para cada uno de los algoritmos seleccionados, se definió un set de hiperparámetros a explorar (Tabla 2) mediante la estrategia de búsqueda en grilla (*GridSearchCV*, scikit-learn). El criterio de optimización utilizado fue la métrica F1-macro (ver ecuación 17), adecuada para escenarios con clases desbalanceadas. El mejor modelo por algoritmo se identificó a partir del valor máximo obtenido con esa métrica (*best_estimator_*).

Tabla 2. Configuración de Hiperparámetros

Modelos	Clasificador	Hiperparámetros
KNN	KNeighborsClassifier()	n_neighbors: [3, 5, 7] weights: ['uniform', 'distance']
SVM	SVC(probability=True)	C: [0.1, 1, 10] kernel: ['linear', 'rbf']
Random Forest	RandomForestClassifier()	n_estimators: [100, 200] max_depth: [None, 10]
XGBoost	XGBClassifier()	n_estimators: [100, 200] max_depth: [3, 5] learning_rate: [0.01, 0.1]

5.4.3. Evaluación de modelos.

El modelo óptimo usando cada uno de los algoritmos de aprendizaje se volvió a entrenar con todo el conjunto de entrenamiento 5-fold CV, obteniendo estimaciones estables de las métricas (*accuracy, balanced accuracy*, F1, MCC, AUC-PR) y ajuste fino de los parámetros. Cada modelo final se evaluó con cuatro conjuntos de evaluación distintos; Test (secuencias completas), SP (secuencias parciales), Sp_nc (secuencias parciales no contenidas) y 2020-24 (secuencias completas colectadas entre los años 2020 - 2024). Para la comparación entre los modelos basados en distintos algoritmos se consideró el F1 Score sobre los cuatro conjuntos de evaluación, el análisis de las matrices de confusión y las curvas AUC-PR (precisión -sensibilidad).

5.4.4. Interpretación de características

Para la combinación modelo-descriptor seleccionada en la etapa comparativa, se procedió a evaluar la importancia de las características con el fin de interpretar su contribución en la identificación del hospedero. Para ello, se aplicó la técnica *Permutation Importance*, implementada en la librería *scikit-learn* del mismo nombre.

Este método consiste en estimar el impacto de cada variable sobre el desempeño del modelo midiendo la disminución en la métrica de evaluación (en este caso F1-macro) cuando los valores de una característica son permutados aleatoriamente. Una reducción

significativa en el desempeño del modelo tras la permutación de una característica dada indica que la misma tiene un peso importante en la capacidad predictiva del modelo.

Inicialmente se identificaron 10 características con mayor importancia, que posteriormente fueron analizadas según su presencia en cada una de las regiones funcionales de la HA (ver Figura 12). Para realizar esto se mapeo cada uno de los péptidos sobre las secuencias completas de HA y luego se procedió a cuantificar la abundancia y posición en la que se encontraba cada uno de estos dipeptidos. Siguiente se procedió a separar por regiones y hospederos, cuantas de estos péptidos se encontraban en cada una de las regiones funcionales de la HAy a que hospedero corresponde cada HA mapeada. Finalmente con estos datos se procedió a realizar un test estadístico de Fisher para evaluar si hay diferencia significativa entre los hospederos y de ser así en qué regiones.

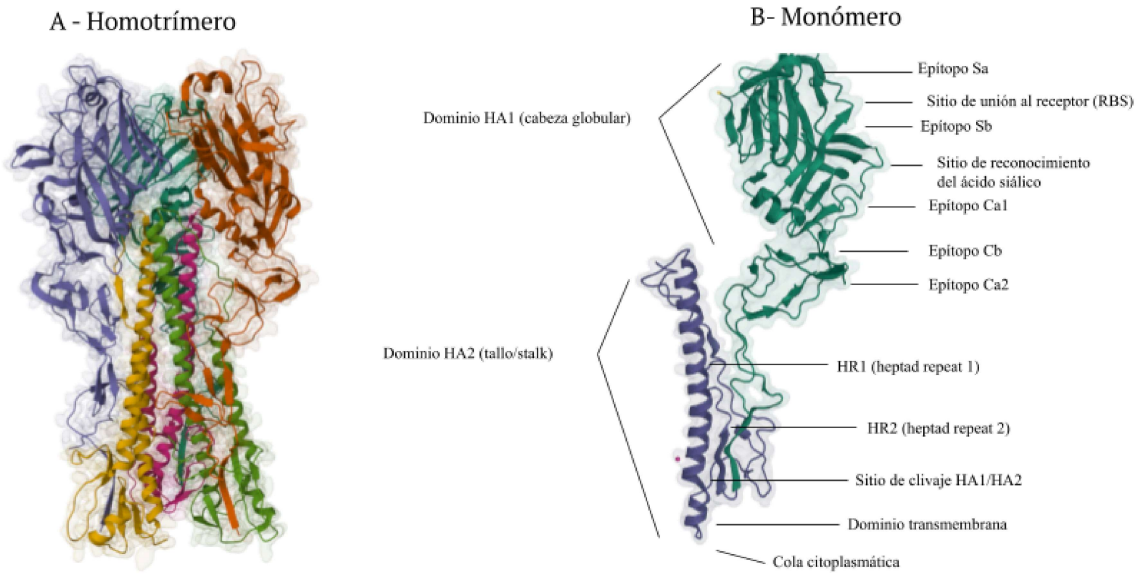


Figura 12. Estructura tridimensional de la hemaglutinina (HA, subtipo H1). La forma funcional (A) se muestra como homotrímero y la estructura de un monómero (B) con las regiones principales indicadas. *Ilustración elaborada a partir de estructuras depositadas en el PDB (ID: 1RUZ)*

5.5. Resultados

5.5.1. Exploración inicial de datos

En esta sección aplicamos dos métodos de reducción de dimensionalidad a las secuencias etiquetadas según subtipo y hospedero de origen y representadas por cada uno de los descriptores detallados más arriba. Una vez que obtuvimos una proyección de las secuencias en un espacio de 3 dimensiones, evaluamos si en el mismo las secuencias pertenecientes a cada subtipo y cada hospedero forman agrupamientos definidos.

5.5.1.1. PCA

Los resultados obtenidos tras el análisis de componentes principales se evaluaron considerando el porcentaje de variabilidad explicada por los primeros tres componentes. En cada caso, la varianza explicada por los tres primeros componentes se observa en el gráfico de la Figura 13.

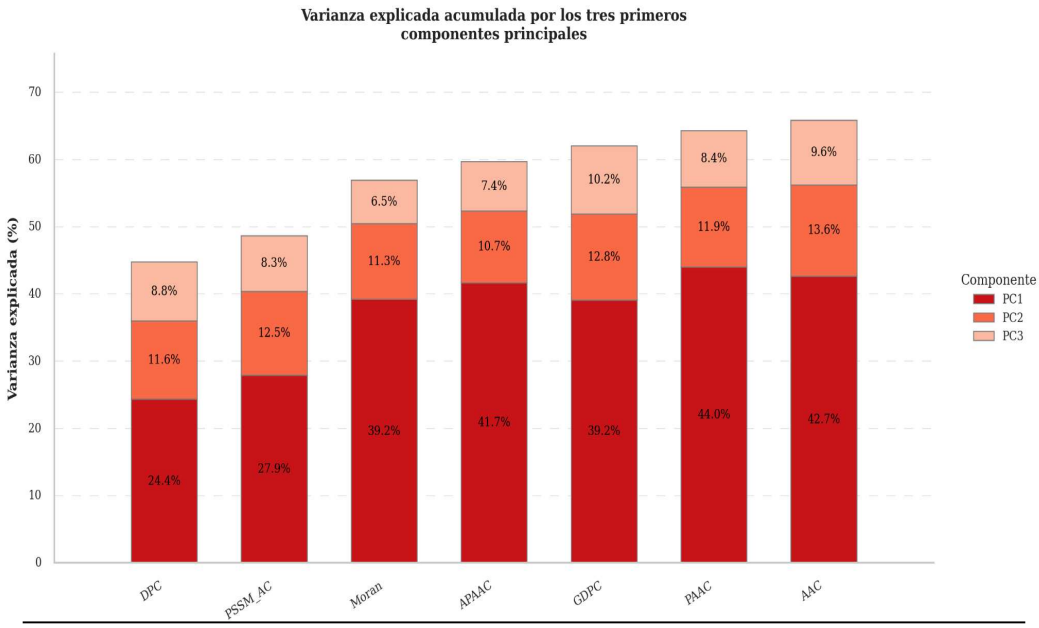


Figura 13. Varianza explicada por los tres primeros componentes principales tras aplicar análisis de componentes principales (PCA) sobre cada matriz de descriptores. Se muestra el porcentaje de varianza explicada por los componentes proyectados (PC1, PC2 y PC3) para cada descriptor.

Los descriptores que tras la reducción de dimensionalidad explican los porcentajes más altos son PAAC y AAC. En ningún caso los tres primeros componentes principales explican altos porcentajes de varianza total, siendo el máximo AAC con 65,9% y DPC el menor con 44,8%. Esto indica que los datos son complejos y no se pueden resumir fácilmente en pocas dimensiones.

En la Figura 14 se muestran las proyecciones obtenidas a partir de los descriptores lineales DPC y Moran, los cuales mostraron patrones de agrupaciones espaciales coincidentes con los subtipos y PSSM AC , como ejemplo de una de las proyecciones donde no se observan patrones coincidentes con subtipo u hospedero. En gráficos cada punto corresponde a una secuencia de HA. Se colorea cada punto según el subtipo de HA o el hospedero de origen de la secuencia, se observa que los mismos presentan una separación espacial relativamente clara en el espacio de proyección. En el caso del descriptor PSSM_AC (Figura 14, panel inferior), se observa una separación entre grupos menos definida, sin una organización espacial clara, patrón que también es observado en las proyecciones correspondientes a los demás descriptores (ver Anexo I).

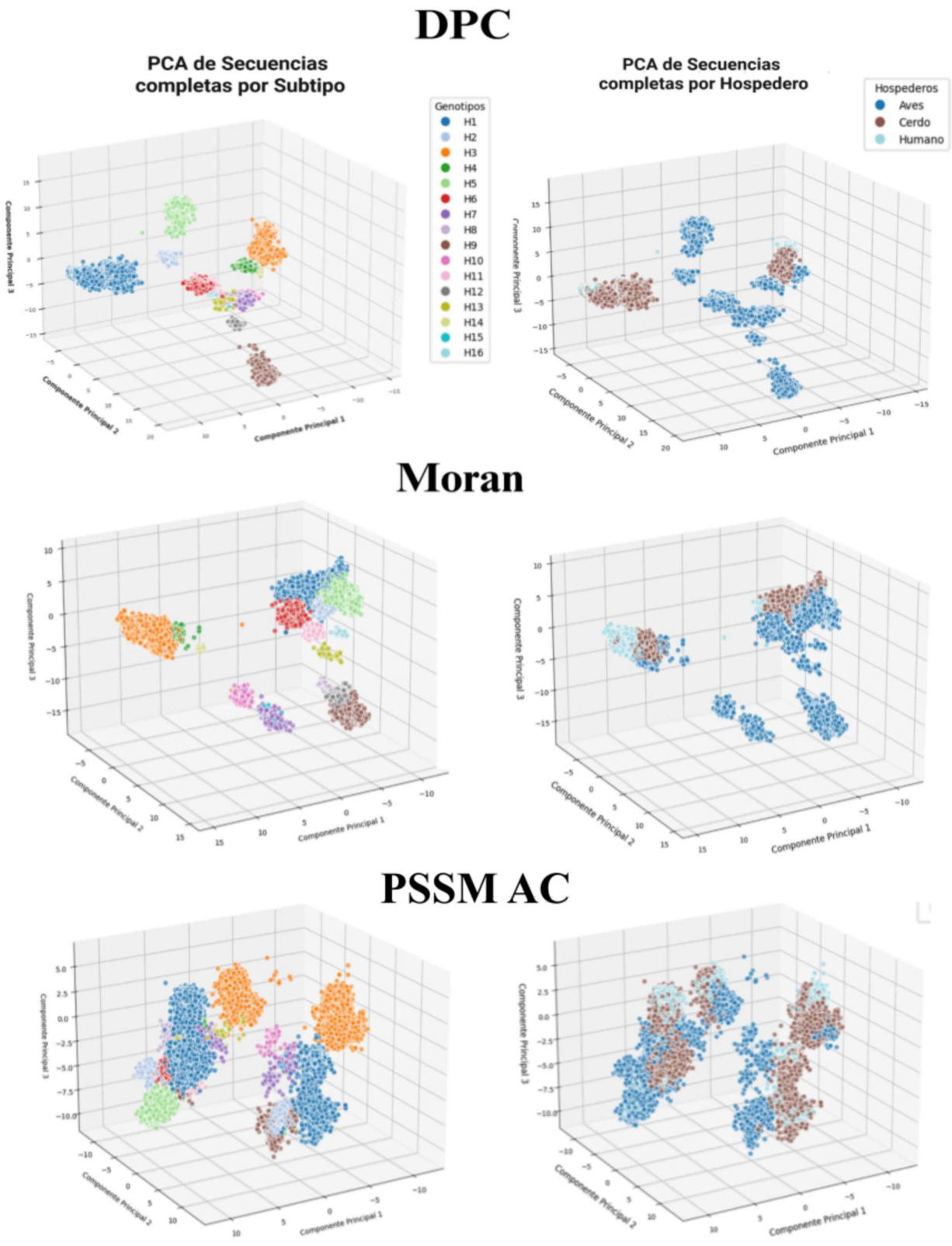


Figura 14. Proyecciones de los tres primeros componentes principales obtenidos a partir de las secuencias representadas mediante los descriptores DPC, Moran y PSSM AC. En la primera columna, cada secuencia está coloreada según el subtipo al que pertenece y en la segunda columna según su hospedero.

5.5.1.2. t-SNE

En el gráfico de la Figura 15 (panel izquierdo), se identifican agrupamientos definidos en distintas regiones, con una separación espacial marcada y sin superposiciones evidentes entre clases. Las clases mayoritarias se encuentran ampliamente dispersas en el espacio de proyección, mientras que las clases minoritarias aparecen menos representadas. En el caso de la clasificación por hospedero, se observa una superposición considerable entre las clases Humano y Cerdo, mientras que las secuencias correspondientes a Aves se agrupan de forma más diferenciada. No obstante, también se identifican algunas zonas de intersección entre las tres clases, principalmente en la región central del gráfico.

Todas las proyecciones generadas a partir de los componentes de t-SNE representan el mismo patrón entre descriptores (ver Anexo II).

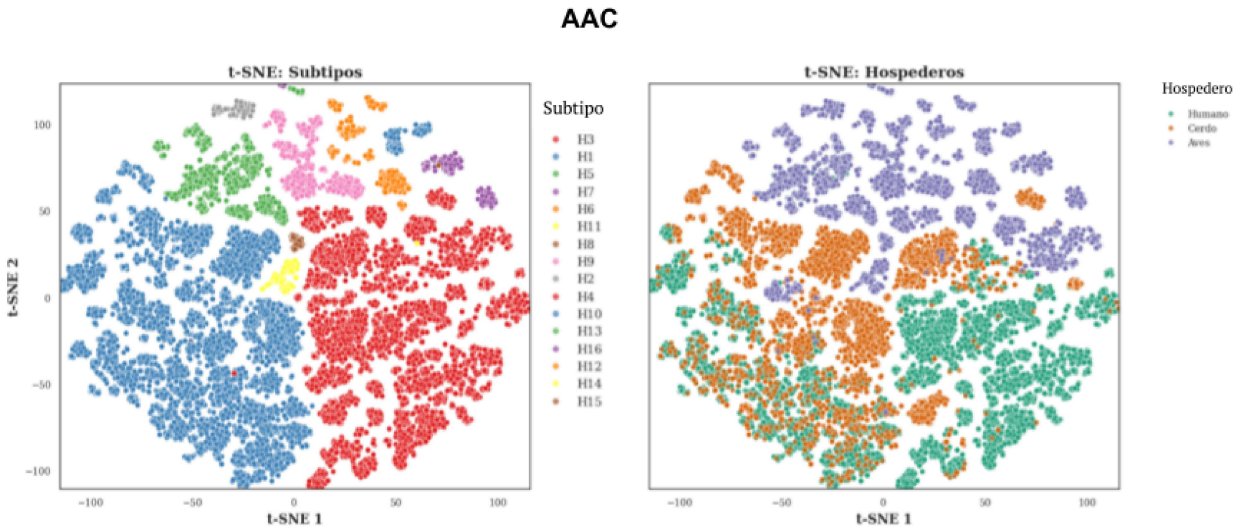


Figura 15. Proyecciones de los componentes del análisis t-SNE para el descriptor AAC sobre el Data Set 1. A la izquierda se colorean los subtipos y a la derecha los hospederos.

5.5.2. Clustering

Se realizó un análisis de agrupamiento sobre las representaciones obtenidas mediante PCA y t-SNE para cada uno de los descriptores. Para ello, se aplicaron dos algoritmos de *clustering* (k-means y GMM) y los resultados se presentan en la Tabla 3.

Tabla 3. Pureza promedio (y desvío estándar) de los clusters obtenidos con Kmeans y GMM sobre las proyecciones de PCA y t-SNE

Descriptor	PCA		t-SNE		PCA		t-SNE	
	Clusters=16				Clusters=3			
	GMM	Kmeans	GMM	Kmeans	GMM	Kmeans	GMM	Kmeans
DPC	0.956 (0.290)	0.960 (0.264)	0.794 (0.319)	0.789 (0.283)	0.718 (0.219)	0.718 (0.219)	0.691 (0.200)	0.668 (0.167)
AAC	0.907 (0.274)	0.903 (0.250)	0.848 (0.334)	0.843 (0.325)	0.647 (0.070)	0.634 (0.040)	0.636 (0.198)	0.689 (0.056)
Moran	0.904 (0.281)	0.899 (0.265)	0.801 (0.309)	0.785 (0.365)	0.699 (0.287)	0.698 (0.288)	0.583 (0.024)	0.564 (0.086)
PAAC	0.896 (0.271)	0.923 (0.282)	0.844 (0.309)	0.854 (0.254)	0.569 (0.071)	0.569 (0.100)	0.727 (0.125)	0.711 (0.074)
GDPC	0.879 (0.313)	0.879 (0.248)	0.815 (0.275)	0.846 (0.277)	0.610 (0.089)	0.689 (0.257)	0.497 (0.075)	0.482 (0.029)
APAAC	0.878 (0.229)	0.882 (0.227)	0.846 (0.393)	0.840 (0.402)	0.685 (0.252)	0.620 (0.006)	0.669 (0.201)	0.613 (0.169)
PSSM	0.808 (0.305)	0.856 (0.269)	0.813 (0.326)	0.815 (0.277)	0.686 (0.148)	0.672 (0.035)	0.634 (0.126)	0.618 (0.063)

Se evaluó cómo se distribuyen los 16 subtipos de secuencias al dividir el dataset en 16 clusters y cómo se distribuyen los hospederos de origen al dividir el dataset en 3 clusters.

Para 16 clústeres en PCA, los valores medios de pureza oscilan entre 0,808 (PSSM - GMM) y 0,960 (DPC- K-means), con desviaciones estándar que van de ±0,229 (APAAC - GMM) a ±0,313 (GDPC - GMM). En el mismo escenario, usando t-SNE, las purezas medias se sitúan entre 0,794 (DPC + GMM) y 0,815 (PSSM + GMM), con desviaciones de ±0,275 a ±0,393. En la configuración de 3 clústeres, con PCA las purezas medias varían de 0,569 (PAAC + GMM/K-means) hasta 0,718 (DPC + GMM/K-means), con desviaciones estándar entre ±0,006 y ±0,288. Bajo t-SNE y 3 clústeres, los valores medios de pureza caen aún más, fluctuando entre 0,482 (GDPC + K-means) y 0,727 (PAAC + GMM), con desviaciones que oscilan entre ±0,029 y ±0,200. Los resultados se muestran en la Tabla 3. A través de los boxplots de la Figura 16 observamos que ambos descriptores (DPC y Moran) exhiben baja dispersión en la pureza de los clusters con la configuración de 16,

DPC obtiene resultados superiores: sus valores se concentran alrededor de 1 y el algoritmo GMM es el que mejor rendimiento ofrece. Kmeans presenta mayor dispersión de los datos y valores atípicos por debajo de 0,8. Por otro lado, en la configuración de 3 clusters, únicamente DPC logra purezas superiores a 0,6 con ambos algoritmos.

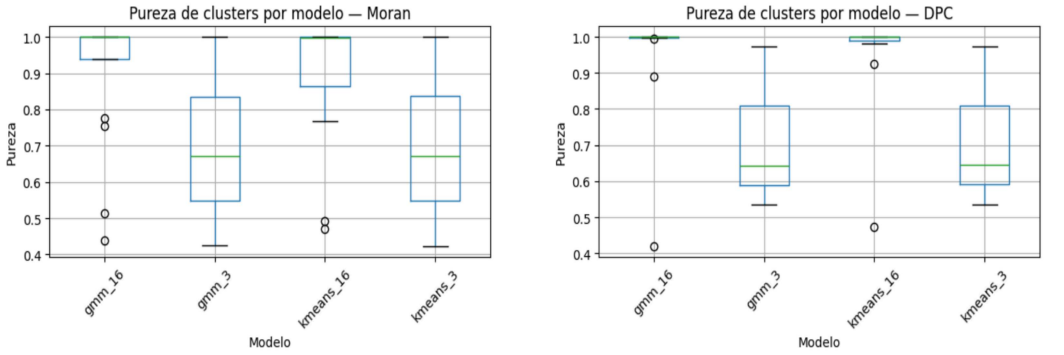


Figura 16. Pureza promedio de los clusters obtenidos con los algoritmos K-Means y GMM sobre la proyección de los tres primeros componentes principales de la representación de las secuencias mediante los descriptores Moran y DPC. La línea dentro de la caja representa la mediana (valor típico), la caja contiene el 50% central de los resultados (variabilidad), los bigotes muestran el rango típico y los puntos son outliers (corridas inusuales). Se comparó la pureza de los clústeres al considerar dos cantidades de clusters diferentes: 16 clusters para discriminar subtipos y 3 clusters para discriminar hospederos.

Los resultados con el descriptor PSSM (Figura 17) revelan que, al agrupar en 16 clusters, GMM alcanza una mediana de pureza alta (≈0,99) pero con una dispersión considerable (IQR ≈0,55–1,0 y bigotes hasta ≈0,40), K-means mantiene una mediana igualmente elevada (≈0,95–0,98) y una variabilidad menor (IQR ≈0,75–1,0, bigotes inferiores a ≈0,45). Para la configuración de 3 clusters, GMM registra una mediana de pureza alrededor de 0,78 con IQR moderado (≈0,70–0,80) y bigotes que bajan a ≈0,50, mientras que K-means exhibe una mediana ligeramente inferior (≈0,68) pero con muy poca dispersión (IQR ≈0,63–0,70 y bigotes apenas por debajo de 0,60).

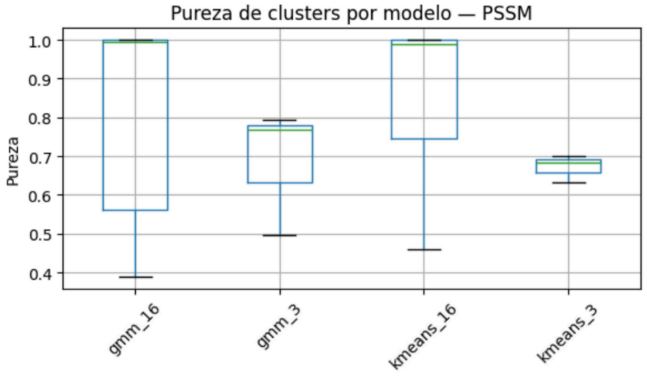


Figura 17. Pureza promedio de los clusters obtenidos con los algoritmos K-Means y GMM sobre la proyección de los tres primeros componentes principales obtenidos a partir de los descriptores PSSM. Se consideraron dos cantidades de clusters: 16 para discriminar genotipos y 3 para discriminar hospederos.

Los box plots de la pureza de los clusters obtenidos con Kmean y GMM sobre las proyecciones obtenidas con PCA y tSNE se encuentran en el material suplementario, Anexo III.

En la Figura 18 se muestran los resultados de los clusters obtenidos por ambos métodos, denotando la robustez de los agrupamientos generados por los descriptores DPC y Moran.

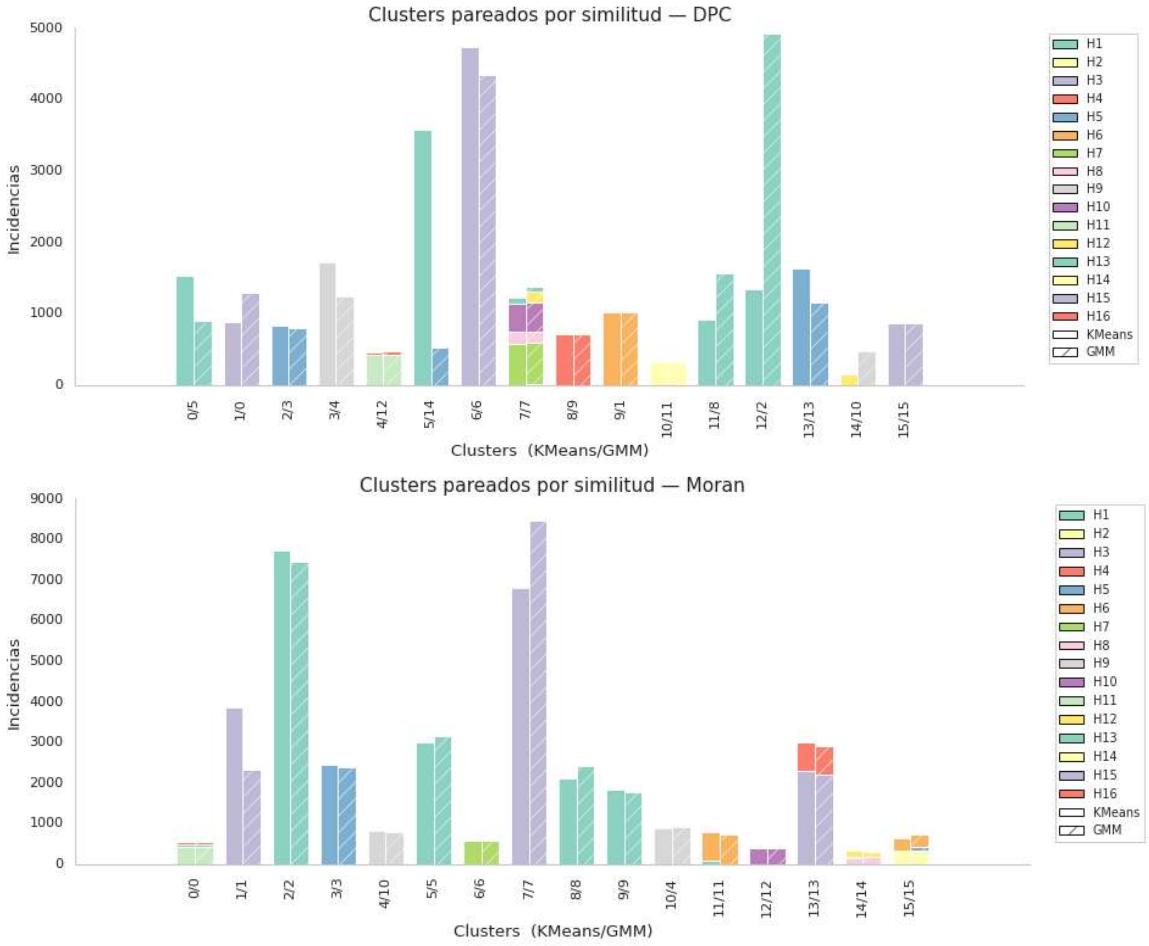


Figura 18. Distribución secuencias entre los 16 clusters para evaluar separacion por subtipo obtenidos con KMeans y GMM a partir de los tres primeros componentes principales de los descriptores DPC (panel superior) y Moran (panel inferior). Se parearon los clusters obtenidos por uno u otro algoritmo en base a la cantidad de coincidencias.

La distribución de las incidencias cuando se evalúa la separación por subtipo (16 clusters) son muy similares cuando se emplean ambos algoritmos de clustering (GMM y KMeans), lo cual sugiere que la estructura de los datos capturada en el espacio reducido por PCA es consistente independientemente del algoritmo de clustering empleado. Para el

caso de la autocorrelación de Moran, si bien los agrupamientos son más definidos en el espacio, como se observa en los boxplots, los mismos no presentan la misma pureza que cuando se utiliza DPC. A pesar de ello, las clases con alta cantidad de incidencias, como H1 y H3, están fuertemente concentradas en clústeres dominantes, estando incluso más concentradas que en los agrupamientos que se obtienen utilizando DPC.

En suma se evidencia, utilizando el análisis de componentes principales, que los descriptores DPC y Moran captaron patrones biológicos de las HA que permiten separarlas por sus subtipos.

5.5.3. Desempeño de los modelos supervisados.

En la Figura 19 se grafica el desempeño de los modelos de clasificación evaluando sobre dos conjuntos de datos que contienen secuencias completas (Test y 2020_24) y sobre dos conjuntos de datos de secuencias parciales, uno con secuencias contenidas en la base de datos de secuencias completas (Sp) y otro con secuencias no contenidas en dicha base de datos (Sp_nc).

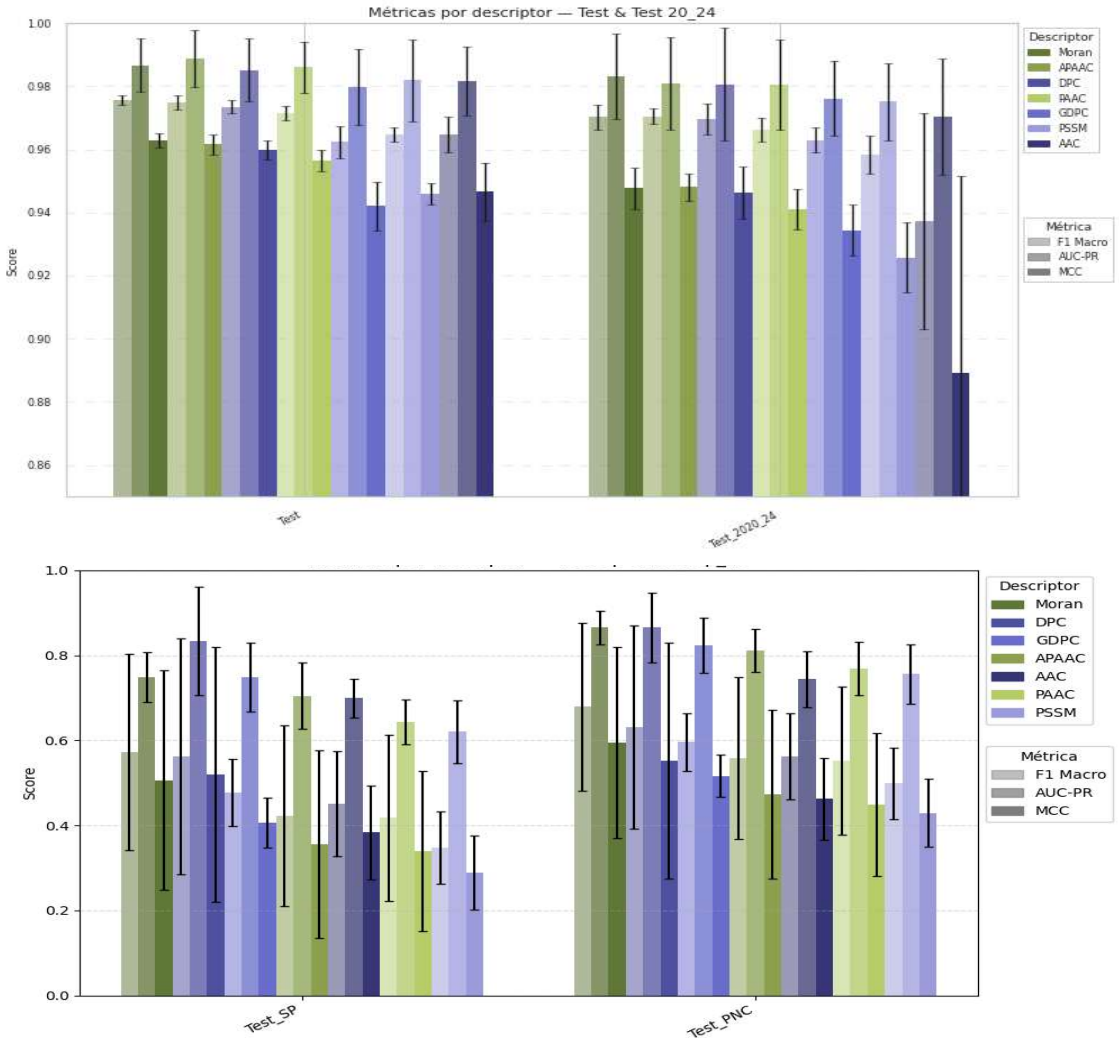


Figura 19. Desempeño de los modeloss sobre cuatro conjuntos de evaluación: *Test*, *20_24*, *Sp* y *Sp_nc*. El panel superior corresponde a las secuencias completas y el panel inferior a las secuencias parciales. Para cada descriptor, se muestra el promedio y el desvío estándar de las métricas F1 Macro, AUC-PR y MCC

Todos los modelos alcanzan valores significativamente superiores a los esperados por azar en las tres métricas utilizadas: F1-macro, AUC-PR y MCC. En particular, se destacaron los descriptores Moran, APAAC y DPC, los cuales presentaron valores superiores a 0,960 en todas las métricas, junto con una baja dispersión entre modelos, lo que sugiere una alta robustez y capacidad discriminativa consistente en diferentes algoritmos de clasificación.

En contraste, descriptores más simples como AAC dieron lugar a una caída notoria en las métricas de evaluación, especialmente en combinación con determinados modelos, lo cual se refleja en la mayor varianza observada en los gráficos. Estos resultados indican que la información contenida en AAC puede ser insuficiente para capturar la complejidad del problema de clasificación multiclase por subtipo de HA.

Cuando se evaluó el desempeño de los modelos al clasificar el conjunto 2020_24, conformado por secuencias completas pertenecientes a un período temporal no contemplado en el entrenamiento, se observaron resultados similares aunque con mayor dispersión entre modelos.

En el caso de la evaluación sobre datasets con secuencias parciales, se observó una caída clara y marcada en el desempeño de todos los modelos (Figura 17, panel inferior). Las métricas descendieron considerablemente y la variabilidad intra modelo aumentó, indicando una mayor sensibilidad a la calidad o longitud de las secuencias de entrada. No obstante, se destaca que los resultados obtenidos con las secuencias parciales no contenidas en los conjuntos de entrenamiento (Sp_nc) fueron sistemáticamente mejores. DPC y Moran resultaron ser los descriptores con el mejor desempeño sobre estos datasets y PSSM AC el de peor desempeño.

En la Tabla 4 se presentan los valores de F1-macro alcanzados por los modelos de clasificación de hospederos sobre distintos conjuntos de evaluación. Los modelos fueron seleccionados por validación cruzada y búsqueda en grilla para la optimización de hiperparámetros.

En el conjunto de prueba (Test), compuesto por secuencias completas, los valores de F1-macro oscilaron entre 0,963, observado en los modelos PSSM-KNN y AAC-XGBoost, y 0,996 correspondiente al par PAAC-KNN, que alcanzó el mejor desempeño. Para el conjunto que incluye secuencias completas obtenidas entre los años 2020 y 2024 (20_24), los valores se situaron en un rango entre 0,923 (AAC-XGBoost) y 0,975, alcanzado por los modelos Moran-SVM y DPC-XGBoost. En cuanto al conjunto de secuencias parciales

contenidas (Sp), se observó una disminución considerable en el desempeño, con valores que variaron entre 0,229 para los modelos SVM-PSSM y SVM-PAAC, y 0,953 en el caso de DPC-KNN, que obtuvo el valor más alto en este escenario. Finalmente, para el conjunto compuesto por secuencias parciales no contenidas en las secuencias completas (Sp_nc), los resultados mostraron un rango de 0,394, correspondiente a PAAC-SVM, hasta 0,959 alcanzado por DPC-KNN.

Tabla 4. F1-macro de los modelos optimizados por validación cruzada, entrenados con diferentes descriptores de secuencias y evaluados sobre diferentes subconjuntos de datos, incluyendo conjuntos con secuencias completas ("Test" y "Test 20-24") y con secuencias parciales ("SP" y "Sp_nc").

Modelo	Data Set	AAC	DPC	GDPC	PSSM	Moran	APAAC	PAAC
KNN	Test [a]	0,970	0,971	0,966	0,963	0,974	0,973	0,996
	20_24 [b]	0,948	0,964	0,964	0,958	0,969	0,968	0,965
	Sp [c]	0,608	0,953	0,576	0,432	0,873	0,723	0,652
	Sp_nc [d]	0,687	0,959	0,692	0,601	0,902	0,828	0,757
SVM	Test	0,967	0,974	0,964	0,968	0,976	0,973	0,972
	20_24	0,962	0,968	0,967	0,966	0,975	0,974	0,970
	Sp	0,370	0,338	0,382	0,229	0,311	0,253	0,229
	Sp_nc	0,499	0,433	0,534	0,396	0,423	0,416	0,394
Random Forest	Test	0,971	0,976	0,967	0,964	0,977	0,976	0,984
	20_24	0,952	0,974	0,967	0,951	0,968	0,970	0,966
	Sp	0,403	0,591	0,472	0,372	0,580	0,423	0,422
	Sp_nc	0,527	0,665	0,576	0,501	0,712	0,551	0,564
XGBoost	Test	0,963	0,974	0,955	0,965	0,975	0,977	0,979
	20_24	0,923	0,975	0,958	0,958	0,966	0,970	0,963
	Sp	0,423	0,392	0,476	0,357	0,567	0,293	0,266
	Sp_nc	0,520	0,475	0,580	0,498	0,715	0,438	0,418

a Test: Data Set de Testeo.
b 20_24: Data Set 4, secuencias comprendidas entre 2020-2024
c Sp: Data Set 2, secuencias parciales.
d Sp_nc: Data set 3 secuencias parciales contenidas.

En la tabla 5 se observan los valores de Balanced Accuracy, Precision macro, MCC y área bajo la curva PR de los descriptores DPC y Moran, que fueron los descriptores que dieron lugar a las mejores métricas.

Estos resultados indican en el caso del descriptor DPC evaluado con el conjunto Test, el clasificador basado en RandomForest registró los mejores valores de Balanced accuracy (0,972), precisión macro (0,981) MCC (0,964), y, junto a XGBoost, la mayor AUCPR (0,992). Sin embargo, al ser evaluado sobre las secuencias parciales, los mejores resultados se obtuvieron usando KNN, que superó al resto en Balanced accuracy (0,961), precisión macro (0,946), MCC (0,948) y PRAUC (0,955). Evaluado con el conjunto Test_2020_24, fue XGBoost el algoritmo que alcanzó las mejores Balanced accuracy (0,972), PRAUC (0,991) y MCC (0,955) y junto a Random Forest, también la precisión macro más alta (0,978). Durante la evaluación con el conjunto de secuencias parciales no contenidas, KNN obtuvo los valores superiores en Balanced accuracy (0,962), precisión macro (0,958), MCC (0,937) y PRAUC (0,948).

Cuando se evaluó el descriptor Moran con el conjunto Test, Random Forest obtuvo la mejor Balanced accuracy (0,974) y el MCC más alto (0,965), SVM obtuvo la mejor precisión macro (0,980) y XGBoost la PR-AUC más elevada (0,992). Utilizando el conjunto Test_2020_24, SVM registró la mejor Balanced accuracy (0,973), precisión macro (0,980) y MCC (0,957), mientras que XGBoost presentó el mejor valor de PR AUC (0,991). Utilizando las secuencias parciales, KNN fue el mejor en Balanced accuracy (0,883), precisión macro (0,865), MCC (0,859) y PR AUC (0,825). Finalmente, utilizando las secuencias parciales no contenidas, KNN obtuvo la mayor Balanced accuracy (0,896), precisión macro (0,915) y MCC (0,854), KNN y XGBoost dieron lugar a valores iguales en PR AUC (0,889).

Tabla 5. Evaluación de los modelos de clasificación de hospedero entrenados con los descriptores DPC y Moran.

Descriptor	Modelo	Data Set	Balanced accuracy	Precisión macro	MCC	PR AUC
DPC	KNN	Test [a]	0,965	0,978	0,956	0,971
		Sp [b]	0,961	0,946	0,948	0,955
		Test_2020_24 [c]	0,960	0,971	0,937	0,954
		Sp_nc [d]	0,962	0,958	0,937	0,948
	SVM	Test	0,970	0,978	0,960	0,987
		Sp	0,510	0,469	0,265	0,716
		Test_2020_24	0,967	0,970	0,943	0,987
		Sp_nc	0,537	0,587	0,314	0,821
	Random Forest	Test	0,972	0,981	0,964	0,992
		Sp	0,774	0,688	0,508	0,927
		Test_2020_24	0,971	0,978	0,954	0,990
		Sp_nc	0,739	0,742	0,568	0,896
	XGBoost	Test	0,970	0,980	0,961	0,992
		Sp	0,641	0,623	0,382	0,731
		Test_2020_24	0,972	0,978	0,955	0,991
		Sp_nc	0,606	0,673	0,389	0,775
Moran	KNN	Test	0,970	0,978	0,960	0,974
		Sp	0,883	0,865	0,859	0,825
		Test_2020_24	0,965	0,975	0,945	0,963
		Sp_nc	0,896	0,915	0,854	0,889
	SVM	Test	0,973	0,980	0,964	0,991
		Sp	0,483	0,459	0,239	0,682
		Test_2020_24	0,973	0,980	0,957	0,989
		Sp_nc	0,531	0,585	0,306	0,806
	Random Forest	Test	0,974	0,980	0,965	0,991
		Sp	0,752	0,688	0,487	0,758
		Test_2020_24	0,965	0,973	0,943	0,988
		Sp_nc	0,782	0,782	0,629	0,862
	XGBoost	Test	0,972	0,978	0,962	0,992
		Sp	0,764	0,589	0,477	0,741
		Test_2020_24	0,963	0,971	0,941	0,991
		Sp_nc	0,786	0,744	0,625	0,889

En la Figura 20 se muestran las curvas de Precisión-Recall (PR) para los descriptores y modelos que alcanzaron los mejores valores de Balanced Accuracy, Precision y MCC sobre los conjuntos “Test” y “SP”. El modelo DPC-KNN obtuvo un PR

AUC de 0,955 en secuencias parciales (SP) (Tabla 5), mientras que Moran-KNN registró 0,825 en ese mismo conjunto. Para las secuencias parciales no contenidas (Sp_nc), Moran-KNN alcanzó 0,889, frente a 0,948 de DPC-KNN. Si bien los modelos Random Forest mostraron los valores más altos en Test (Tabla 4), su PR AUC fue inferior al de KNN tanto en SP como en Sp_nc; Moran-RF presentó PR AUC de 0,758 en SP y 0,862 en Sp_nc.

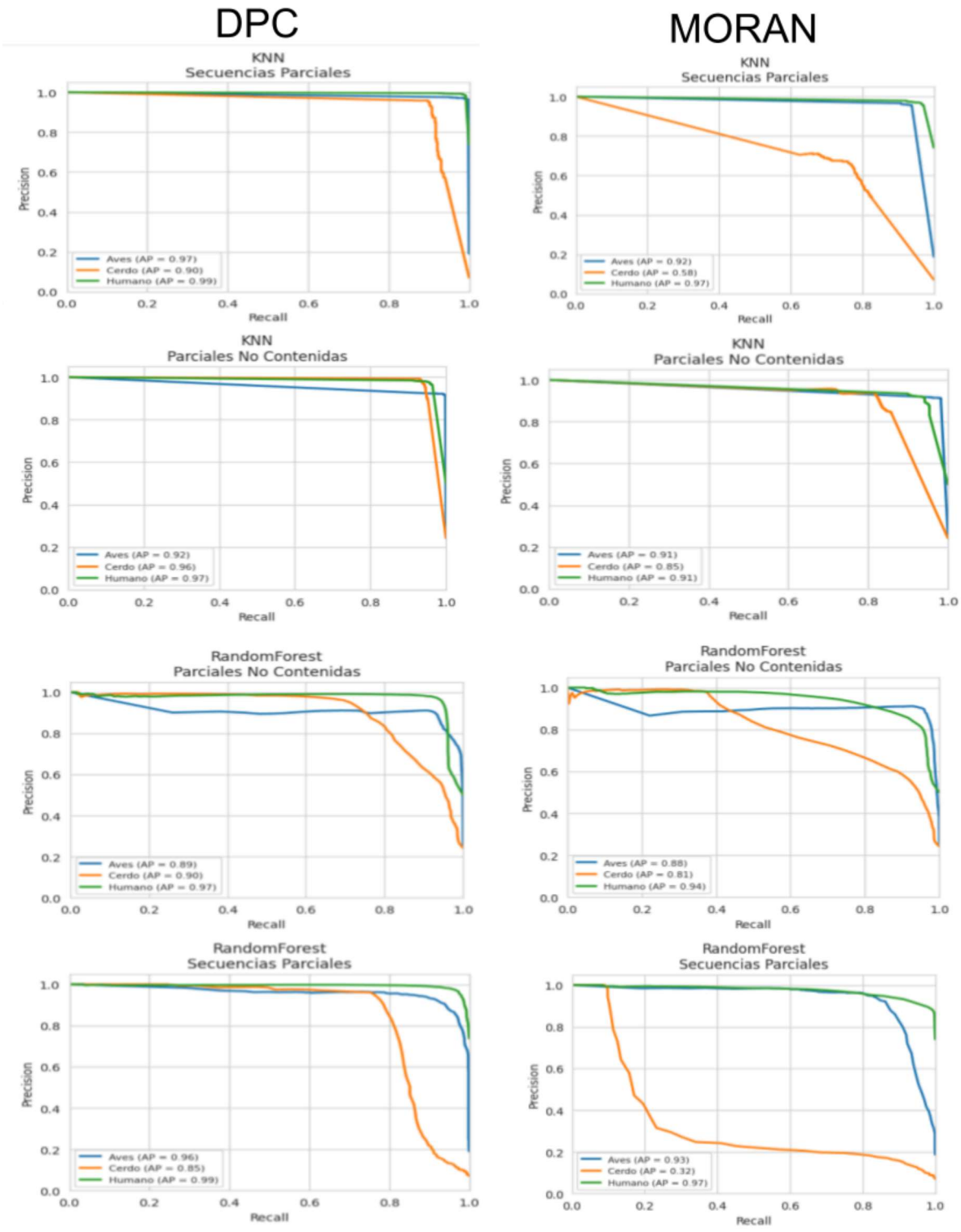


Figura 20. Curvas Precisión-Recall (PR) obtenidas durante la evaluación sobre secuencias parciales de modleos entrenados con los descriptores DPC (panel izquierdo) y autocorrelación de Moran (panel derecho). Se muestran únicamente los modelos con mejores métricas en el conjunto Test. En cada panel, la curva de la izquierda corresponde a KNN y la de la derecha a Random Forest. Los colores distinguen el hospedero: aves (azul), cerdos (naranja) y humanos (verde).

En todos los escenarios evaluados, la clase cerdo fue la que obtuvo los valores más bajos de PR AUC y disminuyó más rápidamente al cambiar el umbral de decisión del modelo. Para el caso de las clases aves y humanos, los valores de área promedio son similares. A partir de las matrices de confusión es posible identificar hacia qué clases los modelos tienden a equivocarse en la clasificación de hospederos. En la Figura 20, correspondiente a las curvas AUC-PR, se observa que, en la mayoría de los casos, los modelos presentan dificultades para clasificar correctamente las secuencias correspondientes a la clase cerdo.

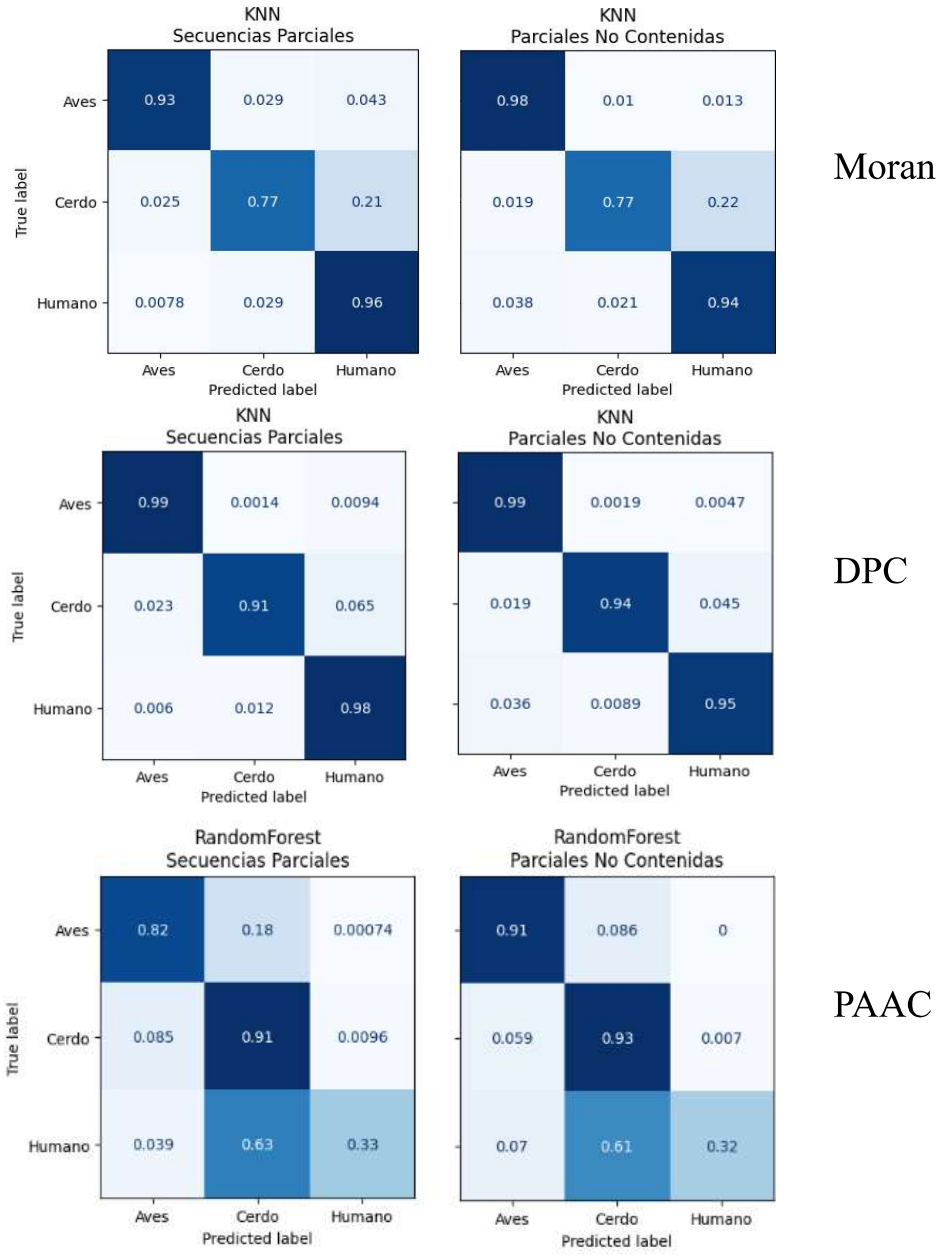


Figura 21. Matrices de confusión de secuencias parciales (Sp y Sp_nc). Los clasificadores entrenados con Moran y DPC obtuvieron los mejores resultados al clasificar secuencias parciales, mientras que el entrenamiento con PAAC-RF dió lugar a una mejor performance clasificando secuencias compeltas.

La matriz de confusión para el descriptor PAAC que se muestra en la Figura 21 (panel inferior), evidencia este comportamiento: si bien este descriptor presenta un desempeño destacado sobre el conjunto de prueba (Test), incluso superior al de DPC y Moran (ver Tabla 4), muestra dificultades específicas para clasificar correctamente las secuencias humanas, que son confundidas mayoritariamente con las de origen porcino, y en un porcentaje menor, con las de Aves (0,18 Sp y 0,86 Sp_nc).

5.5.4. Importancia de las características.

Para ordenar los dípéptidos según su impacto en la métrica F1_macro alcanzada por clasificador DPC-KNN sobre el conjunto de test (400 características) se utilizó el método Permutation Importance (Figura 22).

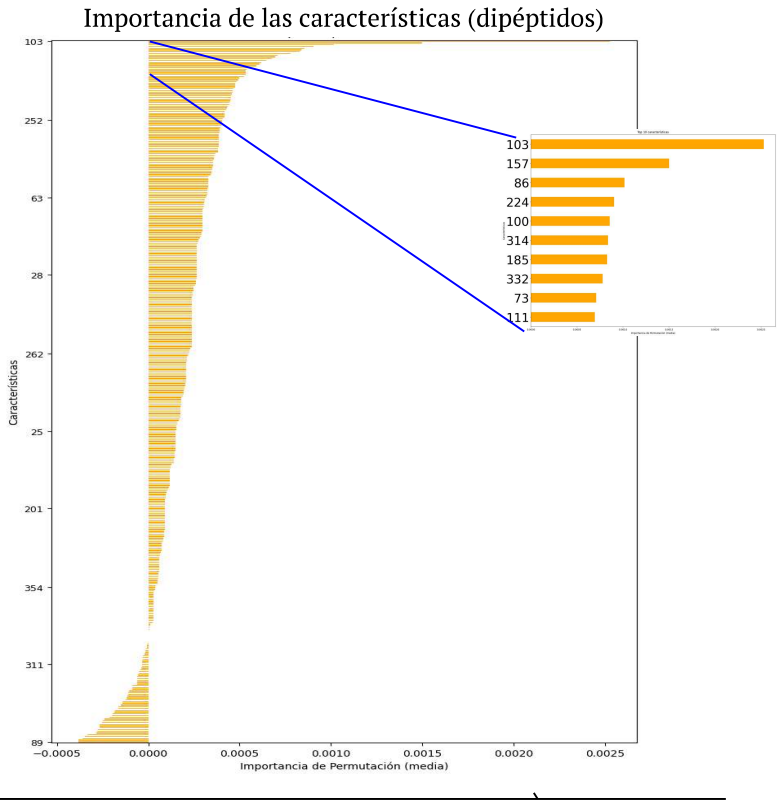


Figura 22. Análisis de importancia de características mediante Permutation Importance. La gráfica principal muestra todas las variables ordenadas por su contribución media a la métrica F1_macro y el recuadro en la esquina superior derecha amplía las 10 características más importantes.

En la Tabla 6 se recogen las diez variables más relevantes, con sus valores de ΔF_1-macro. El dipéptido GE (feature 103) alcanza el máximo ΔF_1-macro = 0,002529, un

valor ≈ 1,7 veces superior al de IV (feature 157, ΔF_1-macro = 0,001501). El ΔF_1-macro del resto de los dípéptidos oscila entre 0,001501 y 0,000695.

Tabla 6. Importancia de las 10 características con mayor contribución al F1-macro para DPC-KNN

Orden	Característica	Dipéptido	ΔF_1-macro
1	103	GE	0,002529
2	157	IV	0,001501
3	86	FH	0,001017
4	224	NF	0,000903
5	100	GA	0,000856
6	314	SR	0,000838
7	185	LG	0,000827
8	332	TP	0,000779
9	73	EQ	0,00071
10	111	GN	0,000695

Cada uno de estos dipéptidos se mapeó a lo largo de las secuencias de HA, que se representan en la Figura 23 con un strip plot. En este gráfico podemos observar que todos los dipéptidos se encuentran distribuídos a lo largo de las secuencias correspondientes a los tres hospederos y que su ubicación parece tener regiones más frecuentes. En Aves, la distribución a lo largo de la secuencia parecería ser más uniforme. En las secuencias de cerdo y humano esta distribución parece más concentrada en ciertas regiones y más similares entre sí.

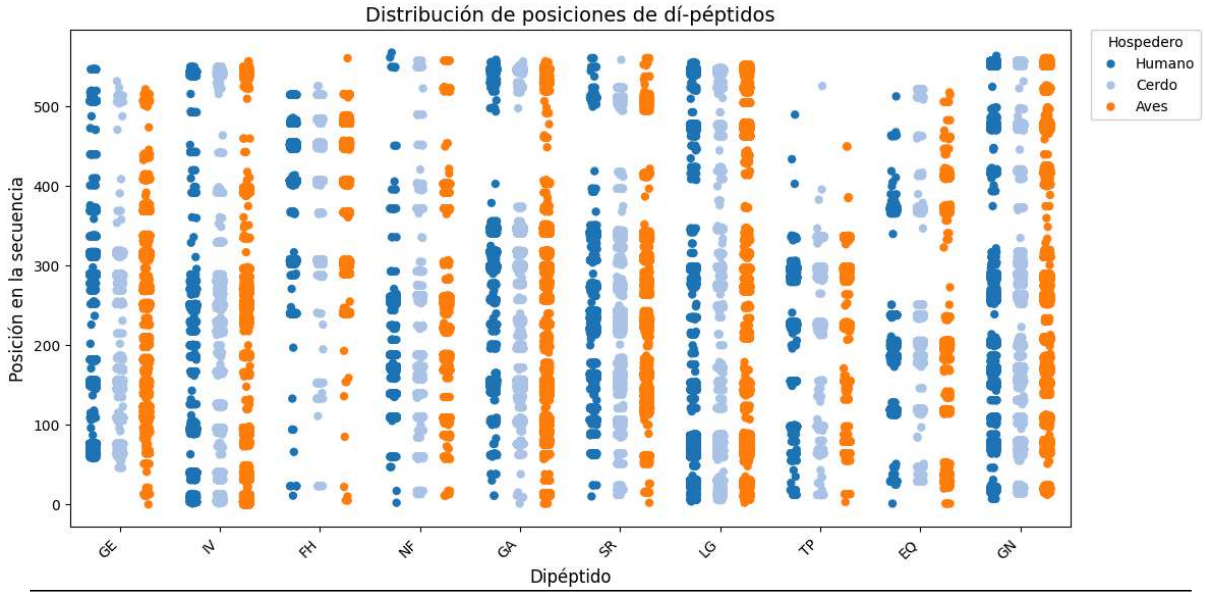


Figura 23. Distribución de los 10 dipéptidos con mayor importancia para la clasificación a lo largo de las secuencias de hemaglutinina, representada mediante un strip plot. Los resultados para cada hospedero (Aves, Humanos y Cerdos) se muestran en distinto color.

A partir del mapeo en la secuencia de HA se obtuvieron pares posición-dipéptido a los cuales se les asoció a una región de la HA definidas (Figura 12). La cantidad de veces

que cada dipétptido fue mapeado en cada una de estas regiones se muestra en la Tabla 7. Se puede observar que a lo largo de ambas regiones ,HA 1 y 2 se concentran una gran cantidad de hits. Algunas regiones más cortas, como la cola citoplasmática, el epítopo cb y el péptido de fusión, también muestran buena representación en hits. El sitio de unión al receptor (RBS), con regiones no contiguas, presenta una cantidad de hits moderada y relativamente equilibrada en los tres hospederos analizados.

Tabla 7. Cantidad de *hits* por regiones funcionales de HA de los 10 dipéptidos más relevantes para la clasificación.

Región	Posición	Hits Humano	Hits Aves	Hits Cerdo	Hits totales
Péptido señal[1]	1–16	9 265	10 671	1 552	21 488
Dominio HA1 (cabeza globular)[1]	17–345	420 037	206 647	206 873	833 557
Sitio de unión al receptor (RBS)[2,3]	131–147, 190–198, 221–228	12 645	12 067	11 531	36 243
Sitio de reconocimiento del ácido siálico[2,3]	98, 135–138, 153, 155, 183, 190, 194–195, 224–226, 228	6 047	4 557	7 501	18 105
Sitio de clivaje HA1/HA2[1]	345–346	1 672	2 370	3 489	7 531
Péptido de fusión (HA2 N-terminal)[3,5]	346–367	23 919	7 527	12 068	43 514
Dominio HA2 (tallo/stalk)[1,3]	346–561	117 566	60 346	69 604	247 516
HR1 (heptad repeat 1)[3]	384–410	12 459	6 861	3 499	22 819
HR2 (heptad repeat 2)[3]	482–519	901	3 845	651	5 397
Epítopo Sa [4]	128–129	282	195	30	507
Epítopo Sb [4]	187–198	12 309	4 177	1 493	17 979
Epítopo Ca1 [4]	169–173, 206–208, 238–240	13 166	4 418	8 083	25 667
Epítopo Ca2 [4]	140–145, 224–225	1 569	4 210	961	6 740
Epítopo Cb [4]	74–79	35 833	15 073	24 111	75 017
Dominio transmembrana[3,6]	519–543	2 105	3 973	997	7 075
Cola citoplasmática[3,6]	544–566	42 794	6 987	25 197	74 978

1 = Burke & Smith, 2014.
2 = de Graaf & Fouchier, 2014.
3 = Skehel & Wiley, 2000.
4 = Caton, Brownlee, Yewdell, & Gerhard, 1982.
5 = Lorieau, Louis, & Bax, 2010.
6 = Veit & Thaa, 2011.

A partir de los conteos de hits por hospedero y por región, se aplicó la prueba exacta de Fisher para evaluar la asociación entre regiones funcionales de la HA y la

ubicación de los dipéptidos importantes implicados en la identificación del hospedero. Adicionalmente, se calculó un odds ratio (OR) ajustado por desbalance de clases, considerando las proporciones normalizadas por el tamaño de cada grupo, con el fin de reducir el sesgo derivado del mayor número de secuencias humanas en el conjunto de datos.

La Tabla. 8 presenta las regiones de HA que mostraron enriquecimiento en hits estadísticamente significativo (p-value < 0,05), tras corregir por desbalance de clase y aplicar la prueba exacta de Fisher. Se incluyeron los resultados de los OR ajustados ⩾ 1.8 ordenadas por orden descendente. Para el caso de Aves se encontraron cuatro regiones que podrían ser importantes para selección de este hospedero: HR2, Epítopo Ca2, Dominio transmembrana y Péptido señal. En todos ellos el OR ajustado presenta valores > 3, lo cual indica que la cantidad de hits en esa región de la HA de aves es estadísticamente significativa, con valores de p-value < 0,001. La cantidad de hits es variada, siendo la región correspondiente al péptido señal la que incluye la mayor cantidad de hits (21.488) y un OR ajustado de 3,52.

Para el caso de humanos solo el Epítopo Sb mostró una cantidad de hits (17.979) significativamente mayor a la esperable por azar, con un OR = 2,19 y un p_value < 0,001, indicando sobre representación localizada en esta región globular. En cerdos se encontraron dos regiones con sobrerrepresentación de hits: el sitio de clivaje HA1/HA2

Tabla 8. Evaluación de la asociación entre regiones funcionales de la hemaglutinina y distribución de dipéptidos mediante la prueba exacta de Fisher, para distintos hospederos. Se muestran los resultados obtenidos con el modelo KNN entrenado con DPC.

Región	Hospedero	OR_clasico	Hits totales	p_value	OR_ajustado
HR2 (heptad repeat 2)	Aves	7,70	5.397	< 0,001	7,27
Epítopo Ca2	Aves	5,18	6.740	< 0,001	5,01
Dominio transmembrana	Aves	3,98	7.075	< 0,001	4,01
Péptido señal	Aves	3,10	21.488	< 0,001	3,52
Epítopo Sb	Humano	2,25	17.979	< 0,001	2,19
Sitio de clivaje HA1/HA2	Cerdo	2,45	7.531	< 0,001	2,04
Sitio de reconocimiento del ácido siálico	Cerdo	2,02	18.105	< 0,001	1,86

(OR = 2,04; p_value < 0,01) y el sitio de reconocimiento de ácido siálico (OR = 1,86; p value < 0,001), con aproximadamente 7.500 y 18.100 hits, respectivamente.

5.5.5. Secuencias ambiguas

Utilizando el modelo que alcanzó las mejores métricas de desempeño, se clasificaron las secuencias "ambiguas", esto es, que estaban asociadas a más de una etiqueta. Estas incidencias corresponden a una misma secuencia que fue aislada de dos organismos diferentes. En la Tabla 9 se muestran, a modo de ejemplo, algunas de estas secuencias y la clasificación que se obtuvo con el modelo aplicado. Se destaca de este análisis que el modelo siempre asignó uno de los dos hospederos de los cuales fue realmente aislado el virus. Por otro lado, las probabilidades asociadas a estas clasificaciones oscilan entre 0,517 y 1.

Tabla 9. Predicciones de los modelos sobre incidencias con etiquetas ambiguas.

Secuencia	Accession	Clase	Predicción	Proba
1	ABV23936 ABV24020	Aves Humano	Aves	0,643
2	ABV24009 ABV24031	Humano Aves	Aves	0,665
3	AFD32765 AFD32745	Humano Cerdo	Humano	1
4	ANZ03534 ADF10107	Cerdo Humano	Humano	0,610
5	AEA29584 ADF10929	Cerdo Humano	Cerdo	0,517
6	AGK24377 AFN87932	Cerdo Humano	Cerdo	0,793
7	AUO37594 ABP64751	Humano Aves	Aves	1
8	ADK22448 ADF83664	Aves Humano	Aves	0,677

5.6.Discusión

5.6.1. PCA, t-SNE y Clustering

La suma total de la varianza explicada por los primeros componentes del PCA es un indicador de cuánta información global del conjunto de datos se está conservando, pero no es el único criterio para evaluar la utilidad de un descriptor para tareas de clasificación

(Xue et al., 2014). Aunque los primeros tres componentes principales explican menos del 50% de la varianza total, como sucede cuando se consideran las secuencias representadas por DPC y Autocorrelación de Moran, el hecho de que en ese espacio se observan agrupamientos que, al menos en algunos de los casos, se corresponden con los subtipos de HA, sugiere que estos descriptores y estas componentes están capturando información discriminativa relevante. Es posible que los primeros componentes principales derivados de descriptores lineales como AAC, que no captan relaciones espaciales, no reflejen la diversidad biológica, aún explicando mayor porcentaje de varianza explicada, lo cual podría deberse a la longitud del vector (20 para AAC). Los descriptores con mayor número de características, como DPC (400 variables), presentan una dispersión más alta en sus valores, lo que hace que el PCA requiere más componentes para capturar adecuadamente un porcentaje importante de la varianza total (Figura 13) .

Al aplicar dos algoritmos de clustering, (Kmeans y GMM) a la proyección de los datos en el espacio de 3 dimensiones definido por los tres primeros componentes principales, se observaron resultados similares en cuanto a la generación de agrupamientos bien definidos tanto cuando se usó DPC como cuando se usó la Autocorrelación de Moran. Éste último presenta mayor reproducibilidad en los clusters generados por ambos algoritmos, debido a que tiene agrupamientos mejor definidos, aunque en la configuración de 16 clusters, los clusters obtenidos con los datos representados por DPC presentan mayor pureza, (0,960 DPC-Kmean vs 0,904 Moran-GMM) . Los demás descriptores no dieron lugar a estructuras bien definidas en el espacio, lo cual se refleja en la baja pureza de los clusters obtenidos. La reducción de dimensión obtenida con t-SNE no capturó información relevante con este tipo de datos. Las proyecciones de los componentes de t-SNE en el espacio son mayormente coherentes con la discriminacion de subtipo, al igual que cuando se usa PCA, pero al no generar agrupamientos definidos, sino más bien solapados en el espacio, es más difícil su asignación a un subtipo particular, lo cual queda reflejado en los cluster obtenidos tanto por kmean como por GMM.

PSSM AC fue el único de los descriptores utilizados con información evolutiva sobre las secuencias, sin embargo, la dispersión espacial que se observa en las proyecciones resultantes no evidencia un patrón claro de agrupamientos. Sería necesaria una exploración más profunda de estas proyecciones para evaluar si las mismas representan cambios evolutivos que puedan estar asociados a los diferentes hospederos o subtipos. Esta dispersión puede deberse a la propia naturaleza del virus, a su variabilidad o al acervo genético dentro de las clases estudiadas (Wang et al., 2017).

En suma, cuando se aplicaron técnicas de reducción de dimensionalidad (PCA, t-SNE) y luego se hizo clustering (K-means, GMM) los descriptores DPC y Moran superaron al resto, dando lugar a los cluster mejor separados y más puros. Entre ambos, DPC se perfila como la mejor opción, ya que ofrece un rendimiento ligeramente superior y tiene un costo computacional menor (O(n)).

5.6.2. Modelos Supervisados

En este estudio se consideraron cuatro algoritmos de aprendizaje supervisado (k vecinos más cercanos, máquinas de vectores de soporte, bosque aleatorio y XGBoost) y siete descriptores de secuencia, con el objetivo de entrenar modelos capaces de inferir el hospedero de origen. Estos algoritmos constituyen enfoques ampliamente utilizados en clasificación supervisada (Cortes & Vapnik, 1995; Breiman, 2001; Chen & Guestrin, 2016).

Cuando estos modelos son entrenados, deben considerarse dos aspectos centrales al evaluar su desempeño: el sobreajuste (*overfitting*) y el subajuste (*underfitting*). El sobreajuste ocurre cuando el modelo se adapta demasiado bien a los datos de entrenamiento —incluyendo variaciones propias del ruido—, por lo que alcanza un rendimiento elevado en ese conjunto, pero generaliza mal y su desempeño cae al evaluarse sobre datos no vistos (Pedregosa et al., 2011). En cambio, el subajuste se produce cuando el modelo no logra capturar la estructura subyacente de los datos, ya sea por una capacidad insuficiente, una parametrización demasiado restrictiva o un conjunto de características poco informativo; en este escenario, el rendimiento es bajo tanto en entrenamiento como en validación o prueba (Pedregosa et al., 2011).

Ambos fenómenos se relacionan con el compromiso sesgo–varianza: modelos demasiado simples tienden a presentar alto sesgo (subajuste), mientras que modelos excesivamente complejos tienden a presentar alta varianza (sobreajuste), lo que afecta la capacidad de generalización (Bishop, 2006; Hastie et al., 2009). En la práctica, una forma útil de diagnóstico consiste en comparar el desempeño entre entrenamiento y validación: una gran brecha entre ambos sugiere sobreajuste, mientras que valores bajos en ambos conjuntos sugieren subajuste (Pedregosa et al., 2011).

Para mitigar el sobreajuste, suelen emplearse estrategias como la separación adecuada de conjuntos (entrenamiento/validación/prueba) y el uso de validación cruzada para estimar el desempeño de generalización y ajustar hiperparámetros sin sesgo optimista (Pedregosa et al., 2011), lo cual se realizó en el presente trabajo.

Asimismo, se aplicaron técnicas de regularización (penalizaciones sobre los parámetros), como control explícito de la complejidad del modelo, con el objetivo de reducir el error de generalización (Goodfellow et al., 2016). Por el contrario, cuando se detecta subajuste, suele ser necesario aumentar la capacidad del modelo (o flexibilizar hiperparámetros), incorporar descriptores más informativos o ajustar el preprocesamiento para capturar mejor la señal presente en los datos (Bishop, 2006; Hastie et al., 2009). Estos pasos de ajustes de los modelos se llevaron a cabo de forma automatizada utilizando validación cruzada y la búsqueda en grilla de mejores hiperparametros previamente establecidos.

Por otro lado, las secuencias biológicas representan un desafío particular para los modelos de AA porque cambios mínimos en posiciones específicas pueden tener consecuencias biológicas importantes a nivel fenotípico. En los VIA subtipos H5 y H7, por ejemplo, la presencia o modificación de un sitio de clivaje polibásico en la hemaglutinina se asocia fuertemente a alta patogenicidad en aves, y variaciones puntuales en esa región pueden modular la virulencia (Suguitan Jr. et al., 2012; Zhang et al., 2012). Del mismo modo, sustituciones en proteínas internas como PB2 (p. ej., en posiciones 627 y 701) se han vinculado con cambios en adaptación a mamíferos y transmisión, ilustrando cómo una o pocas sustituciones pueden impactar propiedades complejas del virus (Steel et al., 2009).

Además, en problemas de clasificación basados en secuencias es frecuente que el conjunto de entrenamiento y el de prueba contengan secuencias altamente relacionadas (con muy pocas diferencias), lo que puede inflar artificialmente las métricas si existen solapamientos o similitudes elevadas entre particiones. Este fenómeno se ha descrito como una forma de fuga de información (“*data leakage*”) o evaluación optimista cuando se utilizan particiones aleatorias en datos biológicos, ya que el modelo puede “resolver” el problema apoyándose principalmente en similitud de secuencia en lugar de aprender reglas generalizables (Bernett et al., 2024 ; Kaufman et al., 2012). Por esta razón, en varios *benchmarks* se recomienda reducir redundancia y controlar la identidad de secuencia entre conjuntos para obtener estimaciones más realistas del desempeño (Xu et al., 2022).

En este contexto, la utilización de un conjunto adicional de secuencias parciales como evaluación externa aporta un criterio más exigente de generalización: obliga al modelo a mantener un desempeño adecuado aun cuando la entrada sea incompleta, lo que aproxima mejor escenarios reales (por ejemplo, secuencias parciales, fragmentadas o de diferente cobertura). De este modo, un modelo con buen poder de generalización debería exhibir un rendimiento consistente no solo sobre el conjunto de prueba con secuencias

completas, sino también sobre el conjunto de secuencias parciales, reduciendo el riesgo de conclusiones basadas únicamente en similitud cercana entre particiones (Bernett et al., 2024).

Se observó que el desempeño de los modelos fue muy alto cuando la evaluación se realizó sobre secuencias completas de hemaglutinina con todos los descriptores. Esto es consistente con la biología de la hemaglutinina: aunque sufre cambios frecuentes por mutación (deriva antigénica) y reordenamiento (cambio antigénico) —y solo raramente por recombinación—, mantiene funciones esenciales evolutivamente conservadas (unión al receptor y fusión de membranas) y una arquitectura estructural global que impone fuertes restricciones funcionales (Shao et al., 2017; Skehel & Wiley, 2000). Además, existen regiones particularmente conservadas relacionadas con el proceso de fusión (por ejemplo, el dominio de fusión en HA2), lo que refuerza la idea de "conservación funcional" aún en un contexto de alta variabilidad (Lorieau et al., 2010; Wu & Wilson, 2020).

En nuestro caso, durante la curación de la base de datos no se eliminó la redundancia de secuencia (p.ej. agrupamiento a ⩾90 % de identidad), lo que implica que secuencias muy similares podrían estar presentes tanto en el conjunto de datos utilizado para el entrenamiento como en el utilizado para la evaluación, como se mencionó anteriormente. Esto se debe a la naturaleza de la base de datos, en la que existe un sesgo inherente a la cantidad y tipo de secuencias contenidas en el servidor (Ao et al, 2022). Por ejemplo los subtipos H1, H3 y H5 son de mayor predominancia, debido a la atención que ha requerido a lo largo del tiempo la vigilancia de estos virus.

Los resultados correspondientes al período 2020-2024 no mostraron comportamientos atípicos en comparación con el conjunto de prueba (Test), a pesar de que en este período surgió la epizootia del virus H5N1, que afectó a varias especies. La decisión de no incluir estos datos en el conjunto de entrenamiento tuvo como objetivo evaluar el desempeño de los modelos con datos de los tres hospederos más relevantes del virus durante la epizootia (que se originó en aves), esperando que, en caso de haberse producido algún cambio significativo en la secuencia de la proteína HA, éste pudiera reflejarse en los resultados. Sin embargo, el comportamiento observado sobre este conjunto fue similar al obtenido en el conjunto de prueba, lo que sugiere que las secuencias del período 2020–2024 no aportan evidencia de variaciones sustanciales en las características detectadas por los modelos.

La performance de los modelos basados en algunos de los descriptores que incorporan autocorrelación secuencial, como PAAC, se vió notablemente afectada al ser evaluada con secuencias parciales. Por ejemplo, el modelo basado en PAAC presentó un F1-score de 0,996 en el set de prueba con secuencias completas, que descendió a 0,652 cuando se evaluó sobre secuencias parciales. Esta caída de rendimiento se debe a que las secuencias parciales presentan no solo frecuencias relativas de aminoácidos alteradas, sino también una distorsión en los valores de correlación secuencial, ya que estos dependen directamente de la posición relativa de los residuos en la secuencia. En particular, propiedades como la hidrofobicidad, la hidrofilicidad y la masa molecular de las cadenas laterales influyen en los términos θ de PAAC, cuya estabilidad se ve comprometida si la proteína fue truncada en regiones funcionales o terminales (Chou, 2001).

La capacidad de los modelos para discriminar resultó estar condicionada por el largo de las secuencias que se utilizaron para la extracción de características. En efecto, todos los modelos tuvieron un mejor desempeño general cuando se utilizaron secuencias parciales no contenidas en las secuencias completas (Sp_nc), que cuando se utilizaron secuencias parciales contenidas (Sp) (Figura 19, panel inferior). Si bien se esperaría que los modelos se comporten mejor al intentar clasificar secuencias parciales contenidas en secuencias usadas para el entrenamiento, parece haber prevalecido el efecto del tamaño de las secuencias (Figura 6). El único descriptor que no presentó este patrón fue DPC, reflejando su robustez y poder de generalización.

Por todo lo antes dicho la selección final del modelo se basó fundamentalmente en la capacidad para utilizar secuencias parciales. En este sentido, dos de los descriptores evaluados presentan desempeño superior: DPC y Moran, en combinación con el algoritmo KNN. El desempeño más bajo del modelo entrenado con Autocorrelación de Moran se debe principalmente a su limitada capacidad para discriminar correctamente al hospedero Cerdo, como se evidencia en las curvas AUC-PR (Figura 20.), donde tanto la sensibilidad como la precisión del modelo para esta clase se ven marcadamente comprometidas al variar el umbral de decisión.

El hospedero Cerdo es reconocido como un "mezclador" viral, dada su capacidad de infectarse simultáneamente con virus aviares y humanos, lo cual facilita la aparición de virus reordenantes (Long et al., 2021). Esta particularidad podría explicar, al menos en parte, la dificultad que enfrentan los modelos para clasificar de forma precisa las secuencias porcinas. Las matrices de confusión muestran una tendencia frecuente a la confusión bidireccional entre las clases Cerdo y Humano, lo que refuerza esta hipótesis.

Por otro lado, aunque el modelo DPC-KNN no fue el que alcanzó el mejor desempeño sobre el conjunto de prueba (Test), fue seleccionado como el modelo óptimo, dado que demostró una mayor robustez frente a la variabilidad de las secuencias parciales, manteniendo una buena capacidad de generalización. Habiendo seleccionado este modelo se continuó con los análisis de importancia de características, con el objetivo de interpretar las características más relevantes para la clasificación correcta de hospederos.

5.6.3. Importancia de características

El análisis de importancia de características mediante el método Permutation Importance aplicado al modelo KNN-DPC, permitió identificar los dipéptidos más relevantes para la clasificación de secuencias de HA según el hospedero. Estos dípeptidos fueron luego mapeados a las regiones funcionales de la HA y se evaluó su distribución diferencial entre las mismas mediante la prueba exacta de Fisher, con el objetivo de detectar asociaciones significativas entre regiones funcionales y la ubicación de los dipéptidos más relevantes para distinguir entre distintos hospederos (humano, cerdo, aves). A pesar de que algunas regiones presentaron un número absoluto de hits relativamente bajo —como HR2 (5.397 hits), el epítopo Ca2 (6.740 hits) o el dominio transmembrana (7.075 hits)—, los resultados estadísticos mostraron asociaciones significativas con determinados hospederos.

Los dipéptidos relevantes ubicados en estas tres regiones fueron significativamente más frecuentes en secuencias clasificadas como aviares, con odds ratios ajustados de 7,27 (HR2), 5,01 (Ca2) y 4,01 (dominio transmembrana), respectivamente ($p < 0,001$ en todos los casos). Este resultado sugiere que estas regiones contienen patrones moleculares distintivos que los modelos de clasificación aprenden a reconocer como característicos del hospedero aviar. Desde el punto de vista biológico, estas asociaciones son altamente plausibles. La región HR2 (heptad repeat 2), ubicada en el tallo de HA2, participa directamente en el proceso de fusión de membranas, siendo determinante para la estabilidad conformacional de la proteína y para la eficiencia del ingreso viral, y de hecho, mutaciones en esta región han sido asociadas con cambios en el pH de fusión y adaptaciones a nuevos hospederos. Por su parte, el epítopo Ca2 (análogo a una región A en H5), es una región antigénica localizada en la cabeza globular de HA1 y su variabilidad ha sido ampliamente documentada como un mecanismo de evasión del sistema inmune, particularmente en el contexto de la deriva antigénica en aves (Luczo et al., 2024). El dominio transmembrana, aunque altamente conservado en términos estructurales, puede presentar diferencias sutiles en la composición de aminoácidos que afectan el anclaje, ensamblaje y liberación de partículas virales, en función del tipo celular del hospedero.

Finalmente en el péptido señal también se encontró diferencias significativas para aves. En la bibliografía no hay evidencias fuertes de la posibilidad de que esta región sea determinante del tropismo del virus por lo que se debería de investigar más al respecto.

En el caso de humanos, tras el mapeo de los 10 dipéptidos más relevantes para la clasificación se encontró una fuerte asociación con las regiones del epítopo Sb. En esta región de la cabeza globular se ha encontrado que mutaciones puntuales de algunos aminoácidos resultan en una disminución de la afinidad del receptor de unión a humanos (Xu et al., 2022), lo cual podría explicar nuestro resultado. Por otro lado, la región del tallo o "stalk" de la HA, altamente conservada, ha sido identificada como un objetivo prometedor para el desarrollo de una vacuna universal contra la influenza (Nuwarda et al., 2021).

Por otro lado, también se encontró una fuerte asociación entre la ubicación de los dipéptidos relevantes para la clasificación y los sitios de clivaje HA1/HA2, que determinan la capacidad infectiva del virus y en el caso de especies aviares también están relacionados con la patogenicidad. Este sitio en el precursor HA0 es clivado por una proteasa y forma los dos monómeros HA1-HA2, unidos por puente disulfuro (Rajao et al., 2019). En el caso de los cerdos estas secuencias son más conservadas que en los demás hospederos estudiados y generalmente contienen aminoácidos monobásicos, una característica que pudo haber sido detectada por el modelo. En virus porcinos el sitio de reconocimiento del ácido Siálico tiene la particularidad de reconocer, los residuos SAα2,3 y SAα2,6, ambos presentes en cerdos, pero que se encuentran mayormente en aves y humanos, respectivamente (Galloway et al., 2013; Pulit-Penaloza et al., 2018).

5.7.Conclusiones

En conjunto, estos resultados no solo ponen de manifiesto la capacidad de los modelos de aprendizaje automático para capturar señales funcionalmente relevantes, sino que también permiten evaluar la contribución específica de ciertas regiones funcionales de la HA a la especificidad por un hospedero u otro del virus de Influenza tipo A. Este tipo de análisis contribuye a una mejor comprensión de los mecanismos moleculares de adaptación viral y puede orientar futuras investigaciones hacia puntos críticos para la vigilancia molecular e incluso ser un aporte al desarrollo de fármacos antivirales.

6. Capítulo 2.

Clasificación del subtipo de HA y patogenicidad

6.1. Resumen

Este capítulo aborda el desarrollo de modelos de aprendizaje automático para la clasificación del subtipo de hemaglutinina (HA) del virus de Influenza A y la identificación automatizada de variantes de alta y baja patogenicidad, con el objetivo de contribuir a sistemas de vigilancia genómica más ágiles y precisos. Dado que el subtipo de HA y la patogenicidad viral constituyen variables críticas para la evaluación del riesgo epidemiológico, se propone una metodología computacional que permite inferir esta información directamente a partir de secuencias completas o parciales de HA.

Se implementaron modelos supervisados basados en el descriptor de composición de dipéptidos (DPC), seleccionado a partir de análisis previos de reducción de dimensionalidad y clustering, que habían demostrado su capacidad para capturar patrones biológicamente relevantes. Se evaluaron distintos algoritmos de clasificación (KNN, SVM, Árboles de Decisión, Random Forest y Gradient Boosting), aplicando validación cruzada estratificada y búsqueda en grilla para la optimización de hiperparámetros. Los modelos fueron evaluados sobre conjuntos independientes que incluyeron tanto secuencias completas como secuencias parciales no contenidas en el entrenamiento.

Los resultados evidencian un desempeño sobresaliente en la clasificación del subtipo de HA, con métricas cercanas a 1 en secuencias completas para todos los modelos. En el escenario más exigente, correspondiente a secuencias parciales, el clasificador basado en Support Vector Machines (SVM) mostró la mayor robustez y estabilidad, alcanzando valores elevados de F1-macro y AUC-PR, y superando a Random Forest en términos de consistencia entre subtipos. Los principales errores de clasificación se observaron en subtipos minoritarios o filogenéticamente cercanos, como H2 y H5, sin comprometer significativamente el desempeño global.

Adicionalmente, se desarrolló un script automatizado para la detección de variantes de alta (HPAI) y baja patogenicidad (LPAI) en subtipos H5 y H7, basado en la identificación de motivos multibásicos en el sitio de clivaje HA1/HA2. Este procedimiento,

sustentado en información curada de OFFLU.org, permitió clasificar correctamente secuencias asociadas a la epizootia de influenza aviar altamente patogénica ocurrida en Uruguay en 2023, mostrando concordancia con análisis manuales y estudios previos.

Finalmente, los modelos de clasificación de hospedero, subtipo y el script de patogenicidad fueron integrados en una plataforma interactiva desarrollada en Streamlit, que permite procesar secuencias de HA y obtener resultados en tiempo real. En conjunto, este capítulo demuestra que la combinación de descriptores proteicos informativos y modelos de aprendizaje automático robustos permite clasificar con alta precisión el subtipo y la patogenicidad del virus Influenza A, incluso a partir de secuencias parciales, constituyendo una herramienta valiosa para la vigilancia epidemiológica y la toma temprana de decisiones sanitarias.

6.2.Introducción

Los virus de la influenza se clasifican a partir de la antigenicidad de sus glicoproteínas de superficie, en particular la hemaglutinina (HA) y la neuraminidasa (NA). En el VIA, se han descrito 18 subtipos de HA (H1–H18) y 11 de NA (N1–N11). Esta clasificación resulta útil para anticipar características relevantes del virus, como el hospedero, la patogenicidad y la capacidad de transmisión entre especies.

De manera tradicional, el subtipo se determina mediante análisis filogenéticos o Blast, un enfoque que suele demandar tiempo, experiencia técnica y recursos computacionales. Si bien existen herramientas bioinformáticas orientadas a esta tarea (por ejemplo, Nexclade y Fluserver), con frecuencia requieren alineamientos complejos o configuraciones técnicas avanzadas.

Dado el aumento del uso de la secuenciación genómica en la vigilancia epidemiológica, cobra especial importancia automatizar la interpretación de las secuencias. En este marco, la identificación rápida del subtipo de HA y de su patogenicidad (alta o baja) puede contribuir a optimizar la respuesta frente a brotes.

En este contexto, el aprendizaje automático se plantea como una estrategia eficaz para detectar patrones predictivos en grandes volúmenes de datos. En el capítulo anterior se observó que métodos no supervisados, como el clustering sobre proyecciones de PCA, reproducen adecuadamente la separación por subtipo, especialmente al utilizar los

descriptores DPC y Moran. En este capítulo se abordan modelos supervisados para clasificar subtipos de HA, empleando el descriptor DPC como variable predictiva principal.

6.3. Objetivos

6.3.1. Objetivo general

Implementar modelos de clasificación de subtipos de HA del virus de influenza tipo A, capaces de clasificar tanto secuencias completas como parciales. Integrados a sistemas de vigilancia genómica, tales modelos podrían contribuir a una evaluación más adecuada del riesgo asociado a la aparición de nuevos casos.

6.3.2. Objetivo específicos

- Implementar y evaluar una metodología basada en modelos de aprendizaje automático que permita la clasificación entre subtipos de HA del Virus de Influenza tipo A a partir de secuencias de HA, utilizando para ello al descriptor DPC como variable predictiva.

- Desarrollar un script de búsqueda de aminoácidos polibásicos específicos en las secuencias de HA. Estos se encuentran en el sitio de clivaje y según el motivo encontrado se pueden asociar a variantes de baja o de alta patogenicidad, denotando también el tipo o caldo, según especificaciones de Offlu (ver Anexo IV).

- Desarrollar una plataforma interactiva que integre los modelos seleccionados para clasificación de subtipo y hospedero con el sistema de asignación de patogenicidad

6.4. Metodología

El proceso comenzó con la base de Datos Integrada, conformada por secuencias completas y parciales de HA del VIA. Luego esta base de datos se subdivide en secuencias completas y parciales. En el clasificador de hospedero el data set 4 (secuencias completas

2020-2024) se utilizó para evaluar el comportamiento sobre este conjunto de datos. Para el desarrollo de clasificador de subtipo de HA este dataset formó parte del data set inicial (Data set 1 + 4) para compensar subrepresentación de clases (ver Figura 24). El set de secuencias parciales no contenidas (Sp_nc) se utilizó para evaluar la performance de los modelos.

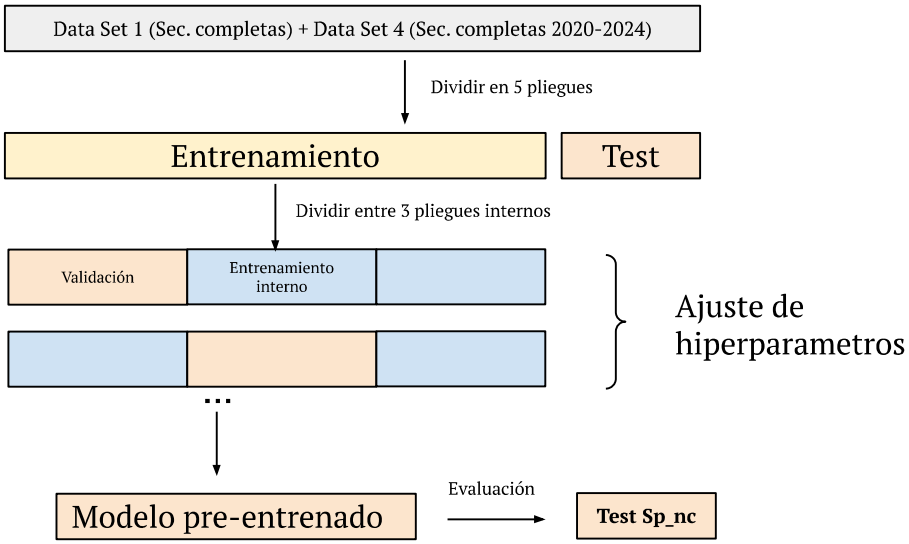


Figura 24. Conjuntos de datos utilizados para el entrenamiento y la validación cruzada anidada (k_externo = 5 y k_interno = 3). Los modelos generados durante el proceso de validación cruzada anidada (es decir, modelos preentrenados) se emplearon posteriormente para realizar predicciones sobre datos no vistos. La evaluación del desempeño se llevó a cabo sobre los conjuntos Test y Sp_nc.

6.4.1. Conjunto de entrenamiento y Test.

Inicialmente se aplicó la técnica Hold-Out sobre el data set inicial (1+4), dividiendo el conjunto de datos en entrenamiento (80%) y prueba (20%). Esta partición aseguró la disponibilidad de un conjunto independiente para la evaluación final (Test). La proporción de clases en el conjunto de entrenamiento y en el de evaluación se mantuvo mediante estratificación (*train_test_split*, scikit-learn) utilizando el parámetro *stratify=y*.

6.4.2. Entrenamiento y ajuste de hiperparámetros

En este trabajo evaluamos los siguientes algoritmos de aprendizaje supervisado: k-Vecinos más cercanos (KNN), Support Vector Machines (SVM), Árboles de decisión, Random Forest y Gradient Boosting Decision Trees (GBDT). Durante la etapa de entrenamiento, se implementó una validación cruzada de 3 pliegues (StratifiedKFold,

scikit-learn), que fue estratificada para asegurar una distribución de clases balanceada en cada partición (ver Figura 24).

Para cada algoritmo, se definió un conjunto de hiperparámetros a explorar (ver Tabla 2) y se aplicó la estrategia de búsqueda en grilla (GridSearchCV, scikit-learn). Cada combinación de hiperparámetros fue evaluada mediante validación cruzada 3-fold, y las métricas de evaluación fueron promediadas para seleccionar la combinación óptima de hiperparámetros. El criterio de optimización utilizado fue la métrica F1-macro, adecuada para escenarios con clases desbalanceadas. El mejor modelo por algoritmo se identificó a partir del valor máximo obtenido en dicha métrica (*best_estimator_*).

6.4.3. Selección de modelo y descriptor.

Cada algoritmo se volvió a entrenar sobre todo el set de Train con CV 3-fold y utilizando los hiperparámetros que dieron lugar a las mayores F1-macro, obteniendo estimaciones estables de las métricas (balanced accuracy, F1, MCC, AUC-PR) y un ajuste fino de los parámetros. Cada uno de los modelos obtenidos se evaluó sobre dos sets distintos con datos no vistos durante el entrenamiento; Test (secuencias completas) y Sp_nc (secuencias parciales no contenidas). La selección final del mejor modelo se realizó en base a los F1 alcanzados sobre los cuatro datasets, el análisis de las matrices de confusión, las curvas AUC-PR, el MCC, la Balanced Accuracy y la Precision macro.

6.4.4. Patogenicidad

En el VIA, los subtipos de HA que se clasifican según sus niveles de patogenicidad son principalmente H5 y H7. Ambos pueden clasificarse como de baja patogenicidad (LPAI, Low Pathogenic Avian Influenza) o alta patogenicidad (HPAI, Highly Pathogenic Avian Influenza), de acuerdo con la composición de aminoácidos en su sitio de clivaje. En particular, la presencia de múltiples aminoácidos básicos (arginina y lisina) en dicha región ha sido asociada con una mayor capacidad del virus para diseminarse sistémicamente en aves, y por lo tanto, con un fenotipo de alta patogenicidad.

Con el objetivo de automatizar la detección de estos motivos, se desarrolló un script en Python que permite identificar la presencia de motivos multibásicos de clivaje directamente a partir de secuencias aminoacídicas de HA. El script implementa una búsqueda por ventana deslizante en torno al dominio conservado “GLF”, y compara los

fragmentos obtenidos contra una base de datos curada de motivos conocidos de clivaje asociados a HPAI. La base de datos se construyó a partir de información obtenida en la plataforma OFFLU.org, respaldada por la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) y la Organización Mundial de Sanidad Animal (WOAH, ex OIE), ver Anexo IV. Esta fuente proporciona descripciones actualizadas de motivos característicos de clados específicos, linajes y variantes aisladas en distintos brotes.

6.5.Resultados

6.5.1. Modelos supervisados

Los resultados de la búsqueda en grilla de los hiperparámetros óptimos a partir del set de entrenamiento utilizando validación cruzada con tres pliegues posterior a Hold Out 80/20 (entrenamiento\test) se observan en la Tabla 10. Estos modelos fueron posteriormente evaluados en los conjuntos “Test” y “secuencias parciales no contenidas” (Sp_nc).

Tabla 10. Hiperparámetros seleccionados por búsqueda de grilla para cada uno de los modelos evaluados utilizando el descriptor DPC.

Modelo	Hiperparámetros seleccionados
KNN	n_neighbors = 3, weights = 'distance'
SVM	C = 0.1, kernel = 'linear', probability = True
RandomForest	max_depth = 10, n_estimators = 200
XGBoost	learning_rate = 0.1, max_depth = 5, n_estimators = 200, eval_metric = 'mlogloss'

Las métricas obtenidas a partir de la clasificación del subtipo de HA evidencian un excelente desempeño de todos los modelos evaluados, en particular cuando se utilizan secuencias completas. Se destacan Random Forest y SVM, con valores cercanos a 1 en todas las métricas (Tabla 11). Por otro lado, el modelo XGBoost presentó el menor

F1-score (0,975). En cuanto el desempeño sobre secuencias parciales no contenidas (Sp_nc) en el entrenamiento, se observa una disminución general de las métricas. En este escenario, SVM logra el mejor desempeño, con un F1-score de 0,983 y un AUC-PR de 0,992, superando notablemente a Random Forest (F1-score de 0,879 y AUC-PR de 0,975). Sin embargo, es importante destacar que el valor del MCC fue superior en Random Forest (0,993) en comparación con SVM (0,968), lo que sugiere un mejor equilibrio entre clases para este modelo, a pesar de la caída en precisión.

Tabla 11. Métricas de evaluación alcanzadas por los modelos durante la clasificación del subtipo de HA.

Modelo	Base	F1	MCC	AUC PR
KNN	Test	0,9995	0,9992	0,9999
	Sp_nc	0,9808	0,9958	0,9667
SVM	Test	0,9999	0,9992	1,0000
	Sp_nc	0,9833	0,9681	0,9917
Random Forest	Test	0,9999	0,9995	0,9999
	Sp_nc	0,8791	0,9932	0,9747
XGBoost	Test	0,9753	0,9982	1,0000
	Sp_nc	0,7948	0,9471	0,9214

En la Figura 25 se muestran algunos indicadores del comportamiento del modelo SVM sobre secuencias parciales no contenidas. Las curvas Precision–Recall se encuentran próximas al vértice superior derecho (1,1), lo que indica un excelente equilibrio entre precisión y sensibilidad. No obstante, se aprecia una caída significativa en la precisión al aumentar la sensibilidad para el subtipo H2, lo cual refleja la tendencia del modelo a clasificar erróneamente instancias de H2 como H1, como refleja la matriz de confusión ubicada a la derecha de la figura.

Por su parte, las secuencias del subtipo H5 también dan lugar a algunos errores, siendo clasificadas como H1, H2 y H3. Estos errores no afectan significativamente el desempeño general del modelo, dado que el la Precisión promedio (AP) se mantiene en 0,99, con aproximadamente el 90% de las instancias correctamente clasificadas. En contraste, el modelo Random Forest mostró una caída más abrupta en la precisión de los subtipos H2 y H16 a medida que se incrementa el recall, lo cual indica una mayor cantidad de falsos positivos. Sin embargo, se observa una ligera mejora en la clasificación del subtipo H1, que alcanza un AP de 1.00 con predicción prácticamente perfecta.

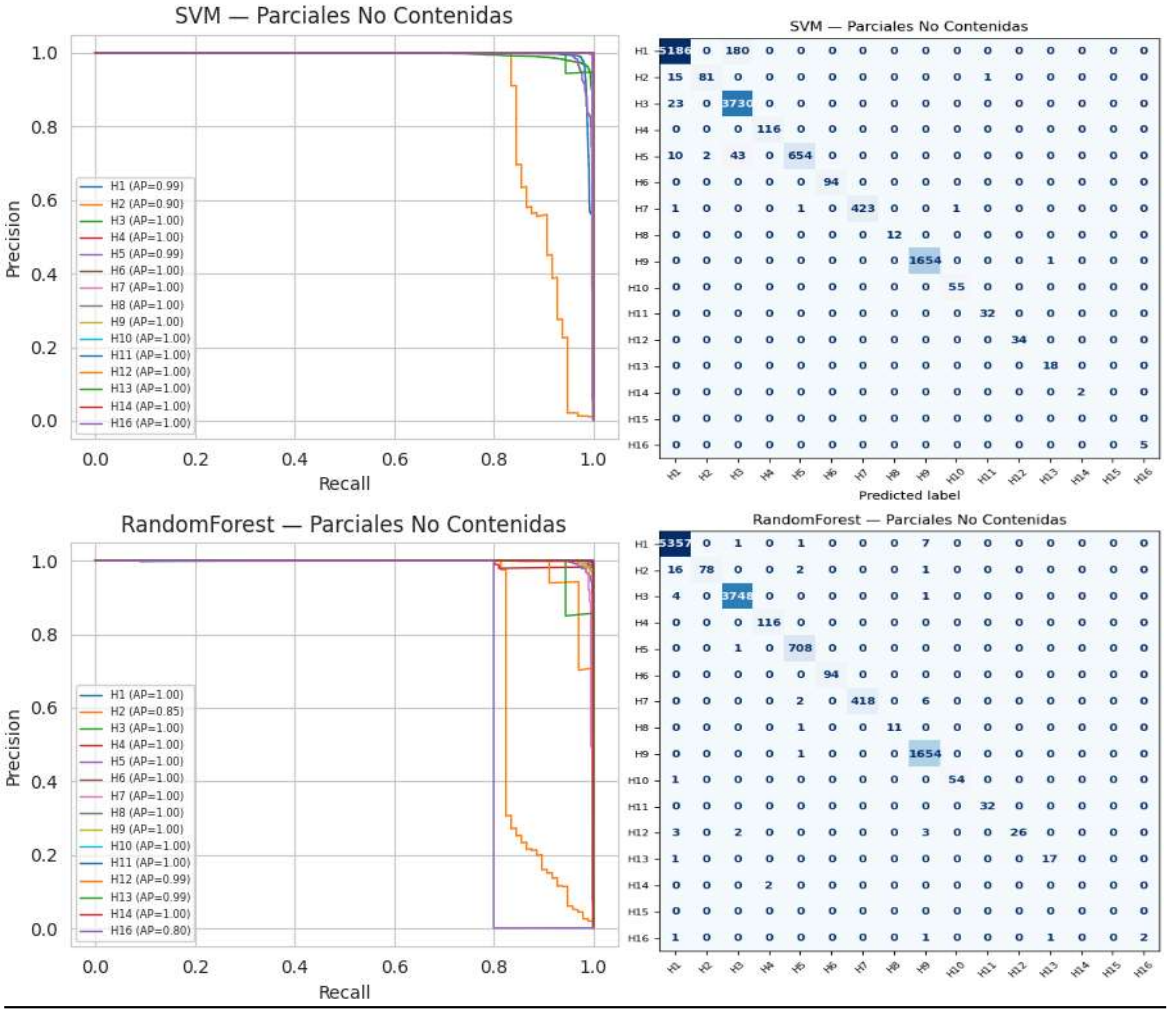


Figura 25. Curvas AUC-PR (izquierda) y las respectivas matrices de confusión (derecha) de los modelos SVM y Random Forest evaluados sobre el conjunto conformado por secuencias parciales no contenidas.

6.5.2. Detección de variantes de alta o baja patogenicidad

Se desarrolló el siguiente *script* que permite identificar los motivos que determinan si las variantes son de alta (HPAI) o baja (LPAI). Se utilizó información extraída de OFFLU.org, que incluye los sitios de clivaje correspondientes a H5 y H7. Inicialmente, se construyó un base de datos a partir de la información publicada en la página oficial de OFFLU.org, que recopila los motivos de sitios de clivaje descritos para subtipos de HA. Este script recorre la secuencia aminoacídica de la proteína HA query en busca del motivo GLF, que corresponde a la región inmediatamente posterior al sitio de clivaje en la proteína HA.

Bloque 1

```
Función detectar_sitio_clivaje(secuencia, ventana_max = 14):

{14 es la longitud máxima de un motivo}
   convertir secuencia a mayúsculas
   inicializar lista vacía encontrados

   for i desde (0: longitud(secuencia) - 2) :
      si secuencia[i : i+3] == "GLF" entonces:
         for tamaño desde (4 hasta ventana_max) hacer:
            inicio ← i - tamaño
            si inicio ⩾ 0 entonces:
               motivo ← secuencia[inicio : i]
               si motivo ∈ motivos_set entonces:
                  agregar a encontrados:
                     - motivo_detectado ← motivo
                     - Subtipo/clado← motivo

   retornar encontrados
```

Bloque 2

```
resultados ← detectar_sitio_clivaje(secuencia_ejemplo)

Si resultados no están vacíos:
   Para cada r en resultados:
      Buscar fila en 'motivos' donde Cleavage_Site == r['motivo_detectado']

      Si se encuentra:
         clado_tipo ← valor de columna 'Clado_Tipo'
         subtipo ← valor de columna 'Subtipo'
      Si no se encuentra:
         clado_tipo ← "Desconocido"
         subtipo ← "Desconocido"

      Imprimir:
         - Motivo detectado
         - Posición antes de GLF
         - Subtipo
         - Clado o tipo
```

Una vez identificado este motivo, se consideró una ventana de entre 4 y 14 aminoácidos anteriores a dicha posición que comparó la secuencia obtenida con todos los motivos registrados en el base de datos que contiene las referencias (Bloque 1). Cuando se detectó una coincidencia exacta, se procedió a recuperar la información asociada desde la base de datos estructurada, incluyendo subtipo o clado viral, patogenicidad (alta o baja) y el motivo de clivaje encontrado. Estos resultados se almacenan y posteriormente se muestran en pantalla, permitiendo identificar de forma automatizada la correspondencia entre la secuencia analizada y los motivos de referencia publicados por OFFLU (Bloque 2). Este script fue testeado con las secuencias de la epizootia de influenza aviar altamente patogénica ocurrida en 2023 y los resultados obtenidos se muestran en la tabla 12, donde se detectaron dos motivos diferentes entre las 32 secuencias de la epizootia utilizadas para probar el script. El segundo de la tabla es el que se encontró en las demás secuencias. Estos fueron coincidentes con los encontrados en la inspección manual de las secuencias (ver Anexo V).

Tabla 12. Clasificación de secuencias de HA utilizando los modelos seleccionados para clasificar hospedero y subtipo y por motivos de clivaje durante la epizootia de influenza A en Uruguay (2023). Se muestran los dos motivos detectados.

Accession	ID	Geolocalización	Hospedero	Subtipo	Resultado del Script de Patogenicidad
XBS26392	A	Uruguay	Ave	H5	Motivo: PLRERRRKR | Posición antes de GLF: 345 | Subtipo: H5 - HPAI | Clado/Tipo: Clade 2.3.4
WWQ73291	145M1	Uruguay	Ave	H5	Motivo: PLREKRRKR | Posición antes de GLF: 345 | Subtipo: H5 - HPAI | Clado/Tipo: Clade 2.3.4

Números de accesión restantes también analizados por el script: WWQ73303, WWQ73315, WPS93889, WPS93890, WPS93891, WPS93892, WPS93893, WPS93894, WPS93895, WPS93896, WPS93897, WPS93898, WPS93899, WPS93900, WPS93901, WPS93902, WPS93991, WPS94003, WLC96837, WLC96849, WLC96861, WLC96873, WLC96885, WLC96895, WLC96909, WLC96921, WLC96933, WLC96945, WLC96957, WLC96969

6.5.3. Plataforma interactiva

Se utilizó la plataforma *Streamlit* (Streamlit Inc., 2024) como *frontend* prototipado y como *backend* el repositorio de *GitHub* (GitHub Inc., 2024). El descriptor utilizado fue DPC, los modelos utilizados fueron KNN y SVM parametrizados y entrenados para clasificación de hospedero y subtipo de HA respectivamente. Los modelos en conjunto con el *script* de detección de motivos de patogenicidad fueron utilizados para generar el flujo de usuario representado en la figura 26.

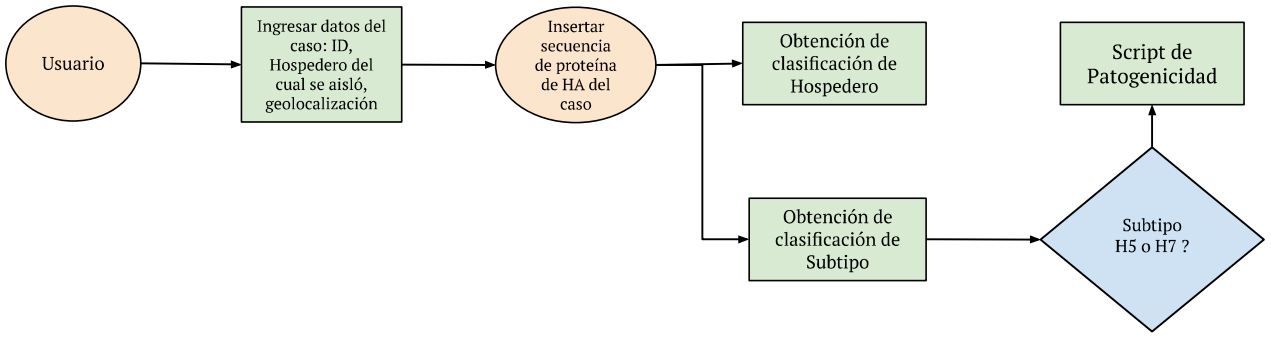


Figura 26. Diagrama de flujo detallado de navegación del usuario en la plataforma. El proceso comienza con el ingreso de los datos del caso (identificador, hospedero y geolocalización), seguido de la incorporación de la secuencia aminoacídica de la HA. El sistema ejecuta la clasificación para determinar el hospedero de origen y el subtipo de HA. Si el subtipo corresponde a H5 o H7, se aplica el script de detección de motivos de patogenicidad. Los resultados se visualizan en un mapa georreferenciado y pueden descargarse en formato CSV.

La plataforma interactiva se encuentra disponible en una version beta y se puede acceder a traves del siguiente enlace: https://aivepi-fcrzgqziiefk6wfvea9erp.streamlit.app/

6.6.Discusión

El hecho de que los modelos hayan presentado una performance muy buena en secuencias completas implica que la elección del descriptor fue buena, lo que a su vez reafirma los resultados de los análisis de reducción de dimensión y clustering. Si bien todos los modelos tienen un buen desempeño al clasificar secuencias completas, SVM aparece como el modelo más robusto frente a información incompleta, condición frecuente en entornos de vigilancia genómica. SVM muestra mayor estabilidad y robustez ante secuencias parciales (Sp_nc), manteniendo una alta precisión incluso cuando se esfuerza por recuperar más positivos (alto recall).

El modelo Random Forest, aunque globalmente muy bueno, tiene menor consistencia, es decir algunas clases muestran una caída significativa de precisión cuando aumenta el recall, lo cual puede ser problemático en tareas donde los falsos positivos deben mantenerse bajos (como vigilancia genómica). La Precision promedio (AP) funciona como un resumen numérico del área bajo cada curva, y en SVM hay mayor número de clases con AP=1.00, lo que indica mayor discriminación incluso en condiciones difíciles.

Trabajos recientes, como el propuesto por Humayun et al. (2021), han utilizado la generación de descriptores directamente a partir de la secuencia nucleotídica, incorporando no sólo ésta sino también propiedades fisicoquímicas mediante la combinación de K-gram, Discrete Wavelet Transformation (DWT) y Multivariate Mutual Information (MMI). Este enfoque presenta la ventaja de no requerir la traducción de la secuencia nucleotídica a aminoácidos, permitiendo su aplicación incluso en escenarios donde las anotaciones proteicas son incompletas o inexistentes. Sin embargo, los resultados reportados en dicho estudio, empleando árboles de decisión como clasificador, alcanzaron un valor de F1-score de 0,952, inferior al obtenido en el presente trabajo para la clasificación de secuencias parciales de HA (F1-score de 0,9833) utilizando el modelo óptimo SVM-DPC. Esta diferencia de rendimiento sugiere que, si bien los métodos basados en características extraídas directamente de la secuencia nucleotídica pueden ser útiles en contextos de datos limitados, el uso de descriptores derivados de la secuencia proteica —como la composición de dipéptidos— en conjunto con modelos de clasificación robustos como el Support Vector Machine, proporciona una mayor capacidad discriminativa. En particular, el presente estudio demuestra que esta estrategia mantiene un desempeño superior incluso cuando las secuencias analizadas son parciales, lo cual resulta relevante para su potencial aplicación en sistemas de vigilancia genómica del VIA.

El empleo del script para identificar la patogenicidad a través de las secuencias de HA de Uruguay tiene concordancia con los encontrado por otros autores (Marandino et. al, 2023), identificando correctamente la patogenicidad del subtipo. En este estudio se encontró el mismo motivo correspondiente a alta patogenicidad y se adjudicó al clado 2.3.4 a todos los virus de influenza de la epizootia de Uruguay. Cabe mencionar que la información proporcionada no es completa ya que, por ejemplo, estos motivos no pueden usarse para discernir entre subclados y falta asociar esta información con las demás proteínas virales. Otros estudios han hecho esto último, para entender mejor la epidemiología y la evolución del virus (Paz et al., 2024) y han asociado a este virus al subclado 2.3.4.4b. Sin embargo, la identificación de clados a través de los motivos sigue siendo más que suficiente para dar alerta sanitaria rápida y poder actuar de forma ágil y precisa, por lo cual entendemos se cumplen los objetivos planteados para este trabajo.

6.7.Conclusiones

Los modelso basados en SVM y Random Forest presentan un desempeño excelente, aunque SVM demuestra mayor estabilidad y discriminación entre subtipos cuando debe clasificar secuencias parciales. Por otro lado, a la luz de estos desempeños, se puede concluir que la elección de descriptores a través de algoritmos de reducción de dimensionalidad fue exitosa. Se logró desarrollar una plataforma amigable con el usuario para implementar estos modelos y así obtener los resultados de forma ágil.

7. Conclusiones finales

El presente trabajo abordó de forma sistemática una problemática actual de la vigilancia epidemiológica utilizando secuencias de proteínas virales y modelos de aprendizaje automático. Se generó un flujo de trabajo que comprende la identificación del hospedero origen del virus, el subtipo al cual corresponde e información de su patogenicidad y clado. Con los modelos implementados y el script desarrollado en el presente trabajo, esta información se puede obtener en tiempo real, proporcionando la agilidad y precisión necesarias para tomar las primeras medidas de contingencia ante brotes del virus de influenza tipo A. Se desarrolló una plataforma interactiva en Streamlit que permite ejecutar los modelos de clasificación entrenados en este trabajo y aplicar el script de detección de motivos asociados a patogenicidad en secuencias de hemaglutinina. Por otro lado, este trabajo llevó a identificar regiones de la HA asociadas fuertemente al tropismo del virus por los hospederos estudiados, lo cual constituye un aporte significativo en la comprensión de los mecanismos moleculares de adaptación interespecie. En particular, se observaron tres regiones altamente conservadas en secuencias de origen aviar, Heptad Repeat 2 (HR2), Epítopo Ca2 y dominio transmembrana. En las secuencias de origen humano, se destaca la presencia de la región correspondiente al Epítopo Sb, característico del dominio globular de HA1, involucrado en el reconocimiento por anticuerpos neutralizantes. Por otra parte, en las secuencias porcinas se identificó una alta conservación de motivos en el sitio de clivaje entre las subunidades HA1/HA2 y el sitio de reconocimiento del ácido siálico. Se destaca la importancia de identificar las regiones estructurales y funcionales que son tenidos en cuenta para la toma de decisiones de los modelos de ML cuando se utilizan secuencias biológicas.

8. Bibliografía

Aguas, R., and Ferguson, N. M. (2013). Feature selection methods for identifying genetic determinants of host species in RNA viruses. PLoS Computational Biology, 9(10), e1003254. https://doi.org/10.1371/journal.pcbi.1003254

Agüero, M., Monne, I., Sánchez, A., Zecchin, B., Fusaro, A., Ruano, M. J., ... and Orejas, J. J. (2023). Highly pathogenic avian influenza A (H5N1) virus infection in farmed minks, Spain, October 2022. Eurosurveillance, 28(3), 2300001.

Alexander, D. J. (2007). An overview of the epidemiology of avian influenza. Vaccine, 25(30), 5637-5644.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic acids research, 25(17), 3389-3402.

Allen, J. E., Gardner, S. N., Vitalis, E. A., & Slezak, T. R. (2009). Conserved amino acid markers from past influenza pandemic strains. *BMC Microbiology, 9*, Article 77. https://doi.org/10.1186/1471-2180-9-77

Attaluri, P. K., Zheng, X., Chen, Z., & Lu, G. (2009). Applying machine learning techniques to classify H1N1 viral strains occurring in 2009 flu pandemic. BIOT-2009, 21.

Attaluri, P. K., Chen, Z., & Lu, G. (2010, May). Applying neural networks to classify influenza virus antigenic types and hosts. In 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (pp. 1-6). IEEE.

Ao, C., Jiao, S., Wang, Y., Yu, L., & Zou, Q. (2022). Biological sequence classification: A review on data and general methods. Research, 2022, 0011.

Ao, C., Zou, Q., & Yu, L. (2022). RFhy-m2G: identification of RNA N2-methylguanosine modification sites based on random forest and hybrid features. Methods, 203, 32-39.

Bernett, J., Blumenthal, D. B., & List, M. (2024). *Cracking the black box of deep sequence-based protein–protein interaction prediction*. *Briefings in Bioinformatics, 25*(2), bbae076.

Basith, S., Lee, G., & Manavalan, B. (2022). STALLION: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction. Brief Bioinform., 23:bbab376. doi: 10.1093/bib/bbab376

BII . Flusurver - Prepared for the next wave. Accessed July 31, 2023

Bhasin, M., & Raghava, G. P. (2004). Classification of nuclear receptors based on amino acid composition and dipeptide composition. Journal of Biological Chemistry, 279(22), 23262-23266.

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. ACM computing surveys (CSUR), 49(2), 1-50.

Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.

Bonetta, R., & Valentino, G. (2020). Machine learning techniques for protein function prediction. Proteins: Structure, Function, and Bioinformatics, 88(3), 397-413.

Brockwell-Staats, C., Webster, R. G., & Webby, R. J. (2009). Diversity of influenza viruses in swine and the emergence of a novel human pandemic influenza A (H1N1). Influenza and other respiratory viruses, 3(5), 207-213.

Brown, I. H. (2001, October). The pig as an intermediate host for influenza A viruses between birds and humans. In International Congress Series (Vol. 1219, pp. 173-178). Elsevier.

Burke, D. F., & Smith, D. J. (2014). A recommended numbering scheme for influenza A HA subtypes. PloS one, 9(11), e112302.

Borkenhagen, L. K., Allen, M. W., & Runstadler, J. A. (2021). Influenza virus genotype to phenotype predictions through machine learning: a systematic review: computational prediction of influenza phenotype. Emerging microbes & infections, 10(1), 1896-1907.

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer

Caton, A. J., Brownlee, G. G., Yewdell, J. W., & Gerhard, W. (1982). The antigenic structure of the influenza virus A/PR/8/34 hemagglutinin (H1 subtype). Cell, 31(2), 417-427.

Cai, L., Ren, X., Fu, X., Peng, L., Gao, M., & Zeng, X. (2021). iEnhancer-XG: interpretable sequence-based enhancers and their strength predictor. Bioinformatics, 37(8), 1060-1067.

Charoenkwan, P., Chiangjong, W., Nantasenamat, C., Hasan, M. M., Manavalan, B., & Shoombuatong, W. (2021). StackIL6: a stacking ensemble model for improving the prediction of IL-6 inducing peptides. Brief Bioinform., 22:bbab172. doi: 10.1093/bib/bbab172

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. https://doi.org/10.1145/2939672.2939785

Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T. T., Wang, Y., ... & Song, J. (2018). iFeature: a python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics, 34(14), 2499-2502.

Chen, J., Zou, Q., & Li, J. (2022). DeepM6ASeq-EL: prediction of human N6-methyladenosine (m6a) sites with LSTM and ensemble learning. Frontiers of Computer Science, 16, 1-7.

Chou, K. C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins, 43, 246-255.

Chou, K. C. (2005). Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. Bioinformatics, 21, 10-19.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Mach Learn., 20, 273–97. doi: 10.1007/bf00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Trans Inf Theory, 13, 21–7. doi: 10.1109/TIT.1967.1053964

Cox, R. J., Brokstad, K. A., & Ogra, P. L. (2004). Influenza virus: immunity and vaccination strategies. Comparison of the immune response to inactivated and live, attenuated influenza vaccines. Scandinavian journal of immunology, 59(1), 1-15.

Cui, F., Zhang, Z., & Zou, Q. (2021). Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. Briefings in Functional Genomics, 20(1), 61-73.

de Graaf, M., & Fouchier, R. A. (2014). Role of receptor binding specificity in influenza A virus transmission and pathogenesis. The EMBO journal, 33(8), 823-841.

Dong, Q., Zhou, S., & Guan, J. (2009). A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. Bioinformatics, 25(20), 2655-2662.

Eng, C. L., Tong, J. C., & Tan, T. W. (2014). Predicting host tropism of influenza A virus proteins using random forest. *BMC Medical Genomics, 7*(Suppl 3), 1–11. https://doi.org/10.1186/1755-8794-7-S3-S1

ElHefnawi, M., & Sherif, F. F. (2014). Accurate classification and hemagglutinin amino acid signatures for influenza A virus host-origin association. *Virology, 449*, 328–338. https://doi.org/10.1016/j.virol.2013.11.017

Feng, Z. P., & Zhang, C. T. (2000). Prediction of membrane protein types based on the hydrophobic index of amino acids. J Protein Chem, 19, 269-275.

Freedman, D. A. (2005). Statistical models: theory and practice. New York: Cambridge University Press.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232.

Galloway, S. E., Reed, M. L., Russell, C. J., & Steinhauer, D. A. (2013). Influenza HA subtypes demonstrate divergent phenotypes for cleavage activation and pH of fusion: implications for host range and adaptation. PLoS pathogens, 9(2), e1003151.

Gorman, O. T., Bean, W. J., Kawaoka, Y., & Webster, R. G. (1990). Evolution of the nucleoprotein gene of influenza A virus. Journal of virology, 64(4), 1487-1497.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. Nature, 457(7232), 1012-1014.

Gilbertson, B., & Subbarao, K. (2023). Mammalian infections with highly pathogenic avian influenza viruses renew concerns of pandemic potential. Journal of Experimental Medicine, 220(8), e20230447.

GitHub Inc. (2024). *GitHub – Where the world builds software* [Plataforma de repositorios de código]. Recuperado de https://github.com/

Gozsari, (2024). *ProtFeat: protein feature extraction tool using POSSUM and iFeature* [Software]. GitHub. https://github.com/gozsari/ProtFeat.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press

Hasan, M. M., Tsukiyama, S., Cho, J. Y., Kurata, H., Alam, M. A., Liu, X., et al. (2022). Deepm5C: A deep-learning-based hybrid framework for identifying human RNA N5-methylcytosine sites using a stacking strategy. Mol Ther., 30, 2856–67. doi: 10.1016/j.ymthe.2022.05.001

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

He, W., Jia, C., & Zou, Q. (2019). 4mCPred: machine learning methods for DNA N4-methylcytosine sites prediction. Bioinformatics, 35(4), 593-601.

Hill, V., Ruis, C., Bajaj, S., Pybus, O. G., & Kraemer, M. U. (2021). Progress and challenges in virus genomic epidemiology. Trends in parasitology, 37(12), 1038-1049.

Humayun, F., Khan, F., Fawad, N., Shamas, S., Fazal, S., Khan, A., ... & Wei, D. Q. (2021). Computational method for classification of avian influenza A virus using DNA sequence information and physicochemical properties. Frontiers in Genetics, 12, 599321.

Horne, D. S. (1988). Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. Biopolymers, 27, 451-477.

Jeon, Y. J., Hasan, M. M., Park, H. W., Lee, K. W., & Manavalan, B. (2022). TACOS: a novel approach for accurate prediction of cell-specific long noncoding RNAs subcellular localization. Brief Bioinform., 23:bbac243. doi: 10.1093/bib/bbac243

Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Springer

Kawashima, S., et al. (2008). AAindex: amino acid index database, progress report 2008. Nucleic Acids Res, 36, D202-205.

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data, 6*(4), Article 15. https://doi.org/10.1145/2382577.2382579

Kennedy, M. (2005). Methodology in diagnostic virology. Veterinary Clinics: Exotic Animal Practice, 8(1), 7-26.

King, D., Miller, Z., Jones, W., et al. (2010). Characteristic sites in the internal proteins of avian and human influenza viruses. *Journal of Biomedical Science and Engineering, 3*(10), 943–955. https://doi.org/10.4236/jbise.2010.310123

Kobak, D., & Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. *Nature Communications, 10*, 5416

Koopmans, M. P. G., Barton Behravesh, C., Cunningham, A. A., Adisasmito, W. B., Almuhairi, S., Bilivogui, P., et al. (2024). The Panzootic Spread of Highly Pathogenic Avian Influenza H5N1 Sublineage 2.3.4.4b: A Critical Appraisal of One Health Preparedness and Prevention. Lancet Infect. Dis., 24, e774–e781.

Kargarfard, F., Sami, A., Mohammadi-Dehcheshmeh, M., & et al. (2016). Novel approach for identification of influenza virus host range and zoonotic transmissible sequences by determination of host-related associative positions in viral genome segments. *BMC Genomics, 17*, 925. https://doi.org/10.1186/s12864-016-3250-9

Kwon, E., Cho, M., Kim, H., & Son, H. S. (2020). A study on host tropism determinants of influenza virus using machine learning. Current Bioinformatics, 15(2), 121-134.

Lazniewski, M., Dawson, W. K., Szczepińska, T., & Plewczynski, D. (2018). The structural variability of the influenza A hemagglutinin receptor-binding site. Briefings in functional genomics, 17(6), 415-427.

Lambert, S. A., Jolma, A., Campitelli, L. F., Das, P. K., Yin, Y., Albu, M., et al. (2018). The human transcription factors. Cell, 172(4), 650-665.

Li, W., O'Neill, K. R., Haft, D. H., DiCuccio, M., Chetvernin, V., Badretdin, A., ... & Thibaud-Nissen, F. (2021). RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation. Nucleic acids research, 49(D1), D1020-D1028.

Liu, T., Zheng, X., & Wang, J. (2010). Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. Biochimie, 92(10), 1330-1334.

Liu, K., & Chen, W. (2020). iMRM: a platform for simultaneously identifying multiple kinds of RNA modifications. Bioinformatics, 36(11), 3336-3342.

Lin, S., Naim, H. Y., Chapin Rodriguez, A., & Roth, M. G. (1998). Mutations in the middle of the transmembrane domain reverse the polarity of transport of the influenza virus hemagglutinin in MDCK epithelial cells. The Journal of cell biology, 142(1), 51-57.

Lin, Z., & Pan, X. M. (2001). Accurate prediction of protein secondary structural content. J Protein Chem, 20, 217-220.

Lorieau, J. L., Louis, J. M., & Bax, A. (2010). The complete influenza hemagglutinin fusion domain adopts a tight helical hairpin arrangement at the lipid: water interface. Proceedings of the National Academy of Sciences, 107(25), 11341-11346.

Long, J. S., Mistry, B., Haslam, S. M., & Barclay, W. S. (2019). Host and viral determinants of influenza A virus species specificity. Nature Reviews Microbiology, 17(2), 67-81.

Lowen, A. C., Baker, A. L., Bowman, A. S., García-Sastre, A., Hensley, S. E., Lakdawala, S. S., et al. (2025). Pandemic Risk Stemming from the Bovine H5N1 Outbreak: An Account of the Knowns and Unknowns. J. Virol., 99, e00052-25.

Luczo, J. M., & Spackman, E. (2024). Epitopes in the HA and NA of H5 and H7 avian influenza viruses that are important for antigenic drift. FEMS Microbiology Reviews, 48(3), fuae014.

Lv, Z., Ao, C., & Zou, Q. (2019). Protein function prediction: from traditional classifier to deep learning. Proteomics, 19(14), 1900119.

Lyu, Z., Wang, Z., Luo, F., Shuai, J., & Huang, Y. (2021). Protein secondary structure prediction with a reductive deep learning method. Frontiers in Bioengineering and Biotechnology, 9, 687426.

Mock, F., Viehweger, A., Barth, E., & Marz, M. (2021). VIDHOP: Viral host prediction with deep learning. *Bioinformatics, 37*(3), 318–325.

Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists.* O'Reilly Media.

Nextclade . Accessed July 31, 2023. https://clades.nextstrain.org

NCBI Virus [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; [2024 04 18]. Available from: https://www.ncbi.nlm.nih.gov/labs/virus/.

Niu, M., Ju, Y., Lin, C., & Zou, Q. (2022). Characterizing viral circRNAs and their application in identifying circRNAs in viruses. Briefings in Bioinformatics, 23(1), bbab404.

Niu, M., Wu, J., Zou, Q., Liu, Z., & Xu, L. (2021). rBPDL: predicting RNA-binding proteins using deep learning. IEEE Journal of Biomedical and Health Informatics, 25(9), 3668-3676.

Nuwarda, R. F., Alharbi, A. A., & Kayser, V. (2021). An Overview of Influenza Viruses and Vaccines. Vaccines 2021, 9, 1032.

Oguzie, J. U., Marushchak, L. V., Shittu, I., Lednicky, J. A., Miller, A. L., Hao, H., et al. (2024). Avian Influenza A(H5N1) Virus among Dairy Cattle, Texas, USA. Emerg. Infect. Dis., 30, 1425–1429.

Pardo-Roa, C., Nelson, M. I., Ariyama, N., Aguayo, C., Almonacid, L. I., Munoz, G., ... & Neira, V. (2023). Cross-species transmission and PB2 mammalian adaptations of highly pathogenic avian influenza A/H5N1 viruses in Chile. bioRxiv.

Padhi, A., Agarwal, A., Saxena, S. K., & Katoch, C. D. S. (2023). Transforming clinical virology with AI, machine learning and deep learning: a comprehensive review and outlook. VirusDisease, 34(3), 345-355.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Pulit-Penaloza, J. A., Pappas, C., Belser, J. A., Sun, X., Brock, N., Zeng, H., et al. (2018). Comparative in vitro and in vivo analysis of H1N1 and H1N2 variant influenza viruses isolated from humans between 2011 and 2016. Journal of virology, 92(22), 10-1128.

Puryear, W., Sawatzki, K., Hill, N., Foss, A., Stone, J. J., Doughty, L., ... & Runstadler, J. (2023). Highly pathogenic avian influenza A (H5N1) virus outbreak in New England seals, United States. Emerging infectious diseases, 29(4), 786.

Rajao, D. S., Vincent, A. L., & Perez, D. R. (2019). Adaptation of human influenza viruses to swine. Frontiers in veterinary Science, 5, 347.

Saravanan, V., & Gautham, N. (2015). Harnessing Computational Biology for Exact Linear B-Cell Epitope Prediction: A Novel Amino Acid Composition-Based Feature Descriptor. OMICS, 19, 648-658.

Shao, W., Li, X., Goraya, M. U., Wang, S., & Chen, J.-L. (2017). Evolution of influenza A virus by mutation and re-assortment. *International Journal of Molecular Sciences, 18*(8), 1650. https://doi.org/10.3390/ijms18081650

Shaltout, N., Moustafa, M., Rafea, A., Moustafa, A., & ElHefnawi, M. (2015, November). Comparing PCA to information gain as a feature selection method for Influenza-A

classification. In 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS) (pp. 279-283). IEEE.

Shaltout, N., Rafea, A., Moustafa, A., Moustafa, M., & ElHefnawi, M. (2016). Optimizing the detection of antiviral-resistant influenza-A strains using machine learning. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 2)..

Sherif, F. F., El Hefnawi, M., & Kadah, Y. (2011). Genomic signatures and associative classification of the hemagglutinin protein for human versus avian versus swine influenza A viruses. In *2011 28th National Radio Science Conference (NRSC)* (pp. 1–9). IEEE. https://doi.org/10.1109/NRSC.2011.5976802

Sokal, R. R., & Thomson, B. A. (2006). Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population. Am J Phys Anthropol, 129, 121-131.

Steel, J., Lowen, A. C., Mubareka, S., & Palese, P. (2009). *Transmission of influenza virus in a mammalian host is increased by PB2 amino acids 627K or 627E/701N. PLoS Pathogens, 5*(1), e1000252.

Stray, S. J., & Pittman, L. B. (2012). Subtype-and antigenic site-specific differences in biophysical influences on evolution of influenza virus hemagglutinin. Virology journal, 9(1), 91.

Streamlit Inc. (2024). *Streamlit – The fastest way to build and share data apps* [Framework de Python]. Recuperado de https://streamlit.io/

scikit-learn. (2025a). *KNeighborsClassifier*. Documentación oficial de scikit-learn. Recuperado el 31 de agosto de 2025 de https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

scikit-learn. (2025b). *SVC*. Documentación oficial de scikit-learn. Recuperado el 31 de agosto de 2025 de https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

scikit-learn. (2025c). *DecisionTreeClassifier*. Documentación oficial de scikit-learn. Recuperado el 31 de agosto de 2025 de

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

scikit-learn. (2025d). *RandomForestClassifier*. Documentación oficial de scikit-learn. Recuperado el 31 de agosto de 2025 de https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Sriwilaijaroen, N., & Suzuki, Y. (2012). Molecular basis of the structure and function of H1 hemagglutinin of influenza virus. Proceedings of the Japan Academy, Series B, 88(6), 226-249.

Suguitan, A. L., Jr., Matsuoka, Y., Lau, Y.-F., Santos, C. P., Vogel, L., Cheng, L. I., Orandle, M., & Subbarao, K. (2012). *The multibasic cleavage site of the hemagglutinin of highly pathogenic A/Vietnam/1203/2004 (H5N1) avian influenza virus acts as a virulence factor in a host-specific manner in mammals*. *Journal of Virology, 86*(5), 2706–2714.

Skehel, J. J., & Wiley, D. C. (2000). Receptor binding and membrane fusion in virus entry: The influenza hemagglutinin. *Annual Review of Biochemistry, 69*, 531–569.

Taubenberger, J. K., & Kash, J. C. (2010). Influenza virus evolution, host adaptation, and pandemic formation. Cell host & microbe, 7(6), 440-451.

Vaidyanathan, A., Gates, A., Brown, C., Prezzato, E., & Bernstein, A. (2024). Heat-Related Emergency Department Visits—United States, May–September 2023. MMWR Morb. Mortal. Wkly. Rep., 73, 324–329.

Wang, J., Yang, B., Revote, J., Leier, A., Marquez-Lago, T. T., Webb, G., ... & Lithgow, T. (2017). POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. Bioinformatics, 33(17), 2756-2758.

World Health Organization. (2022, December 21). *Assessment of risk associated with recent influenza A(H5N1) clade 2.3.4.4b viruses* [Emergency situation update]. World Health Organization

Wu, N. C., & Wilson, I. A. (2020). Influenza hemagglutinin structures and antibody recognition. *Cold Spring Harbor Perspectives in Medicine, 10*(8), a038778. https://doi.org/10.1101/cshperspect.a038778

XGBoost Developers. (2025). *XGBoost Python API*. Documentación oficial de XGBoost. https://xgboost.readthedocs.io/en/stable/python/python_api.html

Xiao, N., et al. (2015). protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. Bioinformatics, 31, 1857-1859.

Xu, C., Zhang, N., Yang, Y., Liang, W., Zhang, Y., Wang, J., et al. (2022). Immune escape adaptive mutations in hemagglutinin are responsible for the antigenic drift of Eurasian avian-like H1N1 swine influenza viruses. Journal of Virology, 96(16), e00971-22.

Xu, M., Zhang, Z., Lu, J., Zhu, Z., Zhang, Y., Ma, C., Liu, R., & Tang, J. (2022). *PEER: A comprehensive and multi-task benchmark for protein sequence understanding* (arXiv:2206.02096). arXiv. https://doi.org/10.48550/arXiv.2206.02096

Xu, X., Subbarao, K., Cox, N. J., & Guo, Y. (1999). Genetic characterization of the pathogenic influenza A/Goose/Guangdong/1/96 (H5N1) virus: similarity of its hemagglutinin gene to those of H5N1 viruses from the 1997 outbreaks in Hong Kong. Virology, 261(1), 15-19.

Xu, B., Tan, Z., Li, K., Jiang, T., & Peng, Y. (2017). Predicting the host of influenza viruses based on the word vector. PeerJ, 5, e3579.

Xue, J., Lee, C., Wakeham, S. G., & Armstrong, R. A. (2011). Using principal components analysis (PCA) with cluster analysis to study the organic geochemistry of sinking particles in the ocean. Organic Geochemistry, 42(4), 356-367.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605.

Yang, Y., Gao, D., Xie, X., Qin, J., Li, J., Lin, H., et al. (2022). DeepIDC: A prediction framework of injectable drug combination based on heterogeneous information and deep learning. Clin Pharmacokinet., 61, 1749–59. doi: 10.1007/s40262-022-01180-9

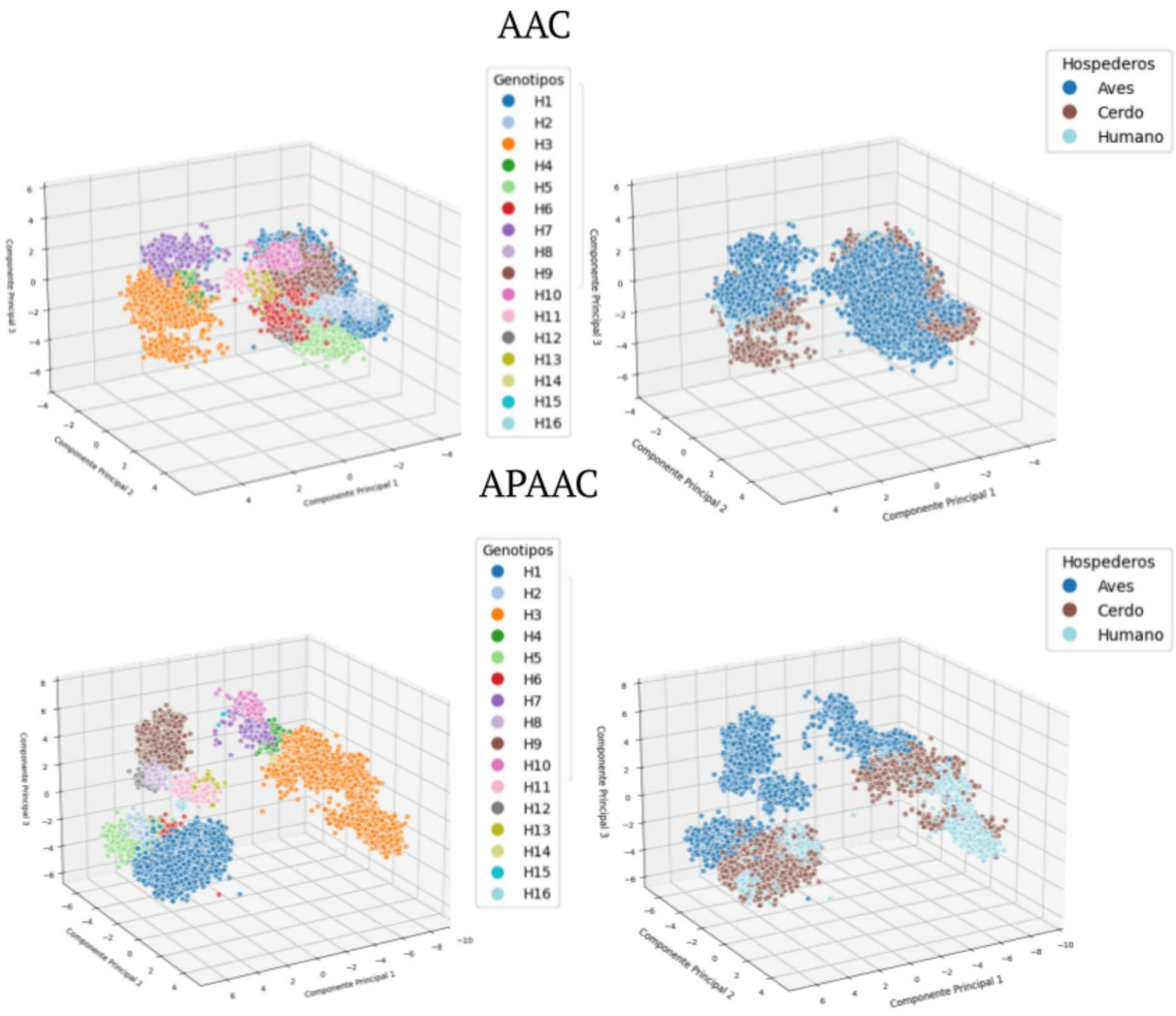
Yin, R., Zhou, X., Zheng, J., & et al. (2018). Computational identification of physicochemical signatures for host tropism of influenza A virus. *Journal of Bioinformatics and Computational Biology, 16*(6), 1–21. https://doi.org/10.1142/S0219720018500264

Yuan, S. S., Gao, D., Xie, X. Q., Ma, C. Y., Su, W., Zhang, Z. Y., et al. (2022). IBPred: A sequence-based predictor for identifying ion binding protein in phage. Comput Struct Biotechnol J., 20, 4942–51. doi: 10.1016/j.csbj.2022.08.053

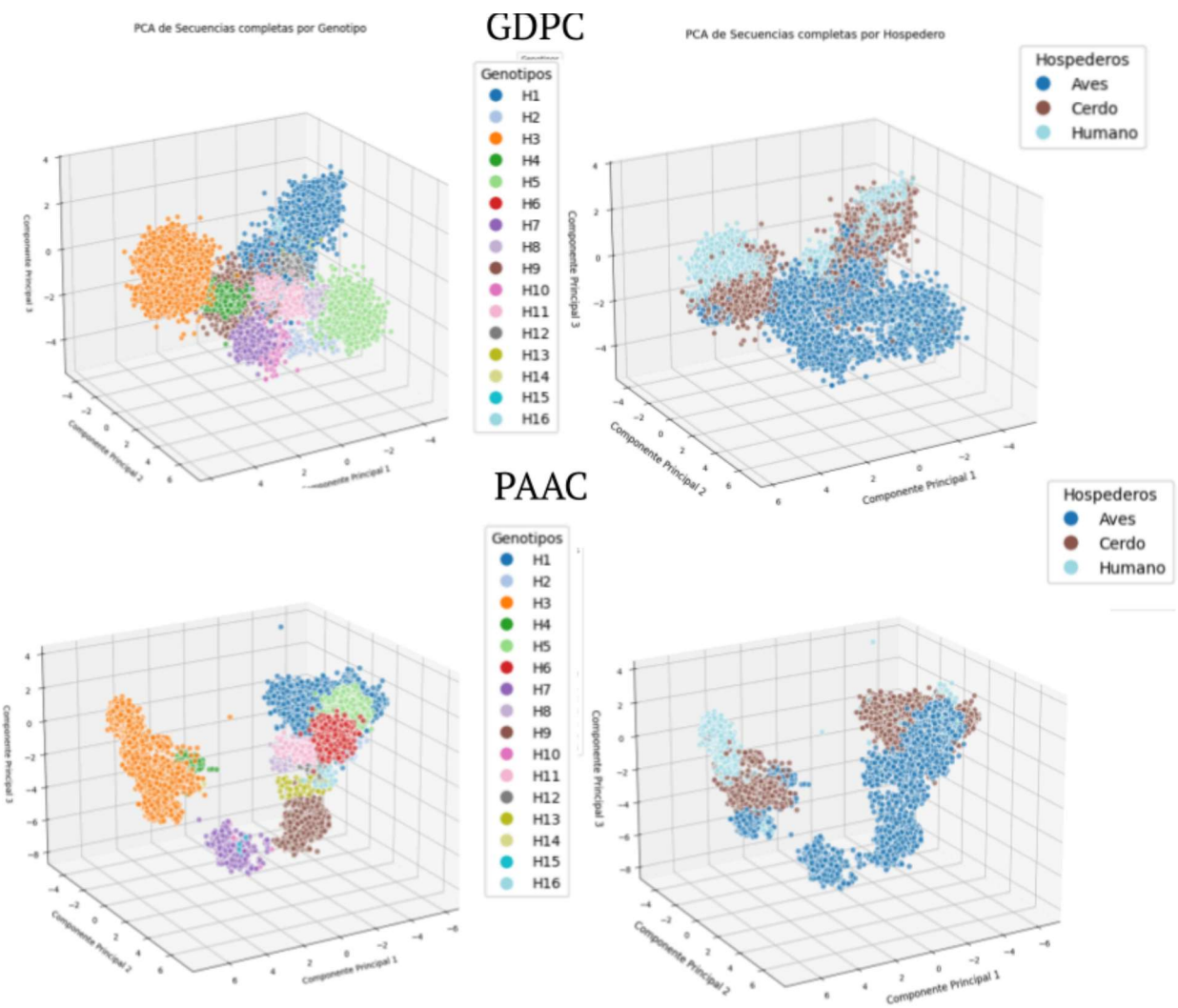
Zhang, Z. Y., Ning, L., Ye, X., Yang, Y. H., Futamura, Y., Sakurai, T., et al. (2022). iLoc-miRNA: extracellular/intracellular miRNA prediction using deep BiLSTM with attention mechanism. Brief Bioinform., 23:bbac395. doi: 10.1093/bib/bbac395

9. Anexos

9.1. Anexo I

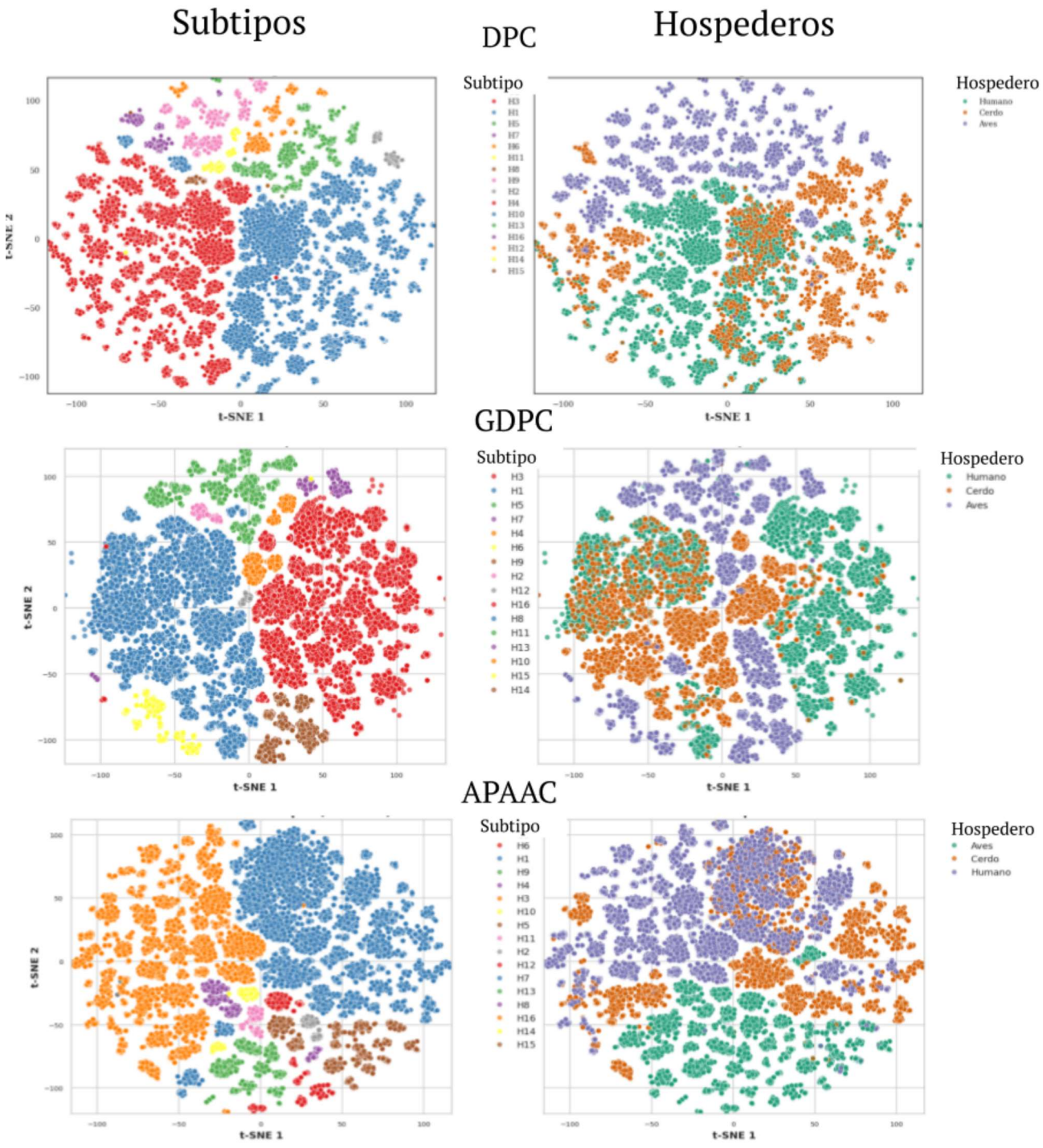


Proyecciones de los tres componentes principales realizados sobre los descriptores AAC y APAAC. En la primer columna se colorean los subtipos de HA y en la segunda columna los hospederos, ambas proceden del mismo análisis.

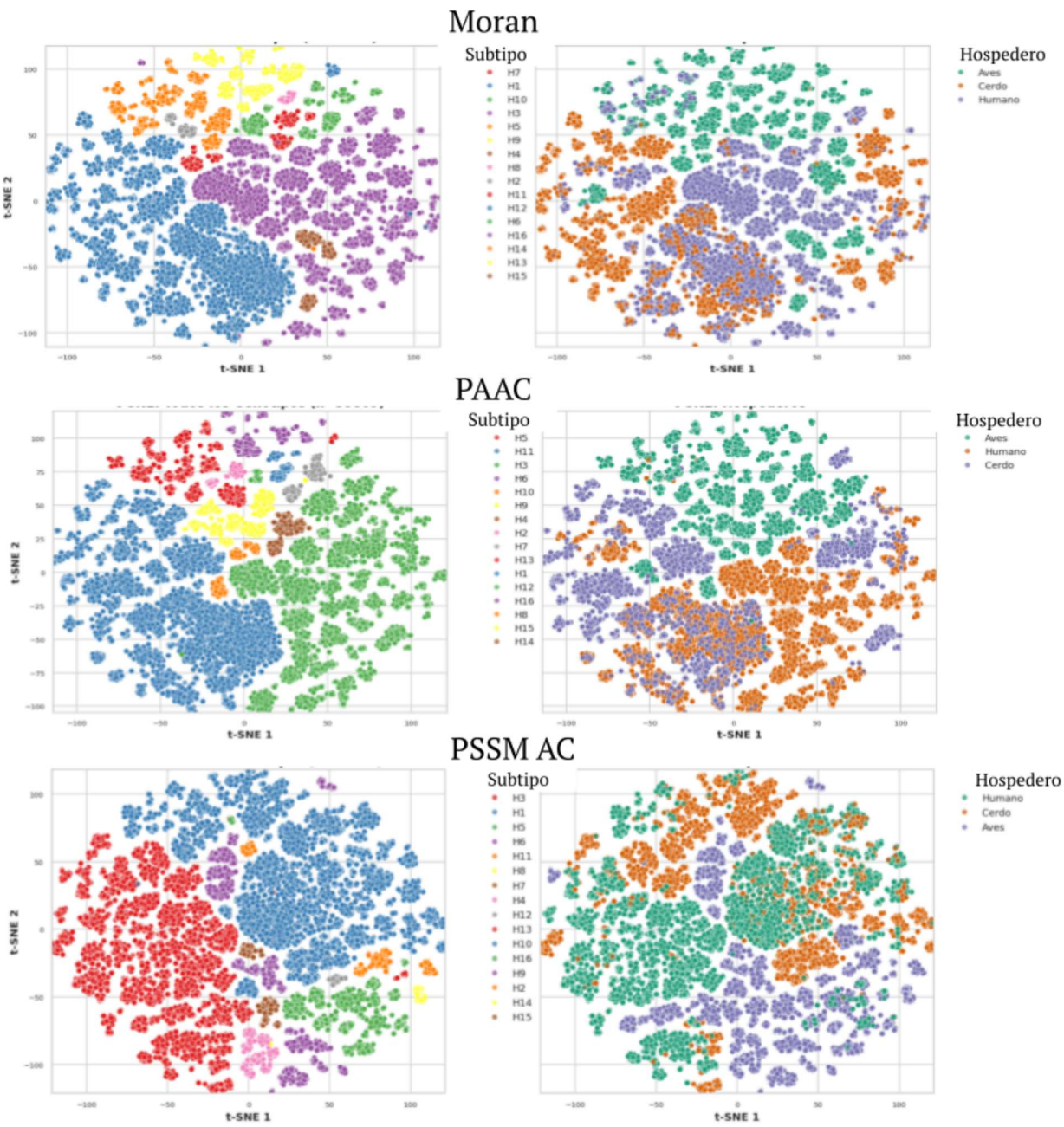


Proyecciones de los tres componentes principales realizados sobre los descriptores GDPC, PAAC.. En la pr columna se colorean los subtipos de HA y en la segunda columna los hospederos, ambas proceden del m análisis.

9.2. Anexo II

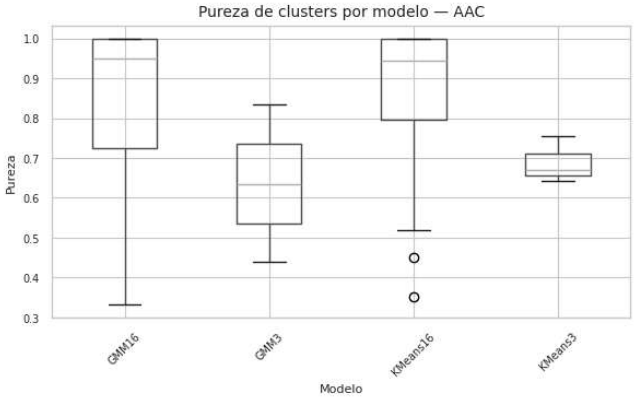


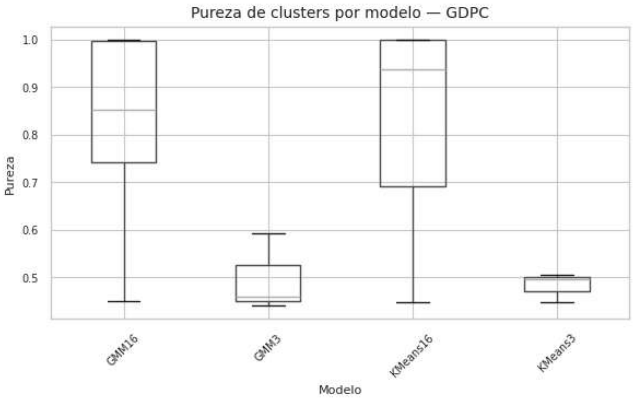
Proyecciones de los componentes del análisis t-SNE para el descriptor DPC, GDPC, APAAC sobre el Data Set 1. A la izquierda se colorean los subtipos y a la derecha los hospederos.

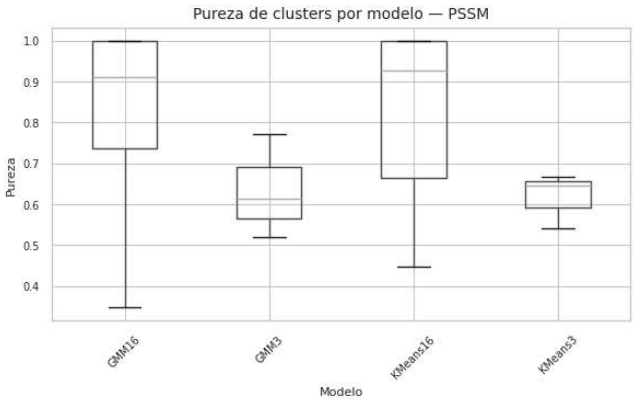


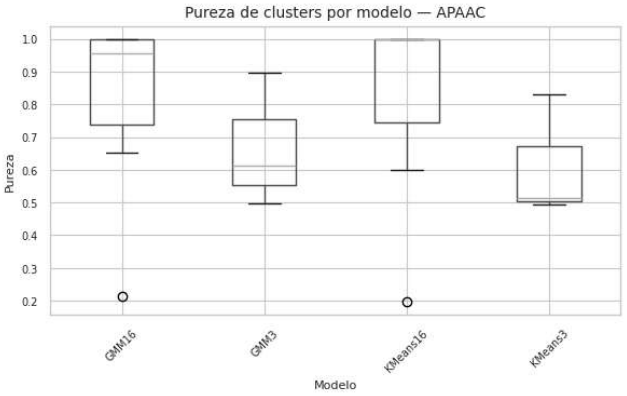
Proyecciones de los componentes del análisis t-SNE para el descriptor Moran, PAAC, PSSM AC sobre el Data Set 1. A la izquierda se colorean los subtipos y a la derecha los hospederos.

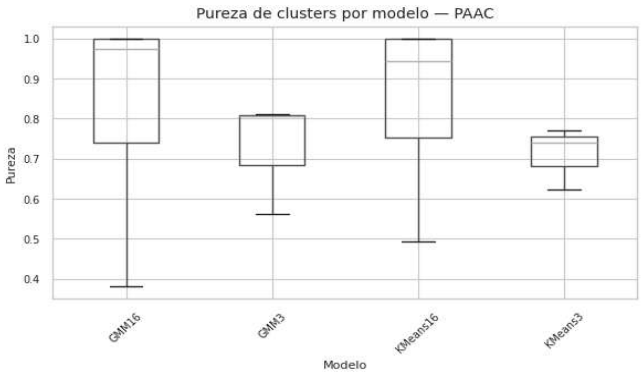
9.3. Anexo III



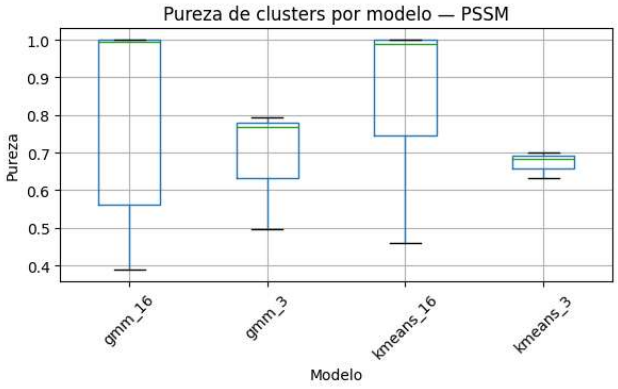


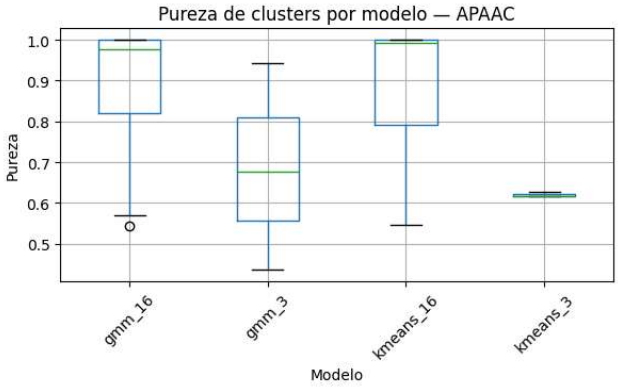


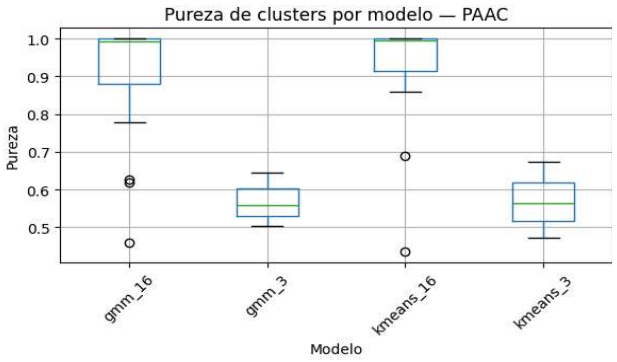


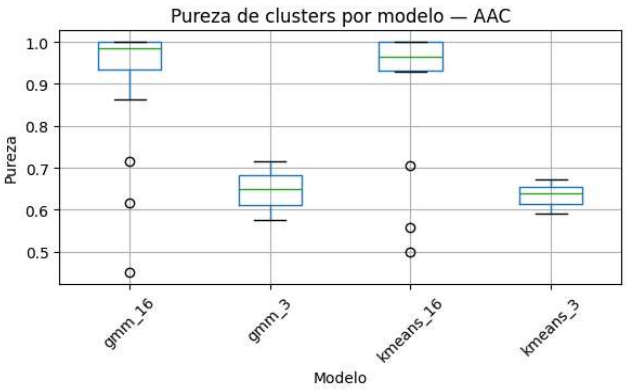


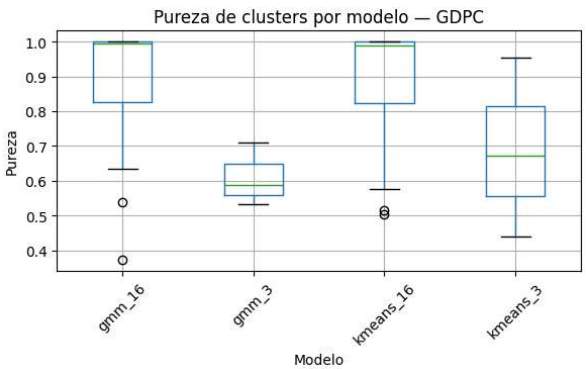
Resultados de la pureza de clusters en configuracion 16 y 3 para los descriptores AAC, GDPC,PSSM,APAAC,PAAC utilizando los algoritmos de culstering GMM y KMeans sobre las proyecciones de T-SNE.











Resultados de la pureza de clusters en configuracion 16 y 3 para los descriptores AAC, GDPC,PSSM,APAAC,PAAC utilizando los algoritmos de culstering GMM y KMeans sobre las proyecciones de T-SNE.

9.4.Anexo IV

Sitios de clivaje polibásicos reportados por OFFLU

Fuente: *OFFLU (FAO/OIE, 2024). "Influenza A Cleavage site update 2022 (4 January)." Disponible en: https://offlu.org/technical-activities/influenza-a-cleavage-site-update-2022-4-january/*

Reproducido íntegramente con fines académicos

Sub-type	Clade[1]/type virus	Cleavage site consensus[2]	critical basic aa[3]	Size of insert
H5	LP	**PQRETR/GLF**	1	0
H5N1	Gs/Gd-lineage	**PQRERRRKKR/GLF**	6	4
H5N1	Clade 1	**PQREERRKKR/GLF** **PQREGRRKKR/GLF** **PQRVGRRKKR/GLF**	5	4
H5N1	Clade 2.1	**PQRESRRKK/GLF** **PQKEGRRKKR/GLF** **PQIERRRKKR/GLF** **PQRERRREKR/GLF**	4-6	3-4
H5N1	Clade 2.2	**PQGEKRRKKR/GLF** **PQGERRRKKR/GLF** **PQGEGRRKKR/GLF** **PQGDRRRKKR/GLF**	5-6	4
H5N1	Clade 2.3.1	**PQRERRRKR/GLF**	5	3

H5N1	Clade 2.3.2	PQRERRRKR/GLF PRRERRRKR/GLF PQREKRRKR/GLF PQKERRRKR/GLF PQIERRRRKR/GLF PQRESRRRKR/GLF	5-6	3-4
H5N1	Clade 2.3.3	PQRERRRKR/GLF	5	3
H5N1, H5N2, H5N3, H5N4, H5N5, H5N6, H5N8	Clade 2.3.4	PLRERRRKR/GLF PLREKRRKR/GLF PPREKRRKR/GLF PLREKRRRKR/GLF PLRERRRKKR/GLF PLRERIRKKR/GLF PLGEKRRKR/GLF PLIEKRRKR/GLF PLRDKRRKR/GLF[4] PLRGKRRKR/GLF[5]	4-6	3-4
H5N1	Clade 2-like	PQRERRRKKR/GLF PQRERRKKR/GLF	5-6	3-4
H5N1	Clade 3	PQRERRRKKR/GLF	6	4
H5N1	Clade 4	PQRERRRKKR/GLF	6	4
H5N1	Clade 5	PQREIRRKKR/GLF	5	4
H5N1	Clade 6	PQRERRRKKR/GLF	6	4
H5N1	Clade 7	PQIEGRRRKR/GLF PQRGRRRKR/GLF PQRGRRRKKR/GLF PQREGGRKKR/GLF PQREGGRRRKR/GLF PQREREGGRRRKR/GLF	4-5	3-4
H5N1	Clade 9	PQRERRRKKR/GLF	6	4
H5N2	A/ostrich/SA/AI2114/11[18] A/ostrich/SA/AI2887/11	PQRRKKR/GLF PQRRRKR/GLF	4 4	1 1
H5N2	A/ostrich/SA/AI1091/06[1, 16]	PQREKRRKKR/GLF	6	4
H5N1	A/gull/Germany/R882/06[29]	PQGERRRKKR/GLF	6	4
H5N2	A/chicken/Italy/1485/97[6, 30]	PQRRRKKR/GLF	5	2
H5N2	A/chicken/Puebla/8623-607/94[4-6]	PQRRKRKTR/GLF PQRKRKRKTR/GLF	5 6	3 4
H5N2	A/chicken/Puebla/8624-602/94[5]	PQRKRKTR/GLF	4	2
H5N1	A/turkey/England/50-92/91[10]	PQRKRRKTR/GLF	5	3
H5N8	A/turkey/Ireland/1378/83[6, 31]	PQRKRKKR/GLF	5	2
H5N9	A/turkey/Ontario/7732/66[6, 32]	PQRRRKKR/GLF	5	2
H5N3	A/tern/South Africa/61[6, 10]	PQRETRRQKR/GLF	4	4

[1] LP – low pathogenic. Gs/Gd = A/goose/Guangdong/1/1996-lineage highly pathogenic avian influenza H5N1; clades include all higher order subclades (i.e clade 7 includes clade 7, 7-like, 7.1, 7.2) unless otherwise specified; numbers in parentheses are references.

[2] Consensus sequence generated from H5 HA sequences available in public databases; red color indicates critical basic residues; / indicates cleavage position; residue insertions are underlined (38).

[3] Basic residue at the -1 position and basic amino acids immediately preceding this position; excludes the -4 position found in the LPAI cleavage site consensus motif.

[4] APHA-Weybridge

[5] Siencsano-Belgium

Table 2: Multi-basic cleavage sites of sporadic H7 HPAI avian influenza A viruses

Sub-type	Pheno-type[1]	location	year	Type virus	cleavage site[2]	Critical basic aa[3]	Size of insert	Accession number[4]	Ref.
H7	LP				**PEIPKGR/GLF,** **PELPKGK/GLF,** **PEPPKGR/GLF,** **PENPKTR/GLF,** **PESPKTR/GLF**	1	0		38
H7N7	HP	Australia	2020	A/chicken/Victoria/20-02865-8/2020	**PEIPGKRERKR/GLF**	4	4	Pending	CSIRO
H7N3	HP	USA	2020	A/turkey/South_Carolina/20-010561-003/2020	**PENPKTDRKSRHRRIR/GLF**[5]	6	9	MT444387	50
H7N9,	HP	China	2017	A/chicken/Guangdong/GD4/2017	**PEVPKGKRTAR/GLF**	3	4	KY855518	43-46
				A/chicken/Guangdong/GD15/2016	**PEVPKRKRTAR/GLF**	4	4	EPI960361	
				A/Guangdong/Th005/2017	**PEVPKGKRIAR/GLF**[5]	3	4	EPI926825	
H7N3	HP	Japan[6]	2018	A/duck/Japan/AQ-HE30-1/2018	**PEVPKRRRTAR/GLF**	4	4	LC416566	47
H7N9	HP	USA	2017	A/chicken/Tennessee/17-007147-2/2017	**PENPKTDRKSRHRRIR/GLF**[5]	6	9	KY818811	42
H7N7	HP	Italy	2016	A/chicken/Italy/16VIR-1873/2016	**PELPKGRKRR/GLF**	4	3	EPI220955	51
H7N8	HP	USA	2016	A/turkey/Indiana/16-001403-1/2016	**PENPKKRKTR/GLF**	4	3	KU558906.1	41
H7N7	HP	UK	2015	A/chicken/England/26352/2015	**PEIPRHRKGR/GLF**	4	3	EPI623939	APHA
H7N7	HP	Germany	2015	A/chicken/Germany/AR1386/2015	**PEIPKRKRR/GLF**	5	3	EPI634885	52
H7N7	HP	Italy	2013	A/chicken/Italy/13VIR4527_11/13	**PETPKRRERR/GLF**	4	3	KF569186.1	39
H7N3	HP	Mexico	2012-2018	A/chicken/Jalisco/CPA1/12	**PENPKDRKSRHRRTR/GLF**	6	8	JX397993.1	23
				A/chicken/Puebla/CPA-04451/16	**PENPKDRKNRHRRTR/GLF**	6	8	KX351916.1	
				A/chicken/Jalisco/CPA-01859/16	**PENPKGKKSRHRKTR/GLF**	6	8	KX351892.1	
H7N7	HP	Spain	2009	A/chicken/Spain/6279-2/2009	**PELPKGTKPRPRR/GLF**	4	6	GU121458.1	24
H7N7	HP	UK	2008	A/chicken/England/1158-11406/08	**PEIPKRKKR/GLF**	4	2	FJ476173.1	25,26
H7N3	HP	Canada	2007	A/chicken/Saskatchewan/HR-00011/07	**PENPKTTKPRPRR/GLF**	4	6	EU500860.1	17
H7N7	HP	North Korea	2005	A/chicken/North Korea/1/2005	**PEIPKGRHRRPKR/GLF**	5	6		13
H7N3	HP	Canada	2004	A/chicken/Canada/rv504/04	**PENPKQAYRKRMTR/GLF**	4	7	CY015006.1	13
					PENPKQAYQKRMTR/GLF	3	7		
					PENPKQAYKKRMTR/GLF	4	7		
					PENPKQAYHKRMTR/GLF	3	7		
					PENPKQAHQKRMTR/GLF	3	7		
					PENPRQAYRKRMTR/GLF	4	7		
					PENPKQACQKRMTR/GLF	3	7		
H7N7	HP	Netherlands	2003	A/chicken/Netherlands/219/03	**PEIPKRRRR/GLF**	4	2	AY338459.1	9,27
H7N3	HP	Chile	2002	A/chicken/Chile/4322/02	**PEKPKTCSPLSRCRETR/GLF**	3	10	AY303631.1	7,28
					PEKPKTCSPLSRCRKTR/GLF	4	10		
H7N1	HP	Italy	1999	A/chicken/Italy/444/99	**PEIPKGSRVRR/GLF**	3	4	AJ704810.1	12
					PEIPKGSRMRR/GLF	3	4		
					PEIPKRSRVRR/GLF	4	4		
H7N4	HP	Australia	1997	A/chicken/New South Wales/2/97	**PEIPRRRKR/GLF**	4	2	CY022693-	19,20
					PEIPRKRKR/GLF	4	2	CY022700	
H7N3	HP	Pakistan	1995	A/chicken/Pakistan/447/95	**PETPKRKRKR/GLF**	5	3	AF202226	2
				A/chicken/Pakistan/CR2/95	**PETPKRRKR/GLF**	4	2	AF202230	
				A/chicken/Pakistan/16/99/95	**PETPKRRNR/GLF**	3	2	AF202233	
H7N3	HP	Australia	1994	A/chicken/Queensland/94	**PEIPRKRKR/GLF**	4	2	CY022685	11,20
H7N1	HP	USA[7]	1994	A/Pekin_robin/California/30412/94	**PEIPKRRR/GLF**	4	1	GU052922	47
H7N3	HP	Australia	1992	A/chicken/Victoria/224/92	**PEIPKKKR/GLF**	4	2	CY025077-	20
					PEIPKRKKR/GLF	4	2	CY025084	
					PEIPKKRKKR/GLF	5	3		
					PEIPKKKKKR/GLF	6	4		
H7N7	HP	Australia	1985	A/chicken/Victoria/85	**PEIPKRERKR/GLF**	4	3	CY025069	10,20
H7N7	HP	Germany	1979	A/chicken/Leipzig/79	**PEIPKKKR/GLF**	4	2	U20459.1	21
				A/goose/Leipzig/137-8/79	**PEIPKRKKR/GLF**	4	2	L43913.1	
				A/goose/Leipzig/192-7/79	**PEIPKKRKKR/GLF**	5	3	L43915.1	
				A/goose/Leipzig/187-7/79	**PEIPKKKKKR/GLF**	6	4	L43914.1	
H7N7	HP	Australia	1976	A/chicken/Victoria/76	**PEIPKRERKR/GLF**	4	3	CY024786	10,20
H7N3	HP	England	1963	A/turkey/England/63	**PETPKRRRR/GLF**	4	2	AF202238	1,10,14

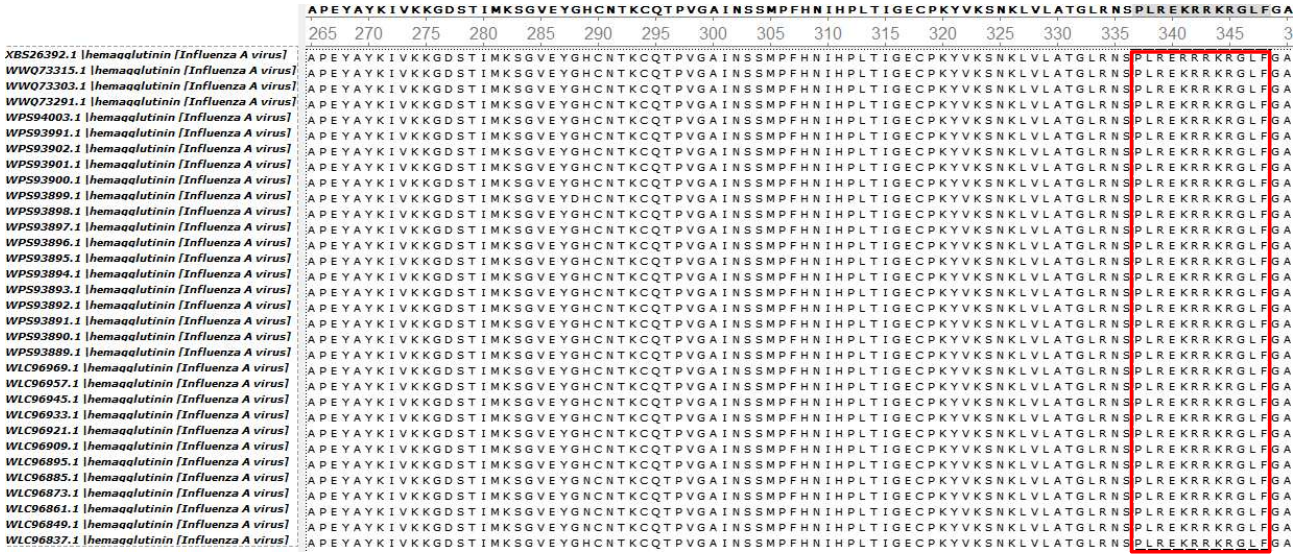
[1] LP = low pathogenic, HP = highly pathogenic as determined by IVPI.
[2] Cleavage site between HA1 and HA2; red color indicates critical basic residues; residue insertions are underlined.
[3] Number of basic residue at immediately preceding and including the -1 position; excludes the -4 position found in the LPAI cleavage site consensus motif.
[4] GenBank or GISAID record number
[5] This cleavage site is presumed HP, but awaits confirmation by IVPI testing in chickens
[6] Virus isolated from Muscovy duck meat from a passenger that originated in China. Sample was confiscated in Japan by authorities.
[7] Virus isolated from Pekin Robin (*Leiothrix lutea*) imported into a USA quarantine station from Asia. The entire lot of quarantined birds were euthanatized.
[8] HA is from similar North American wild bird lineage; same source (host cellular 28s rRNA), sequence and mechanism for insertion into the HA cleavage site; but the insertion events occurred at different times (2017 vs 2020)

Table 3. Unusual, 2-3 basic residue multi-basic cleavage sites with variable LP and HP phenotypes

Sub-type	Pheno-type[1]	location	year	Type virus	cleavage site[2]	Critical basic aa[3]	Size of insert	Accession number[4]	Ref.
H5	LP				**PQRETR/GLF**	1	0		39
H5N1 **H5N2** **H5N9**	HP	France	2015	A/chicken/France/150169a/15 A/duck/France/150233/15 A/duck/France/150236/15	**HQRRKR/GLF**	3	0	KU310447.1 KX014878.1 KX014886.1	53
H5N2	HP	Taiwan	2012	A/chicken/Taiwan/A1997/12	**PQRRKR/GLF**	3	0	KF193394.1	40
H5N2	HP	Taiwan	2008	A/chicken/Taiwan/K703-1/08	**PQRKKR/GLF**[5]	3	0	AB507264.1	34,35
H5N2	LP	USA	2004	A/chicken/Texas/298313/04	**PQRKKR/GLF**[6]	3	0	AY849793.1	6
	LP	Taiwan	2003	A/chicken/Taiwan/1209/03	**PQREKR/GLF**	2	0	AY573917.1	34,35
H5N2	LP/HP	USA	1983	A/chicken/PA/1370/83	**PQKKKR/GLF**[7]	3	0	CY107848.1	6,10,37
H5N1	HP	Scotland	1959	A/chicken/Scotland/59	**PQRKKR/GLF**[8]	3	0	GU052518.1	6
H7	LP				**PEIPKGR/GLF**	1	0		38
H7N4	LP	China	2018	A/chicken/Jiangsu/1/2018	**PELPKGR/GLF**	2	0	EPI_ISL_332358	HVRI, 48
H7N1	LP	UAE	2004	A/houbara bustard/UAE/2004	**PELPKRR/GLF**	2	0		APHA-UK
H7N3	HP	Pakistan	1995	A/chicken/Pakistan/16/99/95	**PETPKRRNR/GLF**	3	2	AF202233	2
H7N7	HP	England	1979	A/turkey/England/199/79	**PEIPKKREK/GLF**	3	2		1,9,14
H7Nx	LP	Australia	1976	A/duck/Victoria/76	**PEIPKKR/GLF**	2	0	U20463.1	33

[1] LP = low pathogenic, HP = highly pathogenic as determined by IVPI. Where both are indicated, age of chickens affected IVPI score or presence/absence of a glycosylation site.
[2] Cleavage site between HA1 and HA2; red color indicates critical basic residues; residue insertions are underlined.
[3] Number of basic residue at immediately preceding and including the -1 position; excludes the -4 position found in the LPAI cleavage site consensus motif.
[4] GenBank record number
[5] Based on OIE mandated 6 week-old chickens, the IVPI = 1.86 and virus was declared HP. However, in 8 week-old chickens the IVPI = 0.89.
[6] LP in 4 week-old chickens (0/8). Parent virus had putative glycosylation site at position 11-13 (NST). Was declare HP based on HA cleavage site sequence similarity to A/chicken/Scotland/1959
[7] Required loss of putative glycosylation at position 11-13 (NSK) for highly pathogenic phenotype
[8] Lack of putative glycosylation site at position 11-13 (KST)

9.5.Anexo V



Alineamiento múltiple de secuencias de la epizootia de VIA HPAI en Uruguay durante 2023. El recuadro en rojo indica los motivos de alta patogenicidad identificados en las secuencias por inspección manual. El motivo conservado es **PLRERRRKR/GLF**, característico de cepas altamente patógenas del subtipo H5. El alineamiento fue realizado utilizando **UniProt**.