Universidad Internacional de La Rioja

Facultad de Ciencias de la Salud

Máster Universitario en Bioinformática

Plan de propuesta. Aplicación de la IA en el estudio del codon usage en el virus del PRRS

Trabajo fin de Estudio presentado por:	Astrid Liliana Vargas Sánchez
Tipo de trabajo:	Propuesta de un proyecto de investigación candidato a presentarse a convocatoria de financiación competitiva.
Directores:	Ismael de la Iglesia San Sebastián José Ignacio Núñez
Fecha:	

Resumen

El virus del síndrome reproductivo y respiratorio porcino (PRRSV) es un patógeno que afecta principalmente a lechones y cerdas gestantes, causando graves pérdidas económicas en la industria porcina. Su impacto se debe a su alta capacidad de transmisión y variabilidad genética, lo que dificulta su control y prevención. El uso de codones en el PRRSV se refiere a la frecuencia con la que el virus emplea distintos codones para codificar un mismo aminoácido. Este aspecto es fundamental para comprender los mecanismos de replicación viral, la eficiencia en la traducción de sus proteínas y su interacción con las células huésped. Además, el análisis del uso de codones puede proporcionar información valiosa sobre la adaptación del virus y su evolución, lo que resulta clave para el desarrollo de estrategias de control. En este estudio se plantea una propuesta para la implementación de herramientas de inteligencia artificial con el objetivo de analizar el uso de codones en el genoma completo del PRRSV. Para ello, se exploran diferentes alternativas de *machine learning* no supervisado, supervisado y *deep learning*. Se concluye que existen múltiples herramientas de inteligencia artificial capaces de abordar este análisis de manera eficiente, proporcionando información relevante para futuras investigaciones.

Palabras clave: Uso de codones, virus del síndrome reproductivo (PRRSV), inteligencia artificial, aprendizaje automático, aprendizaje profundo.

Abstract

The porcine reproductive and respiratory syndrome virus (PRRSV) is a pathogen that primarily affects piglets and pregnant sows, causing significant economic losses in the swine industry. Its impact is due to its high transmission capacity and genetic variability, making its control and prevention challenging. Codon usage in PRRSV refers to the frequency with which the virus employs different codons to encode the same amino acid. This aspect is essential for understanding viral replication mechanisms, protein translation efficiency, and interactions with host cells. Additionally, analyzing codon usage can provide valuable insights into the virus's adaptation and evolution, which is crucial for developing effective control strategies. This study proposes the implementation of artificial intelligence tools to analyze codon usage in the complete PRRSV genome. To achieve this, various approaches in unsupervised learning, supervised learning, and deep learning are explored. The findings indicate that multiple artificial intelligence tools can efficiently perform this analysis, providing relevant information for future research.

Keywords: Codon usage, porcine reproductive and respiratory syndrome virus (PRRSV), artificial intelligence, machine learning, deep learning.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

El virus del síndrome reproductivo y respiratorio porcino (PRRSV, por sus siglas en inglés) es un virus de ARN monocatenario de sentido positivo perteneciente al género Betaarterivirus, dentro de la familia Arteriviridae y el orden Nidovirales (1,2). Este patógeno afecta principalmente a los cerdos domésticos, siendo el causante del síndrome reproductivo y respiratorio porcino (PRRS), una enfermedad de gran relevancia económica para la industria porcina a nivel mundial. El PRRS puede causar fallos reproductivos graves en cerdas gestantes, como abortos y nacimientos prematuros, así como enfermedades respiratorias en cerdos de todas las edades, lo que conduce a una elevada mortalidad en lechones (3,4). Con algunas excepciones, el PRRSV es endémico en las principales regiones productoras de cerdos del mundo, incluyendo América del Norte, Europa y Asia (5,6).

La inteligencia artificial (IA) ha cobrado gran relevancia en el análisis de datos biológicos en los últimos años, impulsada por los avances en el procesamiento de grandes volúmenes de datos genómicos y el desarrollo de potentes herramientas de aprendizaje automático, como el *machine learning* y el *deep learning* (7). La IA puede ser una herramienta valiosa para estudiar el uso de codones en el PRRSV, facilitará la comprensión de la eficiencia de su replicación y su relación con el huésped porcino. Además, puede contribuir al diseño de estrategias antivirales y vacunas más efectivas.

2. Marco teórico

2.1. Conceptos clave

2.1.1. Dogma central de la biología

El concepto del dogma central de la biología fue propuesto por Francis Crick en 1958 y formalizado en 1970. En su trabajo, Crick explicó que numerosos organismos utilizan el ADN de doble cadena (dsADN) como su material genético principal. En estos organismos, la información contenida en el ADN es transcrita en moléculas de ARN mensajero de cadena sencilla (ss), que actúan como intermediarios para transportar las instrucciones genéticas hacia los ribosomas. Durante este proceso, el ARN mensajero (ARNm) es leído y traducido en proteínas con la ayuda de los ribosomas y las moléculas de ARN de transferencia (ARNt), las cuales incorporan aminoácidos en un orden específico, dictado por la secuencia del ARNm (8). Aunque este flujo de información es predominante en la mayoría de los organismos, también se han identificado excepciones al dogma. Los virus presentan una notable diversidad en cuanto a la composición de su material genético. El genoma que se empaqueta dentro de los viriones puede estar constituido por ARN o ADN, y su estructura puede variar entre monocatenaria o bicatenaria (9).

2.1.2. Clasificación de Baltimore

En 1971 David Baltimore propuso un sistema para clasificar a los virus de acuerdo con el ácido nucleico que contienen y su mecanismo de replicación (10). Los grupos se describen a continuación:

Tabla 1. Clasificación de Baltimore

Grupo	Nombre	Genoma	Mecanismo
Grupo I	Virus de ADN bicatenario (dsADN)	ADN de doble cadena.	El ADN se transcribe directamente a ARNm utilizando la maquinaria del huésped.
Grupo II	Virus de ADN monocatenario (ssADN)	ADN de cadena sencilla.	El ADN monocatenario se convierte en bicatenario antes de la transcripción.
Grupo III	Virus de ARN bicatenario (dsARN)	ARN de doble cadena.	Una de las hebras del ARN actúa como molde para producir ARNm.
Grupo IV	Virus de ARN monocatenario de cadena positiva (ssARN+)	ARN de cadena positiva, que funciona directamente como ARNm.	El ARN genómico es traducido directamente por los ribosomas.
Grupo V	Virus de ARN monocatenario de cadena negativa (ssARN−)	ARN de cadena negativa, complementario al ARNm.	El ARN− es transcrito en ARNm por una ARN polimerasa viral.
Grupo VI	Virus de ARN de transcripción inversa	ARN de cadena positiva.	El ARN se retrotranscribe en ADN mediante la enzima transcriptasa inversa, que luego se integra al genoma del huésped.
Grupo VII	Virus de ADN de transcripción inversa	ADN de doble cadena.	El ADN se transcribe en ARN y luego se retrotranscribe para formar nuevos genomas de ADN.

Adaptación de Shors T., 2021

2.1.3. Marcos de lectura abiertos

Los ORF (Open Reading Frames o marcos de lectura abiertos) son secuencias de nucleótidos dentro de un genoma que pueden ser traducidas en proteínas. La traducción comienza en el extremo 5´ del ORF y progresa con la incorporación de aminoácidos según la secuencia de codones hacia el extremo 3´. Un ORF comienza con un codón de inicio (generalmente AUG) y termina en un codón de parada (UAA, UAG o UGA). En los virus, los ORF desempeñan un papel crucial en la codificación de proteínas estructurales y no estructurales necesarias para la replicación, ensamblaje y función del virus (11)

2.2. Virus del síndrome reproductivo y respiratorio porcino

2.2.1. Características del virus PRRS

El PRRSV es un virus envuelto de forma esférica u ovalada, sensible a los solventes lipídicos, con un diámetro que varía entre 50 y 60 nm. Tiene la capacidad de sobrevivir en congelación durante varios años, pero su viabilidad se reduce significativamente en temperaturas más altas, sobreviviendo alrededor de un mes a 4°C y apenas 48 horas a 37°C (12) . Su genoma viral, empaquetado por proteínas de la nucleocápside, tiene una longitud aproximada de 15 kb y consiste en una sola hebra de ARN de cadena positiva (1,3,13).

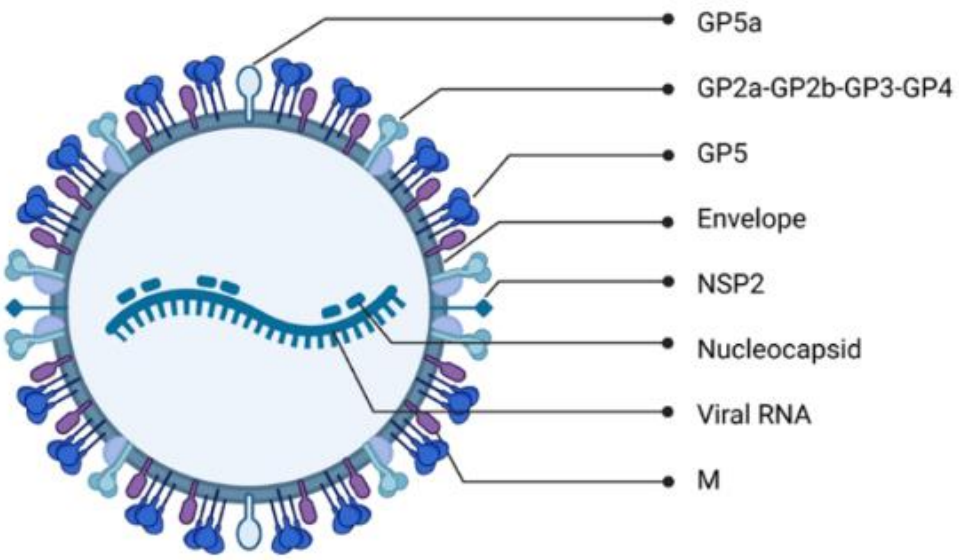

Figura 1. Representación esquemática del virus PRRSV. Tomado de Orosco F., 2024.

El ARN viral contiene once ORFs, los cuales codifican diversas proteínas esenciales tanto para la replicación del virus como para la formación de su estructura.

El gen de la replicasa se compone de los ORF1a y ORF1b, localizados en los tres cuartos proximales del genoma. Estos ORFs codifican las proteínas no estructurales (NSPs), que son fundamentales para los procesos de replicación y transcripción del ARN viral. Entre estas proteínas se encuentran enzimas clave como la ARN polimerasa, proteasas, helicasas, entre otras, que son esenciales para completar el ciclo de replicación viral (1,3).

Dentro de las glicoproteínas asociadas a la membrana, GP5 es la más importante, ya que está involucrada en la adhesión del virus a las células del huésped. Además, es la más abundante en el virión y juega un papel fundamental en la inducción de la respuesta inmune por parte del organismo infectado (1,14).

Tabla 2. Caracteristicas y funciones de las proteinas principales del virus del síndrome respiratorio y reproductivo porcino (PRRSV)

Gen	Proteína	Funciones conocidas
orf1a	nsp1α	Modula la producción de ARNm subgenómico. Puede actuar como antagonista del interferón (IFN).
orf1a	nsp1β	Puede actuar como antagonista del interferón (IFN).
orf1a	nsp2	Enzima que elimina la ubiquitina. Potencial antagonista del interferón (IFN). Proteína transmembrana que participa en la alteración de la membrana, formando parte del complejo de replicación.
orf1a	nsp3	Proteína con un dominio transmembrana que contribuye a la modificación de la membrana y a la formación del complejo replicativo.
orf1a	nsp4	Principal proteasa S. Promueve la apoptosis. Potencial inhibidor del interferón (IFN).

orf1a	nsp5	Proteína transmembrana que probablemente participa en la modificación de la membrana.
orf2a	GP2a	Proteína estructural clave para la capacidad infectiva del virus. Actúa como proteína de unión viral.
orf2b	E	Proteína de la envoltura
orf3	GP3	Proteína estructural clave para la capacidad infectiva del virus. Actúa como proteína de unión viral. Es altamente antigénica y podría jugar un papel en la neutralización del virus.
orf4	GP4	Proteína estructural clave para la capacidad infectiva del virus. Actúa como proteína de unión viral. Es altamente antigénica y podría jugar un papel en la neutralización del virus.
orf5	GP5	Proteína estructural clave para la capacidad infectiva del virus. Actúa como proteína de unión viral. Es altamente antigénica y podría jugar un papel en la neutralización del virus. Los análisis filogenéticos del ORF que codifica esta proteína han revelado que las cepas del PRRSV tipo 2 pueden dividirse en nueve linajes distintos a nivel filogenético.
orf5a	ORF5a	Proteína estructural crucial para la supervivencia del virus.
orf6	M	Proteína de la matriz, que junto con GP5, forma un complejo crucial para la entrada del virus en las células del huésped. También juega un papel clave en el ensamblaje y gemación del virus.

orf7	N	Proteína de nucleocápside (N), que protege el ARN viral y es uno de los principales objetivos de la respuesta inmune.

Adaptación de Lunney JK, et al., 2015; Wu W, et al., 2021

El PRRSV es especialmente propenso a sufrir mutaciones. Esta alta tasa de mutación se debe a la falta de mecanismos de corrección en su ARN polimerasa, lo que incrementa la probabilidad de errores durante la replicación viral. Además, se ha observado que las cepas del PRRSV experimentan recombinación genética, lo que contribuye aún más a su variabilidad. Como resultado, el virus presenta una amplia diversidad de cepas, lo cual complica considerablemente el desarrollo de estrategias efectivas de control, incluidas las vacunas y los tratamientos antivirales (3,12).

En la actualidad, el PRRSV está siendo objeto de intensos estudios con el fin de evaluar la resistencia y susceptibilidad a la infección. Los avances tecnológicos en bioinformática y herramientas de secuenciación han permitido progresos importantes en las áreas de genómica, transcriptómica, proteómica y metabolómica. Estas áreas son clave para entender mejor la interacción entre el virus y el huésped y, por lo tanto, para desarrollar nuevas estrategias de control.

Entre los avances más relevantes, se ha estudiado el papel de los microARNs en la regulación de la respuesta inmune al PRRSV. Algunos ejemplos incluyen:

- **miR-181:** Regula a la baja la expresión del receptor CD163, que es un receptor crucial para la entrada del virus en las células del huésped. Debido a su importancia en la patogénesis del virus, CD163 es un objetivo fundamental en la investigación para el desarrollo de tratamientos antivirales y vacunas que bloqueen la entrada del PRRSV en las células (1).
- **miR-23:** Induce la expresión de interferones tipo I mediante la activación de los factores de transcripción IRF3 e IRF7. Los interferones tipo I son esenciales en la respuesta antiviral y la investigación actual busca promover su activación eficiente para controlar la replicación del PRRSV. Como resultado, se están evaluando estrategias terapéuticas que potencien la producción de interferones tipo I como parte de nuevos tratamientos o vacunas (1).

2.2.2. Características del síndrome reproductivo y respiratorio porcino (PRRS)

La enfermedad se detectó por primera vez en Estados Unidos en la década de 1980, donde fue descrita inicialmente como una "misteriosa enfermedad" debido a su sintomatología inusual y a la falta de un diagnóstico preciso (12). Se han identificado dos grandes genotipos del virus: el europeo (tipo 1) y el norteamericano (tipo 2) (6) .

La principal vía de transmisión es el contacto directo entre cerdos, a través de secreciones como saliva, orina y heces. También puede transmitirse verticalmente de madre a lechones a través de la placenta, así como mediante aerosoles (12).

Los síntomas más comunes son:

- Fallo reproductivo en cerdas: Abortos, mortinatos y nacimientos prematuros.
- Fallos en la fecundación y reducción en el tamaño de las camadas.
- Reducción en la tasa de crecimiento: Los lechones nacidos de madres infectadas suelen ser débiles, con alta mortalidad neonatal.
- Enfermedad respiratoria: Afecta a cerdos de todas las edades, pero es especialmente grave en lechones y cerdos jóvenes. Causa neumonía, dificultad respiratoria, tos y fiebre (3) (5) (6).
- Otros signos asociados incluyen anorexia, fiebre (pirexia), disminución de la producción de leche (agalactia), letargo, predisposición a infecciones secundarias por bacterias u otros patógenos respiratorios. En algunos casos, también se puede observar decoloración de la piel (12).

2.3. Codon usage

De los 64 tripletes de codones, 61 codifican para los 20 aminoácidos, mientras que los tres restantes actúan como señales de parada en la traducción. La degeneración del código genético se refiere a que muchos aminoácidos pueden ser codificados por múltiples codones diferentes (por ejemplo, el aminoácido leucina puede ser codificado por seis codones distintos), los cuales son conocidos como codones sinónimos (15). El estudio del uso de codones (*codon usage*) analiza la frecuencia con la que se utilizan diferentes codones para codificar el mismo aminoácido en un organismo. Algunos codones son preferidos sobre otros para codificar un aminoácido particular, y esta preferencia varía según la especie, los tejidos

dentro de un mismo organismo, los genes funcionalmente relacionados e incluso dentro de un solo gen (16,17).

2.3.1. Factores que influyen en el uso de codones

1. Abundancia de ARNt: Cada codón tiene un ARN de transferencia (ARNt) que lo reconoce y lo empareja con el aminoácido correspondiente. Los organismos tienden a utilizar codones que coinciden con los ARNt más abundantes para aumentar la eficiencia de la traducción (18,19).
2. Eficiencia de la traducción: Los codones más frecuentes tienden a ser traducidos más rápido y con mayor precisión, lo que puede influir en la velocidad de síntesis, el plegamiento y la regulación de las proteínas (19).
3. Adaptación evolutiva: Algunos organismos o genes pueden preferir ciertos codones como resultado de adaptaciones evolutivas para optimizar la expresión génica en condiciones específicas.
4. Restricciones genómicas: El contenido global de GC en el genoma puede influir en la elección de codones, ya que algunos codones tienen más guanina y citosina en sus bases que otros (3,17).

El uso de codones en el virus PRRSV está influenciado por su huésped, ya que la replicación viral depende de la maquinaria celular de este. La evolución del virus puede estar condicionada por la selección de codones, ya que utilizar aquellos preferidos por el huésped puede optimizar la eficiencia de la traducción, lo que resulta en una producción más rápida y eficiente de proteínas virales. Este mecanismo podría influir en factores clave como la virulencia del virus y en su capacidad de supervivencia (20:22).

2.4. Análisis filogenético

El análisis filogenético de ADN, ARN o secuencias de proteínas, se ha convertido en una herramienta importante para estudiar las relaciones evolutivas entre diferentes especies u organismos, sustituyendo la comparación de características morfológicas o fisiológicas. Las relaciones filogenéticas usualmente están representadas en árboles filogenéticos donde las ramas representan la evolución a partir de un ancestro común (23).

2.4.1. Árboles filogenéticos con raíz

En esta representación gráfica se incluye un ancestro común en la base del árbol (Raíz), lo que permite identificar la dirección evolutiva de las especies a lo largo del tiempo. La figura 2 muestra que la especie A diverge de las especies B, C y D 30 millones de años (Ma), la especie B diverge de las especies C y D 22 millones de años (Ma) y la especie C diverge de la especie D 7 millones de años (Ma). El diagrama muestra que las especies C y D son evolutivamente cercanas entre sí, mientras que las especies A y D están distantes (24).

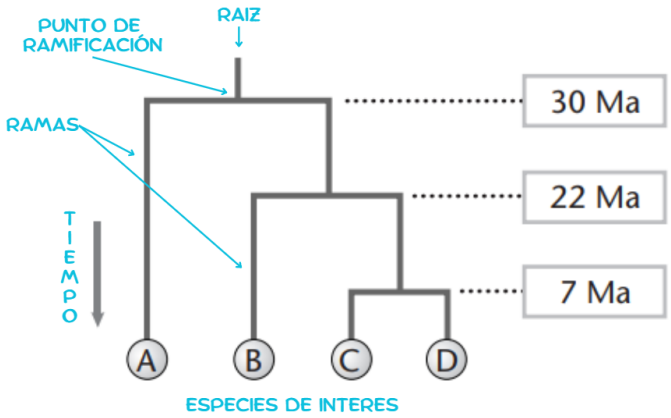

Figura 2. Elementos de un árbol filogenético con raíz. Adaptación de Higgs PG, et al., 2005.

2.4.2. Árboles filogenéticos sin raíz

En esta representación gráfica se observa las relaciones evolutivas entre diferentes especies, pero no se especifica un ancestro común. La figura 3 muestra los nodos internos (i, j) que representan ancestros para los cuales no se tiene datos de la secuencia. La longitud de las ramas varía dependiendo de la distancia evolutiva. Por ejemplo, ha habido muchos cambios a lo largo de la rama de i a A, y relativamente pocos cambios a lo largo de la rama de j a D.

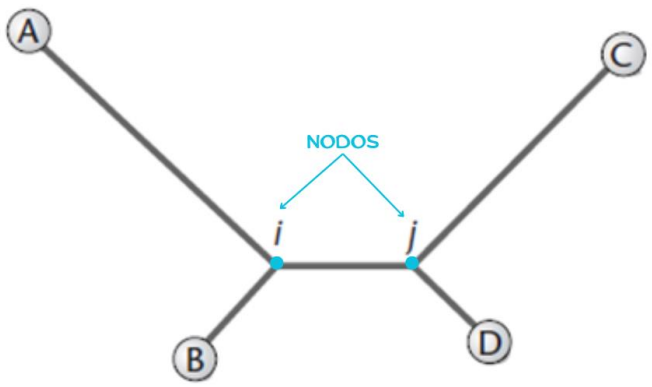

Figura 3. Elementos de un árbol filogenético sin raíz. Adaptación de Higgs PG, et al., 2005.

2.5. Inteligencia artificial

A lo largo de los años, la IA ha sido interpretada y definida de diversas maneras. Sin embargo, una de las descripciones más claras y concisas es que se trata de un campo de la informática que se enfoca en el desarrollo de tecnologías que permitan la automatización de tareas intelectuales, normalmente desarrolladas por humanos (25:27). La IA ofrece la posibilidad de explorar vías novedosas en biología mediante el uso de computadores entrenados para analizar grandes volúmenes de datos. Dentro de este campo, *Machine Learning* y *Deep Learning* son subdisciplinas clave (7).

2.5.1. Machine learning o aprendizaje automático

Fue un término acuñado inicialmente en 1959 por Arthur Samuel, un destacado científico informático de la época. Es un campo que se centra en el desarrollo de algoritmos y modelos que permiten que las computadoras aprendan de los datos sin ser programadas explícitamente para ello (25,28). Estos algoritmos se basan en fundamentos estadísticos para su funcionamiento (29). *Machine Learning* se aplicó por primera vez en medicina durante las décadas de 1980 y 1990, con un enfoque principal en el análisis de imágenes médicas (30).

Machine Learning se subclasifica en tres categorías: aprendizaje no supervisado, aprendizaje supervisado y aprendizaje de refuerzo.

2.5.1.1. Aprendizaje no supervisado

En este tipo de modelos, los datos no se encuentran etiquetados u organizados previamente, lo que implica que los algoritmos deben analizar la información para reconocer y determinar si existen patrones subyacentes. El objetivo principal de estos algoritmos es identificar similitudes entre los datos y agruparlos en categorías (26,28,31). Se consideran no supervisados porque los patrones identificados son generados por el propio algoritmo, sin depender de etiquetas o instrucciones previas. Una vez obtenidos, los patrones identificados deben ser evaluados para determinar su utilidad y relevancia en el análisis (25).

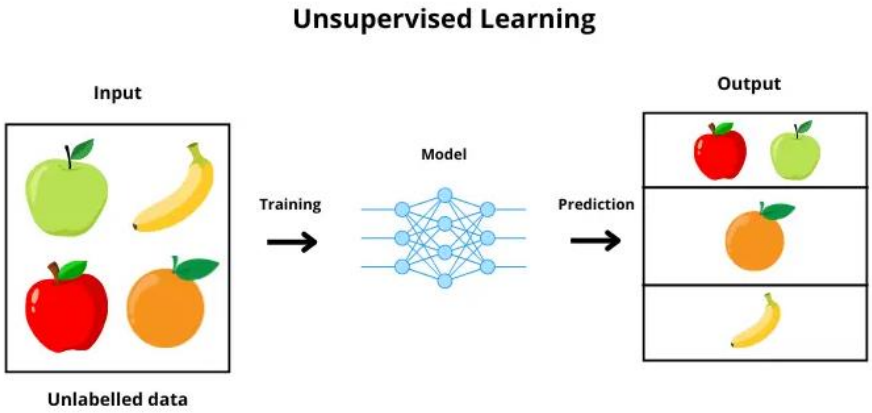

Figura 4. Ejemplo de aprendizaje no supervisado. Tomado de Labs S., 2024.

Cuando se trabaja con grandes volúmenes de datos con múltiples dimensiones, resulta muy útil emplear técnicas de *machine learning* no supervisado para reducir la dimensionalidad y facilitar su análisis (32). Entre las técnicas más utilizadas se encuentran:

- *Análisis de componentes principales (PCA)*

El *PCA* es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos no etiquetados. Esto se logra transformando las variables originales en un nuevo conjunto de variables no relacionadas llamadas componentes principales que preservan la mayor cantidad de información posible de las variables originales. *PCA* es un método lineal, los ejes resultantes son combinaciones lineales de las características originales. La métrica de calidad con la cual se evalúa el PCA es la varianza explicada, generalmente se busca retener al menos el 80 – 90% de los datos (33,34). Podemos implementar el *PCA* utilizando diversos lenguajes de programación. Por ejemplo, en el lenguaje de programación R, se puede acceder al *PCA* mediante librerías especializadas como *prcomp.* Por otro lado, en Python, el *PCA* se puede realizar utilizando la librería *Scikit-learn* (35,36).

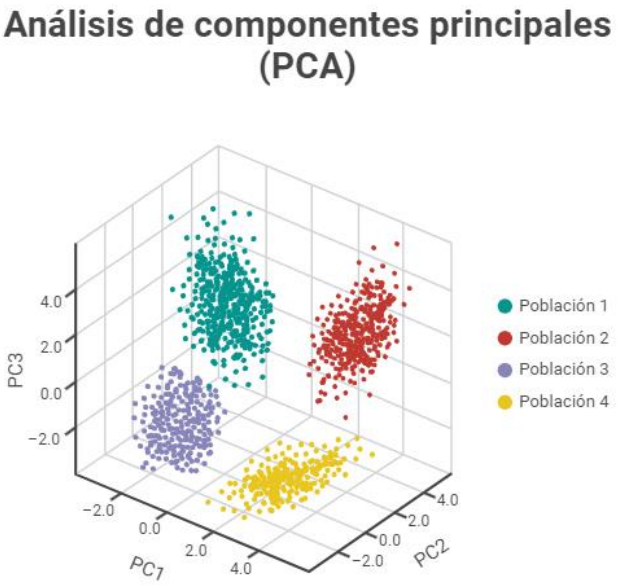

Figura 5. Ejemplo de PCA en 3D. Fuente: Elaboración propia. Realizado con herramientas de biorender.com

Otras técnicas de reducción de dimensionalidad:

- *Multidimensional Scaling (MDS)*

Es una técnica estadística que facilita la visualización de las relaciones entre un conjunto de datos al reducir su dimensionalidad. *MSD* busca preservar las distancias o similitudes entre los puntos de los datos originales (37). La métrica de calidad es el estrés *(Stress function S)*, cuanto menor sea mejor es la representación de los datos en el nuevo espacio dimensional reducido (38,39).

Stress Values for MDS-Scaled Randomly Generated Matricies of Fifteen Objects (800 matrices per dimension)

Dimensions	*Minimum Stress*	*Mean Stress*	*Maximum Stress*
1	0.382	0.449	0.51
2	0.212	0.263	0.3
3	0.141	0.172	0.203
4	0.083	0.116	0.149
5	0.058	0.082	0.113
6	0.033	0.058	0.08
7	0.027	0.042	0.058
8	0.012	0.029	0.046
9	0.009	0.025	0.042
10	0.007	0.019	0.034

NOTE: MDS = multidimensional scaling.

Figura 6. Valores óptimos de estrés en 10 dimensiones. Tomado de Sturrock K et al., 2000.

- *t-SNE (t-Distributed Stochastic Neighbor Embedding):*

Es una técnica no lineal que reduce la dimensionalidad mientras preserva la estructura de los datos. Es ideal para la visualización de datos con muchas dimensiones. Este método mide las similitudes entre pares de puntos basándose en una distribución gaussiana centrada en cada punto. La métrica de calidad es la divergencia de Kullback-Leibler (*KL Divergence)*, cuanto menor sea mejor. Indica qué tan bien se mantienen las relaciones locales (32,40).

- *Isomap (Isometric Mapping)*

Es una técnica de reducción de dimensionalidad que combina conceptos de *PCA* y *MSD*. Este método busca preservar las propiedades geométricas de los datos en un espacio reducido. Utiliza distancias geodésicas (distancias más cortas a lo largo de una superficie) que se representan en un gráfico de vecindad (basado en vecinos más cercanos o un umbral de distancia) La métrica de calidad es el error de reconstrucción (*Residual Variance*), cuanto menor sea mejor. Mide la discrepancia entre distancias originales y proyectadas (41).

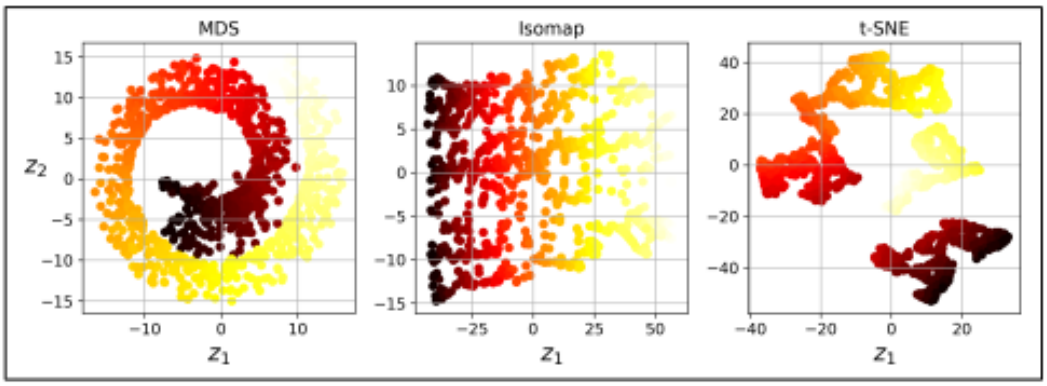

Figura 7. Reducción de dimensionalidad usando varias técnicas. Tomado de Géron A., 2019.

- *K-Means*

Es un algoritmo capaz de realizar agrupamiento o *clustering* de datos de manera eficiente y rápida. Divide un conjunto de datos en un número predefinido de grupos (*K*), basándose en las similitudes entre las observaciones. Este algoritmo utiliza centroides, que representan el centro de los clústeres. Los centroides se calculan como el promedio de los puntos asignados a cada clúster, utilizando métricas como la distancia euclidiana para determinar la proximidad entre los datos. Su objetivo es minimizar la dispersión dentro de los grupos (32,42). La métrica de calidad es la inercia (*Within-cluster Sum of Squares*), cuanto menor sea mejor. Indica la coherencia interna de los clústeres (43).

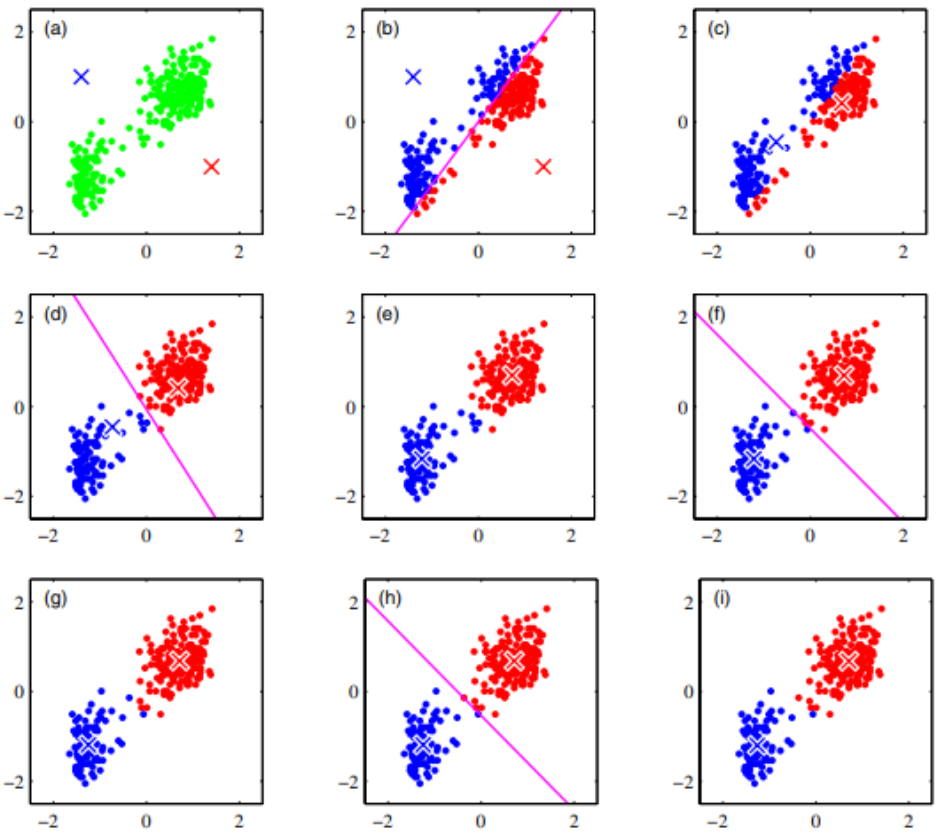

Figura 8. Clusterización con K-means. Tomado de Bishop CM., 2006.

2.5.1.2. Aprendizaje supervisado.

Se basa en algoritmos que emplean datos de entrada o un conjunto de datos de entrenamiento previamente etiquetados u organizados. Estos datos son analizados para predecir valores de salida conocidos, lo que permite identificar patrones y tendencias. A través de este proceso, el algoritmo se entrena y mejora su rendimiento progresivamente, optimizando su capacidad para generar predicciones precisas cuando se le presentan nuevos datos en el conjunto de prueba (26,28,44). Este modelo se ocupa de problemas de clasificación y regresión. La clasificación implica predecir a que categoría pertenece un dato y la regresión implica predecir datos numéricos (25,44).

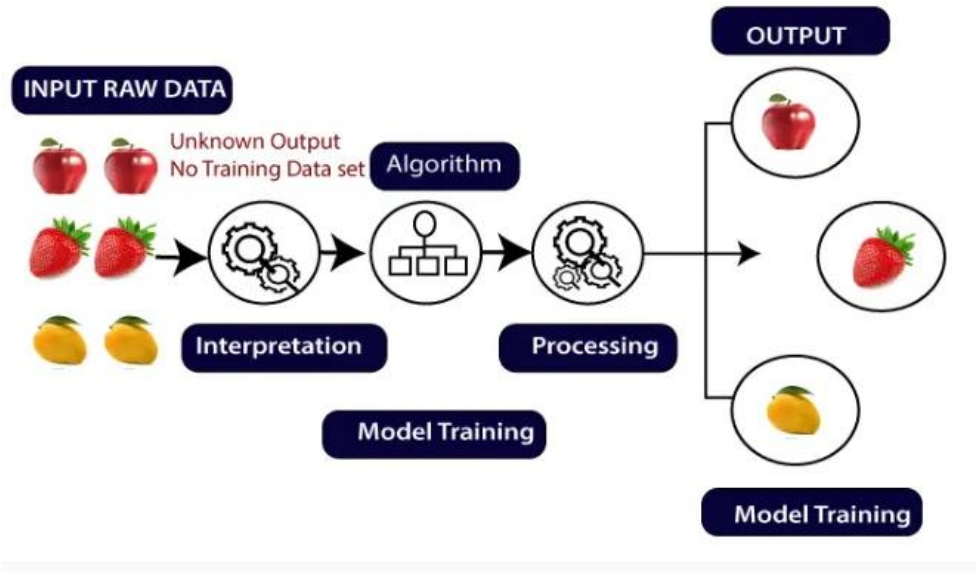

Figura 9. Ejemplo de aprendizaje supervisado. Tomado de Labs S., 2024.

Entre las técnicas más utilizadas se encuentran:

- Regresión lineal

Es un modelo estadístico que se utiliza para describir la relación lineal entre una o más variables independientes o predictoras (X) y una variable dependiente o de respuesta (Y), permitiendo predecir los valores de esta última a partir de los datos observados de las variables independientes. En la regresión lineal simple, la relación se establece con una única variable independiente, mientras que, en la regresión lineal múltiple, se consideran varias. El objetivo de este modelo es encontrar los coeficientes óptimos que minimicen el error entre las predicciones y los valores reales (45,46). Gracias a esta capacidad predictiva, en áreas como la medicina, se pueden identificar factores de riesgo que afectan determinados resultados y generar pronósticos individuales con mayor precisión (46).

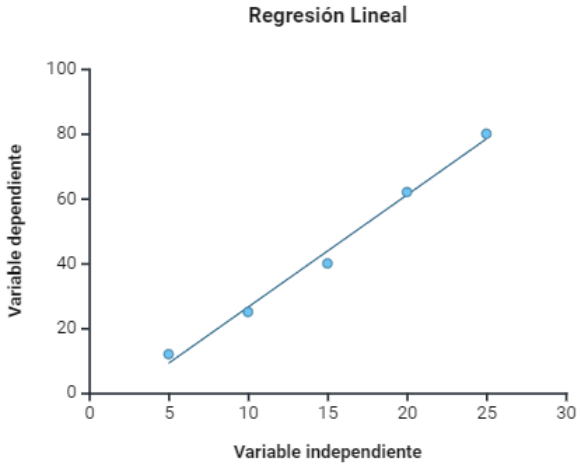

Figura 10. Ejemplo de regresión lineal. Fuente: Elaboración propia. Realizado con herramientas de biorender.com

- Regresión logística

Es un modelo estadístico que se utiliza para clasificar, es muy útil para predecir categorías o clases. Su principal objetivo es estimar la probabilidad de que una observación pertenezca a una categoría específica de una variable dependiente categórica, como "sí/no" o "presente/ausente". A diferencia de la regresión lineal, que predice valores continuos, la regresión logística puede describir la relación entre una o varias variables independientes y una variable dependiente categórica (45:48). La relación entre las variables se mide a través de los *odds ratios* (OR), cuyo valor depende de las variables independientes y refleja cómo influye cada una en la probabilidad del evento de interés (49,50).

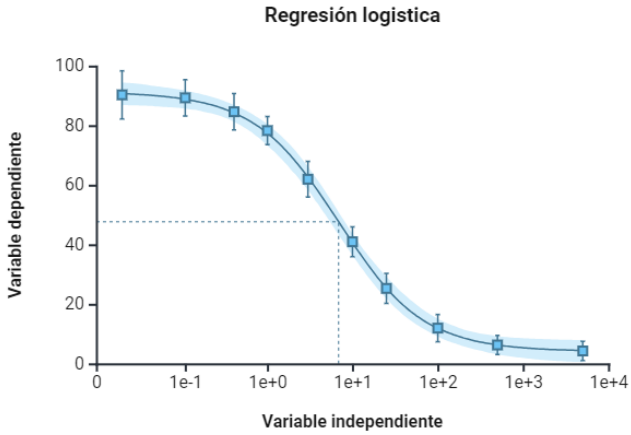

Figura 11. Ejemplo de regresión logistica. Fuente: Elaboración propia. Realizado con herramientas de biorender.com

- *k-Nearest Neighbors (k-NN)*

Es un método utilizado para clasificación y regresión no paramétrico. Los objetos con características similares tienden a pertenecer a la misma categoría o tener valores similares y se agrupan en un espacio de características de *n* dimensiones. En este modelo se suele utilizar métricas como la distancia euclidiana para calcular las diferencias entre las observaciones de datos (51).

- *Support Vector Machines (SVMs)*

Este algoritmo se utiliza tanto para clasificación como para regresión. Su objetivo es encontrar un hiperplano óptimo que separe los datos en diferentes clases, maximizando el margen entre los puntos más cercanos de cada clase. Este proceso se lleva a cabo con una función de elección (52).

Los puntos ubicados en los límites del margen se denominan vectores de soporte (*Support Vectors*) y la línea o plano central del margen representa el hiperplano óptimo de separación (53).

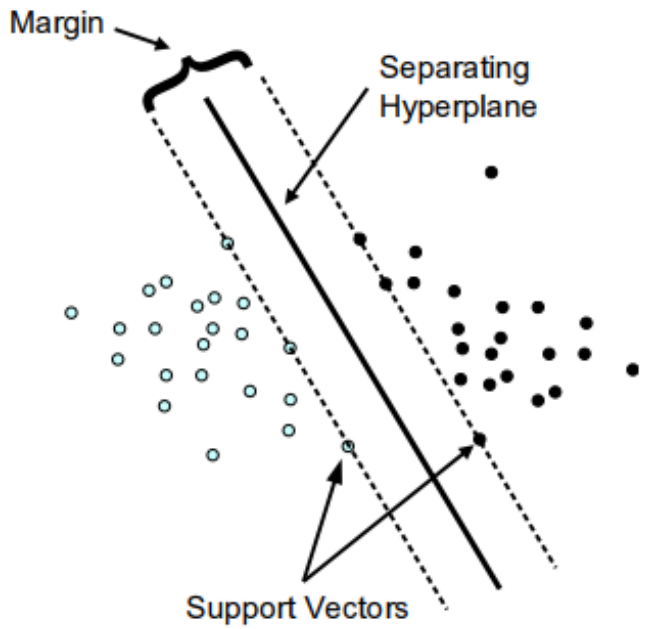

Figura 12. Clasificación con SVM. Tomado de Meyer D et al., 2015.

- Árboles de decisiones

Es un modelo utilizado para clasificación y regresión que representa las decisiones a través de una estructura de árbol. Cada nodo corresponde a una característica especifica, las ramas representan un rango de valores y cada hoja representa una etiqueta de clase o un valor de salida. Se construye recursivamente hasta llegar a un resultado final (hojas) (54).

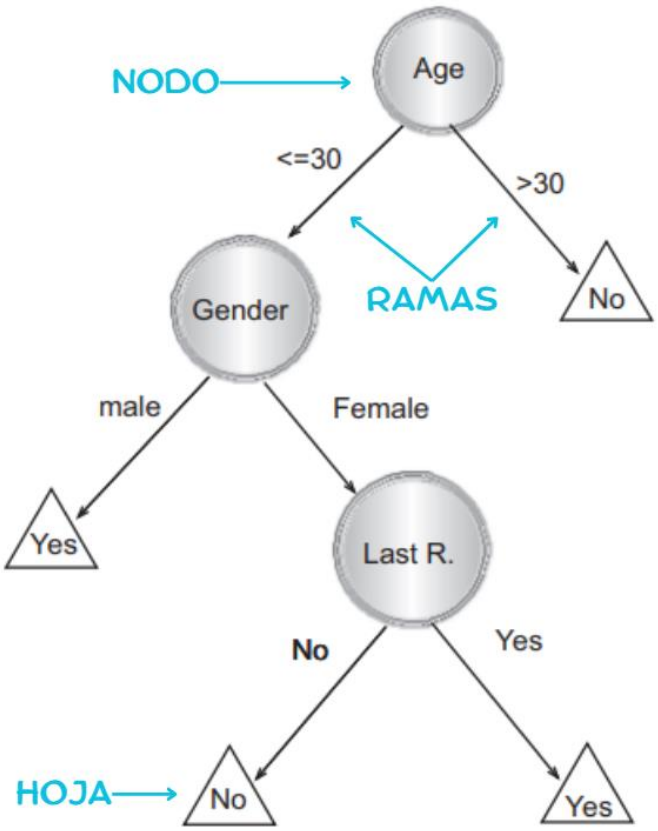

Figura 13. Ejemplo de árbol de decisión. Adaptado de Rokach L et al., 2014.

- *Random Forests*

Random Forests o conjunto de bosques aleatorios es un modelo que combina múltiples árboles de decisión para mejorar la precisión. En cada árbol la aleatorización de los nodos genera múltiples variaciones en las respuestas, y al combinarlas, se obtiene un modelo más preciso y menos propenso al sobreajuste. La predicción final se obtiene mediante el promedio (para regresión) o el voto mayoritario (para clasificación) (54).

2.5.1.3. Aprendizaje por refuerzo

En este modelo se entrena al algoritmo para una tarea específica en la que no hay una respuesta única correcta, pero se desea un resultado general. Es equivalente al aprendizaje humano ya que se basa en un proceso de prueba y error, complementado con recompensas o castigos según las decisiones tomadas. Aunque es un modelo prometedor, sus aplicaciones en medicina y biología actualmente son muy limitadas (25,26).

2.5.2. Deep learning o aprendizaje profundo

Deep Learning es un subcampo del aprendizaje automático que emplea redes neuronales artificiales organizadas en capas para procesar datos complejos. Es eficaz cuando se trabaja con grandes volúmenes de datos (26). Además, estas redes pueden aprender tanto de manera supervisada como no supervisada (28).

Inspiradas en la estructura y el funcionamiento de las neuronas del cerebro humano, las redes neuronales han sido optimizadas para realizar procesos de aprendizaje de manera más eficiente. Están formadas por nodos, que simulan los cuerpos celulares, y conexiones, que imitan el papel de los axones y dendritas (25). Cada neurona recibe un conjunto de entradas, las procesa mediante una función de activación, y transmite el resultado a la siguiente capa de la red (55).

Gracias a estas mejoras, *Deep Learning* es capaz de descubrir patrones y relaciones en los datos que resultan inaccesibles para los algoritmos tradicionales de *Machine learning*, lo que ha permitido avances significativos en áreas como la biología molecular, la genómica y la biomedicina. Con la capacidad de procesar información masiva, estas herramientas están revolucionando el entendimiento de fenómenos biológicos, como la interacción virus-huésped y la evolución genética (7).

Las redes neuronales se dividen en diferentes tipos, mencionaremos algunos de ellos:

2.5.2.1. Redes neuronales *Feedforward* (Perceptrón Multicapa, MLP)

En este tipo de red las neuronas están organizadas por capas donde se incluye una capa de entrada, una o varias capas ocultas dependiendo de la complejidad de los datos y una capa de salida. Los datos fluyen en una sola dirección, sin ciclos o bucles (55).

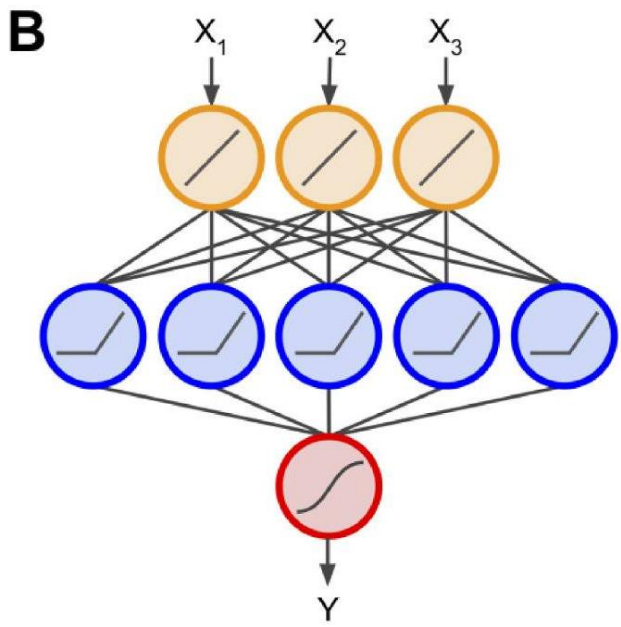

Figura 14. Ejemplo de red neuronal Feedforward MLP. Adaptado de Choi RY et al, 2020.

2.5.2.2. Redes neuronales Recurrentes (RNN)

A diferencia de las redes neuronales *Feedforward*, en este tipo de red la información se retroalimenta a través de bucles. Las RNN son capaces de recordar información previa, lo que las hace especialmente útiles para manejar datos secuenciales y dependientes del tiempo (55).

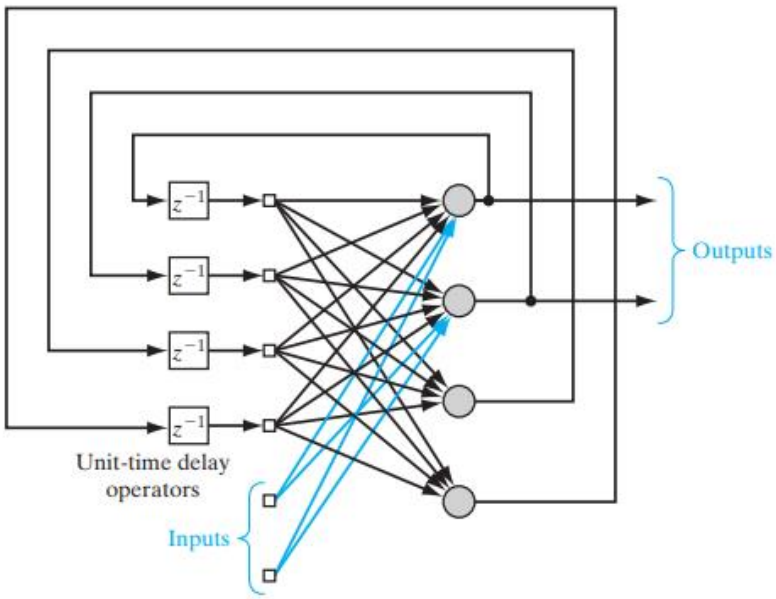

Figura 15. Ejemplo de redes neuronales recurrentes. Tomado de Haykin SS, 2009.

2.5.2.3. Transformers

Es un modelo de red neuronal diseñado para procesar secuencias de datos largos, como texto de manera eficiente y precisa. En comparación con las RNN, los *transformers* son más eficientes ya que analizan toda la secuencia de entrada simultáneamente y no de manera secuencial. Una de las características más importantes de este modelo es que utilizan una técnica llamada "atención". Este mecanismo permite al modelo enfocarse en partes relevantes de la secuencia (56).

2.6. Inteligencia artificial y *codon usage*

Algunos estudios han empleado herramientas de inteligencia artificial como *mBART* (19), *Presyncodon* (57), *JCat* (58), *Gene Designer* (59), *Gene Composer* (60), *Optimizer* (61), entre otras, para predecir el uso de codones en diversos organismos. Muchas de estas herramientas se fundamentan en el uso de redes neuronales profundas y modelos basados en *transformers* (19). Predecir computacionalmente el *codon usage* en diferentes organismos puede proporcionar una herramienta valiosa en biotecnología, ingeniería genética, medicina, agricultura y otros ámbitos de la biología molecular. Esta capacidad permitirá optimizar la expresión génica, mejorar la producción de proteínas heterólogas (no propias del hospedador), mejorar el diseño de genes sintéticos, estudiar la evolución y funcionalidad de los genes, desarrollar vacunas y potencialmente descubrir nuevas formas de tratamiento para enfermedades (19).

Específicamente para el PRRSV, diversos estudios han utilizado herramientas de *machine learning* no supervisado, como el Análisis de Componentes Principales (PCA), para investigar la tendencia en el uso de codones en distintas cepas, enfocándose en regiones como el *orf5a* (3,62); sin embargo, hasta ahora no se han reportado diferencias específicas en el uso de codones entre los distintos linajes en el genoma completo del virus (3).

3. Justificación

La enfermedad causada por el PRRSV provoca pérdidas económicas significativas en la producción porcina, principalmente debido a la reducción en la tasa de crecimiento de los cerdos, el incremento en la mortalidad y los costos adicionales en tratamientos veterinarios. Estos impactos económicos son el resultado directo de un aumento en las tasas de mortalidad y morbilidad dentro de las explotaciones porcinas afectadas. Algunos estudios reportan que los brotes pueden generar una pérdida promedio de $255 USD por cerdo (6). Por otro lado, investigaciones realizadas en los Estados Unidos estiman que la enfermedad puede costar a la industria porcina al menos 600 millones de dólares al año (63).

El uso de codones en el PRRSV, es un aspecto relevante que influye en la replicación viral, la eficiencia en la traducción de sus proteínas y la interacción con las células huésped. Al ser un virus con una alta tasa de mutación y variabilidad genética, diferentes cepas de PRRSV pueden mostrar diferencias en su preferencia de uso de codones. Estas diferencias podrían estar relacionadas con la adaptación a diferentes poblaciones de cerdos o con la evasión del sistema inmunológico. Este sesgo puede variar entre las dos principales variantes del virus (tipo 1, europeo, y tipo 2, norteamericano) (62). Comparar los patrones de composición de codones entre el virus y el cerdo nos permite entender las adaptaciones que el virus ha desarrollado a lo largo de su evolución (3).

El análisis de la preferencia en el uso de codones en virus es de gran relevancia en biología molecular, ya que este conocimiento permite el diseño de vacunas basadas en proteínas (64). Además, proporciona información clave sobre fenómenos como las transferencias horizontales de genes. Este análisis también es fundamental para revelar relaciones evolutivas entre especies, al comparar cómo diferentes organismos utilizan sus codones. Asimismo, contribuye significativamente a estudios de ingeniería genética (17).

4. Planteamiento del problema

Algunos estudios han explorado los patrones de uso de codones en el PRRSV, con un enfoque particular en el *ORF5a* (65). Sin embargo, a pesar de su relevancia, la información disponible sobre los patrones de uso de codones en el resto del genoma del PRRSV sigue siendo limitada (62).

4.1. Pregunta de investigación

¿Permiten las herramientas de *machine learning* y *deep learning* diferenciar el uso de codones del PRRSV en su genoma completo?

4.2. Hipótesis

El uso de herramientas bioinformáticas basadas en inteligencia artificial para analizar el uso de codones del PRRSV permitirá identificar patrones específicos y contribuirá a un mejor entendimiento de la variabilidad viral y la evolución del virus.

4.3. Objetivo general

Este estudio tiene como objetivo proponer un plan de propuesta para implementar herramientas de inteligencia artificial que permitan analizar el uso de codones en el genoma completo del PRRSV.

4.3.1. Objetivos específicos

- Revisar y analizar las metodologías actuales sobre el uso de codones en PRRSV.
- Proponer el desarrollo de un conjunto de herramientas bioinformáticas basadas en inteligencia artificial que permitan analizar el uso de codones en el genoma completo de PRRSV.
- Utilizar los datos obtenidos para profundizar en la comprensión de los mecanismos de replicación y adaptación del PRRSV en el huésped porcino que permitan establecer una correlación entre el uso de codones y virulencia de las cepas de PRRSV.

5. Metodología

5.1. Recopilación de datos

Las secuencias utilizadas para este análisis se podrán descargar desde la página de *Genbank* del Centro Nacional de Información Biotecnológica (NCBI). Estas incluirán genomas completos de diversas cepas, provenientes de distintas regiones del mundo.

5.2. Análisis filogenético

El siguiente paso consiste en llevar a cabo un alineamiento de secuencias con el objetivo de identificar las regiones conservadas entre las diversas cepas analizadas. Este proceso puede realizarse utilizando distintas herramientas, como *ClustalW* (66), *Clustal Omega* (67), *MAFFT* (68), entre otras. Los eventos de recombinación de las secuencias alineadas se podrán estudiar con herramientas como *RDP5* (69), *SimPlot* (70), entre otras.

A partir del alineamiento se generará un árbol filogenético para observar las relaciones evolutivas entre las cepas. Para este análisis, se empleará el método de Máxima Verosimilitud (ML), ya que ofrece una alta precisión en la estimación de las relaciones evolutivas y es particularmente adecuado para manejar grandes volúmenes de datos (71). El modelo de substitución más adecuado se seleccionará mediante el *Bayesian Information Criterion*, implementado en MEGA X (72). Además, herramientas como *IQ-tree* (73), entre otras, pueden emplearse para la construcción del árbol filogenético. Los resultados de este análisis se pueden visualizar en *iTOL* (74).

5.3. Composición de nucleótidos

Se analizarán diversos aspectos relacionados con el uso de codones, incluyendo la frecuencia de nucleótidos (A%, C%, U% y G%), las frecuencias de nucleótidos en la tercera posición (%A3, %C3, %U3 y %G3), el contenido de G+C (GC), el contenido de GC en la primera, segunda o tercera posición (GC1, GC2, GC3, respectivamente (75). Una herramienta muy útil que permitirá llevar a cabo este proceso de manera eficiente es *CodonW* (76).

5.4. Análisis de abundancia relativa de dinucleótidos

El análisis de abundancia relativa de dinucleótidos es una herramienta que evalúa la frecuencia con la que aparecen combinaciones específicas de dos nucleótidos en secuencias genómicas o transcriptómicas con el fin de identificar patrones en la composición del genoma. Se basa en la comparación entre la frecuencia esperada y la frecuencia observada (3,77). Este valor se puede calcular de acuerdo a la siguiente ecuación:

$$\rho_{xy} = \frac{f_{xy}}{f_x f_y}$$

Figura 16. Ecuación para calcular el análisis de abundancia relativa de dinucleótidos. Tomado de Wu W et al., 2021.

Donde:

- **fxy** es la frecuencia observada del dinucleótido XY.
- **fx** es la frecuencia del nucleótido X.
- **fy** es la frecuencia del nucleótido Y.

Un valor de **Pxy** indica que el dinucleótido XY aparece con la frecuencia esperada. Valores menores a 0.78 sugieren que el dinucleótido está subrepresentado, mientras que valores mayores a 1.23 indican que el dinucleótido está sobrerrepresentado (3,77). Estas frecuencias pueden ser calculadas mediante *CodonW* (76).

5.5. Índice de adaptación de codones (CAI)

Es una métrica utilizada para predecir el nivel de expresión génica, evaluar la adaptación de los genes virales a sus huéspedes y comparar el uso de codones entre diferentes organismos. Sus valores varían de 0 a 1, donde un CAI más alto indican una mayor adaptabilidad al huésped (3). Para calcular este índice de manera eficiente, una herramienta ampliamente utilizada es *CAIcal* (78,79).

5.6. Cálculo del uso relativo de codones sinónimos (RSCU)

El RSCU es una métrica que mide la frecuencia con la que un codón se utiliza para codificar un aminoácido y lo compara con la frecuencia esperada. Este valor se puede calcular de acuerdo a la siguiente ecuación:

$$\mathrm{RSCU} = \frac{gij}{\sum_{j}^{ni} gij} \cdot ni$$

Figura 17. Ecuación para calcular el RSCU. Tomado de Sharp PM et al., 1987.

Donde ***gij*** es la frecuencia observada del codón ***(j)*** que codifica el aminoácido ***(i)*** en la secuencia analizada. Es el número de veces que aparece un codón específico para ese aminoácido. En el denominador de la ecuación se calcula la suma total de las frecuencias observadas de todos los codones sinónimos ***(j)*** que codifican para el aminoácido ***(i)***. Esto da el número total de veces que aparece el aminoácido ***(i)*** en la secuencia, sin importar qué codón lo codifique. Y **ni** es el número de codones sinónimos disponibles para el aminoácido ***(i)***. Por ejemplo: Para Leucina, ni = 6 (porque hay 6 codones que la codifican). Cuando RSCU es <1 significa que la frecuencia de uso del codón es menor o >1 significa que la frecuencia de uso del codón es mayor. Cuando RSCU es = 1 significa que el codón no tiene preferencia (79). Los valores de RSCU se pueden calcular con el paquete *seqinr* de R (80).

5.7. PCA

El *PCA* ha sido utilizado por varios autores (3,62) en el estudio del *codon usage* en el virus del PRRS para reducir la dimensionalidad de los datos y encontrar relaciones entre las variables (los codones con su valor RSCU) y las muestras (las secuencias). Las secuencias fueron representadas como vectores de 59 dimensiones, excluyendo los codones UGG (triptofano), AUG (inicio) y los tres codones de parada, ya que estos no tienen sinónimos y no contribuyen al sesgo de uso de codones. Posteriormente, la dimensionalidad de los vectores se redujo a dos componentes principales, lo que permitió visualizar mejor las relaciones entre las secuencias. Se evidencia que los datos representados en estos estudios explican menos del 50% de la variabilidad de los datos en dos componentes principales. Esto es problemático

porque indica que la mayor parte de la variabilidad de los datos no está siendo capturada, lo que puede llevar a una representación incompleta.

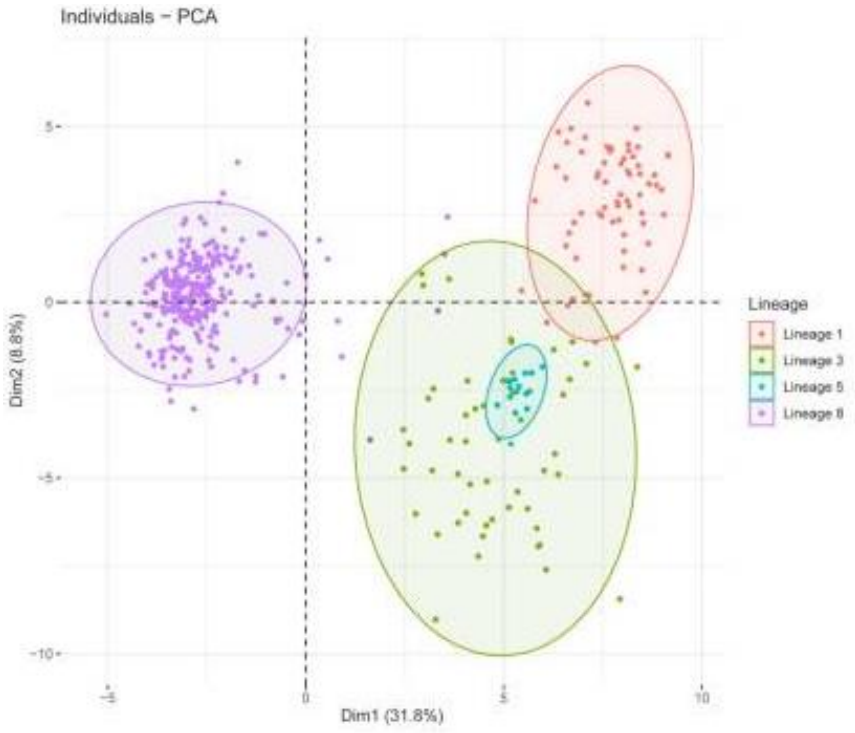

Figura 18. PCA en estudio del codon usage en el virus del PRRS. Tomado de Wu W et al., 2021.

En nuestro estudio el *PCA* podría ser una técnica útil; sin embargo, también se considera la posibilidad de evaluar otras técnicas de *machine learning* no supervisado como: *MDS*, *t-SNE, Isomap, K-Means,* entre otras. Se espera capturar la mayor cantidad de datos que expliquen la variabilidad del uso de codones, se considerará la representación gráfica en hasta tres dimensiones si es necesario.

La selección de la técnica se basará en las características del conjunto de datos analizado y en la comparación de las métricas de calidad utilizadas.

Tabla 3. Comparación de métricas de calidad entre técnicas de *machine learning* no supervisado

Técnica	Tipo	Objetivo principal	Métrica de calidad	Cómo se interpreta
PCA	Reducción de dimensionalidad lineal	Maximizar la varianza retenida en menos dimensiones	Varianza explicada (%)	Cuanto mayor, mejor; generalmente se busca retener al menos el 80-90%
MDS	Reducción de dimensionalidad basada en distancias	Preservar distancias originales en menos dimensiones	Estrés	Cuanto menor, mejor (idealmente < 0.1)
t-SNE	Reducción de dimensionalidad no lineal	Preservar estructuras locales de los datos	Divergencia de Kullback-Leibler	Cuanto menor, mejor; indica qué tan bien se mantienen relaciones locales
Isomap	Reducción de dimensionalidad basada en geodésicas	Preservar relaciones geométricas globales	Error de reconstrucción	Cuanto menor, mejor; mide la discrepancia entre distancias originales y proyectadas
K-Means	Agrupamiento	Minimizar la dispersión dentro de los grupos	Inercia	Cuanto menor, mejor; indica la coherencia interna de los clústeres

Fuente: Elaboración propia

El objetivo del algoritmo elegido será agrupar las secuencias de manera más eficiente, identificando patrones o relaciones subyacentes entre ellas que faciliten su análisis e interpretación.

Estas pruebas pueden implementarse y visualizarse fácilmente en R o Python, utilizando bibliotecas como *vegan, tsne, stats, FactoMineR, MASS* y *ggplot2* en R, o *scikit-learn, Matplotlib* y *Seaborn* en Python. Un ejemplo del código correspondiente se encuentra en el Anexo A.

5.8. Análisis con técnicas de *machine learning* supervisado

5.8.1. Creación de algoritmos basados en modelos tradicionales

Dado que no contamos con datos numéricos, modelos como la regresión lineal y la regresión logística no serían apropiados. En su lugar, se puede considerar la creación de nuevos algoritmos basados en modelos de *machine learning* supervisado, como *KNN*, *SVMs*, árboles de decisión, y *Random Forest*, los cuales permiten analizar patrones en datos categóricos o de estructura más compleja.

Los modelos supervisados elegidos se pueden evaluar con métricas como *Accuracy* y *Kappa*:

5.8.1.1. Accuracy

Mide el porcentaje de predicciones correctas sobre el total de muestras evaluadas (81).

Este valor se puede calcular de acuerdo a la siguiente ecuación:

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

Figura 19. Fórmula para calcular accuracy. Tomado de la página oficial de scikit-learn.

Donde:

- ***nsamples*** es el número total de muestras en el conjunto de datos.
- ***yi*** es el valor real de la muestra ***i***.
- ***y^i*** es la predicción del modelo para la muestra ***i***.
- ***1(y^i = yi)*** es una función indicadora que devuelve 1 si la predicción es correcta y 0 si es incorrecta.

La puntuación obtenida varia de 0 a 1, donde valores más altos indican un mejor rendimiento del modelo (81).

5.8.1.2. *Kappa de Cohen* (Coeficiente Kappa, κ)

Es una métrica que evalúa la concordancia entre las predicciones del modelo y los valores reales. Tiene en cuenta el efecto del azar (82,83).

Este valor se puede calcular de acuerdo a la siguiente ecuación:

$$\kappa = \frac{P_o - P_c}{1 - P_c}$$

Figura 20. Fórmula para calcular Kappa. Tomado de Sim J et al.,2005.

Donde:

- ***Po*** es el porcentaje de aciertos del modelo.
- ***Pc*** es el porcentaje de aciertos esperados al azar.

Tabla 4. Interpretación del valor de *Kappa*

Valor de *Kappa*	**Concordancia**
≤ 0	Sin concordancia o peor que el azar
0.00-0.20	Leve
0.21-0.40	Regular
0.41-0.60	Moderada
0.61-0.80	Sustancial
0.81-1.00	Casi perfecta

Adaptación de Landis JR, et al., 1977.

Los modelos creados pueden implementarse y visualizarse fácilmente en R o Python, utilizando bibliotecas como *e1071*, *caret*, *randomForest*, *tidymodels* y *ggplot2* en R, o *scikit-learn, Matplotlib* y *Seaborn* en Python. Estas librerías permiten calcular métricas de evaluación como *accuracy* y *kappa*, facilitando la comparación entre modelos para seleccionar aquel con el mejor desempeño según los criterios establecidos.

5.8.2. Algoritmo Boruta

Un estudio sobre el uso de codones en diferentes especies de bacterias (84) empleó el algoritmo Boruta, basado en técnicas de *Random Forest*. Este algoritmo utiliza características de sombra, que son versiones aleatorias de cada variable original, y las utiliza como referencia para evaluar la importancia de las variables originales. Posteriormente, a través de *Random Forest*, descarta las variables de menor relevancia y conserva aquellas más significativas sin perder información relevante (85).

En nuestro estudio, se puede considerar la implementación de este algoritmo y su evaluación mediante métricas como *accuracy*, lo que permitiría determinar su efectividad en la selección de codones relevantes.

5.9. Análisis con técnicas de *deep learning*

Un estudio describe una herramienta muy útil llamada *mBART*, un modelo de inteligencia artificial basado en la arquitectura de *Transformers* (19). Este modelo ha sido utilizado para estudiar el uso de codones en eucariotas como: *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe* y las bacterias *Escherichia coli* y *Bacillus subtilis*.

En nuestro estudio, se podría considerar el uso de herramientas como *mBART* o la creación de nuevas redes neuronales capaces de capturar patrones más complejos, extraer relaciones entre los datos, manejar grandes volúmenes de información y generar representaciones útiles automáticamente. Sin embargo, es importante considerar que, si un modelo más simple ya proporciona buenos resultados, el uso de redes neuronales podría ser innecesario. Por esta razón, es fundamental evaluar ambas alternativas antes de tomar una decisión.

Las redes neuronales elegidas se pueden evaluar con métricas como *Accuracy* y *Validation Loss:*

5.9.1. Validation Loss

Es una métrica que mide el error del modelo al realizar predicciones en los datos de validación. Su análisis, junto con métricas como *accuracy* permite evaluar el sobreajuste (*Overfitting)*, que ocurre cuando un algoritmo aprende en exceso los detalles y el ruido del conjunto de entrenamiento interpretándolos erróneamente como patrones significativos (28,86). Además, la métrica de *validation loss* es útil para determinar la época óptima en el entrenamiento, evitando que el modelo continúe aprendiendo de manera ineficiente (86).

Deep learning pueden implementarse y visualizarse fácilmente en R o Python, utilizando bibliotecas como *Keras*, *TensorFlow*, *torch*, *MXNet* y *ggplot2* en R, o *TensorFlow, Keras, PyTorch, Matplotlib* y *Seaborn* en Python. Estas librerías permiten calcular métricas de evaluación como *accuracy* y *validation los*s para evaluar el rendimiento del modelo.

5.10.Análisis estadístico

En el estudio de Wu W, et al. los valores de CAI en diferentes linajes fueron analizados mediante la prueba de Kruskal-Wallis, ya que los datos no cumplían con los supuestos de normalidad. El objetivo fue comparar si existían diferencias significativas entre los grupos, utilizando los siguientes criterios:

- $p \leq 0{,}001$: Relación extremadamente significativa.
- 0,001 - 0,01: Relación altamente significativa.
- 0,01 - 0,05: Relación significativa.
- $p > 0{,}05$: No significativa.

Estas relaciones se pudieron visualizar en diagramas de caja generados en R (3).

Por otro lado, en el estudio de Liu YS, et al. se empleó el análisis de correlación por rangos de Spearman para determinar si la composición del ARN influye en el uso de codones sinónimos (62).

Para nuestro estudio, la elección de la prueba estadística dependerá de la distribución de los datos. Si los datos no están normalizados, podríamos aplicar Kruskal-Wallis o Spearman, como en los estudios mencionados. En cambio, si los datos cumplen con la normalidad, podríamos utilizar ANOVA para comparar los grupos. Estas pruebas pueden implementarse y visualizarse fácilmente en R.

5.11.Cronograma

Tabla 5. Cronograma de actividades

Actividad	Semana 1-2	Semana 3-4	Semana 5-6	Semana 7-8	Semana 9-10	Semana 11-12	Semana 13-16
Recopilación de datos	X						
Análisis filogenético	X						
Cálculo de composición de nucleótidos, análisis de abundancia relativa de dinucleótidos y cálculo del uso relativo de codones sinónimos (RSCU)		X					
Evaluación de métodos de *machine learning* no supervisado			X				
Evaluación de métodos de *machine learning* supervisado			X				
Evaluación de técnicas de *deep learning*				X			
Análisis estadístico					X		
Obtención y análisis de resultados finales						X	
Redacción y publicación de los hallazgos							X

Fuente: Elaboración propia

6. Financiación

Para la realización de este trabajo, los recursos necesarios son accesibles en comparación con los estudios experimentales en laboratorio. Los datos genómicos están disponibles en bases de datos públicas como NCBI y pueden descargarse sin costo. En el caso de *machine learning,* es posible trabajar con computadores de gama media o utilizar servidores en la nube, como Google colab. Por otro lado, para *deep learning* lo más recomendable es usar GPUs de alto rendimiento que optimicen el procesamiento y el entrenamiento de los modelos (87).

7. Limitaciones para su implementación

Aunque la información se extraerá de bases de datos, muchas secuencias pueden estar incompletas o contener errores de anotación, por lo que es fundamental seleccionar aquellas que cumplan con los criterios de aceptabilidad para garantizar resultados confiables.

Otra limitación es el requerimiento de recursos computacionales. Si se utilizan técnicas de *deep learning*, se necesitarán GPUs más potentes para entrenar modelos grandes.

8. Resultados esperados

Obtener un conjunto de secuencias completas de diferentes cepas del virus a partir de distintas bases de datos como Genbank, asegurando que no presenten eventos de recombinación. Estas secuencias serán utilizadas en la construcción de un árbol filogenético. Posteriormente, se calcularán varias métricas como: Composición de nucleótidos, CAI, análisis de abundancia relativa de dinucleótidos y RSCU. Se espera identificar patrones en la preferencia del uso de codones, determinar los nucleótidos más frecuentes y evaluar la conservación de posiciones entre cepas. Además, se identificarán nucleótidos correlacionados entre sí y se cuantificará el sesgo en el uso de codones en función de la cepa. Para ello, se integrará información fenotípica de las bases de datos, como el grado de patogenicidad, con el fin de establecer correlaciones entre uso de codones y la virulencia.

A continuación, se aplicarán técnicas de reducción de dimensionalidad, como PCA u otros algoritmos de *machine learning* no supervisado, utilizando la métrica de RSCU y las secuencias virales. La técnica con mejores métricas de calidad será seleccionada para visualizar similitudes y diferencias en el uso de codones entre genes y cepas, minimizando la pérdida de información relevante.

Como alternativa, se explorará el uso de *machine learning* supervisado para minimizar la pérdida de datos en la reducción de dimensionalidad. Se evaluará la creación de un nuevo algoritmo basado en modelos de *machine learning* supervisado o la implementación de técnicas como el algoritmo Boruta. La selección del mejor modelo se basará en métricas como *kappa* y *accuracy*. Asimismo, dependiendo de la complejidad del análisis, se explorará el uso de técnicas de *deep learning*, incluyendo modelos como *mBART* o el desarrollo de nuevas redes neuronales que serán evaluadas con las métricas de *validation loss* y *accuracy*. Se espera identificar de manera más efectiva los codones preferenciales en distintos genes y cepas.

Finalmente, se realizará un análisis estadístico comparativo, evaluando valores como la abundancia relativa de dinucleótidos, el CAI y la composición general de nucleótidos (A%, U%, C%, G%) en comparación con la composición en la tercera posición del codón (A_3%, U_3%, C_3%, G_3%) y métricas como (C+G) % frente a (C_3+G_3) %. Dependiendo de la normalidad de los datos, se emplearán diferentes pruebas estadísticas, como Kruskal-Wallis, Spearman o ANOVA, para identificar diferencias significativas en la optimización del uso de codones.

9. Conclusiones

Existen diversas herramientas de inteligencia artificial que permiten analizar el uso de codones en el genoma completo del PRRSV. Dentro de ellas se puede considerar herramientas de *machine learning* no supervisado como el PCA, herramientas de *machine learning* supevisado, como el algoritmo de Boruta, y redes neuronales como *mBART.*

Dado que analizaríamos el genoma completo, es probable que obtengamos grandes volúmenes de datos. Por ello, es fundamental elegir una herramienta que sea capaz de procesar eficientemente la mayor cantidad de información sin pérdida de datos importantes, garantizando resultados confiables y precisos. Además, el uso de técnicas basadas en inteligencia artificial facilita la identificación de patrones en la selección de codones, lo que puede proporcionar una visión más profunda de la dinámica evolutiva del PRRSV.

Finalmente, los datos obtenidos en este estudio se podrán utilizar para contribuir a una mejor comprensión de los mecanismos de replicación y adaptación del PRRSV en su huésped porcino. Esto permitirá establecer una correlación entre el uso de codones y virulencia de las cepas del virus lo cual es crucial para el desarrollo de estrategias de control y prevención.

Referencias bibliográficas

1. Lunney JK, Fang Y, Ladinig A, Chen N, Li Y, Rowland B, et al. Porcine reproductive and respiratory syndrome virus (PRRSV): Pathogenesis and interaction with the immune system. Annu Rev Anim Biosci [Internet]. 2015;4:129–54. Disponible en: http://dx.doi.org/10.1146/annurev-animal-022114-111025.

2. Sun Q, Xu H, An T, Cai X, Tian Z, Zhang H. Recent progress in studies of porcine reproductive and respiratory syndrome virus 1 in China. Viruses [Internet]. 2023;15(7). Disponible en: http://dx.doi.org/10.3390/v15071528.

3. Wu W, Ge X, Zhang Y, Han J, Guo X, Zhou L, et al. Evolutionary patterns of Codon usage in major lineages of porcine reproductive and respiratory syndrome virus in China. Viruses [Internet]. 2021;13(6). Disponible en: http://dx.doi.org/10.3390/v13061044.

4. Kuhn JH, Lauck M, Bailey AL, Shchetinin AM, Vishnevskaya TV, Bào Y, et al. Reorganization and expansion of the nidoviral family Arteriviridae. Arch Virol [Internet]. 2016 [citado el 17 de septiembre de 2024];161(3):755–68. Disponible en: http://dx.doi.org/10.1007/s00705-015-2672-z.

5. Ruedas-Torres I, Rodríguez-Gómez IM, Sánchez-Carvajal JM, Larenas-Muñoz F, Pallarés FJ, Carrasco L, et al. The jigsaw of PRRSV virulence. Vet Microbiol [Internet]. 2021;260(109168):109168. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S0378113521001917.

6. Neumann EJ, Kliebenstein JB, Johnson CD, Mabry JW, Bush EJ, Seitzinger AH, et al. Assessment of the economic impact of porcine reproductive and respiratory syndrome on swine production in the United States. J Am Vet Med Assoc [Internet]. 2005;227(3):385–92. Disponible en: http://dx.doi.org/10.2460/javma.2005.227.385.

7. Joiret M, Leclercq M, Lambrechts G, Rapino F, Close P, Louppe G, et al. Cracking the genetic code with neural networks. Front Artif Intell [Internet]. 2023;6. Disponible en: http://dx.doi.org/10.3389/frai.2023.1128153.

8. Crick F. Central dogma of molecular biology. Nature [Internet]. 1970 [citado el 19 de noviembre de 2024];227(5258):561–3. Disponible en: https://www.nature.com/articles/227561a0.

9. Koonin EV, Dolja VV. A virocentric perspective on the evolution of life. Curr Opin Virol [Internet]. 2013;3(5):546–57. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S1879625713001028.

10. Shors T. Virus: estudio molecular con orientación clínica. Ed. Médica Panamericana; 2009.

11. Watson JD, Baker TA, Bell SP, Gann A, Levine M, Losick R. Biología molecular del gen. Ed. Médica Panamericana; 2005.

12. Done SH, Paton DJ, White ME. Porcine reproductive and respiratory syndrome (PRRS): a review, with emphasis on pathological, virological and diagnostic aspects. Br Vet J [Internet]. 1996;152(2):153–74. Disponible en: http://dx.doi.org/10.1016/s0007-1935(96)80071-6.

13. Orosco F. From nature's pharmacy to swine health: Harnessing natural compounds against prrsv infection. Slov Vet Zb [Internet]. 2024 [citado el 19 de noviembre de 2024];61(1):9–28. Disponible en: https://www.slovetres.si/index.php/SVR/article/view/1789.

14. Sun L, Li Y, Liu R, Wang X, Gao F, Lin T, et al. Porcine reproductive and respiratory syndrome virus ORF5a protein is essential for virus viability. Virus Res [Internet]. 2013;171(1):178–85. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S0168170212004509.

15. Quax TEF, Claassens NJ, Söll D, van der Oost J. Codon bias as a means to fine-tune gene expression. Mol Cell [Internet]. 2015;59(2):149–61. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S1097276515004025.

16. Sharp PM, Li WH. An evolutionary perspective on synonymous codon usage in unicellular organisms. J Mol Evol [Internet]. 1986;24(1–2):28–38. Disponible en: http://dx.doi.org/10.1007/BF02099948.

17. Parvathy ST, Udayasuriyan V, Bhadana V. Codon usage bias. Mol Biol Rep [Internet]. 2022;49(1):539–65. Disponible en: http://dx.doi.org/10.1007/s11033-021-06749-4.

18. Ikemura T. Codon usage and tRNA content in unicellular and multicellular organisms. Mol Biol Evol [Internet]. 1985;2(1):13–34. Disponible en: http://dx.doi.org/10.1093/oxfordjournals.molbev.a040335.

19. Sidi T, Bahiri-Elitzur S, Tuller T, Kolodny R. Predicting gene sequences with AI to study codon usage patterns [Internet]. bioRxiv. 2024. p. 2024.02.11.579798. Disponible en: http://biorxiv.org/content/early/2024/02/12/2024.02.11.579798.abstract.

20. Wu H, Bao Z, Mou C, Chen Z, Zhao J. Comprehensive analysis of Codon usage on porcine Astrovirus. Viruses [Internet]. 2020 [citado el 20 de octubre de 2024];12(9):991. Disponible en: https://www.mdpi.com/1999-4915/12/9/991.

21. Pan S, Mou C, Wu H, Chen Z. Phylogenetic and codon usage analysis of atypical porcine pestivirus (APPV). Virulence [Internet]. 2020;11(1):916–26. Disponible en: http://dx.doi.org/10.1080/21505594.2020.1790282.

22. Li G, Wang R, Zhang C, Wang S, He W, Zhang J, et al. Genetic and evolutionary analysis of emerging H3N2 canine influenza virus. Emerg Microbes Infect [Internet]. 2018;7(1):1–15. Disponible en: http://dx.doi.org/10.1038/s41426-018-0079-0.

23. Nei M, Kumar S. Molecular evolution and phylogenetics. Londres, Inglaterra: Oxford University Press; 2000.

24. Higgs PG, Attwood TK. Bioinformatics and Molecular Evolution. Hoboken, NJ, Estados Unidos de América: Wiley-Blackwell; 2005.

25. Choi RY, Coyner AS, Kalpathy-Cramer J, Chiang MF, Campbell JP. Introduction to machine learning, neural networks, and deep learning. Transl Vis Sci Technol [Internet]. 2020; 9(2):14. Disponible en: http://dx.doi.org/10.1167/tvst.9.2.14, 51.

26. Rouhiainen L. Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro. Alienta Editorial; 2018.

27. Shortliffe E. Computer-Based Medical Consultations: MYCIN. Elsevier; 2012.

28. Handelman GS, Kok HK, Chandra RV, Razavi AH, Lee MJ, Asadi H. eDoctor: machine learning and the future of medicine. J Intern Med [Internet]. 2018; 284(6):603–19. Disponible en: http://dx.doi.org/10.1111/joim.12822.

29. Breiman L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). Stat Sci [Internet]. 2001 [citado el 15 de enero de 2025];16(3):199–231. Disponible en: https://projecteuclid.org/journals/statistical-science/volume-16/issue-3/Statistical-Modeling--The-Two-Cultures-with-comments-and-a/10.1214/ss/1009213726.full.

30. Theodosiou AA, Read RC. Artificial intelligence, machine learning and deep learning: Potential resources for the infection clinician. J Infect [Internet]. 2023; 87(4):287–94. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S0163445323003791.

31. Labs S. What are the three types of artificial intelligence and their specific features? [Internet]. Skyld.io. [citado el 15 de enero de 2025]. Disponible en: https://skyld.io/What-Are-the-Three-Types-of-Artificial-Intelligence-Learning-and-Their-Specific-Features.

32. Géron A. Hands-on machine learning with Scikit-Learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. Heidelberg, Alemania: O'Reilly; 2019.

33. Principal component analysis. New York: Springer-Verlag; 2002.

34. Mina Nashed. Population genetics 3D principal component analysis (PCA) [Internet]. Biorender.com. [citado el 23 de enero de 2025]. Disponible en: https://www.biorender.com/template/population-genetics-3d-principal-component-analysis-pca.

35. Faraway JJ. Linear models with R, Second Edition. 2a ed. Boca Ratón, FL, Estados Unidos de América: CRC Press; 2014.

36. Principal Component Analysis (PCA) on Iris dataset [Internet]. scikit-learn. [citado el 17 de enero de 2025]. Disponible en: https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_iris.html.

37. Mugavin ME. Multidimensional scaling: a brief overview. Nurs Res [Internet]. 2008;57(1):64–8. Disponible en: http://dx.doi.org/10.1097/01.NNR.0000280659.88760.7c.

38. Kruskal JB. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika [Internet]. 1964;29(1):1–27. Disponible en: http://dx.doi.org/10.1007/bf02289565.

39. Sturrock K, Rocha J. A multidimensional scaling stress evaluation table. Field Methods [Internet]. 2000;12(1):49–60. Disponible en: http://dx.doi.org/10.1177/1525822x0001200104.

40. Maaten L, Hinton GE. Visualizing Data using t-SNE. Journal of Machine Learning Research [Internet]. 2008 [citado el 23 de enero de 2025];9(86):2579–605. Disponible en: https://jmlr.org/papers/v9/vandermaaten08a.html.

41. Tenenbaum JB, Silva V de, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science [Internet]. 2000;290(5500):2319–23. Disponible en: http://dx.doi.org/10.1126/science.290.5500.2319.

42. Bishop CM. Pattern recognition and machine learning. Springer Verlag; 2006.

43. Clustering [Internet]. scikit-learn. [citado el 19 de febrero de 2025]. Disponible en: https://scikit-learn.org/stable/modules/clustering.html.

44. Nelli F. Python data analytics: With pandas, NumPy, and matplotlib. Berkeley, CA: Apress; 2023.

45. James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning: With applications in R. New York, NY: Springer US; 2021.

46. Schneider A, Hommel G, Blettner M. Linear regression analysis: part 14 of a series on evaluation of scientific publications. Dtsch Arztebl Int [Internet]. 2010;107(44):776–82. Disponible en: http://dx.doi.org/10.3238/arztebl.2010.0776.

47. Stoltzfus JC. Logistic regression: A brief primer. Acad Emerg Med [Internet]. 2011;18(10):1099–104. Disponible en: http://dx.doi.org/10.1111/j.1553-2712.2011.01185.x.

48. Wang QQ, Yu SC, Qi X, Hu YH, Zheng WJ, Shi JX, et al. Overview of logistic regression model analysis and application. Zhonghua Yu Fang Yi Xue Za Zhi [Internet]. 2019;53(9):955–60. Disponible en: http://dx.doi.org/10.3760/cma.j.issn.0253-9624.2019.09.018.

49. Hosmer DW Jr, Lemeshow S, Sturdivant RX. Applied logistic regression. Wiley; 2013.

50. Núñez E, Steyerberg EW, Núñez J. Estrategias para la elaboración de modelos estadísticos de regresión. Rev Esp Cardiol [Internet]. 2011;64(6):501–7. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S0300893211003502.

51. Hallee L, Khomtchouk BB. Machine learning classifiers predict key genomic and evolutionary traits across the kingdoms of life. Sci Rep [Internet]. 2023 [citado el 11 de febrero de 2025];13(1):1–14. Disponible en: https://www.nature.com/articles/s41598-023-28965-7.

52. Cortes C, Vapnik V. Support-vector networks. Mach Learn [Internet]. 1995;20(3):273–97. Disponible en: http://dx.doi.org/10.1007/bf00994018.

53. Meyer, D. and Wien, F.t. (2015) Support Vector Machines. The Interface to Libsvm in Package, e1071. - references - scientific research publishing [Internet]. Scirp.org. [citado el 13 de febrero de 2025].

54. Rokach L, Maimon O. Data mining with decision trees: Theory and applications. World scientific; 2014.

55. Haykin SS. Neural Networks and Learning Machines. Upper Saddle River, NJ, Estados Unidos de América: Pearson; 2009.

56. Vaswani A, Shazeer NM, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. Neural Inf Process Syst [Internet]. 2017 [citado el 2 de febrero de 2025];5998–6008. Disponible en: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

57. Tian J, Yan Y, Yue Q, Liu X, Chu X, Wu N, et al. Predicting synonymous codon usage and optimizing the heterologous gene for expression in E. coli. Sci Rep [Internet]. 2017 [citado el 21 de noviembre de 2024];7(1):1–9. Disponible en: https://www.nature.com/articles/s41598-017-10546-0.

58. Grote A, Hiller K, Scheer M, Munch R, Nortemann B, Hempel DC, et al. JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. Nucleic Acids Res [Internet]. 2005 [citado el 21 de noviembre de 2024];33(Web Server):W526–31. Disponible en: https://academic.oup.com/nar/article/33/suppl_2/W526/2505472 noviembre de 2024];7(1):1–9. Disponible en: https://www.nature.com/articles/s41598-017-10546-0.

59. Villalobos A, Ness JE, Gustafsson C, Minshull J, Govindarajan S. Gene Designer: a synthetic biology tool for constructing artificial DNA segments. BMC Bioinformatics [Internet]. 2006;7(1). Disponible en: http://dx.doi.org/10.1186/1471-2105-7-285.

60. Lorimer D, Raymond A, Walchli J, Mixon M, Barrow A, Wallace E, et al. Gene Composer: database software for protein construct design, codon engineering, and gene synthesis. BMC Biotechnol [Internet]. 2009;9(1). Disponible en: http://dx.doi.org/10.1186/1472-6750-9-36.

61. Puigbo P, Guzman E, Romeu A, Garcia-Vallve S. OPTIMIZER: a web server for optimizing the codon usage of DNA sequences. Nucleic Acids Res [Internet]. 2007 [citado el 21 de noviembre de 2024];35(Web Server):W126–31. Disponible en: https://academic.oup.com/nar/article/35/suppl_2/W126/2920747.

62. Liu Y-S, Zhou J-H, Chen H-T, Ma L-N, Ding Y-Z, Wang M, et al. Analysis of synonymous codon usage in porcine reproductive and respiratory syndrome virus. Infect Genet Evol [Internet]. 2010;10(6):797–803. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S1567134810001085.

63. Holck JT, Polson DD. Financial impact of PRRS. In:Zimmerman JJ, Yoon K-J, eds. The porcine reproductive and respiratory syndrome compendium. 2nd ed. Des Moines: National Pork Board, 2003;51–58.

64. Pintó RM, Burns CC, Moratorio G. Editorial: Codon usage and dinucleotide composition of virus genomes: From the virus-host interaction to the development of vaccines. Front Microbiol [Internet]. 2021;12. Disponible en: http://dx.doi.org/10.3389/fmicb.2021.791750.

65. Robinson SR, Abrahante JE, Johnson CR, Murtaugh MP. Purifying selection in porcine reproductive and respiratory syndrome virus ORF5a protein influences variation in envelope glycoprotein 5 glycosylation. Infect Genet Evol [Internet]. 2013;20:362–8. Disponible en: http://dx.doi.org/10.1016/j.meegid.2013.09.022.

66. Larkin MA, Blackshields G, Brown NP, Chenna R, McGettigan PA, McWilliam H, et al. Clustal W and Clustal X version 2.0. Bioinformatics [Internet]. 2007 [citado el 28 de noviembre de 2024];23(21):2947–8. Disponible en: https://academic.oup.com/bioinformatics/article/23/21/2947/371686?login=false.

67. Sievers F, Higgins DG. Clustal omega. Curr Protoc Bioinformatics [Internet]. 2014;48:3.13.1-3.13.16. Disponible en: http://dx.doi.org/10.1002/0471250953.bi0313s48.

68. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol [Internet]. 2013;30(4):772–80. Disponible en: http://dx.doi.org/10.1093/molbev/mst010.

69. Martin DP, Varsani A, Roumagnac P, Botha G, Maslamoney S, Schwab T, et al. RDP5: a computer program for analyzing recombination in, and removing signals of recombination from, nucleotide sequence datasets. Virus Evol [Internet]. 2020;7(1):veaa087. Disponible en: http://dx.doi.org/10.1093/ve/veaa087.

70. Samson S, Lord É, Makarenkov V. SimPlot++: a Python application for representing sequence similarity and detecting recombination. Bioinformatics [Internet]. 2022;38(11):3118–20. Disponible en: http://dx.doi.org/10.1093/bioinformatics/btac287.

71. Yang S, De Angelis D. Maximum likelihood. En: Methods in Molecular Biology. Totowa, NJ: Humana Press; 2013. p. 581–95.

72. Kumar S, Stecher G, Li M, Knyaz C, Tamura K. MEGA X: Molecular Evolutionary Genetics Analysis across computing platforms. Mol Biol Evol [Internet]. 2018;35(6):1547–9. Disponible en: http://dx.doi.org/10.1093/molbev/msy096.

73. Nguyen L-T, Schmidt HA, von Haeseler A, Minh BQ. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol Biol Evol

[Internet]. 2014;32(1):268–74. Disponible en: http://dx.doi.org/10.1093/molbev/msu300.

74. Letunic I, Bork P. Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. Nucleic Acids Res [Internet]. 2021;49(W1):W293–6. Disponible en: http://dx.doi.org/10.1093/nar/gkab301.

75. Zhao Z-Y, Yu D, Ji C-M, Zheng Q, Huang Y-W, Wang B. Comparative analysis of newly identified rodent arteriviruses and porcine reproductive and respiratory syndrome virus to characterize their evolutionary relationships. Front Vet Sci [Internet]. 2023;10. Disponible en: http://dx.doi.org/10.3389/fvets.2023.1174031.

76. Sharp P. Correspondence Analysis of Codon Usage [Internet]. Sourceforge.net. [citado el 30 de noviembre de 2024]. Disponible en: https://codonw.sourceforge.net.

77. Kariin S, Burge C. Dinucleotide relative abundance extremes: a genomic signature. Trends Genet [Internet]. 1995;11(7):283–90. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S0168952500890769.

78. Puigbò P, Bravo IG, Garcia-Vallve S. CAIcal: a combined set of tools to assess codon usage adaptation. Biol Direct [Internet]. 2008;3:38. Disponible en: http://dx.doi.org/10.1186/1745-6150-3-38.

79. Sharp PM, Li WH. The codon Adaptation Index--a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Res [Internet]. 1987;15(3):1281–95. Disponible en: http://dx.doi.org/10.1093/nar/15.3.1281.

80. Seqinr [Internet]. R-project.org. [citado el 28 de enero de 2025]. Disponible en: https://seqinr.r-forge.r-project.org.

81. Metrics and scoring: quantifying the quality of predictions [Internet]. scikit-learn. [citado el 19 de febrero de 2025]. Disponible en: https://scikit-learn.org/stable/modules/model_evaluation.html.

82. Sim J, Wright CC. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. Phys Ther. 2005;85(3):257–68.

83. Landis JR, Koch GG. The measurement of observer agreement for categorical data. Biometrics. 1977;33(1):159–74.

84. Sen P, Kurmi A, Ray SK, Satapathy SS. Machine learning approach identifies prominent codons from different degenerate groups influencing gene expression in bacteria. Genes Cells [Internet]. 2022;27(10):591–601. Disponible en: http://dx.doi.org/10.1111/gtc.12977.

85. Kursa, M. B., Jankowski, A., & Rudnicki, W. R. Boruta—A system for feature selection.Fundamenta Informaticae[Internet]. 2010;(101):271–285.

86. Chollet F. Deep learning with Python [Internet]. Manning Publications. 2017. [citado el 20 de febrero de 2025]. Disponible en: https://www.manning.com/books/deep-learning-with-python.

87. Lucas WR, Leon YC, Samuel WR, Karthik R, Aravind K, Ruining D, et al. Exploring shared memory architectures for end-to-end gigapixel deep learning [Internet]. arXiv [cs.CV]. 2023. Disponible en: http://arxiv.org/abs/2304.12149.

Anexo A. Aplicación de tres métodos de aprendizaje no supervisado en R

Repositorio en el que se aplican tres métodos de aprendizaje no supervisado en R, utilizando el conjunto de datos ***Iris***:

https://github.com/Liliana223/Machine_learning_aprendizaje_no_supervisado/tree/main