Modelo de clasificación de péptidos anticancerígenos utilizando herramientas de inteligencia artificial y bioinformática

Brayan Arturo Navarro Sinuco y Ivan Andres Mendoza Oñate

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Nydia Paola Rondón Villarreal

Doctora en Ingeniería, Área Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2026

Agradecimientos

Los autores expresan su gratitud a la Universidad Industrial de Santander y a la Escuela de Ingeniería de Sistemas e Informática por el entorno académico y los recursos institucionales que hicieron posible este trabajo.

A la doctora Nydia Paola Rondón Villarreal, directora de este trabajo de grado, por su orientación rigurosa, sus observaciones precisas y su disposición permanente a lo largo de todas las etapas del proyecto.

A nuestras familias, por el apoyo incondicional durante el proceso de formación y especialmente durante el desarrollo de este trabajo.

Tabla de Contenido

Pág.
Introducción 10
1. Planteamiento y Justificación del Problema 11
2. Objetivos 15
2.1 Objetivo General 15
2.2 Objetivos Específicos 15
3. Marco de Referencia 16
3.1 Marco Conceptual 16
3.1.1 Cáncer Como Problema de Salud Pública 16
3.1.2 Péptidos 17
3.1.3 Péptidos Anticancerígenos (ACPs) 19
3.1.4 Bases de Datos Bioinformáticas 21
3.1.5 Bioinformática e Inteligencia Artificial en el Descubrimiento de Fármacos 22
3.1.6 Procesamiento de Lenguaje Natural Aplicado a Secuencias 24
3.1.6.1 ESM-2 (Evolutionary Scale Modeling 2). 25
3.1.6.2 ProtBERT 26
3.1.7 Descriptores Fisicoquímicos 27
3.1.8 Clasificadores de Aprendizaje Automático 28
3.1.9 Validación Cruzada Anidada 30
3.1.10 Desbalance de Clases y SMOTE 31
3.1.11 Herramientas Computacionales en Python 33
3.1.12 Evaluación y Métricas de Modelos Predictivos 34
3.1.13 Consideraciones Éticas y Legales 35
3.2 Estado del Arte 36
4. Metodología 42
4.1 Fase 1 — Construcción del Modelo Predictivo con Descriptores Fisicoquímicos 43
4.1.1 Preprocesamiento y Cálculo de Descriptores 44
4.1.2 Entrenamiento y Evaluación 44
4.2 Fase 2 — Construcción del Modelo Predictivo con Embeddings de ESM-2 45
4.2.1 Generación de Embeddings 45
4.2.2 Entrenamiento y Evaluación 46
4.3 Fase 3 — Construcción del Modelo Predictivo con Embeddings de ProtBERT 48

4.3.1 Generación de Embeddings 48
4.4 Fase 4 — Construcción del Modelo Multimodal 49
4.4.1 Construcción del Vector Multimodal 49
4.4.2 Entrenamiento y Evaluación 49
4.5 Fase 5 — Desarrollo del Paquete de Python 50
4.5.1 Selección e Integración del Modelo Final 50
4.5.2 Desarrollo del Paquete 50
4.5.3 Documentación 51
4.6 Alcance y Nivel de Madurez Tecnológica 52
5. Resultados 53
5.1 Caracterización del Conjunto de Datos de Referencia 53
5.1.1 Distribución de Longitudes de Secuencia 53
5.1.2 Composición Aminoacídica y Propiedades Fisicoquímicas 54
5.2 Modelos Basados en Descriptores Fisicoquímicos 56
5.3 Modelos Basados en Embeddings de ESM-2 56
5.4 Modelos Basados en Embeddings de ProtBERT 58
5.5 Análisis de Ablación: Capacidad del Modelo y Estrategia de Pooling 59
5.6 Modelo Multimodal 61
5.7 Comparación Global y Selección del Modelo Final 64
5.8 Viabilidad Computacional 66
6. Discusión 67
6.1 Sobre el Efecto del Protocolo de Evaluación 67
6.2 Sobre el Desempeño Comparativo de los Modelos de Lenguaje Proteico 69
6.3 Sobre el Desempeño de los Clasificadores Lineales 70
6.4 Sobre el Papel del Desbalance de Clases y las Técnicas de Remuestreo 70
6.5 Sobre la Integración Multimodal 71
6.6 Sobre la Robustez de los Resultados 73
7. Limitaciones 75
8. Conclusiones 77
9. Trabajo Futuro 80
Referencias Bibliográficas 81
Apéndices 87

Lista de Tablas

Pág.

Tabla 1. Estado del arte. 37
Tabla 2. Distribución de longitudes de secuencia (residuos) por subconjunto. 53
Tabla 3. Composición aminoacídica diferencial entre ACPs y no-ACPs. 54
Tabla 4. Propiedades fisicoquímicas globales por clase (media ± desviación típica). 55
Tabla 5. Descriptores fisicoquímicos (577 dimensiones) sobre el conjunto de prueba independiente de mACPpred 2.0. 56
Tabla 6. ESM-2 8M sobre el conjunto de prueba independiente de mACPpred 2.0. 57
Tabla 7. ProtBERT (prot_bert_bfd) sobre el conjunto de prueba independiente. 58
Tabla 8. Análisis de ablación sobre el conjunto de prueba independiente. 59
Tabla 9. Evolución metodológica de la configuración basada en ESM-2 8M. 59
Tabla 10. Modelo multimodal por concatenación directa (897 dimensiones) sobre el conjunto de prueba independiente. 62
Tabla 11. Estrategia de fusión tardía (w = 0.5) sobre el conjunto de prueba independiente. 63
Tabla 12. Comparación global sobre el conjunto de prueba independiente de mACPpred 2.0. 64
Tabla 13. Herramientas de referencia con AUC-ROC publicado sobre el conjunto de prueba independiente. 65
Tabla 14. Requisitos computacionales de los modelos de lenguaje evaluados. 67
Tabla 15. Diferencias entre la vía de evaluación descartada y el protocolo adoptado. 68
Tabla 16. AUC-ROC por rango de longitud sobre el conjunto de prueba independiente. 74
Tabla A1. Opciones de la interfaz de línea de comandos del paquete acp-classifier. 88
Tabla A2. Rendimiento del paquete acp-classifier sobre un lote de 100 péptidos. 90
Tabla B1. Descriptores fisicoquímicos sobre el conjunto de prueba independiente: resultados completos. 92
Tabla B2. Hiperparámetros seleccionados para los modelos finales de cada representación. 92
Tabla B3. Rejillas de hiperparámetros exploradas por cada clasificador. 93
Tabla C1. Composición del conjunto de referencia empleado, verificada sobre los archivos. 94
Tabla C2. Bibliotecas empleadas y función de cada una. 95

Lista de Figuras

Pág.

Figura 1. Estructura general del trabajo en cinco fases. 43

Figura 2. Flujo del proceso de clasificación de péptidos anticancerígenos mediante descriptores fisicoquímicos. 45

Figura 3. Flujo del proceso de clasificación de péptidos anticancerígenos mediante representaciones contextuales (embeddings). 47

Figura 4. Arquitectura del paquete acp-classifier para la clasificación de péptidos anticancerígenos desde línea de comandos. 51

Lista de Apéndices

Apéndice A. Manual del Paquete acp-classifier 87
Apéndice B. Tablas Extendidas de Resultados 91
Apéndice C. Reproducibilidad y Acceso al Código 93

Resumen

Título: Modelo de clasificación de péptidos anticancerígenos utilizando herramientas de inteligencia artificial y bioinformática *

Autor: Brayan Arturo Navarro Sinuco y Ivan Andres Mendoza Oñate **

Palabras Clave: péptidos anticancerígenos, aprendizaje automático, modelos de lenguaje proteico, ESM-2, bioinformática, clasificación de secuencias.

Descripción:

Los péptidos anticancerígenos (ACPs) constituyen una alternativa terapéutica prometedora frente a la quimioterapia convencional, por su selectividad hacia las células tumorales y su menor toxicidad sistémica. Su identificación experimental resulta inviable a gran escala, lo que motiva el desarrollo de herramientas de cribado computacional.

Este trabajo evaluó de forma sistemática tres representaciones numéricas de secuencias peptídicas: descriptores fisicoquímicos (577 características), representaciones contextuales del modelo de lenguaje proteico ESM-2 en sus variantes de 8 y 150 millones de parámetros, y representaciones de ProtBERT. Sobre cada una se optimizaron siete clasificadores mediante validación cruzada anidada de cinco pliegues externos y cinco internos, con el conjunto de referencia de mACPpred 2.0: 1.176 ACPs y 1.176 no-ACPs para entrenamiento, y 610 ACPs y 2.760 no-ACPs para la prueba independiente.

La combinación de ESM-2 8M con Regresión Logística obtuvo el mejor desempeño sobre la prueba independiente: AUC-ROC de 0,8745, F1-score de 0,6230 y coeficiente de correlación de Matthews de 0,5341, por delante de nueve de las once herramientas con AUC-ROC publicado sobre ese mismo conjunto y por debajo únicamente de MLACP 2.0 (0,893) y mACPpred 2.0 (0,887). El modelo de menor capacidad superó a las variantes de mayor dimensionalidad, resultado consistente con la longitud mediana de nueve residuos de los péptidos evaluados. La integración multimodal, por concatenación directa y por fusión tardía, no mejoró ese resultado (0,8676 y 0,8731), por lo que el modelo final se construyó únicamente sobre los embeddings de ESM-2 8M y se implementó en acp-classifier, un paquete de Python de libre acceso, instalable mediante pip y ejecutable desde línea de comandos sin unidad de procesamiento gráfico, lo que permite integrarlo en flujos de trabajo bioinformáticos automatizados.

---

* Trabajo de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Nydia Paola Rondón Villarreal. Doctora en Ingeniería, Área Ingeniería Electrónica.

Abstract

Title: Anticancer peptide classification model using artificial intelligence and bioinformatics tools *

Author: Brayan Arturo Navarro Sinuco and Ivan Andres Mendoza Oñate **

Key Words: anticancer peptides, machine learning, protein language models, ESM-2, bioinformatics, sequence classification.

Description:

Anticancer peptides (ACPs) are a promising therapeutic alternative to conventional chemotherapy, owing to their selectivity towards tumour cells and their lower systemic toxicity. Large-scale experimental identification is unfeasible, which motivates the development of computational screening tools.

This work systematically evaluated three numerical representations of peptide sequences: physicochemical descriptors (577 features), contextual embeddings from the ESM-2 protein language model in its 8- and 150-million-parameter variants, and ProtBERT embeddings. Seven classifiers were optimised on each representation through nested cross-validation with five outer and five inner folds, using the mACPpred 2.0 benchmark: 1,176 ACPs and 1,176 non-ACPs for training, and 610 ACPs and 2,760 non-ACPs for independent testing.

ESM-2 8M embeddings combined with a Logistic Regression classifier achieved the best performance on the independent test set: an AUC-ROC of 0.8745, an F1-score of 0.6230 and a Matthews correlation coefficient of 0.5341, ahead of nine of the eleven predictors with a published AUC-ROC on that same set and below only MLACP 2.0 (0.893) and mACPpred 2.0 (0.887). The smallest model outperformed the higher-dimensional variants, a result consistent with the nine-residue median length of the evaluation peptides. Multimodal integration, through direct concatenation and late fusion, did not improve that result (0.8676 and 0.8731), so the final model was built on ESM-2 8M embeddings alone and released as acp-classifier, an open-access Python package installable via pip and executable from the command line without a graphics processing unit, allowing direct integration into automated bioinformatics workflows.

* Degree Work

** Faculty of Physicomechanical Engineering. School of Systems and Computer Engineering. Director: Nydia Paola Rondón Villarreal. PhD in Engineering, Electronic Engineering Area.

Introducción

El cáncer representa en la actualidad una de las mayores problemáticas de salud a nivel mundial. De acuerdo con la Organización Mundial de la Salud (OMS), en el año 2022 se registraron más de 20 millones de nuevos diagnósticos y un aproximado de 9,7 millones de muertes asociadas a esta enfermedad (Bray et al., 2024). La magnitud de estas cifras, junto a su creciente número de casos, pone en evidencia la urgencia de fortalecer la investigación biomédica y promover el desarrollo de nuevas alternativas terapéuticas.

A pesar de que existen técnicas ampliamente usadas como la quimioterapia y la radioterapia, estas presentan desventajas como efectos secundarios severos, resistencia tumoral y costos elevados. En este contexto, los péptidos anticancerígenos (ACPs) emergen como una alternativa prometedora. Un péptido es una molécula compuesta por una cadena corta de aminoácidos (los bloques constructores de las proteínas) unidos mediante enlaces químicos. Desde una perspectiva computacional, cada péptido puede representarse como una secuencia de caracteres sobre un alfabeto de 20 símbolos (los 20 aminoácidos estándar), lo que permite aplicar técnicas de procesamiento de secuencias y aprendizaje automático. Los ACPs, al ser típicamente cargados positivamente a pH fisiológico, interactúan de manera selectiva con las membranas de células cancerosas que presentan carga negativa en su superficie exterior, a diferencia de las células sanas, actuando de forma análoga a un filtro discriminador en procesamiento de señales. Esta selectividad, junto con su baja toxicidad y biocompatibilidad, los convierte en candidatos atractivos frente a los tratamientos tradicionales (Bauso et al., 2024; Chinnadurai et al., 2023).

En el marco de este trabajo, la clasificación computacional de ACPs se aborda como un problema de clasificación binaria supervisada: dada una secuencia peptídica, el modelo determina

si dicha secuencia posee o no propiedades anticancerígenas, con base en características numéricas derivadas de su composición y propiedades fisicoquímicas.

No obstante, el estudio de estas moléculas presenta algunos desafíos, debido a que la identificación de péptidos con propiedades anticancerígenas requiere la exploración de un enorme número de posibles combinaciones en sus secuencias; en este sentido, la bioinformática y la inteligencia artificial (IA) son herramientas útiles para el descubrimiento de estas moléculas, permitiendo realizar análisis más rápidos, económicos y precisos (Cheong et al., 2024).

Por lo anterior, el presente trabajo plantea la construcción de un modelo de clasificación de péptidos anticancerígenos, haciendo uso de herramientas de IA y bioinformática, implementado como un paquete de Python que facilite su integración con flujos de trabajo de investigación, contribuyendo al desarrollo de nuevas terapias contra el cáncer y apoyando el avance científico y tecnológico en el área de la salud (Cock et al., 2009).

1. Planteamiento y Justificación del Problema

El cáncer, más allá de ser un gran problema para la salud pública, constituye un reto científico por la dificultad que demanda encontrar nuevas terapias que realmente funcionen y que sean seguras para los pacientes. Actualmente existen estrategias terapéuticas ampliamente utilizadas, pero estas suelen presentar limitaciones significativas; en muchos casos generan efectos secundarios severos, daño de tejidos sanos, debilitamiento del sistema inmune y en general pérdida de calidad de vida, y aún con todos estos efectos negativos, no garantizan una respuesta duradera debido al desarrollo de resistencia tumoral (Bray et al., 2024). Por otra parte, los elevados costos de los tratamientos restringen el acceso para una gran parte de la población, y en conjunto, todos

estos factores demuestran la necesidad urgente de desarrollar moléculas innovadoras capaces de ofrecer mayor eficacia, selectividad y seguridad, superando las limitaciones de los tratamientos tradicionales.

En este escenario, los péptidos anticancerígenos (ACPs) han generado grandes expectativas gracias a sus características únicas. Los péptidos son moléculas formadas por cadenas de aminoácidos, que son los bloques fundamentales que componen las proteínas, unidos entre sí mediante enlaces químicos llamados enlaces peptídicos. Gracias a su estructura, los ACPs pueden interactuar de manera más selectiva con las células tumorales y activar mecanismos biológicos naturales como la apoptosis, proceso mediante el cual el organismo elimina células dañinas de forma controlada, o la estimulación del sistema inmune para combatir el tumor. Estas características, junto con buena tolerancia y baja toxicidad, los convierten en candidatos con alto potencial para el desarrollo de nuevas alternativas a los tratamientos tradicionales (Bauso et al., 2024; Chinnadurai et al., 2023). La evidencia actual respalda este interés, debido a que algunos péptidos ya se encuentran en ensayos clínicos avanzados e incluso algunos ya han sido aprobados por agencias regulatorias, lo cual confirma su viabilidad terapéutica.

La complejidad de la búsqueda de nuevos ACPs, se comprende al considerar la magnitud de posibles combinaciones que presentan los péptidos, los cuales están conformados por la unión de aminoácidos a través de enlaces peptídicos. Teniendo en cuenta que existen 20 aminoácidos estándar distintos, incluso secuencias cortas de 10 residuos podrían generar más de 10.240.000.000.000 combinaciones posibles. Considerando que la longitud de los péptidos normalmente se encuentra entre 2 y 50 aminoácidos, el tamaño del espacio de búsqueda comprende miles de millones de distintas variantes y explorar esta gran cantidad de datos a través de ensayos

experimentales resultaría en una labor inviable debido al tiempo y recursos necesarios para su exploración (Cheong et al., 2024).

Ante esta problemática, la bioinformática y la inteligencia artificial (IA) presentan una gran oportunidad, ya que estas herramientas han demostrado ser cruciales en el descubrimiento de fármacos, esto gracias a su capacidad de procesar y analizar grandes volúmenes de datos biológicos, que no sería posible lograr únicamente en un laboratorio. Modelos computacionales que han sido entrenados con información de péptidos conocidos previamente, pueden aprender patrones asociados a las propiedades anticancerígenas, facilitando la clasificación y predicción de nuevos péptidos candidatos. Es así que el uso y desarrollo de estos modelos computacionales, permite optimizar recursos, acelerar procesos de investigación e incrementar las posibilidades de encontrar moléculas con mayor eficiencia terapéutica (Chen et al., 2021).

El presente trabajo se justifica en tres dimensiones fundamentales. Primero, en el plano científico, el cual busca contribuir a la búsqueda de nuevas moléculas con potencial actividad anticancerígena (Bauso et al., 2024). Segundo, en el plano tecnológico, se plantea la construcción de un modelo de clasificación implementado como un paquete de Python, lo cual facilita su integración en flujos de investigación bioinformática. Finalmente, en el plano social y a largo plazo, este proyecto responde a la necesidad de terapias más seguras y eficaces, lo cual impactaría directamente en la calidad de vida de los pacientes y en la sostenibilidad de los sistemas de salud (Bray et al., 2024).

Con base en lo anterior, se plantea la siguiente pregunta de investigación: ¿Un modelo multimodal que integre descriptores fisicoquímicos y embeddings de modelos de lenguaje proteico modernos como ESM-2 y ProtBERT alcanza un desempeño competitivo, bajo métricas estándar de clasificación como AUC-ROC y F1-score, frente a modelos predictivos basados en una sola

representación y frente a herramientas de referencia como AntiCP 2.0, MLACP 2.0 y mACPpred 2.0 para la clasificación de péptidos anticancerígenos?

Con base en estos elementos, se formula la siguiente hipótesis: la integración de descriptores fisicoquímicos y embeddings de modelos de lenguaje proteico modernos como ESM-2 y ProtBERT en una arquitectura multimodal permite desarrollar un modelo de clasificación de péptidos anticancerígenos con un desempeño competitivo, medido mediante AUC-ROC y F1-score, frente a modelos basados en una sola representación y frente a herramientas de referencia del estado del arte. Adicionalmente, se evaluará si la integración de ambas fuentes de información aporta mejoras significativas frente a los modelos individuales.

2. Objetivos

2.1 Objetivo General

Desarrollar un modelo que permita clasificar péptidos con potencial actividad anticancerígena, mediante la combinación de descriptores fisicoquímicos y representaciones contextuales basadas en modelos de lenguaje proteico, con el fin de generar un paquete de Python instalable desde consola.

2.2 Objetivos Específicos

- Construir un modelo predictivo de péptidos anticancerígenos utilizando un conjunto de datos de referencia, descriptores fisicoquímicos y técnicas de inteligencia artificial.
- Desarrollar un modelo predictivo de péptidos anticancerígenos a partir de técnicas de procesamiento de lenguaje natural aplicadas a secuencias peptídicas de un conjunto de datos de referencia.
- Integrar ambas representaciones en un modelo multimodal que combine ambas representaciones mediante diferentes estrategias utilizando el conjunto de datos de referencia.
- Evaluar el desempeño de los modelos predictivos desarrollados mediante métricas de clasificación estándar (exactitud, F1-Score, AUC-ROC y coeficiente de correlación de Matthews), para seleccionar el modelo que obtenga mejor rendimiento en el conjunto de datos independiente.
- Implementar el modelo predictivo seleccionado en un paquete de Python, documentado y de libre acceso para su integración en flujos de trabajo bioinformáticos.

3. Marco de Referencia

El siguiente marco de referencia contiene los fundamentos teóricos, conceptuales y normativos que avalan el desarrollo de un modelo de clasificación de péptidos anticancerígenos utilizando herramientas de inteligencia artificial y bioinformática. Se presentan los conceptos necesarios sobre péptidos, bioinformática, inteligencia artificial y su ayuda en la lucha contra el cáncer. Además, se incluye el estado del arte que reporta investigaciones y avances relacionados con las tecnologías aplicadas. En últimas se busca ofrecer una base sólida que permita entender la necesidad y justificación del proyecto presentado, dentro del ámbito científico y académico.

3.1 Marco Conceptual

3.1.1 Cáncer Como Problema de Salud Pública

El cáncer representa una de las principales causas de mortalidad a nivel mundial y constituye un desafío crítico para los sistemas de salud pública contemporáneos. Esta enfermedad se caracteriza por el crecimiento descontrolado de células anormales que adquieren la capacidad de invadir tejidos circundantes y propagarse a sitios distantes del organismo mediante un proceso conocido como metástasis (Bray et al., 2024).

Según datos publicados por la OMS, esta es la segunda causa principal de muerte a nivel mundial. Se estima que causa alrededor de 10 millones de muertes anualmente, y se proyecta que estos números continúen aumentando con el tiempo (Bray et al., 2024). Por esta razón, el cáncer es hoy uno de los problemas médicos públicos más significativos en el mundo. Los cánceres de próstata, mama, colon y estómago son los tipos más comunes de cáncer en Colombia, como se indica en el Observatorio Nacional de Cáncer (Instituto Nacional de Cancerología, 2023).

Los tratamientos convencionales contra el cáncer, que incluyen cirugía, quimioterapia, radioterapia y terapias hormonales, presentan limitaciones significativas que afectan tanto su eficacia como la calidad de vida de los pacientes (Chinnadurai et al., 2023). La quimioterapia, por ejemplo, aunque puede ser efectiva en la reducción de tumores, carece de especificidad celular, lo que resulta en daños colaterales a tejidos sanos, particularmente aquellos con alta tasa de división celular como la médula ósea, el tracto gastrointestinal y los folículos pilosos (Bray et al., 2024). Esta falta de selectividad genera efectos secundarios severos incluyendo inmunosupresión, náuseas, alopecia y fatiga crónica, entre otros.

Estas limitaciones fundamentales de los tratamientos tradicionales han impulsado la búsqueda de alternativas terapéuticas que ofrezcan mayor selectividad hacia células tumorales, menores efectos adversos, y costos de producción más accesibles (Bauso et al., 2024). En este contexto, los péptidos anticancerígenos emergen como candidatos prometedores que podrían complementar o, en algunos casos, sustituir las terapias convencionales, ofreciendo un perfil de seguridad mejorado y mecanismos de acción innovadores que podrían superar algunos de los obstáculos actuales en el tratamiento del cáncer (Chinnadurai et al., 2023; Bauso et al., 2024).

3.1.2 Péptidos

Los péptidos son moléculas fundamentales compuestas por cadenas de aminoácidos unidos mediante enlaces peptídicos, los cuales se forman a través de una reacción de condensación entre el grupo carboxilo (-COOH) de un aminoácido y el grupo amino ($-NH_2$) del siguiente (Chinnadurai et al., 2023). A diferencia de las proteínas, que típicamente contienen cientos o miles de residuos, los péptidos son estructuras significativamente más cortas, generalmente entre 2 y 50 aminoácidos (Chinnadurai et al., 2023; Bauso et al., 2024)

La versatilidad funcional de los péptidos en sistemas biológicos es extraordinaria. Estas moléculas participan en una amplia gama de procesos fisiológicos críticos, incluyendo la señalización celular a través de hormonas peptídicas como la insulina y el glucagón, la regulación de funciones endocrinas mediante péptidos neuroactivos, y la modulación de respuestas inmunológicas a través de citoquinas y quimioquinas (Chinnadurai et al., 2023). Esta diversidad funcional refleja la capacidad de los péptidos para adoptar conformaciones tridimensionales específicas que les permiten interactuar selectivamente con receptores celulares, enzimas y otras moléculas diana.

Una característica particularmente valiosa de los péptidos es su especificidad molecular (Bauso et al., 2024). La secuencia precisa de aminoácidos en un péptido determina no sólo su estructura tridimensional, sino también su afinidad y selectividad hacia objetivos biológicos específicos. Esta especificidad de secuencia-función permite el diseño racional de péptidos con actividades biológicas predefinidas, un concepto fundamental en el desarrollo de terapias peptídicas dirigidas (Chinnadurai et al., 2023).

La biocompatibilidad de los péptidos representa otra ventaja crucial, especialmente en el contexto farmacológico. Al estar compuestos por aminoácidos naturales, los mismos componentes que forman las proteínas del organismo, los péptidos son generalmente bien tolerados por el cuerpo humano y se metabolizan a través de vías fisiológicas normales mediante proteasas y peptidasas endógenas (Chinnadurai et al., 2023). Esta característica reduce significativamente el riesgo de acumulación tóxica y efectos adversos sistémicos en comparación con muchas moléculas sintéticas pequeñas utilizadas en farmacoterapia convencional, las cuales pueden requerir procesos metabólicos hepáticos complejos y generar metabolitos potencialmente tóxicos (Bauso et al., 2024).

En el contexto del desarrollo de fármacos, los péptidos ocupan un nicho único entre las moléculas pequeñas tradicionales y las terapias biológicas de gran tamaño como los anticuerpos monoclonales (Chinnadurai et al., 2023). Combinan la especificidad y potencia de las proteínas grandes con la relativa facilidad de síntesis, modificación química y optimización de las moléculas pequeñas. Esta posición intermedia, junto con los avances recientes en química de péptidos y estrategias para mejorar su estabilidad *in vivo* (como la ciclización, la incorporación de aminoácidos no naturales, o modificaciones N-terminal y C-terminal), ha llevado a un interés creciente en los péptidos como plataforma para el desarrollo de nuevos medicamentos (Bauso et al., 2024). Particularmente en áreas terapéuticas donde se requiere alta selectividad y baja toxicidad, como es el caso de las terapias anticancerígenas, los péptidos representan una alternativa prometedora con un perfil de riesgo y beneficio favorable (Chinnadurai et al., 2023; Bauso et al., 2024).

3.1.3 Péptidos Anticancerígenos (ACPs)

Existen péptidos que poseen propiedades anticancerígenas, ya que poseen diferentes mecanismos que ayudan a detener el crecimiento de células tumorales y cancerígenas (Bauso et al., 2024). Estos péptidos sobresalen por su habilidad de unirse de manera específica a las membranas de células cancerosas y afectarlas; sus modos de acción abarcan desde la promoción de la apoptosis, modulación de las membranas celulares, inhibición del proceso de angiogénesis y la activación de respuestas del sistema inmunológico. Existen investigaciones recientes que han demostrado la actividad biológica de ciertos péptidos anticancerígenos, así como su selectividad, lo que sugiere su utilidad como tratamiento complementario y, a su vez, una posible alternativa a la quimioterapia convencional (Cheong et al., 2024).

La base molecular de esta selectividad reside en las diferencias biofísicas y bioquímicas entre las membranas plasmáticas de células tumorales y células sanas (Cheong et al., 2024). Las células cancerosas presentan una serie de alteraciones características en la composición y organización de su membrana celular que las hacen susceptibles a la acción de los ACPs. Entre estas alteraciones se encuentra un incremento significativo en la densidad de fosfolípidos aniónicos en la monocapa externa de la membrana plasmática, particularmente fosfatidilserina y glicoproteínas O-glicosiladas con carga negativa (Bauso et al., 2024; Cheong et al., 2024). En contraste, en células normales estos componentes aniónicos se localizan predominantemente en la monocapa interna de la membrana, mantenida por la acción de flipasas específicas.

Los ACPs, que son típicamente catiónicos a pH fisiológico debido a la presencia de residuos de lisina y arginina, establecen inicialmente interacciones electrostáticas con estos componentes aniónicos de la membrana tumoral (Chinnadurai et al., 2023; Bauso et al., 2024). Esta atracción electrostática constituye el primer paso en el mecanismo de acción de muchos ACPs y es fundamental para su selectividad hacia células cancerosas. Una vez unidos a la membrana, los ACPs pueden adoptar conformaciones anfipáticas, particularmente estructuras de hélice α, que les permiten insertarse en la bicapa lipídica y formar poros transmembrana o causar la desestabilización general de la membrana (Bauso et al., 2024).

El potencial terapéutico de los ACPs se ve reforzado por varios estudios clínicos en curso, donde algunos péptidos anticancerígenos ya han alcanzado fases avanzadas de desarrollo e incluso han recibido aprobación regulatoria para indicaciones específicas (Chinnadurai et al., 2023). Este progreso valida el concepto de los péptidos como agentes anticancerígenos viables y subraya la importancia de desarrollar herramientas computacionales para acelerar la identificación y

caracterización de nuevos candidatos terapéuticos peptídicos (Bauso et al., 2024; Cheong et al., 2024).

3.1.4 Bases de Datos Bioinformáticas

El estudio sobre los ACPs necesita la disponibilidad de una gran cantidad de datos cuidadosamente seleccionados y precisos. Entre las bases de datos existentes sobre péptidos se pueden encontrar algunas más destacadas y confiables como: CancerPPD, que contiene más de 3400 péptidos y proteínas con actividad anticancerígena (Tyagi et al., 2015); APD3, que posee más de 3000 péptidos con propiedades bioactivas (Wang et al., 2016); DRAMP (Database of Research on Antimicrobial Peptides) ofrece información exhaustiva sobre péptidos antimicrobianos, incluyendo datos experimentales, estructuras tridimensionales, propiedades fisicoquímicas y referencias bibliográficas, lo cual facilita el análisis comparativo entre distintos tipos de péptidos bioactivos (Kang et al., 2019); DBAASP (Database of Antimicrobial Activity and Structure of Peptides), una base de datos integral que proporciona información sobre la actividad antimicrobiana y citotóxica de péptidos, incluyendo datos estructurales y experimentales útiles para el desarrollo de modelos predictivos (Pirtskhalava et al., 2021); CAMP (Collection of Anti-Microbial Peptides), que contiene secuencias de péptidos antimicrobianos con anotaciones detalladas sobre su actividad biológica (Waghu et al., 2016).

Finalmente, UniProt (The UniProt Consortium, 2023) representa una de las bases de datos más completas y ampliamente utilizadas en bioinformática para la consulta de información general sobre proteínas y péptidos. Esta plataforma proporciona secuencias revisadas, anotaciones funcionales, dominios estructurales, interacciones moleculares y referencias experimentales que permiten contextualizar la función biológica de los péptidos dentro de sistemas más complejos. En conjunto, estas bases de datos facilitan en gran medida el acceso a secuencias y ofrecen una amplia

variedad de detalles sobre características fisicoquímicas, estructurales y datos experimentales, los cuales resultan imprescindibles para el desarrollo, entrenamiento y validación de modelos computacionales destinados a la identificación y diseño de nuevos péptidos con potencial terapéutico (Tyagi et al., 2015; Jhong et al., 2022; Pirtskhalava et al., 2021).

3.1.5 Bioinformática e Inteligencia Artificial en el Descubrimiento de Fármacos

La bioinformática es un campo de estudio que representa y fusiona la biología, la informática y la estadística, todo esto para llevar a cabo un análisis exhaustivo usando datos biológicos a gran escala. En el ámbito oncológico, esta disciplina ayuda en gran medida a detectar patrones en las secuencias de péptidos o proteínas y además facilita el hallazgo de nuevas moléculas terapéuticas y acelera los métodos de validación experimental (Cheong et al., 2024).

La bioinformática ha evolucionado dramáticamente hasta convertirse en un componente esencial de la investigación biomédica moderna. En el contexto específico del descubrimiento de fármacos, esta disciplina permite el análisis sistemático de vastos conjuntos de datos moleculares, incluyendo secuencias genómicas, estructuras proteicas tridimensionales, perfiles de expresión génica y datos de interacciones moleculares, facilitando la identificación computacional de candidatos terapéuticos antes de la validación experimental (Bauso et al., 2024).

La inteligencia artificial ofrece una amplia gama de herramientas computacionales para analizar y predecir datos en el ámbito biológico. Gracias al aprendizaje automático, se han podido desarrollar clasificadores bastante eficientes mediante algoritmos como Máquinas de Vectores de Soporte (**SVM**, *Support Vector Machines*), Bosques Aleatorios (*Random Forest) y* k-Vecinos más Cercanos (**k-NN**, *k-Nearest Neighbors*), que han demostrado su eficacia en la predicción de péptidos bioactivos (Cheong et al., 2024). Estos modelos aprenden patrones complejos a partir de características fisicoquímicas derivadas de secuencias de aminoácidos, tales como composición,

carga neta, hidrofobicidad y propensión a formar estructuras secundarias específicas, entre otras. La arquitectura SVM, por ejemplo, ha mostrado particular robustez en la clasificación de péptidos anticancerígenos al encontrar hiperplanos óptimos en espacios de alta dimensionalidad, mientras que Random Forest proporciona estimaciones de importancia de características que revelan qué propiedades fisicoquímicas son más relevantes para la actividad anticancerígena (Phan et al., 2022). Estos enfoques de aprendizaje automático han sido implementados exitosamente en herramientas como AntiCP 2.0 y MLACP 2.0, que representan el estado del arte actual en la predicción computacional de péptidos anticancerígenos basados en descriptores fisicoquímicos tradicionales.

Adicionalmente, el aprendizaje profundo ha impulsado enormemente el uso de nuevas arquitecturas como Redes Neuronales Convolucionales (**CNN**, *Convolutional Neural Networks*) que han demostrado efectividad en la extracción automática de motivos secuenciales locales y patrones estructurales relevantes para la actividad anticancerígena, como ilustra el modelo xDeep-AcPEP desarrollado por Chen et al. (2021), que logra predicciones precisas mediante la combinación de capas convolucionales para detección de patrones con aprendizaje multitarea para optimización simultánea de múltiples propiedades peptídicas relacionadas. Las Redes Neuronales Recurrentes (**RNN**, *Recurrent Neural Networks*) y Memoria de Corto y Largo Plazo (**LSTM**, *Long Short-Term Memory*) son particularmente adecuadas para modelar dependencias secuenciales de largo alcance en cadenas peptídicas, capturando cómo aminoácidos distantes en la secuencia primaria pueden interactuar funcionalmente en la estructura tridimensional plegada (Yan et al., 2020). Adicionalmente, los modelos de conjunto (*ensemble*) que combinan predicciones de múltiples algoritmos han demostrado mejorar la robustez y generalización, como se evidencia en MLACP 2.0 que integra 8 clasificadores diferentes (Phan et al., 2022). Estas arquitecturas eliminan

la necesidad de diseño manual exhaustivo de características, permitiendo que el modelo descubra automáticamente representaciones óptimas de las secuencias peptídicas.

Sin embargo, uno de los mayores desafíos sigue siendo el sobreajuste (overfitting) derivado de la escasez de datos de alta calidad en comparación con la dimensionalidad de los espacios de características y la capacidad de los modelos de aprendizaje profundo (Chinnadurai et al., 2023). Para mitigar este problema, se emplean estrategias metodológicas como la validación cruzada anidada (nested cross-validation) para evaluación rigurosa del rendimiento predictivo sin sesgo optimista (Varma y Simon, 2006), técnicas de regularización mediante *dropout* y penalización de pesos, y la selección cuidadosa de características fisicoquímicas biológicamente interpretables. La integración de enfoques bioinformáticos tradicionales basados en descriptores fisicoquímicos con técnicas avanzadas de inteligencia artificial constituye una dirección prometedora, donde modelos híbridos combinan el conocimiento del dominio acumulado con el poder de generalización del aprendizaje automático, aprovechando las fortalezas de ambos paradigmas para mejorar la precisión y confiabilidad en el cribado virtual de bibliotecas peptídicas.

3.1.6 Procesamiento de Lenguaje Natural Aplicado a Secuencias

La aplicación del procesamiento del lenguaje natural en informática se basa en la comparación de cadenas de aminoácidos con cadenas textuales. Cada aminoácido se interpreta como un símbolo o token, lo que permite el uso de modelos diseñados para el lenguaje humano en el análisis de secuencias biológicas. Se han adoptado incrustaciones creadas a partir de modelos como Word2Vec, un modelo que genera representaciones vectoriales de palabras (o en este caso, aminoácidos) basándose en su contexto de aparición; ProtBERT (Protein Bidirectional Encoder Representations from Transformers), un modelo basado en la arquitectura BERT adaptado específicamente para secuencias proteicas que captura relaciones contextuales bidireccionales

entre aminoácidos; ProtT5, basado en la arquitectura T5 (Text-to-Text Transfer Transformer) y entrenado en grandes conjuntos de datos proteicos para generar representaciones semánticas profundas; y ESM (Evolutionary Scale Modeling), una familia de modelos que aprenden representaciones proteicas a partir de millones de secuencias evolutivas, capturando información estructural y funcional implícita. Estos modelos ayudan a capturar características semánticas, estructurales y evolutivas de las secuencias peptídicas, permitiendo una mejor comprensión de sus propiedades biológicas (Tran et al., 2023; Cheong et al., 2024; Elnaggar et al., 2021; Rao et al., 2021; Lin et al., 2023).

Una ventaja fundamental de estas representaciones contextuales es su capacidad para capturar información que va más allá de la simple composición de aminoácidos. A diferencia de los descriptores fisicoquímicos tradicionales que tratan cada posición de forma independiente, los embeddings generados por modelos de lenguaje proteico aprenden dependencias de largo alcance y patrones evolutivos conservados a través del entrenamiento en millones de secuencias naturales (Lin et al., 2023). Por ejemplo, ESM-2, entrenado en más de 65 millones de secuencias proteicas, ha demostrado capturar información estructural tridimensional implícita sin requerir datos experimentales de estructura, logrando predicciones de plegamiento comparables a métodos basados en alineamientos múltiples, pero con velocidades 60 veces superiores (Lin et al., 2023).

3.1.6.1 ESM-2 (Evolutionary Scale Modeling 2). ESM-2 (Evolutionary Scale Modeling 2) es una familia de modelos de lenguaje proteico desarrollada por Meta AI Research y presentada por Lin et al. (2023). Su entrenamiento se realizó sobre aproximadamente 65 millones de secuencias proteicas únicas procedentes de la base de datos UniRef, agrupadas a partir de los clústeres de UniRef50, utilizando como objetivo el modelado de lenguaje enmascarado: dado que algunos aminoácidos de la secuencia son ocultados al modelo, este debe predecirlos a partir

del contexto restante, lo que obliga al modelo a aprender relaciones biológicas entre posiciones de la secuencia.

La arquitectura interna de ESM-2 corresponde a un transformer bidireccional, en el que cada posición de la secuencia puede relacionarse con todas las demás simultáneamente a través del mecanismo de autoatención. Esta característica permite capturar dependencias entre aminoácidos separados por decenas de posiciones, algo que los descriptores fisicoquímicos convencionales no logran representar.

Una de las ventajas más relevantes de ESM-2 para este trabajo es su disponibilidad en múltiples tamaños, lo que permite seleccionar una variante acorde con los recursos computacionales disponibles. En este proyecto se evaluaron dos variantes: esm2_t6_8M_UR50D con 8 millones de parámetros que genera vectores de 320 dimensiones, y esm2_t30_150M_UR50D con 150 millones de parámetros que genera vectores de 640 dimensiones.

3.1.6.2 ProtBERT. ProtBERT forma parte de la familia de modelos ProtTrans, desarrollada por Elnaggar et al. (2021) con el objetivo de adaptar las arquitecturas transformer del procesamiento de lenguaje natural al dominio de las secuencias biológicas. En particular, ProtBERT toma como base la arquitectura BERT original de Google y la reentrena desde cero sobre secuencias proteicas, sin transferir los pesos aprendidos sobre texto en lenguaje natural.

La variante utilizada en este trabajo, denominada prot_bert_bfd, fue entrenada sobre la base de datos BFD (Big Fantastic Database), que agrupa aproximadamente 2.100 millones de secuencias proteicas provenientes de diversas fuentes públicas, convirtiéndola en uno de los conjuntos de entrenamiento más extensos disponibles para modelos de lenguaje proteico. Como

resultado de este entrenamiento, el modelo genera vectores de representación de 1.024 dimensiones.

Una diferencia práctica importante respecto a ESM-2 es que ProtBERT requiere que los aminoácidos de la secuencia estén separados por espacios antes de la tokenización, dado que el tokenizador trata cada aminoácido como una unidad léxica independiente. Esto implica que la secuencia "ACDEF" debe ser transformada a "A C D E F" antes de ser procesada por el modelo.

3.1.7 Descriptores Fisicoquímicos

Los descriptores fisicoquímicos constituyen una forma de representación numérica de las propiedades químicas y físicas de una secuencia peptídica, derivadas directamente de la composición de aminoácidos mediante fórmulas y tablas de referencia establecidas en la literatura bioquímica. A diferencia de los embeddings generados por modelos de lenguaje, los descriptores fisicoquímicos no requieren entrenamiento previo ni recursos computacionales especiales, y son completamente interpretables en términos biológicos.

En este trabajo se calcularon descriptores pertenecientes a cuatro categorías, empleando las librerías peptides.py, adaptación a Python del paquete Peptides (Osorio et al., 2015), y propy3 (Cao et al., 2013):

La primera categoría comprende propiedades globales de la secuencia: peso molecular, carga neta a pH fisiológico (7.0), punto isoeléctrico, hidrofobicidad calculada bajo tres escalas reconocidas en la literatura (Eisenberg, Kyte-Doolittle y Hopp-Woods), momento hidrofóbico calculado con ángulos de 100° y 160°, índice de inestabilidad e índice alifático, para un total de diez propiedades. Estas propiedades capturan características macroscópicas del péptido directamente relacionadas con su comportamiento en solución y su capacidad de interacción con membranas celulares.

La segunda categoría corresponde a la composición de aminoácidos (AAC), un vector de 20 valores que representa la frecuencia relativa de cada aminoácido estándar en la secuencia. Este descriptor, aunque simple, ha demostrado alta capacidad discriminativa en múltiples estudios de clasificación de péptidos bioactivos.

La tercera categoría es la composición de dipéptidos (DPC), que extiende la AAC considerando pares de aminoácidos consecutivos, generando un vector de 400 valores. Esta representación captura información sobre la estructura local del péptido que no está presente en la AAC.

Finalmente, los descriptores CTD (Composición, Transición y Distribución) agrupan los aminoácidos según propiedades compartidas como polaridad, volumen molecular y solubilidad, describiendo cómo se distribuyen estas propiedades a lo largo de la secuencia.

3.1.8 Clasificadores de Aprendizaje Automático

Con el propósito de identificar el clasificador más adecuado para cada tipo de representación, en este trabajo se evaluaron siete algoritmos de aprendizaje automático ampliamente utilizados en problemas de clasificación binaria en bioinformática.

La Máquina de Vectores de Soporte (SVM) con kernel RBF busca el hiperplano de máximo margen que separa las dos clases en un espacio de características de alta dimensionalidad. Su comportamiento está controlado por dos hiperparámetros: C, que penaliza las clasificaciones incorrectas, y gamma, que determina el radio de influencia de cada muestra de entrenamiento. Este clasificador es especialmente robusto cuando el número de características es comparable o mayor al número de muestras, condición frecuente en problemas de bioinformática.

El Random Forest construye un conjunto de árboles de decisión entrenados sobre subconjuntos aleatorios de los datos y las características, combinando sus predicciones mediante

votación por mayoría. Su principal fortaleza es la reducción de la varianza del modelo sin sacrificar significativamente el sesgo.

Los *Extra Trees* (*Extremely Randomized Trees*) introducen un nivel adicional de aleatoriedad respecto al Random Forest al seleccionar los umbrales de división de los nodos de forma completamente aleatoria, lo que reduce aún más la varianza y produce modelos más rápidos de entrenar con resultados frecuentemente comparables.

El Gradient Boosting construye el modelo de forma secuencial, donde cada árbol nuevo corrige los errores residuales del modelo anterior. A diferencia de los métodos de bagging como Random Forest, el boosting optimiza explícitamente una función de pérdida, lo que le permite producir modelos más precisos a costa de mayor tiempo de entrenamiento.

La Regresión Logística es un modelo lineal que estima la probabilidad de pertenencia a cada clase mediante la función sigmoide. A pesar de su simplicidad conceptual, ha demostrado ser altamente competitiva cuando las representaciones de entrada son de buena calidad, como es el caso de los embeddings generados por modelos de lenguaje proteico preentrenados.

El clasificador K-Nearest Neighbors (KNN) asigna a cada muestra la clase más frecuente entre sus k vecinos más cercanos en el espacio de características, según una métrica de distancia predefinida. Su comportamiento es no paramétrico y depende fuertemente de la escala de las características, por lo que requiere normalización previa.

Finalmente, el Perceptrón Multicapa (MLP) es una red neuronal con capas completamente conectadas que aprende representaciones no lineales mediante retropropagación. En este trabajo se evaluaron arquitecturas de una y dos capas ocultas con entre 128 y 512 neuronas por capa.

3.1.9 Validación Cruzada Anidada

Una de las decisiones metodológicas más importantes en el diseño de este trabajo fue la adopción de la validación cruzada anidada (nested cross-validation) para la evaluación de los modelos. Esta decisión se fundamenta en los hallazgos de Varma y Simon (2006), quienes demostraron que el uso del mismo conjunto de datos para optimizar hiperparámetros y estimar el rendimiento del modelo produce estimaciones optimistas del error de generalización, fenómeno conocido como sesgo de selección.

La validación cruzada anidada resuelve este problema mediante dos bucles independientes. El bucle externo, configurado con 5 pliegues, es responsable exclusivamente de estimar el rendimiento real del modelo sobre datos que no participaron en ninguna etapa de optimización. El bucle interno, también con 5 pliegues, opera sobre los datos de entrenamiento del bucle externo y se encarga de identificar los mejores hiperparámetros mediante GridSearchCV optimizando el AUC-ROC.

Esta configuración de 5×5 pliegues implica 25 ciclos de entrenamiento y validación por cada combinación de hiperparámetros de la rejilla, lo que proporciona estimaciones estadísticamente robustas del rendimiento real. Las métricas reportadas en la validación cruzada corresponden al promedio y la desviación estándar sobre los 5 pliegues externos, permitiendo evaluar no solo el rendimiento medio sino también la estabilidad del modelo ante diferentes particiones de los datos.

La validación cruzada anidada se aplica exclusivamente sobre el conjunto de entrenamiento. Una vez estimado el rendimiento, el modelo final de cada clasificador se obtiene repitiendo la búsqueda en rejilla —con la misma validación interna de cinco pliegues— sobre la totalidad del conjunto de entrenamiento y reentrenando con la mejor configuración encontrada.

Solo ese modelo se aplica, una única vez, al conjunto de prueba independiente, que no interviene en ninguna etapa de la validación cruzada ni de la selección de hiperparámetros. Los hiperparámetros recogidos en la Tabla B2 corresponden a este reentrenamiento final.

3.1.10 Desbalance de Clases y SMOTE

El conjunto de prueba independiente de mACPpred 2.0 presenta una distribución significativamente desbalanceada entre las dos clases del problema: 610 ACPs (18.1 %) frente a 2.760 no-ACPs (81.9 %). Lejos de ser una limitación del dataset, este desbalance refleja la realidad biológica del problema de cribado computacional de péptidos, donde la proporción de moléculas con actividad anticancerígena es inherentemente baja respecto al total de secuencias candidatas en una librería de péptidos.

Esta condición tiene implicaciones directas sobre la elección de las métricas de evaluación y la estrategia de entrenamiento. La exactitud (*accuracy*) resulta una métrica inapropiada bajo desbalance severo: un clasificador trivial que predijera siempre la clase mayoritaria obtendría una accuracy del 81.9 % sin haber aprendido ningún patrón discriminativo. Por esta razón, en este trabajo se adopta el AUC-ROC como métrica principal de evaluación, dado que cuantifica la capacidad discriminativa del modelo de forma independiente al umbral de decisión y es la métrica que reportan las herramientas de referencia sobre este conjunto, lo que permite la comparación directa. Puesto que bajo desbalance la curva de precisión-exhaustividad resulta más informativa que la curva ROC (Saito y Rehmsmeier, 2015), para el modelo final se reporta además el área bajo esa curva (AUPRC). Complementariamente se reportan el F1-score y el coeficiente de correlación de Matthews (MCC), ambos más informativos que la accuracy bajo condiciones de desbalance.

El conjunto de entrenamiento del benchmark empleado está balanceado por diseño: sus autores submuestrearon aleatoriamente la clase mayoritaria hasta igualar ambas clases en 1.176

secuencias. El desbalance, por tanto, es una condición exclusiva del conjunto de prueba (610 ACPs frente a 2.760 no-ACPs), y esa asimetría entre las condiciones de entrenamiento y de evaluación modifica sustancialmente el papel que puede desempeñar cualquier técnica de balanceo: sobre un conjunto de entrenamiento ya equilibrado, el remuestreo sintético no tiene clase minoritaria que ampliar.

SMOTE es una técnica de sobremuestreo que genera muestras sintéticas de la clase minoritaria interpolando linealmente entre pares de muestras reales existentes en el espacio de características. A diferencia del sobremuestreo aleatorio simple, que replica muestras existentes con riesgo de sobreajuste, SMOTE crea nuevas muestras que amplían la región del espacio ocupada por la clase minoritaria, favoreciendo la generalización del clasificador hacia regiones no representadas en los datos originales. Su efecto, no obstante, está condicionado por la existencia de una clase minoritaria en el conjunto de entrenamiento: cuando ambas clases se encuentran igualmente representadas, el algoritmo no genera muestra sintética alguna y su aplicación resulta inocua.

Un aspecto metodológico crítico en la aplicación de SMOTE es su correcta ubicación dentro del proceso de validación. En este trabajo, SMOTE se incluyó dentro de cada pliegue de entrenamiento del bucle externo de la validación cruzada anidada, garantizando que las muestras sintéticas nunca contaminen los conjuntos de validación ni de prueba. Esta práctica es fundamental para obtener estimaciones de rendimiento no sesgadas, ya que aplicar SMOTE antes de la partición de los datos introduciría fuga de información entre los conjuntos de entrenamiento y evaluación, inflando artificialmente las métricas reportadas. La verificación de su comportamiento efectivo se reporta en la sección 6.4.

3.1.11 Herramientas Computacionales en Python

Python se ha consolidado como el lenguaje de programación predominante en bioinformática debido a su sintaxis sencilla, filosofía de código abierto, amplia comunidad de desarrolladores y gran cantidad de bibliotecas especializadas. Esta popularidad se refleja en el ecosistema científico moderno, donde Python actúa como lenguaje integrador entre diferentes herramientas computacionales, desde el análisis exploratorio de datos hasta la implementación de modelos de aprendizaje profundo en producción. La versatilidad de Python permite a investigadores sin formación profunda en ciencias de la computación desarrollar flujos de trabajo reproducibles y escalables, democratizando el acceso a técnicas computacionales avanzadas en el ámbito biomédico (Cock et al., 2009).

Entre las bibliotecas más relevantes para el desarrollo de modelos predictivos de péptidos anticancerígenos se encuentra Biopython, que facilita el manejo, análisis y manipulación de secuencias biológicas en múltiples formatos (FASTA, GenBank, PDB), así como la comunicación con bases de datos bioinformáticas remotas como NCBI y UniProt (Cock et al., 2009). Scikit-learn proporciona implementaciones optimizadas de algoritmos de aprendizaje automático supervisado y no supervisado, incluyendo SVM, Random Forest, k-NN, además de herramientas para preprocesamiento de datos, selección de características, validación cruzada y evaluación de modelos mediante métricas estándar (Pedregosa et al., 2011). Para el desarrollo de arquitecturas de aprendizaje profundo, PyTorch y TensorFlow ofrecen frameworks flexibles que permiten la construcción de redes neuronales personalizadas, diferenciación automática, optimización mediante GPU y despliegue de modelos entrenados (Paszke et al., 2019; Abadi et al., 2016).

Adicionalmente, librerías especializadas como peptides.py y ProPy3 proporcionan funciones específicas para calcular descriptores fisicoquímicos y bioinformáticos de secuencias

proteicas, incluyendo composición de aminoácidos, índices de hidrofobicidad, perfiles de carga, pseudo-composición de aminoácidos (PseAAC) y descriptores de composición-transición-distribución (CTD), facilitando la extracción de características numéricas a partir de secuencias peptídicas para su uso en modelos predictivos (Osorio et al., 2015; Cao et al., 2013). Estas herramientas permiten integrar técnicas de análisis de secuencias con modelos de incrustación de proteínas y aprendizaje profundo (Cock et al., 2009).

3.1.12 Evaluación y Métricas de Modelos Predictivos

La evaluación y validación de los modelos de predicción es de suma importancia para garantizar la fiabilidad de los resultados en un ámbito biomédico. Entre las métricas más utilizadas se encuentra la exactitud (accuracy), que mide el porcentaje total de predicciones correctas realizadas por el modelo. Aunque es una medida general del rendimiento, puede resultar insuficiente cuando se trabaja con conjuntos de datos desbalanceados, donde una clase es más frecuente que otra.

La precisión (*precision*) evalúa la proporción de predicciones positivas correctas respecto al total de predicciones positivas realizadas. Esta métrica es especialmente relevante en contextos biomédicos donde un falso positivo puede generar diagnósticos incorrectos o tratamientos innecesarios.

Por otro lado, la recuperación o sensibilidad (*recall*) mide la capacidad del modelo para identificar correctamente los casos positivos reales. En aplicaciones como la detección de enfermedades, una alta sensibilidad es crucial, ya que reduce la probabilidad de falsos negativos, es decir, casos reales que el modelo no logra detectar.

La puntuación F1 (F1-score) integra precisión y recuperación en una sola métrica, calculando su media armónica. Esta medida resulta útil cuando se busca un equilibrio entre ambas,

especialmente en problemas donde tanto falsos positivos como falsos negativos tienen consecuencias significativas.

El coeficiente de correlación de Matthews (MCC) mide la correlación entre las clases reales y las predichas a partir de los cuatro valores de la matriz de confusión: $MCC = (VP \cdot VN - FP \cdot FN) / \sqrt{[(VP+FP)(VP+FN)(VN+FP)(VN+FN)]}$, donde VP, VN, FP y FN son los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Toma valores entre −1 y 1, donde 0 corresponde a un clasificador aleatorio y 1 a una clasificación perfecta. A diferencia de la exactitud y del F1-score, considera las cuatro celdas de la matriz de confusión, lo que lo hace robusto frente al desbalance de clases (Powers, 2011).

Otra métrica ampliamente utilizada es el ROC-AUC (Área bajo la curva ROC), que evalúa la capacidad del modelo para distinguir entre clases. Un valor de ROC-AUC cercano a 1 indica un excelente desempeño del modelo, mientras que valores cercanos a 0.5 sugieren una capacidad de clasificación similar al azar (Cheong et al., 2024).

Sin embargo, en conjuntos de datos desbalanceados, como es frecuente en problemas biomédicos, las curvas *Precision-Recall* pueden ser más informativas que las curvas ROC para evaluar el rendimiento del clasificador (Saito y Rehmsmeier, 2015).

3.1.13 Consideraciones Éticas y Legales

Todos los trabajos relacionados con la biomedicina deben cumplir con estándares éticos y legales que garanticen la vida y la dignidad humana, recogidos internacionalmente en la Declaración de Helsinki (World Medical Association, 2013). El presente trabajo no involucra sujetos humanos, muestras biológicas ni datos personales: emplea exclusivamente secuencias peptídicas de acceso público, distribuidas por los autores del conjunto de referencia (Sangaraju et al., 2024). En coherencia con los principios de ciencia abierta, el código y el modelo resultantes se

publican bajo licencia MIT. Por último, dado que la herramienta produce predicciones computacionales, sus resultados deben entenderse como hipótesis que requieren validación experimental y no como evidencia de actividad terapéutica; el uso responsable de la inteligencia artificial en biomedicina exige comunicar de forma explícita esas limitaciones a quien emplee el modelo.

3.2 Estado del Arte

El estado actual del arte en la clasificación de péptidos anticancerígenos ha mostrado un progreso notable en las últimas décadas, gracias a la creación de bases de datos especializadas y la implementación de técnicas de inteligencia artificial (Cheong et al., 2024). Inicialmente, la investigación se centró en el uso de descriptores fisicoquímicos y algoritmos tradicionales de aprendizaje automático. Sin embargo, recientemente se han integrado modelos de aprendizaje profundo, así como representaciones de secuencias sofisticadas mediante procesamiento del lenguaje natural. A continuación, se describen algunos de los estudios más significativos que sirven de base a este proyecto.

Tabla 1

Estado del arte.

Ref.	Título	Año	Revista / Fuente	Contribución
[1]	Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries (Bray et al.)	2024	CA: A Cancer Journal for Clinicians	Presenta estadísticas actualizadas de incidencia y mortalidad de cáncer a nivel mundial. Proporciona el contexto epidemiológico necesario para justificar la relevancia de nuevas alternativas terapéuticas como los péptidos anticancerígenos.
[2]	Cáncer en cifras – Instituto Nacional de Cancerología, Colombia	2023	Instituto Nacional de Cancerología (INC), Colombia	Ofrece datos oficiales sobre incidencia y mortalidad de cáncer en el contexto colombiano. Respalda la pertinencia del proyecto en el ámbito nacional y su aporte potencial al sistema de salud local.
[3]	Current research status of anti-cancer peptides: Mechanism of action, production, and clinical applications (Chinnadurai et al.)	2023	Biomedicine & Pharmacotherapy	Revisión exhaustiva del estado actual de los ACPs, incluyendo mecanismos de acción, métodos de producción y aplicaciones clínicas. Brinda fundamentos conceptuales sólidos para el marco de referencia.
[4]	Identification of anticancer peptides from the genome of Candida albicans: In silico screening, *in vitro* and *in vivo* validations (Cheong et al.)	2024	Journal of Chemical Information and Modeling	Ejemplo de integración de bioinformática con validaciones experimentales en la identificación de ACPs. Conecta directamente con el enfoque del proyecto propuesto de combinar herramientas computacionales con aplicaciones biomédicas.
[5]	Biological activity of natural and synthetic peptides as anticancer agents (Bauso et al.)	2024	International Journal of Molecular Sciences	Analiza la actividad biológica de péptidos naturales y sintéticos con potencial anticancerígeno, destacando su versatilidad estructural y su capacidad de interacción selectiva con células tumorales.

Ref.	Título	Año	Revista / Fuente	Contribución
[6]	dbAMP 2.0: Updated resource for antimicrobial peptides with an enhanced scanning method for genomic and proteomic data (Jhong et al.)	2022	Nucleic Acids Research	Presenta una base de datos actualizada de péptidos antimicrobianos, relevante para el entrenamiento y validación de modelos bioinformáticos utilizados en la clasificación de ACPs.
[7]	DBAASP v3: Database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics (Pirtskhalava et al.)	2021	Nucleic Acids Research	Ofrece una base de datos de péptidos con actividad citotóxica y antimicrobiana, usada como fuente de información estructural para el desarrollo de modelos predictivos de péptidos anticancerígenos.
[8]	AntiCP 2.0: An updated model for predicting anticancer peptides (Agrawal et al.)	2021	Briefings in Bioinformatics	Presenta un clasificador de ACPs basado exclusivamente en descriptores fisicoquímicos como composición de aminoácidos, dipéptidos y perfiles binarios, utilizando algoritmos como SVM, Random Forest, Extra Trees y k-NN. Su principal limitación es la dependencia total de descriptores fisicoquímicos sin incorporar representaciones contextuales, lo que restringe su capacidad para capturar patrones evolutivos y estructurales de las secuencias peptídicas. Adicionalmente, solo está disponible como servidor web sin paquete instalable localmente.
[9]	MLACP 2.0: An updated machine learning tool for anticancer peptide prediction (Phan et al., 2022)	2022	Computational and Structural Biotechnology Journal	Desarrolla un meta-predictor de ACPs que combina 11 encodings con 8 clasificadores, incluyendo CNN. Aunque incorpora el embedding seq2vec basado en ELMo (2018), este modelo es significativamente menos potente que arquitecturas modernas como ESM-2 o ProtBERT. Los propios autores reconocen como trabajo futuro el desarrollo de sistemas de codificación independientes de propiedades fisicoquímicas y de modelos

Ref.	Título	Año	Revista / Fuente	Contribución
				híbridos que combinen enfoques convencionales con aprendizaje profundo. Solo disponible como servidor web sin paquete instalable.
[10]	Evolutionary-scale prediction of atomic level protein structure with a language model (Lin et al.)	2023	Science	Presenta ESM-2, una familia de modelos de lenguaje (language models) de proteínas escalados desde 8 millones hasta 15 mil millones de parámetros. Demuestra que la estructura atómica de proteínas emerge en modelos de lenguaje entrenados únicamente con secuencias mediante el objetivo de modelado de lenguaje enmascarado (masked language modeling).
[11]	Deep-AmPEP30: Improve Short Antimicrobial Peptides Prediction with Deep Learning (Yan et al.)	2020	Molecular Therapy - Nucleic Acids	Desarrolla un clasificador de péptidos antimicrobianos cortos (≤30 aminoácidos) utilizando Redes Neuronales Convolucionales (Convolutional Neural Networks, CNN) y composición de aminoácidos reducidos (Reduced Amino Acid Composition, RAAC). Emplea características basadas en PseKRAAC (pseudo K-tuple Reduced Amino Acid Composition) que agrupan aminoácidos según información evolutiva, matrices de sustitución, hidrofobicidad y energía de contacto. Aunque se enfoca en péptidos antimicrobianos en lugar de anticancerígenos, resulta pertinente para este trabajo porque aborda específicamente secuencias de longitud reducida, condición que caracteriza al conjunto de prueba independiente empleado aquí, cuya mediana de longitud es de nueve residuos.
[12]	ACP-ESM: A novel *framework* for classification of anticancer peptides using protein-oriented transformer approach (Kilimci & Yalcin)	2024	Artificial Intelligence in Medicine	Propone un framework basado en cuatro modelos transformer (ESM, ProtBERT, BioBERT y SciBERT) para la clasificación de ACPs a partir de secuencias de aminoácidos. El modelo ESM alcanza una precisión del 96.45 % en el dataset AntiCp2, estableciendo nuevos benchmarks en el estado del arte. Constituye el antecedente más directo al enfoque de embeddings propuesto en este trabajo.

Ref.	Título	Año	Revista / Fuente	Contribución
[13]	mACPpred 2.0: Stacked Deep Learning for Anticancer Peptide Prediction with Integrated Spatial and Probabilistic Feature Representations (Sangaraju et al.)	2024	Journal of Molecular Biology	Desarrolla un predictor de ACPs mediante aprendizaje profundo apilado con bloques CNN 1D, integrando descriptores fisicoquímicos con 16 embeddings NLP preentrenados no especificados explícitamente. Construye el dataset de referencia utilizado en el presente trabajo (1176 ACPs y 1176 no-ACPs entrenamiento; 610 ACPs y 2760 no-ACPs prueba). Aunque dispone de un repositorio GitHub para uso local, no ofrece un paquete instalable mediante gestores estándar como pip o Conda, lo que limita su adopción práctica. Adicionalmente, no especifica ni compara modelos de lenguaje proteico modernos como ESM-2 o ProtBERT, y utiliza una arquitectura de apilamiento de predicciones en lugar de fusión directa de representaciones.
[14]	ACP-EPC: An Interpretable Deep Learning Framework for Anticancer Peptide Prediction Utilizing Pre-trained Protein Language Model and Multi-view Feature Extracting Strategy (Lv et al.)	2025	Molecular Diversity	Desarrolla un marco de aprendizaje profundo interpretable que integra representaciones contextuales de ESM-2 con descriptores fisicoquímicos mediante un mecanismo de atención cruzada (Cross-Attention) para fusión multimodal, alcanzando precisiones de 0.935 y 0.984 en los conjuntos de prueba ACP135 y ACP99 respectivamente. Constituye el antecedente más directo a la presente propuesta al combinar modelos de lenguaje proteico con descriptores fisicoquímicos para clasificación de ACPs. Sus limitaciones concretas incluyen: utilizar únicamente ESM-2 sin comparar con modelos alternativos como ProtBERT; ser evaluado exclusivamente sobre datasets pequeños (ACP135 con 135 péptidos y ACP99 con 99 péptidos) sin comparación directa con herramientas de referencia sobre datasets de mayor escala como el de mACPpred 2.0; y estar disponible únicamente como servidor web en http://www.bioai-lab.com/ACP-EPC sin ofrecer un paquete instalable localmente desde consola.

Ref.	Título	Año	Revista / Fuente	Contribución
[15]	PLMACPred: Prediction of Anticancer Peptides Based on Protein Language Model and Wavelet Denoising Transformation (Arif et al.)	2024	Scientific Reports	Desarrolla un predictor de ACPs que combina embeddings de ESM-2 y ProtT5 con descriptores fisicoquímicos y evolutivos, aplicando un algoritmo de denoising wavelet 2D para reducir el ruido de los vectores de características. Utiliza un clasificador de bosque profundo en cascada (CDF) logrando precisiones superiores al 94 % en tres datasets de referencia. Confirma que la combinación de modelos de lenguaje proteico con descriptores tradicionales mejora el desempeño predictivo, aunque no ofrece herramienta instalable desde consola y fue evaluado en datasets diferentes al utilizado en el presente trabajo.

El análisis del estado del arte revela una evolución progresiva en los enfoques computacionales para la clasificación de ACPs: desde clasificadores basados exclusivamente en descriptores fisicoquímicos como AntiCP 2.0 (Agrawal et al., 2021) y MLACP 2.0 (Phan et al., 2022), hasta modelos que incorporan embeddings de modelos de lenguaje proteico como ACP-ESM (Kilimci y Yalcin, 2024), y más recientemente trabajos que integran ambas representaciones en arquitecturas multimodales como mACPpred 2.0 (Sangaraju et al., 2024) y ACP-EPC (Lv et al., 2025). Este último representa el antecedente más directo a la presente propuesta, al integrar ESM-2 con descriptores fisicoquímicos mediante un mecanismo de atención cruzada para la fusión multimodal de características. Sin embargo, ACP-EPC presenta limitaciones que este trabajo busca superar: primero, utiliza únicamente ESM-2 como modelo de lenguaje proteico, sin evaluar alternativas como ProtBERT, que captura relaciones contextuales bidireccionales entre aminoácidos y ha demostrado desempeño competitivo en tareas similares; segundo, fue evaluado sobre datasets pequeños (ACP135 y ACP99), sin comparación directa con herramientas de referencia como mACPpred 2.0 sobre un dataset de mayor escala; y tercero, como la mayoría de las herramientas existentes, solo está disponible como servidor web, sin ofrecer un paquete

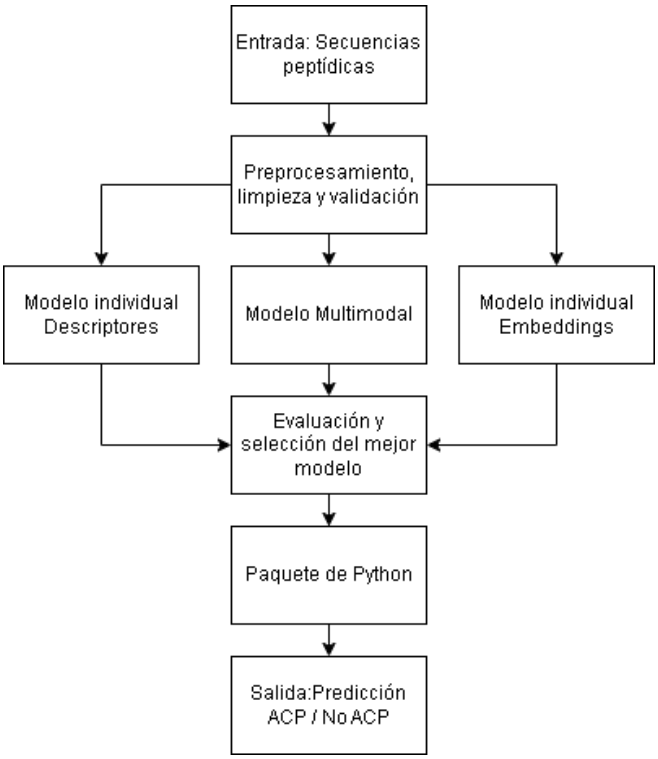
instalable localmente desde consola, lo que limita su adopción en flujos de trabajo bioinformáticos automatizados. El presente trabajo busca cubrir estas brechas mediante el desarrollo de una arquitectura que compare ESM-2 y ProtBERT y combine el mejor de ellos con descriptores fisicoquímicos, evaluada sobre el dataset de referencia de mACPpred 2.0 para permitir una comparación directa y objetiva con el estado del arte, e implementada como un paquete de Python instalable desde consola con dependencias mínimas.

4. Metodología

El desarrollo de este trabajo se estructuró en cinco fases secuenciales e interdependientes. El conjunto de datos utilizado corresponde al benchmark oficial de mACPpred 2.0 (Sangaraju et al., 2024), compuesto por 1.176 ACPs y 1.176 no-ACPs para entrenamiento y 610 ACPs y 2.760 no-ACPs para el conjunto de prueba independiente. Los modelos se entrenaron exclusivamente sobre el archivo de entrenamiento y se evaluaron exclusivamente sobre el de prueba, respetando la partición oficial. La generación de las representaciones de los modelos de lenguaje proteico se ejecutó en Google Colab Pro con GPU NVIDIA A100-SXM4-80GB; el cálculo de los descriptores y el entrenamiento de los clasificadores se ejecutaron en procesador, tanto en Colab como en el equipo local descrito en el Anexo C.3. Se emplearon las librerías scikit-learn, PyTorch y *HuggingFace Transformers*.

Figura 1

Estructura general del trabajo en cinco fases.



Nota. Las secuencias peptídicas se preprocesaron una sola vez y alimentaron tres modelos independientes: uno basado en descriptores fisicoquímicos, otro en representaciones contextuales y un tercero que integró ambos. Los tres se evaluaron bajo el mismo protocolo y el de mejor desempeño se implementó en el paquete de Python.

4.1 Fase 1 — Construcción del Modelo Predictivo con Descriptores Fisicoquímicos

El objetivo de esta fase fue construir un modelo de clasificación de ACPs basado exclusivamente en representaciones fisicoquímicas de las secuencias peptídicas, sin emplear modelos de lenguaje preentrenados.

4.1.1 Preprocesamiento y Cálculo de Descriptores

Las secuencias peptídicas fueron leídas desde archivos en formato FASTA y validadas para garantizar que contuvieran únicamente los 20 aminoácidos estándar. Los descriptores fisicoquímicos se calcularon utilizando las librerías peptides.py (Osorio et al., 2015) y propy3 (Cao et al., 2013), agrupados en cuatro categorías:

- **Propiedades globales (10):** peso molecular, carga neta a pH 7.0, punto isoeléctrico, hidrofobicidad (escalas Eisenberg, Kyte-Doolittle y Hopp-Woods), momento hidrofóbico con ángulos de 100° y 160°, índice de inestabilidad e índice alifático**.**
- **Composición de aminoácidos (AAC):** vector de 20 valores con la frecuencia relativa de cada aminoácido estándar.
- **Composición de dipéptidos (DPC):** vector de 400 valores con la frecuencia de cada par de aminoácidos consecutivos.
- **Descriptores CTD:** composición, transición y distribución de propiedades fisicoquímicas agrupadas por polaridad, volumen y solubilidad.

4.1.2 Entrenamiento y Evaluación

Las secuencias de Training.fasta (1.176 ACPs y 1.176 no-ACPs) fueron usadas para entrenamiento y las de Independent.fasta (610 ACPs y 2.760 no-ACPs) para evaluación. Los vectores de descriptores fueron normalizados mediante StandardScaler antes del entrenamiento. Se evaluaron siete clasificadores: SVM con kernel RBF, Random Forest, Extra Trees, Gradient Boosting, Regresión Logística, KNN y MLP. La selección de hiperparámetros se realizó mediante GridSearchCV con validación cruzada anidada de 5×5 pliegues optimizando AUC-ROC.

Los resultados completos de esta fase se presentan en la sección 5.2.

Figura 2

Flujo del proceso de clasificación de péptidos anticancerígenos mediante descriptores fisicoquímicos.



Nota. Las secuencias en formato FASTA se convirtieron en un vector de 577 descriptores (10 propiedades globales, 20 de composición de aminoácidos, 400 de composición de dipéptidos y 147 de composición, transición y distribución), normalizado mediante estandarización Z-score. Sobre esa representación se optimizaron siete clasificadores mediante validación cruzada anidada de cinco pliegues externos y cinco internos, entrenando con Training.fasta y evaluando sobre el conjunto de prueba independiente de mACPpred 2.0.

4.2 Fase 2 — Construcción del Modelo Predictivo con Embeddings de ESM-2

Esta fase tuvo como objetivo construir modelos predictivos basados en representaciones contextuales generadas por el modelo de lenguaje proteico ESM-2, evaluando el impacto de diferentes estrategias metodológicas afectaban el rendimiento final.

4.2.1 Generación de Embeddings

Se utilizó la variante esm2_t6_8M_UR50D de ESM-2, con 8 millones de parámetros y vectores de representación de 320 dimensiones, cargada desde HuggingFace Transformers (Wolf

et al., 2020). Las secuencias fueron tokenizadas y procesadas en lotes para aprovechar eficientemente la GPU disponible (NVIDIA A100-SXM4-80GB, Google Colab Pro).

Para la obtención del vector de representación final de cada péptido se aplicó mean pooling sobre los estados ocultos de la última capa del modelo, excluyendo del promedio el token especial de inicio (CLS), que no corresponde a ningún aminoácido y cuyo peso relativo en el promedio aumenta cuanto más corto es el péptido. Por un desplazamiento de índice en el enmascarado, detectado durante la verificación del paquete, el procedimiento excluyó además la posición del último aminoácido y conservó el token de fin de secuencia (EOS). Se mantuvo esta variante porque es la que produjo los resultados reportados y la que implementa el modelo distribuido; la exclusión exacta de CLS y EOS se evaluó por separado y obtuvo un AUC-ROC de 0.8704, un F1-score de 0.6243 y un MCC de 0.5355, valores prácticamente equivalentes (sección 5.5). Las secuencias de Training.fasta (1.176 ACPs y 1.176 no-ACPs) se usaron para entrenamiento y las de Independent.fasta (610 ACPs y 2.760 no-ACPs) para evaluación.

4.2.2 Entrenamiento y Evaluación

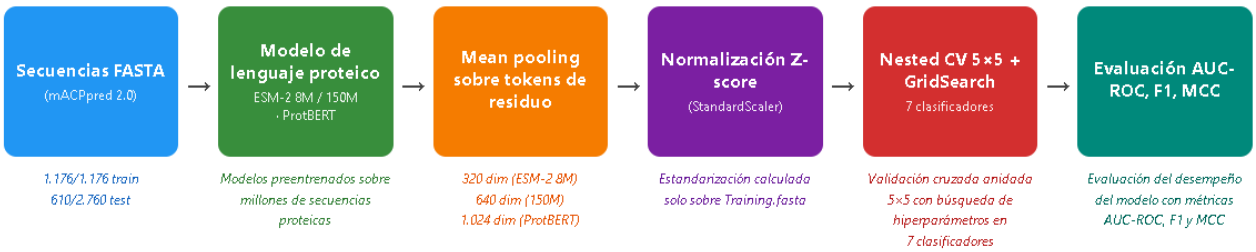
Los embeddings generados fueron normalizados mediante StandardScaler antes del entrenamiento. Se evaluaron los mismos siete clasificadores de la Fase 1: SVM con kernel RBF, Random Forest, Extra Trees, Gradient Boosting, Regresión Logística, KNN y MLP. La selección de hiperparámetros se realizó mediante GridSearchCV con validación cruzada anidada de 5×5 pliegues (Nested 5×5-Fold CV), optimizando AUC-ROC como métrica principal, dado que el conjunto de prueba independiente presenta un desbalance de clases significativo (18 % ACPs frente a 82 % no-ACPs) que hace a esta métrica más informativa que la exactitud (Saito y Rehmsmeier, 2015). La validación cruzada anidada se aplicó exclusivamente sobre Training.fasta: el bucle interno seleccionó los hiperparámetros y el externo estimó el desempeño sin sesgo de

selección. Fijada la configuración, cada clasificador se reentrenó con la totalidad de Training.fasta y el modelo resultante se evaluó una sola vez sobre Independent.fasta. El conjunto de prueba independiente no intervino en ningún momento en la validación cruzada ni en la selección de hiperparámetros.

Los resultados completos de esta fase se presentan en la sección 5.3.

Figura 3

Flujo del proceso de clasificación de péptidos anticancerígenos mediante representaciones contextuales (embeddings).



Nota. Las secuencias fueron procesadas por un modelo de lenguaje proteico (ESM-2: Lin et al., 2023; ProtBERT: Elnaggar et al., 2021), del que se obtuvo mediante mean pooling sobre los tokens de residuo un vector denso de 320, 640 o 1.024 dimensiones según la variante empleada. El vector se estandarizó y alimentó a los mismos siete clasificadores, optimizados con validación cruzada anidada de cinco pliegues externos y cinco internos. entrenando con Training.fasta y evaluando sobre el conjunto de prueba independiente de mACPpred 2.0.

4.3 Fase 3 — Construcción del Modelo Predictivo con Embeddings de ProtBERT

Con el objetivo de comparar el rendimiento de diferentes modelos de lenguaje proteico para la clasificación de ACPs, se repitió el proceso de generación de embeddings utilizando ProtBERT (variante prot_bert_bfd) como modelo alternativo a ESM-2.

4.3.1 Generación de Embeddings

El modelo prot_bert_bfd fue cargado desde HuggingFace Transformers utilizando *BertTokenizer* y *BertModel*. A diferencia de ESM-2, ProtBERT requiere que los aminoácidos de la secuencia estén separados por espacios antes de la tokenización, por lo que se implementó un preprocesamiento que transforma cada secuencia al formato requerido. El modelo genera vectores de representación de 1.024 dimensiones. Dado el mayor tamaño del modelo respecto a ESM-2 (aproximadamente 1.7 GB), el procesamiento se realizó en lotes de 8 secuencias para evitar errores de memoria en la GPU.

Para obtener la representación final de cada péptido se aplicó mean pooling sobre los estados ocultos de la última capa con el mismo enmascarado descrito en la sección 4.2.1, que excluye el token inicial —[CLS] en la tokenización de ProtBERT— y la última posición de residuo. Los embeddings resultantes se procesaron con el protocolo establecido en la sección 4.2.2, sin variación alguna: la validación cruzada anidada de 5×5 pliegues se aplicó únicamente sobre Training.fasta para seleccionar los hiperparámetros, y el modelo reentrenado sobre la totalidad de ese conjunto se evaluó una sola vez sobre Independent.fasta.

Los resultados completos de esta fase se presentan en la sección 5.4.

4.4 Fase 4 — Construcción del Modelo Multimodal

Esta fase integró las representaciones de descriptores fisicoquímicos y embeddings de ESM-2 8M mediante dos estrategias de fusión —concatenación directa de vectores y fusión tardía por promedio de probabilidades—, con el objetivo de determinar si la combinación de ambas representaciones mejora el rendimiento respecto a los modelos individuales.

4.4.1 Construcción del Vector Multimodal

El vector multimodal se construyó concatenando el vector de 577 descriptores fisicoquímicos de la Fase 1 con el vector de 320 dimensiones de embeddings de ESM-2 8M de la Fase 2, seleccionado por ser el embedding con mejor rendimiento individual sobre el conjunto de prueba independiente. La concatenación produjo un vector combinado de 897 dimensiones por péptido. Cada bloque fue normalizado por separado mediante StandardScaler antes de la concatenación, dado que sus escalas son sustancialmente distintas.

4.4.2 Entrenamiento y Evaluación

El vector concatenado fue evaluado con los mismos siete clasificadores, la misma rejilla de hiperparámetros y la misma metodología de Nested 5×5-Fold CV de las fases anteriores, garantizando comparabilidad directa. La fusión tardía, que exige recorrer la rejilla por separado para cada bloque y duplica el coste de cómputo, se evaluó sobre los tres clasificadores de mejor comportamiento en las fases anteriores.

Ninguna de las dos estrategias de integración superó al modelo basado únicamente en embeddings de ESM-2 8M; los resultados completos se presentan en la sección 5.6 y se analizan en la sección 6.5.

4.5 Fase 5 — Desarrollo del Paquete de Python

4.5.1 Selección e Integración del Modelo Final

Con base en los resultados obtenidos en las fases anteriores, se seleccionó la combinación de embeddings de ESM-2 8M (esm2_t6_8M_UR50D) con un clasificador de Regresión Logística como modelo final para su implementación en el paquete de Python. Esta selección se fundamentó en tres criterios: el mayor AUC-ROC obtenido sobre el conjunto de prueba independiente de mACPpred 2.0 (0.**8745**), que constituye el único protocolo comparable con el estado del arte; la menor complejidad computacional del clasificador, que reduce el riesgo de sobreajuste y facilita la interpretación del modelo; y los menores requisitos de despliegue del modelo de lenguaje empleado, de 31 MB frente a los 1.7 GB que exige ProtBERT. El modelo fue reentrenado sobre el conjunto de entrenamiento completo con los mejores hiperparámetros identificados, y serializado junto con el objeto StandardScaler mediante joblib para su distribución en el paquete.

4.5.2 Desarrollo del Paquete

Se desarrolló el paquete acp-classifier, instalable desde consola mediante:

```
pip install acp-classifier
```

El paquete recibe como entrada un archivo en formato FASTA con una o múltiples secuencias peptídicas y genera un archivo CSV con los resultados de predicción para cada péptido, incluyendo la etiqueta predicha (ACP o no-ACP) y la probabilidad asociada. Se incluyó un archivo de configuración de dependencias compatible con pip y Conda para garantizar la reproducibilidad del entorno de ejecución.

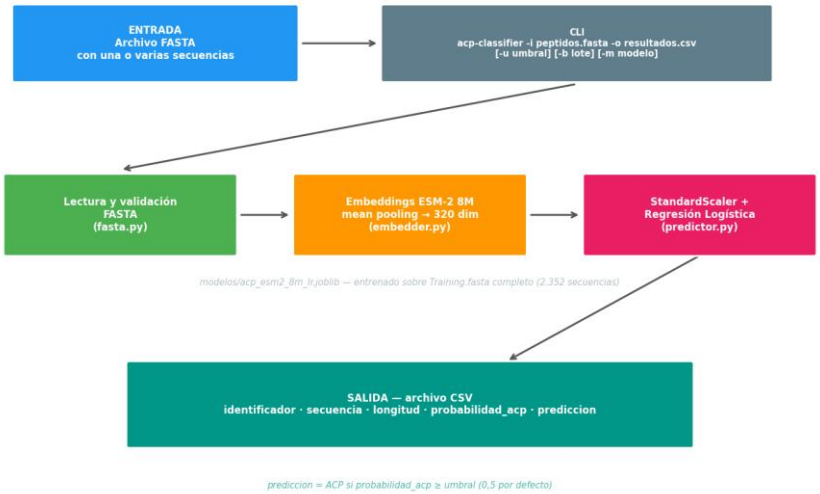
4.5.3 Documentación

Se generó documentación completa del código incluyendo README con instrucciones de instalación y uso, ejemplos de uso con datasets de prueba, y notebooks reproducibles que ilustran el flujo completo desde la instalación hasta la obtención de resultados en diferentes entornos bioinformáticos.

La fase concluyó con dos productos. El primero es el paquete acp-classifier, funcional e instalable desde consola, que integra la combinación de embeddings de ESM-2 8M con un clasificador de Regresión Logística, seleccionada por su AUC-ROC sobre el conjunto de prueba independiente de mACPpred 2.0. El segundo es la propia herramienta documentada, replicable y de libre acceso, publicada en un repositorio público para su uso en investigaciones posteriores.

Figura 4

Arquitectura del paquete acp-classifier para la clasificación de péptidos anticancerígenos desde línea de comandos.



Nota. El usuario provee un archivo FASTA desde la interfaz de línea de comandos; el paquete valida las secuencias, genera sus representaciones con ESM-2 8M mediante mean pooling, aplica los parámetros del modelo de Regresión Logística distribuidos con el paquete y escribe un archivo CSV con el identificador, la secuencia, su longitud, la probabilidad estimada de actividad anticancerígena y la clasificación resultante.

El paquete final integra el modelo de mejor desempeño identificado en la fase de evaluación, que resultó ser la combinación de embeddings de ESM-2 8M con un clasificador de Regresión Logística.

4.6 Alcance y Nivel de Madurez Tecnológica

El presente proyecto contempló el desarrollo y la evaluación comparativa de modelos de clasificación basados en dos enfoques de representación —descriptores fisicoquímicos y embeddings de modelos de lenguaje proteico (ESM-2 y ProtBERT)—, así como una arquitectura multimodal que combina ambas representaciones. El paquete final de Python integró el modelo de clasificación con mejor desempeño identificado en la evaluación comparativa sobre el conjunto de prueba independiente. El nivel de madurez tecnológica (Technology Readiness Level, TRL) alcanzado corresponde a TRL 4: validación del componente en entorno de laboratorio computacional (Mankins, 1995). El alcance se limita a la validación computacional sobre datos de referencia públicos; la validación experimental *in vitro* o *in vivo* de los péptidos identificados queda fuera del alcance y se plantea como trabajo futuro.

5. Resultados

5.1 Caracterización del Conjunto de Datos de Referencia

El análisis del conjunto de datos de mACPpred 2.0 reveló dos características con implicaciones directas sobre la interpretación de los resultados: una diferencia sistemática en la longitud de las secuencias entre los conjuntos de entrenamiento y prueba, y un perfil de composición aminoacídica que confirma sobre los datos propios el mecanismo de selectividad descrito en el marco conceptual.

5.1.1 Distribución de Longitudes de Secuencia

La Tabla 2 presenta la distribución de longitudes de las secuencias peptídicas en cada uno de los subconjuntos. Los péptidos anticancerígenos del conjunto de prueba independiente presentan una longitud mediana de 9 residuos, frente a los 20 residuos de los péptidos anticancerígenos del conjunto de entrenamiento. Esta diferencia implica que los modelos son entrenados sobre un régimen de longitud y evaluados sobre otro sustancialmente distinto, condición conocida en la literatura como desplazamiento de covariables (covariate shift).

Tabla 2

Distribución de longitudes de secuencia (residuos) por subconjunto.

Subconjunto	n	Media	Mediana	Mínimo	Máximo	P95
Entrenamiento — ACP	1.176	22.5	20	6	50	43
Entrenamiento — no-ACP	1.176	26.0	25	6	50	46
Prueba independiente — ACP	610	11.7	9	6	47	28
Prueba independiente — no-ACP	2.760	21.6	19	6	50	39

Esta observación resulta relevante por tres motivos. En primer lugar, explica por qué una división estratificada del conjunto combinado —ensayada durante el desarrollo y descartada por las razones que se exponen en la sección 6.1— produce estimaciones artificialmente optimistas: al mezclar ambos archivos antes de particionar, las dos distribuciones de longitud quedan repartidas entre entrenamiento y prueba, y el problema se vuelve más fácil de lo que es. En segundo lugar, condiciona el comportamiento de los descriptores basados en composición: la composición de dipéptidos genera un vector de 400 posiciones, mientras que un péptido de nueve residuos aporta únicamente ocho dipéptidos observados, produciendo representaciones extremadamente dispersas. En tercer lugar, ofrece una explicación estructural al desempeño comparativo de los modelos de lenguaje proteico de distinta capacidad, discutido en la sección 5.5.

5.1.2 Composición Aminoacídica y Propiedades Fisicoquímicas

La Tabla 3 presenta los cinco aminoácidos más enriquecidos y los cuatro más empobrecidos en los péptidos anticancerígenos respecto de los no anticancerígenos, calculados sobre las 5.722 secuencias del conjunto completo.

Tabla 3

Composición aminoacídica diferencial entre ACPs y no-ACPs.

Aminoácido	ACP (%)	no-ACP (%)	Diferencia	Razón
C — Cisteína	6.55	1.93	+4.62	3.40
K — Lisina	10.54	6.45	+4.09	1.63
G — Glicina	9.03	7.25	+1.78	1.25
R — Arginina	7.41	5.99	+1.43	1.24
W — Triptófano	2.58	1.34	+1.24	1.92
Q — Glutamina	2.39	3.96	−1.57	0.60
V — Valina	4.75	6.68	−1.92	0.71
D — Ácido aspártico	2.59	4.97	−2.38	0.52
E — Ácido glutámico	2.55	6.17	−3.63	0.41

El perfil obtenido concuerda con el mecanismo de acción descrito en la sección 3.1.3. Los residuos catiónicos lisina y arginina se encuentran enriquecidos en los péptidos anticancerígenos (razones de 1.63 y 1.24 respectivamente), mientras que los residuos aniónicos ácido aspártico y ácido glutámico aparecen reducidos aproximadamente a la mitad (razones de 0.52 y 0.41). Este patrón es coherente con la hipótesis de que la selectividad de los ACPs hacia membranas tumorales tiene un componente electrostático dominante (Bauso et al., 2024; Chinnadurai et al., 2023). El enriquecimiento observado en cisteína sugiere además una mayor presencia de péptidos estabilizados mediante puentes disulfuro, aunque su caracterización estructural excede el alcance del presente trabajo.

El análisis de propiedades globales, calculado sobre las 5.722 secuencias del conjunto completo (1.786 ACPs y 3.936 no-ACPs), confirma cuantitativamente esta interpretación (Tabla 4).

Tabla 4

Propiedades fisicoquímicas globales por clase (media ± desviación típica).

Propiedad	**ACP**	**no-ACP**
Carga neta a pH 7.0	+2.37 ± 3.24	+0.32 ± 2.84
Hidrofobicidad (Eisenberg)	0.00 ± 0.35	0.00 ± 0.28
Momento hidrofóbico (ángulo 100°)	0.47 ± 0.22	0.44 ± 0.16
Secuencias con carga neta positiva	74.8 %	52.3 %

La carga neta media de los péptidos anticancerígenos supera en 2.05 unidades a la de los no anticancerígenos (+2.37 frente a +0.32), y la proporción de secuencias con carga neta positiva a pH fisiológico asciende al 74.8 % frente al 52.3 %. En contraste, la hidrofobicidad media es prácticamente idéntica en ambas clases, y la diferencia en el momento hidrofóbico, aunque estadísticamente detectable dado el número de secuencias, es pequeña (0.47 frente a 0.44), lo que sugiere que la carga constituye el factor discriminante dominante entre las propiedades globales

evaluadas. Este resultado respalda la pertinencia de los descriptores fisicoquímicos como representación de partida del problema.

5.2 Modelos Basados en Descriptores Fisicoquímicos

Los descriptores fisicoquímicos fueron evaluados sobre el conjunto de prueba independiente de mACPpred 2.0, entrenando exclusivamente con Training.fasta y aplicando el mismo esquema de validación cruzada anidada empleado en las fases posteriores. La Tabla 5 recoge los resultados.

Tabla 5

Descriptores fisicoquímicos (577 dimensiones) sobre el conjunto de prueba independiente de mACPpred 2.0.

Clasificador	AUC-ROC CV	AUC-ROC Test	F1 Test	Exactitud Test	MCC Test
SVM RBF	**0.8834 ± 0.0138**	**0.8416**	**0.5669**	**0.8136**	**0.4614**
Gradient Boosting	0.8900 ± 0.0138	0.8228	0.5112	0.7861	0.3889
Extra Trees	0.8918 ± 0.0131	0.8207	0.4915	0.7881	0.3655
Random Forest	0.8895 ± 0.0132	0.8151	0.5068	0.7834	0.3831
Logistic Regression	0.8680 ± 0.0146	0.7968	0.5098	0.7843	0.3870
MLP	0.8569 ± 0.0105	0.7575	0.4780	0.7680	0.3451
KNN	0.8640 ± 0.0217	0.7163	0.4100	0.6849	0.2477

Mejor resultado obtenido: **SVM RBF** con AUC-ROC de **0.8416**, F1-score de **0.5669** y MCC de **0.4614** sobre el conjunto de prueba independiente de mACPpred 2.0.

5.3 Modelos Basados en Embeddings de ESM-2

La Tabla 6 presenta el desempeño de los siete clasificadores evaluados sobre los embeddings de 320 dimensiones generados con la variante esm2_t6_8M_UR50D, aplicando mean pooling con exclusión del token CLS y de la última posición de residuo (sección 4.2.1).

Tabla 6

ESM-2 8M sobre el conjunto de prueba independiente de mACPpred 2.0.

Clasificador	AUC-ROC CV	AUC-ROC Test	F1 Test	Exactitud Test	MCC Test
Logistic Regression	**0.8923 ± 0.0152**	**0.8745**	**0.6230**	**0.8362**	**0.5341**
Random Forest	0.8938 ± 0.0126	0.8681	0.6213	0.8315	0.5330
Extra Trees	0.8948 ± 0.0153	0.8670	0.6115	0.8341	0.5189
SVM RBF	0.8981 ± 0.0117	0.8654	0.5958	0.8297	0.4985
Gradient Boosting	0.8880 ± 0.0128	0.8598	0.5988	0.8234	0.5031
KNN	0.8904 ± 0.0116	0.8250	0.5436	0.7997	0.4311
MLP	0.8826 ± 0.0110	0.8233	0.5298	0.7819	0.4133

Mejor resultado obtenido: **Regresión Logística** con AUC-ROC de **0.8745**, F1-score de **0.6230** y MCC de **0.5341** sobre el conjunto de prueba independiente. El modelo no supera a las dos herramientas de referencia más recientes evaluadas sobre ese conjunto, MLACP 2.0 (AUC-ROC de **0.893**) y mACPpred 2.0 (AUC-ROC de **0.887**), de las que lo separan **0.019** y **0.013** puntos de AUC-ROC respectivamente; sí supera a las nueve herramientas restantes con AUC-ROC publicado sobre el mismo conjunto. Las más competitivas entre ellas —AMPfun (0.802), AntiCP 2.0 en su variante alterna (0.835), mACPpred (0.842) e iDACP (0.852)— quedan todas por debajo del límite inferior del intervalo de confianza del modelo (0.859, sección 6.6); el resto presenta valores inferiores a 0.75, incluidas dos variantes de ACPred-BMF con valores por debajo de 0.5. La ordenación se mantiene en F1-score y MCC, donde el modelo ocupa también el tercer lugar, por detrás de las mismas dos herramientas. La diferencia frente a estas debe leerse considerando que mACPpred 2.0 es una arquitectura de aprendizaje profundo apilado que integra múltiples representaciones de modelos de lenguaje y decenas de características probabilísticas, y MLACP 2.0 un metapredictor que combina, mediante una red convolucional, las salidas de múltiples clasificadores, frente a un único modelo de ocho millones de parámetros con un clasificador lineal. Resulta destacable que el modelo lineal más simple del conjunto evaluado obtenga el mejor rendimiento, comportamiento que se analiza en el capítulo de discusión. Se observa asimismo una

diferencia sistemática entre las métricas de validación cruzada (AUC-ROC en torno a 0.89) y las obtenidas sobre el conjunto independiente (AUC-ROC en torno a 0.87), consistente con el desplazamiento de distribución caracterizado en la sección 5.1.1.

5.4 Modelos Basados en Embeddings de ProtBERT

Con el objetivo de comparar el desempeño de distintos modelos de lenguaje proteico, se repitió el procedimiento empleando ProtBERT en su variante prot_bert_bfd, que genera vectores de 1.024 dimensiones. Los resultados se presentan en la Tabla 7.

Tabla 7

ProtBERT (prot_bert_bfd) sobre el conjunto de prueba independiente.

Clasificador	AUC-ROC CV	AUC-ROC Test	F1 Test	Exactitud Test	MCC Test
SVM RBF	**0.8884 ± 0.0173**	**0.8517**	**0.5827**	**0.8181**	**0.4819**
Logistic Regression	0.8854 ± 0.0149	0.8370	0.5658	0.8178	0.4601
Extra Trees	0.8907 ± 0.0171	0.8241	0.5363	0.7973	0.4216
Gradient Boosting	0.8844 ± 0.0154	0.8238	0.5298	0.7920	0.4131
Random Forest	0.8868 ± 0.0171	0.8212	0.5447	0.7961	0.4326
KNN	0.8895 ± 0.0186	0.8032	0.5220	0.7908	0.4030
MLP	0.8775 ± 0.0138	0.7967	0.5079	0.7780	0.3841

Mejor resultado obtenido: SVM con kernel RBF con AUC-ROC de **0.8517**, F1-score de **0.5827** y MCC de **0.4819**. ProtBERT, pese a disponer de una capacidad considerablemente mayor que ESM-2 8M —420 millones de parámetros frente a 8 millones, y **1.024** dimensiones de representación frente a 320—, obtuvo un rendimiento inferior en **0.0228** puntos de AUC-ROC sobre el mismo conjunto de evaluación.

5.5 Análisis de Ablación: Capacidad del Modelo y Estrategia de Pooling

Con el fin de aislar la contribución de las distintas decisiones de diseño adoptadas durante la fase de embeddings, se compararon cuatro configuraciones evaluadas sobre el mismo conjunto de prueba independiente. La Tabla 8 resume los mejores resultados de cada configuración.

Tabla 8

Análisis de ablación sobre el conjunto de prueba independiente.

Configuración	Dim.	Pooling	Mejor clasificador	AUC-ROC Test
ESM-2 150M	640	Todos los tokens	Extra Trees	0.8477
ESM-2 8M	320	Todos los tokens	Extra Trees	0.8554
ESM-2 8M	320	Sin CLS ni último residuo	Logistic Regression	0.8745
ProtBERT	1.024	Sin CLS ni último residuo	SVM RBF	0.8517

De forma complementaria, la Tabla 9 documenta la evolución metodológica de la configuración basada en ESM-2 8M, desde la evaluación inicial sin optimización de hiperparámetros hasta la configuración final.

Tabla 9

Evolución metodológica de la configuración basada en ESM-2 8M.

Configuración metodológica	AUC-ROC Test
ESM-2 8M sin optimización de hiperparámetros	0.8390
ESM-2 8M con búsqueda en malla (GridSearchCV)	0.8555
ESM-2 8M con validación cruzada anidada, sin SMOTE	0.8554
ESM-2 8M con validación cruzada anidada, SMOTE y pooling sin CLS ni último residuo	0.8745

Dos observaciones se desprenden de esta progresión. La optimización de hiperparámetros aportó una mejora de 0.0165 puntos respecto de la configuración inicial, mientras que la adopción de la validación cruzada anidada no modificó el desempeño sobre el conjunto de prueba (0.8555 frente a 0.8554); esto último es esperable, dado que la finalidad de dicho esquema no es mejorar el rendimiento sino eliminar el sesgo optimista en su estimación (Varma y Simon, 2006).

La última transición de la tabla incorpora simultáneamente dos modificaciones —la aplicación de SMOTE y el cambio en la estrategia de pooling—, lo que en principio impediría atribuir la mejora resultante a una u otra. Sin embargo, en este caso particular ambas contribuciones sí resultan separables, dado que el efecto de SMOTE sobre este protocolo pudo verificarse de forma independiente: al estar el conjunto de entrenamiento balanceado por diseño (1.176 secuencias de cada clase), el algoritmo no genera ninguna muestra sintética, ni sobre el conjunto completo ni dentro de los pliegues de la validación cruzada, cuyas particiones estratificadas preservan el balance. Al ser SMOTE una operación de efecto nulo bajo estas condiciones, la totalidad de la mejora de 0.0191 puntos es atribuible al cambio en la estrategia de pooling. Este razonamiento se desarrolla con mayor detalle en la sección 6.4.

El análisis permite extraer dos conclusiones. En primer lugar, el incremento de capacidad del modelo de lenguaje no se traduce en mejor desempeño: bajo condiciones idénticas de pooling y validación, la variante de 8 millones de parámetros superó a la de 150 millones en los dos clasificadores para los que se completó la comparación —Random Forest (0.8522 frente a 0.8454) y Extra Trees (0.8554 frente a 0.8477)—, y ProtBERT, el modelo de mayor capacidad evaluado, se situó por debajo de la mejor configuración de ESM-2 8M. En segundo lugar, la mayor mejora observada en toda la fase, de 0.0191 puntos de AUC-ROC, no provino del tamaño del modelo sino de la estrategia de agregación: excluir del mean pooling el token CLS —junto con la última posición de residuo, según se detalla en la sección 4.2.1— elevó el AUC-ROC de 0.8554 a 0.8745 sobre la misma representación de 320 dimensiones. La exclusión exacta de CLS y EOS, evaluada por separado, alcanza 0.8704, de modo que la mejora se sostiene con cualquiera de las dos variantes. La comparación clasificador a clasificador confirma que el efecto no se debe al cambio de clasificador: con el nuevo pooling, la Regresión Logística pasó de 0.8189 a 0.8745, SVM RBF

de 0.8055 a 0.8654, Random Forest de 0.8522 a 0.8681, Gradient Boosting de 0.8241 a 0.8598 y Extra Trees de 0.8554 a 0.8670; KNN fue la única excepción, con un ligero descenso de 0.8303 a 0.8250. Este resultado es coherente con que, en péptidos cortos, los tokens especiales —en particular CLS— introduzcan una distorsión apreciable en el vector de representación, dado que su peso relativo dentro del promedio es inversamente proporcional a la longitud de la secuencia.

5.6 Modelo Multimodal

Esta sección presenta los resultados de la Fase 4, dirigida al tercer objetivo específico: determinar si la integración de descriptores fisicoquímicos y embeddings de ESM-2 8M mejora el rendimiento respecto de cada representación por separado. Se evaluaron dos estrategias de fusión —concatenación directa de vectores y fusión tardía por promedio de probabilidades— bajo el mismo esquema de validación cruzada anidada 5×5 y sobre el mismo conjunto de prueba independiente empleado en las secciones anteriores, de modo que las cifras sean directamente comparables con las de las Tablas 5, 6 y 7.

El bloque de descriptores fisicoquímicos fue verificado antes del entrenamiento, confirmando 577 descriptores sin valores ausentes ni infinitos y sin ninguna columna de varianza nula. La concatenación con los 320 embeddings de ESM-2 8M produjo un vector combinado de 897 dimensiones por péptido.

La Tabla 10 recoge el rendimiento de los siete clasificadores sobre el vector concatenado de 897 dimensiones. La columna de F1 en validación cruzada corresponde al F1-score binario sobre la clase positiva, promediado sobre los cinco pliegues externos.

Tabla 10

Modelo multimodal por concatenación directa (897 dimensiones) sobre el conjunto de prueba independiente.

Clasificador	AUC-ROC CV	F1 CV	AUC-ROC Test	F1 Test	Exactitud Test	MCC Test
SVM RBF	**0.8937 ± 0.0122**	**0.8168**	**0.8676**	**0.6197**	**0.8365**	**0.5296**
Random Forest	0.8991 ± 0.0129	0.8190	0.8657	0.6157	0.8315	0.5250
Extra Trees	0.8988 ± 0.0149	0.8228	0.8622	0.6033	0.8341	0.5081
Gradient Boosting	0.8943 ± 0.0132	0.8178	0.8568	0.6045	0.8214	0.5116
Logistic Regression	0.8883 ± 0.0143	0.8163	0.8413	0.5661	0.8217	0.4610
KNN	0.8914 ± 0.0158	0.8148	0.7952	0.4969	0.7567	0.3697
MLP	0.8774 ± 0.0097	0.7993	0.7732	0.4983	0.7819	0.3724

El mejor resultado correspondió a SVM RBF, con un AUC-ROC de 0.8676 sobre el conjunto de prueba independiente. El SVM y los tres clasificadores basados en árboles se situaron en una banda estrecha, entre 0.8568 y 0.8676, mientras que KNN y MLP quedaron claramente por debajo, con 0.7952 y 0.7732. La Regresión Logística, que es el mejor clasificador sobre ESM-2 8M en solitario, cedió posiciones al incorporar el bloque de descriptores y descendió a 0.8413.

La segunda estrategia entrena un clasificador independiente sobre cada bloque y promedia las probabilidades que ambos emiten, con un peso de 0.5 para cada uno. Como exige recorrer la rejilla de hiperparámetros por separado para cada bloque, su coste computacional se duplica, por lo que se evaluó sobre los tres clasificadores con mejor comportamiento en las fases anteriores y no sobre los siete. La Tabla 11 presenta los resultados.

Tabla 11

Estrategia de fusión tardía (w = 0.5) sobre el conjunto de prueba independiente.

Clasificador	AUC-ROC CV	F1 CV	AUC-ROC Test	F1 Test	Exactitud Test	MCC Test
SVM RBF	**0.9031 ± 0.0105**	**0.8275**	**0.8731**	**0.6102**	**0.8362**	**0.5169**
Extra Trees	0.9020 ± 0.0133	0.8155	0.8584	0.5931	0.8261	0.4951
Logistic Regression	0.8921 ± 0.0125	0.8228	0.8564	0.5953	0.8273	0.4979

La fusión tardía con SVM RBF alcanzó el mejor AUC-ROC de toda la fase, 0.8731, y también el mejor valor en validación cruzada, 0.9031 ± 0.0105. Superó a la concatenación directa en AUC-ROC, aunque no en F1-score ni en MCC, donde la concatenación con SVM RBF conservó la ventaja: 0.6197 y 0.5296 frente a 0.6102 y 0.5169.

Para explorar el aporte de cada representación se examinaron los coeficientes de la Regresión Logística ajustada sobre el vector concatenado, agrupando su valor absoluto según el bloque de origen. Los 577 descriptores fisicoquímicos acumulan el 61.2 % del peso total, con una suma de coeficientes absolutos de 6.48, frente al 38.8 % de las 320 dimensiones de ESM-2 8M, con 4.11; por unidad de dimensión la relación se invierte, con una media de 0.0112 frente a 0.0129.

Esta lectura es solo descriptiva y no debe interpretarse como una medida de importancia. La regularización L2 reparte el peso entre variables correlacionadas y nunca anula coeficientes, de modo que el reparto refleja en buena medida el número de dimensiones de cada bloque. Además, en la concatenación la Regresión Logística ocupa el quinto lugar de siete (0.8413), por lo que no representa a la mejor configuración multimodal. La evidencia más directa del aporte de cada bloque la ofrece la comparación entre modelos: con SVM RBF, añadir los descriptores a los embeddings mejora ligeramente el AUC-ROC —de 0.8654 a 0.8676 por concatenación y a 0.8731 por fusión tardía—, mientras que con la Regresión Logística lo reduce, de 0.8745 a 0.8413.

Ninguna de las dos estrategias de integración superó al modelo basado únicamente en embeddings de ESM-2 8M, que alcanzó un AUC-ROC de 0.8745 sobre el mismo conjunto de prueba (Tabla 6). La diferencia es pequeña —0.0014 respecto de la fusión tardía y 0.0069 respecto de la concatenación— y no resulta estadísticamente significativa (sección 6.6), pero se produce en la dirección contraria a la esperada en las dos estrategias. La integración multimodal no aporta, por tanto, una mejora de rendimiento que justifique el coste adicional de calcular 577 descriptores en tiempo de inferencia, y el modelo final del trabajo se mantiene sobre ESM-2 8M. El tercer objetivo específico queda cumplido: se construyeron y evaluaron dos modelos multimodales bajo un protocolo comparable al de las representaciones individuales, y el resultado, aún siendo negativo, constituye en sí mismo un hallazgo del trabajo.

5.7 Comparación Global y Selección del Modelo Final

La Tabla 12 reúne el mejor resultado de cada representación evaluada, junto con los valores reportados por las herramientas de referencia sobre el mismo conjunto de prueba independiente. Todas las filas se evaluaron sobre el mismo conjunto de prueba independiente, que sus autores construyeron sin solapamiento con los datos de entrenamiento de las herramientas existentes (Sangaraju et al., 2024). Cada herramienta, no obstante, se entrenó con sus propios datos, de modo que la comparación es directa en cuanto al conjunto de evaluación, pero no controla las diferencias en los datos de entrenamiento.

Tabla 12

Comparación global sobre el conjunto de prueba independiente de mACPpred 2.0.

Representación	Mejor clasificador	AUC-ROC	F1-score	MCC
Descriptores fisicoquímicos (577 dim)	SVM RBF	0.8416	0.5669	0.4614

Representación	Mejor clasificador	AUC-ROC	F1-score	MCC
ESM-2 8M (320 dim)	**Logistic Regression**	**0.8745**	**0.6230**	**0.5341**
ProtBERT (1.024 dim)	SVM RBF	0.8517	0.5827	0.4819
Multimodal, concatenación (897 dim)	SVM RBF	0.8676	0.6197	0.5296
Multimodal, fusión tardía (w = 0.5)	SVM RBF	0.8731	0.6102	0.5169
mACPpred 2.0 (referencia)	SDL + CNN	0.887	0.671	0.597
MLACP 2.0 (referencia)	Meta-predictor	0.893	0.639	0.557

Tabla 13

Herramientas de referencia con AUC-ROC publicado sobre el conjunto de prueba independiente de mACPpred 2.0.

Herramienta	Enfoque	AUC-ROC	F1-score
MLACP 2.0	Meta-predictor	0.893	0.639
mACPpred 2.0	SDL + CNN	0.887	0.671
ESM-2 8M (este trabajo)	**Regresión Logística**	**0.8745**	**0.6230**
iDACP	SVM	0.852	0.549
mACPpred	Meta-predictor	0.842	0.586
AntiCP 2.0 (Alternate)	Extra Trees	0.835	0.422
AMPfun	Random Forest	0.802	0.537
ACPred	SVM	0.739	0.426
TriNet	Red neuronal	0.718	0.383
iAMPCN	Red convolucional	0.681	0.230
CancerGram	Random Forest	0.673	0.114
AntiCP 2.0 (Main)	Extra Trees	0.597	0.351
ACPred-BMF (Main)	Red neuronal	0.326	0.378
ACPred-BMF (Alternate)	Red neuronal	0.263	0.399

Nota. Valores publicados por Sangaraju et al. (2024, Tabla 1) sobre el mismo conjunto de prueba independiente. Se incluyen las once herramientas con AUC-ROC reportado; iDACP figura una sola vez porque sus cinco variantes de umbral comparten AUC-ROC, que es independiente del umbral de decisión. Las herramientas sin AUC-ROC publicado sobre este conjunto (ACP-MHCNN, iAMP-RAAC y PreTP-2L) no aparecen. En negrita, el modelo de este trabajo.

Con base en estos resultados se seleccionó la combinación de embeddings de ESM-2 8M con un clasificador de Regresión Logística como modelo final para su integración en el paquete

de Python. La selección se fundamenta en tres criterios: el mayor AUC-ROC obtenido sobre el conjunto de prueba independiente, que es el único protocolo comparable con el estado del arte; la menor complejidad computacional entre las alternativas evaluadas, tanto en el modelo de lenguaje empleado como en el clasificador; y la reducción de requisitos de despliegue que supone distribuir un modelo de 31 MB frente a los 1.7 GB que exige ProtBERT. La Tabla 13 sitúa el modelo entre las once herramientas con AUC-ROC publicado sobre este mismo conjunto: queda por delante de nueve de ellas y solo por detrás de MLACP 2.0 y mACPpred 2.0.

Conviene precisar que el criterio de selección empleó el conjunto de prueba independiente y no la validación cruzada. En validación cruzada, las configuraciones de cada representación se sitúan en una banda estrecha, entre 0.88 y 0.90 de AUC-ROC, con desviaciones típicas en torno a 0.015 que hacen indistinguibles sus diferencias: la Regresión Logística sobre ESM-2 8M (0.8923 ± 0.0152) queda a 0.006 puntos del SVM RBF sobre la misma representación (0.8981 ± 0.0117). Además, la validación cruzada se calcula sobre la distribución de longitudes del conjunto de entrenamiento, distinta de la del conjunto de prueba (sección 5.1.1), por lo que su ordenación no anticipa la del conjunto independiente. Entre configuraciones equivalentes en validación cruzada, el criterio de parsimonia respalda además la elección del clasificador más simple. Seleccionar sobre el conjunto de prueba introduce, no obstante, un sesgo optimista en la cifra de 0.8745, que se reconoce en el capítulo de Limitaciones y cuya incertidumbre se cuantifica en la sección 6.6.

5.8 Viabilidad Computacional

Dado que uno de los objetivos del trabajo es la distribución del modelo como paquete instalable, se evaluaron los requisitos computacionales de cada alternativa (Tabla 14).

Tabla 14

Requisitos computacionales de los modelos de lenguaje evaluados.

Modelo	Parámetros	Tamaño en disco	Dimensiones
ESM-2 8M (esm2_t6_8M_UR50D)	8 millones	≈ 31 MB	320
ESM-2 150M (esm2_t30_150M_UR50D)	150 millones	≈ 600 MB	640
ProtBERT (prot_bert_bfd)	420 millones	≈ 1.7 GB	1.024

El cálculo de los 577 descriptores fisicoquímicos para las 2.352 secuencias de entrenamiento y las 3.370 de prueba se realizó sobre un procesador AMD Ryzen 7 5700G sin aceleración por GPU, con un tiempo de procesamiento de aproximadamente 12 segundos. La generación de embeddings con ESM-2 8M y ProtBERT requirió aceleración por GPU y se ejecutó en Google Colab Pro con una unidad NVIDIA A100-SXM4-80GB de 80 GB de memoria; el entrenamiento y la evaluación de los clasificadores sobre los embeddings generados se realizó posteriormente en CPU. El modelo final basado en ESM-2 8M permite su ejecución en inferencia sin GPU, lo que facilita su distribución como paquete instalable en equipos sin hardware especializado, a diferencia de alternativas como ProtBERT que requieren aproximadamente 1.7 GB de almacenamiento y mayor capacidad de memoria para su carga.

6. Discusión

6.1 Sobre el Efecto del Protocolo de Evaluación

Durante el desarrollo se ensayó una vía de evaluación alternativa que finalmente se descartó: una división estratificada 80/20 construida sobre la unión de los dos archivos del benchmark. Se documenta aquí porque las razones de su descarte constituyen, en sí mismas, un

aporte metodológico del trabajo. La Tabla 15 resume en qué se diferencia del protocolo finalmente adoptado.

Tabla 15

Diferencias entre la vía de evaluación descartada y el protocolo adoptado.

Aspecto	**Exploración descartada — división 80/20**	**Protocolo adoptado — conjunto independiente**
Conjunto de partida	5.722 secuencias (ambos archivos combinados)	Training.fasta y Independent.fasta por separado
Entrenamiento	4.577 secuencias (31 % / 69 %)	2.352 secuencias (50 % / 50 %)
Evaluación	1.145 secuencias, partición estratificada	3.370 secuencias del conjunto oficial (18 % / 82 %)
Descriptores fisicoquímicos	70 características con selección por ANOVA	577 descriptores, sin selección
Esquema de validación	Validación cruzada anidada 10×5	Validación cruzada anidada 5×5
Métrica de selección	Exactitud (accuracy)	AUC-ROC
Remuestreo	SMOTE (1.719 muestras sintéticas)	SMOTE (ninguna muestra sintética generada)
Pooling de embeddings	Todos los tokens	Sin CLS ni último residuo
Comparable con el estado del arte	No	Sí

La diferencia determinante es la construcción del conjunto de evaluación. Aquella vía reconstruía una partición propia a partir de la combinación de ambos archivos, de modo que secuencias pertenecientes al conjunto de prueba oficial pasaban a formar parte del entrenamiento; sus métricas describen, por tanto, un problema distinto del que resuelven las herramientas de referencia. A ello se añade que tampoco era reproducible: el procedimiento eliminaba duplicados mediante una estructura de conjunto cuyo orden varía entre ejecuciones, por lo que la partición seleccionaba secuencias distintas cada vez pese a fijar la semilla aleatoria. El protocolo adoptado respeta la partición oficial y es el único cuyos resultados se reportan en el capítulo de Resultados.

Las cifras de aquella vía no se contrastan aquí con las del protocolo adoptado, porque ambas difieren en más de un aspecto a la vez: además de la construcción del conjunto de evaluación, la vía descartada empleó un conjunto de descriptores distinto —70 características con

selección por ANOVA, frente a los 577 descriptores de la Fase 1— y optimizó la exactitud en lugar del AUC-ROC. Sus resultados no permiten, por tanto, cuantificar el efecto del protocolo de evaluación. La lección que sí se desprende es de orden práctico: los resultados de las herramientas de predicción de ACPs solo admiten comparación cuando se obtienen sobre particiones idénticas, y la práctica de reconstruir divisiones propias sobre conjuntos de referencia dificulta la evaluación acumulativa del progreso.

6.2 Sobre el Desempeño Comparativo de los Modelos de Lenguaje Proteico

Los resultados no muestran una ventaja asociada a la capacidad del modelo de lenguaje: ESM-2 8M superó tanto a ESM-2 150M como a ProtBERT, el modelo de mayor capacidad evaluado. La observación debe leerse con cautela, porque ESM-2 150M solo se evaluó con el pooling sobre todos los tokens —con el que, bajo búsqueda en rejilla simple —protocolo distinto del de la Tabla 8, que emplea validación cruzada anidada—, igualó a ESM-2 8M, 0.8555 en ambos casos— y, bajo validación cruzada anidada, únicamente con dos clasificadores; ProtBERT, además, pertenece a otra familia de modelos con otro corpus de preentrenamiento, de modo que la comparación no aísla el efecto del tamaño. Dos factores resultan compatibles con los datos. Por un lado, los parámetros del clasificador se estiman sobre solo 2.352 ejemplos de entrenamiento, y una representación de 640 o 1.024 dimensiones aumenta el riesgo de sobreajuste frente a una de 320. Por otro, los modelos de lenguaje proteico se preentrenaron sobre proteínas completas, de cientos de residuos, mientras que los péptidos anticancerígenos del conjunto de prueba presentan una longitud mediana de nueve; sobre secuencias tan cortas, el contexto que el modelo puede explotar es mínimo y el promedio de los estados ocultos descansa sobre muy pocas posiciones. Ninguna de las dos hipótesis se ha contrastado de forma aislada en este trabajo (Elnaggar et al., 2021; Lin et al., 2023).

6.3 Sobre el Desempeño de los Clasificadores Lineales

La Regresión Logística, el modelo de menor complejidad entre los siete evaluados, obtuvo el mejor desempeño sobre los embeddings de ESM-2. Lejos de constituir un resultado anómalo, este comportamiento es consistente con la naturaleza de las representaciones empleadas: es plausible que los modelos de lenguaje proteico preentrenados generen espacios de representación en los que las clases resultan aproximadamente separables mediante fronteras lineales; en ese caso, la capacidad adicional de los clasificadores no lineales no aporta poder discriminante y sí incrementa el riesgo de sobreajuste. El resultado admite, por tanto, una lectura positiva: la calidad de la representación es tal que un clasificador simple resulta suficiente, lo que además favorece la interpretabilidad del modelo y reduce sus requisitos de despliegue. Conviene matizar, no obstante, que la ventaja de la Regresión Logística se observa sobre el conjunto de prueba y no en la validación cruzada, donde el SVM RBF obtuvo un valor ligeramente superior (0.8981 frente a 0.8923). Lo que el resultado indica, por tanto, es que el clasificador lineal generaliza mejor ante el cambio de distribución entre entrenamiento y prueba, más que una mayor capacidad discriminante sobre los datos de entrenamiento.

6.4 Sobre el Papel del Desbalance de Clases y las Técnicas de Remuestreo

El conjunto de prueba independiente de mACPpred 2.0 presenta una distribución marcadamente desbalanceada, con un 18.1 % de instancias positivas frente a un 81.9 % de negativas. Sin embargo, el conjunto de entrenamiento del mismo benchmark está balanceado por diseño, con 1.176 secuencias de cada clase. Esta asimetría entre las condiciones de entrenamiento y de evaluación tiene una consecuencia metodológica que conviene explicitar.

La incorporación de SMOTE dentro de cada pliegue de entrenamiento fue verificada empíricamente. Bajo el protocolo adoptado el algoritmo no genera ninguna muestra sintética, dado

que ambas clases se encuentran igualmente representadas en el conjunto de entrenamiento: devuelve la matriz de entrada sin modificar un solo valor. Los resultados reportados en las secciones 5.2 a 5.6 son, por tanto, idénticos a los que se obtendrían sin aplicar la técnica. Únicamente tendría efecto sobre una partición de entrenamiento desbalanceada, como la que generaba la división descartada en la sección 6.1, donde sí producía 1.719 muestras sintéticas.

Esta verificación confirma que el desbalance del benchmark constituye una condición exclusiva de la fase de evaluación y no del entrenamiento, por lo que no es abordable mediante remuestreo: sobremuestrear el conjunto de prueba invalidaría la evaluación. La respuesta metodológica adecuada consiste en la selección de métricas robustas frente al desbalance. En consecuencia, y tal como se argumenta en la sección 3.1.10, este trabajo adopta el AUC-ROC como métrica principal y reporta de forma complementaria el F1-score y el coeficiente de correlación de Matthews, evitando la exactitud, que bajo esta distribución otorgaría un 81.9 % a un clasificador trivial que asigna siempre la clase mayoritaria (Saito y Rehmsmeier, 2015).

6.5 Sobre la Integración Multimodal

La integración de descriptores fisicoquímicos y embeddings de ESM-2 8M no mejoró el desempeño respecto del mejor modelo unimodal: la concatenación directa alcanzó un AUC-ROC de 0.8676 y la fusión tardía un 0.8731, frente al 0.8745 obtenido por ESM-2 8M con Regresión Logística. Las diferencias son pequeñas y no resultan estadísticamente significativas (sección 6.6). Tampoco son uniformes entre clasificadores: con SVM RBF, ambas estrategias superan al mismo clasificador entrenado solo sobre ESM-2 8M (0.8676 y 0.8731 frente a 0.8654), mientras que con los otros seis clasificadores la concatenación queda por debajo de su equivalente unimodal. Lo que se mantiene es que ninguna configuración multimodal supera a la mejor configuración unimodal.

Los datos no indican que el bloque fisicoquímico carezca de información: por sí solo alcanza un AUC-ROC de 0.8416, y con SVM RBF su incorporación a los embeddings mejora ligeramente el resultado. Lo que no se observa es que la unión de ambos bloques aporte más de lo que el bloque de embeddings ya alcanza por sí solo en su mejor configuración.

La explicación más plausible —que este trabajo no contrasta de forma directa— es la redundancia entre ambas representaciones. Un modelo de lenguaje proteico se entrena sobre la secuencia de aminoácidos, de modo que la composición y las propiedades locales que los descriptores calculan de forma explícita están ya presentes, de forma implícita, en los embeddings. Incorporarlas por segunda vez amplía el espacio de entrada de 320 a 897 dimensiones —casi el triple— sin añadir señal nueva apreciable, y el coste de estimar esos parámetros adicionales sobre 2.352 ejemplos de entrenamiento supera al beneficio marginal que aportan.

A esta redundancia se suma una circunstancia propia del conjunto de evaluación. Como se documenta en la sección 5.1.1, los péptidos anticancerígenos del conjunto de prueba independiente tienen una longitud mediana de nueve residuos. Una parte sustancial de los 577 descriptores —en particular las 400 frecuencias dipeptídicas— pierde capacidad informativa en secuencias tan cortas: un péptido de nueve residuos contiene únicamente ocho dipéptidos, por lo que el vector correspondiente queda casi vacío. El bloque fisicoquímico es, por tanto, más débil precisamente sobre las instancias que determinan la evaluación, mientras conserva un peso agregado elevado dentro del modelo.

El contraste entre las dos estrategias aporta un matiz adicional. La fusión tardía, que mantiene los bloques en espacios separados hasta el momento de promediar las probabilidades, superó a la concatenación directa en AUC-ROC (0.8731 frente a 0.8676). Cuando las representaciones son redundantes y de dimensionalidad dispar, combinar únicamente la decisión

final puede resultar preferible a fundirlas en un vector único. La observación no es general —con Extra Trees el orden se invierte, 0.8584 frente a 0.8622— y, en cualquier caso, ninguna de las dos alternativas justifica aquí el coste adicional.

Desde el punto de vista de la implementación, la conclusión tiene una consecuencia práctica favorable. El cálculo de los 577 descriptores en tiempo de inferencia exige dependencias y tiempo de cómputo adicionales que el modelo final evita: al no aportar mejora, prescindir de ellos simplifica el paquete acp-classifier sin pérdida alguna de desempeño.

6.6 Sobre la Robustez de los Resultados

La diferencia de longitudes documentada en la sección 5.1.1 plantea una pregunta que conviene responder de forma explícita: si los modelos aprenden propiedades de la secuencia o se limitan a aprovechar que, en el conjunto de prueba, los péptidos anticancerígenos son más cortos que los no anticancerígenos. Para responderla se evaluó una línea base que no emplea ningún modelo y ordena los péptidos únicamente por su longitud, asignando mayor puntuación a los más cortos. Sobre el conjunto de prueba independiente esa línea base alcanza un AUC-ROC de 0.8493, superior al de los descriptores fisicoquímicos (0.8416) y cercano al de ProtBERT (0.8517); sobre el conjunto de entrenamiento, donde la diferencia de longitudes entre clases es menor, se queda en 0.6074.

La Tabla 16 descompone el desempeño por rangos de longitud. Dentro de cada rango la longitud deja de discriminar —su AUC-ROC cae a valores entre 0.51 y 0.66—, mientras que los tres modelos conservan valores entre 0.74 y 0.87. Los modelos, por tanto, no se limitan a reproducir la longitud: capturan información de la secuencia que distingue entre péptidos de longitud similar. La ordenación entre representaciones se mantiene además en el rango de 6 a 10 residuos, que concentra el 63 % de los péptidos anticancerígenos del conjunto de prueba, donde

ESM-2 8M (0.8667) supera a ProtBERT (0.8336) y a los descriptores (0.7396). El resultado es coherente con lo expuesto en la sección 5.1.1: la composición dipeptídica pierde información en los péptidos cortos, y es precisamente en ellos donde los descriptores se sitúan más lejos de las representaciones contextuales.

Tabla 16

AUC-ROC por rango de longitud sobre el conjunto de prueba independiente.

Modelo	Global	6–10 residuos	11–20 residuos	21–50 residuos
Solo longitud (línea base)	0.8493	0.5853	0.6559	0.5124
Descriptores + SVM RBF	0.8416	0.7396	0.7515	0.8580
ESM-2 8M + Logistic Regression	0.8745	0.8667	0.7793	0.8683
ProtBERT + SVM RBF	0.8517	0.8336	0.7618	0.8722

Nota. Número de péptidos por rango: 6–10 residuos, 386 ACPs y 196 no-ACPs; 11–20, 176 y 1.390; 21–50, 48 y 1.174. La línea base asigna a cada péptido una puntuación igual a su longitud con signo negativo. Los tres modelos se reentrenaron con los hiperparámetros de la Tabla B2 y reproducen exactamente el AUC-ROC reportado en las Tablas 5, 6 y 7.

La incertidumbre de las cifras se estimó mediante remuestreo bootstrap del conjunto de prueba, con 2.000 réplicas. El intervalo de confianza del 95 % del AUC-ROC del modelo final es [0.859, 0.887]. Las diferencias frente a la fusión tardía (0.0014) y frente a la concatenación (0.0069), calculadas sobre las mismas réplicas, tienen intervalos que incluyen el cero —[−0.006, 0.008] y [−0.002, 0.015]—, de modo que no son estadísticamente significativas: el resultado de la integración multimodal debe leerse como ausencia de mejora y no como un empeoramiento. La diferencia frente a la línea base de longitud, en cambio, es positiva en todo su intervalo, [0.003, 0.047]. En cuanto a las herramientas de referencia, el valor publicado de mACPpred 2.0 (0.887) coincide con el límite superior del intervalo, mientras que el de MLACP 2.0 (0.893) queda fuera de él.

Por último, dado que bajo desbalance la curva de precisión-exhaustividad resulta más informativa que la curva ROC (Saito y Rehmsmeier, 2015), se calculó el área bajo esa curva para el modelo final: 0.506, frente al 0.181 que obtendría un clasificador aleatorio, igual a la proporción de péptidos anticancerígenos en el conjunto de prueba. El valor es coherente con la precisión de 0.534 señalada en el capítulo de Limitaciones.

7. Limitaciones

El presente trabajo está sujeto a las siguientes limitaciones, cuya explicitación resulta necesaria para delimitar el alcance de las conclusiones:

Desplazamiento de distribución entre entrenamiento y evaluación. Como se documenta en la sección 5.1.1, los péptidos anticancerígenos del conjunto de prueba presentan una longitud mediana sustancialmente menor que los del conjunto de entrenamiento. Los modelos se entrenan, en consecuencia, sobre un régimen de longitud distinto del empleado en su evaluación, condición que limita la extrapolación de los resultados a poblaciones de péptidos con otras distribuciones de longitud.

Validación exclusivamente computacional. Los resultados se limitan a la validación computacional sobre datos de referencia públicos, con el alcance y el nivel de madurez tecnológica descritos en la sección 4.6; su confirmación experimental queda como línea de trabajo futuro.

Evaluación sobre un único conjunto de referencia. Los modelos fueron evaluados exclusivamente sobre el benchmark de mACPpred 2.0. No se incorporaron conjuntos alternativos como los de AntiCP 2.0, ACP135 o ACP99, cuya utilización no es inmediata. El conjunto de mACPpred 2.0 se construyó reuniendo los conjuntos de once herramientas previas, entre ellas

AntiCP 2.0, de modo que parte de esas secuencias ya forma parte del entrenamiento empleado: según los identificadores de los archivos, 297 secuencias de Training.fasta proceden de los conjuntos de AntiCP 2.0. Evaluar sobre ellos mediría en parte la memoria del modelo y no su capacidad de generalización, y exigiría antes depurar por similitud las secuencias compartidas. ACP135 y ACP99, por su parte, contienen 135 y 99 péptidos, un tamaño con el que el intervalo de confianza del AUC-ROC resultaría considerablemente más amplio que el obtenido sobre las 3.370 secuencias del conjunto independiente. En consecuencia, la estabilidad del desempeño ante otras construcciones del problema no está verificada.

Alcance de las estrategias de fusión evaluadas. La integración multimodal se abordó mediante concatenación de representaciones y fusión tardía de probabilidades. No se implementaron mecanismos de fusión basados en atención cruzada como los propuestos por Lv et al. (2025), cuya evaluación queda propuesta como trabajo futuro.

Alcance de la interpretabilidad. Aunque los descriptores fisicoquímicos son interpretables por construcción, las representaciones generadas por modelos de lenguaje proteico no admiten una lectura biológica directa, lo que limita la capacidad del modelo final para orientar el diseño racional de nuevos péptidos.

Dependencia de la precisión respecto de la prevalencia. Las métricas reportadas corresponden a un conjunto de prueba con un 18.1 % de instancias positivas. Con el umbral de decisión en 0.5, la precisión del modelo final sobre ese conjunto es de 0.534, de modo que aproximadamente la mitad de los péptidos señalados como ACP son falsos positivos. En un escenario de cribado real, donde la proporción de ACPs es sensiblemente menor, esa precisión disminuye. Se trata de una propiedad de la tarea y no de una deficiencia del modelo, pero condiciona su uso práctico y debe ser conocida por quien emplee la herramienta.

Selección del modelo final sobre el conjunto de prueba. El modelo final se eligió por su AUC-ROC sobre el conjunto de prueba independiente, dado que la validación cruzada no distinguía entre las configuraciones candidatas (sección 5.7). Esa elección introduce un sesgo optimista en la cifra de 0.8745, que debe leerse junto con su intervalo de confianza (sección 6.6). La misma consideración se aplica a la estrategia de pooling, adoptada tras observar su efecto sobre el conjunto de prueba.

Similitud y origen de las secuencias. Los autores del conjunto de referencia redujeron la redundancia aplicando CD-HIT con un umbral de identidad del 85 % (Sangaraju et al., 2024; Li y Godzik, 2006), y en este trabajo se verificó que no existen secuencias idénticas entre los archivos de entrenamiento y de prueba; no se realizó, sin embargo, un análisis propio de similitud parcial. Además, 602 de los 610 péptidos anticancerígenos del conjunto de prueba proceden de una misma colección de origen, la de MLACP 2.0, según consta en los identificadores de los archivos del conjunto de referencia, lo que limita la diversidad de la clase positiva sobre la que se evalúa el modelo.

8. Conclusiones

El presente trabajo abordó el desarrollo comparativo de modelos de clasificación de péptidos anticancerígenos basados en descriptores fisicoquímicos, en representaciones contextuales generadas por modelos de lenguaje proteico y en la integración de ambas aproximaciones. De los resultados obtenidos se derivan las siguientes conclusiones:

Se construyeron y evaluaron modelos predictivos para las tres representaciones planteadas en los objetivos específicos, empleando un protocolo de validación cruzada anidada de 5×5

pliegues y evaluando sobre el conjunto de prueba independiente de mACPpred 2.0, lo que permite la comparación directa con el estado del arte.

La combinación de embeddings de ESM-2 8M con un clasificador de Regresión Logística alcanzó un AUC-ROC de 0.8745 sobre el conjunto de prueba independiente de mACPpred 2.0, por delante de nueve de las once herramientas con AUC-ROC publicado sobre ese conjunto y por debajo de MLACP 2.0 (0.893) y mACPpred 2.0 (0.887). El resultado se obtiene con un único modelo de ocho millones de parámetros y un clasificador lineal, sin ensamblados ni hardware especializado en inferencia, lo que sitúa la contribución del trabajo en la relación entre desempeño y coste computacional antes que en la mejora del máximo absoluto.

La capacidad del modelo de lenguaje proteico no determina el desempeño en la tarea. ESM-2 8M, el modelo de menor tamaño evaluado, superó tanto a ESM-2 150M como a ProtBERT, resultado consistente con la longitud reducida de los péptidos del conjunto de evaluación y con el tamaño limitado del conjunto de entrenamiento, aunque la comparación realizada no aísla el efecto del tamaño del modelo.

La estrategia de agregación de los estados ocultos constituye una decisión de diseño de impacto comparable al de la elección del modelo. La exclusión del token CLS en el cálculo del mean pooling produjo una mejora de 0.0191 puntos de AUC-ROC sobre la misma representación —0.0150 con la exclusión exacta de CLS y EOS—, superior a la diferencia observada entre modelos de distinta capacidad.

El protocolo de evaluación condiciona de forma determinante las métricas reportadas. Una división estratificada construida sobre la unión de los archivos del benchmark, ensayada y descartada durante el desarrollo, produce estimaciones optimistas que no admiten comparación

con las herramientas de referencia, lo que subraya la necesidad de respetar las particiones oficiales de los conjuntos de referencia.

El modelo de mejor desempeño resultó ser también el de menores requisitos computacionales, lo que hace viable su distribución como paquete instalable ejecutable en equipos sin hardware especializado.

La integración multimodal de descriptores fisicoquímicos y embeddings no mejoró el desempeño de la mejor representación individual. Ni la concatenación directa, con un AUC-ROC de 0.8676, ni la fusión tardía, con 0.8731, superaron el 0.8745 obtenido por ESM-2 8M en solitario, aun cuando el bloque fisicoquímico contiene por sí solo información discriminante (AUC-ROC de 0.8416). La explicación más plausible es que la información que los descriptores aportan de forma explícita se encuentra ya representada, de manera implícita, en las representaciones del modelo de lenguaje, de modo que su incorporación amplía la dimensionalidad del problema sin añadir señal discriminante. El resultado, de signo negativo, simplifica el modelo final y reduce las dependencias del paquete distribuido.

En respuesta a la pregunta de investigación, los modelos multimodales alcanzaron un desempeño cercano al del mejor modelo individual, pero no superior, y las diferencias entre ambos no son estadísticamente significativas. La hipótesis de que la integración de descriptores fisicoquímicos y embeddings aporta mejoras frente a los modelos basados en una sola representación no se confirma. En consecuencia, el objetivo general se cumplió en su dimensión metodológica —se desarrollaron y evaluaron los modelos individuales y su combinación mediante dos estrategias—, y el paquete distribuido incorpora el modelo individual, que resultó el de mejor desempeño y el de menores requisitos computacionales.

9. Trabajo Futuro

A partir de las limitaciones identificadas y de los resultados obtenidos, se proponen las siguientes líneas de continuación:

Evaluar mecanismos de fusión multimodal basados en atención cruzada, que permitan una integración de las representaciones más expresiva que la concatenación directa y la fusión tardía evaluadas en este trabajo.

Extender la evaluación a conjuntos de referencia adicionales, con el fin de verificar la estabilidad del desempeño ante distintas construcciones del problema y distintas distribuciones de longitud de secuencia.

Definir un protocolo de validación previa para el empleo del paquete sobre colecciones distintas a la utilizada en este trabajo. La herramienta acepta cualquier archivo FASTA, pero su desempeño depende del perfil de longitudes y del criterio con que se hayan definido los ejemplos negativos en cada colección, y la salida no advierte de esa circunstancia. Antes de emplearla sobre un conjunto nuevo conviene validarla sobre una muestra etiquetada de ese mismo conjunto y, en su caso, reajustar el umbral de decisión a la prevalencia esperada.

Explorar estrategias de ajuste fino (fine-tuning) de los modelos de lenguaje proteico sobre el dominio específico de los péptidos anticancerígenos, en lugar de emplear las representaciones preentrenadas de forma directa.

Incorporar técnicas de interpretabilidad que permitan relacionar las dimensiones de los embeddings con propiedades fisicoquímicas conocidas, con el objetivo de orientar el diseño racional de nuevos candidatos.

Validar experimentalmente *in vitro* una selección de los péptidos identificados por el modelo, como paso siguiente al alcance definido en la sección 4.6.

Referencias Bibliográficas

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 265–283. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Agrawal, P., Bhagat, D., Mahalwal, M., Sharma, N., & Raghava, G. P. S. (2021). AntiCP 2.0: An updated model for predicting anticancer peptides. *Briefings in Bioinformatics, 22*(3), bbaa153. https://doi.org/10.1093/bib/bbaa153

Arif, M., Musleh, S., Fida, H., & Alam, T. (2024). PLMACPred prediction of anticancer peptides based on protein language model and wavelet denoising transformation. *Scientific Reports, 14*(1), 16992. https://doi.org/10.1038/s41598-024-67433-8.

Bauso, L. V., La Fauci, V., Munaò, S., Bonfiglio, D., Armeli, A., Maimone, N., Longo, C., & Calabrese, G. (2024). Biological activity of natural and synthetic peptides as anticancer agents. *International Journal of Molecular Sciences, 25*(13), 7264. https://doi.org/10.3390/ijms25137264

Bray, F., Laversanne, M., Sung, H., Ferlay, J., Siegel, R. L., Soerjomataram, I., & Jemal, A. (2024). Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians, 74*(3), 229–263. https://doi.org/10.3322/caac.21834

Cao, D. S., Xu, Q. S., & Liang, Y. Z. (2013). Propy: A tool to generate various modes of Chou's PseAAC. *Bioinformatics, 29*(7), 960–962. https://doi.org/10.1093/bioinformatics/btt072

Chen, J., Cheong, H. H., & Siu, S. W. I. (2021). xDeep-AcPEP: Deep learning method for anticancer peptide activity prediction based on convolutional neural network and multitask learning. *Journal of Chemical Information and Modeling, 61*(8), 3789–3803. https://doi.org/10.1021/acs.jcim.1c00181

Cheong, H.-H., Zuo, W., Chen, J., Un, C.-W., Si, Y.-W., Wong, K. H., Kwok, H. F., & Siu, S. W. I. (2024). Identification of anticancer peptides from the genome of *Candida albicans*: In silico screening, in vitro and in vivo validations. Journal of Chemical Information and Modeling, 64(15), 6174–6189. https://doi.org/10.1021/acs.jcim.4c00501

Chinnadurai, R. K., Khan, N., Meghwanshi, G. K., Ponne, S., Althobiti, M., & Kumar, R. (2023). Current research status of anti-cancer peptides: Mechanism of action, production, and clinical applications. *Biomedicine & Pharmacotherapy, 164*, 114996. https://doi.org/10.1016/j.biopha.2023.114996

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & de Hoon, M. J. L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics, 25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2021). ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(10), 7112–7127. https://doi.org/10.1109/TPAMI.2021.3095381

Instituto Nacional de Cancerología. (2023, octubre). *Cáncer en cifras*. https://www.cancer.gov.co/recursos_user/imagenes/Infografias/CANCER_EN_CIFRAS_2022_APROBADA.pdf

Jhong, J.-H., Yao, L., Pang, Y., Li, Z., Chung, C.-R., Wang, R., Li, S., Li, W., Luo, M., Ma, R., Huang, Y., Zhu, X., Zhang, J., Feng, H., Cheng, Q., Wang, C., Xi, K., Wu, L.-C., Chang, T.-H., & Lee, T.-Y. (2022). dbAMP 2.0: Updated resource for antimicrobial peptides with an enhanced scanning method for genomic and proteomic data. *Nucleic Acids Research, 50*(D1), D460–D470. https://doi.org/10.1093/nar/gkab1080

Kang, X., Dong, F., Shi, C., Liu, S., Sun, J., Chen, J., Li, H., Xu, H., Lao, X., & Zheng, H. (2019). DRAMP 2.0, an updated data repository of antimicrobial peptides. *Scientific Data, 6*(1), 148. https://doi.org/10.1038/s41597-019-0154-y

Kilimci, Z. H., & Yalcin, M. (2024). ACP-ESM: A novel framework for classification of anticancer peptides using protein-oriented transformer approach. *Artificial Intelligence in Medicine, 156*, 102951. https://doi.org/10.1016/j.artmed.2024.102951

Li, W., & Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics, 22*(13), 1658–1659. https://doi.org/10.1093/bioinformatics/btl158

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science, 379*(6637), 1123–1130. https://doi.org/10.1126/science.ade2574

Lv, J., Li, K., Wang, Y., Xu, J., Meng, Y., Cui, F., Wei, L., Zhang, Q., & Zhang, Z. (2025). ACP-EPC: An interpretable deep learning framework for anticancer peptide prediction utilizing

pre-trained protein language model and multi-view feature extracting strategy. *Molecular Diversity*. https://doi.org/10.1007/s11030-025-11352-x

Mankins, J. C. (1995). *Technology readiness levels: A white paper*. NASA Office of Space Access and Technology. https://aiaa.kavi.com/apps/group_public/download.php/2212/TRL_Mankins1995.pdf.

Osorio, D., Rondón-Villarreal, P., & Torres, R. (2015). Peptides: A package for data mining of antimicrobial peptides. *The R Journal, 7*(1), 4–14. https://doi.org/10.32614/RJ-2015-001

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*, 8024–8035.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Phan, L. T., Park, H. W., Pitti, T., Madhavan, T., Jeon, Y.-J., & Manavalan, B. (2022). MLACP 2.0: An updated machine learning tool for anticancer peptide prediction. *Computational and Structural Biotechnology Journal, 20*, 4473–4480. https://doi.org/10.1016/j.csbj.2022.07.043

Pirtskhalava, M., Amstrong, A. A., Grigolava, M., Chubinidze, M., Alimbarashvili, E., Vishnepolsky, B., Gabrielian, A., Rosenthal, A., Hurt, D. E., & Tartakovsky, M. (2021). DBAASP v3: Database of antimicrobial/cytotoxic activity and structure of peptides as a

resource for development of new therapeutics. *Nucleic Acids Research, 49*(D1), D288–D297. https://doi.org/10.1093/nar/gkaa991

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies, 2*(1), 37–63.

Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., & Rives, A. (2021). MSA Transformer. *Proceedings of the 38th International Conference on Machine Learning, PMLR 139*, 8844–8856. https://proceedings.mlr.press/v139/rao21a.html

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE, 10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Sangaraju, V. K., Pham, N. T., Wei, L., Yu, X., & Manavalan, B. (2024). mACPpred 2.0: Stacked deep learning for anticancer peptide prediction with integrated spatial and probabilistic feature representations. *Journal of Molecular Biology, 436*(17), 168687. https://doi.org/10.1016/j.jmb.2024.168687

Tran, C., Khadkikar, S., & Porollo, A. (2023). Survey of protein sequence embedding models. *International Journal of Molecular Sciences, 24*(4), 3775. https://doi.org/10.3390/ijms24043775

Tyagi, A., Tuknait, A., Anand, P., Gupta, S., Sharma, M., Mathur, D., Joshi, A., Singh, S., Gautam, A., & Raghava, G. P. S. (2015). CancerPPD: A database of anticancer peptides and proteins. *Nucleic Acids Research, 43*(D1), D837–D843. https://doi.org/10.1093/nar/gku892

The UniProt Consortium. (2023). UniProt: The Universal Protein Knowledgebase in 2023. *Nucleic Acids Research, 51*(D1), D523–D531. https://doi.org/10.1093/nar/gkac1052

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics, 7*(1), 91. https://doi.org/10.1186/1471-2105-7-91

Waghu, F. H., Barai, R. S., Gurung, P., & Idicula-Thomas, S. (2016). CAMPR3: A database on sequences, structures and signatures of antimicrobial peptides. *Nucleic Acids Research, 44*(D1), D1094–D1097. https://doi.org/10.1093/nar/gkv1051

Wang, G., Li, X., & Wang, Z. (2016). APD3: The antimicrobial peptide database as a tool for research and education. *Nucleic Acids Research, 44*(D1), D1087–D1093. https://doi.org/10.1093/nar/gkv1278

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

World Medical Association. (2013). World Medical Association Declaration of Helsinki: Ethical principles for medical research involving human subjects. *JAMA, 310*(20), 2191–2194. https://doi.org/10.1001/jama.2013.281053

Yan, J., Bhadra, P., Li, A., Sethiya, P., Qin, L., Tai, H. K., Wong, K. H., & Siu, S. W. I. (2020). Deep-AmPEP30: Improve short antimicrobial peptides prediction with deep learning. *Molecular Therapy - Nucleic Acids, 20*, 882–894. https://doi.org/10.1016/j.omtn.2020.05.006

Apéndices

Apéndice A. Manual del Paquete acp-classifier

Este anexo documenta la instalación y el uso del paquete desarrollado como resultado del quinto objetivo específico. El paquete implementa el modelo seleccionado —representaciones de ESM-2 8M con un clasificador de Regresión Logística— y permite clasificar secuencias peptídicas desde la línea de comandos o desde código Python.

A.1 Requisitos e Instalación

El paquete requiere Python 3.9 o superior y no necesita unidad de procesamiento gráfico. La primera ejecución descarga los pesos del modelo ESM-2 8M (aproximadamente 31 MB) desde el repositorio de HuggingFace y los conserva en caché local para los usos siguientes.

```
pip install acp-classifier
```

Alternativamente, desde el código fuente o mediante Conda:

```
git clone https://github.com/ACP-Classifier-test/acp-classifier.git
cd acp-classifier
pip install -e .


conda env create -f environment.yml
conda activate acp-classifier
```

A.2 Uso Desde la Línea de Comandos

```
acp-classifier -i peptidos.fasta                      # resultados por pantalla

acp-classifier -i peptidos.fasta -o resultados.csv    # resultados a un archivo

acp-classifier -i peptidos.fasta -o salida.csv -u 0.8 # umbral más exigente
```

Tabla A1

Opciones de la interfaz de línea de comandos del paquete acp-classifier.

Opción	Descripción	Valor por defecto
-i, --input	Archivo FASTA de entrada (obligatorio)	—
-o, --output	Archivo CSV de salida	salida por pantalla
-u, --umbral	Probabilidad mínima para clasificar como ACP	0.5
-b, --batch-size	Secuencias procesadas por lote	32
-m, --modelo	Artefacto .joblib alternativo	el incluido
-q, --quiet	Suprime los mensajes de progreso	—
--version	Muestra la versión instalada	—

A.3 Formatos de Entrada y de Salida

La entrada es un archivo FASTA estándar con una o varias secuencias. Se admiten registros repartidos en varias líneas. Los residuos no estándar (X, B, Z, U y O) se descartan, igual que durante el entrenamiento del modelo: si el preprocesamiento no coincidiera entre entrenamiento e inferencia, el clasificador recibiría datos distintos de aquellos con los que aprendió.

```
>peptido_1
GLFSVVTGVLKAVGKNVAKNVGGSLLEQLKCKKISGGC
>peptido_2
NNEQIRRFSQVLLDAGIVTTIRKTRGDDIDA
```

La salida es un archivo CSV con cinco columnas: identificador, secuencia, longitud, probabilidad estimada de actividad anticancerígena y clasificación resultante.

```
identificador,secuencia,longitud,probabilidad_acp,prediccion
peptido_1,GLFSVVTGVLKAVGKNVAKNVGGSLLEQLKCKKISGGC,38,0.7350,ACP
peptido_2,NNEQIRRFSQVLLDAGIVTTIRKTRGDDIDA,31,0.0643,no-ACP
```

A.4 Uso Desde Código Python

```
from acp_classifier import AcpClassifier

clf = AcpClassifier()

for p in clf.predecir_secuencias(["GIGKFLHSAKKFGKAFVGEIMNS"]):
    print(p.identificador, p.probabilidad_acp, p.prediccion)

predicciones = clf.predecir_fasta("peptidos.fasta", umbral=0.7)
probas = clf.probabilidades(["GIGKFLHSAKKFGKAFVGEIMNS", "AAAAAAAAAA"])
```

A.5 Arquitectura Interna

El procesamiento sigue cuatro etapas, descritas en la Figura 4:

- Lectura y limpieza de las secuencias, conservando únicamente los veinte aminoácidos estándar.
- Generación de la representación mediante el modelo esm2_t6_8M_UR50D, promediando los estados ocultos de la última capa con exclusión del token CLS y de la última posición de residuo —la variante con la que se entrenó el modelo, descrita en la sección 4.2.1 y registrada en los metadatos del artefacto para que entrenamiento e inferencia coincidan— para obtener un vector de 320 dimensiones por péptido.
- Normalización de cada dimensión con la media y la desviación típica calculadas sobre el conjunto de entrenamiento, almacenadas junto con el modelo.

- Clasificación mediante Regresión Logística (C = 0,01, solver saga) y escritura del archivo de salida.

Los parámetros del modelo —coeficientes, intercepto, media y desviación típica— se distribuyen como arreglos numéricos y no como objetos serializados de la biblioteca de aprendizaje automático, de modo que la predicción no depende de la versión de esa biblioteca instalada en el equipo del usuario.

A.6 Rendimiento

Medido sobre un procesador AMD Ryzen 7 de ocho núcleos, sin unidad de procesamiento gráfico:

Tabla A2

Rendimiento del paquete acp-classifier sobre un lote de 100 péptidos.

Operación	Tiempo
Carga del modelo	1,22 s
Clasificación de 100 péptidos, con el modelo ya cargado	0,52 s
Proceso completo desde consola, incluido el arranque	entre 9,7 y 10,1 s
Rendimiento sostenido	aproximadamente 193 péptidos por segundo

La diferencia entre los 0,52 segundos de clasificación y los cerca de diez del proceso completo corresponde al tiempo de importación de las bibliotecas de aprendizaje profundo, no al cálculo. En procesamiento por lotes grandes, ese coste se paga una sola vez.

A.7 Verificación

El paquete incluye quince pruebas automatizadas que cubren la lectura de archivos FASTA, el manejo de errores, los metadatos del modelo, el efecto del umbral de decisión y el funcionamiento de la interfaz de línea de comandos. Se ejecutan con:

```
pip install -e ".[dev]"
pytest -q
```

A.8 Limitaciones de Uso

- La validación es exclusivamente computacional (sección 4.6). Una predicción positiva constituye una hipótesis a verificar experimentalmente, no una afirmación de actividad biológica.
- El modelo fue entrenado sobre péptidos de entre 6 y 50 residuos; su comportamiento fuera de ese rango no ha sido caracterizado.
- El desempeño está caracterizado sobre un único conjunto de referencia.
- Existe un desplazamiento de distribución conocido entre los conjuntos de entrenamiento y de prueba del benchmark, descrito en la sección 5.1.

Apéndice B. Tablas Extendidas de Resultados

Las tablas del capítulo de Resultados recogen las métricas principales de cada configuración. Este anexo reúne los resultados completos de cada clasificador, incluidas las métricas de validación cruzada y los hiperparámetros seleccionados, para permitir la verificación y la reproducción de los experimentos.

B.1 Protocolo Adoptado — Descriptores Fisicoquímicos Sobre el Conjunto de Prueba Independiente

Siete clasificadores optimizados mediante validación cruzada anidada de cinco pliegues externos y cinco internos, evaluados sobre las 3.370 secuencias del conjunto de prueba independiente. Estos son los resultados comparables con el estado del arte.

Tabla B1

Descriptores fisicoquímicos sobre el conjunto de prueba independiente: resultados completos.

Clasificador	AUC-ROC	F1	MCC	Exactitud	AUC en CV
SVM RBF	**0.8416**	**0.5669**	**0.4614**	**0.8136**	**0.8834 ± 0.0138**
Gradient Boosting	0.8228	0.5112	0.3889	0.7861	0.8900 ± 0.0138
Extra Trees	0.8207	0.4915	0.3655	0.7881	0.8918 ± 0.0131
Random Forest	0.8151	0.5068	0.3831	0.7834	0.8895 ± 0.0132
Logistic Regression	0.7968	0.5098	0.3870	0.7843	0.8680 ± 0.0146
MLP	0.7575	0.4780	0.3451	0.7680	0.8569 ± 0.0105
KNN	0.7163	0.4100	0.2477	0.6849	0.8640 ± 0.0217

B.2 Hiperparámetros Seleccionados

Configuración elegida por la búsqueda en rejilla, repetida sobre la totalidad del conjunto de entrenamiento, para el mejor clasificador de cada representación; con ella se entrenó el modelo que se evaluó sobre el conjunto de prueba independiente.

Tabla B2

Hiperparámetros seleccionados para los modelos finales de cada representación.

Representación	Clasificador	Hiperparámetros
Descriptores fisicoquímicos (577 dim)	SVM RBF	{'C': 1, 'gamma': 'scale'}
ESM-2 8M (320 dim)	Regresión Logística	{'C': 0.01, 'solver': 'saga'}
ProtBERT (1.024 dim)	SVM RBF	{'C': 1, 'gamma': 'auto'}
Multimodal, concatenación (897 dim)	SVM RBF	{'C': 1, 'gamma': 'scale'}
Multimodal, fusión tardía (w = 0.5)	SVM RBF	desc={'C': 1, 'gamma': 'auto'} | emb={'C': 10, 'gamma': 0.001}

La fila de fusión tardía recoge dos configuraciones porque la estrategia entrena dos modelos independientes: desc= corresponde al clasificador del bloque de descriptores y emb= al del bloque de embeddings.

B.3 Rejillas de Búsqueda Empleadas

Las mismas para todas las representaciones, de modo que las comparaciones entre filas sean atribuibles a la representación y no al espacio de búsqueda.

Tabla B3

Rejillas de hiperparámetros exploradas por cada clasificador.

Clasificador	Rejilla
SVM RBF	C: 0.1, 1, 10, 100 · gamma: scale, auto, 0.001, 0.01
Random Forest	n_estimators: 100, 200, 300 · max_depth: None, 10, 20
Extra Trees	n_estimators: 100, 200, 300 · max_depth: None, 10, 20
Gradient Boosting	n_estimators: 100, 200 · learning_rate: 0.05, 0.1, 0.2 · max_depth: 3, 5
Regresión Logística	C: 0.001, 0.01, 0.1, 1, 10, 100 · solver: lbfgs, saga
KNN	n_neighbors: 3, 5, 7, 11 · weights: uniform, distance · metric: euclidean, cosine
MLP	hidden_layer_sizes: (256,), (512,), (512,256), (256,128) · alpha: 0.0001, 0.001, 0.01 · learning_rate_init: 0.001, 0.0001

Apéndice C. Reproducibilidad y Acceso al Código

C.1 Repositorio

El código del paquete, el modelo serializado, las pruebas automatizadas y los ejemplos están disponibles públicamente bajo licencia MIT en:

```
https://github.com/ACP-Classifier-test/acp-classifier
```

El repositorio incluye un archivo CITATION.cff con los metadatos de citación, un registro de cambios y un flujo de integración continua que ejecuta las pruebas sobre Python 3.9, 3.10, 3.11 y 3.12.

C.2 Datos

Se empleó el conjunto de referencia de mACPpred 2.0 (Sangaraju et al., 2024), distribuido por sus autores junto con el programa de uso local. Está compuesto por dos archivos en formato FASTA cuyos contenidos se verificaron antes de su uso:

Tabla C1

Composición del conjunto de referencia empleado, verificada sobre los archivos.

Archivo	ACPs	no-ACPs	Total
Training.fasta	1.176	1.176	2.352
Independent.fasta	610	2.760	3.370

C.3 Entorno de Ejecución

Los experimentos de la Fase 1, basados en descriptores fisicoquímicos, se ejecutaron en un equipo local con procesador AMD Ryzen 7 de ocho núcleos, sin unidad de procesamiento gráfico. La generación de las representaciones de los modelos de lenguaje proteico y los experimentos de las Fases 2 a 4 se realizaron en Google Colab Pro; la unidad de procesamiento gráfico NVIDIA A100 se empleó únicamente para generar las representaciones, ya que los clasificadores de scikit-learn se ejecutan en procesador en ambos entornos.

C.4 Reproducción de los Experimentos

- Fase 1 sobre el conjunto de prueba independiente: main_descriptores_test_independiente.py. El cálculo de los 577 descriptores para las

5.722 secuencias toma unos seis segundos; la validación cruzada anidada, entre veinticinco y cuarenta minutos aprovechando los ocho núcleos.

- Fases 2 a 4, representaciones contextuales y modelo multimodal: notebook de Colab, que requiere unidad de procesamiento gráfico únicamente para generar las representaciones.
- Reentrenamiento del modelo del paquete: entrenar_modelo.py, que regenera el artefacto desde los archivos FASTA y reimprime las métricas sobre el conjunto de prueba independiente. Toma alrededor de un minuto en procesador.

Todas las particiones y búsquedas emplean una semilla fija (42), de modo que las ejecuciones son reproducibles.

La reproducibilidad del resultado principal se verificó de forma directa: partiendo de las representaciones de ESM-2 8M almacenadas y entrenando una Regresión Logística con los hiperparámetros de la Tabla B2 sobre la partición de entrenamiento, la evaluación sobre el conjunto de prueba independiente devuelve un AUC-ROC de 0.8745, un F1-score de 0.6230, una exactitud de 0.8362 y un coeficiente de correlación de Matthews de 0.5341, coincidentes en las cuatro cifras con los valores reportados en la sección 5.3. Las quince pruebas automatizadas del paquete se ejecutan sin fallos sobre el entorno descrito.

C.5 Dependencias Principales

Tabla C2

Bibliotecas empleadas y función de cada una.

Biblioteca	Función
PyTorch y Transformers	Carga y ejecución de los modelos de lenguaje proteico
scikit-learn	Clasificadores, validación cruzada anidada y métricas
imbalanced-learn	Implementación de SMOTE

Biblioteca	**Función**
peptides.py y ProPy3	Cálculo de los descriptores fisicoquímicos
NumPy	Manipulación de las representaciones numéricas
joblib	Serialización del modelo y del normalizador