

**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
ESCUELA DE POSGRADO
UNIDAD DE POSGRADO DE LA FACULTAD DE
INGENIERIA DE PRODUCCIÓN Y SERVICIOS**



**Detección *in Silico* de Neoantígenos Utilizando
Transformers y Transfer Learning en el Marco de
Desarrollo de Vacunas Personalizadas para Tratar el
Cáncer**

Tesis presentada por el Magister:
Vicente Enrique Machaca Arceda

Para optar el Grado de:
Doctor en Ciencia de la Computación

Asesor:
Prof. Dr. Cristian Lopez Del Alamo

Arequipa - Perú
2024

Declaración de autenticidad

Yo, Vicente Enrique Machaca Arceda, declaro que la tesis titulada, ‘Detección *in Silico* de Neoantígenos Utilizando Transformers y Transfer Learning en el Marco de Desarrollo de Vacunas Personalizadas para Tratar el Cáncer’ y el trabajo presentado en este documento, son de mi propiedad intelectual y confirmo que:

- Este trabajo fue desarrollado durante mi candidatura a grado de doctor de esta universidad.
- Ninguna parte de esta tesis ha sido presentado para otro grado de esta universidad o cualquier otra institución.
- Cuando cito a otros autores, las fuentes han sido brindadas, con excepción de estas citas, el trabajo es de mi autoría.
- He agradecido las principales fuentes de ayuda.
- En caso de que mi tesis haya sido desarrollada con un equipo de trabajo, he sido claro y he detallado la parte exacta de mi autoría.

Firma:

Fecha:

“Con fe, disciplina y desinteresada devoción al deber, no hay nada que merezca la pena que no puedas lograr.”

Muhammad Ali Jinnah

Dedico este trabajo a mi esposa Pamela Laguna Laura, quien me ha acompañado durante todo este proceso, me ha motivado y sobre todo me ha dado su amor, me ha ayudado a prevalecer y siempre seguir adelante. De igual forma, a mis padres Vicente Machaca Chino y Victoria Arceda Arenas, de ellos he aprendido el valor de la disciplina, la fuerza por emprender y la importancia de los valores sin importar las circunstancias; gracias a ellos he logrado cumplir mis objetivos.

Resumen

El cáncer es el mayor problema de salud mundial en la actualidad, frente a esto han surgido nuevos tratamientos basados en inmunoterapia como el desarrollo de vacunas personalizadas basadas en neoantígenos. Sin embargo, el proceso para identificar neoantígenos, es complejo y existen varias etapas para lograrlo, desde el secuenciamiento de muestras tumorales, alineamiento con muestras de tejido saludable, identificación y anotación de mutaciones, para luego proseguir con la predicción de la unión de péptidos con el MHC y posteriormente la unión del pMHC con el TCR. Si esta unión procede, el péptido en cuestión es un fuerte candidato a neoantígeno. En este proceso, una de las fases más críticas es la predicción de la unión pMHC, lo cual ha motivado el desarrollo de esta tesis.

Además, las redes neuronales Transformers han revolucionado el campo del procesamiento natural del lenguaje y se han aplicado en muchas otras áreas como en Proteómica. Esto porque las proteínas al ser representadas como secuencias de aminoácidos, son muy similares a las secuencias de palabras en una oración. Es así, que otras investigaciones han aplicado el uso de Transformers y redes neuronales con mecanismos de atención para la predicción de la unión pMHC. Adicionalmente, existen modelos pre-entrenados como TAPE, ProtBert y ESM2, estos han sido entrenados con grandes volúmenes de datos para varias tareas de Proteómica. Basados en lo anterior, en esta tesis se propone el uso de aplicar *fine-tuning* a TAPE, ProtBert, ESM2(t6), ESM2(t12), ESM2(t30) y ESM2(t33) para la tarea de predicción de la unión pMHC, el *fine-tunning* consistió en agregar un bloque BiLSTM al final del modelo. Además, se ha evaluado el uso de *Gradient Accumulation Steps* (GAS) y una metodología de congelamiento de capas.

Luego de los experimentos, los modelos con mejores resultados fueron TAPE-GAS, que resultó de aplicar GAS a TAPE y ESM2(t6)-Freeze, que resultó de aplicar la metodología de congelamiento a ESM2. Finalmente, se compararon estos modelos con los métodos de mejor resultado en el estado del arte, tales como: NetMHCpan4.1, MHCflurry2.0, ACME, Anthem y MixMHCpred2.2. Al finalizar los experimentos, TAPE-GAS y ESM2-Freeze superaron a los otros métodos en *accuracy*, AUC, *precision*, *f1-score* y MCC. En términos de AUC, TAPE-GAS y ESM2(t6)-Freeze obtuvieron 0.9841 y 0.9830 respectivamente, frente a 0.9557 y 0.9642 de NetMHCpan4.1 y MhcFlurry2.0.

Índice general

Declaración de autenticidad	I
Resumen	IV
Índice de figuras	VIII
Índice de tablas	X
Abreviaciones	XI
1. Introducción	1
1.1. Contexto y Motivación	1
1.2. Problema	3
1.2.1. Formulación del Problema	5
1.3. Objetivos	5
1.3.1. Objetivo General	5
1.3.2. Objetivos Específicos	6
1.4. Contribuciones	6
1.5. Organización del Trabajo	7
2. Marco Conceptual	8
2.1. Bioinformática	8
2.1.1. Bioinformática	8
2.1.2. DNA, RNA y Proteínas	8
2.1.3. Mutaciones	11
2.1.4. Fusión de Genes	11
2.2. Sistema Inmunitario	13
2.2.1. Células T y APC	14
2.2.2. MHC I y II	14
2.2.3. Neoantígenos	16
2.2.4. Inmunoinformática	16
2.3. <i>Machine Learning</i>	17
2.3.1. Algoritmos de Aprendizaje	17
2.3.1.1. La Tarea, T	17
2.3.1.2. El Desempeño, P	18

2.3.1.3. La Experiencia, <i>E</i>	19
2.3.2. Redes Neuronales	20
2.4. Deep Learning	22
2.4.1. Redes Neuronales Profundas	23
2.4.2. Redes Neuronales Convolucionales	25
2.4.3. Redes Neuronales Recurrentes	27
2.4.4. <i>Transformers</i>	28
2.4.4.1. <i>Self-attention</i>	29
2.4.4.2. <i>Mulitple head self-attention</i>	31
2.4.5. BERT	32
2.4.5.1. Pre-entrenamiento	32
2.4.5.2. <i>Fine-tuning</i>	33
3. Estado del Arte	35
3.1. Metodología	35
3.1.1. Preguntas de Investigación	35
3.1.2. Criterios de Inclusión y Exclusión	36
3.1.3. Estrategia de Búsqueda	36
3.2. Resultados	37
3.2.1. Consideraciones Iniciales	37
3.2.2. Detección de Neoantígenos	38
3.2.3. Priorización de Neoantígenos	38
3.2.3.1. Bases de Datos	38
3.2.3.2. Predicción de la Unión pMHC	39
3.2.3.3. Predicción de la Unión pMHC-TCR	42
3.2.4. <i>Pipelines</i>	44
3.2.5. Ensayos Clínicos	45
4. Propuesta	51
4.1. Metodología	51
4.2. Modelos BERT	52
4.2.1. Modelos de lenguaje de preoteínas	53
4.2.2. TAPE	56
4.2.3. ProtBert-BFD	56
4.2.4. ESM2	57
4.3. <i>Fine-tuning</i>	57
4.4. <i>Gradient Accumulation Steps</i>	60
4.5. Hiperparámetros	61
4.6. Ambiente de trabajo	61
4.7. Bases de datos	62
4.8. Clasificación binaria y Métricas	63
5. Experimentos	65
5.1. Modelos BERT	65
5.2. Congelación de Capas y GAS	66
6. Resultados	67
6.1. Congelación de Capas	67

6.2. <i>Vanish Gradient</i> y GAS	70
6.3. Entrenamiento (30 epochs)	71
6.4. Comparación con los Métodos del Estado del Arte	72
6.5. Herramienta para Predicción de la Unión pMHC	76
6.6. Discusión	77
6.6.1. <i>Fine-tuning ESM2</i>	77
6.6.2. Congelamiento de Capas y GAS	77
6.6.3. TAPE, ProtBert-BFD y ESM2	78
7. Conclusiones y Trabajos Futuros	80
7.1. Conclusiones	80
7.2. Trabajos Futuros	82
7.3. Agradecimiento	82

Índice de figuras

1.1.	<i>Pipeline</i> para el desarrollo de vacunas basadas en neoantígenos	3
2.1.	Localización y estructura del DNA. Fuente: NCI (2022).	9
2.2.	Transcripción y traducción. Fuente: NCI (2020).	10
2.3.	<i>Alternative Splicing</i> . Fuente: NCI (2020).	10
2.4.	Ejemplos de SNV en el DNA. Fuente: Socratic.org (2022)	12
2.5.	Ejemplos de variaciones en el DNA. Fuente: PacBio (2021)	12
2.6.	Representación de la fusión de genes.	13
2.7.	Representación de <i>Trans-splicing</i> y <i>Cis-splicing</i>	14
2.8.	Presentación de antígenos por MHC-I. Fuente: Zhang et al. (2019)	15
2.9.	Presentación de antígenos por MHC-II. Fuente: Zhang et al. (2019)	15
2.10.	Proceso para la detección de neo antígenos y generación de vacunas personalizadas. Fuente: (Mattos et al., 2020)	16
2.11.	Tipos de aprendizaje en <i>Machine Learning</i> . Fuente: (Prince, 2023)	20
2.12.	Representación de una neurona. Fuente: Raff (2022).	21
2.13.	Representación de una red neuronal.	21
2.14.	Red neuronal con una entrada, cuatro unidades ocultas y dos salidas. Fuente: Prince (2023)	22
2.15.	Relación entre Inteligencia Artificial, <i>Machine Learning</i> y <i>Deep Learning</i> . Fuente: El Naqa and Murphy (2022).	23
2.16.	Representación de un <i>Deep Feedforward Network</i> . Fuente: El Naqa and Murphy (2022).	24
2.17.	Notación matricial de una red neuronal profunda. Fuente: Prince (2023).	24
2.18.	Red neuronal convolucional 1D con un <i>kernel</i> de tamaño 3. Fuente: Prince (2023).	25
2.19.	Ejemplo del uso de <i>stride</i> , <i>kernel size</i> y <i>dilation</i> en la convolución 1D. Fuente: Prince (2023)	25
2.20.	Ejemplo de una convolución 2D en procesamiento de imágenes. Fuente: Shuchen (2022).	26
2.21.	Ejemplo de una capa convolucional 2D. Fuente: Prince (2023)	27
2.22.	Arquitectura de LeNet-5, una CNN para el reconocimiento de dígitos. Fuente: LeCun et al. (1998).	27
2.23.	Ejemplo del procesamiento del <i>input gate</i> , <i>forget gate</i> y <i>output gate</i> de LSTM. Fuente: Zhang et al. (2021)	28
2.24.	Ejemplo de como calcular <i>self-attention</i> . Fuente: Prince (2023)	30
2.25.	Proceso para procesar el <i>self-attention</i> de forma matricial. Fuente: Prince (2023)	30
2.26.	Proceso para procesar el <i>multiple head self-attention</i> . Fuente: Prince (2023)	31

2.27. Pre-entrenamiento de BERT. Fuente: Prince (2023)	32
2.28. <i>Fine-tuning</i> de BERT	33
3.1. Una visión general de cada fase del proceso de generación de vacunas personalizadas basadas en neoantígenos.	36
4.1. Propuesta de <i>transfer learning</i> de ESM-1b y una red neuronal paralela para la predicción de la afinidad entre un péptido y MHC (peptide MHC binding).	52
4.2. Arquitectura del modelo BERT	53
4.3. Ejemplo de traducción de aminoácidos	54
4.4. Estructura 3D de la proteína Mioglobulina	54
4.5. Estructura 3D de una proteína normal y una proteína mutada	55
4.6. Ejemplo de aplicación de <i>Fine-tuning</i>	59
4.7. Fine-tuning al modelo BERT	59
4.8. Ejemplo de aplicación de <i>Gradient Accumulation Steps</i>	60
4.9. Cuantificación de las muestras por <i>k-mers</i> dentro de los conjuntos de entrenamiento, validación y pruebas. El conjunto de datos se obtuvo de Anthem (Mei et al., 2021).	62
6.1. Comparación del AUC de modelos <i>Transformers</i> entrenados con 3 <i>epochs</i>	69
6.2. Gradientes del modelo ESM2(t6)	70
6.3. Gradientes del modelo ESM2(t30)	71
6.4. Comparación del AUC con métodos del estado del arte	74
6.5. Comparación de ROC con métodos del estado del arte	74
6.6. La distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 <i>epochs</i> para 8, 9, 10 y 11-mers; junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MHCflurry2.0.	75
6.7. La distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 <i>epochs</i> para 12 y 13-mers; junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MHCflurry2.0.	76
6.8. La distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 <i>epochs</i> para los péptidos 14- <i>mer</i> ; junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MHCflurry2.0.	76

Índice de tablas

3.1.	Criterios de inclusión y exclusión.	36
3.2.	Cadenas de búsqueda utilizadas para cada fase de detección de neoantígenos.	37
3.3.	Bases de datos públicas de unión pMHC e interacción pMHC-TCR	39
3.4.	Métodos basados en <i>Transformers</i> y DL con mecanismos de atención utilizados para la predicción de la unión pMHC.	47
3.5.	Métodos basados en <i>Transformers</i> y DL con mecanismos de atención utilizados para la predicción de la unión pMHC-TCR.	48
3.6.	Lista de <i>pipelines</i> desarrollados desde 2018 hasta la fecha para la detección de neoantígenos. GN: <i>Gene Expression</i> , VA: <i>Variant Annotation</i>	49
3.7.	Lista de pruebas clínicas que han utilizado vacunas personalizadas basadas en neoantígenos desde el 2028. M: muestra, FC: Fase de Cáncer, FE: Fase de ensayo.	50
4.1.	Diferencias significativas entre los modelos TAPE, ProtBert-DFB y ESM2. HS: <i>Hidden size</i> ; AH: <i>Attention heads</i>	56
4.2.	Características del bloque BiLSTM utilizado en el <i>fine-tuning</i>	60
4.3.	Ejemplo de algunas muestras de la base de datos	63
5.1.	Nomenclatura utilizada para los modelos entrenados.	66
6.1.	Comparación de los modelos <i>Transformer</i> entrenados por 3 <i>epochs</i>	68
6.2.	Comparación de los modelos <i>Transformer</i> entrenados por 30 <i>epochs</i>	72
6.3.	Evaluación del desempeño de los modelos de <i>Transformer</i> TAPE-GAS y ESM2(t6)-Freeze, entrenados durante 30 <i>epochs</i> , en comparación con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MhcFlurry2.0.	75

Abreviaciones

ANN	Artificial Neural Network
AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
bp	Base pair in DNA
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DNA	Deoxyribonucleic Acid
GNN	Graph Neural Netowrk
G-BERT	Graph Bidirectional Encoder Representations from Transformers
HLA	Human Leukocyte Antigens
MCC	Matthews Correlation Coefficient
MHC-I	Major Histocompatibility Complex Class I
MHC-II	Major Histocompatibility Complex Class II
MHC-III	Major Histocompatibility Complex Class III
mRNA	Messenger Ribonucleic Acid
NLP	Natural Language Processing
pMHC	Peptide-MHC ligand
pMHC-TCR	pMHC T-cell receptor ligand
RNA	Ribonucleic Acid
RoBERTa	Optimized BERT
RSL	Revisión Sistemática de la Literatura
tRNA	Transfer Ribonucleic Acid
TCR	T-cell receptor

Capítulo 1

Introducción

1.1. Contexto y Motivación

El cáncer representa el desafío de salud global más significativo ([Siegel et al., 2023](#)). Además, según el Instituto de Investigación del Cáncer del Reino Unido, se registraron más de 18 millones de nuevos casos y 10 millones de muertes en 2020 ([UK, 2023b](#)). Además, se predice que habrá alrededor de 28 millones de nuevos casos anualmente para alrededor de 2040 si la incidencia se mantiene estable y el crecimiento de la población y el envejecimiento continúan según las tendencias recientes ([UK, 2023a](#)). Esto representa un aumento del 54.9 % desde 2020, con un aumento esperado mayor en hombres (60.6 %) que en mujeres (48.8 %).

En este contexto, se sabe que los métodos tradicionales basados en cirugía, radioterapia y quimioterapia tienen baja eficacia y efectos secundarios adversos ([Peng et al., 2019](#)). Por lo tanto, ha surgido el desarrollo de la inmunoterapia contra el cáncer, con el objetivo de estimular el sistema inmunológico del paciente ([Borden et al., 2022](#)). Existen tratamientos como vacunas personalizadas, terapias con linfocitos T adoptivos e inhibidores de puntos de control inmunológico. De entre estos, las vacunas basadas en neoantígenos han mostrado un gran potencial al potenciar las respuestas de los linfocitos T y se consideran las más propensas a tener éxito ([Borden et al., 2022](#)). Además, los neoantígenos se utilizan en la terapia de bloqueo de puntos de control inmunológico. Los neoantígenos se consideran biomarcadores predictivos y objetivos para el tratamiento sinérgico en la inmunoterapia contra el cáncer ([Fang et al., 2022a](#)).

El desarrollo de vacunas personalizadas contra el cáncer es un proceso largo que depende de la detección precisa de neoantígenos (ver Figura 1.1). Estos neoantígenos son péptidos que se encuentran exclusivamente en las células cancerosas. El objetivo de un tratamiento

basado en vacunas personalizadas es entrenar a los linfocitos (células T) del paciente para que reconozcan estos neoantígenos y activen el sistema inmunológico ([Mattos et al., 2020](#); [Peng et al., 2019](#)). El proceso se resume en la Figura 1.1(b) y consta de las siguientes fases:

1. Obtener muestras de tejidos cancerosos y sanos. Ambos tejidos se secuencian para obtener ADN y/o ARN. Algunos enfoques incluyen información del *immunopeptidome* obtenida mediante *Mass Spectrometry* (MS).
2. En la etapa *in-silico*, se realiza el alineamiento de secuencias, se desarrolla un proceso de llamada de variantes para detectar variaciones y/o mutaciones, y se anotan estas variantes (posible detección de neoantígenos). Hay disponibles varias herramientas con buen rendimiento para esta etapa.
3. En esta etapa *in-silico*, se priorizan los neoantígenos. Este paso es crucial y ha recibido una atención significativa en la investigación en los últimos años debido a su complejidad y la baja efectividad de los enfoques actuales. Aquí, se evalúa la afinidad de los candidatos neoantígenos (péptidos) de la etapa anterior con el *Major Histocompatibility Complex* (MHC), conocido como la unión pMHC. Luego, se evalúa la afinidad de pMHC para unirse al *T-cell Receptor* (TCR). Al final de esta etapa, se obtienen los neoantígenos.
4. En la etapa *in-vitro*, en el laboratorio se inducen las células T del paciente para que reconozcan los neoantígenos. En este punto, se desarrollan las vacunas. Esta etapa la llevan a cabo biotecnólogos y biólogos.
5. Finalmente, el oncólogo realiza una evaluación clínica de la vacuna.

La detección *in-silico* de neoantígenos se basa en las etapas segunda y tercera representadas en la Figura 1.1(b). En este contexto, debido a la complejidad del proceso y la variedad de métodos disponibles, se han desarrollado herramientas de *software* y flujos de trabajo para agilizar el uso de estas herramientas. Además, los *Transformers* han marcado el comienzo de una nueva era en la inteligencia artificial, demostrando logros destacados en una variedad de tareas de procesamiento del lenguaje natural ([Patwardhan et al., 2023](#)). Estos modelos también han encontrado aplicación en la detección de neoantígenos, especialmente en la tercera etapa de la Figura 1.1(b). Se han propuesto modelos BERT y redes de aprendizaje profundo con mecanismos de atención para predecir la unión péptido-MHC y pMHC-TCR obteniendo resultados prometedores. Sin embargo, aún existe mucho camino por recorrer y con el incremento constante

de muestras de ADN/proteínas, sumado a los nuevos mecanismos para entrenar modelos *Transformers* con miles de millones de parámetros (10^9), se espera lograr avances significativos en este campo de estudio.

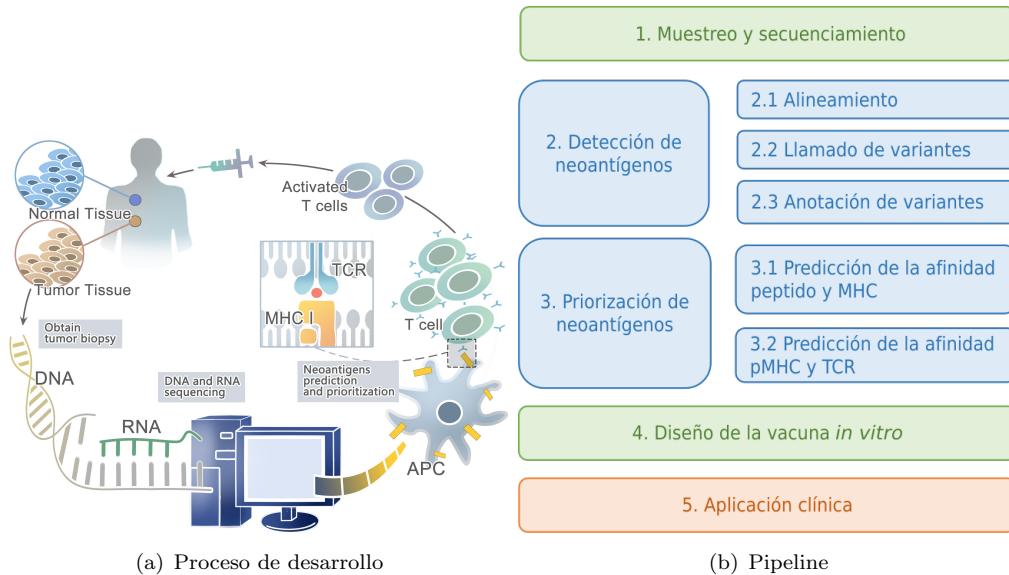


FIGURA 1.1: Marco de desarrollo para la creación de vacunas personalizadas contra el cáncer basadas en neoantígenos. (a) proporciona una visión general de cada etapa (Han et al., 2020). (b) una visión general de cada fase con un énfasis en el desarrollo *in-silico*.

1.2. Problema

Los neoantígenos son péptidos mutados específicos de tumores y son considerados los principales causantes de una respuesta inmune (Borden et al., 2022; Chen et al., 2021b; Gopanenko et al., 2020). Es así que surgen varios esfuerzos e investigación en la Inmunoterapia del cáncer, concentradas en el estudio y detección de neoantígenos. Es así, que el desarrollo de vacunas personalizadas basadas en neoantígenos es considerado uno de los métodos con mayor probabilidad de éxito (Borden et al., 2022). Incluso varias compañías como BioNTech, Genocea Biosciences, Neon Therapeutics y Gritstone Oncology realizan investigación y ofrecen el servicio de generar vacunas personalizadas a pacientes de cáncer.

Además, el MHC representa un factor clave, en la detección de neoantígenos, al ser el encargado de unirse al neoantígeno y presentarlo a la superficie de la célula. Debido a esto, este trabajo enfoca en el desarrollo de un método para la predicción del enlace entre neoantígenos y MHC (*pMHC binding*), esto corresponde a la Fase 3.1 de la Figura 1.1(b) dentro del *pipeline* de detección de neoantígenos. Existen dos tipos: el MHC clase I (MHC-I) y MHC clase II (MHC-II), ambos presentan péptidos en la

superficie celular a las células T CD8+ y CD4+, respectivamente ([Janeway Jr, 1997](#); [Abualrous et al., 2021](#)). En detalle, el ciclo de vida de los neoantígenos que se unen a MHC-I se puede resumir de la siguiente manera. Primero, una proteína cancerígena se degrada en péptidos en el citoplasma. Luego, los péptidos se unen al MHC (*pMHC binding*). Despues, este compuesto sigue un camino hasta llegar a la membrana celular (*pMHC presentation*). Finalmente, el pMHC es reconocido por el TCR, desencadenando el sistema inmunológico ([Janeway Jr, 1997](#); [Wieczorek et al., 2017](#); [Gasser et al., 2021](#)). Por lo tanto, la unión pMHC es un paso muy importante para la inmunidad celular, y la predicción y comprensión de esta unión tienen un valioso potencial. Lamentablemente, la mayoría de las ligandos pMHC no llegan a la membrana celular ([Mattos et al., 2020](#)).

Adicionalmente, las proteínas MHC están codificadas por genes altamente polimórficos, llamados *Antígenos Leucocitarios Humanos* (HLA); la considerable naturaleza polimórfica de los genes MHC proporciona una variación sustancial en la unión con los neoantígenos, lo que influye en el conjunto de neoantígenos presentados a las células T ([Abualrous et al., 2021](#)). En consecuencia, los métodos propuestos se categorizan como *pan-specific* y *allele-specific*. Los métodos *allele-specific* ([Rammensee et al., 1999](#); [Reche et al., 2002](#); [Kim et al., 2009](#); [Nielsen and Andreatta, 2016](#); [Vang and Xie, 2017](#); [Shao et al., 2020](#); [Bravi et al., 2021](#)) entran un modelo para cada *allele* del MHC; mientras que los métodos *pan-specific* ([Hu et al., 2019](#); [Liu et al., 2019](#); [Wu et al., 2019](#); [Phloyphisut et al., 2019](#); [O'Donnell et al., 2018, 2020](#); [Reynisson et al., 2020a](#); [Venkatesh et al., 2020](#); [Ye et al., 2021](#); [Mei et al., 2021](#); [Chu et al., 2022](#); [Zhang et al., 2022b](#); [Mei et al., 2021](#); [Hu et al., 2019](#); [Gfeller et al., 2023](#)) entran un modelo global que toma péptidos (neoantígenos) y MHC como entradas. Además, la naturaleza polimórfica del MHC eleva bastante la complejidad de este problema, se cree que existen las 10000 diferentes MHC *alleles* ([Abelin et al., 2017](#)), esto complica mucho la detección de neo antígenos. Por lo tanto, los métodos *pan-specific* surgen con una alta posibilidad de futuras aplicaciones.

Lamentablemente, a pesar de varios esfuerzos en el desarrollo de métodos para la detección de neoantígenos, menos del 5 % de neoantígenos detectados activan el sistema inmune ([Mattos et al., 2020](#); [Mill et al., 2022](#); [Bulik-Sullivan et al., 2019](#); [Bassani-Sternberg et al., 2015](#); [Yadav et al., 2014](#)). Según los autores de los métodos, las razones son:

1. La no inclusión en conjunto de varias fuentes de información como DNA-seq, RNA-seq, y datos de *Mass Spectrometry* (MS) ([Kim et al., 2018](#)). Por ejemplo, la mayoría de propuestas no utiliza datos de MS; en la actualidad, existe una creciente información de estos datos y se están aplicando a varios campos de la Bioinformática.
2. Uso herramientas de bajo desempeño para la predicción del enlace péptido-MHC (pMHC) (etapa 3.1 de la Figura 1.1(b)). La mayoría de aplicaciones, se basa en el

uso de MHCFlurry ([O'Donnell et al., 2020](#)) y NetMHCpan4.1 ([Reynisson et al., 2020a](#)). Sin embargo, actualmente, se cuenta con herramientas de mejor desempeño basado en *Transformers* ([Arceda, 2023](#)). Esta tesis, se enfoca en resolver este problema.

3. Para la etapa 3.2 de la Figura [1.1\(b\)](#), los autores no consideran la predicción del enlace pMHC al TCR (pMHC-TCR), varios autores consideran incluir esta tarea en trabajos futuros ([Rubinsteyn et al., 2018](#)).
4. Finalmente, no utilizar información de eventos de *alternative splicing*, variaciones estructurales en el ADN y las mutaciones de fusión de genes, está información esta fuertemente relacionada con varios tipos de cáncer ([Wood et al., 2020](#)).

En conclusión, la detección de neoantígenos es un desafío que consta de múltiples etapas, y las herramientas actuales en el estado del arte presentan un rendimiento insuficiente. Uno de los factores clave detrás de este bajo rendimiento está relacionado con la predicción del enlace pMHC. Por esta razón, esta tesis se centra en abordar este problema mediante la propuesta de un método basado en *Transformers* para la predicción del enlace/unión pMHC.

1.2.1. Formulación del Problema

El presente estudio se centra en el problema de predicción del enlace pMHC-I (*pMHC binding prediction*). Esto representa un problema de clasificación binaria que toma como entrada la secuencia de aminoácidos de un péptido y el MHC. Un péptido podría representarse como: $p = \{A, \dots, Q\}$ y una representación similar para el MHC sería: $q = \{A, N, \dots, G\}$. Finalmente, necesitamos conocer la probabilidad de afinidad entre p y q . Si esta probabilidad es lo suficientemente alta, es posible que el péptido se enlace al MHC y por lo tanto, el péptido p en cuestión, sería un excelente candidato a neoantígeno.

1.3. Objetivos

1.3.1. Objetivo General

Implementar un método *in silico* basado en *Transformers* y *Transfer Learning* para la detección de neoantígenos, enfocados en la predicción de la unión pMHC.

1.3.2. Objetivos Específicos

- (a) Analizar los métodos que utilizan *Transformers* para la predicción del enlace pMHC en el contexto de detección de neoantígenos.
- (b) Analizar los modelos basados en *Transformers* TAPE, ProtBert-BFD, y EMS2 pre-entredados para diversas tareas en Proteómica y de los cuales se puede aplicar *Transfer Learning*.
- (c) Implementar *fine-tuning* a los modelos TAPE, ProtBert-BFD, y EMS2 para la tarea de predicción del enlace pMHC, aplicando *Gradient Accumulation Steps* (GAS) y una metodología de congelamiento de capas.
- (d) Comparar los modelos de mejor desempeño con las herramientas del estado del arte como: NetMHCpan4.1, MHCFlurry2.0, Anthem, ACME y MixMHCpred2.2.

1.4. Contribuciones

Las principales contribuciones de este trabajo son:

- (a) Revisión y análisis de los métodos basados en *Transformers* para la detección de neoantígenos. Se cuenta con dos publicaciones: “*Deep Learning and Transformers in MHC-Peptide Binding and Presentation Towards Personalized Vaccines in Cancer Immunology: A Brief Review*” ([Machaca et al., 2023](#)) y “*Transformers Meets Neoantigen Detection: A Systematic Literature Review*”.
- (b) *Fine-tuning* a seis modelos de *Transformers* para la predicción del enlace pMHC; además, se ha evaluado el uso de GAS y una metodología de congelamiento de capas. Se cuenta con dos publicaciones: “*Neoantigen Detection Using Transformers and Transfer Learning in the Cancer Immunology Context*” ([Arceda, 2023](#)) y “*Fine-tuning Transformers for Peptide-MHC Class I Binding Prediction*”.
- (c) Comparación de los métodos propuestos con herramientas del estado del arte como: NetMHCpan4.1, MHCFlurry2.0, Anthem, ACME y MixMHCpred2.2. Los métodos propuestos tienen los mejores resultados en *accuracy*, *Area Under the Curve* (AUC), *recall*, *f1-score* y *Matthews Correlation Coefficient* (MCC).

1.5. Organización del Trabajo

El restante de este trabajo está organizado de la siguiente manera:

- En el Capítulo 2 se presentan los conceptos básicos sobre Bioinformática e inmunoterapia del Cáncer, también son abordados temas sobre *Deep Learning* y redes neuronales *Transformers*.
- Luego, en el Capítulo 3 se describen los trabajos relacionados a la presente tesis. Debido a la gran cantidad de publicaciones, solo se ha considerado trabajos desde el 2018 y que hacen uso de *Transformers* o redes neuronales con mecanismos de atención.
- El Capítulo 4, presenta la propuesta de la tesis. Esta se basa en un método para desarrollar *fine-tuning* a *Transformers* pre-entrenados para diversas tareas de Proteómica.
- Luego, en el Capítulo 6, se presentan los resultados de la investigación. Además, se presenta una comparación con los métodos del estado del arte.
- Finalmente, en el Capítulo 7 son expuestos las conclusiones del presente trabajo así como también las direcciones para continuar con el mismo en la sección de trabajos futuros.

Capítulo 2

Marco Conceptual

El proyecto pertenece al área de Bioinformática y específicamente a la Inmunoinformática, en este contexto el marco teórico detalla conceptos de Biología Molecular (ADN, ARN y proteínas), Inmunología y Ciencias de la Computación.

2.1. Bioinformática

En esta sección, describiremos los principales conceptos referentes a Bioinformática que serán considerados en la propuesta de la tesis.

2.1.1. Bioinformática

Según [Luscombe et al. \(2001\)](#), la Bioinformática involucra la tecnología que utiliza las computadoras para el almacenamiento, manipulación y distribución de información relacionada a la Biología Molecular como DNA, RNA y proteínas. También podemos considerar que la Bioinformática se enfoca al análisis de secuencias, estructuras y funciones de los genes y proteínas. Adicionalmente, algunos autores como [Xiong \(2006\)](#) consideran a la Biología Molecular Computacional como un sinónimo a la Bioinformática.

2.1.2. DNA, RNA y Proteínas

Deoxyribonucleic Acid (DNA) es una molécula dentro de las células que contiene información genética responsable del desarrollo y función del organismo ([NCI, 2022](#)). Gran parte del DNA se sitúa dentro del núcleo de las células (en organismos Eucariontes). Por ejemplo en la Figura [2.1](#), vemos como el DNA, forma parte de los cromosomas

y estos a su vez están en el núcleo. Luego, podemos notar, que los genes representan segmentos del DNA. Finalmente, en la Figura 2.1, notamos las bases nitrogenadas que componen el DNA: *Guanine*, *Cytosine*, *Adenine* y *Thymine*; normalmente, estas bases serán representadas por las letras: G, C, A, T respectivamente.

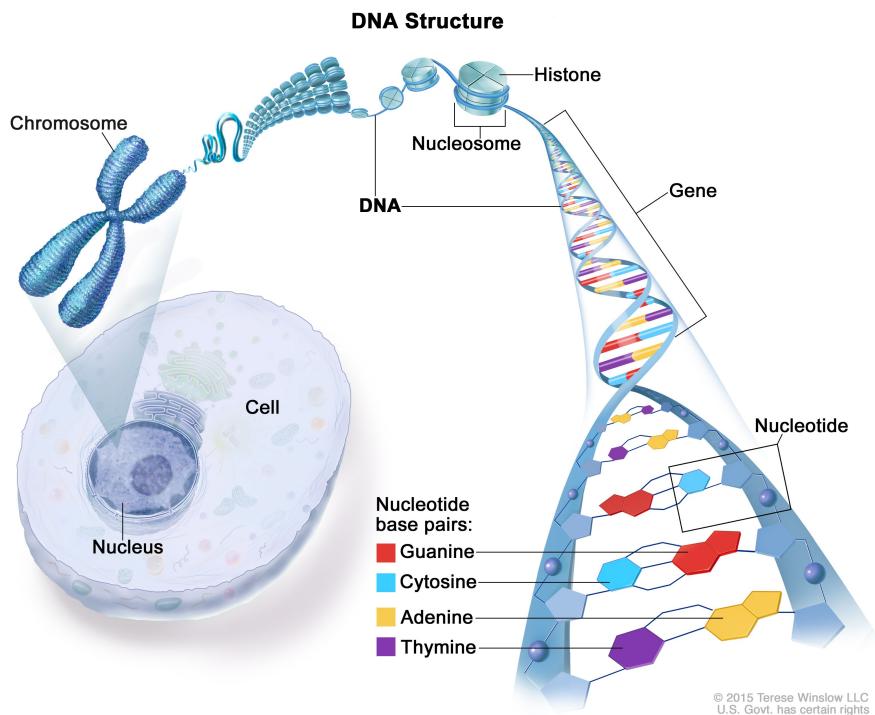


FIGURA 2.1: Localización y estructura del DNA. Fuente: [NCI \(2022\)](#).

Durante el ciclo de vida de la célula, ocurre un proceso llamado Transcripción (ver Figura 2.2), en este proceso se generan cadenas de *Ribonucleic Acid* (RNA) a partir de la cadena de DNA ([NCI, 2022](#)). Durante este proceso la base nitrogenada *Thymine* (T) es reemplazada por *Uracil* (U). El proceso mencionado, ocurre dentro del núcleo de la célula y en esta etapa el RNA es llamado *messenger RNA* (mRNA). Una vez el mRNA sale del núcleo, es transportado por *transfer RNA* (tRNA) hacia los Ribosomas (ver Figura 2.2). En esta última etapa ocurre la Traducción, cada grupo de tres bases nitrogenadas (codones) se convierten en un aminoácido diferente, luego estos aminoácidos forman cadenas polipeptídicas y estas a su vez forman las proteínas; normalmente, cada gen genera una proteína ([Xiong, 2006](#); [NCI, 2022](#)).

Durante el proceso de Traducción, puede ocurrir un fenómeno llamado *Alternative Splicing*. Por ejemplo, en la Figura 2.3, notamos como un gen puede generar tres proteínas distintas, cada una con funciones distintas. Este fenómeno, complica bastante el análisis de DNA.

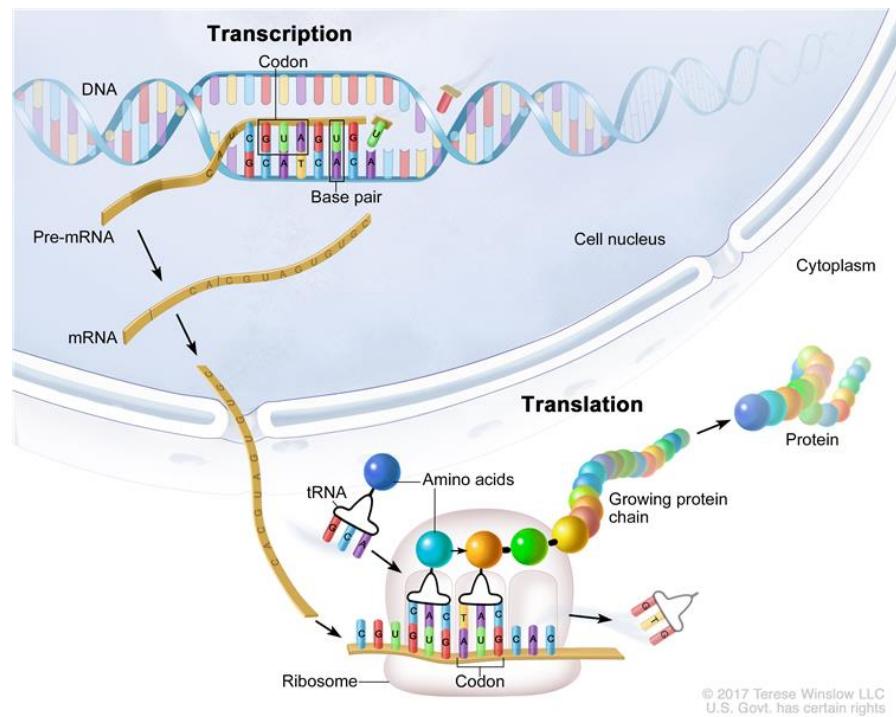
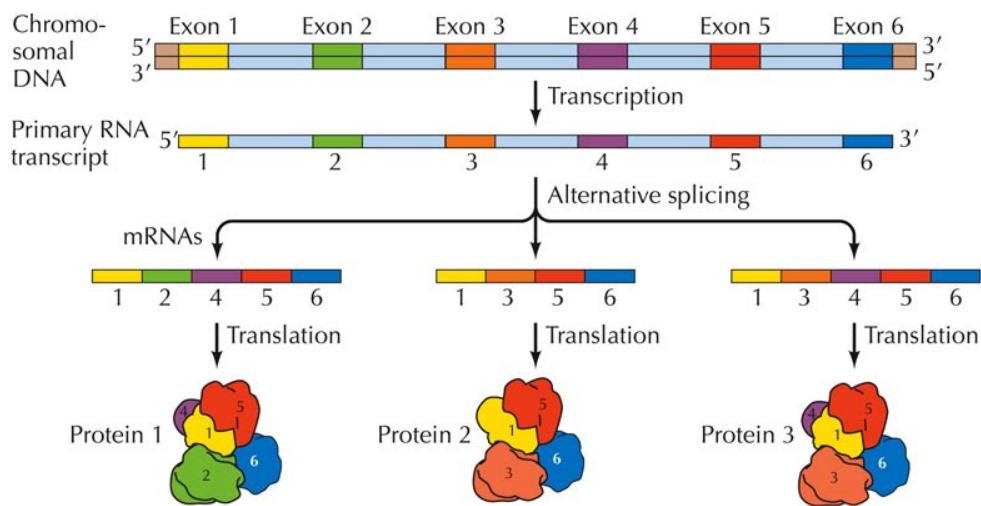


FIGURA 2.2: Transcripción y traducción. Fuente: NCI (2020).



THE CELL, Fourth Edition, Figure 5.5 © 2006 ASM Press and Sinauer Associates, Inc.

FIGURA 2.3: Alternative Splicing. Fuente: NCI (2020).

2.1.3. Mutaciones

Las mutaciones también llamadas variaciones, representan cualquier cambio en la secuencia de DNA, estos pueden ocurrir durante la división celular o por la exposición a agentes químicos o radioactivos. Estas mutaciones pueden ser beneficiosas, dañinas (cuando afectan la generación de proteínas) o no tener algún efecto (NCI, 2022). Varios tipos de Cáncer son ocasionados por estas mutaciones (Borden et al., 2022; Chen et al., 2021b; Mattos et al., 2020).

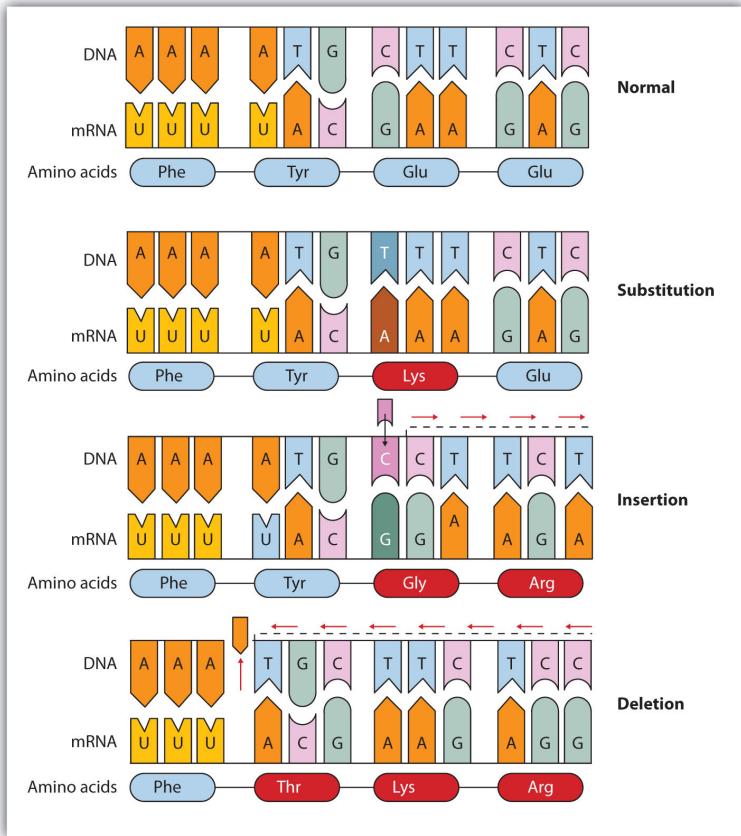
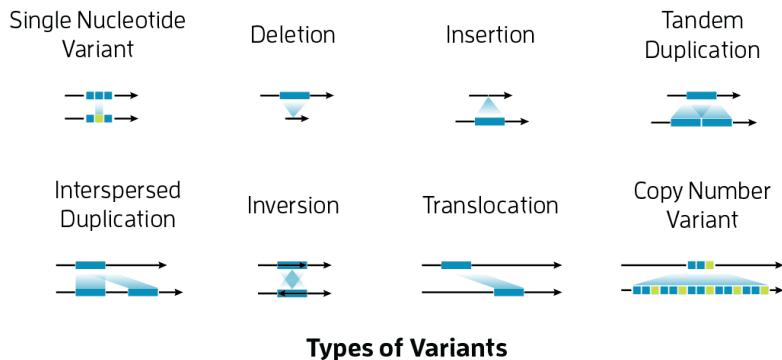
Según el tipo de célula afectada, tenemos: mutaciones somáticas y mutaciones *germline* (una mutación en estas células puede ser heredada a la descendencia) (Clancy, 2008). Según (Xu, 2018), las variaciones genómicas pueden clasificarse en tres grupos: *Single-Nucleotide Variant* (SNV), inserciones y eliminaciones (INDELS) y *Structural Variation* (SV). Una mutación se considera SNV cuando las variaciones afectan a menos de 10 bases.

En la Figura 2.4, presentamos ejemplos de SNV. Por ejemplo, las sustituciones pueden afectar la generación de un aminoácido, pero las inserciones o eliminaciones pueden afectar en cadena la generación de varios aminoácidos, a este tipo de fenómeno se le conoce como *frameshift mutation* (Xu, 2018).

En la Figura 2.5, mostramos algunos tipos de SV. En este caso, también se pueden presentar INDELS, *Tandem duplication*, inversiones, translocaciones y *Copy Number Variants* (CNV). Los CNVs, representan fuertes candidatos para ser biomarcadores de varios tipos de Cáncer (Pan et al., 2019; Lucito et al., 2007). Otra mutación importante, es referente a la fusión de genes, en estos casos dos o más genes se fusionan y forman una proteína completamente diferente, este tipo de mutación también está fuertemente relacionado a varios tipos de Cáncer (Kerbs et al., 2022; Kim and Zhou, 2019; Heyer and Blackburn, 2020).

2.1.4. Fusión de Genes

Según Williford and Betrán (2013), la fusión de genes es un proceso mediante el cual las secuencias completas o parciales de dos o más genes distintos se fusionan en un solo gen, como resultado de reordenamientos derivados de ADN o ARN. Este fenómeno es ampliamente distribuido y se ha observado en todos los dominios de la vida. Además, la fusión de genes contribuyen de manera destacada al cambio evolutivo al proporcionar una fuente continua de nuevos genes. Las duplicaciones génicas a menudo preceden a las fusiones génicas, permitiendo la evolución de genes quiméricos, al mismo tiempo que preservan las funciones originales. A pesar de la reputación de las fusiones génicas

FIGURA 2.4: Ejemplos de SNV en el DNA. Fuente: [Socratic.org \(2022\)](https://www.socratic.org/)FIGURA 2.5: Ejemplos de variaciones en el DNA. Fuente: [PacBio \(2021\)](https://www.pacbionet.com/)

como impulsores de la evolución adaptativa, pueden tener consecuencias devastadoras, a menudo dando lugar a trastornos genómicos o cáncer.

Las fusiones de genes suelen ser causadas por alteraciones en la estructura genómica resultantes de daños en el ADN y por la posterior recombinación y replicación erróneas. Como se ve en la Figura 2.6, las reorganizaciones genómicas pueden ocurrir entre uno o dos genes independientes a través de seis mecanismos conocidos: translocación, inserción, inversión, duplicación en tandem, eliminación y cromotripsis ([Taniue and Akimitsu, 2021](#); [Dai et al., 2018](#)).

Adicionalmente, la fusión de genes puede ser generada por eventos *Trans-splicing* y *Cis-splicing* (ver Figura 2.7). En *Trans-splicing*, los exones de diferentes transcripciones de ARN se empalman y fusionan para producir un solo ARNm maduro. Otro mecanismo de empalmado es el *Cis-splicing*, en el cual dos genes vecinos son transcritos en un solo ARN precursor mediante la lectura continua de la transcripción (Taniue and Akimitsu, 2021).

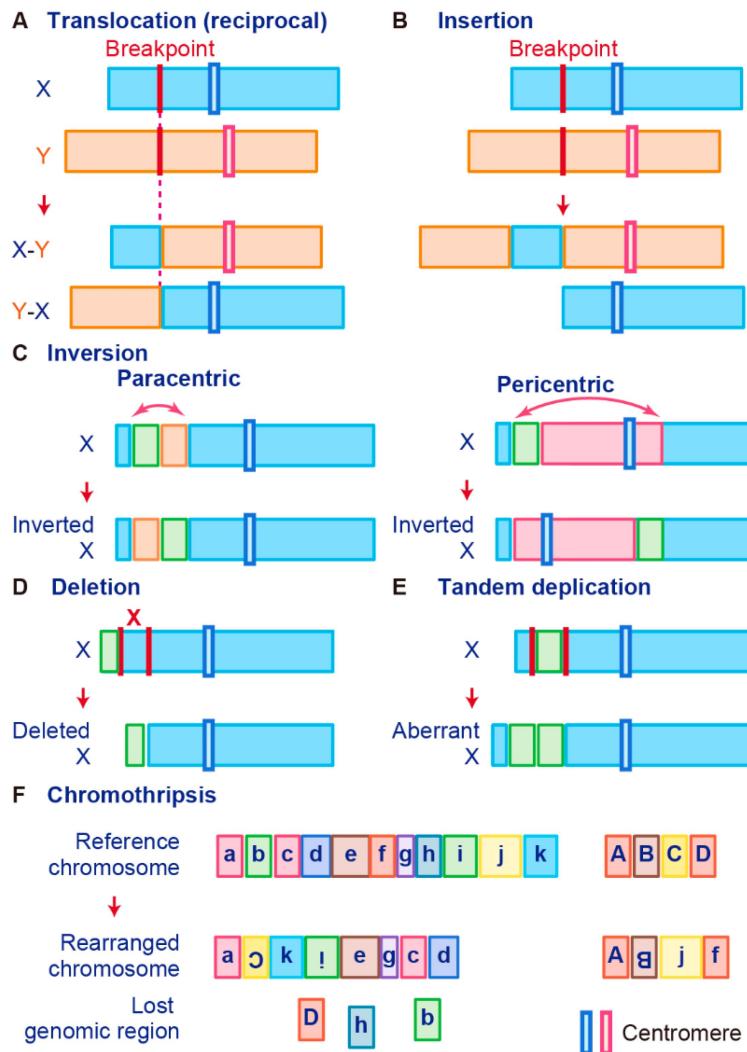


FIGURA 2.6: Representación esquemática de la formación de fusión de genes mediante reordenamientos cromosómicos estructurales. (A) Translocación. (B) Inserción. (C) Inversión. (D) Eliminación. (E) Duplicación en tandem. (F) Cromotripsis. Fuente: [Taniue and Akimitsu \(2021\)](#)

2.2. Sistema Inmunitario

El sistema inmunitario hace referencia al conjunto de células y procesos químicos que tiene como función protegernos de agentes extraños como: microbios, bacterias, células de

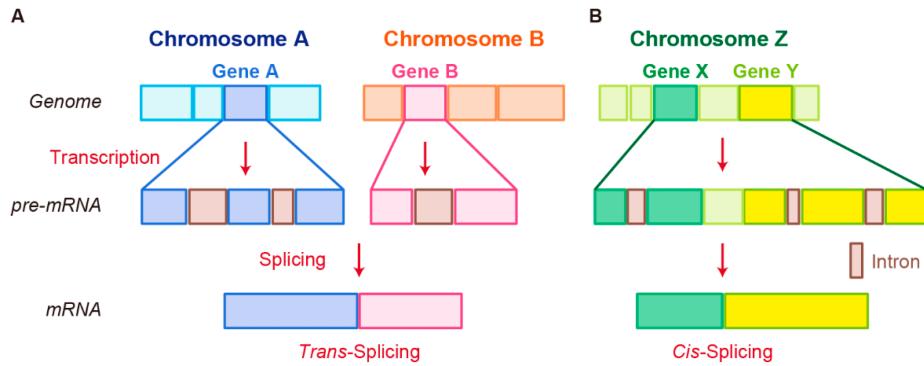


FIGURA 2.7: Representación esquemática de la formación de ARN de fusión mediante reordenamientos cromosómicos no estructurales. (A) (A) *Trans-splicing*. (B) *Cis-splicing*. Fuente: [Taniue and Akimitsu \(2021\)](#)

Cáncer, toxinas, etc. [Marshall et al. \(2018\)](#). En esta sección, se explicará de forma breve el comportamiento del sistema inmunitario frente cuando un agente extraño (antígeno) ingresa al cuerpo humano.

2.2.1. Células T y APC

Las células T también llamadas linfocitos T, se forman a partir de la médula ósea y son los encargados de eliminar agentes extraños (antígenos) [NCI \(2022\)](#). Estas células están compuestas por un *T-cell Receptor* (TCR), que es el encargado de reconocer y enlazar a los antígenos. Luego, algunas células T, requieren de la acción de los *Antigen Presenting Cells* (APC), estas células APC son: células dendríticas, macrófagos, células B, fibroblastos y células epiteliales. Normalmente, los APC devoran los antígenos y luego los presentan a las células T para su eliminación ([Marshall et al., 2018](#)).

2.2.2. MHC I y II

Las proteínas *Major Histocompatibility Complex* (MHC) I y II desempeñan un rol importante en el sistema inmunitario. Ambas proteínas tienen la función de presentar péptidos (antígenos) en la superficie de las células, para que sean reconocidas por las células T ([Abualrous et al., 2021](#)). MHC-I se encarga de la presentación de las células con núcleo, mientras que MHC-II, de las células APC.

El proceso de presentación de los antígenos por MHC-I es el siguiente (Figura 2.8): la proteína foránea es degradada por el proteasoma y se producen péptidos (posibles antígenos), luego estos péptidos son transportados al *Endoplasmic Reticulum* (ER) con la ayuda de *Transporter associated Antigen Processing* (TAP), luego es migrado al aparato de Golgi para ser presentado en la superficie de la célula y es enlazado a la proteína

MHC-I, una vez en la superficie, el antígeno puede ser reconocido por las células CD8+T ([Zhang et al., 2019](#)).

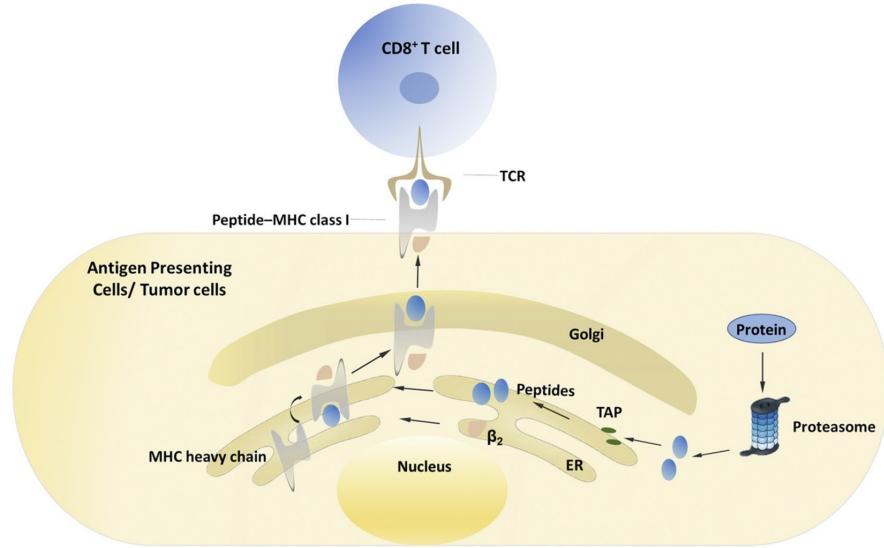


FIGURA 2.8: Presentación de antígenos por MHC-I. Fuente: [Zhang et al. \(2019\)](#)

Para el caso de MHC-II, es un proceso similar (Figura 2.9): primero, los patógenos son devorados por fagocitosis, los péptidos asociados a MHC-II son producidos en el *Endoplasmic Reticulum* (ER), para luego ser trasladados al aparato de Golgi, y luego ser transportados a la superficie de las células una vez enlazadas con MHC-II, finalmente, son reconocidas por las células CD4+T ([Zhang et al., 2019](#)).

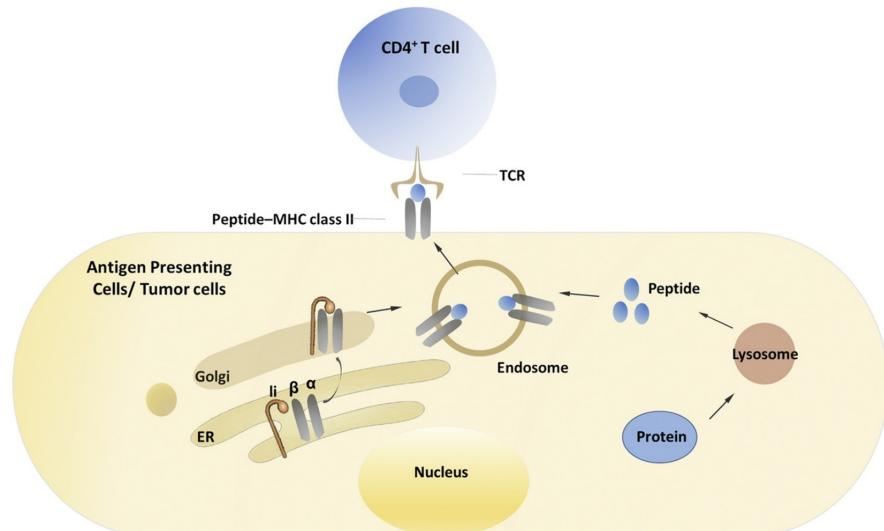


FIGURA 2.9: Presentación de antígenos por MHC-II. Fuente: [Zhang et al. \(2019\)](#)

2.2.3. Neoantígenos

Es una proteína que se forma en las células de Cáncer cuando ocurre mutaciones en el DNA. Los neo antígenos cumplen un rol importante al estimular una respuesta inmune en contra de células de Cáncer. En la actualidad, se estudia su uso en el desarrollo de vacunas contra el Cáncer [NCI \(2022\)](#). Una característica importante de los neo antígenos, es que solo están presentes en células tumorales y no en células sanas, debido a eso son considerados factores clave en la inmunoterapia del Cáncer [Borden et al. \(2022\)](#). En la actualidad hay varios métodos para detectar a predecir neo antígenos, pero solo una pequeña porción de ellos logran estimular al sistema inmune [Chen et al. \(2021b\); Hao et al. \(2021\)](#).

Este proceso para la detección de neo antígenos, generalmente consiste en: (1) extracción del tejido tumoral, (2) identificación de mutaciones, (3) detección de neo antígenos y predicción de inmunogenicidad, (4) desarrollo de experimentos *in vitro* y (5) desarrollo de la vacuna ([Mattos et al., 2020; Peng et al., 2019](#)) (ver Figura 2.10).

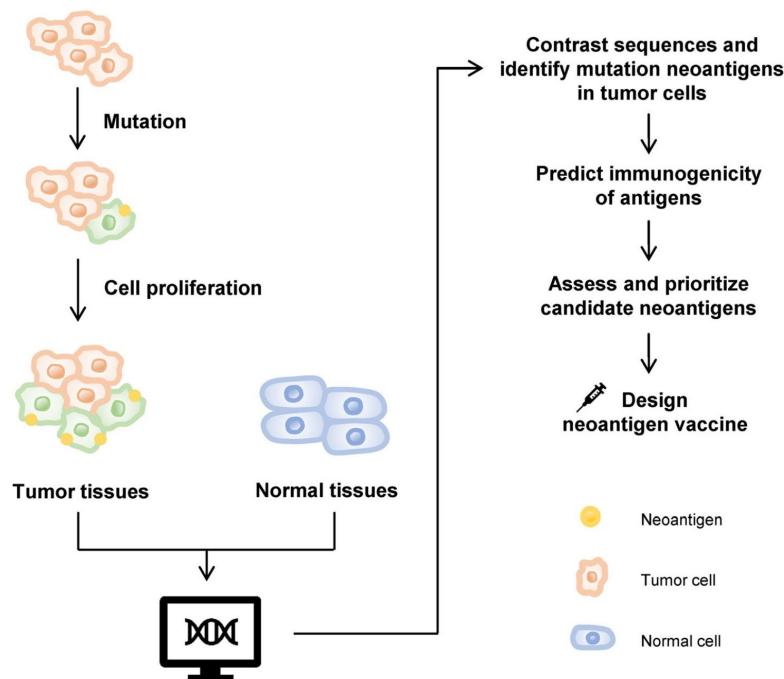


FIGURA 2.10: Proceso para la detección de neo antígenos y generación de vacunas personalizadas. Fuente: ([Mattos et al., 2020](#))

2.2.4. Inmunoinformática

Según [Tong and Ren \(2009\)](#), la Inmunoinformática, combina la inmunología tradicional con ciencias de la computación, matemáticas, química, bioquímica, genómica y proteómica para el análisis a gran escala de la función del sistema inmunológico, ofrece

nuevas oportunidades para la investigación futura, desde el laboratorio hasta el paciente. Gracias a las nuevas tecnologías y algoritmos en las Ciencias de la Computación, este campo ha generado bastante investigación y promete ser una área de riguroso estudio en el futuro. Además, esta tesis se sitúa en este campo de estudio, al aplicar metodologías de Inteligencia Artificial a la Inmunología.

2.3. *Machine Learning*

Machine Learning (ML) es una categoría de algoritmos computacionales capaces de emular algunas acciones inteligentes. Es el resultado de varias disciplinas como: inteligencia artificial, probabilidad, estadística, ciencia de la computación, teoría de la computación, psicología y filosofía ([El Naqa and Murphy, 2022](#)). *Machine Learning* tiene varias definiciones, pero una de las mas acertadas, según [Samuel \(1967\)](#): “Campo de estudio que brinda a las computadoras la habilidad de aprender sin haber sido explícitamente programado”.

2.3.1. Algoritmos de Aprendizaje

Un algoritmo de aprendizaje o *machine learning algorithm*, es aquel algoritmo que no debe ser programado explícitamente, este aprende de la experiencia, a partir de datos ([Goodfellow et al., 2016](#)). Según [Mitchell \(1997\)](#): “*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*”. La traducción a español indicaría: “Un programa de computadora puede aprender de una experiencia E , para una tarea T y con una métrica de desempeño P , si el desempeño de la tarea T , medido con P , mejorar con la experiencia E ”. Esto, nos da a entender que un programa de computadora puede aprender si mejora su desempeño según aumente su experiencia o datos.

2.3.1.1. La Tarea, T

La tarea T de ML, puede ser descrito como de la forma en que el sistema de ML procesa una muestra o ejemplo. Según [Goodfellow et al. \(2016\)](#) las tareas más comunes de ML son:

- **Clasificación.** En este caso, el algoritmo de ML debe predecir la clase a la que pertenece la muestra como una función: $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$. También puede escribirse como: $y = f(x)$, aquí x representa la entrada y la función f determinará la clase a la que pertenece.
- **Regresión.** El algoritmo debe producir una función: $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Es decir, dada como entrada un vector x de números reales, el algoritmo de ML debe predecir un valor en los números reales.
- **Transcripción.** En este caso, dada como entrada datos no estructurados, el algoritmo de ML debe generar información de forma textual. Por ejemplo: dada una imagen como entrada, la salida sería el texto encontrado en la imagen.
- **Maquinas de traducción.** Como el nombre indica, la entrada es un texto en un lenguaje y la salida es un texto en otro lenguaje.
- **Salida estructurada.** En este caso la salida es un vector o alguna estructura de datos de varios valores. El procesamiento natural de lenguaje es un buen ejemplo, la entrada es un texto y la salida es un árbol que denota la estructura gramatical y semántica de la entrada.
- **Detección de anomalías.** En este tipo de problemas el algoritmo de ML, busca detectar eventos anómalos, es decir muestras que no corresponden a la distribución normal de los datos. Un ejemplo, es la detección de transacciones fraudulentas.
- **Síntesis y muestreo.** En este caso, el algoritmo de ML debe generar nuevas muestras a partir de un conjunto de entrenamiento. Esto se aplica en los videojuegos, para la generación automática de texturas para objetos de gran tamaño.

2.3.1.2. El Desempeño, P

Es muy importante medir el desempeño de un algoritmo de ML, usualmente la métrica utilizada puede variar según la tarea T . Para tareas de clasificación, usualmente se suele aplicar *Precision* y *Recall*, estos están detallados en las Ecuaciones 2.1 y 2.2 respectivamente ([Dalianis, 2018](#)).

$$\text{Precision: } P = \frac{tp}{tp + fp} \quad (2.1)$$

$$\text{Recall: } R = \frac{tp}{tp + fn} \quad (2.2)$$

donde tp , hace referencia a la cantidad de muestras que eran verdaderas y han sido reconocidas como verdaderas; fp , son las muestras que eran falsas, pero fueron reconocidas como verdaderas; fn , son las muestras que eran negativas y fueron reconocidas como negativas. Otra métrica importante es el *F-score*, este puede ser definido como el peso promedio de *Precision* y *Recall* (Dalianis, 2018). En la Ecuación 2.3, presentamos la definición.

$$F\text{-score: } F_{\beta} = (1 + \beta^2) * \frac{P * R}{\beta^2 * P + R} \quad (2.3)$$

Cuando $\beta = 1$:

$$F\text{-score: } F_1 = 2 * \frac{P * R}{P + R} \quad (2.4)$$

Finalmente otra métrica, aunque no muy recomendada para datos no balanceados es el *accuracy*. Este representa el porcentaje de muestras reconocidas correctamente.

$$Accuracy: acc = \frac{tp + tn}{tp + tn + fp + fn} \quad (2.5)$$

Para otro tipo de problemas, como regresión se puede aplicar el *error rate*, esta es una medida en los números reales y nos indica que tan diferente es la predicción realizada por un algoritmo de ML (Goodfellow et al., 2016).

2.3.1.3. La Experiencia, E

Según el tipo de experiencia que realizan los algoritmos de ML, se pueden clasificar en: Aprendizaje supervisado y Aprendizaje no supervisado Goodfellow et al. (2016).

- **Aprendizaje supervisado.** En este caso, cada muestra par el entrenamiento tiene los datos de entrada x y una etiqueta l . La idea es que el algoritmo de ML, pueda aprender de estos datos y luego realizar predicción de la etiqueta j tomando como entrada sólo los datos x . Según, Prince (2023) los modelos de aprendizaje supervisado definen una relación entre los datos de entrada y una predicción de salida. Generalmente, podemos en esta área tenemos tratamos problemas de regresión y clasificación. El primero predice un número real mientras que el otro clasifica a un tipo de clase.
- **Aprendizaje no supervisado.** En este caso, solo se cuenta con muestras no etiquetadas. Entonces el algoritmo de ML, debe agrupar los datos en *clusters*.

Un ejemplo de estos problemas es la segmentación de clientes, segmentación de noticias, etc. Adicionalmente, tenemos los modelos generativos, que aprenden a sintetizar nuevos ejemplos de datos que son estadísticamente indistinguibles de los datos de entrenamiento. Algunos modelos generativos describen explícitamente la distribución de probabilidad sobre los datos de entrada, y nuevos ejemplos se generan muestreando de esta distribución. Otros simplemente aprenden un mecanismo para generar nuevos ejemplos sin describir explícitamente su distribución ([Prince, 2023](#)).

- **Aprendizaje por refuerzo.** La última área de aprendizaje automático es el aprendizaje por refuerzo. Este paradigma introduce la idea de un agente que vive en un mundo y puede realizar ciertas acciones en cada paso de tiempo. Las acciones cambian el estado del sistema, pero no necesariamente de manera determinista. Tomar una acción también puede producir recompensas, y el objetivo del aprendizaje por refuerzo es que el agente aprenda a elegir acciones que conduzcan a altas recompensas en promedio ([Prince, 2023](#)).

En la Figura 2.11 disponemos de los tipos de aprendizaje en ML. Pueden dividirse de manera general en aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Las redes neuronales profundas contribuyen a cada una de estas áreas ([Prince \(2023\)](#)).

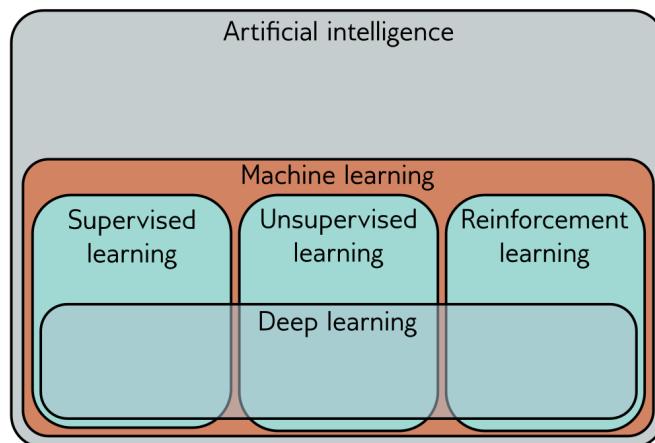


FIGURA 2.11: Tipos de aprendizaje en *Machine Learning*. Fuente: ([Prince, 2023](#))

2.3.2. Redes Neuronales

Uno de los modelos más representativos de ML son las redes neuronales. Estas se basan en unidades llamadas neuronas (perceptrón). En la Figura 2.12, se muestra esta representación, donde x_i representa un atributo, w_i es el peso que se asigna al atributo x_i ,

de esta forma la neurona representa el resultado de multiplicar un peso a un atributo: $\sum_{i=1}^d x_i \cdot w_i$, una representación vectorial sería: $\mathbf{x}^T \mathbf{w}$ (Nielsen, 2015). Luego, a dicho resultado se aplica una función de activación, la función mas utilizada es la función sigmoidea (Ecuación 2.6 y 2.7).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.6)$$

donde $z = \sum_i w_i \cdot x_i - b$.

$$\frac{1}{1 + e^{-\sum_i w_i \cdot x_i - b}} \quad (2.7)$$

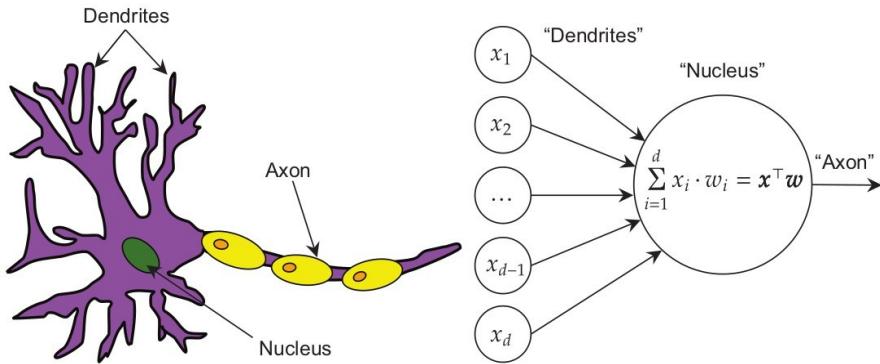


FIGURA 2.12: Representación de una neurona. Fuente: Raff (2022).

El perceptrón, es capaz de solucionar varios problemas, pero para casos complejos puede formar una red, como se presenta en la Figura 2.13.

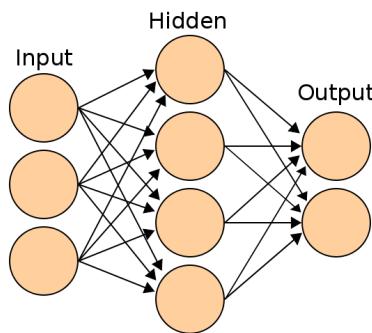


FIGURA 2.13: Representación de una red neuronal.

En resumen, para la red neuronal de la Figura 2.14, una red neuronal con una entrada escalar x , cuatro unidades ocultas h_1, h_2, h_3, h_4 , y una salida 2D $y = [y_1, y_2]^T$, se define con las Ecuaciones 2.8 y 2.9. Además, $a[\bullet]$ es la función de activación, generalmente para las capas ocultas es representada por la función ReLU.

$$\begin{aligned}
 h_1 &= a[\theta_{10} + \theta_{11}x] \\
 h_2 &= a[\theta_{20} + \theta_{21}x] \\
 h_3 &= a[\theta_{30} + \theta_{31}x] \\
 h_4 &= a[\theta_{40} + \theta_{41}x]
 \end{aligned} \tag{2.8}$$

$$\begin{aligned}
 y_1 &= \phi_{10} + \phi_{11}h_1 + \phi_{12}h_2 + \phi_{13}h_3 + \phi_{14}h_4 \\
 y_2 &= \phi_{20} + \phi_{21}h_1 + \phi_{22}h_2 + \phi_{23}h_3 + \phi_{24}h_4
 \end{aligned} \tag{2.9}$$

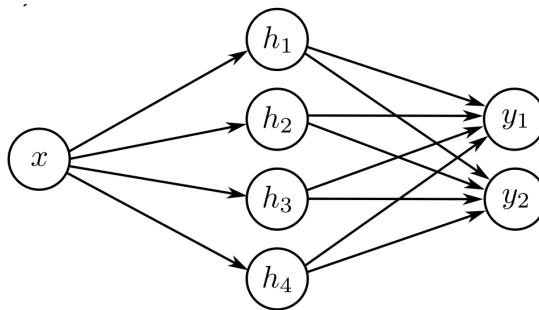


FIGURA 2.14: Red neuronal con una entrada, cuatro unidades ocultas y dos salidas.
Fuente: Prince (2023)

2.4. Deep Learning

Deep learning (DL) es una subcategoría de *Machine Learning*, a diferencia de los algoritmos tradicionales de ML, usualmente DL trata con señales sin pre-procesamiento, los modelos (basados en redes neuronales) son mucho mas complejos tanto en dimensión como en el método de aprendizaje (El Naqa and Murphy, 2022). Por ejemplo, en la Figura 2.15, presentamos la relación entre Inteligencia Artificial (IA), ML y DL, de ahí podemos concluir que ML es parte de la IA y DL es parte de ML (El Naqa and Murphy, 2022).

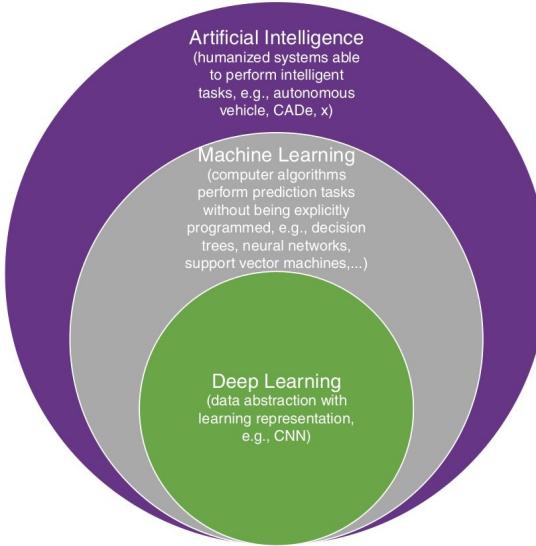


FIGURA 2.15: Relación entre Inteligencia Artificial, *Machine Learning* y *Deep Learning*.

Fuente: [El Naqa and Murphy \(2022\)](#).

Además, a medida que aumenta el número de unidades ocultas, las redes neuronales mejoran su capacidad descriptiva. De hecho, con suficientes unidades ocultas, las redes pueden describir funciones arbitrariamente complejas. Sin embargo, resulta que para algunas funciones, el número necesario de unidades ocultas es prácticamente grande [Prince \(2023\)](#).

2.4.1. Redes Neuronales Profundas

Deep Neural Networks o Redes Neuronales Profundas, son perceptrones multicapa o *multilayer perceptrons*(MLP). Su objetivo es aproximar una función f^* , para el caso de clasificación, podría modelarse como $y = f^*(x)$. Luego, un *feedforward network*, define un mapeo $y = f(x; \theta)$ y aprende los valores de los parámetros θ [Goodfellow et al. \(2016\)](#). Entonces una red neuronal profunda, es una red neuronal tradicional pero con un número grande de neuronas y capas (Figura 2.16).

Por ejemplo una red neuronal como la presentada en la Figura 2.17, se puede representar por la Ecuación 2.10. Donde se describe al vector de unidades ocultas en la capa k como \mathbf{h}_k . El vector de *bias* que contribuye a las capa oculta $k + 1$ es β_k . Los pesos que son aplicados a la k^{th} capa y contribuyen a la capa $(k + 1)^{th}$ es Ω_k . Entonces, una red neuronal profunda $y = f[x, \phi]$ con k capas, se representa como:

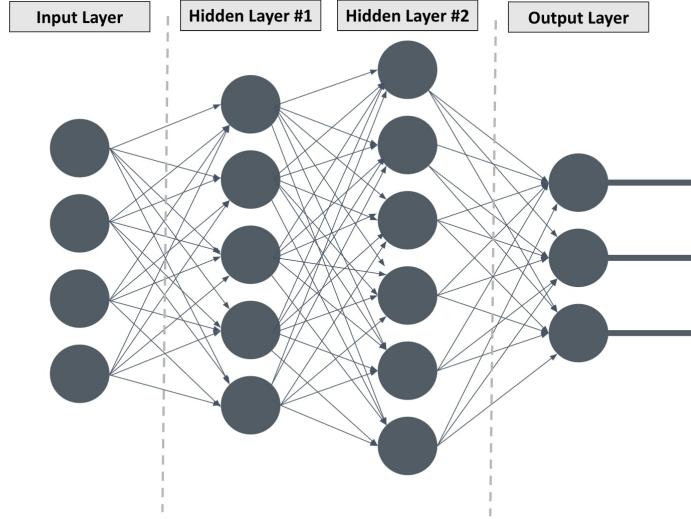


FIGURA 2.16: Representación de un *Deep Feedforward Network*. Fuente: [El Naqa and Murphy \(2022\)](#).

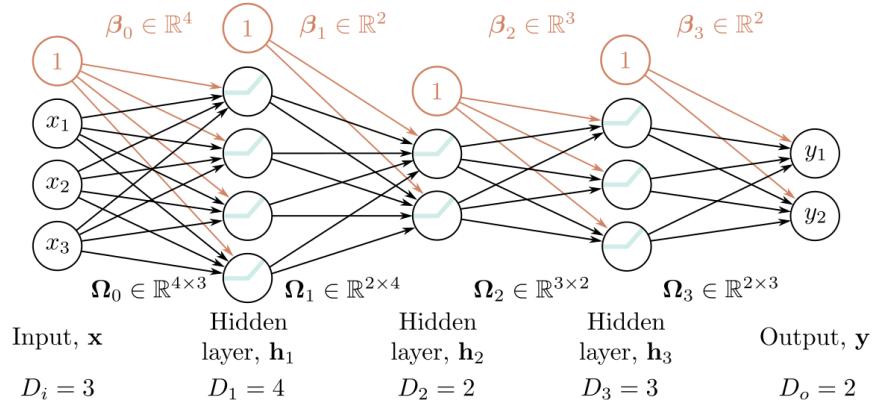


FIGURA 2.17: Notación matricial de una red neuronal profunda. Fuente: [Prince \(2023\)](#).

$$\begin{aligned}
 \mathbf{h}_1 &= \mathbf{a}[\beta_0 + \Omega_0 \mathbf{x}] \\
 \mathbf{h}_2 &= \mathbf{a}[\beta_1 + \Omega_1 \mathbf{h}_1] \\
 \mathbf{h}_3 &= \mathbf{a}[\beta_2 + \Omega_2 \mathbf{h}_2] \\
 &\dots \\
 \mathbf{h}_k &= \mathbf{a}[\beta_{k-1} + \Omega_{k-1} \mathbf{h}_{k-1}] \\
 \mathbf{y} &= \beta_k + \Omega_k \mathbf{h}_k
 \end{aligned} \tag{2.10}$$

2.4.2. Redes Neuronales Convolucionales

Una *Convolutional Neural Networks* (CNN) o Red Neuronal Convolucional, es una red neuronal basada en la operación de convoluciones (utilizada en procesamiento de imágenes). Generalmente estas redes neuronales se aplican a problemas de visión computacional (Zhang et al., 2021). Por ejemplo en la Figura 2.18, se presenta una red neuronal convolucional 1D con un *kernel* de tamaño 3. Cada salida z_i es una suma ponderada de los tres inputs más cercanos x_{i-1}, x_i, x_{i+1} , donde los pesos son $[\omega_1, \omega_2, \omega_3]$. La salida z_2 se calcula con $z_2 = \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3$. La salida $z_3 = \omega_1 x_2 + \omega_2 x_3 + \omega_3 x_4$. Además, podemos variar el tamaño del *kernel*, el *stride* y *dilation* como se ve en la Figura 2.19.

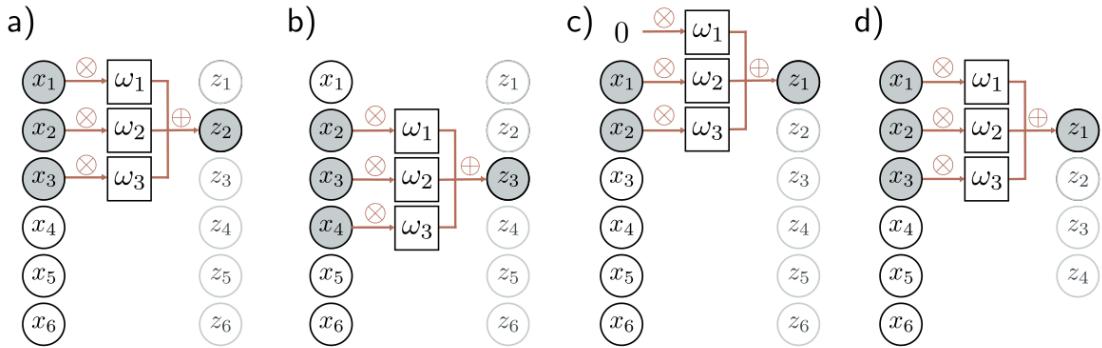


FIGURA 2.18: Red neuronal convolucional 1D con un *kernel* de tamaño 3. Fuente: Prince (2023).

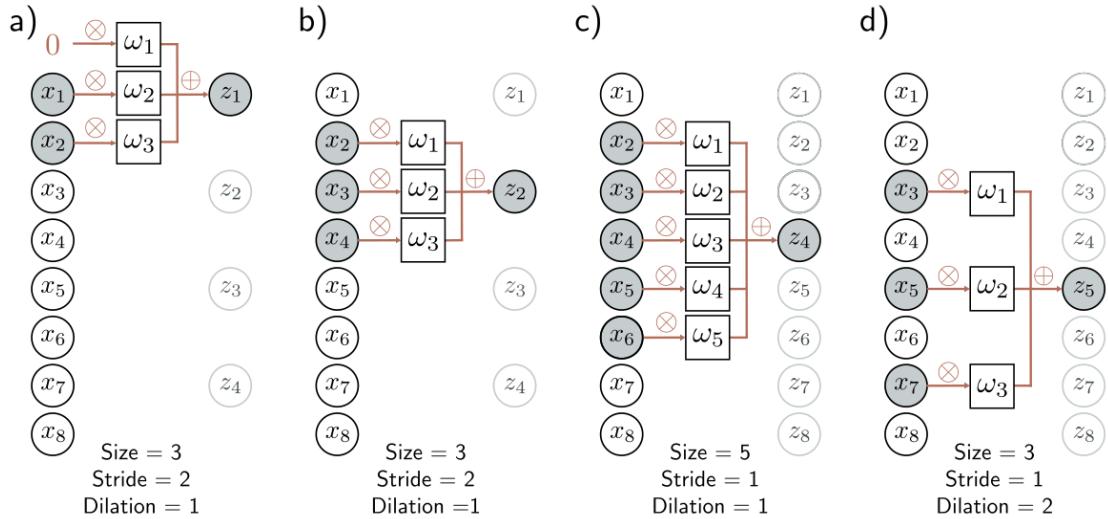


FIGURA 2.19: Ejemplo del uso de *stride*, *kernel size* y *dilation* en la convolución 1D. Fuente: Prince (2023).

Entonces, una capa convolucional calcula su salida convolucionando la entrada, sumando un *bias* β , y pasando cada resultado a través de una función de activación $a[\bullet]$. Con un

tamaño de *kernel* de tres, *stride* y *dilation* igual a 1, la i -ésima unidad oculta h_i se calcularía como:

$$h_i = a[\beta + \omega_1 x_{i-1} + \omega_2 x_i + \omega_3 x_{i+1}]$$

$$h_i = a \left[\beta + \sum_{j=1}^3 \omega_j x_{i-j} \right] \quad (2.11)$$

Adicionalmente, la operación básica es la convolución 2D, se presenta en la Figura 2.20. Se toman pequeñas ventanas de una imagen y se realiza el producto punto con un *kernel* ya establecido. Según los diferentes valores del *kernel*, se pueden obtener diferentes resultados en la imagen de salida como: detección de bordes, suavizados, dilatación, etc. También puede ver la Figura 2.21, para ver el cómputo de una capa convolucional 2D.

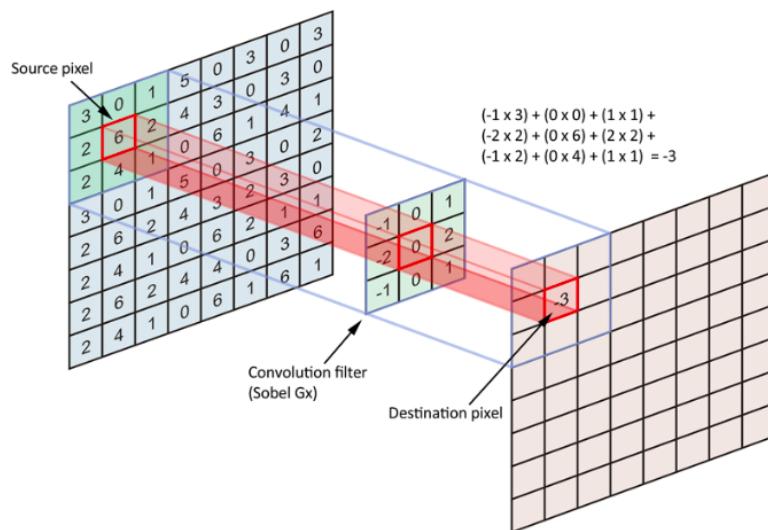
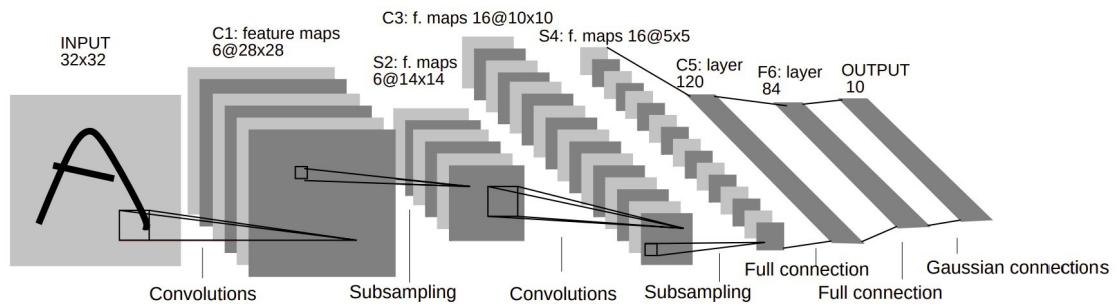
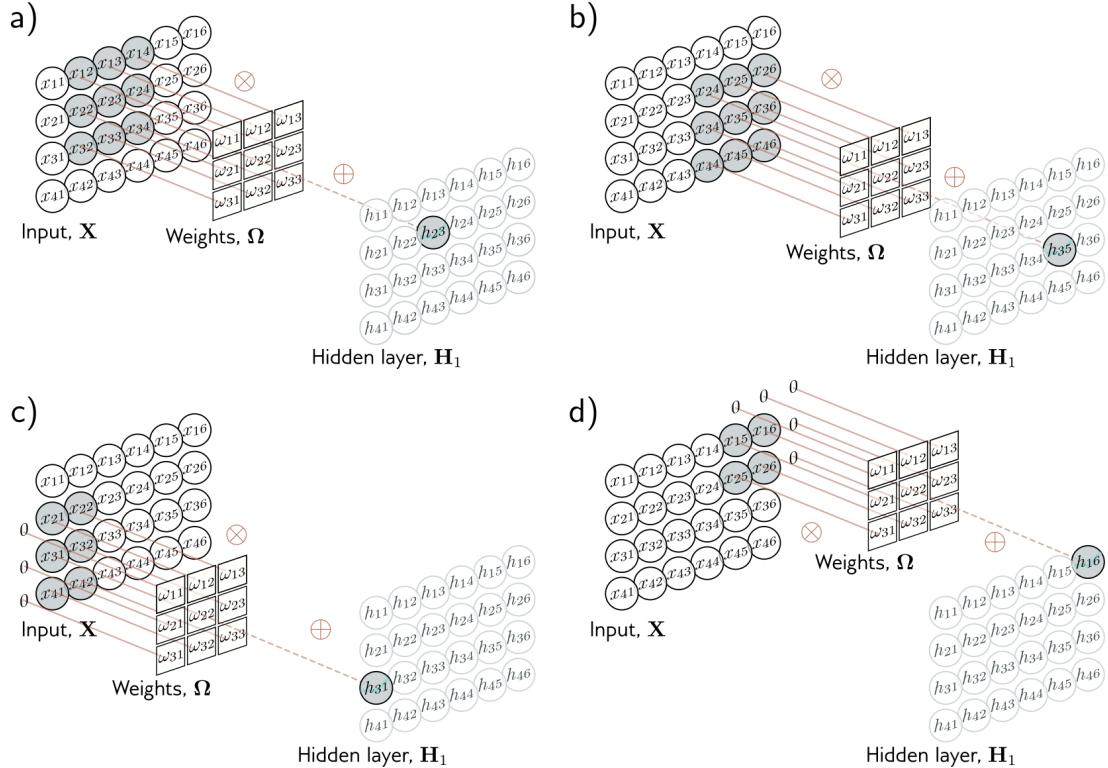


FIGURA 2.20: Ejemplo de una convolución 2D en procesamiento de imágenes. Fuente: [Shuchen \(2022\)](#).

Con inspiración en la operación de convolución, se plantean las CNN por primera vez por LeCun et al. (1998). En la Figura 2.22, se presenta la LeNet-5, planteado por los autores. Luego, surgen diversas propuestas como AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014), GoogleNet (Szegedy et al., 2015) y ResNet (He et al., 2016).



2.4.3. Redes Neuronales Recurrentes

Mientras que las CNN están especializadas para manejar información espacial, las *Recurrent Neural Networks* (RNN), se especializan en información secuencial (Zhang et al., 2021). En este campo, se habla del tiempo como una variable y se tratan problemas de series temporales por ejemplo.

El término RNN, aparece por primera vez en los trabajos de Rumelhart et al. (1985) y Jordan (1997). Algunos autores, comentan también que el inicio de las RNN fue con las redes de Hopfield (Hopfield, 1982). En general estas RNN, tienen dos entradas: estado

actual y estado anterior; luego la RNN predice el siguiente estado. El problema de estas redes neuronales surgen por una falta de memoria, es decir cuando tenemos varios estados, el estado inicial va a influenciar cada vez menos a los estados futuros.

Como alternativa de solución al problema mencionado anteriormente, surge *Long Short-Term Memory* (LSTM), propuesta por [Hochreiter and Schmidhuber \(1997\)](#). Una red neuronal LSTM, es capaz de recordar un dato relevante de una secuencia y almacenarlo varios instantes de tiempo. En la Figura 2.23, explicamos brevemente el funcionamiento de LSTM, los datos que ingresan a una compuerta (*gate*), son los datos de entrada en un tiempo específico y el estado oculto anterior. Luego, es procesado por tres capas totalmente conectadas: *input gate*, *forget gate* y *output gate* ([Zhang et al., 2021](#)).

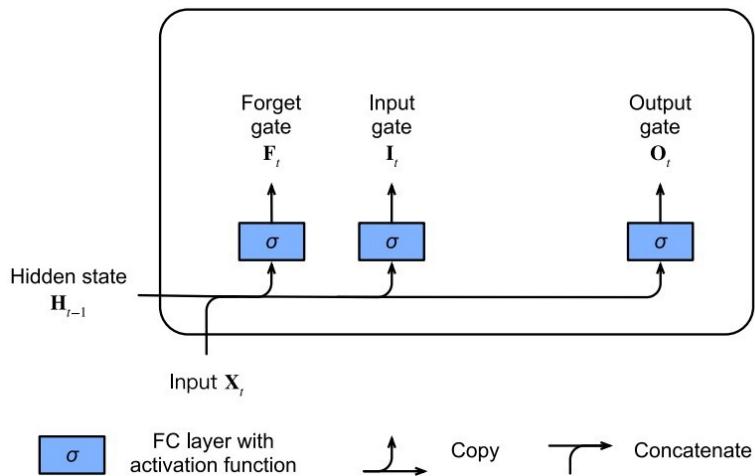


FIGURA 2.23: Ejemplo del procesamiento del *input gate*, *forget gate* y *output gate* de LSTM. Fuente: [Zhang et al. \(2021\)](#).

2.4.4. *Transformers*

El concepto del mecanismo de atención fue introducido inicialmente por Bahdanau en 2014 ([Bahdanau et al., 2014](#)) para abordar las limitaciones asociadas con vectores de codificación de longitud fija. Este enfoque novedoso produjo resultados comparables a los estados del arte en la traducción de inglés a francés. Posteriormente, el mecanismo de atención encontró aplicación en la inferencia de lenguaje natural ([Parikh et al., 2016](#)), lo que llevó a la propuesta de una red de atención estructurada ([Kim et al., 2017](#)). Sin embargo, es importante señalar que estos módulos de atención se utilizaban típicamente en conjunto con redes recurrentes. Ocurrió un cambio significativo en 2017 con la publicación del innovador artículo “*Attention Is All You Need*” propuesta por [Vaswani et al. \(2017\)](#), que presentó una nueva arquitectura de red conocida como *Transformer*. Esta arquitectura se basó exclusivamente en mecanismos de atención y representó una

partida fundamental de los enfoques tradicionales. En 2018, Devlin et al. (2018) introdujo el modelo bidireccional de *Transformer Bidirectional Encoder Representations from Transformers* (BERT). Desde entonces, se ha convertido en uno de los modelos de *Transformer* más reconocidos e influyentes en el campo. El *Transformer* se basa en el concepto de *self-attention*, que se refiere a cuánta atención presta una palabra a otras palabras. Por ejemplo, en la siguiente oración: “El animal no cruzó la calle porque estaba muy cansado”, *self-attention* permite asociar “estaba” con “animal” (Prince, 2023).

2.4.4.1. Self-attention

El bloque principal de un *Transformer*, es la autoatención o *self-attention* $sa[\bullet]$, que toma N entradas x_n , cada una de dimensión $D \times 1$, y devuelve N vectores de salida del mismo tamaño. En el procesamiento del lenguaje natural, cada entrada x_n representa una palabra; mientras que en secuencias de proteínas, representa un aminoácido. Luego, por cada entrada, se calcula un conjunto de valores $v_m = \beta_v + \Omega_v x_m$, donde β_v y Ω_v son los sesgos y pesos, respectivamente. Así, el bloque de *self-attention* se calcula mediante la Ecuación 2.13. Además, $a[x_m, x_n]$ es la atención que la salida x_n presta a x_m y se calcula mediante el producto punto entre k_m^T y q_n . Adicionalmente, se prefiere trabajar con un *self-attention* escalado (Ecuación 2.14); aquí, D_q es la dimensión de q_n y k_n . En la Figura 2.24, se ejemplifica el proceso para calcular *self-attention*, mientras que en la Figura 2.25, se presenta lo mismo, pero forma matricial (Prince, 2023).

$$\begin{aligned} v_m &= \beta_v + \Omega_v x_m \\ q_n &= \beta_q + \Omega_q x_n \\ k_n &= \beta_k + \Omega_k x_m \\ a[x_m, x_n] &= \text{softmax}[k_m^T \cdot q_n] \end{aligned} \tag{2.12}$$

$$Sa[X] = V \cdot \text{softmax}[k^T \cdot q_n] \tag{2.13}$$

$$Sa[X] = V \cdot \text{softmax} \left[\frac{k^T \cdot q_n}{\sqrt{D_q}} \right] \tag{2.14}$$

Además, aplicar varios *multi-head self-attention* logra mejores resultados (Prince, 2023). Entonces, la concatenación de varios *head attentions* se presenta en la Ecuación 2.15.

$$MhSa[X] = \Omega_c[Sa_1[X]; Sa_2[X]; \dots; Sa_H[X];] \tag{2.15}$$

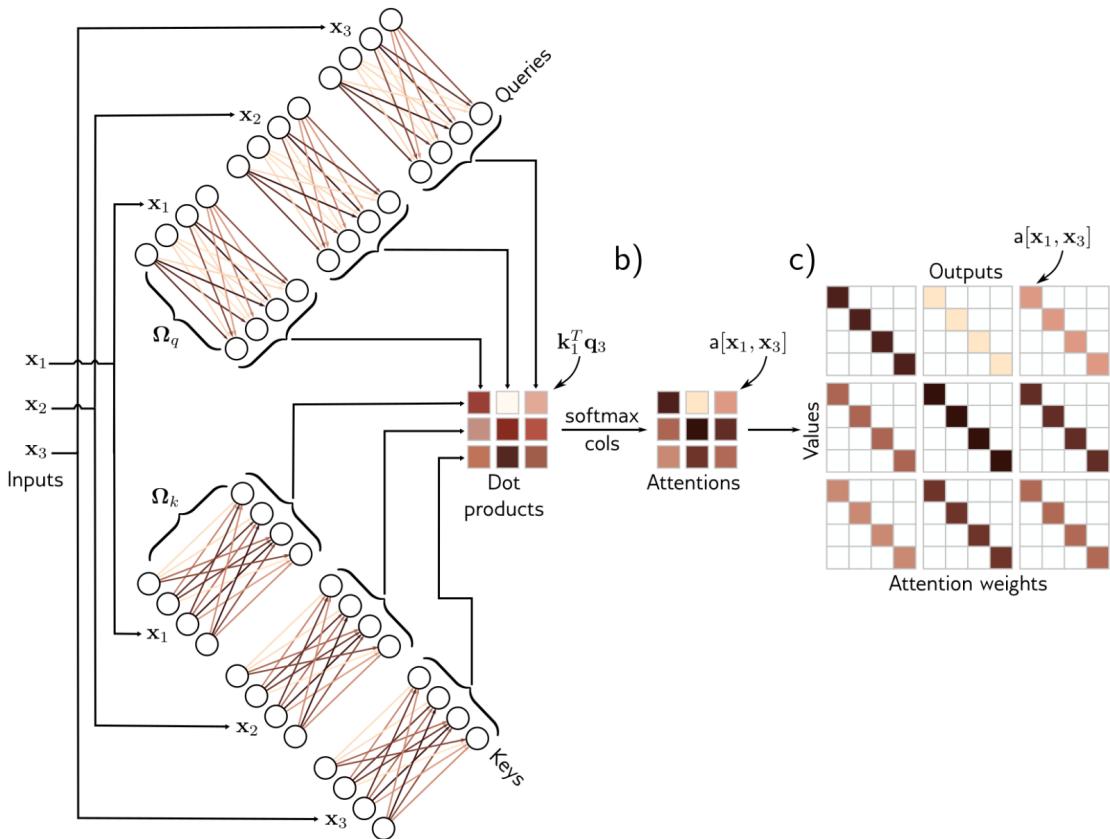


FIGURA 2.24: Ejemplo de como calcular *self-attention*. Fuente: Prince (2023)

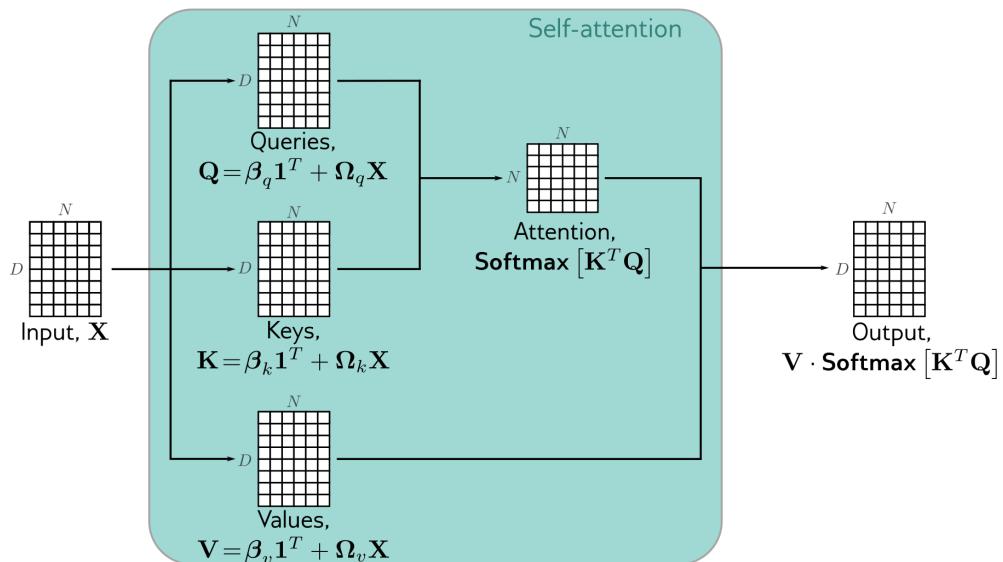


FIGURA 2.25: Proceso para procesar el *self-attention* de forma matricial. Fuente: Prince (2023)

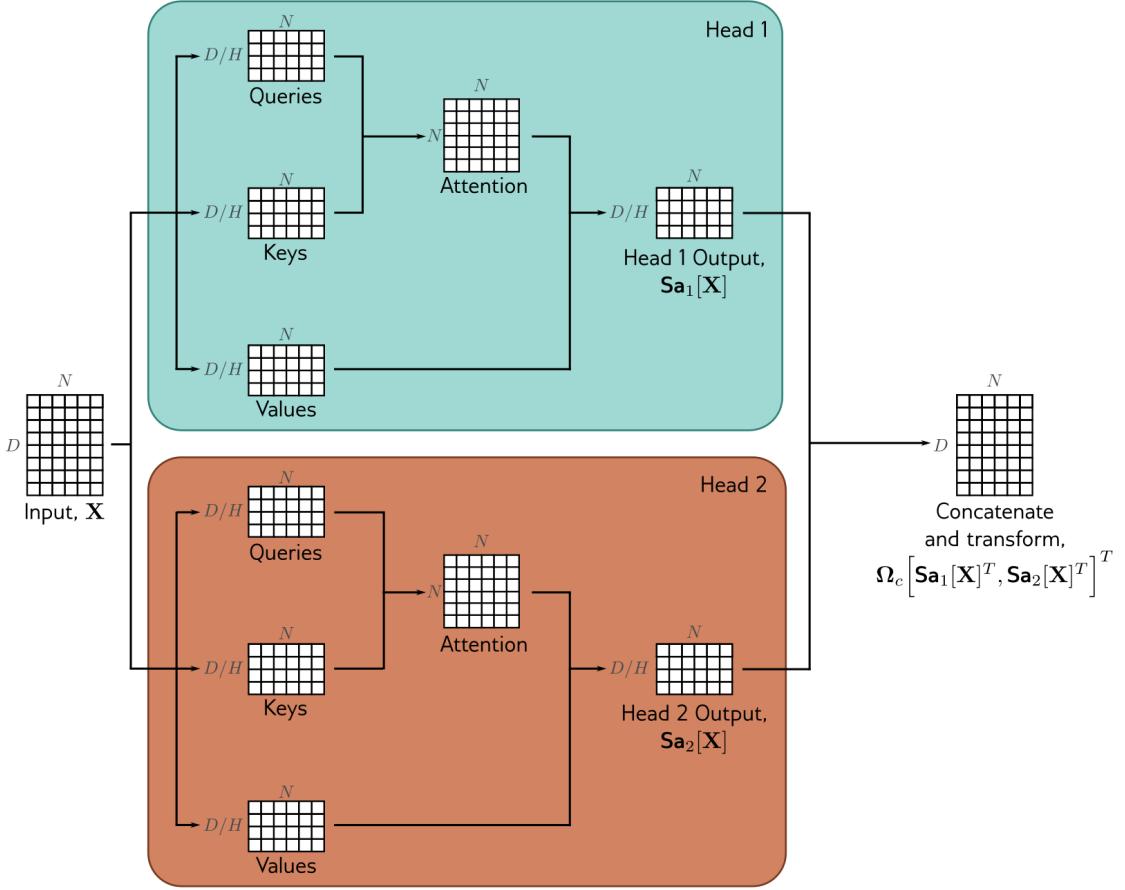


FIGURA 2.26: Proceso para procesar el *multiple head self-attention*. Fuente: Prince (2023)

2.4.4.2. *Mulitple head self-attention*

Normalmente se aplican múltiples mecanismos de *self-attention* en paralelo, y esto se conoce como *multiple head self-attention*. Ahora se calculan H conjuntos diferentes de v , k y q (Prince, 2023):

$$\begin{aligned} \mathbf{V}_m &= \beta_{vh}\mathbf{1}^T + \Omega_{vh}\mathbf{X} \\ \mathbf{Q}_n &= \beta_{qh}\mathbf{1}^T + \Omega_{qh}\mathbf{X} \\ \mathbf{K}_n &= \beta_{kh}\mathbf{1}^T + \Omega_{kh}\mathbf{X} \end{aligned} \quad (2.16)$$

El h^{th} *self-attention* se calcula con:

$$\mathbf{Sa}_h[\mathbf{X}] = \mathbf{V}_h \cdot \text{Softmax} \left[\frac{\mathbf{K}_h^T \mathbf{Q}_h}{\sqrt{\mathbf{D}_q}} \right] \quad (2.17)$$

Un resumen, es presentado en la Figura 2.26.

2.4.5. BERT

Bidirectional Encoder Representations from Transformers (BERT), propuesta por [Devlin et al. \(2018\)](#), está inspirada por la red *Transformer* y su mecanismo de atención, la cuál entiende la relación contextual entre diferentes palabras. A diferencia de una RNN, BERT no tiene dirección, es decir lee la secuencia entera. Esta característica, le permite al modelo aprender información contextual de una palabra con respecto a las otras ([Kelvin, 2022](#)).

BERT es un modelo de codificador que utiliza un vocabulario de 30000 *tokens*. Los *tokens* de entrada se convierten en *word-embedding* de dimensión 1024 y se pasan a través de 24 *Transformers*. Cada uno de ellos contiene un mecanismo de *self-attention* con 16 *heads*. Los *queries*, *keys* y *values* para cada *head* tienen una dimensión de 64. La dimensión de la única capa oculta en la red completamente conectada en el *Transformer* es de 4096. El número total de parámetros es aproximadamente 340 millones. Cuando se introdujo BERT, esto se consideraba grande, pero ahora es mucho más pequeño que los modelos de vanguardia ([Prince, 2023](#)).

2.4.5.1. Pre-entrenamiento

En la etapa de pre-entrenamiento, la red se entrena utilizando *self-supervision*. Esto permite el uso de cantidades enormes de datos sin necesidad de etiquetas manuales. En el caso de BERT, la tarea de *self-supervision* consiste en predecir las palabras faltantes en oraciones tomadas de un gran corpus de internet (Figura 2.27). Durante el entrenamiento, la longitud máxima de entrada es de 512 *tokens* y el tamaño del *batch size* es de 256. El sistema se entrena durante un millón de *steps*, lo que equivale aproximadamente a 50 *epochs* del corpus de 3.3 mil millones de palabras ([Prince, 2023](#)).

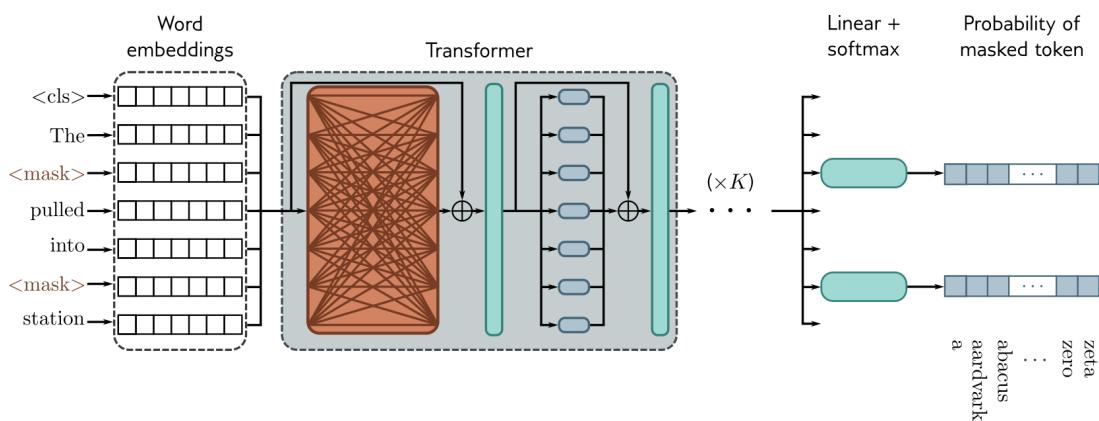


FIGURA 2.27: Pre-entrenamiento de BERT. Fuente: [Prince \(2023\)](#)

2.4.5.2. Fine-tuning

En la etapa de *fine-tuning*, los parámetros del modelo se ajustan para especializar la red en una tarea particular. Se añade una capa adicional a la red del *Transformer* para convertir los vectores de salida al formato de salida deseado (Prince, 2023). Por ejemplo, se puede realizar para la tarea de clasificación de textos (Figura 2.28.a); donde, se coloca un token especial conocido como el token de clasificación o *cls* al comienzo de cada cadena durante el pre-entrenamiento. Para tareas de clasificación de texto como el análisis de sentimientos (en el cual el pasaje se etiqueta como tener un tono emocional positivo o negativo), el vector asociado con el token *cls* se asigna a un número único y se pasa a través de una función sigmoidea (Prince, 2023).

También, se puede realizar nuevas tareas como clasificación de palabras. El objetivo es clasificar cada palabra como un tipo de entidad (por ejemplo, persona, lugar, organización o ninguna entidad), como se describe en la Figura 2.28.b (Prince, 2023).

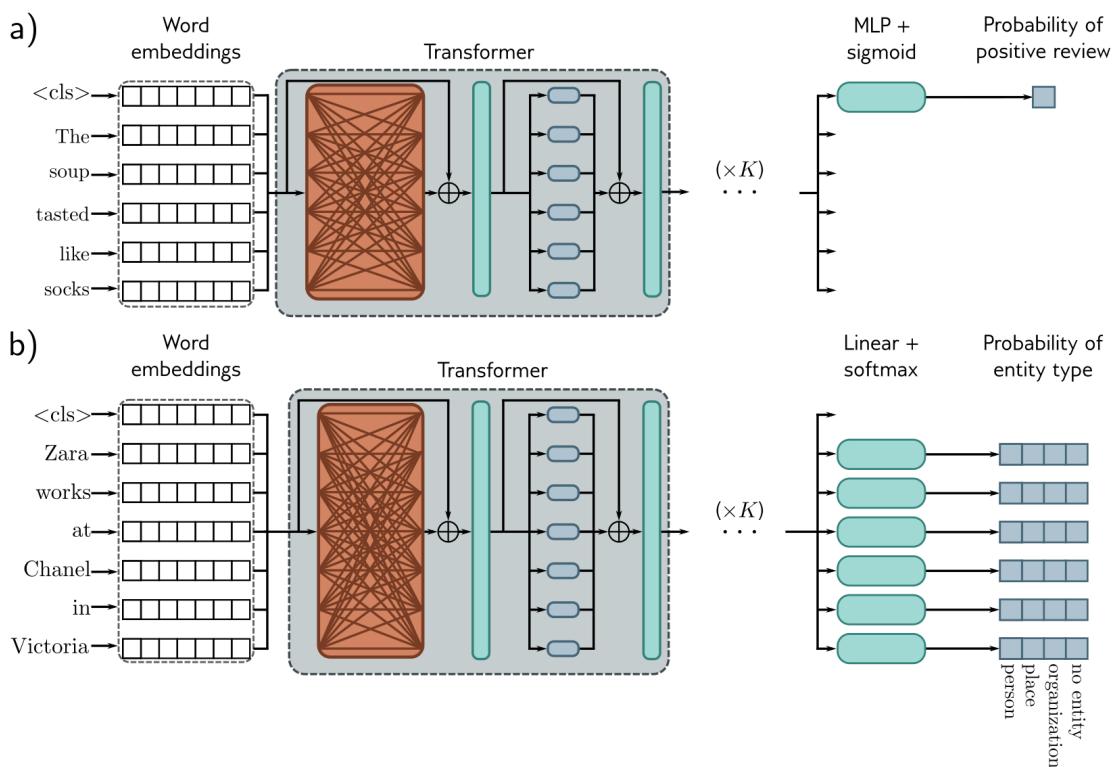


FIGURA 2.28: *Fine-tuning* de BERT, luego del pre-entrenamiento se realiza *fine-tuning* sobre un conjunto de datos etiquetados manualmente para una tarea específica. Fuente: Prince (2023)

El *fine-tuning* de modelos pre-entrenados mediante *self-supervision* ha mejorado significativamente el rendimiento de vanguardia en tareas de procesamiento del lenguaje

natural (Zhang et al., 2020). Sin embargo, a pesar de los éxitos significativos, el *fine-tuning* sigue siendo inestable, especialmente cuando se utiliza la variante grande de BERT en conjuntos de datos pequeños, donde el pre-entrenamiento tiene el potencial de proporcionar el beneficio más significativo. Procesos de aprendizaje idénticos con diferentes *seeds* aleatorios a menudo resultan en modelos significativamente diferentes y a veces degenerados después del *fine-tuning*, incluso cuando solo algunos aspectos aparentemente insignificantes del proceso de aprendizaje se ven afectados por el *seed* aleatorio (Zhang et al., 2020; Prince, 2023).

Capítulo 3

Estado del Arte

En este capítulo presentamos los trabajos relacionados sobre detección de neoantígenos utilizando *Transformers*. Adicionalmente, hemos incluido los *pipelines* desarrollados y pruebas clínicas de aplicación de vacunas personalizadas para demostrar la efectividad e importancia de investigaciones basadas en neoantígenos.

3.1. Metodología

Se ha desarrollado parcialmente las etapas de la metodología PRISMA ([Yepes-Nuñez et al., 2021](#)) para hacer revisiones sistemáticas de la literatura.

El enfoque principal se centra en la priorización de neoantígenos (ver Figura 3.1), ya que esta área ha sido objeto de una cantidad significativa de investigaciones que utilizan *Transformers*. Sin embargo, hemos incluido un análisis de *pipelines* y estudios de ensayos clínicos para obtener información sobre los hallazgos más recientes en cuanto a la aplicación de la detección de neoantígenos en vacunas personalizadas contra el cáncer.

3.1.1. Preguntas de Investigación

Se ha propuesto las siguientes preguntas de investigación:

- **Q1.** ¿Como se aplican los modelos *Transformers* para la detección de neoantígenos?
- **Q2.** ¿Que problemas y limitaciones hacen frente los modelos *Transformers* en la detección de neoantígenos?
- **Q3.** ¿Que *pipelines* se han desarrollado para la detección de neoantígenos?

- **Q4.** ¿Que pruebas clínicas, de vacunas personalizadas de neoantígenos, han sido aplicadas?

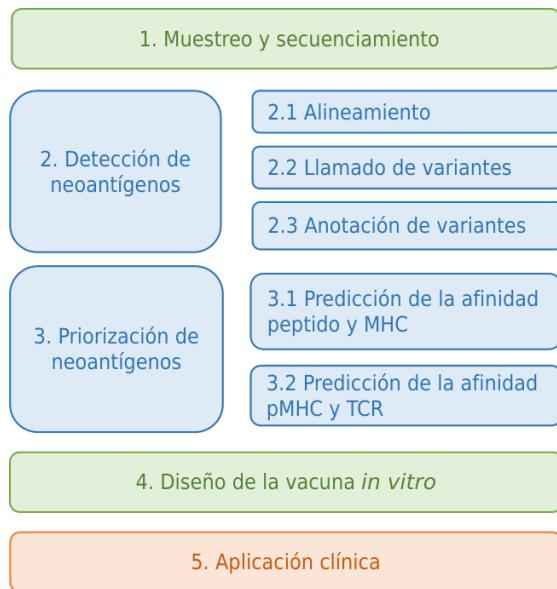


FIGURA 3.1: Una visión general de cada fase del proceso de generación de vacunas personalizadas basadas en neoantígenos.

3.1.2. Criterios de Inclusión y Exclusión

En la Tabla 3.1, presentamos los criterios de inclusión y exclusión considerados para obtener los artículos relacionados a la tesis.

TABLA 3.1: Criterios de inclusión y exclusión.

Criterio de inclusión	Criterio de exclusión
Artículos en revistas Q1 o Q2 según Scimago o Conferencias ERA (A o B).	Artículos de Conferencias de bajo nivel.
Artículos publicados desde el 2018.	Publicaciones que no son del área de computación o Bioinformática.
Artículos que hagan uso de <i>Transformers</i> o <i>Deep Learning</i> con mecanismos de atención.	

3.1.3. Estrategia de Búsqueda

Según las cadenas de búsqueda de la Tabla 3.2 y considerando los criterios de inclusión y exclusión de la Tabla 3.1, se realizo la búsqueda en Scopus, Web of Science, PubMed y Google Scholar. Se realizo un filtro por título, obteniendo un total de 151 artículos.

TABLA 3.2: Cadenas de búsqueda utilizadas para cada fase de detección de neoantígenos.

Categoría	Cadena de búsqueda
Priorización de neoantígenos	(mhc OR hla) AND (peptide OR epitope OR antigen) AND (specificity OR immunogenicity OR binding OR affinity OR predict* OR detection OR presentation OR classification) AND (transformer* OR bert* OR attention OR 'transfer learning' OR method* OR predict*), (tcr OR 't cell' OR t-cell) AND (mhc OR peptide OR epitope OR antigen) AND (specificity OR immunogenicity OR binding OR affinity OR predict* OR detection OR presentation OR classification) AND (transformer* OR bert* OR attention OR 'transfer learning' OR method* OR predict*)
Pipelines	(pipeline OR toolkit) AND (tcr OR 't cell' OR t-cell OR mhc OR hla OR peptide OR epitope OR antigen* OR neoantigen*) (pipeline OR tool* OR workflow OR application OR web*) AND (peptide OR epitope OR antigen* OR neoantigen* OR neoepitope*) AND (immunotherapy OR detection OR identify* OR predict* OR presentation*)
Ensayos clínicos	(neoantigen OR neoepitope OR dendritic cell) AND (vaccines OR immunology)

Luego, se seleccionó un subconjunto en función de los criterios de inclusión, y finalmente, se revisó el *abstract* de cada uno para llegar a un número de 79 artículos.

3.2. Resultados

3.2.1. Consideraciones Iniciales

La detección de neoantígenos es una área multidisciplinaria que involucra a las ciencias de la computación, biología molecular y medicina. Debido a eso, existe una gran cantidad de artículos relacionados a neoantígenos y esto puede presentar sesgos en algunas publicaciones no identificadas. Para disminuir este sesgo, se ha seguido las principales etapas de la metodología PRISMA; se desarrollado cuidadosamente varias cadenas de búsqueda que retornaban la mayoría de publicaciones, se ha incluido todos los sinónimos conocidos y se ha enfatizado en realizar las búsquedas en Scopus y Pubmed, por ser estas donde la mayoría de publicaciones se indexan. Como resultado, se ha obtenido dos publicaciones: “Deep Learning and Transformers in MHC-Peptide Binding and Presentation Towards Personalized Vaccines in Cancer Immunology: A Brief Review” ([Machaca et al., 2023](#)) y “Transformers Meets Neoantigen Detection: A Systematic Literature Review”.

3.2.2. Detección de Neoantígenos

La detección de neoantígenos se basa en la identificación inicial de candidatos, seguida de su posterior priorización. En esta sección, explicaremos el proceso de detección de candidatos a neoantígenos (etapa 2 en la Figura 3.1).

Durante esta etapa, se utilizan datos de secuenciación de ADN (DNA-seq) y de ARN (RNA-seq) para identificar candidatos a neoantígenos. Sin embargo, en este campo, se han adoptado ampliamente varias herramientas bien establecidas, y los *Transformers* no se utilizan regularmente. En primer lugar, se encarga de tomar datos de DNA-seq, RNA-seq y *Mass Spectrometry* (MS) como entrada. Luego, procede a alinear estas secuencias utilizando herramientas como BWA-MEM y Bowtie2. Además, STAR podría ser utilizado porque alinea muestras de tumores de manera más efectiva ([Rubinsteyn et al., 2018](#)). La salida de esta etapa consiste en archivos de alineación BAM. Para la llamada de variantes, se podrían emplear MuTect y Strelka. Posteriormente, la información de ambos métodos se podría combinar, siguiendo el enfoque utilizado por [Zhou et al. \(2021\)](#) y [Rubinsteyn et al. \(2018\)](#). La salida consiste en archivos VCF. A continuación, está la etapa de anotación de variantes, donde se utilizan archivos con formato VCF para derivar péptidos generados a partir de estas variaciones o mutaciones; Isovar y ANNOVAR podrían ser utilizados en esta tarea. Finalmente, para determinar el tipo de HLA del paciente, la herramienta OptiType es una opción. Al final, tenemos varios candidatos a neoantígenos y los tipos de HLA del paciente.

3.2.3. Priorización de Neoantígenos

La priorización de neoantígenos es la tercera etapa en el desarrollo de vacunas contra el cáncer (Figura 3.1). En esta etapa, se toman los candidatos a neoantígenos y se predice su afinidad con el MHC, un problema conocido como predicción del enlace pMHC. Luego, este complejo pMHC se utiliza para predecir la interacción con el TCR. Ambos problemas toman dos secuencias de proteínas como entrada, y el objetivo es predecir su afinidad (regresión) o unión (clasificación).

3.2.3.1. Bases de Datos

Para priorizar neoantígenos, los investigadores a menudo recopilan muestras de diversas fuentes, generalmente extrayendo datos de estudios previos y recursos similares. Sin embargo, existen conjuntos de datos públicos disponibles, como se enumeran en la Tabla 3.3, que se centran específicamente en la interacción entre péptidos y MHC (péptido-MHC) ([Wu et al., 2018](#); [Zhou et al., 2019](#); [Tan et al., 2020](#); [Lu et al., 2022](#)), así como en la

interacción entre pMHC y TCR ([Shugay et al., 2018](#); [Bagaev et al., 2020](#)). Es importante destacar que un estudio reciente proporciona estructuras tridimensionales de péptidos y HLA, lo que introduce una nueva perspectiva de investigación. Finalmente, la *Immune Epitope Database* (IEDB) ([Vita et al., 2018](#)) se destaca como un recurso ejemplar en este campo.

TABLA 3.3: Bases de datos públicas de unión pMHC e interacción pMHC-TCR

Nombre	Referencia	Descripción
VDJdb	Shugay et al. (2018) ; Bagaev et al. (2020)	Base de datos de unión del TCR al pMHC, contiene 5491 muestras.
IEDB	Vita et al. (2018)	Es la base de datos más grande que contiene información de <i>epitopes</i> de células T de humanos y otros organismos.
TSNAdb	Wu et al. (2018)	Involucra 7748 muestras de mutaciones y HLA de 16 tipos de cáncer.
NeoPeptide	Zhou et al. (2019)	Incorpora muestras de neoantígenos resultantes de mutaciones somáticas y elementos relacionados. También contiene 1818137 <i>epitopes</i> de más de 36000 neoantígenos.
pHLA3D	Oliveira et al. (2019)	Presenta 106 estructuras 3D de las cadenas α , $\beta 2M$ y péptidos de las moléculas HLA-I.
dbPepNeo	Tan et al. (2020)	Contiene muestras validadas de la unión del pMHC a partir de MS. Incluye 407794 muestras de baja calidad, 247 de calidad media y 295 de alta calidad.
dbPepNeo2.0	Lu et al. (2022)	Recopila una lista de neoantígenos y moléculas HLA. Presenta 801 HLAs de alta calidad y 842,289 de baja calidad. Además, 55 neoantígenos de clase II y 630 neoantígenos con unión al TCR.
IntroSpect	Zhang et al. (2022a)	Es una herramienta para construir bases de datos sobre la unión pMHC. Utiliza datos de MS.
IPD-IMGT	Robinson et al. (2020)	Tiene 25000 moléculas MHC y 45 <i>alleles</i> .

3.2.3.2. Predicción de la Unión pMHC

Los enfoques para predecir la unión pMHC se pueden clasificar ampliamente en dos categorías: métodos *allele-specific* y métodos *pan-specific*. Los métodos *allele-specific* implican entrenar un modelo distinto para cada *allele* específico, mientras que los métodos *pan-specific* implican el entrenamiento de un modelo universal aplicable a una variedad de *alleles*. Luego, en la Tabla 3.4, presentamos una comparación de modelos *Transformer* y métodos de aprendizaje profundo que utilizan mecanismos de atención.

Dado que trabajamos con entradas de proteínas, cada aminoácido se representa utilizando una fila de la matriz BLOSUM. Algunos estudios han utilizado BLOSUM62 ([Jin et al., 2021](#); [Ye et al., 2021](#); [Zhao et al., 2019](#); [O'Donnell et al., 2018](#)) y BLOSUM50

(Yang et al., 2021; Hu et al., 2019). Además, ciertos autores han utilizado una combinación de codificación *one-hot* y codificación BLOSUM (Liu et al., 2021; Jokinen et al., 2021; Zeng and Gifford, 2019b,a). Alternativamente, se han empleado métodos como el codificador universal de Google (Kubick and Mickael, 2021), AAindex (Kawashima and Kanehisa, 2000; Li et al., 2021) (una base de datos de índices numéricos que representan propiedades fisicoquímicas y bioquímicas de los aminoácidos), coordenadas tridimensionales de aminoácidos (Shi et al., 2020), y la consideración de las propiedades fisicoquímicas de aminoácidos individuales (Moris et al., 2021; Montemurro et al., 2021; Luu et al., 2021). Más recientemente, algunos estudios han incorporado *eluted ligands* de la membrana celular, extraídos mediante datos MS (Zhou et al., 2022; Reynisson et al., 2020a,b; O'Donnell et al., 2020; Alvarez et al., 2019).

Actualmente, NetMHCPan4.1 (Reynisson et al., 2020a) es un método de referencia, este es una red neuronal artificial profunda que consiste en 40 redes neuronales artificiales ensambladas; cabe destacar que maneja eficazmente conjuntos de datos de MS, al igual que el MHCflurry2.0 (O'Donnell et al., 2020).

Existen modelos de *Convolutional Neural Networks* (CNN) que incorporan un mecanismo de atención, como ACME (Hu et al., 2019). ACME utiliza una CNN con un módulo de atención que asigna pesos a posiciones de residuos individuales, con el objetivo de asignar mayores pesos a los residuos de mayor importancia en las interacciones pMHC. ACME logró un Coeficiente de Correlación de Rango de Spearman (SRCC) de 0.569, lo cual es superior a NetMHCPan 4.0. A continuación, tenemos MHCAttNet (Venkatesh et al., 2020), que utiliza una CNN seguida de una capa de atención. La capa de atención se utiliza para generar un mapa de calor sobre los aminoácidos, indicando las subsecuencias importantes presentes en la secuencia de aminoácidos. Otro modelo basado en CNN es DeepAttentionPan (Jin et al., 2021), que utiliza una CNN profunda para codificar péptidos y MHC en vectores de dimensiones $40 \times 10 \times 11$ antes de emplear un módulo de atención para calcular pesos posicionales. También contamos con DeepNetBim (Yang et al., 2021), que incorpora un módulo de atención similar a ACME y DeepAttentionPan. Sin embargo, utiliza dos CNN separadas para predecir la unión pMHC y la immunogenicidad, que luego se combinan en las capas finales. Además, en su estudio sobre SpConvM (Chen et al., 2021c), los autores demostraron que la incorporación de núcleos globales en CNN con atención produjo un mejor rendimiento. Además, sus experimentos incluyeron una comparación de diferentes métodos de codificación de aminoácidos, incluyendo *one-hot*, BLOSUM y Deep. Según sus hallazgos, la combinación de *one-hot*, BLOSUM y Deep juntos dio como resultado mejores resultados. Recientemente, ha surgido el uso de *Capsule Neural Network* (CapsNet) para modelar relaciones jerárquicas. CapsNet-MHC (Kalemati et al., 2023) se propone para predecir la unión pMHC-I, y superó a otras

herramientas como HLAB, ACME, Anthem y NetMHCpan4.1 para péptidos pequeños de 8 a 11 *mers*.

Además, se han introducido varias *Recurrent Neural Networks* (RNN), como DeepHLApan (Wu et al., 2019), que es un modelo *allele-specific* que considera datos de unión pMHC e inmunogenicidad. El modelo presenta tres capas de *Bidirectional Gated Recurrent Unit* (BiGRU) y una capa de atención, produciendo finalmente las predicciones de unión e inmunogenicidad. Además, este enfoque incorporó epitopos de células T CD8+ y datos de MS; logró una precisión que supera 0.9 para 43 alelos HLA. Además, el modelo *allele-specific* DeepSeqPanII (Liu et al., 2021) utilizó una combinación de codificación BLOSUM62 y *one-hot*, con un enfoque específico en MHC-II. El modelo incluyó dos capas de *Long Short-Term Memory* (LSTM) con 100 unidades y un bloque de atención para extraer información ponderada. El bloque de atención consistía en cuatro capas de convolución 1-D, y se emplearon tres capas completamente conectadas para predecir la afinidad. DeepSeqPanII superó a NetMHCIIPan 3.2 para 26 de los 54 *alleles*. Otra RNN es MATHLA (Ye et al., 2021), que utilizó una BiLSTM para aprender las dependencias entre los residuos de aminoácidos y aplicó *multi-head self-attention* para obtener información posicional para la salida de BiLSTM. La salida se procesó aún más a través de capas convolucionales 2-D. MATHLA logró un AUC de 0.964, superando el rendimiento de NetMHCpan 4.0, MHCflurry y ACME, que obtuvieron puntajes de 0.945, 0.925 y 0.905, respectivamente. Recientemente, el modelo *allele-specific* DapNet-HLA (Jing et al., 2023) introdujo un conjunto de datos adicional de Swiss-Prot para muestras negativas. El método utilizó un método de *embedding* para cada token y su posición absoluta, que se comparó con varias técnicas de codificación, incluyendo la *Desviación de Dipeptide Deviation from Expected mean* (DDE), *Amino Acid Composition* (AAC), *Dipeptide Composition* (DPC), y *Encoding based on Grouped Weight* (EGBW). Recientemente, DapNet-HLA combinó las ventajas de CNN, SENet (para agrupamiento) y LSTM, logrando buenos resultados, aunque no se comparó directamente con métodos de vanguardia.

BERTMHC (Cheng et al., 2021) fue uno de los trabajos pioneros en incorporar la arquitectura BERT. Este predictor *pan-specific* de unión/presentación de pMHC-II utilizó el aprendizaje por transferencia de *Tasks Assessing Protein Embeddings* (TAPE) (Rao et al., 2019), un modelo entrenado con datos de la base de datos Pfam que comprende treinta y un millones de proteínas. Los autores integraron TAPE seguido de una capa *Fully Connected* (FC). En experimentos, BERTMHC superó a NetMHCIIPan3.2 y PUFFIN, logrando un AUC de 0.8822 en comparación con 0.8774. Del mismo modo, ImmunoBERT (Gasser et al., 2021) aprovechó el aprendizaje por transferencia de TAPE, centrándose en la predicción de pMHC-I. El modelo también utilizó capas FC después del modelo TAPE. El análisis de los autores concluyó que los aminoácidos en proximidad

a los extremos N/C del péptido son de alta relevancia, según análisis de LIME y SHAP. Además, CapTransformer (Chen et al., 2021a) introdujo un innovador mecanismo de *cross self-attention* que alinea y agrega eficazmente las características de los residuos de pMHC de manera conjunta. Al utilizar tanto la *self-attention* como *cross self-attention*, facilita el aprendizaje de representaciones de características para los residuos individuales y la información global de unión pMHC, lo que resulta en un rendimiento superior en comparación con NetMHCpan4.0.

Otros métodos que utilizaron el aprendizaje por transferencia incluyen MHCRoBERTa (Wang et al., 2022a) y HLAB (Zhang et al., 2022b). El primero empleó cinco *encoders* con doce *multi-head self-attention*. Inicialmente, el enfoque utilizó un entrenamiento auto-supervisado con datos de las bases de datos UniProtKB y Swiss-Prot. El método también aplicó la Tokenización de *subtokens* y superó a NetMHCpan4.0 y MHCflurry2.0, logrando un coeficiente de correlación de Spearman Rank (SRCC) de 0.543. HLAB aprovechó el aprendizaje por transferencia de ProtBert-BFD (Elnaggar et al., 2021), que fue entrenado con datos del conjunto de datos BFD que contiene 2,122 millones de proteínas. HLAB empleó un modelo BiLSTM al final de ProtBert-BFD y logró un rendimiento superior a NetMHCpan4.1. Además, una investigación adicional examinó la aplicación del aprendizaje por transferencia y la aplicación de *padding* (Arceda, 2023). Finalmente, TransPHLA (Chu et al., 2022) aplica la *self-attention* a los péptidos, TransPHLA superó a NetMHCpan4.1, y ofrece la ventaja de ser efectivo para péptidos y *alleles* de MHC de diferentes tamaños.

Una propuesta interesante implica el uso del modelo *Star-Transformer*, SMHCpan (Ye et al., 2023), un modelo liviano en el que la estructura de FC se reemplaza por una topología en forma de estrella. Además, las *Graph Neural Networks* (GNN) se han utilizado en varios problemas de *Protein-Protein Interaction* (PPI) debido a que gestionan las relaciones entre proteínas. En este contexto, surgió una propuesta novedosa, ESM-GAT (Hashemi et al., 2023), que utilizó arquitecturas BERT y aprendizaje por transferencia de los modelos ESM1b y ESM2, luego apiló una *Graph Attention Network* (GAT). Superó a NetMHCpan4.1; sin embargo, los autores no compararon la propuesta con otras herramientas del estado del arte.

3.2.3.3. Predicción de la Unión pMHC-TCR

Los modelos, en general, tienen como entrada secuencias de TCR y pMHC. La representación de estos datos y la extracción de características son esenciales para obtener mejores resultados. Muchos modelos utilizan la matriz BLOSUM, mientras que algunos modelos emplean la codificación *one-hot*, y otros combinan BLOSUM y la codificación

one-hot. Sin embargo, algunos optan por utilizar *Granularity vectors* ([Xu et al., 2022](#)). En la Tabla 3.5, presentamos un resumen de los métodos basados en *Transformers* y DL con mecanismos de atención para la predicción de la unión pMHC-TCR.

Las CNN con mecanismos de atención se emplean porque capturan y procesan eficazmente los datos. MIX-TPI ([Yang et al., 2023](#)), un marco computacional multimodal, utiliza CNN con *self-attention*. También, se utilizan para construir *sequence-based extractors* (SE) y *physicochemical-based extractors* (PE), que se encargan de aprender características refinadas de secuencias y características fisicoquímicas, respectivamente. Finalmente, *self-attention* se emplea para fusionar estas representaciones y predecir la unión pMHC-TCR.

Por otro lado, las RNN realizan tareas que involucran datos secuenciales de aminoácidos de TCR y pMHC, y existen varios estudios de RNN con mecanismos de atención, como DLpTCR ([Xu et al., 2021](#)) y AttnTAP ([Xu et al., 2022](#)). DLpTCR es un conjunto de tres arquitecturas de aprendizaje profundo que incluyen *Fully Connected Layers* (FCN), LeNet-5 y ResNet-20 para predecir la probabilidad de interacción péptido-TCR y utiliza un mecanismo de atención en ResNet-20 para mejorar la calidad de las salidas generadas. AttnTAP es un marco de aprendizaje profundo de doble entrada que incluye una BiLSTM y una perceptrón multicapa (MLP), y un mecanismo de atención para extraer características de TCR y péptidos por separado y realizar la predicción de la unión péptido-TCR. Algunos estudios utilizaron CNN, RNN y mecanismos de atención, como TcellMatch ([Fischer et al., 2020](#)). TcellMatch es un conjunto de modelos de aprendizaje profundo con múltiples arquitecturas y utiliza una tecnología llamada de *single-cell*, que permite la secuenciación simultánea de las cadenas TCR α y β y la determinación de la especificidad de las células T. Este estudio también realiza múltiples comparaciones y demuestra que los modelos que incluyen tanto la cadena α como la cadena β tienen una ventaja predictiva sobre los modelos que solo incluyen la cadena β , aunque la diferencia es pequeña pero significativa.

En la actualidad, se han desarrollado una variedad de modelos basados en *Transformers*. ATM-TCR ([Cai et al., 2022](#)) introdujo el uso de mecanismos de atención como parte principal y presentó un modelo que utiliza una red de *multi-head self-attention*. Este modelo consta de dos *encoders* para secuencias de TCR y *epitopes* y un *decoder* lineal para determinar la unión. Utiliza una red de *multi-head self-attention* para obtener representaciones contextuales de cada secuencia. Cada una de las secuencias de TCR y *epitopes* se alinea a través de IMGT y se utiliza la distancia euclídea para calcular si hay unión o no. Luego, AVIB ([Grazioli et al., 2022](#)), una generalización de múltiples secuencias del *Variational Information Bottleneck*, introdujo un novedoso método llamado *Attention of Experts* (AoE). AoE puede aprovechar los abundantes datos disponibles

cuando falta la secuencia de la cadena CDR3 α o CDR3 β . Tambien surge BERTrand (Myronov et al., 2023), la arquitectura, BERT, se pre-entreno y realizo *fine-tuning* para la predicción de la unión péptido-TCR. Inicialmente, los péptidos y TCR se representan como secuencias de tokens, y cada péptido y TCR se concatenan en una secuencia unificada que conserva su información posicional y tipológica. Otro modelo es *Hybrid gMLP* (Zhang et al., 2023a) que combinó el modelo gMLP con el mecanismo de atención, donde la información se obtuvo a través de gMLP, y luego se utilizaron *multi-head self-attention* y *local attention* para extraer la información de correlación del CDR3 pMHC-TCR. Este marco puede manejar los problemas causados por diferentes longitudes de TCR. Además, demostró que los modelos entrenados con datos emparejados de cadenas CDR3- α y CDR3- β son mejores que aquellos entrenados solo con datos de CDR3- α o CDR3- β . Este modelo puede tener potencial, pero le falta una base de datos grande para introducir su modelo poderoso. Finalmente, TCRdock (Bradley, 2023) es una versión especializada del predictor de redes neuronales *AlphaFold*, que demuestra que el predictor *AlphaFold* se puede utilizar para distinguir con precisión los *epitopes* de péptidos correctos de los incorrectos.

Otros métodos que utilizaron el aprendizaje por transferencia incluyen diffRBM (Bravi et al., 2023) y pMTattn (Shang et al., 2022). DiffRBM es un enfoque basado en secuencias que utiliza aprendizaje por transferencia y *Restricted Boltzmann Machines* RBM. Las RBM se utilizan principalmente en la representación y generación de datos. Este enfoque depende de dos conjuntos de datos diferentes genéricamente denominados conjuntos de datos “selected” y “background” de gran tamaño. Ambos conjuntos se entranan con el modelo de RBM. pMTattn fue uno de los primeros modelos en adoptar un mecanismo de *cross-attention*, lo que permite centrarse en los sitios de unión importantes.

3.2.4. Pipelines

La detección *in silico* de neoantígenos se basa en las etapas segunda y tercera representadas en la Figura 3.1. En este contexto, debido a la complejidad del proceso y a la variedad de métodos disponibles, se han desarrollado herramientas de software y *pipelines* para agilizar el uso de estas herramientas. En la Tabla 3.6, presentamos los *pipelines* publicados desde 2018. Estos utilizan diversos tipos de información como entrada. Por ejemplo, PGV Pipeline (Rubinsteyn et al., 2018) y PEPPRMINT (Zhou et al., 2023) utilizan secuenciación de ADN, mientras que otras herramientas como PGNneo (Tan et al., 2023), NAP-CNB (Wert-Carvajal et al., 2021), NaoANT-HILL (Coelho et al., 2020), ProGeo-neo (Li et al., 2020), ScanNeo (Wang et al., 2019), y Neopepse (Kim et al., 2018) utilizan RNA-seq porque estas secuencias capturan mejor la información sobre mutaciones y regiones no codificantes del ADN (Tan et al., 2023).

Para reducir la complejidad de los *pipelines*, algunas propuestas han optado por utilizar el *Variant Calling Files* (VCF) como entrada. Estos archivos contienen información sobre mutaciones y se obtienen mediante métodos de alineamiento y llamada de variantes (etapas 2.1 y 2.2 en la Figura 3.1). Herramientas como HLA3D (Li et al., 2022), Neoepiscope (Wood et al., 2020), pVACtools (Hundal et al., 2020) y NeoPredPipe (Schenck et al., 2019) reducen así el número de herramientas utilizadas en la detección de neoantígenos. Sin embargo, los resultados obtenidos pueden ser inferiores en comparación con las herramientas que utilizan secuenciación de ADN y ARN.

Además, para una detección precisa de neoantígenos, es necesario contar con la secuenciación de las proteínas del MHC. Estas proteínas son esenciales porque se utilizan para predecir la unión entre neoantígenos potenciales y MHC (pMHC: etapa 3.1 en la Figura 3.1). Dado que estas proteínas son codificadas por genes altamente polimórficos, se produce una variación sustancial en la unión de péptidos (neoantígenos), lo que influye en el conjunto de péptidos presentados a las células T (Abualrous et al., 2021). En este contexto, los *pipelines* NeoPredPipe (Schenck et al., 2019) y Neopepsee (Kim et al., 2018) solicitan estas proteínas MHC como entrada, mientras que otros predicen esta información a partir de la secuenciación de ADN. Desde el punto de vista de la facilidad de uso, obtener los tipos de MHC implica un esfuerzo adicional para el usuario.

3.2.5. Ensayos Clínicos

En todos los ensayos clínicos revisados que se han concluido hasta la fecha (Tabla 3.7), se ha observado un punto común: la terapia celular adoptiva es segura, con efectos secundarios manejables, y genera una respuesta inmunológica lo suficientemente efectiva como para ayudar en la lucha contra diferentes tipos de cáncer. También contribuye positivamente a otros tipos de tratamientos, especialmente el uso de inhibidores de puntos de control (Awad et al., 2022; Ott et al., 2020; Holm et al., 2022; Rocconi et al., 2022; Poran et al., 2020; Bassani-Sternberg et al., 2019). Esto se refleja en el aumento del tiempo libre de enfermedad o la tasa de supervivencia general de los pacientes que participaron en estos estudios en comparación con los tratamientos convencionales. Esto, por supuesto, depende del tipo de neoplasia, la etapa en la que se encuentra y la naturaleza más o menos agresiva de cada neoplasia. Es particularmente relevante el resultado de un ensayo clínico aleatorio que muestra que las vacunas personalizadas basadas en células dendríticas cargadas *in vivo* con neoantígenos tumorales demostraron generar respuestas inmunológicas más fuertes con menos efectos secundarios que otros tipos de terapia celular adoptiva, específicamente vacunas de células tumorales expuestas a antígenos (Dillman et al., 2018).

En todos los ensayos clínicos, se administraron vacunas personalizadas basadas en neoantígenos a pacientes con tumores sólidos, y en la mayoría de los estudios, a pacientes en etapas avanzadas de la enfermedad (Cheng et al., 2021; Cafri et al., 2020; Awad et al., 2022; Ott et al., 2020; Palmer et al., 2022; Yu et al., 2023; Holm et al., 2022; Mueller et al., 2022; Ellingsen et al., 2022; Shou et al., 2022; Aggarwal et al., 2019; Poran et al., 2020; Dillman et al., 2018), considerando una etapa avanzada aquella en la que hay metástasis o extensión de la enfermedad. Esto no excluye la posibilidad de que este tipo de tratamiento también pueda aplicarse a los cánceres hematológicos.

Lamentablemente, solo dos de los estudios revisados fueron aleatorizados (Rocconi et al., 2022; Dillman et al., 2018). Todos los estudios son de intervención, lo que demuestra la intención de los investigadores de establecer los parámetros de seguridad, eficacia o ambos para las vacunas personalizadas basadas en neoantígenos, ya sea como terapia individual o en combinación con otros tipos de tratamientos, como se discutió anteriormente.

Cerca de la mitad de los estudios son ensayos de fase II (Cheng et al., 2021; Cafri et al., 2020; Cai et al., 2021; Yu et al., 2023; Holm et al., 2022; Mueller et al., 2022; Ellingsen et al., 2022; Aggarwal et al., 2019; Kloor et al., 2020; Podaza et al., 2020; Sater et al., 2020; Dillman et al., 2018), documentando la eficacia de las vacunas personalizadas basadas en neoantígenos en el tratamiento de diversos tipos de cáncer. Es interesante destacar que los neoantígenos podrían tener otras aplicaciones no solo terapéuticas contra el cáncer, sino también como predictores de la respuesta a tratamientos de inmunoterapia como los inhibidores de puntos de control, que podrían definirse mediante el análisis de su interacción con los linfocitos T CD8 (Holm et al., 2022).

TABLA 3.4: Métodos basados en *Transformers* y DL con mecanismos de atención utilizados para la predicción de la unión pMHC.

Referencia	Nombre	Entrada	Modelo
Hashemi et al. (2023)	ESM-GAT	<i>One-hot</i>	BERT con transferencia de aprendizaje de ESM1b y ESM2 <i>fine-tuned</i> con una <i>Graph Attention Network</i> (GAT) al final. Superó a NetMHCPan4.1.
Kalemati et al. (2023)	CapsNet-MHC	BLOSUM62	<i>Capsule Neural Network</i> , superó a las herramientas de vanguardia para péptidos pequeños de 8 a 11 mer.
Ye et al. (2023)	STMHCPan	<i>One-hot</i>	Un modelo <i>Star-Transformer</i> , útil para péptidos de cualquier longitud y ampliable para predecir respuestas de células T.
Jing et al. (2023)	DapNet-HLA	<i>Fused word embedding</i>	Combina las ventajas de CNN, SENet (para agrupación) y LSTM con atención.
Zhang et al. (2022b)	HLAB	<i>One-hot</i>	BERT del modelo pre-entrenado ProtBert seguido de una BiLSTM.
Wang et al. (2022a)	MHC RoBERTa	<i>One-hot</i>	RoBERTa pre-entrenado seguido de 12 <i>multi-head self-attention</i> y capas totalmente conectadas, superó a NetMHCPan3.0.
Chu et al. (2022)	TransPHLA	<i>One-hot</i>	Utiliza un mecanismo de self-attention basado en cuatro bloques, superó ligeramente a NetMHCPan4.1 y es más rápido en hacer predicciones.
Chen et al. (2021a)	CapTransformer	<i>One-hot</i>	<i>Transformer</i> con <i>cross self-attention</i> para capturar información local y global.
Gasser et al. (2021)	ImmunoBERT	<i>One-hot</i>	BERT de TAPE pre-entrenado seguido de una capa lineal. Los autores afirman que los terminales N y C son altamente relevantes después de un análisis con SHAP y LIME.
Cheng et al. (2021)	BERTMHC	<i>One-hot</i>	BERT de TAPE pre-entrenado seguido de una capa lineal. Superó a NetMHCIIpan3.2 y PUF-FIN.
Ye et al. (2021)	MATHLA	BLOSUM	Integra BiLSTM con <i>multi-head self-attention</i> . Obtuvo una puntuación AUC de 0.964, en comparación con 0.945, 0.925 y 0.905 para NetMHCPan 4.0, MHCflurry y ACME respectivamente.
Liu et al. (2021)	DeepSeqPanII	BLOSUM62 y <i>one-hot</i>	Tiene dos capas LSTM, un bloque de atención y tres capas totalmente conectadas. Obtuvo mejores resultados que NetMHCIIpan 3.2 en 26 de 54 alelos.
Yang et al. (2021)	DeepNetBim	BLOSUM50	Utiliza CNN separadas para la predicción de la unión pMHC y <i>immunogenetic</i> con un módulo de atención. Obtuvo 0.015 MAE para la unión y 94.7 de precisión para la inmunogenicidad.
Jin et al. (2021)	DeepAttention Pan	BLOSUM62	CNN con un mecanismo de atención. Es <i>allele-specific</i> y obtuvo resultados ligeramente mejores que ACME a nivel de <i>alleles</i> .
Chen et al. (2021c)	SpConvM	<i>One-hot</i> , BLOSUM y Deep	Capa 1D de CNN, una capa de atención y una capa totalmente conectada. Además, emplearon <i>global kernels</i> para mejorar sus resultados, junto con una combinación de <i>one-hot</i> , BLOSUM y Deep.
Venkatesh et al. (2020)	MHCAttNet	<i>One-hot</i>	CNN seguido de una capa de atención para generar un mapa de calor sobre los aminoácidos.
Hu et al. (2019)	ACME	BLOSUM50	CNN con atención, extrae patrones interpretables sobre la unión pMHC. Además, obtuvo un SRCC de 0.569, un AUC de 0.9 para HLA-A y 0.88 para HLA-B.
Wu et al. (2019)	DeepHLApan	<i>One-hot</i>	Modelo <i>allele-specific</i> con tres capas de GRU Bidireccional (BiGRU) con una capa de atención. Obtuvo una precisión > 0,9 en 43 <i>alleles</i> HLA.

TABLA 3.5: Métodos basados en *Transformers* y DL con mecanismos de atención utilizados para la predicción de la unión pMHC-TCR.

Referencia	Nombre	Entrada	Modelo
Bravi et al. (2023)	diffRBM	BLOSUM62	Emplea una arquitectura RBM, aprendizaje por transferencia y <i>Restricted Boltzmann Machines</i> .
Grazioli et al. (2022)	AVIB	BLOSUM50	Utiliza <i>Attention of Experts</i> (AoE) (AoE) y <i>multi-head self-attention</i> para predecir las interacciones entre TCRs y péptidos.
Myronov et al. (2023)	BERTrand	BLOSUM62	Emplea un modelo BERT con más de 2.5 millones de parámetros, y utiliza una estrategia de pre-entrenamiento no supervisado para compensar la cantidad de parámetros.
Zhang et al. (2023a)	Hybrid gMLP	BLOSUM50	Deep learning con mecanismos de atención para predecir la interacción de péptidos MHC y TCR, y puede manejar los problemas causados por las diferentes longitudes de los TCR.
Yang et al. (2023)	MIX-TPI	BLOSUM62	Emplea CNNs con <i>self-attention</i> para construir un extractor basado en secuencias y características fisicoquímicas. Luego las fusiona con una capa de <i>self-attention</i> para predecir las interacciones TCR-péptido.
Bradley (2023)	TCRdock	BLOSUM62	Una versión especializada de AlphaFold para generar modelos de interacciones pMHC-TCR que se pueden utilizar para distinguir los <i>epitopes</i> de los incorrectos.
Fang et al. (2022b)	ATMTCR	BLOSUM50	ATMTCR se alimenta en dos capas completamente conectadas, un modelo <i>Contrastive learning-based</i> y NetMHCpan para predecir la unión del TCR y el complejo PMHC.
Cai et al. (2022)	ATM-TCR	<i>One-hot</i> & BLOSUM	Consta de dos <i>encoders</i> y un <i>decoder</i> lineal. Cada secuencia se alinea mediante IMGT. Calcula la similitud de los mapas de atención y los mapas de referencia para confirmar si hay una unión.
Xu et al. (2022)	AttnTAP	Vectores de Granularidad	Incluye una red BiLSTM, con mecanismos de atención y un perceptrón multicapa para extraer las características del TCR y el péptido y predecir la unión pMHC-TCR.
Shang et al. (2022)	pMTattn	Incrustación	Emplea cross <i>self-attention</i> para aprender información de interacción entre pMHCs y TCRs.
Xu et al. (2021)	DLpTCR	<i>One-hot</i>	DLpTCR consiste una red totalmente conectada, LeNet-5 y ResNet-20 para predecir la unión péptido-CDR3 $\alpha(\beta)$. También implementa un mecanismo de atención en ResNet-20.

TABLA 3.6: Lista de *pipelines* desarrollados desde 2018 hasta la fecha para la detección de neoantígenos. GN: *Gene Expression*, VA: *Variant Annotation*.

Nombre	Referencia	Entrada	Salida	Herramientas
PEPPRMINT	Zhou et al. (2023)	DNA-seq	Neoantigens	BWA, Mutect, Strelka, ANNOVAR, OptiType, PEPPRMINT, netMHCpan4.1.
PGNneo	Tan et al. (2023)	VCF, RNA-seq, MS data	Neoantigens	Trimmomatic, BWA, SAMtools, GATK, Picard, OptiType, Annovar, Bedtools, MaxQuant, NetMHCpan4.1, Blastp.
HLA3D	Li et al. (2022)	VCF, HLA, SMG, HBV	Neoantigens	MHCcluster, SAVES, PROCHECK, CoDockPP, Verify 3D, ERRAT, ClusterW2, 3Dmol, PSRPRED4.0, MHCflurry.
NextNEOpI	Rieder et al. (2022)	WES/WGS, RNA-seq	Neoantigens	OptiType, pVACseq, NetMHCpan, MHCflurry, NeoFuse, MiXCR.
Seq2Neo	Diao et al. (2022)	WES/WGS, RNA-seq	Neoantigens	Mutect2, STARFusion, ANNOVAR, Agfusion, NetMHCpan, MHCflurry, Pick-Pocket, NetMHCcon, TPMcalculator, NetCTLpan.
NAP-CNB	Wert-Carvajal et al. (2021)	RNA-seq	Neoantigens	Star, Picard, GATK, SplitNCigarsReads, MuTect2, Cufinks, Epi-Seq, pVACseq, Neoantimon, MuPeXI, BLOSUM62.
NeoANT-HILL	Coelho et al. (2020)	RNA-seq, VCF	Neoantigens, GE	GATK, Mutect2, Optitype, NetMHC, NetMHCpan, NetMHCCcons, NetMHCCstapan, PickPoket, SMM, SMMPMBEC, MHCflurry, NetMHCIIpan, NN-align, SMM-align, Sturniolo, Kallisto.
Neoepiscope	Wood et al. (2020)	VCF, BAM	Neoantigens	BWA, Bowtie2, Pindel, MuSE, RADIA, SomaticSniper, VarScan2, GATK, HapCUT2.
OpenVax	Kodysh and Rubinsteyn (2020)	DNA-seq, RNA-seq	Neoantigens	GATK 3.7, STAR, MuTect 1.1.7, Mutect 2, Strelka, NetMHCpan, NetMHCCcons, SMM, SMM with a Peptide.
ProGeo-neo	Li et al. (2020)	RNA-seq, VCF	Neoantigens	SRA Toolkit, BWA, GATK, Bcftools, ANNOVAR, Kallisto, OptiType, NetMHCpan4.0.
pVACtools	Hundal et al. (2020)	VCF	Neoantigens	CWL36, Cromwell37, ADNC38, BWA-MEM25, HaplotypeCaller28, MHCflurry14, MHCnuggets15, NetChop17, INTEGRATE-Neo19.
TruNeo	Tang et al. (2020)	DNA-seq, RNA-seq	Neoantigens	BWA, GATK v3.3, Somatic SNVs, STAR v2.5.3a, RSEM v1.3.0, NetMHCpan 3.0, netChop.
NeoPredPipe	Schenck et al. (2019)	VCF, HLA	Neoantigens, VA	ANNOVAR, POLYSOLVER, netMHCpan, PeptideMatch.
ScanNeo	Wang et al. (2019)	RNA-seq	Neoantigens	HISAT2, BEDTools, BWA-MEM, pVAC-Seq, NetMHC, NetMHCpan.
Neopepsee	Kim et al. (2018)	RNA-seq, VCF, HLA	Neoantigens, GE	NetCTLpan, Swiss-Prot.
PGV Pipeline	Rubinsteyn et al. (2018)	DNA-seq	Neoantigens	BWA-MEN, BQSR, MuTect, Strelka, STAR, seq2hla, Vaxrank, Isovar, MHCtools, Varcode, pyEnsembl.

TABLA 3.7: Lista de pruebas clínicas que han utilizado vacunas personalizadas basadas en neoantígenos desde el 2028. M: muestra, FC: Fase de Cáncer, FE: Fase de ensayo.

Referencia	M	Tiempo	Tipo de Cáncer	FC	FE
BioNTech (2023)	16	Dec 2019 - Aug 2021	Pancreatic ductal adenocarcinoma	-	I
Rojas et al. (2023)	28	Dec 2019 - Aug 2021	Pancreatic ductal adenocarcinoma	-	I
Yu et al. (2023)	6	Oct 2019 - Aug 2020	MSS-Colorectal Cancer	Advanced	I, II
Holm et al. (2022)	24	12 meses, hasta 5 años	Urothelial carcinoma	Advanced	II
Awad et al. (2022)	16	May 2018 - Apr 2019	Non-squamous non-small cell lung Cancer	Advanced	I
Palmer et al. (2022)	14	32 semanas	Non-small Cell Lung Cancer, MSS-colorectal cancer, gastroesophageal adenocarcinoma and urothelial cancer	Advanced	I
Wang et al. (2022b)	20	5 años desde Jul 2019	Genomic Unstable Solid Tumors	-	I
Ellingsen et al. (2022)	12	Oct 2015 hasta 5 años	Melanoma	Advanced	I, II
Shou et al. (2022)	28	Feb 2018 - May 2021	Different malignant solid tumors	Advanced	I
Rocconi et al. (2022)	24	May 2017 - May 2022	Relapsed Ovarian Cancer	-	I
Cheng et al. (2021)	12	Nov 2017 - Sep 2019	Lung Cancer	Advanced	I, II
Cai et al. (2021)	7	33 meses	Hepatocellular carcinoma	-	I, II
Platten et al. (2021)	28	May 2015 - Nov 2018	Glioma	-	I
Cafri et al. (2020)	4	Mar 2018 - Nov 2019	Gastric Cancer	Advanced	I, II
Ott et al. (2020)	62	23 semanas	Melanoma, Non-small Cell Lung Cancer, or Bladder Cancer	Advanced	I
Kloor et al. (2020)	16	6 meses	Preventive vaccine for pacientes with Lynch Syndrome	-	I, II
Poran et al. (2020)	21	104 semanas	Melanoma	Advanced	I
Engelhard et al. (2020)	12	12 - 26 semanas	Melanoma	-	I
Podaza et al. (2020)	13	2 años	Melanoma	-	II
Sater et al. (2020)	27	May 2014 - Jan 2018	Prostate Cancer	-	II
Mueller et al. (2022)	29	Nov 2016 - Mar 2019	Diffuse midline glioma	Advanced	I, II
Keskin et al. (2019)	8	20 semanas	Glioblastoma	-	I
Aggarwal et al. (2019)	22	May 2014 - Aug 2016	Head and Neck Cancer	Advanced	I, II
Bassani-Sternberg et al. (2019)	12	Sep 2020 - Sep 2028	Pancreatic Adenocarcinoma	-	I
Dillman et al. (2018)	42	5 años	Melanoma	Advanced	II

Capítulo 4

Propuesta

La detección de neoantígenos es un proceso largo, descrito anteriormente. Debido a esto, esta investigación se ha centrado en la predicción de la unión pMHC, porque es una de las etapas con mayor investigación en el estado del arte y sin embargo, los resultados aún carecen de buen desempeño. En resumen, en este trabajo hemos realizado *fine-tuning* a modelos *Transformer*, para la tarea de predicción de la unión pMHC.

4.1. Metodología

Esta investigación se enfoca en la tarea de predecir la unión pMHC, descrito en la etapa 3.1 del proceso general para generar vacunas personalizadas basadas en neoantígenos (ver Figura 4.1). Se ha evaluado seis modelos *Transformers* pre-entrenados en diversas tareas de Proteómica como: predicción de estructura de proteínas, predicción de la función de proteínas, etc. Los modelos *Transformer* son: TAPE ([Rao et al., 2019](#)), ProtBert-BFD ([Elnaggar et al., 2021](#)) y ESM2 ([Lin et al., 2023](#)) (ESM2(t6), ESM2(t12), ESM2(t30), ESM2(t33)). Durante la evaluación se realizó *fine-tuning* a los modelos agregando un bloque de BiLSTM al final, de igual forma que lo realizó HLAB ([Zhang et al., 2022b](#)). También se evaluó el uso de *Gradient Accumulation Steps* (GAS) y el uso de una metodología para congelar las capas del modelo *Transformer*. En la Figura 4.1, describimos la propuesta: primero tomamos como entrada el péptido y el MHC, luego estos son concatenados y son recibidos por el modelo *Transformer* y el bloque BiLSTM respectivamente para predecir su afinidad o unión.



FIGURA 4.1: Propuesta de *transfer learning* de ESM-1b y una red neuronal paralela para la predicción de la afinidad entre un péptido y MHC (peptide MHC binding).

4.2. Modelos BERT

El modelo *Bidirectional Encoder Representations from Transformers* (BERT) fue presentado en el artículo “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” Devlin et al. (2018). Este modelo se trata de un Transformer bidireccional preentrenado con información de las bases de datos de Libros de Toronto y Wikipedia. Es bidireccional porque lee el texto en ambas direcciones. El modelo está compuesto por un *encoder* y un *decoder* (ver Figura 4.2), donde el *encoder* tiene la funcionalidad de codificar los *tokens*, mientras que el *decoder* los decodifica y los convierte nuevamente a texto.

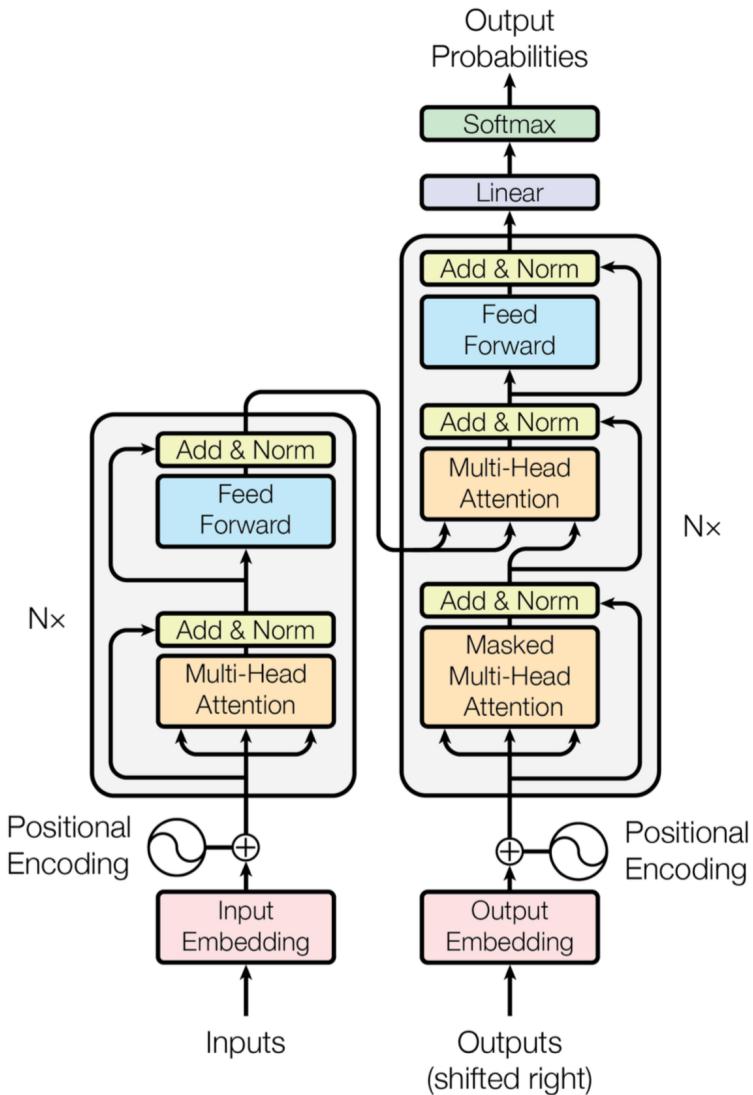


FIGURA 4.2: Arquitectura del modelo BERT diferenciando los *encoders* y *decoders*. Fuente: [Devlin et al. \(2018\)](#).

4.2.1. Modelos de lenguaje de proteínas

El modelo BERT ha permitido grandes avances en el área de procesamiento natural del lenguaje y también se ha extendido hacia otras aplicaciones, como el análisis de secuencias de aminoácidos (proteínas). Por ejemplo, en la Figura 4.3, mostramos un ejemplo de una cadena de aminoácidos y cómo se forma a partir de la secuencia mRNA. Cada tres bases conforman un posible aminoácido y tenemos veintidós en total.

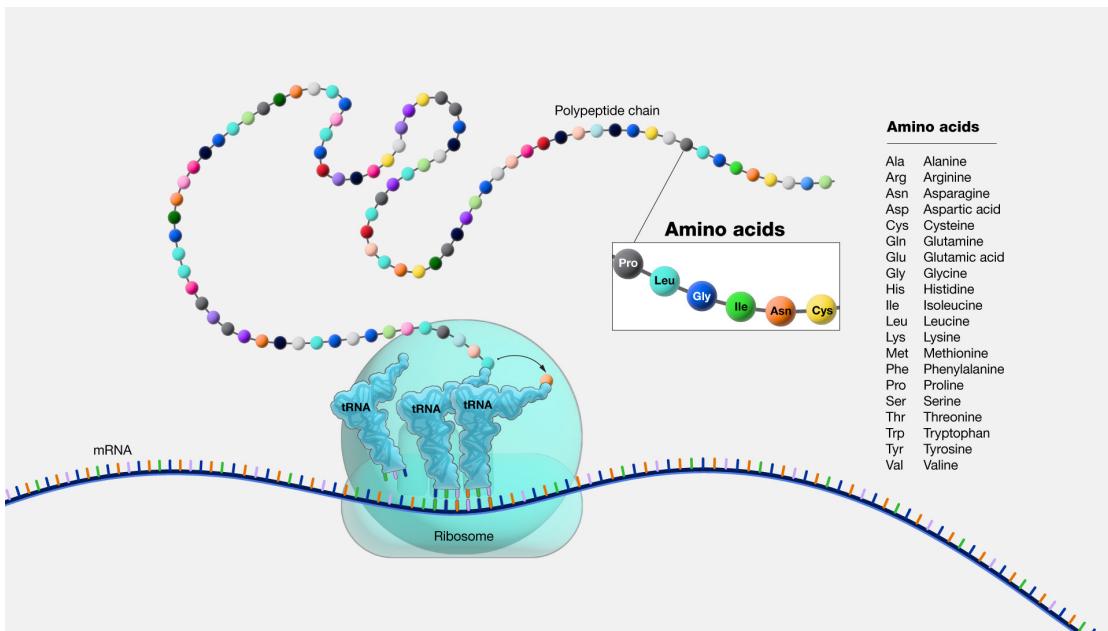


FIGURA 4.3: Ejemplo de como ocurre la traducción de aminoácidos a partir de una secuencia mRNA. Cada tres bases conforman un aminoácido y en total tenemos veintidós diferentes aminoácidos. Fuente: [NIH \(2024\)](#).

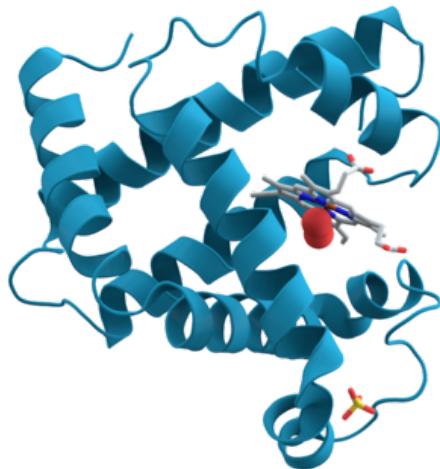


FIGURA 4.4: Estructura 3D de la proteína Mioglobulina. Esta estructura 3D es- ta compuesta por 154 aminoácidos: *MGLSDGEWQ LVNVWGKVE ADIPGHG-QEV LIRLFKGHPE TLEKFDKFHK LKSEDEMkas EDLKKHGATV LTALG-GILKK KGHHEAEIKP LAQSHATKHK IPVVKYLEFIS ECIIQVLQSK HPGDFGA-DAQ GAMNKALELF RKDMASNYKE LGFQG*. Fuente: [UniProt \(2024\)](#), identifica- ción P02144.

Según la cadena de aminoácidos, la estructura y función de una proteína es determinada ([Rastogi et al., 2022](#); [Kihara and Kihara, 2017](#); [Rangwala and Karaypis, 2010](#)). Por ejemplo la secuencia de aminoácidos: *MGLS DGEWQ LVNV WGKVE ADIPG HGQEV LIRL FKGHPE TLEKF DKFKH LKSE DEMKAS EDLK KHGATV LTALG GILKK KGHH EAEIKP LAQSH ATKHK IPVKY LEFIS ECIIQ VLQSK HPGD FGADAQ*

GAMNK ALELF RKDM ASNYKE LGFQG, representa la proteína Mioglobulina (ver Figura 4.4). Según la posición de cada aminoácido y el tamaño de la secuencia, la proteína cumple con cierta función. Entonces, si una mutación ocurre a nivel de aminoácidos por ende la función de la proteína se verá afectada, este es origen de varios tipos de cáncer (Xie et al., 2023; Biswas et al., 2023). Por ejemplo, en la Figura 4.5, presentamos una mutación por sustitución, donde el aminoácido K cambio por E, este cambio pequeño modificó la estructura y función de la proteína.

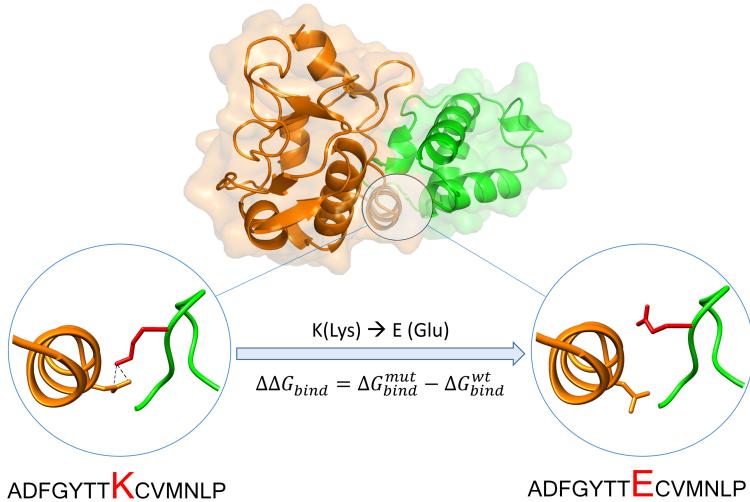


FIGURA 4.5: Estructura 3D de una proteína normal y una proteína mutada. La mutación ocurrió en el aminoácido K, que cambió y se volvió el aminoácido E. Esta mutación generó un cambio en la estructura y función de la proteína. Fuente: [MutaBind \(2024\)](#)

Entonces, una secuencia de aminoácidos es muy similar a una cadena de texto. En una cadena de texto, las palabras pueden ser los *tokens*; mientras que en la secuencia de aminoácidos, cada aminoácido sería un *token*. Además, el orden de los aminoácidos determina la estructura y función de las proteínas. Basándonos en estas premisas, varios autores han considerado el estudio de proteínas de manera similar a una tarea de procesamiento natural del lenguaje. De esta forma, existen trabajos que han utilizado Transformers para diversas tareas en Proteómica, como la predicción de la estructura de una proteína, la predicción de la función y del *contact map*, así como la predicción del enlace entre dos proteínas. Esta área de estudio es tan importante que, de manera similar a los modelos de lenguaje grandes como Llama2 y Mistral, se han entrenado modelos BERT con millones de secuencias de proteínas. Incluso, se han utilizado metodologías similares para el entrenamiento de modelos de lenguaje. Por ejemplo, tradicionalmente los modelos de lenguaje utilizan una técnica de enmascaramiento en su entrenamiento auto supervisado, la cual consiste en ocultar algunas palabras y entrenar al modelo para que pueda predecir la palabra oculta. De igual forma, se ha realizado con las secuencias de aminoácidos, donde se ocultaron ciertos aminoácidos.

Evaluamos seis modelos BERT: TAPE (Rao et al., 2019), ProtBert-BFD (Elnaggar et al., 2021) y ESM2 (Lin et al., 2023) (ESM2(t6), ESM2(t12), ESM2(t30), ESM2(t33)). Estos modelos fueron entrenados con grandes conjuntos de datos de secuencias de proteínas como Pfam (El-Gebali et al., 2019), BFD y UniRef50 (Suzek et al., 2015). Además, se realizó *fine-tuning* para la predicción de unión pMHC-I. En la Tabla 4.1, presentamos las características de cada modelo.

TABLA 4.1: Diferencias significativas entre los modelos TAPE, ProtBert-DFB y ESM2.
HS: *Hidden size*; AH: *Attention heads*.

Modelo	BD	Muestras	Capas	HS	AH	Params.
TAPE	Pfam	30M	12	768	12	92M
ProtBert-BFD	BFD	2122M	30	1024	16	420M
ESM2(t6)	Uniref50	60M	6	320	20	8M
ESM2(t12)	Uniref50	60M	12	480	20	35M
ESM2(t30)	Uniref50	60M	30	640	20	150M
ESM2(t33)	Uniref50	60M	33	1280	20	650M

4.2.2. TAPE

Tasks Assessing Protein Embeddings (TAPE) (Rao et al., 2019) es el primer intento de evaluar el aprendizaje semi-supervisado en secuencias de proteínas. TAPE consta de doce capas de 512 unidades con ocho *attention-heads*, lo que resulta en un total de 92 millones de parámetros. Los autores aplicaron entrenamiento semi-supervisado con la base de datos Pfam (El-Gebali et al., 2019), que contiene treinta millones de dominios de proteínas. Además, el conjunto de datos Pfam representa un subconjunto del *Knowledge Base UniProt* (UniProtKB) (Consortium et al., 2018); en particular, Pfam utilizó secuencias de *Reference Proteomes* (Finn et al., 2016) en lugar de utilizar todo el conjunto de datos de UniProtKB. En consecuencia, Pfam tiene casi la mitad de las secuencias de proteínas que otras bases de datos extraídas de UniProtKB.

4.2.3. ProtBert-BFD

ProtBert-BFD es parte de una familia de modelos de ProtTrans (Elnaggar et al., 2021). Los autores evaluaron varias arquitecturas de aprendizaje profundo con los conjuntos de datos BFD, UniRef50 y UniRef100, cada uno con 2122, 45 y 216 millones de secuencias.

Añadido a esto, BFD se considera la colección más extensa de secuencias de proteínas; fusiona UniProt (Consortium, 2019) y proteínas de múltiples proyectos de secuenciación de metagenómica. Mientras tanto, UniRef (Suzek et al., 2015) proporciona un conjunto *clusterized* de secuencias de proteínas de UniProtKB. Es importante destacar que el conjunto de datos más grande, BFD, las muestras tienen ruido y contiene errores en las secuencias (Elnaggar et al., 2021).

Algunos de los modelos propuestos son ProtBert-BFD, ProtT5-XL y ProtT5-XXL, que tienen 420 millones, 3 mil millones y 11 mil millones de parámetros, respectivamente. ProtBert-BFD se entrenó con BFD; mientras tanto, los modelos ProtT5 se entrenaron inicialmente con BFD y luego con UniRef50, lo que mejoró el rendimiento en un 2.8 % y un 1.4 % para ProtT5-XL y ProtT5-XXL, respectivamente. Sin embargo, ProtT5-XL superó tanto a ProtBert-BFD como al modelo más grande, ProtT5-XXL. Los autores afirmaron que la cantidad de muestras mejoraba el rendimiento, pero no observaron una similitud consistente con el tamaño del modelo. Sugerían que modelos más grandes ven menos muestras con la misma potencia de cálculo, por lo que los modelos más grandes necesitan conjuntos de datos más grandes. Por esta razón, hemos optado por ProtBert, ya que es más pequeño que ProtT5-XL y creemos que se adapta mejor al tamaño del conjunto de datos actual para esta investigación.

4.2.4. ESM2

ESM-2 (Lin et al., 2023) es una familia de modelos *Transformer* que tienen desde 8 millones hasta 15 billones de parámetros. El modelo se basa en BERT (Devlin et al., 2018) y supera a su versión anterior, ESM-1b (Rives et al., 2021), al eliminar las capas de *dropout* en las capas ocultas y de atención. Además, los autores sugirieron que los métodos de codificación de posición absoluta no se extrapolan bien; en consecuencia, utilizaron la *Rotary Position Embedding* (RoPE). Significativamente, el uso de RoPE aumenta ligeramente el costo de entrenamiento; al mismo tiempo, mejora la calidad del modelo para modelos pequeños (Lin et al., 2023). Además, los autores utilizaron el conjunto de datos no redundante UniRef50 (Suzek et al., 2015) de UniProt, que contiene 60 millones de secuencias de proteínas.

4.3. *Fine-tuning*

Una manera de realizar *transfer learning* de modelos de *deep learning* grandes, es realizando *fine-tuning*. Esta metodología, generalmente se aplica cuando tenemos un modelo entrenado con una base de datos muy grande, como por ejemplo una base de datos

de texto de todas las páginas de internet. Luego, queremos adaptar dicho modelo para una tarea específica y en donde no contamos una base de datos muy grande, como por ejemplo un conjunto de muestras de análisis de sentimientos; entonces, podemos aplicar *fine-tuning* al modelo entrenado con todo el texto de internet para una nueva tarea, entrenandolo de nuevo pero esta vez con el objetivo de predecir el sentimiento de un texto. En resumen esta técnica consiste en modificar las ultimas capas del modelo pre-entrenado. La modificación puede ser eliminando las ultimas capas y agregando capas lineales, convoluciones, o recurrentes. Luego, se vuelve a entrenar el modelo una vez mas con el objetivo que el modelo se adapte al nuevo problema. En la Figura 4.6, mostramos un ejemplo de este procedimiento.

Luego, se sabe que al entrenar modelos de *Transformers* grandes, las capas finales experimentan cambios más significativos, mientras que las capas iniciales, más cercanas a la entrada, sufren modificaciones relativamente menores. En otras palabras, las ultimas capas son mas específicas, mientras que las capas iniciales, representan características generales del problema (Merchant et al., 2020; Lee et al., 2019; Kovaleva et al., 2019). Debido a esto, es comun en los procesos de *fine-tuning*, eliminar las ultimas capas y agregar otras capas al final del modelo. En este trabajo, apilamos en cascada un bloque BiLSTM al final del modelo pre-entrenado (ver Figura 4.7), luego se agrego una capa lineal con dos neuronas como salida para el problema de clasificación pMHC. Finalmente, este modelo se entreno con una base de datos de muestras de pMHC. Las características de los modelos BERT pre-entrenados se detallan en la Tabla 4.1 mientras que las características del bloque LSTM utilizado en el proceso de *fine-tuning* fueron inspirados por el trabajo de Zhang et al. (2022b) y se detallan en la Tabla 4.2.

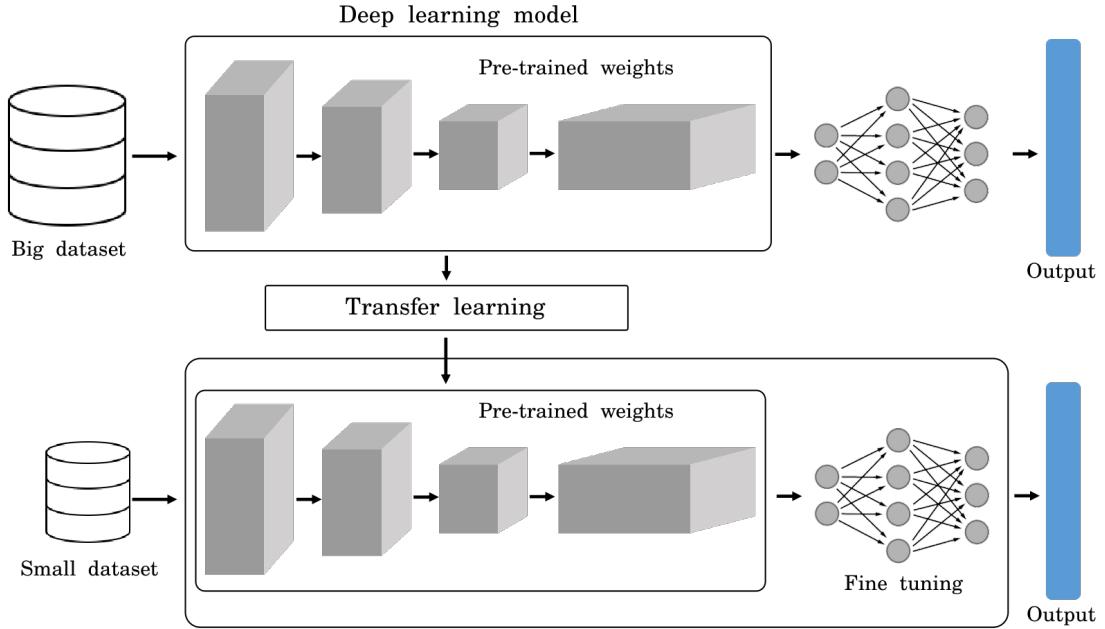


FIGURA 4.6: Ejemplo de aplicación de *Fine-tuning*. Primero, se entrena un modelo para una tarea x con una gran base de datos, luego puede aprovecharse ese aprendizaje para otra tarea similar y que no tenga una base de datos grande. Generalmente, en este proceso, se eliminan las ultimas capas y se agregan otras capas lineales, convoluciones o recurrentes. Fuente: Adaptado de Prince (2023).

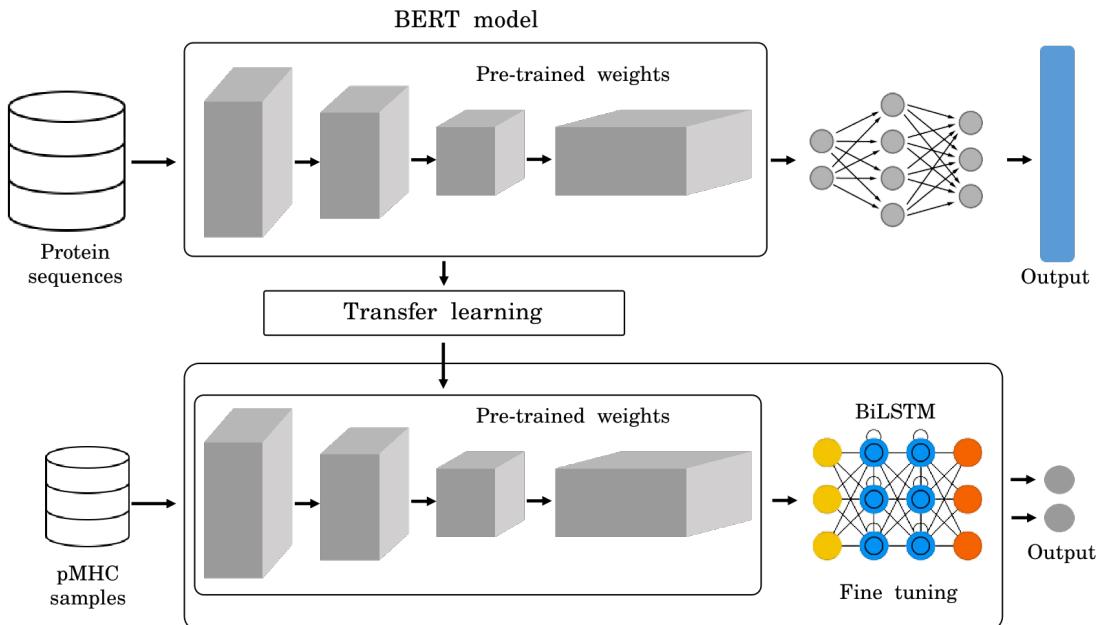


FIGURA 4.7: *Fine-tuning* propuesto. Se toma las primeras capas de un modelo BERT pre-entrenado con bases de datos de secuencias de proteínas. Luego, se agrega un bloque de capas BiLSTM al final y se entrena de nuevo con una base de datos pequeña de muestras pMHC. Fuente: Adaptado de Prince (2023).

TABLA 4.2: Características del bloque BiLSTM utilizado en el *fine-tuning*.

Tipó	Entada	Salida	Capas
BiLSTM	Salida del modelo BERT (Tabla 4.1, columna HS)	768	2
Dropout	0.1 %	-	-
Lineal	1536	2	1

4.4. Gradient Accumulation Steps

Adicionalmente, los modelos de *Transformer* grandes utilizan bastante memoria de la GPU. Por lo tanto, inspirados en trabajos similares sobre entrenamiento de modelos grandes de *Transformers* para problemas de NLP (Anil et al., 2021; Zhang et al., 2023b; Huang et al., 2023), evaluamos los resultados de aplicar *Gradient Accumulation Steps* durante el entrenamiento. El proceso se explica en la Figura 4.8, por ejemplo en un entrenamiento normal durante el proceso de actualización de parámetros en el algoritmo de aprendizaje, se considera un solo *mini-batch* para obtener las gradientes y luego actualiza los parámetros; mientras que en el enfoque utilizando *Gradient Accumulation Steps*, se consideran varios *mini-batches*, se suman sus gradientes y luego estas gradientes acumuladas son utilizadas para actualizar los parámetros.

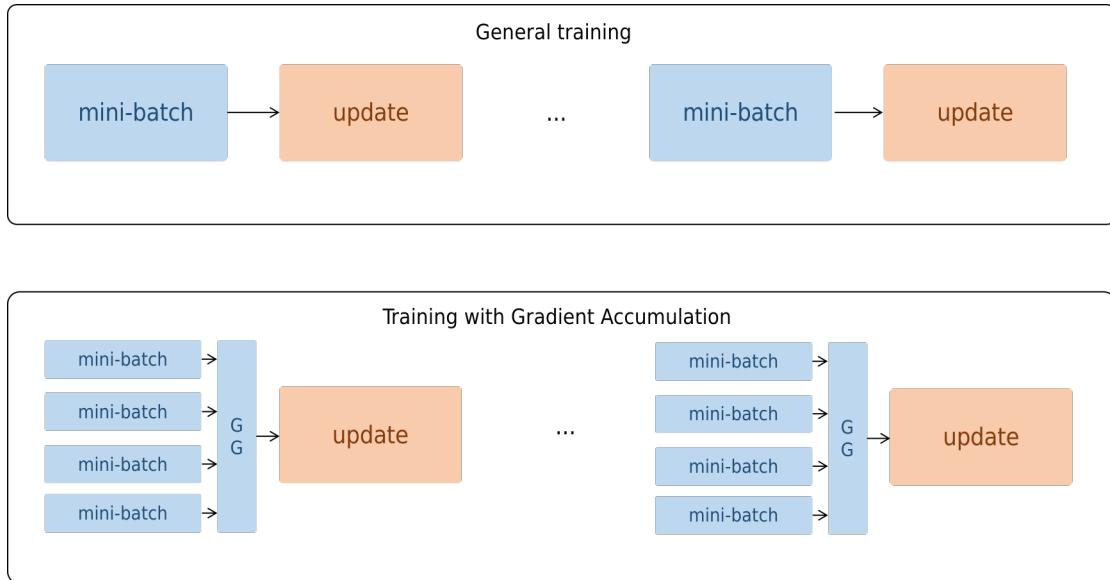


FIGURA 4.8: Ejemplo de aplicación de *Gradient Accumulation Steps* durante el entrenamiento de modelos de Deep Learning. En este caso, un entrenamiento general considera un solo *mini-batch* para obtener las gradientes y luego actualiza los parámetros; mientras que en el enfoque utilizando *Gradient Accumulation Steps*, se consideran varios *mini-batches*, se suman sus gradientes, luego estas gradientes acumuladas son utilizadas para actualizar los parámetros. Fuente: Adaptado de Prince (2023).

Generalmente, los parámetros de un modelo de *machine learning* se actualizan según la ecuación 4.1, donde $\partial^i = \frac{\partial J(\theta)}{\partial \theta}$ es la derivada parcial de θ respecto a la función costo $J(\theta)$ en la iteración i. Entonces, el uso de *Gradient Accumulation Steps* (GAS) propone acumular esas derivadas durante un número de iteraciones como se representa en la ecuación 4.2 (en este caso se acumuló las derivadas por tres iteraciones). Como resultado, al acumular las gradientes la actualización de parámetros se realiza cada cierto número de iteraciones reduciendo así el consumo de memoria.

$$\theta = \theta - \alpha \partial^i \quad (4.1)$$

$$\theta = \theta - \alpha (\partial^{i-1} + \partial^i + \partial^{i+1}) \quad (4.2)$$

En este proyecto, se planteo comprobar los beneficios de utilizar GAS en el entrenamiento de los modelos. Se realizo una comparación de los modelos entrenados con GAS y sin GAS. Para los modelos TAPE, ESM2(t6), ESM2(t12) y ESM2(t30) se acumularán las gradientes por 64 iteraciones. Mientras que para los modelos mas grandes como ProtBert-BFD y ESM2(t33), se acumuló por 128 iteraciones. Se planteo esta diferencia porque los modelos mas grandes requerian un mayor consumo de memoria.

4.5. Hiperparámetros

Finalmente, utilizamos los siguientes hiperparámetros: tasa de aprendizaje de 5e-5, *weight decay* de 0.0001, *momentum* de 0.9, *warn-up steps* de 1000 con *linear decay*, optimizador ADAM ($\beta_1 = 0.9, \beta_2 = 0.999$) y *early stopping*. Estos valores fueron utilizados por BERTMHC ([Cheng et al., 2021](#)) después de buscar los mejores parámetros utilizando *grid search*.

4.6. Ambiente de trabajo

Para entrenar los modelos pequeños de *deep learning*, con menos de 200 millones de parámetros, se utilizo una tarjeta RTX3070, esta cuenta con 8GB de memoria dedicada de vídeo. Para entrenar los modelos grandes, mas de 200 millones de parámetros, se utilizo GPUs en la nube. La plataforma utilizda fue Paperspace, esta brinda diferentes opciones de GPUs, las utilizadas en este proyecto fueron la A100 y la A4000.

4.7. Bases de datos

Utilizamos secuencias de péptidos del conjunto de datos Anthem ([Mei et al., 2021](#)). Este conjunto de datos consta de 539,019 muestras para entrenamiento, 179,673 para validación y 172,580 para pruebas. Por ejemplo, en la Tabla 4.3, mostramos algunas muestras de esta base de datos. La columna MHC especifica el tipo de MHC o HLA, luego la columna pseudosecuencia, representa la secuencia de aminoácidos del MHC; esta pseudosecuencia fue obtenida de la base de datos de NetMHCpan4.1. Seguidamente, prosigue la secuencia de aminoácidos del péptido. Finalmente, la ultima columna es la clase o etiqueta, tendrá un valor de uno cuando existe una unión entre el pMHC y cero en caso contrario.

Adicionalmente, en la Figura 4.9, presentamos la distribución de las muestras por *k-mers*. En Bioinformática, se hace referencia al termino *k-mer* para representar secuencias biológicas de ADN, ARN o proteínas. Para el caso específico de proteínas, una secuencia de *8-mer*, defina que la proteína en cuestión esta compuesta por 8 aminoácidos. Para la base de datos utilizada en esta investigación, los péptidos de *9-mers* constituyen la mayoría de las muestras; mientras que los péptidos *14-mer* son los mas escasos. Además, estos péptidos *14-mer*, usualmente presentan comportamientos distintos. Esto ha ocasionado que los umbrales utilizados por los clasificadores varia dependiendo del *k-mer* y por tipo de clasificador.

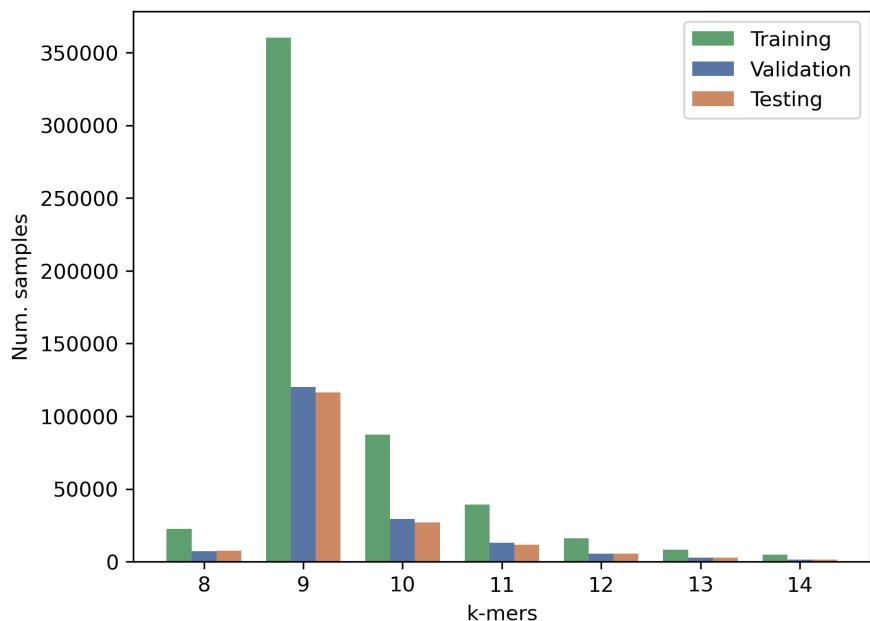


FIGURA 4.9: Cuantificación de las muestras por *k-mers* dentro de los conjuntos de entrenamiento, validación y pruebas. El conjunto de datos se obtuvo de Anthem ([Mei et al., 2021](#)).

TABLA 4.3: Ejemplo de algunas muestras de la base de datos. Cada MHC es representado por una pseudosecuencia procesadas por NetMHCpan4.1. Luego los péptidos son cadenas de aminoácidos entre 8 a 14 amino ácidos. La clase o etiqueta es un cero si existe una unión entre el péptido y el MHC o cero en caso contrario.

MHC	Pseudosecuencia del MHC	Péptido	Clase
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	LFGRDLSY	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	TDKKTHLY	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	RSDTPLIY	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	NSDLVQKY	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	LSDLWDWK	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	LLQNDGFF	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	DSDMQTLV	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	TDYHVRVY	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	VLDSEGYL	1
HLA-A*01:01	YFAMYQENMAHTDANTLYIIYRDYTWWVARVYRGY	SDFHNNRY	1
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	FDGRVVTRSYLEKQ	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	KPCCPDIDIFVDGK	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	QDLKDFMRQAGEVT	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	EGYPKSKKQFEEV	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	GNHISALKRRYTRR	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	RHLRTHTGEKPYVC	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	RGLNNGITPLNSIS	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	SDFALKNPFSLEM	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	ALDSDGDASPGTWSG	0
HLA-C*06:02	YDSGYREKYLQADVNKLYLWYDSYTWAESWAYTWY	QLVLYMKAQQLAA	0

4.8. Clasificación binaria y Métricas

El problema de predicción de unión pMHC es un problema de regresión. Sin embargo, basado en el conjunto de datos utilizado en este estudio, también podría abordarse como un problema de clasificación binaria al seleccionar un umbral apropiado. Las métricas de aprendizaje automático utilizadas en este trabajo son: *accuracy*, *precision*, *recall*, *f-1 score*, *Area Under the Curve (AUC)*, y *Matthews Correlation Coefficient (MCC)*. Todas las métricas están descritas en las ecuaciones siguientes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.4)$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \quad (4.5)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 \times TP}{2 * TP + FP + FN} \quad (4.6)$$

$$Specificity = \frac{TN}{FP + TN} \quad (4.7)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4.8)$$

donde TP , hace referencia a la cantidad de muestras que eran verdaderas y han sido reconocidas como verdaderas; TN , hace referencia a la cantidad de muestras que eran verdaderas y han sido reconocidas como falsas; FP , son las muestras que eran falsas, pero fueron reconocidas como verdaderas; FN , son las muestras que eran falsas y fueron reconocidas como falsas.

Capítulo 5

Experimentos

En este capítulo, detallamos la metodología utilizada para los experimentos. Esta metodología involucró utilizar la base de datos Anthem ([Mei et al., 2021](#)) para entrenar los modelos, se comparó los modelos TAPE, ESM2 y ProtBert-BFD después de aplicar *fine-tuning*. Adicionalmente, se comparó el desempeño de estos al aplicar *Gradient Accumulation Steps* (GAS) y una metodología de congelación de capas. Finalmente, se comparó estos modelos con herramientas del estado del arte: NetMHCpan4.1 ([Reynisson et al., 2020a](#)) y MHCFlurry2.0 ([O'Donnell et al., 2020](#)), Anthem ([Mei et al., 2021](#)), Acme ([Hu et al., 2019](#)) y MixMHCpred2.2 ([Gfeller et al., 2023](#)).

5.1. Modelos BERT

Como se detalló en el Capítulo 4, los modelos BERT pre-entrenados para realizar *fine-tuning* fueron: TAPE ([Rao et al., 2019](#)), ProtBert-BFD ([Elnaggar et al., 2021](#)) y ESM2 (ESM2(t6), ESM2(t12), ESM2(t30), ESM2(t33)) ([Lin et al., 2023](#)). Tanto ProtBert como ESM2 constituyen una familia de varios modelos; sin embargo, para esta tesis se escogió ProtBert-BFD de la familia de modelos ProtBert, y ESM2(t6), ESM2(t12), ESM2(t30), ESM2(t33) de la familia de modelos ESM2. Todos estos modelos se basan en una red *Transformer* BERT que ha sido pre-entrenada con grandes volúmenes de secuencias de proteínas. En la Tabla 4.1 se presenta una comparación a nivel de arquitectura de estos modelos. El pre-entrenamiento fue realizado por los autores de estos modelos y los parámetros del modelo están disponibles de forma gratuita en la plataforma HuggingFace. Basándonos en lo mencionado, este proyecto se enfocó en realizar *fine-tuning* a los modelos BERT para adaptarlos a la tarea de predicción del enlace pMHC.

TABLA 5.1: Nomenclatura utilizada para los modelos entrenados.

Modelo	Descripción
[model]-Normal	Modelo [model] despues de aplicar fine-tuning sin utilizar el congelamiento de capas y GAS
[model]-Freeze	Modelo [model] despues de aplicar fine-tuning utilizando solo el congelamiento de capas
[model]-GAS	Modelo [model] despues de aplicar fine-tuning utilizando solo GAS
[model]-Freeze-GAS	Modelo [model] despues de aplicar fine-tuning utilizando el congelamiento de capas y GAS

5.2. Congelación de Capas y GAS

Para la metodología de congelación de capas, congelamos todos los parámetros del *Transformer* y solo entrenamos el bloque BiLSTM. Utilizar este método acelera el entrenamiento y mantiene el buen rendimiento, como se ha discutido en trabajos previos ([Merchant et al., 2020](#); [Lee et al., 2019](#); [Kovaleva et al., 2019](#)). Adicionalmente, se ha evaluado el efecto de utilizar GAS durante el entrenamiento. Durante, este primer bloque de entrenamiento se ha utilizado tres *epochs*, de igual forma como fue utilizado por otros autores ([Zhang et al., 2022b](#)).

Capítulo 6

Resultados

En este capítulo, detallamos los resultados del *fine-tuning* de los modelos *Transformers* y el efecto de aplicar *Gradient Accumulation Steps* (GAS) y una metodología de congelación de capas. Al principio, entrenamos cada modelo durante tres *epochs*; luego, seleccionamos los modelos con los mejores resultados y los entrenamos nuevamente durante 30 *epochs* con *early stopping*.

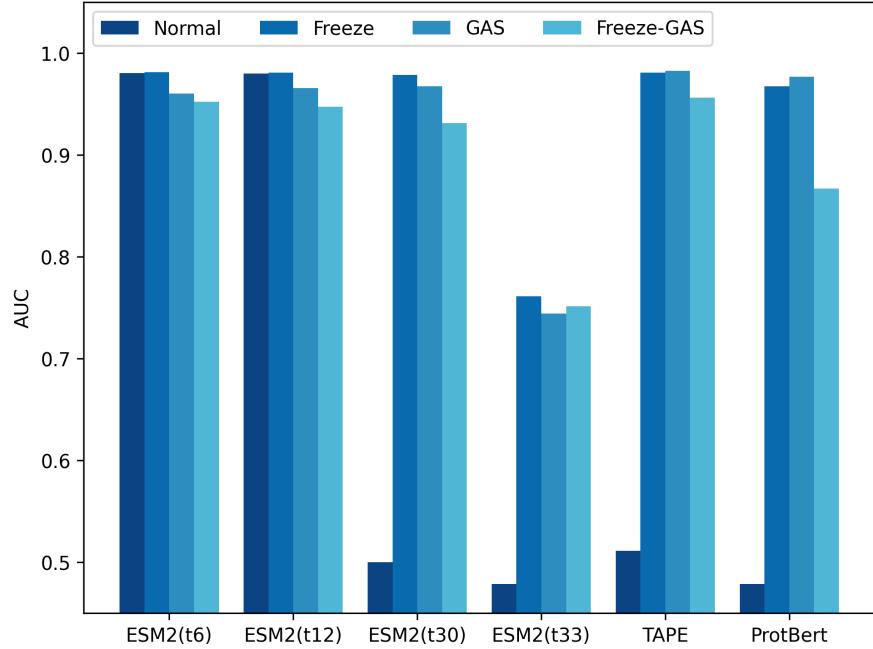
6.1. Congelación de Capas

La comparación de desempeño se presenta en la Tabla 6.1; además, la Figura 6.1(a) muestra el AUC de cada modelo utilizando GAS y la metodología de congelación de capas. En esta comparación, el sufijo 'Normal' representa el entrenamiento utilizando todos los hiperparámetros de la Sección 4.3; el sufijo 'GAS' indica la integración de *Gradient Accumulation Steps* (GAS), mientras que el sufijo 'Freeze' indica la aplicación de la metodología de congelación de capas a los modelos. Según los resultados obtenidos, es evidente que la implementación de la metodología de congelación de capas conduce a mejoras en el rendimiento en todos los modelos ESM2. Además, se observó que los modelos más grandes, como ESM2(t30)-Normal, ESM2(t33)-Normal, TAPE-Normal y ProtBert-Normal, no lograron converger debido a un problema de *vanish gradients*, lo cual se desarrollará en la sección posterior. Además, la utilización de la metodología de congelación de capas permite que estos modelos converjan de manera efectiva.

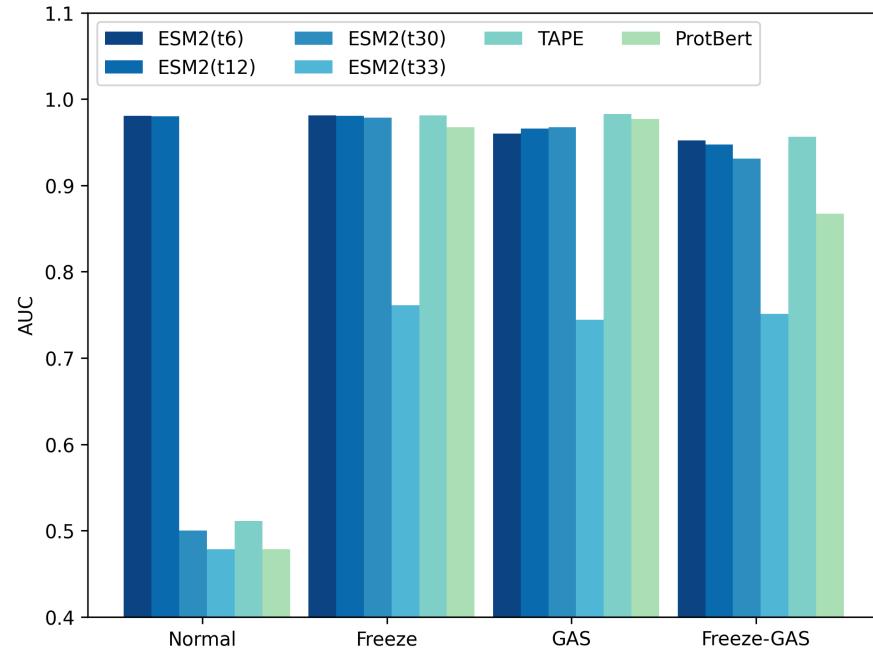
TABLA 6.1: Evaluación del rendimiento de los modelos de *Transformer* utilizando *Gradient Accumulation Steps* (GAS) y la metodología de congelación de capas, **entrenados durante tres epochs**. Además, el sufijo 'Normal' representa el entrenamiento clásico utilizando los hiperparámetros de la Sección 4.3. La inclusión del sufijo 'GAS' en cada modelo indica la integración de *Gradient Accumulation Steps*, mientras que el sufijo 'Freeze' señala la aplicación de la metodología de congelación de capas a los modelos.

Además, el guion '-' en cada celda indica que el modelo no logró converger.

Modelo	Accuracy	Precision	Recall	F1-score	AUC	MCC
ESM2(t6)-Normal	0.9344	0.9334	0.9354	0.9344	0.9805	0.8689
ESM2(t6)-Freeze	0.9351	0.9253	0.9464	0.9357	0.9812	0.8704
ESM2(t6)-GAS	0.8986	0.8966	0.9007	0.8986	0.9602	0.7973
ESM2(t6)-Freeze-GAS	0.8869	0.8913	0.8806	0.8860	0.9520	0.7738
ESM2(t12)-Normal	0.9327	0.9243	0.9422	0.9332	0.9799	0.8655
ESM2(t12)-Freeze	0.9344	0.9251	0.9451	0.9350	0.9808	0.8690
ESM2(t12)-GAS	0.9010	0.9279	0.8692	0.8976	0.9655	0.8037
ESM2(t12)-Freeze-GAS	0.8805	0.8556	0.9149	0.8843	0.9475	0.7629
ESM2(t30)-Normal	-	-	-	-	-	-
ESM2(t30)-Freeze	0.9303	0.9185	0.9440	0.9311	0.9786	0.8609
ESM2(t30)-GAS	0.9090	0.9167	0.8993	0.9079	0.9675	0.8181
ESM2(t30)-Freeze-GAS	0.8565	0.8156	0.9206	0.8649	0.9312	0.7191
ESM2(t33)-Normal	-	-	-	-	-	-
ESM2(t33)-Freeze	0.6818	0.7139	0.6044	0.6546	0.7613	0.3677
ESM2(t33)-GAS	0.6767	0.6312	0.8467	0.7233	0.7442	0.3763
ESM2(t33)-Freeze-GAS	0.6738	0.6254	0.8633	0.7254	0.7514	0.3763
TAPE-Normal	-	-	-	-	-	-
TAPE-Freeze	0.9342	0.9276	0.9415	0.9345	0.9809	0.8684
TAPE-GAS	0.9371	0.9290	0.9463	0.9376	0.9826	0.8744
TAPE-Freeze-GAS	0.8914	0.8851	0.8989	0.8920	0.9564	0.7828
ProtBert-Normal	-	-	-	-	-	-
ProtBert-Freeze	0.9083	0.9176	0.8968	0.9071	0.9673	0.8168
ProtBert-GAS	0.9138	0.9569	0.8662	0.9093	0.9767	0.8313
ProtBert-Freeze-GAS	0.7864	0.7333	0.8988	0.8076	0.8669	0.5881



(a) Comparación por modelo.

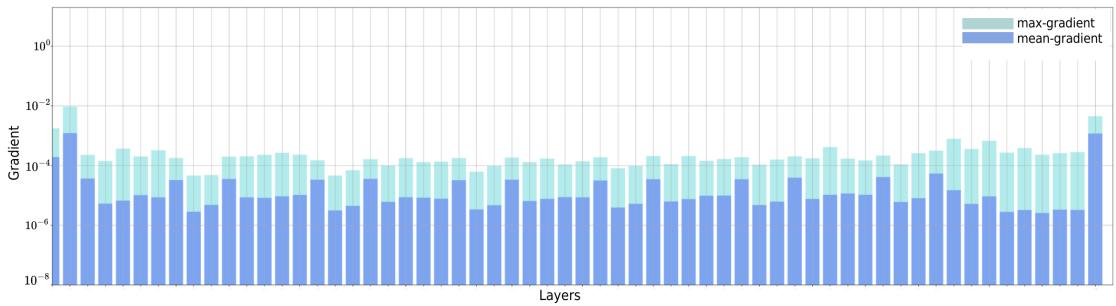


(b) Comparación por la metodología de entrenamiento.

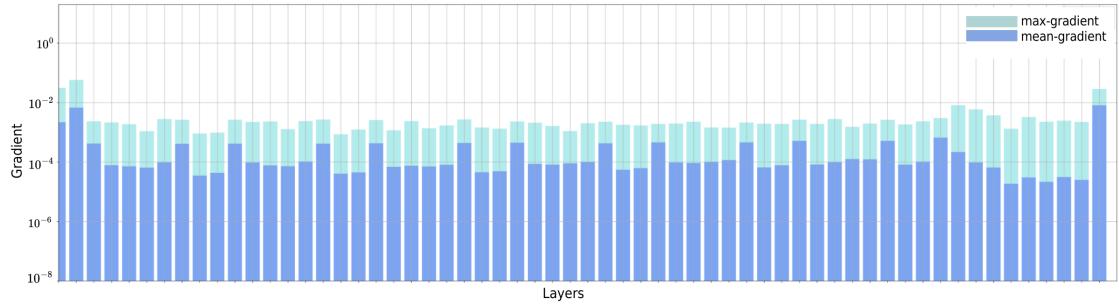
FIGURA 6.1: Análisis comparativo de AUC en arquitecturas de modelos *Transformers* utilizando diversas metodologías de entrenamiento, incluyendo el entrenamiento estándar (Normal), el entrenamiento con una metodología de congelación de capas (Freeze), el entrenamiento con *Gradient Accumulation Steps* (GAS) y el entrenamiento combinando GAS con la congelación de capas (Freeze-GAS). Además, entramos todos los modelos durante tres *epochs*.

6.2. *Vanish Gradient* y GAS

En la Tabla 6.1, notamos que los modelos grandes como ESM2(t30)-Normal, ESM2(t33)-Normal, TAPE-Normal y ProtBert-Normal no lograron converger. Por lo tanto, graficamos los gradientes de estos modelos durante el entrenamiento para descubrir sus causas. En primer lugar, graficamos los gradientes promedio por capa para el modelo más pequeño, ESM2(t6)-Normal, en la Figura 6.2 (este modelo no tuvo problemas de *vanish gradient*). Luego, graficamos los gradientes promedio para ESM2(t30)-Normal (este modelo experimentó un problema de *vanish gradient*) en la Figura 6.3. Como se desprende de los resultados, el modelo más grande, ESM2(t30), mostró gradientes promedio casi nulos en todas las capas de BERT después de solo tres épocas; como resultado, este fenómeno hizo que el modelo no logre converger de manera efectiva.



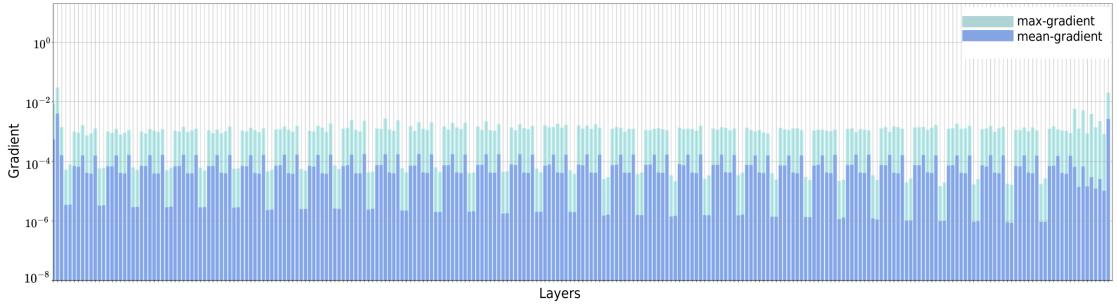
(a) Gráfico de los gradientes por capa antes de entrenar ESM2(t6).



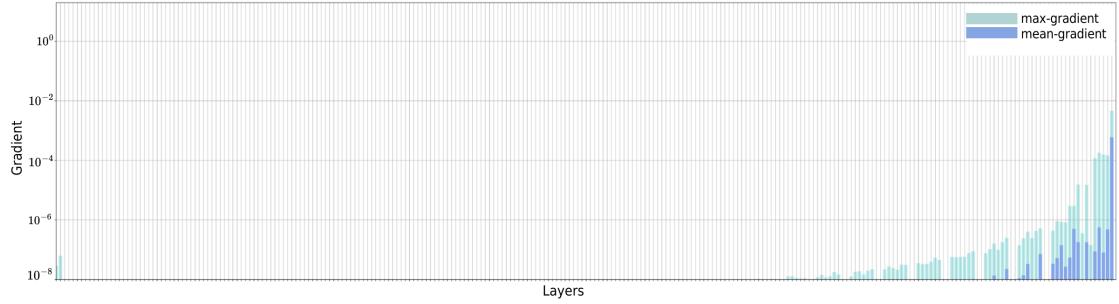
(b) Gráfico de los gradientes por capa después de entrenar ESM2(t6) durante tres *epochs*.

FIGURA 6.2: Promedio y máximo de gradientes por capa para el modelo ESM2(t6). (a) Representa los gradientes antes del entrenamiento, (b) representa los gradientes después del entrenamiento durante tres *epochs*.

Normalmente, el método de *Gradient Accumulation Steps* (GAS) se utiliza para reducir el problema de consumo excesivo de memoria de la GPU durante el entrenamiento (Zhang et al., 2023b; Huang et al., 2023). Sin embargo, esta metodología, puede aliviar ligeramente el problema de *vanish gradient* acumulando gradientes durante algunas iteraciones y solo actualizando el optimizador después de haber realizado un número



(a) Gráfico de los gradientes por capa antes de entrenar ESM2(t30).



(b) Gráfico de los gradientes por capa después de entrenar ESM2(t30) durante tres epochs.

FIGURA 6.3: Promedio y máximo de gradientes por capa para el modelo ESM2(t30).
(a) Representa los gradientes antes del entrenamiento, (b) representa los gradientes después del entrenamiento durante tres epochs.

de iteraciones. Por ejemplo, en la Tabla 6.1, mostramos los resultados después de aplicar GAS a todos los modelos de *Transformer*, después de entrenar durante tres *epochs*. De estos resultados notamos que los modelos ESM2(t30)-Normal, ESM2(t33)-Normal, TAPE-Normal y ProtBert-Normal no convergieron sin GAS; sin embargo, si los entramos con GAS, alcanzaron resultados aceptables.

En particular, la incorporación de *Gradient Accumulation Steps* (GAS) en TAPE-GAS no solo mitiga el problema de *vanish gradient*, sino que también mejora el rendimiento en comparación con otros experimentos que involucran a TAPE, como TAPE-Normal, TAPE-Freeze y TAPE-Freeze-GAS. El mismo fenómeno se observó también en los modelos de ProtBert.

6.3. Entrenamiento (30 epochs)

Para una comparación más detallada, ampliamos los *epochs* de entrenamiento de los modelos con mejor rendimiento de la Tabla 6.1, incluyendo ESM2(T6) y TAPE, a 30 *epochs* con *early stopping*. Además, incluimos a ESM2(t30) para investigar si los modelos más grandes logran mejores resultados durante un período de entrenamiento más largo. Cabe señalar que ProtBert-BFD se excluyó del análisis debido a su bajo rendimiento.

TABLA 6.2: Evaluación del rendimiento de los modelos de *Transformer* con *Gradient Accumulation Steps* (GAS) y la metodología de congelación de capas **entrenados durante treinta (30) epochs**. Además, el sufijo 'Normal' representa el entrenamiento clásico utilizando los hiperparámetros de la Sección 4.3. La inclusión del sufijo 'GAS' en cada modelo indica la integración de *Gradient Accumulation Steps*, mientras que el sufijo 'Freeze' señala nuestra aplicación de la metodología de congelación de capas a los modelos. Además, el guion '-' en cada celda indica que el modelo no pudo converger.

	Accuracy	Precision	Recall	F1-score	AUC	MCC	Termino
ESM2(t6)-Normal	0.9390	0.9333	0.9453	0.9392	0.9797	0.8780	9 epochs
ESM2(t6)-Freeze	0.9401	0.9398	0.9402	0.9400	0.9830	0.8802	6 epochs
ESM2(t6)-GAS	0.9366	0.9322	0.9413	0.9368	0.9818	0.8732	15 epochs
ESM2(t6)-Freeze-GAS	0.9354	0.9326	0.9383	0.9355	0.9813	0.8708	17 epochs
ESM2(t30)-Normal	-	-	-	-	-	-	-
ESM2(t30)-Freeze	0.9393	0.9304	0.9493	0.9397	0.9787	0.8787	14 epochs
ESM2(t30)-GAS	0.9346	0.9337	0.9352	0.9345	0.9808	0.8691	17 epochs
ESM2(t30)-Freeze-GAS	0.9363	0.9319	0.9411	0.9365	0.9818	0.8726	27 epochs
TAPE-Normal	-	-	-	-	-	-	-
TAPE-Freeze	0.9395	0.9404	0.9382	0.9393	0.9815	0.8790	9 epochs
TAPE-GAS	0.9415	0.9352	0.9484	0.9418	0.9841	0.8831	5 epochs
TAPE-Freeze-GAS	0.9359	0.9297	0.9428	0.9362	0.9820	0.8719	18 epochs

Como se indica en la Tabla 6.2, los modelos ESM2 obtienen sus mejores resultados cuando se aplica la metodología de congelación de capas. En cambio, para TAPE, los mejores resultados se logran al utilizar GAS sin congelación. Es importante destacar que TAPE-GAS y ESM2(t6)-Freeze produjeron los resultados más favorables, con TAPE-GAS superando ligeramente a ESM2(t6)-Freeze en este aspecto.

6.4. Comparación con los Métodos del Estado del Arte

Además, comparamos los mejores modelos, ESM2(t6)-Freeze y TAPE-GAS, entrenados durante 30 epochs (consulte la Tabla 6.2), con métodos de vanguardia. Cubrimos NetMHCpan4.1 (Reynisson et al., 2020a) y MHCFlurry2.0 (O'Donnell et al., 2020) porque son métodos de referencia bien conocidos; y tres herramientas más recientes como Anthem (Mei et al., 2021), Acme (Hu et al., 2019) y MixMHCpred2.2 (Gfeller et al., 2023).

Durante la evaluación de estas herramientas en el conjunto de datos de *testing*, encontramos consideraciones específicas para ACME. Para garantizar una evaluación justa, excluimos los siguientes *alleles* de la evaluación para ACME: HLA-C01:02, HLA-C02:02, HLA-C03:03, HLA-C03:04, HLA-C04:01, HLA-C05:01, HLA-C06:02, HLA-C07:01, HLA-C07:02, HLA-C07:04, HLA-C08:02, HLA-C12:03, HLA-C14:02, HLA-C15:02, HLA-C16:01, HLA-C17:01, HLA-A02:50, HLA-A24:06, HLA-A24:13, HLA-A32:15, HLA-B45:06 y HLA-B83:01. Esta exclusión fue necesaria ya que ACME no pudo predecir la unión péptido-MHC para estos *alleles* particulares.

Es importante señalar que la elección del umbral para predecir la unión de pMHC puede variar según la herramienta específica y el *k-mer* utilizado. Esta variabilidad hace que el AUC sea una métrica ideal para comparar métodos, ya que proporciona una evaluación robusta que no es sensible a las diferencias de umbral. Por esta razón, en la Figura 6.4 y 6.5, presentamos el AUC y la curva ROC respectivamente de TAPE-GAS, ESM2(t6)-Freeze, NetMHCpan4.1 y MHCflurry2.0, Anthem, Acme y MixMHCpred2.2. Según estos gráficos, TAPE-GAS y ESM2(t6)-Freeze obtuvieron el valor más alto de AUC.

Además, al evaluar métricas de rendimiento de clasificación binaria, estandarizamos el umbral para TAPE-GAS y ESM2(t6) en 0.5. Mantuvimos un umbral de 0.5 para NetMHCpan4.1, de acuerdo con su configuración recomendada, mientras que para ACME, seguimos un umbral de 0.42, según lo recomendado en su documentación. En el caso de Anthem, la herramienta proporcionó predicciones binarias de unión directamente. Sin embargo, para MixMHCpred2.2 y MHCfurry, determinamos los valores óptimos de umbral a partir del conjunto de datos de *testing*, resultando en 2.7308 y 0.09439, respectivamente. En la Tabla 6.3, presentamos una comparación exhaustiva entre TAPE-GAS y ESM2(t6)-Freeze (entrenados durante 30 *epochs*) y las herramientas del estado del arte. Los resultados demuestran claramente que TAPE-GAS y ESM2(t6)-Freeze superan consistentemente a las herramientas del estado del arte en todas estas métricas: *AUC*, *precisión*, *recall*, *F1-score* y *MCC*.

Finalmente, mostramos la distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 épocas, junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MHCflurry2.0. En este caso, evaluamos cada distribución por *k-mer* (Figura 6.6, 6.7 y 6.8). En todos los *k-mer*, TAPE-GAS y ESM2(t6)-Freeze presentan la puntuación más alta. Además, es importante destacar que el modelo pequeño ESM2(t6)-Freeze ofrece resultados superiores para péptidos más largos con longitudes de 11, 12, 13 y 14.

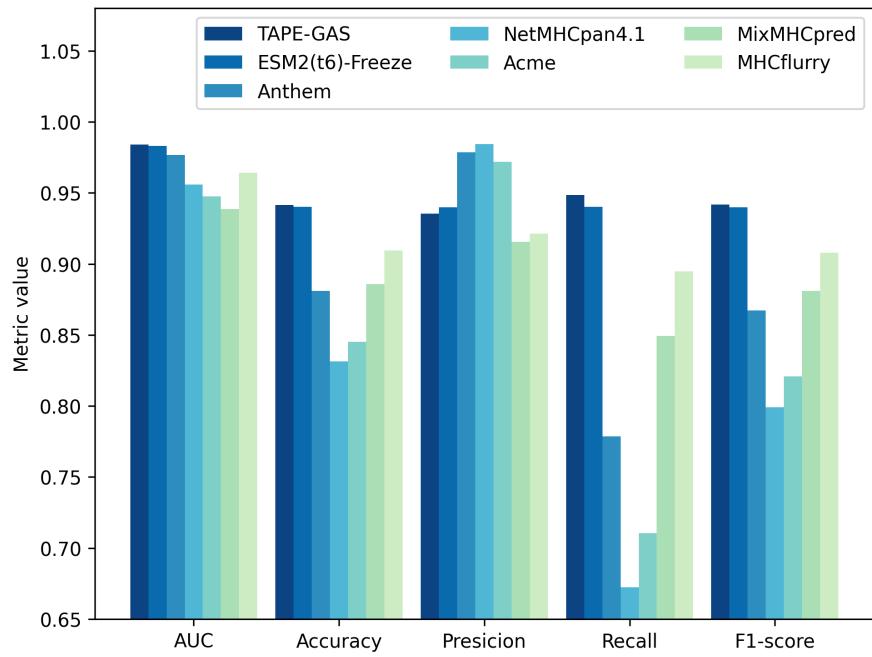


FIGURA 6.4: Los valores de AUC para TAPE-GAS y ESM2(t6) entrenados durante 30 epochs, en comparación con las herramientas de vanguardia.

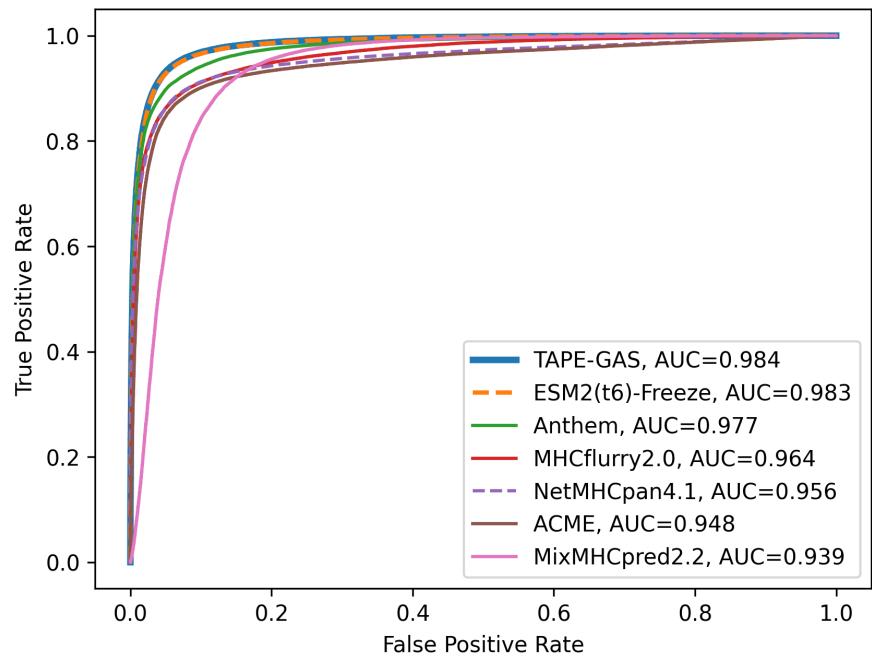


FIGURA 6.5: Las curvas ROC para TAPE-GAS y ESM2(t6) entrenados durante 30 epochs, en comparación con las herramientas de vanguardia.

TABLA 6.3: Evaluación del desempeño de los modelos de *Transformer* TAPE-GAS y ESM2(t6)-Freeze, entrenados durante 30 epochs, en comparación con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MhcFlurry2.0.

	Accuracy	Precision	Recall	F1-score	AUC	MCC
TAPE-GAS	0.9415	0.9352	0.9484	0.9418	0.9841	0.8831
ESM2(t6)-Freeze	0.9401	0.9398	0.9402	0.9400	0.9830	0.8802
Anthem	0.8811	0.9786	0.7787	0.8673	0.9768	0.7785
NetMHCpan4.1	0.8312	0.9844	0.6724	0.7991	0.9557	0.6982
ACME	0.8452	0.9717	0.7105	0.8208	0.9476	0.7165
MixMHCpred2.2	0.8857	0.9155	0.8493	0.8811	0.9386	0.7733
MhcFlurry2.0	0.9093	0.9211	0.8948	0.9078	0.9642	0.8189

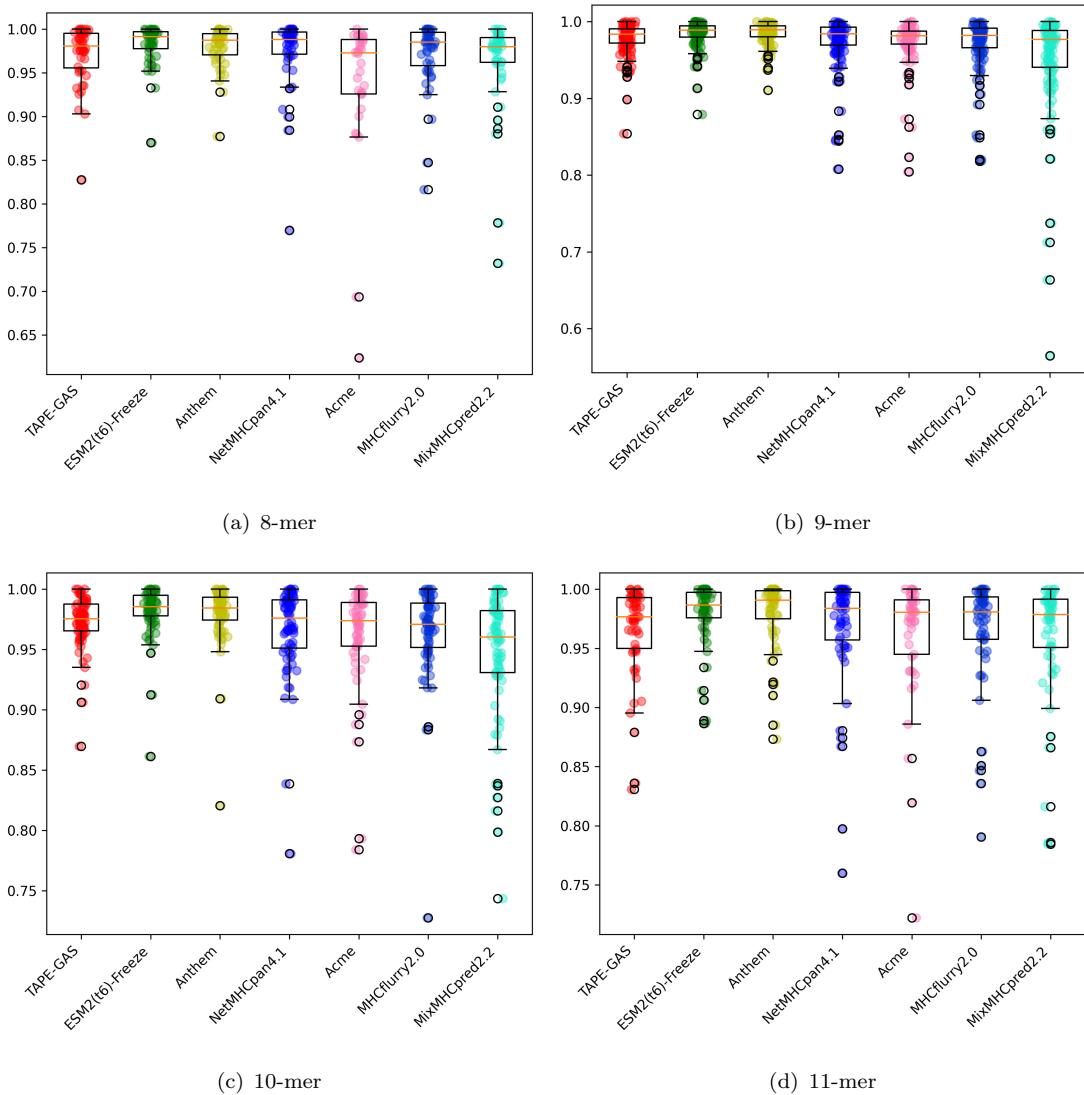


FIGURA 6.6: La distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 epochs para 8, 9, 10 y 11-mers; junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y Mhcflurry2.0.

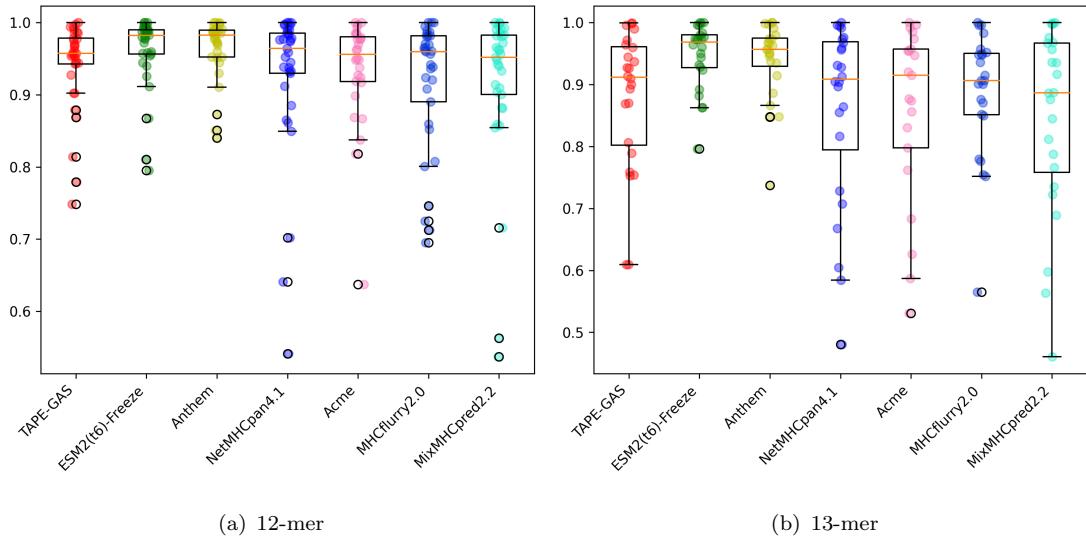


FIGURA 6.7: La distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 *epochs* para 12 y 13-*mers*; junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MHCflurry2.0.

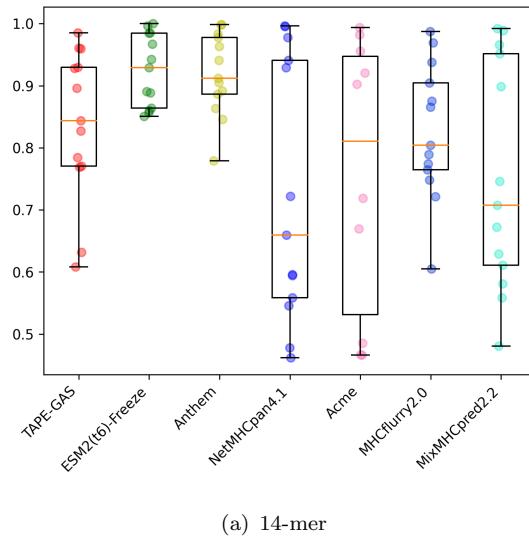


FIGURA 6.8: La distribución de AUC para TAPE-GAS y ESM2(t6)-Freeze, ambos entrenados durante 30 *epochs* para los péptidos 14-*mer*; junto con Anthem, NetMHCpan4.1, ACME, MixMHCpred2.2 y MHCflurry2.0.

6.5. Herramienta para Predicción de la Unión pMHC

En base a los resultados obtenidos se ha desarrollado una herramienta de línea de comandos. En esta herramienta podemos hacer predicciones de los modelos con mejor desempeño desarrollados en esta tesis: ESM2(t6)-Freeze y TAPE-GAS, para la predicción de la

unión pMHC. Además, podemos reentrenar estos modelos con otras bases de datos. La herramienta es código libre y esta en este repositorio: <https://github.com/arceda/pmh>.

6.6. Discusión

6.6.1. *Fine-tuning* ESM2

Cuando se consideran específicamente los modelos *Transformer* de ESM2, los resultados más favorables se obtuvieron con el modelo más pequeño, ESM2(t6), como se indica en la Tabla 6.1. Sin embargo, es importante destacar que los autores de ESM2 informaron en su artículo que, para diversas otras tareas, modelos más grandes como ESM2(t30) y ESM2(t33) superaron a los más pequeños como ESM2(t6) y ESM2(t12) (Lin et al., 2023). Además, está bien establecido que los modelos más grandes tienden a aprender más rápido pero requieren conjuntos de datos de entrenamiento más extensos (Elnaggar et al., 2021). En el caso de la predicción de la unión pMHC-I, nuestro estudio empleó un conjunto de datos que consta de 559,019 muestras, que no consideramos lo suficientemente grande para ESM2(t33), un modelo que cuenta con 650 millones de parámetros. En futuras investigaciones, planeamos evaluar el rendimiento de modelos más grandes utilizando conjuntos de datos más extensos.

Otra razón potencial para el rendimiento superior de ESM2(t6) podría atribuirse al uso de *Rotary Position Embedding* (RoPE) (RoPE) en lugar de la codificación posicional absoluta. Si bien RoPE puede llevar a un ligero aumento en el costo de entrenamiento, se ha observado que mejora la calidad de los resultados, especialmente para modelos más pequeños (Lin et al., 2023).

6.6.2. Congelamiento de Capas y GAS

Durante el entrenamiento de los modelos *Transformer*, exploramos la implementación de una metodología de congelación de capas. Este enfoque implica bloquear el modelo *Transformer* mientras se actualizan solo los parámetros de BiLSTM. Como se informa en varios estudios sobre metodologías de congelación en *Transformers* (Merchant et al., 2020; Lee et al., 2019; Kovaleva et al., 2019), este método generalmente es adecuado para acelerar el proceso de entrenamiento, aunque puede implicar un ligero sacrificio en el desempeño. Sorprendentemente, para los modelos ESM2, esta metodología arrojó los mejores resultados, mientras que para TAPE y ProtBert-BFD, produjo los resultados esperados (ver Tabla 6.1).

Además, nos encontramos con un problema recurrente de *vanish gradient* al entrenar modelos grandes como TAPE-normal, ProtBert-normal, ESM2(t30)-normal y ESM2(t33)-normal (ver Tabla 6.1). Este desafío es un fenómeno común al entrenar modelos de lenguaje grandes, ya que los gradientes tienden a acercarse a valores cercanos a cero después de varias etapas de entrenamiento. Para abordar esto, evaluamos la efectividad de emplear *Gradient Accumulation Steps* (GAS). Este método está diseñado para reducir el consumo de memoria durante el entrenamiento acumulando gradientes durante un cierto número de iteraciones antes de actualizar los parámetros del modelo. En los experimentos, adoptamos la técnica de acumular gradientes durante 64 y 128 iteraciones. Esta técnica alivia ligeramente el problema de gradientes que desaparecen al evitar que los gradientes disminuyan hasta valores cercanos a cero, como se informa para los modelos: ESM2(t30)-GAS, ESM2(t33)-GAS, TAPE-GAS y ProtBert-GAS en la Tabla 6.1. Sin embargo, es importante tener en cuenta que esta técnica principalmente extiende la cantidad de iteraciones de entrenamiento que se pueden realizar antes de que el modelo posiblemente vuelva a enfrentar el problema de *vanish gradient*. Como ejemplo, al entrenar ProtBert-Normal, los gradientes se acercaban a cero, después del primer *epoch*. Sin embargo, con la introducción de GAS (ProtBert-GAS), logramos extender el entrenamiento a tres *epochs* antes de volver a enfrentar el problema de *vanish gradients*, que resurgió después de cuatro *epochs*.

6.6.3. TAPE, ProtBert-BFD y ESM2

En esta investigación comparamos el desempeño de TAPE, ProtBert-BFD y ESM2, cada uno de los cuales se describe en la Tabla 4.1. Las métricas se presentan en la Tabla 6.1. Según esta información, ProtBert-BFD obtuvo el peor resultado a pesar de que este modelo fue pre-entrenado con el conjunto de datos más grande, BFD, que contiene 2122 millones de muestras y tiene 420 millones de parámetros. Creemos que este resultado se debe a la ruido en las muestras y a los errores en las secuencias en el conjunto de datos BFD ([Elnaggar et al., 2021](#)). Además, los modelos *Transformer* grandes requieren más datos para el entrenamiento ([Elnaggar et al., 2021](#)), y en nuestro caso este modelo se entreno con 559,019 muestras.

Además, es destacable que TAPE logró los mejores resultados, con ESM2(t6) siguiendo de cerca (como se muestra en la Tabla 6.1). Los modelos TAPE fueron pre-entrenados utilizando el conjunto de datos Pfam, que es el conjunto de datos más pequeño en esta comparación, con aproximadamente 30 millones de muestras. Es importante mencionar que el conjunto de datos Pfam se deriva de UniProtKB y selectivamente incluye secuencias que pertenecen a *Reference Proteomes* en lugar de abarcar toda la base de datos

de UniProtKB [Finn et al. \(2016\)](#). En consecuencia, Pfam cubre la mitad de las secuencias de proteínas en comparación con otros conjuntos de datos basados en UniProtKB, pero sus muestras son de mayor calidad. Por lo tanto, es lógico suponer que TAPE encapsula una representación más completa y refinada de la información de proteínas en comparación con otros modelos pre-entrenados.

ESM2(t6) logró resultados que compiten estrechamente con el rendimiento de TAPE, como se demuestra en la Tabla [6.2](#). Es importante destacar que ESM2(t6) consta de solo 8 millones de parámetros, en comparación con los 92 millones de parámetros de TAPE. Además, ambos modelos fueron entrenados en muestras de UniProtKB, aunque TAPE utilizó un subconjunto de este conjunto de datos. Además, ESM2(t6) superó a TAPE para péptidos más largos, que van desde 11 a 14 mers, como se muestra en la Figura [6.6](#). Estos hallazgos sitúan firmemente a ESM2(t6) como un candidato destacado para análisis futuros debido a su notable rendimiento y eficiencia.

Capítulo 7

Conclusiones y Trabajos Futuros

7.1. Conclusiones

PRIMERA: Se revisó y analizó los métodos que utilizan *Transformers* para la predicción del enlace pMHC. Este análisis reforzó la hipótesis del uso de *Transformers* y *transfer learning* en los trabajos recientes, estos motivados por el gran auge de los modelos BERT en el procesamiento natural de lenguaje. Adicionalmente a esto, también se incluyó los métodos adyacentes necesarios para la detección de neoantígenos en el marco de desarrollo de vacunas personalizadas contra el cáncer. Estos métodos incluyen el análisis de métodos para la predicción de la unión pMHC-TCR, *pipelines* y ensayos clínicos. Este análisis ha demostrado que existen varios problemas en el proceso de detección de neoantígenos como: la no inclusión de datos de MS, bajo rendimiento en las herramientas de predicción pMHC y PMHC-TCR, así como la falta de métodos para la detección de fusión de genes y *alternative splicing*; sin embargo, a pesar de estas limitaciones, ya se han realizado ensayos clínicos con resultados motivadores.

SEGUNDA: Se analizó los modelos *Transformers* pre-entrenados como: TAPE, ProtBert-BFD, y la familia de modelos de EMS2. Estos modelos fueron entrenados en grandes bases de datos de proteínas para tareas como: Predicción de estructuras de proteínas, predicción del *contact map*, predicción de la función de proteínas, etc. De estos, ProtBert-BFD fue entrenado con la base de datos mas grande proteínas actualmente (2122 millones de muestras); sin embargo, las muestras tienen ruido y el modelo presentó resultados pobres en los experimentos de esta tesis. Adicionalmente, TAPE se entreno con la base de datos mas pequeña (30 millones de muestras); sin embargo, presentó mejores resultados en los experimentos porque las muestras pertenecían a *Reference Proteomas*, y esto generó que el modelo represente mejor la información de una proteína.

TERCERA: Se aplicó *fine-tuning* a los seis modelos pre-entrenados agregando un bloque BiLSTM. Adicionalmente, se evaluó el uso de *Gradient Accumulation Steps* (GAS) y una metodología para congelar las capas del modelo *Transformer*. Como resultado, se verificó que en los modelos ESM2, al aplicar el congelamiento de capas, se mejoró el desempeño y además el modelo mas pequeño ESM2(t6), obtuvo los mejores resultados. Esto puede ser causado por el uso RoPE y al tener una base datos aún pequeña para la complejidad de estos datos. ProtBert, era uno de los modelos mas grandes y obtuvo un desempeño pobre, esto porque fue entrenado con una base de datos con mucho ruido. Finalmente, TAPE obtuvo los mejores resultados, el cual se debe a la calidad de las muestras utilizadas en su pre-entrenamiento.

Adicionalmente, en el análisis comparativo de los seis modelos *Transformers*: TAPE, ProtBert-BFD, ESM2(t6), ESM2(t12), ESM2(t30) y ESM2(t33) con la incorporación de GAS y la técnica de congelación de capas. Observamos que ESM2(t6)-Freeze y TAPE-GAS lograron los resultados más favorables. Además, observamos que el uso de GAS ofreció una mitigación menor del problema de *vanish gradientes*, lo que permitió el entrenamiento efectivo de modelos *Transformer* más grandes. Además, descubrimos que la metodología de congelación de capas aceleró el proceso de entrenamiento y produjo los resultados más favorables para los modelos ESM2. En contraste, el uso de GAS condujo a los mejores resultados para TAPE y ProtBert.

CUARTA: También, se volvió a entrenar los modelos de mejor desempeño ESM2(t6)-Freeze y TAPE-GAS, esta vez por 30 *epochs*, para mejorar aún más su desempeño y luego se realizó una comparación con los métodos de mejor desempeño en el estado del arte, tales como: NetMHCpan4.1, MHCflurry2.0, Anthem, ACME y MixMHCpred2.2. Se realizó la comparación en términos de diversas métricas de rendimiento como el AUC, *accuracy*, *precision*, *recall*, *f1-score* y MCC. De esta comparativa, los modelos propuestos ESM2(t6)-Freeze y TAPE-GAS superaron a los demás métodos del estado del arte en *AUC*, *accuracy*, *recall*, *f1-score* y MCC. Esto demuestra las ventajas de aplicar *fine-tuning* a modelos *Transformer* grandes para predecir la unión péptido-MHC.

QUINTA: Finalmente, se ha logrado cumplir con el objetivo general de la tesis. Se ha implementado un método *in silico* basado en *Transformers* y *Transfer Learning* para la detección de neoantígenos, enfocados en la predicción de la unión pMHC. Para llevar a cabo esta implementación, se evaluó varios modelos *Transformers* pre-entrenados y se realizó *Transfer Learning* aplicando *fine-tuning* a los modelos *Transformer*. Finalmente, se comparó estos modelos con las herramientas de mejor desempeño en el estado del arte. Luego de las comparaciones, el método implementado obtuvo el mejor desempeño en términos de AUC, *accuracy*, *recall*, *f1-score* y MCC.

7.2. Trabajos Futuros

En este trabajo, evaluamos los modelos TAPE, ProtBert-BFD, ESM2(t6), ESM2(t12), ESM2(t30) y ESM2(t33), cada uno con 92, 420, 8, 35, 150 y 650 millones de parámetros respectivamente. Sin embargo, existen otras alternativas como ProtT5-XL y ProtT5-XXL, ESM2(t36) y ESM2(t48), cada uno con 3, 11, 3 y 15 billones de parámetros respectivamente. No evaluamos estos modelos debido al tamaño reducido del conjunto de datos y el costo de entrenamiento. No obstante, planeamos evaluar estos enormes modelos *Transformer* con un conjunto de datos más grande que contenga muestras del conjunto de datos Anthem, MixMHCpred2.2 y la evaluación más reciente de herramientas de predicción de unión pMHC.

Además, dada la considerable inversión de recursos asociada al entrenamiento de modelos *Transformer* grandes, planeamos investigar las posibles ventajas de utilizar DistilBERT y LoRA para tareas de entrenamiento y predicciones.

Además, en este trabajo, se aplicó *fine-tuning* al modelo *Transformer* agregando un bloque BiLSTM al final, basado en el trabajo de HLAB. En el futuro, planeamos evaluar la eficacia de un bloque Star-*Transformer*, similar a la metodología empleada en SMHCpan. Además, considerando los resultados prometedores demostrados en ESM-GAT, creemos que la inclusión de una Red de Atención de Grafos (GAT) podría mejorar significativamente el rendimiento de nuestro modelo en investigaciones futuras. Por último, nos gustaría evaluar la metodología utilizada por TransPHLA, debido a su efectividad demostrada en el manejo de péptidos de diferentes longitudes.

7.3. Agradecimiento

Esta investigación fue respaldada parcialmente por la Universidad La Salle y la Universidad Católica San Pablo, con código de proyecto: 01-CPICI-2021. Este fue un financiamiento conjunto para promover la investigación entre ambas universidades.

También, agradezco a mi asesor Dr. Cristian Lopez por su ayuda técnica y soporte en la metodología de la tesis. Adicionalmente, agradezco a los miembros del jurado, por sus comentarios y observaciones de la tesis, estos han sido tomados en cuenta y han mejorado la calidad de la tesis.

Bibliografía

- Abelin, J. G., Keskin, D. B., Sarkizova, S., Hartigan, C. R., Zhang, W., Sidney, J., Stevens, J., Lane, W., Zhang, G. L., Eisenhaure, T. M., et al. (2017). Mass spectrometry profiling of hla-associated peptidomes in mono-allelic cells enables more accurate epitope prediction. *Immunity*, 46(2):315–326.
- Abualrous, E. T., Sticht, J., and Freund, C. (2021). Major histocompatibility complex (mhc) class i and class ii proteins: impact of polymorphism on antigen presentation. *Current Opinion in Immunology*, 70:95–104.
- Aggarwal, C., Cohen, R. B., Morrow, M. P., Kraynyak, K. A., Sylvester, A. J., Knoblock, D. M., Bauml, J. M., Weinstein, G. S., Lin, A., Boyer, J., et al. (2019). Immunotherapy targeting hpv16/18 generates potent immune responses in hpv-associated head and neck cancer. *Clinical Cancer Research*, 25(1):110–124.
- Alvarez, B., Reynisson, B., Barra, C., Buus, S., Ternette, N., Connelley, T., Andreatta, M., and Nielsen, M. (2019). Nnalign_ma; mhc peptidome deconvolution for accurate mhc binding motif characterization and improved t-cell epitope predictions. *Molecular & Cellular Proteomics*, 18(12):2459–2477.
- Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. (2021). Large-scale differentially private bert. *arXiv preprint arXiv:2108.01624*.
- Arceda, V. E. M. (2023). Neoantigen detection using transformers and transfer learning in the cancer immunology context. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 97–102. Springer.
- Awad, M. M., Govindan, R., Balogh, K. N., Spigel, D. R., Garon, E. B., Bushway, M. E., Poran, A., Sheen, J. H., Kohler, V., Esaulova, E., et al. (2022). Personalized neoantigen vaccine neo-pv-01 with chemotherapy and anti-pd-1 as first-line treatment for non-squamous non-small cell lung cancer. *Cancer Cell*, 40(9):1010–1026.
- Bagaev, D. V., Vroomans, R. M., Samir, J., Stervbo, U., Rius, C., Dolton, G., Greenshields-Watson, A., Attaf, M., Egorov, E. S., Zvyagin, I. V., et al. (2020). Vdjdb

- in 2019: database extension, new analysis infrastructure and a t-cell receptor motif compendium. *Nucleic Acids Research*, 48(D1):D1057–D1062.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Bassani-Sternberg, M., Digklia, A., Huber, F., Wagner, D., Sempoux, C., Stevenson, B. J., Thierry, A.-C., Michaux, J., Pak, H., Racle, J., et al. (2019). A phase ib study of the combination of personalized autologous dendritic cell vaccine, aspirin, and standard of care adjuvant chemotherapy followed by nivolumab for resected pancreatic adenocarcinoma—a proof of antigen discovery feasibility in three patients. *Frontiers in immunology*, 10:1832.
- Bassani-Sternberg, M., Pletscher-Frankild, S., Jensen, L. J., and Mann, M. (2015). Mass spectrometry of human leukocyte antigen class i peptidomes reveals strong effects of protein abundance and turnover on antigen presentation*[s]. *Molecular & Cellular Proteomics*, 14(3):658–673.
- BioNTech (2023). Personalized mRNA vaccine immunogenic against PDAC. *Cancer Discovery*, 13(7):1504–1504.
- Biswas, N., Chakrabarti, S., Padul, V., Jones, L. D., and Ashili, S. (2023). Designing neoantigen cancer vaccines, trials, and outcomes. *Frontiers in immunology*, 14:1105420.
- Borden, E. S., Buetow, K. H., Wilson, M. A., and Hastings, K. T. (2022). Cancer neoantigens: Challenges and future directions for prediction, prioritization, and validation. *Frontiers in Oncology*, 12.
- Bradley, P. (2023). Structure-based prediction of t cell receptor: peptide-mhc interactions. *Elife*, 12:e82813.
- Bravi, B., Di Gioacchino, A., Fernandez-de Cossio-Diaz, J., Walczak, A. M., Mora, T., Cocco, S., and Monasson, R. (2023). A transfer-learning approach to predict antigen immunogenicity and t-cell receptor specificity. *Elife*, 12:e85126.
- Bravi, B., Tubiana, J., Cocco, S., Monasson, R., Mora, T., and Walczak, A. M. (2021). Rbm-mhc: a semi-supervised machine-learning method for sample-specific prediction of antigen presentation by hla-i alleles. *Cell systems*, 12(2):195–202.
- Bulik-Sullivan, B., Busby, J., Palmer, C. D., Davis, M. J., Murphy, T., Clark, A., Busby, M., Duke, F., Yang, A., Young, L., et al. (2019). Deep learning using tumor hla peptide mass spectrometry datasets improves neoantigen identification. *Nature biotechnology*, 37(1):55–63.

- Cafri, G., Gartner, J. J., Zaks, T., Hopson, K., Levin, N., Paria, B. C., Parkhurst, M. R., Yossef, R., Lowery, F. J., Jafferji, M. S., et al. (2020). mrna vaccine-induced neoantigen-specific t cell immunity in patients with gastrointestinal cancer. *The Journal of clinical investigation*, 130(11):5976–5988.
- Cai, M., Bang, S., Zhang, P., and Lee, H. (2022). Atm-tcr: Tcr-epitope binding affinity prediction using a multi-head self-attention model. *Frontiers in Immunology*, 13:893247.
- Cai, Z., Su, X., Qiu, L., Li, Z., Li, X., Dong, X., Wei, F., Zhou, Y., Luo, L., Chen, G., et al. (2021). Personalized neoantigen vaccine prevents postoperative recurrence in hepatocellular carcinoma patients with vascular invasion. *Molecular Cancer*, 20:1–13.
- Chen, C., Qiu, Z., Yang, Z., Yu, B., and Cui, X. (2021a). Jointly learning to align and aggregate with cross attention pooling for peptide-mhc class i binding prediction. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 18–23. IEEE.
- Chen, I., Chen, M., Goedegebuure, P., and Gillanders, W. (2021b). Challenges targeting cancer neoantigens in 2021: a systematic literature review. *Expert Review of Vaccines*, 20(7):827–837.
- Chen, Z., Min, M. R., and Ning, X. (2021c). Ranking-based convolutional neural network models for peptide-mhc class i binding prediction. *Frontiers in Molecular Biosciences*, 8:634836.
- Cheng, J., Bendjama, K., Rittner, K., and Malone, B. (2021). Bertmhc: improved mhc-peptide class ii interaction prediction with transformer and multiple instance learning. *Bioinformatics*, 37(22):4172–4179.
- Chu, Y., Zhang, Y., Wang, Q., Zhang, L., Wang, X., Wang, Y., Salahub, D. R., Xu, Q., Wang, J., Jiang, X., et al. (2022). A transformer-based model to predict peptide-hla class i binding and optimize mutated peptides for vaccine design. *Nature Machine Intelligence*, 4(3):300–311.
- Clancy, S. (2008). Genetic mutation. *Nature Education*, 1(1):187.
- Coelho, A. C. M., Fonseca, A. L., Martins, D. L., Lins, P. B., da Cunha, L. M., and de Souza, S. J. (2020). neoant-hill: an integrated tool for identification of potential neoantigens. *BMC Medical Genomics*, 13(1):1–8.
- Consortium, U. (2019). Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515.

- Consortium, U. et al. (2018). Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 46(5):2699.
- Dai, X., Theobard, R., Cheng, H., Xing, M., and Zhang, J. (2018). Fusion genes: A promising tool combating against cancer. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1869(2):149–160.
- Dalianis, H. (2018). Evaluation metrics and evaluation. In *Clinical text mining*, pages 45–53. Springer.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Diao, K., Chen, J., Wu, T., Wang, X., Wang, G., Sun, X., Zhao, X., Wu, C., Wang, J., Yao, H., Gerarduzzi, C., and Liu, X.-S. (2022). Seq2neo: a comprehensive pipeline for cancer neoantigen immunogenicity prediction. *Int J Mol Sci*, 23(19).
- Dillman, R. O., Cornforth, A. N., Nistor, G. I., McClay, E. F., Amatruda, T. T., and Depriest, C. (2018). Randomized phase ii trial of autologous dendritic cell vaccines versus autologous tumor cell vaccines in metastatic melanoma: 5-year follow up and additional analyses. *Journal for immunotherapy of cancer*, 6(1):1–10.
- El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., et al. (2019). The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432.
- El Naqa, I. and Murphy, M. J. (2022). Machine and deep learning in oncology, medical physics and radiology.
- Ellingsen, E. B., Bounova, G., Kerzeli, I., Anzar, I., Simnica, D., Aamdal, E., Guren, T., Clancy, T., Mezheyevski, A., Inderberg, E. M., et al. (2022). Characterization of the t cell receptor repertoire and melanoma tumor microenvironment upon combined treatment with ipilimumab and hert vaccination. *Journal of Translational Medicine*, 20(1):1–13.
- Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. (2021). Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127.
- Engelhard, V. H., Obeng, R. C., Cummings, K. L., Petroni, G. R., Ambakhutwala, A. L., Chianese-Bullock, K. A., Smith, K. T., Lulu, A., Varhegyi, N., Smolkin, M. E., et al.

- (2020). Mhc-restricted phosphopeptide antigens: preclinical validation and first-in-humans clinical trial in participants with high-risk melanoma. *Journal for immunotherapy of cancer*, 8(1).
- Fang, X., Guo, Z., Liang, J., Wen, J., Liu, Y., Guan, X., and Li, H. (2022a). Neoantigens and their potential applications in tumor immunotherapy. *Oncology Letters*, 23(3):1–9.
- Fang, Y., Liu, X., and Liu, H. (2022b). Attention-aware contrastive learning for predicting t cell receptor-antigen binding specificity. *bioRxiv*.
- Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., et al. (2016). The pfam protein families database: towards a more sustainable future. *Nucleic acids research*, 44(D1):D279–D285.
- Fischer, D. S., Wu, Y., Schubert, B., and Theis, F. J. (2020). Predicting antigen specificity of single t cells based on tcr cdr 3 regions. *Molecular systems biology*, 16(8):e9416.
- Gasser, H.-C., Bedran, G., Ren, B., Goodlett, D., Alfaro, J., and Rajan, A. (2021). Interpreting bert architecture predictions for peptide presentation by mhc class i proteins. *arXiv preprint arXiv:2111.07137*.
- Gfeller, D., Schmidt, J., Croce, G., Guillaume, P., Bobisse, S., Genolet, R., Queiroz, L., Cesbron, J., Racle, J., and Harari, A. (2023). Improved predictions of antigen presentation and tcr recognition with mixmhcpred2. 2 and prime2. 0 reveal potent sars-cov-2 cd8+ t-cell epitopes. *Cell Systems*, 14(1):72–83.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Gopanenko, A. V., Kosobokova, E. N., and Kosorukov, V. S. (2020). Main strategies for the identification of neoantigens. *Cancers*, 12(10):2879.
- Grazioli, F., Machart, P., Mösch, A., Li, K., Castorina, L. V., Pfeifer, N., and Min, M. R. (2022). Attentive variational information bottleneck for tcr-peptide interaction prediction. *Bioinformatics*, 39(1):btac820.
- Han, X.-J., Ma, X.-l., Yang, L., Wei, Y.-q., Peng, Y., and Wei, X.-w. (2020). Progress in neoantigen targeted cancer immunotherapies. *Frontiers in Cell and Developmental Biology*, 8:728.
- Hao, Q., Wei, P., Shu, Y., Zhang, Y.-G., Xu, H., and Zhao, J.-N. (2021). Improvement of neoantigen identification through convolution neural network. *Frontiers in immunology*, 12.

- Hashemi, N., Hao, B., Ignatov, M., Paschalidis, I. C., Vakili, P., Vajda, S., and Kozakov, D. (2023). Improved prediction of mhc-peptide binding using protein language models. *Frontiers in Bioinformatics*, 3.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Heyer, E. E. and Blackburn, J. (2020). Sequencing strategies for fusion gene detection. *BioEssays*, 42(7):2000016.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Holm, J. S., Funt, S. A., Borch, A., Munk, K. K., Bjerregaard, A.-M., Reading, J. L., Maher, C., Regazzi, A., Wong, P., Al-Ahmadi, H., et al. (2022). Neoantigen-specific cd8 t cell responses in the peripheral blood following pd-l1 blockade might predict therapy outcome in metastatic urothelial carcinoma. *Nature Communications*, 13(1):1935.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.
- Hu, Y., Wang, Z., Hu, H., Wan, F., Chen, L., Xiong, Y., Wang, X., Zhao, D., Huang, W., and Zeng, J. (2019). Acme: pan-specific peptide–mhc class i binding prediction through attention-based deep neural networks. *Bioinformatics*, 35(23):4946–4954.
- Huang, Z., Jiang, B., Guo, T., and Liu, Y. (2023). Measuring the impact of gradient accumulation on cloud-based distributed training. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 344–354. IEEE.
- Hundal, J., Kiwala, S., McMichael, J., Miller, C. A., Xia, H., Wollam, A. T., Liu, C. J., Zhao, S., Feng, Y.-Y., Graubert, A. P., et al. (2020). Pvactools: a computational toolkit to identify and visualize cancer neoantigens. *Cancer immunology research*, 8(3):409–420.
- Janeway Jr, C. A. (1997). Immunobiology the immune system in health and disease. *Artes Medicas*.
- Jin, J., Liu, Z., Nasiri, A., Cui, Y., Louis, S.-Y., Zhang, A., Zhao, Y., and Hu, J. (2021). Deep learning pan-specific model for interpretable mhc-i peptide binding prediction

- with improved attention mechanism. *Proteins: Structure, Function, and Bioinformatics*, 89(7):866–883.
- Jing, Y., Zhang, S., and Wang, H. (2023). Dapnet-hla: Adaptive dual-attention mechanism network based on deep learning to predict non-classical hla binding sites. *Analytical Biochemistry*, 666:115075.
- Jokinen, E., Huuhtanen, J., Mustjoki, S., Heinonen, M., and Lähdesmäki, H. (2021). Predicting recognition between t cell receptors and epitopes with tcrgp. *PLoS computational biology*, 17(3):e1008814.
- Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier.
- Kalemati, M., Darvishi, S., and Koohi, S. (2023). Capsnet-mhc predicts peptide-mhc class i binding based on capsule neural networks. *Communications Biology*, 6(1):492.
- Kawashima, S. and Kanehisa, M. (2000). Aaindex: amino acid index database. *Nucleic acids research*, 28(1):374–374.
- Kelvin, J. (2022). Rnns, lstms, cnns, transformers and bert.
- Kerbs, P., Vosberg, S., Krebs, S., Graf, A., Blum, H., Swoboda, A., Batcha, A. M., Mansmann, U., Metzler, D., Heckman, C. A., et al. (2022). Fusion gene detection by rna-sequencing complements diagnostics of acute myeloid leukemia and identifies recurring nrrip1-mir99ahg rearrangements. *haematologica*, 107(1):100.
- Keskin, D. B., Anandappa, A. J., Sun, J., Tirosh, I., Mathewson, N. D., Li, S., Oliveira, G., Giobbie-Hurder, A., Felt, K., Gjini, E., et al. (2019). Neoantigen vaccine generates intratumoral t cell responses in phase ib glioblastoma trial. *Nature*, 565(7738):234–239.
- Kihara, D. and Kihara (2017). *Protein Function Prediction*. Springer.
- Kim, P. and Zhou, X. (2019). Fusiongdb: fusion gene annotation database. *Nucleic acids research*, 47(D1):D994–D1004.
- Kim, S., Kim, H. S., Kim, E., Lee, M., Shin, E.-C., and Paik, S. (2018). Neopepsee: accurate genome-level prediction of neoantigens by harnessing sequence and amino acid immunogenicity information. *Annals of Oncology*, 29(4):1030–1036.
- Kim, Y., Denton, C., Hoang, L., and Rush, A. M. (2017). Structured attention networks. *arXiv preprint arXiv:1702.00887*.
- Kim, Y., Sidney, J., Pinilla, C., Sette, A., and Peters, B. (2009). Derivation of an amino acid similarity matrix for peptide: Mhc binding and its application as a bayesian prior. *BMC bioinformatics*, 10:1–11.

- Kloor, M., Reuschenbach, M., Pauligk, C., Karbach, J., Rafiyan, M.-R., Al-Batran, S.-E., Tariverdian, M., Jagger, E., and von Knebel Doeberitz, M. (2020). A frameshift peptide neoantigen-based vaccine for mismatch repair-deficient cancers: a phase i/ii clinical trial. *Clinical Cancer Research*, 26(17):4503–4510.
- Kodysh, J. and Rubinsteyn, A. (2020). Openvax: An open-source computational pipeline for cancer neoantigen prediction. *Methods in molecular biology (Clifton, N.J.)*, 2120:147–160.
- Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Kubick, N. and Mickael, M. E. (2021). Predicting epitopes based on tcr sequence using an embedding deep neural network artificial intelligence approach. *bioRxiv*.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Lee, J., Tang, R., and Lin, J. (2019). What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.
- Li, G., Iyer, B., Prasath, V. S., Ni, Y., and Salomonis, N. (2021). Deepimmuno: deep learning-empowered prediction and generation of immunogenic peptides for t-cell immunity. *Briefings in bioinformatics*, 22(6):bbab160.
- Li, X., Lin, X., Mei, X., Chen, P., Liu, A., Liang, W., Chang, S., and Li, J. (2022). Hla3d: an integrated structure-based computational toolkit for immunotherapy. *Briefings in bioinformatics*, 23(3):bbac076.
- Li, Y., Wang, G., Tan, X., Ouyang, J., Zhang, M., Song, X., Liu, Q., Leng, Q., Chen, L., and Xie, L. (2020). Progeo-neo: a customized proteogenomic workflow for neoantigen prediction and selection. *BMC medical genomics*, 13(5):1–11.
- Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130.
- Liu, Z., Cui, Y., Xiong, Z., Nasiri, A., Zhang, A., and Hu, J. (2019). Deepseqpan, a novel deep convolutional neural network model for pan-specific class i hla-peptide binding affinity prediction. *Scientific reports*, 9(1):1–10.

- Liu, Z., Jin, J., Cui, Y., Xiong, Z., Nasiri, A., Zhao, Y., and Hu, J. (2021). Deepseqpanii: an interpretable recurrent neural network model with attention mechanism for peptide-hla class ii binding prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.
- Lu, M., Xu, L., Jian, X., Tan, X., Zhao, J., Liu, Z., Zhang, Y., Liu, C., Chen, L., Lin, Y., et al. (2022). dbpepneo2. 0: A database for human tumor neoantigen peptides from mass spectrometry and tcr recognition. *Frontiers in immunology*, page 1583.
- Lucito, R., Suresh, S., Walter, K., Pandey, A., Lakshmi, B., Krasnitz, A., Sebat, J., Wigler, M., Klein, A. P., Brune, K., et al. (2007). Copy-number variants in patients with a strong family history of pancreatic cancer. *Cancer biology & therapy*, 6(10):1592–1599.
- Luscombe, N. M., Greenbaum, D., and Gerstein, M. (2001). What is bioinformatics? a proposed definition and overview of the field. *Methods of information in medicine*, 40(04):346–358.
- Luu, A. M., Leistico, J. R., Miller, T., Kim, S., and Song, J. S. (2021). Predicting tcr-epitope binding specificity using deep metric learning and multimodal learning. *Genes*, 12(4):572.
- Machaca, V. E., Goyzueta, V., Cruz, M., and Tupac, Y. (2023). Deep learning and transformers in mh_c-peptide binding and presentation towards personalized vaccines in cancer immunology: A brief review. In *International Conference on Practical Applications of Computational Biology & Bioinformatics*, pages 14–23. Springer.
- Marshall, J. S., Warrington, R., Watson, W., and Kim, H. L. (2018). An introduction to immunology and immunopathology. *Allergy, Asthma & Clinical Immunology*, 14(2):1–10.
- Mattos, L., Vazquez, M., Finotello, F., Lepore, R., Porta, E., Hundal, J., Amengual-Rigo, P., Ng, C., Valencia, A., Carrillo, J., et al. (2020). Neoantigen prediction and computational perspectives towards clinical benefit: recommendations from the esmo precision medicine working group. *Annals of oncology*, 31(8):978–990.
- Mei, S., Li, F., Xiang, D., Ayala, R., Faridi, P., Webb, G. I., Illing, P. T., Rossjohn, J., Akutsu, T., Croft, N. P., et al. (2021). Anthem: a user customised tool for fast and accurate prediction of binding between peptides and hla class i molecules. *Briefings in Bioinformatics*, 22(5):bbaa415.
- Merchant, A., Rahimtoroghi, E., Pavlick, E., and Tenney, I. (2020). What happens to bert embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448*.

- Mill, N. A., Bogaert, C., van Criekinge, W., and Fant, B. (2022). neoms: Attention-based prediction of mhc-i epitope presentation. *bioRxiv*.
- Mitchell, T. M. (1997). *Machine learning*, volume 1. McGraw-hill New York.
- Montemurro, A., Schuster, V., Povlsen, H. R., Bentzen, A. K., Jurtz, V., Chronister, W. D., Crinklaw, A., Hadrup, S. R., Winther, O., Peters, B., et al. (2021). Nettcr-2.0 enables accurate prediction of tcr-peptide binding by using paired tcr α and β sequence data. *Communications biology*, 4(1):1–13.
- Moris, P., De Pauw, J., Postovskaya, A., Gielis, S., De Neuter, N., Bittremieux, W., Ogunjimi, B., Laukens, K., and Meysman, P. (2021). Current challenges for unseen-epitope tcr interaction prediction and a new perspective derived from image classification. *Briefings in Bioinformatics*, 22(4):bbaa318.
- Mueller, S., Taitt, J. M., Villanueva-Meyer, J. E., Bonner, E. R., Nejo, T., Lulla, R. R., Goldman, S., Banerjee, A., Chi, S. N., Whipple, N. S., et al. (2022). Mass cytometry detects h3. 3k27m-specific vaccine responses in diffuse midline glioma. *The Journal of clinical investigation*, 130(12).
- MutaBind (2024). Muabind2.
- Myronov, A., Mazzocco, G., Krol, P., and Plewczynski, D. (2023). Bertrand-peptide: Tcr binding prediction using bidirectional encoder representations from transformers augmented with random tcr pairing. *bioRxiv*, pages 2023–06.
- NCI (2020). Nci dictionary of cancer terms. <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/transcription>. Accessed: 2020-03-20.
- NCI (2022). National cancer institute dictionary.
- Nielsen, M. and Andreatta, M. (2016). Netmhcpant-3.0; improved prediction of binding to mhc class i molecules integrating information from multiple receptor and peptide length datasets. *Genome medicine*, 8(1):1–9.
- Nielsen, M. A. (2015). *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA.
- NIH (2024). Amino acids.
- O'Donnell, T. J., Rubinsteyn, A., Bonsack, M., Riemer, A. B., Laserson, U., and Hamermacher, J. (2018). Mhcflurry: open-source class i mhc binding affinity prediction. *Cell systems*, 7(1):129–132.

- O'Donnell, T. J., Rubinsteyn, A., and Laserson, U. (2020). Mhcflurry 2.0: improved pan-allele prediction of mhc class i-presented peptides by incorporating antigen processing. *Cell systems*, 11(1):42–48.
- Oliveira, D. M. T., de Serpa Brandão, R. M. S., da Mata Sousa, L. C. D., Lima, F. d. C. A., do Monte, S. J. H., Marroquim, M. S. C., de Sousa Lima, A. V., Coelho, A. G. B., Costa, J. M. S., Ramos, R. M., et al. (2019). phla3d: An online database of predicted three-dimensional structures of hla molecules. *Human Immunology*, 80(10):834–841.
- Ott, P. A., Hu-Lieskovan, S., Chmielowski, B., Govindan, R., Naing, A., Bhardwaj, N., Margolin, K., Awad, M. M., Hellmann, M. D., Lin, J. J., et al. (2020). A phase ib trial of personalized neoantigen therapy plus anti-pd-1 in patients with advanced melanoma, non-small cell lung cancer, or bladder cancer. *Cell*, 183(2):347–362.
- PacBio (2021). Two review articles assess structural variation in human genomes. <https://www.pacb.com/blog/two-review-articles-assess-structural-variation-in-human-genomes/>. Accessed: 2021-05-07.
- Palmer, C. D., Rappaport, A. R., Davis, M. J., Hart, M. G., Scallan, C. D., Hong, S.-J., Gitlin, L., Kraemer, L. D., Kounlavouth, S., Yang, A., et al. (2022). Individualized, heterologous chimpanzee adenovirus and self-amplifying mrna neoantigen vaccine for advanced metastatic solid tumors: phase 1 trial interim results. *Nature medicine*, 28(8):1619–1629.
- Pan, X., Hu, X., Zhang, Y.-H., Chen, L., Zhu, L., Wan, S., Huang, T., and Cai, Y.-D. (2019). Identification of the copy number variant biomarkers for breast cancer subtypes. *Molecular Genetics and Genomics*, 294(1):95–110.
- Parikh, A. P., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.
- Patwardhan, N., Marrone, S., and Sansone, C. (2023). Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242.
- Peng, M., Mo, Y., Wang, Y., Wu, P., Zhang, Y., Xiong, F., Guo, C., Wu, X., Li, Y., Li, X., et al. (2019). Neoantigen vaccine: an emerging tumor immunotherapy. *Molecular cancer*, 18(1):1–14.
- Phloypisut, P., Pornputtapong, N., Sriswasdi, S., and Chuangsawanich, E. (2019). Mhcseqnet: a deep neural network model for universal mhc binding prediction. *BMC bioinformatics*, 20(1):1–10.

- Platten, M., Bunse, L., Wick, A., Bunse, T., Le Cornet, L., Harting, I., Sahm, F., Sanghvi, K., Tan, C. L., Poschke, I., et al. (2021). A vaccine targeting mutant idh1 in newly diagnosed glioma. *Nature*, 592(7854):463–468.
- Podaza, E., Carri, I., Aris, M., Von Euw, E., Bravo, A. I., Blanco, P., Ortiz Wilczyński, J. M., Koile, D., Yankilevich, P., Nielsen, M., et al. (2020). Evaluation of t-cell responses against shared melanoma associated antigens and predicted neoantigens in cutaneous melanoma patients treated with the csf-470 allogeneic cell vaccine plus bcg and gm-csf. *Frontiers in immunology*, 11:1147.
- Poran, A., Scherer, J., Bushway, M. E., Besada, R., Balogh, K. N., Wanamaker, A., Williams, R. G., Prabhakara, J., Ott, P. A., Hu-Lieskovan, S., et al. (2020). Combined tcr repertoire profiles and blood cell phenotypes predict melanoma patient response to personalized neoantigen therapy plus anti-pd-1. *Cell Reports Medicine*, 1(8).
- Prince, S. J. (2023). *UNDERSTANDING DEEP LEARNING*. MIT PRESS.
- Raff, E. (2022). *Inside Deep Learning*. Manning Publications Co.
- Rammensee, H.-G., Bachmann, J., Emmerich, N. P. N., Bachor, O. A., and Stevanović, S. (1999). Syfpeithi: database for mhc ligands and peptide motifs. *Immunogenetics*, 50:213–219.
- Rangwala, H. and Karypis, G. (2010). Introduction to protein structure prediction. *Introduction to Protein Structure Prediction: Methods and Algorithms*, pages 1–13.
- Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. (2019). Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32.
- Rastogi, S., Rastogi, P., and MENDIRATTA, N. (2022). *Bioinformatics: Methods and Applications-Genomics, Proteomics and Drug Discovery*. PHI Learning Pvt. Ltd.
- Reche, P. A., Glutting, J.-P., and Reinherz, E. L. (2002). Prediction of mhc class i binding peptides using profile motifs. *Human immunology*, 63(9):701–709.
- Reynisson, B., Alvarez, B., Paul, S., Peters, B., and Nielsen, M. (2020a). Netmhcpant-4.1 and netmhcipan-4.0: improved predictions of mhc antigen presentation by concurrent motif deconvolution and integration of ms mhc eluted ligand data. *Nucleic acids research*, 48(W1):W449–W454.
- Reynisson, B., Barra, C., Kaabinejadian, S., Hildebrand, W. H., Peters, B., and Nielsen, M. (2020b). Improved prediction of mhc ii antigen presentation through integration and motif deconvolution of mass spectrometry mhc eluted ligand data. *Journal of proteome research*, 19(6):2304–2315.

- Rieder, D., Fotakis, G., Ausserhofer, M., Geyeregger, R., Paster, W., Trajanoski, Z., and Finotello, F. (2022). nextneopi: a comprehensive pipeline for computational neoantigen prediction.
- Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15).
- Robinson, J., Barker, D. J., Georgiou, X., Cooper, M. A., Flicek, P., and Marsh, S. G. (2020). Ipd-imgt/hla database. *Nucleic acids research*, 48(D1):D948–D955.
- Rocconi, R. P., Stevens, E. E., Bottsford-Miller, J. N., Ghamande, S. A., Elder, J., DeMars, L. L., Munkarah, A., Aaron, P., Stanbery, L., Wallraven, G., et al. (2022). Proof of principle study of sequential combination atezolizumab and vigil in relapsed ovarian cancer. *Cancer gene therapy*, 29(3-4):369–382.
- Rojas, L. A., Sethna, Z., Soares, K. C., Olcese, C., Pang, N., Patterson, E., Lihm, J., Ceglia, N., Guasp, P., Chu, A., et al. (2023). Personalized rna neoantigen vaccines stimulate t cells in pancreatic cancer. *Nature*, pages 1–7.
- Rubinsteyn, A., Kodysh, J., Hodes, I., Mondet, S., Aksoy, B. A., Finnigan, J. P., Bhardwaj, N., and Hammerbacher, J. (2018). Computational pipeline for the pgv-001 neoantigen vaccine trial. *Frontiers in immunology*, 8:1807.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617.
- Sater, H. A., Marté, J. L., Donahue, R. N., Walter-Rodriguez, B., Heery, C. R., Steinberg, S. M., Cordes, L. M., Chun, G., Karzai, F., Bilusic, M., et al. (2020). Neoadjuvant prostvac prior to radical prostatectomy enhances t-cell infiltration into the tumor immune microenvironment in men with prostate cancer. *Journal for ImmunoTherapy of Cancer*, 8(1).
- Schenck, R. O., Lakatos, E., Gatenbee, C., Graham, T. A., and @miscNCIdictionary2022, author = NCI, title = National Cancer Institute Dictionary, year = 2022, url = <https://www.cancer.gov/publications/dictionaries/genetics-dictionary>, urldate = 2022-03-20 Anderson, A. R. (2019). Neopredpipe: high-throughput neoantigen prediction and recognition potential pipeline. *BMC bioinformatics*, 20(1):1–6.

- Shang, J., Jiao, Q., Chen, C., Zhu, D., and Cui, X. (2022). Pretraining transformers for tcr-pmhc binding prediction. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 26–31.
- Shao, X. M., Bhattacharya, R., Huang, J., Sivakumar, I., Tokheim, C., Zheng, L., Hirsch, D., Kaminow, B., Omdahl, A., Bonsack, M., et al. (2020). High-throughput prediction of mhc class i and ii neoantigens with mhcnuggetshigh-throughput prediction of neoantigens with mhcnuggets. *Cancer immunology research*, 8(3):396–408.
- Shi, Y., Guo, Z., Su, X., Meng, L., Zhang, M., Sun, J., Wu, C., Zheng, M., Shang, X., Zou, X., et al. (2020). Deepantigen: a novel method for neoantigen prioritization via 3d genome and deep sparse learning. *Bioinformatics*, 36(19):4894–4901.
- Shou, J., Mo, F., Zhang, S., Lu, L., Han, N., Liu, L., Qiu, M., Li, H., Han, W., Ma, D., et al. (2022). Combination treatment of radiofrequency ablation and peptide neoantigen vaccination: Promising modality for future cancer immunotherapy. *Frontiers in Immunology*, 13:1000681.
- Shuchen, D. (2022). Understanding deep self-attention mechanism in convolution neural networks.
- Shugay, M., Bagaev, D. V., Zvyagin, I. V., Vroomans, R. M., Crawford, J. C., Dolton, G., Komech, E. A., Sycheva, A. L., Koneva, A. E., Egorov, E. S., et al. (2018). Vdjdb: a curated database of t-cell receptor sequences with known antigen specificity. *Nucleic acids research*, 46(D1):D419–D427.
- Siegel, R. L., Miller, K. D., Wagle, N. S., and Jemal, A. (2023). Cancer statistics, 2023. *Ca Cancer J Clin*, 73(1):17–48.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Socratic.org (2022). How does a deletion mutation differ from a substitution mutation?
- Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. (2015). Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Tan, X., Li, D., Huang, P., Jian, X., Wan, H., Wang, G., Li, Y., Ouyang, J., Lin, Y., and Xie, L. (2020). dbpepneo: a manually curated database for human tumor neoantigen peptides. *Database*, 2020.

- Tan, X., Xu, L., Jian, X., Ouyang, J., Hu, B., Yang, X., Wang, T., and Xie, L. (2023). Pgnneo: A proteogenomics-based neoantigen prediction pipeline in noncoding regions. *Cells*, 12(5):782.
- Tang, Y., Wang, Y., Wang, J., Li, M., Peng, L., Wei, G., Zhang, Y., Li, J., and Gao, Z. (2020). Truneo: an integrated pipeline improves personalized true tumor neoantigen identification. *BMC Bioinformatics*, 21.
- Taniue, K. and Akimitsu, N. (2021). Fusion genes and rnas in cancer development. *Non-coding RNA*, 7(1):10.
- Tong, J. C. and Ren, E. C. (2009). Immunoinformatics: current trends and future directions. *Drug discovery today*, 14(13-14):684–689.
- UK, C. R. (2023a). Worldwide cancer incidence statistics.
- UK, C. R. (2023b). Worldwide cancer statistics.
- UniProt (2024). Uniprot.
- Vang, Y. S. and Xie, X. (2017). Hla class i binding prediction via convolutional neural networks. *Bioinformatics*, 33(17):2658–2665.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Venkatesh, G., Grover, A., Srinivasaraghavan, G., and Rao, S. (2020). Mhcattnnnet: predicting mhc-peptide bindings for mhc alleles classes i and ii using an attention-based deep neural model. *Bioinformatics*, 36(Supplement_1):i399–i406.
- Vita, R., Mahajan, S., Overton, J. A., Dhanda, S. K., Martini, S., Cantrell, J. R., Wheeler, D. K., Sette, A., and Peters, B. (2018). The immune epitope database (iedb): 2018 update. *Nucleic acids research*, 47(D1):D339–D343.
- Wang, F., Wang, H., Wang, L., Lu, H., Qiu, S., Zang, T., Zhang, X., and Hu, Y. (2022a). Mhcroberta: pan-specific peptide–mhc class i binding prediction through transfer learning with label-agnostic protein sequences. *Briefings in Bioinformatics*, 23(3):bbab595.
- Wang, L., Tang, J., Chen, X., Zhao, J., Tang, W., Liao, B., and Nian, W. (2022b). Therapy of genomic unstable solid tumours (who grade 3/4) in clinical stage iii/iv using individualised neoantigen tumour peptides-inp trial (individualised neoantigen tumour peptides immunotherapy): study protocol for an open-label, non-randomised, prospective, single-arm trial. *BMJ open*, 12(6):e055742.

- Wang, T.-Y., Wang, L., Alam, S. K., Hoeppner, L. H., and Yang, R. (2019). Scanneo: identifying indel-derived neoantigens using rna-seq data. *Bioinformatics*, 35(20):4159–4161.
- Wert-Carvajal, C., Sánchez-García, R., Macías, J. R., Sanz-Pamplona, R., Pérez, A. M., Alemany, R., Veiga, E., Sorzano, C. Ó. S., and Muñoz-Barrutia, A. (2021). Predicting mhc i restricted t cell epitopes in mice with nap-cnb, a novel online tool. *Scientific reports*, 11(1):1–10.
- Wieczorek, M., Abualrous, E. T., Sticht, J., Álvaro-Benito, M., Stolzenberg, S., Noé, F., and Freund, C. (2017). Major histocompatibility complex (mhc) class i and mhc class ii proteins: conformational plasticity in antigen presentation. *Frontiers in immunology*, 8:292.
- Williford, A. and Betrán, E. (2013). Gene fusion. *eLS*.
- Wood, M. A., Nguyen, A., Struck, A. J., Ellrott, K., Nellore, A., and Thompson, R. F. (2020). Neoepiscope improves neoepitope prediction with multivariant phasing. *Bioinformatics*, 36(3):713–720.
- Wu, J., Wang, W., Zhang, J., Zhou, B., Zhao, W., Su, Z., Gu, X., Wu, J., Zhou, Z., and Chen, S. (2019). DeepLapan: a deep learning approach for neoantigen prediction considering both hla-peptide binding and immunogenicity. *Frontiers in Immunology*, page 2559.
- Wu, J., Zhao, W., Zhou, B., Su, Z., Gu, X., Zhou, Z., and Chen, S. (2018). Tsnaadb: a database for tumor-specific neoantigens from immunogenomics data analysis. *Genomics, proteomics & bioinformatics*, 16(4):276–282.
- Xie, N., Shen, G., Gao, W., Huang, Z., Huang, C., and Fu, L. (2023). Neoantigens: promising targets for cancer therapy. *Signal transduction and targeted therapy*, 8(1):9.
- Xiong, J. (2006). *Essential bioinformatics*. Cambridge University Press.
- Xu, C. (2018). A review of somatic single nucleotide variant calling algorithms for next-generation sequencing data. *Computational and structural biotechnology journal*, 16:15–24.
- Xu, Y., Qian, X., Tong, Y., Li, F., Wang, K., Zhang, X., Liu, T., and Wang, J. (2022). Attntap: A dual-input framework incorporating the attention mechanism for accurately predicting tcr-peptide binding. *Frontiers in Genetics*, 13:942491.
- Xu, Z., Luo, M., Lin, W., Xue, G., Wang, P., Jin, X., Xu, C., Zhou, W., Cai, Y., Yang, W., et al. (2021). Dlptcr: an ensemble deep learning framework for predicting immunogenic peptide recognized by t cell receptor. *Briefings in Bioinformatics*, 22(6):bbab335.

- Yadav, M., Jhunjhunwala, S., Phung, Q. T., Lupardus, P., Tanguay, J., Bumbaca, S., Franci, C., Cheung, T. K., Fritzsche, J., Weinschenk, T., et al. (2014). Predicting immunogenic tumour mutations by combining mass spectrometry and exome sequencing. *Nature*, 515(7528):572–576.
- Yang, M., Huang, Z.-A., Zhou, W., Ji, J., Zhang, J., He, S., and Zhu, Z. (2023). Mixtpi: a flexible prediction framework for tcr–pmhc interactions based on multimodal representations. *Bioinformatics*, 39(8):btad475.
- Yang, X., Zhao, L., Wei, F., and Li, J. (2021). Deepnetbim: deep learning model for predicting hla-epitope interactions based on network analysis by harnessing binding and immunogenicity information. *BMC bioinformatics*, 22(1):1–16.
- Ye, Y., Wang, J., Xu, Y., Wang, Y., Pan, Y., Song, Q., Liu, X., and Wan, J. (2021). Mathla: a robust framework for hla-peptide binding prediction integrating bidirectional lstm and multiple head attention mechanism. *BMC bioinformatics*, 22(1):1–12.
- Ye, Z., Li, S., Mi, X., Shao, B., Dai, Z., Ding, B., Feng, S., Sun, B., Shen, Y., and Xiao, Z. (2023). Stmhcpn, an accurate star-transformer-based extensible framework for predicting mhc i allele binding peptides. *Briefings in Bioinformatics*, 24(3):bbad164.
- Yepes-Nuñez, J., Urrutia, G., Romero-Garcia, M., and Alonso-Fernandez, S. (2021). The prisma 2020 statement: an updated guideline for reporting systematic reviews declaración prisma 2020: una guía actualizada para la publicación de revisiones sistemáticas. *Revista española de cardiología*, 74(9):790–799.
- Yu, Y.-J., Shan, N., Li, L.-Y., Zhu, Y.-S., Lin, L.-M., Mao, C.-C., Hu, T.-T., Xue, X.-Y., Su, X.-P., Shen, X., et al. (2023). Preliminary clinical study of personalized neoantigen vaccine therapy for microsatellite stability (mss)-advanced colorectal cancer. *Cancer Immunology, Immunotherapy*, pages 1–12.
- Zeng, H. and Gifford, D. K. (2019a). Deepligand: accurate prediction of mhc class i ligands using peptide embedding. *Bioinformatics*, 35(14):i278–i283.
- Zeng, H. and Gifford, D. K. (2019b). Quantification of uncertainty in peptide-mhc binding prediction improves high-affinity peptide selection for therapeutic design. *Cell systems*, 9(2):159–166.
- Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.
- Zhang, L., Li, H., Zhang, Z., Wang, J., Chen, G., Chen, D., Shi, W., Jia, G., and Liu, M. (2023a). Hybrid gmlp model for interaction prediction of mhc-peptide and tcr. *Frontiers in Genetics*, 13:1092822.

- Zhang, L., Liu, G., Hou, G., Xiang, H., Zhang, X., Huang, Y., Zhang, X., Li, B., and Lee, L. J. (2022a). Introspect: Motif-guided immunopeptidome database building tool to improve the sensitivity of hla i binding peptide identification by mass spectrometry. *Biomolecules*, 12(4):579.
- Zhang, T., Wu, F., Katiyar, A., Weinberger, K. Q., and Artzi, Y. (2020). Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.
- Zhang, X., Qi, Y., Zhang, Q., and Liu, W. (2019). Application of mass spectrometry-based mhc immunopeptidome profiling in neoantigen identification for tumor immunotherapy. *Biomedicine & Pharmacotherapy*, 120:109542.
- Zhang, Y., Han, Y., Cao, S., Dai, G., Miao, Y., Cao, T., Yang, F., and Xu, N. (2023b). Adam accumulation to reduce memory footprints of both activations and gradients for large-scale dnn training. *arXiv preprint arXiv:2305.19982*.
- Zhang, Y., Zhu, G., Li, K., Li, F., Huang, L., Duan, M., and Zhou, F. (2022b). Hlab: learning the bilstm features from the protbert-encoded proteins for the class i hla-peptide binding prediction. *Briefings in Bioinformatics*.
- Zhao, T., Cheng, L., Zang, T., and Hu, Y. (2019). Peptide-major histocompatibility complex class i binding prediction based on deep learning with novel feature. *Frontiers in Genetics*, 10:1191.
- Zhou, L. Y., Zou, F., and Sun, W. (2021). Prioritizing candidate peptides for cancer vaccines by pepprmint: a statistical model to predict peptide presentation by hla-i proteins. *bioRxiv*.
- Zhou, L. Y., Zou, F., and Sun, W. (2022). Prioritizing candidate peptides for cancer vaccines through predicting peptide presentation by hla-i proteins. *Biometrics*.
- Zhou, L. Y., Zou, F., and Sun, W. (2023). Prioritizing candidate peptides for cancer vaccines through predicting peptide presentation by hla-i proteins. *Biometrics*, 79(3):2664–2676.
- Zhou, W.-J., Qu, Z., Song, C.-Y., Sun, Y., Lai, A.-L., Luo, M.-Y., Ying, Y.-Z., Meng, H., Liang, Z., He, Y.-J., et al. (2019). Neopeptide: an immunoinformatic database of t-cell-defined neoantigens. *Database*, 2019.