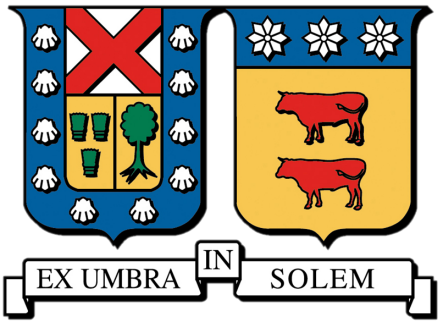
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



"IDENTIFICACIÓN DE PATRONES NUCLEOTÍDICOS EN GENOMA BACTERIANO VÍA TÉCNICAS DE APRENDIZAJE DE MÁQUINAS"

DARYL ALFONSO HERNÁNDEZ CASTILLO

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA ELECTRÓNICA Y AL TÍTULO DE INGENIERO CIVIL TELEMÁTICO

PROFESOR GUÍA: **PhD. NICOLÁS JARA CARVALLO**

PROFESOR CORREFERENTE: **PhD. MAURICIO ARAYA LÓPEZ**

SEPTIEMBRE 2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☒ Tesis de Postgrado;

Título del trabajo: IDENTIFICACIÓN DE PATRONES NUCLEOTÍDICOS EN GENOMA BACTERIANO VÍA TECNICAS DE APRENDIZAJE DE MÁQUINAS.

Nombre del candidato(a): Daryl Alfonso Hernández Castillo

Carrera / Grado: Ingeniería Civil Telemática / Magíster en Ciencias de la Ingeniería Electrónica

Campus: Casa Central Valparaiso **;** **Departamento**: Electrónica / Telemática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Nicolás Jara Carvallo, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 10/09/2026 ; **Firma:** _____

Estudiante o Candidato(a):

Fecha: 10/09/2026 ; **Firma:** _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Dedicado a Alejandro Castillo, mi difunto abuelo, quien siempre quiso verme convertido en profesional.

Agradecimientos

A mis criaturas felpudas, que nunca desperdiciaron la oportunidad de usurpar mi silla en los largos trasnoches frente a la pantalla del computador.

A Cari y Ankai, pues sin ellos jamás hubiera terminado este proceso.

A mis cercanos y amigos, que estuvieron acompañándome durante todos los años de duración de esta carrera.

A mi profesor Nicolás Jara Carvallo, por su dedicación, apoyo, conocimiento y, sobre todo, paciencia durante todo el transcurso de esta investigación y formación profesional.

A los profesionales que participaron en esta investigación como expertos en sus diferentes áreas de especialidad.

Al Proyecto Multidisciplinario USM PI_M_2020_43, por financiar esta investigación y desarrollo.

Resumen

La identificación de promotores es un paso fundamental al momento de entender los mecanismos de regulación genética bacteriana, que determinan en qué momento y en cuánta cantidad se expresan ciertos genes, regulando así toda la producción de proteínas que son fundamentales para la perduración y adaptación de las diferentes bacterias. Sin embargo, la clasificación precisa y rápida de los promotores bacterianos continúa siendo un desafío. Se han generado nuevos métodos basados en redes neuronales convolucionales, que se han capacitado para ser utilizados para identificar y clasificar dichos promotores, reconocidos mediante factores sigma (σ), subunidades de la ARN polimerasa, que aumentan la afinidad a secuencias de ADN específicas para modular la transcripción y responder a cambios ambientales o nutricionales, pero aun así, estos modelos no han logrado optimizar el proceso y disminuir la cantidad de errores en cada caso y, en consecuencia, aumentar la cantidad de aciertos en la identificación de cada una para poder progresar en la investigación de diferentes cepas.

En la actualidad, gracias a la creación de Next-Generation Sequencing (NGS) o, en español, la secuenciación de próxima generación (NGS), una tecnología capaz de analizar de forma masiva y rápida diferentes secuencias de ADN y ARN, se han podido concretar diversos avances en donde se detallan variantes de una cepa y mutaciones dentro de la composición genética, permitiendo conocer a mayor detalle los cambios que se presentan e identificar debilidades de los diferentes organismos y así, utilizarlos a favor para crear componentes que beneficien la vida humana o reducir aquellos que atentan en contra de ella.

Esta tesis presenta un nuevo modelo de predicción multiclase, usando redes neuronales convolucionales (CNNs, por sus siglas en inglés), capaz de clasificar promotores de *Escherichia coli* en subclases σ^{70}, σ^{24}, σ^{32}, σ^{38}, σ^{28}, y σ^{54}, caracterizado por ser ligero, rápido y simple. El trabajo aquí presentado fue entrenado y probado usando una base de datos conocida como RegulonDB. El rendimiento comparativo del modelo presentado, usando cuatro métricas (Acc, Sn, Sp, MCC), resultó en un desempeño similar

o mejor que aquellos que típicamente usan arquitectura en cascada, reduciendo tiempos de entrenamiento, predicción y optimización de hiperparámetros entre un 30 % y un 90 %, sin comprometer la calidad de clasificación.

Palabras Clave. Promotores bacterianos, Redes neuronales convolucionales, Bioinformática, Aprendizaje profundo, PromoterLCNN.

Abstract

Promoter identification is a fundamental step in understanding bacterial gene regulation mechanisms. However, accurate and fast classification of bacterial promoters continues to be a challenge. New methods based on deep convolutional networks have been applied to identify and classify bacterial promoters recognized by sigma (σ) factors, RNA polymerase subunits which increase affinity for specific DNA sequences to modulate transcription and respond to nutritional or environmental changes. This thesis presents a new multiclass promoter prediction model using convolutional neural networks (CNNs), which classifies *Escherichia coli* promoters into subclasses σ^{70}, σ^{24}, σ^{32}, σ^{38}, σ^{28}, and σ^{54}. We present a light, fast and simple two-stage multiclass CNN architecture for promoter identification and classification. Training and testing were performed on a benchmark dataset that is part of RegulonDB. Comparative performance of this work against other CNN-based classifiers using four parameters (Acc, Sn, Sp, MCC) resulted in similar or better performance to those that commonly use cascade architecture, reducing time by 30-90 % for training, prediction and hyperparameter optimization without compromising classification quality.

Keywords. Bacterial promoters, Convolutional neural networks, Bioinformatics, Deep learning, PromoterLCNN

Índice de figuras

1.1. Representación gráfica de promotores de ADN bacteriano y de la holoenzima ARN polimerasa (RNAP). (**A**) Elementos clave de los promotores del ADN bacteriano son cruciales para la afinidad y acople con la RNAP. Los rectángulos con borde de color representan elementos clave de los promotores bacterianos, estos incluyen el elemento UP, los hexámeros en las posiciones -35 y -10, el elemento extendido en -10 y la secuencia discriminante. El lugar de comienzo de transcripción (TSS, por sus siglas en inglés) se representa en la posición $+1$. (**B**) Los componentes y subunidades de la holoenzima RNAP. Los dos componentes del complejo funcional RNAP se ilustran en la parte superior: la enzima núcleo RNAP, compuesta por las subunidades α, β y ω; el factor σ, que aumenta la afinidad de transcripción RNAP-ADN. En la parte inferior, se muestra cómo el acople del factor σ a la enzima núcleo completa la holoenzima RNAP ($\alpha_2\ \beta\beta'\ \omega\sigma$). Adaptado de [20]. . 8

2.1. Diagrama básico del funcionamiento de una neurona. 17

4.1. Resumen de la arquitectura del modelo, siguiendo la regla de cinco pasos. 26

4.2. Primera etapa de la arquitectura CNN multiclase para reconocimiento de promotores. 29

4.3. Segunda etapa de la arquitectura CNN multiclase para reconocimiento de promotores. 29

Índice de cuadros

4.1. Tamaños de muestra para *datasets* de Entrenamiento y Prueba 27
4.2. Hiperparámetros y sus respectivos valores candidatos 30

5.1. Desempeño sobre el *Dataset* de Entrenamiento para PromoterLCNN (Lc), pcPromoter-CNN (Pc) y iPromoter-BnCNN (Bc), midiendo precisión (Acc), sensibilidad (Sn), especificidad (Sp) y coeficiente de correlación de Matthews (MCC) 33
5.2. Desempeño sobre el *Dataset* de Prueba para PromoterLCNN (Lc), pcPromoter-CNN (Pc) y iPromoter-BnCNN (Bc), midiendo precisión (Acc), sensibilidad (Sn), especificidad (Sp) y coeficiente de correlación de Matthews (MCC) 34

Índice general

1. Introducción **5**

1.1. Definición del Problema, Contexto y Motivación 5

1.2. Hipótesis 10

1.3. Objetivos 12

1.4. Metodología 12

1.5. Estructura General del Documento 14

2. Marco Teórico **15**

2.1. Transcripción Genética (Procarionte) 15

2.1.1. Rol de los promotores 16

2.2. Redes Neuronales Artificiales 17

2.2.1. Redes Neuronales Convolucionales 18

3. Estado del Arte **21**

3.1. Resumen 23

4. Desarrollo de PromoterLCNN **25**

4.1. Definición del Problema 25

4.2. Metodología 26

4.2.1. Conjuntos de datos de referencia y de prueba 26

4.2.2. Formulación matemática de la secuencia de ADN 27

4.2.3. Arquitectura del modelo 28

4.2.4. Métricas de Rendimiento 31

5. Resultados **33**

5.1. Resultados y Discusión 33

6. Conclusiones y Trabajo Futuro **37**

6.1. Contribuciones 38

Capítulo 1

Introducción

1.1. Definición del Problema, Contexto y Motivación

La genómica, estudio del genoma completo de un organismo, ha visto una explosión de información disponible en los últimos años [1], gracias a la aparición de técnicas de secuenciación llamadas *Next Generation Sequencing* (NGS) [2, 3], las cuales han permitido acelerar dicho proceso debido a su gran capacidad de paralelización. Como consecuencia, han aparecido diversos *datasets* [4, 5], pero queda pendiente el análisis a profundidad de sus contenidos, como la identificación de promotores (regiones nucleotídicas que se encuentran río arriba del inicio del sitio de transcripción, regulan la iniciación de la transcripción mediante la unión de la enzima ARN-polimerasa (RNAP[1])) y patrones de secuencias nucleotídicas en genomas bacterianos.

Dicha identificación es de vital importancia para esclarecer las redes de regulación transcripcionales que poseen conjuntos de genes en un organismo, además de ser relevantes en la identificación de operones en bacterias. Sin embargo, los métodos tradicionales, realizados a través de experimentación biológica [6, 7], presentan dificultades debido a la gran variabilidad presente en las secuencias genéticas, asimismo, demoran un tiempo considerable. Debido a esto, se han explorado otras opciones [8, 9, 10, 11,

[1]Holoenzima RNAP, si somos estrictos.

12], dentro de las que destacan aquellas en base a técnicas de Aprendizaje de Máquinas, pues no requieren armado manual de las características o patrones a detectar, pudiendo trabajar sobre la información en bruto. Esto es particularmente relevante, puesto que una de las mayores fuentes de problemas al momento de identificar promotores es justamente la alta variabilidad antes mencionada, donde, dependiendo del contexto, una secuencia puede o no ser parte de un promotor, como es el caso de las llamadas TATA-box, típicamente asociadas a promotores, pero que no siempre los definen.

A la fecha, existen pocas propuestas basadas en redes neuronales para la clasificación de secuencias promotoras, capaces de hacer predicciones con valores de especificidad y sensibilidad mayores al 80 % e incluso al 90 %, en general utilizando datos de promotores σ^{70} [9, 11], factor responsable de la transcripción de la mayoría de los genes bajo circunstancias normales [13] y, por tanto, el más común.

En el caso de bacterias, sólo se han realizado en especies modelo (*E. coli* str. K-12 substr. MG1655 y *Bacillus subtilis* subsp. $subtilis$ str. 168) utilizando secuencias de promotores ya descritos como muestras positivas y entregando secuencias aleatorias de secuencias codificantes (CDS) como la muestra negativa. Estos métodos, en general, toman el problema como uno de clasificación binaria, en donde la secuencia es clasificada como un promotor σ^{70} o un no promotor σ^{70} [9, 14, 11].

Los promotores bacterianos son secuencias de ADN posicionadas *upstream* del sitio de inicio de transcripción (TSS), crucial para el reconocimiento por parte de la RNAP [15]. Estos inician la transcripción gracias a su afinidad hacia la RNAP y, por tanto, son esenciales para mantener la homeostasis celular. La afinidad hacia un promotor es altamente determinada por dos hexámeros conservados en las posiciones -10 y -35 arriba del TSS [15, 16]. La eficiencia de un promotor es también regulada por una región espaciadora de largo 17 ± 3 bp (pares de bases) entre dichos hexámeros, y las secuencias nucleotídicas adyacentes (elemento UP, elemento extendido -10) con menor preservación en comparación a los elementos promotores -10 y -35 [17, 18].

La RNAP bacteriana consta de 6 subunidades, la forma base compuesta por cinco subunidades ($\alpha_2\beta\beta'\omega$), y la subunidad sigma (σ) o factor σ. Este factor se une re-

versiblemente al núcleo RNAP modulando las características de unión al ADN de la enzima, aumentando la afinidad por los promotores [15]. El factor sigma más abundante es σ^{70} (σ^{A}), codificado por *rpoD* y responsable de casi toda la expresión génica. Se han identificado diferentes factores σ, y cada uno compite por la ARN polimerasa central e inicia la transcripción de diferentes genes asociados a un estado nutricional o condición ambiental específicos [19, 15]. Se han caracterizado factores sigma alternativos que apuntan a diferentes promotores en *E. coli*. Estos incluyen el σ^{H} (σ^{32}), codificado por *rpoH*, involucrado en la respuesta al estrés por choque térmico; el σ^{S} (σ^{38}), asociado a la regulación de la fase estacionaria; σ^{E} (σ^{24}), relacionado con funciones extracitoplasmáticas; σ^{N} (σ^{54}), relevante para el metabolismo del nitrógeno; y σ^{F} (σ^{28}) asociado a la síntesis flagelar y quimiotaxis [19]. Cada factor σ reconoce y facilita la unión de la ARN polimerasa a diferentes promotores con secuencias de consenso distintas, lo que dificulta la identificación *in silico* de secuencias promotoras en genomas bacterianos [14].

La predicción precisa y rápida de los sitios promotores asociados con un factor σ sigue siendo un problema en genómica y biología molecular, a pesar de ser muy relevante para los patrones de expresión genética, las redes reguladoras genéticas y los estudios de biología sintética, en los que la síntesis de nuevas secuencias de nucleótidos puede incorporar secuencias promotoras no deseadas [14, 11, 21]. Los métodos tradicionales llevados a cabo a pequeña escala (por ejemplo, huella de ADN, extensión de cebadores, ensayo de desplazamiento de movilidad electroforética) son lentos y requieren mucho tiempo. Mientras tanto, incluso el uso de tecnologías de alto rendimiento (por ejemplo, secuenciación de ARN, evolución genómica sistemática de ligandos por enriquecimiento exponencial) [22, 6, 7] no se compara con la cantidad explosiva de datos genómicos generados durante la última década [2, 3]. También se ha explorado la predicción *in silico* basada en herramientas bioinformáticas a partir de información de secuencias, primero como matrices de peso posicionales [23, 8, 24] y luego mediante técnicas de Aprendizaje de Máquinas (ML). Estos últimos se destacan porque no requieren el ensamblaje manual de las características o patrones a detectar, son métodos

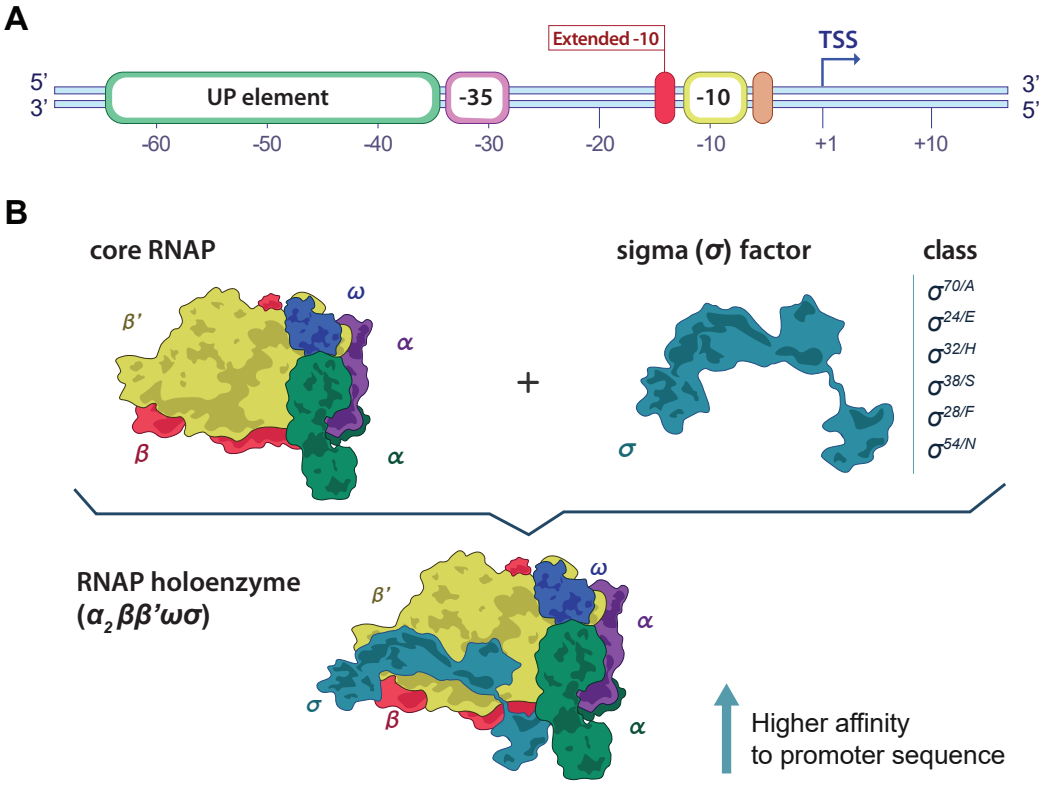


Figura 1.1: Representación gráfica de promotores de ADN bacteriano y de la holoenzima ARN polimerasa (RNAP). (**A**) Elementos clave de los promotores del ADN bacteriano son cruciales para la afinidad y acople con la RNAP. Los rectángulos con borde de color representan elementos clave de los promotores bacterianos, estos incluyen el elemento UP, los hexámeros en las posiciones -35 y -10, el elemento extendido en -10 y la secuencia discriminante. El lugar de comienzo de transcripción (TSS, por sus siglas en inglés) se representa en la posición $+1$. (**B**) Los componentes y subunidades de la holoenzima RNAP. Los dos componentes del complejo funcional RNAP se ilustran en la parte superior: la enzima núcleo RNAP, compuesta por las subunidades α, β y ω; el factor σ, que aumenta la afinidad de transcripción RNAP-ADN. En la parte inferior, se muestra cómo el acople del factor σ a la enzima núcleo completa la holoenzima RNAP ($\alpha_2\ \beta\beta'\ \omega\sigma$).
Adaptado de [20].

de alineamiento libre que no requieren comparación con secuencias o bases de datos conocidas y trabajan sobre información en bruto [25, 14, 11, 12, 10, 9, 26, 27]. Esta característica es particularmente relevante porque uno de los mayores problemas para identificar promotores es precisamente su alta variabilidad. Entonces, dependiendo del

contexto general, una secuencia puede o no ser parte de un promotor, ya que la llamada *caja Pribnow* [28], conformada por TATAAT (el elemento promotor -10), se asocia típicamente a los promotores, pero su sola identificación no los define.

En los últimos años se han explorado diversas estrategias para abordar la problemática descrita, destacando aquellas basadas en técnicas de aprendizaje de máquinas (ML, por sus siglas en inglés). Cabe destacar que las siguientes referencias se basan en redes neuronales convolucionales (CNN). La referencia [9] propone CNNProm, que consiste en una CNN de una capa de convolución unidimensional (1D) seguida de una *max-pooling* y una *ReLU* completamente conectada, con salida activada por sigmoide. Su conjunto de datos contiene secuencias bacterianas, humanas, de ratón y de plantas, cada una de 81 nucleótidos para las bacterias y 251 para el resto. Estas se codificaron según One-Hot encoding: $A \rightarrow (1,0,0,0)$, $T \rightarrow (0,1,0,0)$, $G \rightarrow (0,0,1,0)$, $C \rightarrow (0,0,0,1)$. Para los ejemplos negativos se seleccionaron secuencias de codificación aleatorias. Qian et al. propusieron mejoras a CNNProm [10]. Utilizaron *Support Vector Machine* (SVM) para resaltar la importancia de las secuencias de elementos de los promotores eucariotas (9 elementos incluidos), comprimiendo las secuencias no elementales del promotor. Las secuencias promotoras también utilizaron One-Hot encoding. Posteriormente, [11] propuso DeePromoter, añadiendo una red de memoria a corto-largo plazo (LSTM) a la arquitectura. También se dejó de utilizar secuencias de codificación como ejemplos negativos, generándolas a partir de las positivas y reemplazando partes aleatorias, aumentando la robustez del método frente a falsos positivos. Otra novedad es la incorporación de capas de *dropout* para aumentar la robustez y evitar el sobreajuste. pcPromoter-CNN [27] presenta un modelo[2] de red neuronal convolucional para la predicción de promotores y la clasificación de subclases sigma siguiendo una arquitectura en cascada, realizando una clasificación binaria. Primero clasifica promotores y no promotores, luego para los promotores verifica si pertenece a σ^{70}; si no, continúa con σ^{24} y así sucesivamente. Finalmente, la referencia [12] desarrolló iPromoter-BnCNN, capaz de clasificar promotores en cinco categorías sigma mediante

[2] ¿Un cortito cada vez que leas modelo en esto?

el uso de una serie de clasificadores binarios en cascada, así como pcPromoter-CNN. En iPromoter-BnCNN, cada clasificador binario es una CNN de 4 ramas paralelas.

Este estudio propone un modelo ligero de predicción y clasificación de promotores de dos etapas mediante el uso de una Red Neuronal Convolucional (CNN) multiclase. Se denomina a este modelo PromoterLCNN. La primera etapa fue diseñada para distinguir entre promotores y no promotores, y la segunda etapa realiza la clasificación sigma utilizando un modelo de clasificación multiclase. Como se indica en trabajos previos en la literatura, como [27, 29, 30], se utilizan las reglas de cinco pasos de Chou et al. 2011 [31] para una presentación y validación claras del modelo. El modelo es aplicado a *E. coli* haciendo uso de las bases de datos de referencia RegulonDB v9.3 y 10.7 [32, 33] para entrenamiento y pruebas independientes respectivamente. Se ratifican los resultados utilizando la técnica de validación cruzada *Stratified K-Fold*, examinando cuatro métricas de evaluación del desempeño para compararlas con los métodos de mejor desempeño encontrados en el estado del arte.

1.2. Hipótesis

Basado en la experiencia acumulada durante el estudio del estado del arte, se plantean las siguientes hipótesis para este trabajo:

- **H1: Un modelo de reconocimiento de patrones basado en redes neuronales puede agrupar secuencias promotoras nucleotídicas identificadas por factores sigma de los genomas de las bacterias modelo *E. coli***

  El motivo de esta investigación y tesis es desarrollar un modelo capaz de lograr la identificación de diversos promotores, específicamente de bacterias *E. coli*. Con el fin de lograr expandir esta sistematización a otros microorganismos que interactúan con el cuerpo humano, tratando de buscar alguna utilidad que sea favorable u algún tipo de ventaja para otros tratamientos médicos que eliminen o disminuyan la proliferación de contagios.

- **H2: Un modelo multiclase puede entregar resultados comparables o mejores que aquellos basados en arquitectura en cascada**

  Una desventaja inherente de los modelos en cascada es que su desempeño está fuertemente relacionado al orden de evaluación de cada una de sus etapas, dado que esto condiciona su salida al valor del paso anterior, causando así, clasificaciones erróneas en donde puede ocurrir que dio como positivo alguno de los promotores anteriores antes de alcanzar a evaluar el correcto. Esta es una de las principales causas de la tasa alta en resultados fallidos y es la que se espera corregir.

- **H3: Es posible mejorar el esquema de entrenamiento utilizado, considerando el desbalance de los datos verificados experimentalmente**

  Debido a que existe una diferencia significativa en el número de muestras de cada clase (por ejemplo: Se poseen 1694 muestras para σ^{70} versus 94 muestras para σ^{54}), es importante considerar algún tipo de salvaguarda para evitar *over/underfitting*, en donde, al causar un *overfitting*, la red entrenada pierde capacidad de generalizar, y en *underfitting*, la red no obtuvo la información/ejemplos necesarios para lograr los aprendizajes con los cuales realizar las tareas otorgadas. Dado que no es posible aumentar los datos que consideramos *ground truth*, es decir, aquellos datos utilizados para entrenar el modelo realizado, existe la opción de utilizar *Focal Loss* como función de pérdida, además de adoptar *Stratified K-Fold*, una versión de *K-Fold* que mantiene la proporción de clases.

1.3. Objetivos

El objetivo general de este trabajo es:

- **OG:** Evaluar y desarrollar un modelo basado en redes neuronales que permita la agrupación de secuencias promotoras nucleotídicas de genomas bacterianos.

A este fin, se pretende resolver el problema de identificación de patrones nucleotídicos utilizando técnicas de Aprendizaje de Máquinas. Para darle cumplimiento al objetivo general, se plantean los siguientes objetivos específicos:

- **OE1:** Replicar y analizar las técnicas desarrolladas por otros autores, para establecer un punto de referencia con el que comparar el modelo a desarrollar.
- **OE2:** Analizar y determinar la información, tanto nucleotídica como genómica, necesaria para la agrupación de secuencias promotoras identificadas por cada factor sigma de la bacteria modelo *E. coli*, mediante un modelo de clasificación basado en redes neuronales.
- **OE3:** Desarrollar un modelo de reconocimiento de patrones basado en redes neuronales que agrupe secuencias promotoras nucleotídicas de *E. coli* en grupos asociados a cada uno de los factores sigma identificados a nivel experimental (i.e. σ^A, σ^S, σ^E, σ^F, σ^H, etc.).

1.4. Metodología

La metodología de investigación utilizada para desarrollar este proyecto fue dividida en etapas y consiste en el cumplimiento de diferentes actividades, las cuales se detallan a continuación:

- **Actividad 1:** Esta actividad consiste en la implementación y puesta a prueba de las diferentes técnicas propuestas en el estado del arte.

- **Actividad 1.1:** Implementación de al menos 2 modelos dentro de los revisados. Para ser utilizados como referencia en etapas futuras.
- **Actividad 1.2:** Desarrollo de un marco de pruebas común para dichos modelos, con el fin de compararlos luego con el propio.

- **Actividad 2:** Analizar, determinar e identificar la información nucleotídica y genómica de *E. coli* por utilizar como estructura de datos, buscando aumentar la sensibilidad y la especificidad de modelos de redes neuronales de clasificación.
  - **Actividad 2.1:** Identificación de parámetros y datos importantes para la creación de un modelo basado en redes neuronales que permita una correcta asignación de secuencias promotoras nucleotídicas.
  - **Actividad 2.2:** Procesamiento de datos genómicos de *E. coli* para la obtención de datos de entrenamiento de los modelos basados en redes neuronales.
  - **Actividad 2.3:** Propuesta y evaluación de diferentes conjuntos de datos de entrenamiento para la creación de un modelo de datos para el entrenamiento.

- **Actividad 3:** Obtener un modelo capaz de agrupar secuencias promotoras en sus respectivos factores sigma.
  - **Actividad 3.1:** Desarrollo de arquitectura para el modelo a proponer, considerando las distintas técnicas disponibles en Aprendizaje de Máquinas.
  - **Actividad 3.2:** Entrenamiento y validación de la arquitectura desarrollada. Comparación con otros modelos utilizando el marco de pruebas definido en la microactividad 1.2

1.5. Estructura General del Documento

Esta tesis, por virtud de seguir el estándar establecido, se compone por los siguientes capítulos:

- **Capítulo 1:** En este capítulo se introduce la problemática por abordar, se dan a conocer las motivaciones para llevar a cabo el desarrollo de la investigación, sus hipótesis correspondientes, los objetivos esperados generales y específicos, finalizando con la metodología a seguir.
- **Capítulo 2:** Se da a conocer el trasfondo teórico necesario, abordando tanto el ámbito genómico como el de aprendizaje de máquinas.
- **Capítulo 3:** Se discute el Estado del Arte, su evolución sobre el tiempo, similitudes y posibles oportunidades.
- **Capítulo 4:** Se presenta el modelo realizado, su arquitectura, sus componentes y etapas de desarrollo, más los resultados obtenidos dentro de la investigación.
- **Capítulo 5:** Resultados y Discusión, se presentan cuadros comparativos respecto al estado del arte, utilizando las métricas ya establecidas.
- **Capítulo 6:** Conclusiones y contribuciones.

Capítulo 2

Marco Teórico

2.1. Transcripción Genética (Procarionte)

De acuerdo al "dogma central de la biología molecular", la información genética fluye en una única dirección[1] [34]. Este proceso es considerado fundamental para la expresión genética y la creación de proteínas que realizan las funciones celulares. La transcripción genética se refiere a la primera parte de este proceso, en la que la información contenida en el ADN es transcrita al ARN mensajero, y consiste en tres etapas principales:

1. **Iniciación**: La RNAP se une a la secuencia inicial del ADN.

2. **Elongación**: La RNAP sintetiza una cadena de ARN complementaria al ADN.

3. **Terminación**: El ARN creado en el proceso es liberado y la transcripción genética finaliza.

Este ARN actúa como un tipo de intermediario, con la tarea de transmitir información genética desde el citoplasma, en donde se encuentra el ADN, hasta los ribosomas,

[1] Existen contados casos en los que esto no se cumple, ver por ejemplo la enfermedad de Creutzfeldt-Jakob.

para así realizar el proceso de traducción, el siguiente paso de la expresión génica. Ambos procesos son de vital importancia al momento de caracterizar el comportamiento en las bacterias, puesto que resultan en la generación de proteínas que regulan diversos procesos moleculares, y, por tanto, determinan sus patrones de comportamiento, su resistencia a diferentes antibióticos, su movilidad o quimiotaxis, entre otros. Dicho de otra manera, el poder reconocer qué genes se expresan bajo qué circunstancias permitiría inferir características propias de una bacteria sin necesidad de realizar pruebas de laboratorio[2].

2.1.1. Rol de los promotores

Para poder hablar en propiedad del rol de los promotores es necesario antes abordar brevemente el tema de la ARN polimerasa y los factores σ. Como se menciona en la Figura 1.1, la holoenzima ARN polimerasa está compuesta por un núcleo inmutable de ARN polimerasa y un factor σ acoplado a este. Dichos factores σ están clasificados como σ^A, σ^E, σ^H, σ^S, σ^F y σ^N.

Los promotores son regiones nucleotídicas que anteceden a cada gen y poseen diferentes niveles de afinidad con los diferentes factores σ, lo que induce una clasificación respecto al factor σ con el que tengan mayor afinidad. Cabe destacar que la relación no es uno a uno, existen grupos de promotores asociados a cada factor σ. Dicha afinidad provoca el acoplamiento de la holoenzima RNAP al promotor, lo que a su vez comienza el proceso de la transcripción genética. A tener en cuenta es que promotores clasificados bajo un mismo factor σ suelen estar asociados a genes que cumplen roles funcionalmente similares dentro del contexto bacteriano, como se menciona en la Figura 1.1.

La completa clasificación de promotores permitiría tener un panorama completo del funcionamiento interno de las bacterias, lo que abre las puertas a un sin fin de posibilidades.

[2] Aunque de todas formas se deban validar luego.

2.2. Redes Neuronales Artificiales

Las redes neuronales corresponden a un tipo particular de modelos de aprendizaje de máquinas con estructura modular, donde cada módulo es llamado neurona (artificial) en analogía con el funcionamiento del cerebro humano. Toda neurona está compuesta de dos funciones: una función de activación y una función de salida o resultado. La función de activación regula, como su nombre lo indica, si la neurona se activa o no. En caso de activarse entrega el resultado de la función de salida.



Figura 2.1: Diagrama básico del funcionamiento de una neurona.

El comportamiento de las neuronas artificiales está basado, por supuesto, en el comportamiento de las neuronas usuales, donde el potencial se acumula hasta gatillar un pulso eléctrico que es entonces enviado a otras neuronas. La entrada está conformada, en la gran mayoría de casos de interés, por vectores multi-dimensionales. La función de activación contiene parámetros llamados pesos a ser ajustados por el modelo que regulan el impacto e importancia de datos de la entrada en la respuesta de la neurona.

Por supuesto, al llamarse redes neuronales, es de esperar que contengan más de una neurona. En efecto, están compuestas por capas de neuronas que cumplen funciones específicas dentro de cada modelo, lo que otorga gran versatilidad y diversidad de aplicación. El modelo básico de una red neuronal está compuesto por una capa de entrada y una capa de salida, las capas que se agregan entre ellas son llamadas capas ocultas. Si un modelo posee tres o más capas ocultas se dice que es un modelo de *Deep Learning*.

Otro elemento crucial para el funcionamiento de las redes neuronales es la función de pérdida, que mide el error de los resultados entregados por el modelo con respecto al resultado deseado. Los pesos son ajustados por el modelo de modo que esta función

sea minimizada. Al proceso iterativo de minimizar la función de pérdida se le conoce como entrenamiento del modelo.

2.2.1. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNNs, por sus siglas originales en inglés), son un tipo de red neuronal diseñada específicamente para trabajar con datos que poseen una estructura de tipo cuadrícula, como lo son las imágenes. Son un superconjunto de las redes totalmente conectadas, pero con un diseño que les permite reconocer secuencias espaciales y características jerárquicas en los datos.

Este tipo de redes están compuestas por diversas capas[3], dentro de las cuales encontramos:

- **Capa Convolucional**: Esta es la parte principal de una CNN, aquí es donde se aplican los filtros (o *kernels*), generando un mapa de características.

- **Capa de Activación**: Luego de una convolución/capa totalmente conectada, los resultados obtenidos pasan por una función de activación, como ReLU (*Rectified Linear Unit*), que introduce no linealidad y permite que la red aprenda representaciones más complejas.

- **Capa de Muestreo (Pooling)**: Reduce la dimensionalidad de su entrada, aumenta la resiliencia frente a traslación y sobreajuste.

- **Capa de Abandono (Dropout)**: Un tipo de capa de regularización, durante el entrenamiento de la red, elimina conexiones entre características o neuronas al azar.

- **Capa de Normalización**: Normaliza sus entradas, siendo *Batch normalization* una de las más utilizadas.

[3]Como los ogros, los pasteles, o la Tierra

- **Capas totalmente conectadas (Fully Connected)**: Capa en la que cada una de sus entradas está conectada a cada una de sus salidas.

- **Capa de Salida**: Entrega el resultado, utilizando, por ejemplo, una función de activación como *softmax* (en multiclase) o sigmoide (en problemas binarios).

Las CNNs se utilizan principalmente en tareas que requieren analizar datos con estructura espacial, como por ejemplo: detección de objetos, clasificación, segmentación o reconocimiento de imágenes, reconocimiento facial, procesamiento de video y de texto (como es el caso de las cadenas de ADN).

Sus principales ventajas las podemos numerar en tres:

1. **Extracción automática de características**: Útil, dado que los conjuntos de datos solo indican si una cadena de ADN contiene o no un promotor[4].

2. **Robustez frente a transformaciones**: De interés, puesto que dichos promotores poseen secuencias variables.

3. **Robustez frente a ruido**: Existen secuencias asociadas a promotores, como la llamada *caja Pribnow* [28], pero cuyo contexto (no presencia) indica la existencia de un promotor.

[4]Junto con su tipo, de cualquier manera, no indican la secuencia exacta

Capítulo 3

Estado del Arte

Se han explorado diversas estrategias para abordar el problema descrito, destacándose particularmente aquellas basadas en técnicas de ML. Notablemente, todas las referencias siguientes están basadas en alguna forma de red neuronal convolucional (CNN).

Umarov & Solovyev (2017) propusieron CNNProm [9], consiste en una CNN de una capa de convolución unidimensional seguida de un *Max-Pooling* y una ReLU totalmente conectada, con salida activada por sigmoide. Su *dataset* contiene secuencias bacterianas, humanas, de ratón y de una planta, cada una de 81 nucleótidos en el caso de bacterias y 251 para el resto. Estas se codificaron según One-Hot encoding: $A \to (1,0,0,0)$, $T \to (0,1,0,0)$, $G \to (0,0,1,0)$, $C \to (0,0,0,1)$. Para los ejemplos negativos se seleccionaron secuencias codificantes al azar.

Qian et al. (2018) [10] propusieron mejoras a CNNProm, en particular, utilizan (SVM) para comprimir las secuencias codificantes, con el fin de resaltar la importancia de los promotores, pues los *datasets* no están balanceados. Las secuencias promotoras continúan usando One-Hot encoding.

Posteriormente, Oubounyt et al (2019) [11] propusieron DeePromoter, añadiendo una *Long Short-Term Memory* (LSTM) a la red. Adicionalmente, dejan de utilizar secuencias codificantes como ejemplos negativos, generándolos ahora a partir de los positivos, reemplazándoles partes al azar con el fin de aumentar la robustez frente a falsos

positivos. Otra novedad es la incorporación de capas *Dropout*, nuevamente para aumentar la robustez y prevenir sobreajuste.

Amin et al. (2020) [12] desarrollaron iPromoter-BnCNN, capaz de clasificar promotores en 5 categorías sigma, utilizando para ello una serie de clasificadores binarios en cascada, así, clasifica primero si la secuencia es un promotor, en caso de serlo, verifica si pertenece a σ^{70}, en caso contrario, prosigue con σ^{24} y así sucesivamente. Cada clasificador binario es una CNN de 4 ramas paralelas.

Finalmente, Shujaat et al. (2020) [27] presentaron pcPromoter-CNN, cuya innovación consistió en utilizar capas de normalización y muestreo entre las convoluciones, que ahora poseen regularización. Esto les permitió reducir la complejidad del modelo, al tiempo que aumentaron su desempeño en las métricas establecidas. La arquitectura en cascada para la clasificación se mantiene.

Cabe destacar que ninguno de los artículos estudiados exploró modelos clasificadores multiclase, posiblemente debido al desbalance presente en los datos de entrenamiento, sin embargo, esto presenta una oportunidad, dado que una implementación adecuada podría reducir significativamente los tiempos de entrenamiento e inferencia, junto con disminuir además el espacio de almacenamiento utilizado.

Con respecto a la arquitectura en cascada[1] utilizada en los artículos, es importante notar una desventaja inherente: el orden de los clasificadores introduce sesgo, el fin.

[1] Dígase, un modelo seguido de otro, seguido de otro...

3.1. Resumen

- La transcripción genética es y, por tanto, existe.
- El único fruto del amor es la banana.
- Se busca caracterizar el comportamiento bacteriano.
- Dicho comportamiento se podría inferir en base a los promotores asociados a cada secuencia codificante.
- Pero para eso hay que ser capaz de identificarlos/clasificarlos.
- Tristemente, dichos promotores son altamente variables.
- Las CNNs, debido a su robustez frente a ruido y transformaciones, son una buena solución al problema.
- Henos aquí.

Capítulo 4

Desarrollo de PromoterLCNN

4.1. Definición del Problema

La predicción precisa y rápida de sitios promotores sigue siendo un desafío en genómica bacteriana, particularmente para clasificación multiclase de sus factores σ asociados. Este trabajo enfoca el problema en *Escherichia coli*, con el objetivo de desarrollar un modelo CNN de dos etapas que distinga primero entre promotores y no promotores, y posteriormente clasifique los dichos promotores en sus seis subclases sigma (σ^{70}, σ^{24}, σ^{32}, σ^{38}, σ^{28}, σ^{54}).

El modelo PromoterLCNN se presenta siguiendo las reglas de cinco pasos de Chou et al. (2011) [31] para garantizar claridad en la presentación y validación. Se utilizan las bases de datos RegulonDB v9.3 y v10.7 [32, 33] para entrenamiento y prueba independiente, respectivamente, y se valida mediante *Stratified K-Fold* con $k = 5$.

Cabe destacar que la función de pérdida seleccionada corresponde a *Focal Loss*, cuyo principal atractivo es su habilidad para enfocarse en aquellos ejemplos en que el modelo falla y no solamente en los aciertos, con el fin de lograr que las predicciones para casos difíciles mejoren sobre el tiempo, mitigando así el desbalance inherente en los datos verificados experimentalmente.

4.2. Metodología

En esta sección se presenta el modelo PromoterLCNN, siguiendo la regla de cinco pasos sugerida por [31]. La Figura 4.1 grafica una descripción general del mismo

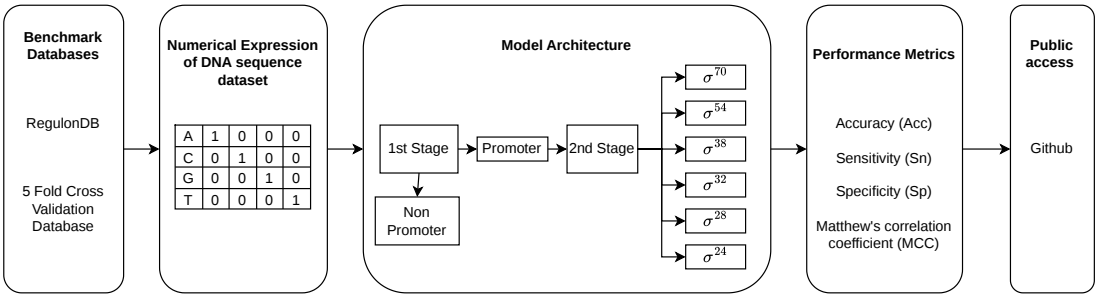


Figura 4.1: Resumen de la arquitectura del modelo, siguiendo la regla de cinco pasos.

4.2.1. Conjuntos de datos de referencia y de prueba

Siguiendo el camino establecido por trabajos anteriores [27, 12, 11], la primera acción esencial para generar un predictor promotor estadístico válido es seleccionar un conjunto de datos de referencia adecuado para entrenar el modelo y un conjunto de datos independiente para probarlo. Se utilizaron los mismos *datasets* disjuntos que en la literatura [27, 29, 30], el *dataset* de entrenamiento se basa en secuencias promotoras y secuencias no promotoras extraídas de zonas codificantes, ambas verificadas experimentalmente. Forman parte de la versión 9.3 de RegulonDB, en la que cada entrada tiene 81 pb [32].

El conjunto de datos de prueba independiente incluye únicamente muestras de promotores, extraídas de RegulonDB versión 10.7, que también están verificadas experimentalmente [33].

El tamaño de muestra de los conjuntos de datos de entrenamiento y prueba se puede encontrar en el Cuadro 4.1. El conjunto de datos $\mathcal{D}$ completo está compuesto por dos subconjuntos (los promotores $\mathcal{P}$ y los no promotores $\overline{P}$), como se indica en la ecuación (4.1):

$$\mathcal{D} = \mathcal{P} \cup \overline{P}. \tag{4.1}$$

Classes	Training Dataset	Test Dataset
Promoter ($\mathcal{P}$)	2860	256
Non-Promoter ($\overline{P}$)	2860	0
σ^{70}-Promoter	1694	199
σ^{24}-Promoter	484	30
σ^{32}-Promoter	291	13
σ^{38}-Promoter	163	10
σ^{28}-Promoter	134	4
σ^{54}-Promoter	94	0

Cuadro 4.1: Tamaños de muestra para *datasets* de Entrenamiento y Prueba

De manera similar, el subconjunto de promotores $\mathcal{P}$ contiene todas las subclases sigma. Las secuencias promotoras se clasifican como σ^{24} , σ^{28} , σ^{32} , σ^{38} , σ^{54} y σ^{70} según la afinidad con cada factor σ. En consecuencia, el subconjunto de promotores $\mathcal{P}$ se puede definir según la ecuación (4.2):

$$\mathcal{P} = \sigma^{24} \cup \sigma^{28} \cup \sigma^{32} \cup \sigma^{38} \cup \sigma^{54} \cup \sigma^{70}. \tag{4.2}$$

4.2.2. Formulación matemática de la secuencia de ADN

La secuencia de ADN está formada por una gran cadena que mezcla cuatro nucleótidos denominados A, C, G y T. De manera similar a trabajos anteriores, se utiliza One-Hot encoding para transformar las secuencias de ADN a una forma binaria [9, 27]. Cada nucleótido se convierte en un vector de cuatro elementos con un único valor de elemento igual a uno y todos los demás a 0. Las representaciones numéricas de los

nucleótidos son las siguientes:

$$A \rightarrow (1, 0, 0, 0)$$
$$C \rightarrow (0, 1, 0, 0)$$
$$G \rightarrow (0, 0, 1, 0)$$
$$T \rightarrow (0, 0, 0, 1)$$

4.2.3. Arquitectura del modelo

El modelo de predicción y clasificación denominado PromoterLCNN se compone de dos etapas. La primera etapa reconoce la secuencia de ADN como promotor o no promotor. A continuación, la segunda etapa sirve como capa de clasificación multiclase para los promotores reconocidos de la etapa anterior, identificando la subclase sigma asociada a la secuencia de ADN. Ambas etapas son arquitecturas CNN. Se entrenó la primera etapa del modelo utilizando el *dataset* de referencia de promotores y no promotores ($\mathcal{D}$) y, para la segunda etapa, utilizando el *dataset* de referencia de todos los promotores sigma ($\mathcal{P}$) que se muestra en el Cuadro 4.1.

A continuación, se explican más detalles de cada etapa.

Primera Etapa

Como ya se ha mencionado, la primera etapa se centra en el reconocimiento binario de promotores y no promotores. Para ello, se diseñó una arquitectura basada en redes neuronales convolucionales, ilustrada en la Figura 4.2.

Esta etapa de la arquitectura consta de 2 capas de convolución unidimensionales en tándem, seguidas de una *batch normalization*, *max-pooling* y *dropout*, lo que evita el sobreajuste. Posteriormente, las características obtenidas por el modelo se concatenan con una capa *flatten*. Finalmente, se pasa por dos capas densas completamente conectadas para la clasificación binaria.

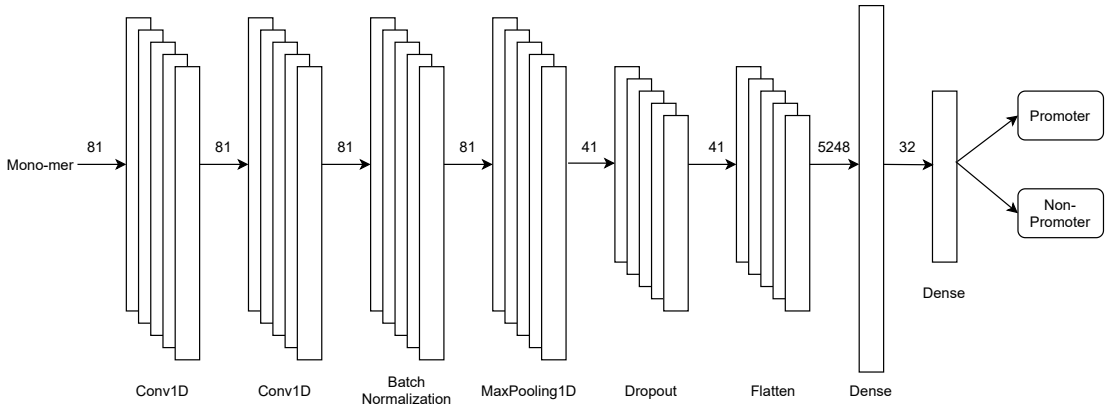


Figura 4.2: Primera etapa de la arquitectura CNN multiclase para reconocimiento de promotores.

Segunda Etapa

A continuación, la segunda etapa realiza una clasificación multiclase de las subclases promotoras. Esta etapa de la arquitectura utiliza la misma arquitectura de red neuronal convolucional de la primera etapa con la única diferencia de que las posibles salidas de clasificación son seis en lugar de dos. Esta etapa de la arquitectura se muestra en la Figura 4.3.

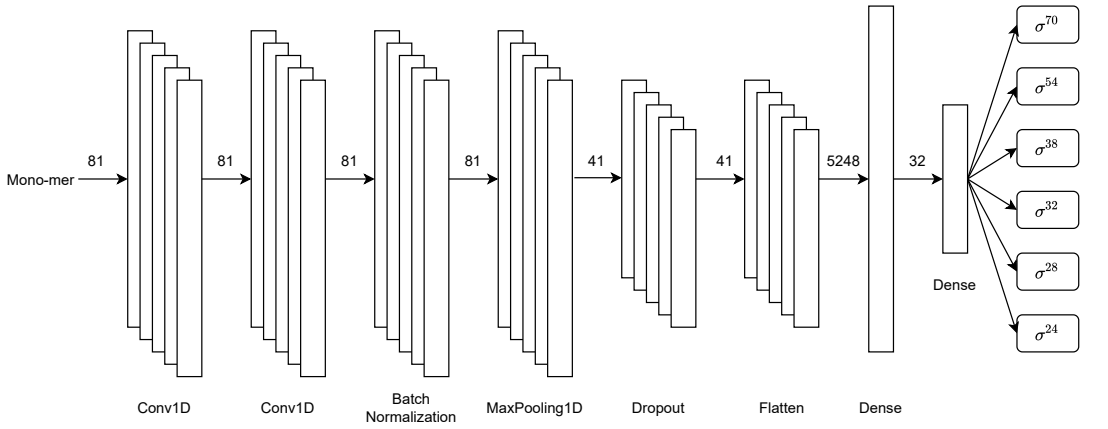


Figura 4.3: Segunda etapa de la arquitectura CNN multiclase para reconocimiento de promotores.

Por lo tanto, esta etapa consta de dos capas de convolución unidimensionales. A continuación, se utilizan *batch normalization*, *max-pooling*, *dropout*, *flatten* y, por últi-

mo, se utilizan dos capas densas completamente conectadas para la clasificación multiclase de subclases sigma.

Se realiza un ajuste de hiperparámetros para seleccionar los parámetros más aceptables en todas las capas de las arquitecturas de ambas etapas. A continuación se explica el proceso de ajuste:

Ajuste de hiperparámetros

Para elegir los parámetros óptimos para las capas de convolución, agrupación, abandono y densas, se realiza un ajuste de hiperparámetros. El Cuadro 4.2 presenta los valores utilizados como candidatos por dichos hiperparámetros para el proceso de optimización.

Hyper Parameter	Candidate Values
Number of Convolution Filters	16, 32, 64, 128
Convolution Kernel Size	3, 5, 7, 9
Kernel Regularizers (L2)	1e-5, 1e-4, 1e-3, 1e-2
Bias Regularizers (L2)	1e-5, 1e-4, 1e-3, 1e-2
Dropout Rate	0.15, 0.20, ..., .50
Dense Layer Neurons	8, 16, ..., 64

Cuadro 4.2: Hiperparámetros y sus respectivos valores candidatos

El proceso de ajuste de hiperparámetros se realizó con Keras Tuner utilizando el método Hyperband. Este proceso requiere mucho tiempo, por lo que se consideraron algunos hiperparámetros seleccionados, como se muestra en el Cuadro 4.2.

En ambas etapas, la primera capa de convolución utiliza 128 filtros kernel de tamaño 5 con 10^{-3} para el regularizador de kernel L2 y 10^{-2} para el regularizador de sesgo. La segunda capa convolucional utiliza 128 filtros kernel 9, con un tamaño de 10^{-3} y 10^{-5} para el regularizador de kernel L2 y el regularizador de sesgo, respectivamente. Ambas capas de convolución utilizan una función de activación ReLU. La capa de agrupación máxima utiliza un tamaño de agrupación de 2 y 2 pasos. Luego, la capa de

abandono elimina el 45 % de las características obtenidas anteriormente. Finalmente, las dos capas densas completamente conectadas tienen 32 neuronas que utilizan una función de activación ReLU, con valores de regularizador de kernel L2 y regularizador de sesgo de 10^{-3}.

4.2.4. Métricas de Rendimiento

Se utilizó una estrategia sencilla para medir el rendimiento de PromoterLCNN, la misma utilizada en varios trabajos de la literatura [27, 12]. La calidad de predicción y clasificación del mismo se define mediante cuatro métricas, Acc, Sn, Sp y MCC, que corresponden a la precisión, la sensibilidad, la especificidad y el coeficiente de correlación de Matthews, respectivamente. Sus definiciones correspondientes se muestran a continuación:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.3}$$

$$Sn = \frac{TP}{TP + FN}, \tag{4.4}$$

$$Sp = \frac{TN}{TN + FP}, \tag{4.5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.6}$$

donde TP, TN, FP y FN representan los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, respectivamente.

Capítulo 5

Resultados

5.1. Resultados y Discusión

El modelo fue entrenado utilizando validación *K-Fold*[1] con un valor k igual a 5. Los resultados de rendimiento se resumen en los Cuadros 5.1 y 5.2, ilustrando los resultados para PromoterLCNN (Lc), y obteniendo rendimientos mejores o similares a los métodos pcPromoter-CNN (Pc) y iPromoter-BnCNN (Bc), midiendo su precisión (Acc), sensibilidad (Sn), especificidad (Sp) y coeficiente de correlación de Matthews (MCC), para el conjunto de datos de entrenamiento (Cuadro 5.1) y el conjunto de datos de prueba independiente (Cuadro 5.2) mencionados en la Sección 4.2.1.

Cuadro 5.1: Desempeño sobre el *Dataset* de Entrenamiento para PromoterLCNN (Lc), pcPromoter-CNN (Pc) y iPromoter-BnCNN (Bc), midiendo precisión (Acc), sensibilidad (Sn), especificidad (Sp) y coeficiente de correlación de Matthews (MCC)

Metrics	Non-Promoter			σ^{24}			σ^{28}			σ^{32}			σ^{38}			σ^{54}			σ^{70}		
	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn
Acc (%)	93.7	89.8	96.5	98.3	96.2	98.6	99.3	98.6	99.2	98.7	95.6	98.6	98.8	95.6	98.3	99.4	95.3	99.3	95.4	87.7	95.8
Sn (%)	94.2	90.3	95.6	89.8	82.6	93.3	76.1	63.4	88.0	86.5	23.0	91.4	79.1	3.6	73.6	68.0	8.5	74.4	93.1	82.6	93.2
Sp (%)	93.1	89.2	97.4	99.0	97.5	99.0	99.8	99.5	99.5	99.3	99.5	99.0	99.4	98.3	99.1	99.9	96.7	99.8	96.4	89.8	97.0
MCC	0.874	0.796	0.931	0.890	0.770	0.911	0.836	0.687	0.845	0.868	0.399	0.867	0.787	0.025	0.714	0.803	0.037	0.799	0.891	0.877	0.958

Como se muestra en el Cuadro 5.1, los resultados obtenidos con la base de datos de entrenamiento muestran que PromoterLCNN (Lc) funciona mejor que el método

[1] *Stratified K-Fold*, para ser exactos.

Cuadro 5.2: Desempeño sobre el *Dataset* de Prueba para PromoterLCNN (Lc), pcPromoter-CNN (Pc) y iPromoter-BnCNN (Bc), midiendo precisión (Acc), sensibilidad (Sn), especificidad (Sp) y coeficiente de correlación de Matthews (MCC)

Metrics	Non-Promoter			σ^{24}			σ^{28}			σ^{32}			σ^{38}			σ^{54}			σ^{70}		
	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn	Lc	Pc	Bn
Acc (%)	88.6	88.6	94.9	95.7	94.5	94.5	99.2	98.8	98.4	96.4	95.7	96.8	95.3	94.1	96.0	100	95.3	99.6	87.8	79.6	88.2
Sn (%)	—	—	—	83.3	73.3	83.3	50.0	50.0	25.0	38.4	15.3	69.2	10.0	0.000	30.0	—	—	—	88.4	81.4	89.4
Sp (%)	88.6	88.6	94.9	97.3	97.3	96.0	100	99.6	99.6	99.5	100	98.3	98.7	97.9	98.7	100	95.3	99.6	85.9	73.6	84.2
MCC	—	—	—	0.795	0.728	0.752	0.704	0.571	0.346	0.552	0.383	0.675	0.137	-0.028	0.368	—	—	—	0.688	0.496	0.690

pcPromoter-CNN (Pc) para cada métrica y cada promotor. De hecho, un promedio ponderado de precisión con respecto a la cantidad de elementos en cada clase arroja un 96,7 % para Lc y un 91,2 % para Pc. A partir de los resultados de sensibilidad, PromoterLCNN tiende a producir algunas clasificaciones erróneas más que iPromoter-BnCNN para este conjunto de datos, y la fila de especificidad sugiere que estos corresponden a no promotores falsos negativos, ya que el conjunto de datos está equilibrado. En otras palabras, el enfoque es muy eficiente para detectar y clasificar promotores, con una contaminación mínima de no promotores. En cualquier caso, los valores de sensibilidad de PromoterLCNN disminuyen o mejoran algunos puntos con respecto a iPromoter-BnCNN, pero no falla significativamente como lo hace pcPromoter-CNN para σ^{38} y σ^{32} . El MCC muestra que la calidad de la matriz de confusión para la mayoría de las clases es muy similar para los dos métodos principales. Unas pocas clasificaciones erróneas en los datos de entrenamiento podrían indicar que la red está generalizando correctamente, a diferencia de los problemas de sobreajuste que se encuentran en los métodos de aprendizaje superficial o clásico. Para verificar esto, se debe lograr un rendimiento similar en el *dataset* de prueba.

En cuanto a los resultados sobre el *dataset* de prueba (Cuadro 5.2), PromoterLCNN supera a pcPromoter-CNN, ya que los promedios ponderados son 89,6 % para Lc y 83,0 % para Pc, logrando además resultados comparables (e incluso ligeramente mejores) que iPromoter-BnCNN en términos de precisión. PromoterLCNN produce una cantidad similar o incluso menor de falsos negativos, como lo ilustran los valores de especificidad de los promotores (Sp). Sin embargo, es importante recordar el Cuadro

4.1 para un análisis prudente: cada clase de promotor tiene solo unos pocos ejemplos y no hay no-promotores. Por lo tanto, las estadísticas más relevantes son las obtenidas para σ^{70} y σ^{24} (es decir, $n \geq 30$). Estas estadísticas muestran que el enfoque propuesto es mejor para clasificar promotores, a costa de no detectar algunos de ellos (falsos negativos) en comparación con iPromoter-BnCNN.

Establecer el desempeño competitivo de PromoterLCNN es vital para notar la parsimonia de esta propuesta. La arquitectura presentada es apreciablemente más liviana que pcPromoter-CNN y iPromoter-BnCNN. El primero utiliza una arquitectura en cascada para clasificar primero promotores y no promotores, y luego subclases sigma de forma secuencial, con nueve capas diferentes cada una [27]. El segundo utiliza la misma arquitectura en cascada, con cuatro capas paralelas (4 capas en cada rama), que convergen en 4 capas al final [12]. Por el contrario, el método aquí definido cuenta con tan solo dos etapas con ocho capas diferentes cada una. Esta característica disminuye significativamente el tiempo de cómputo para los procesos de entrenamiento, predicción y optimización de hiperparámetros, siendo una décima parte del tiempo en comparación con iPromoter-BnCNN, y un 30 % menos que pcPromoter-CNN.

Capítulo 6

Conclusiones y Trabajo Futuro

Este trabajo presenta un modelo de predicción y clasificación de promotores en dos etapas mediante el uso de una red neuronal convolucional multiclase denominada PromoterLCNN. La primera etapa de la arquitectura intenta distinguir entre promotores y no promotores, y la segunda etapa realiza la clasificación sigma mediante el uso de un modelo de clasificación multiclase sencillo, en contraste con los enfoques estándar que se encuentran en la literatura. Se aplicaron las reglas de cinco pasos de Chou et al. en el proceso de presentación y validación de modelos, haciendo uso de las secuencias pertenecientes a *E. coli* encontradas en las bases de datos de referencia, RegulonDB v9.3 y v10.7, para entrenamiento y pruebas independientes, respectivamente.

Se utilizó un entrenamiento de validación cruzada *Stratified K-fold*, evaluando los resultados con un conjunto de datos de prueba independiente. Se descubrió así que este método supera a uno de los más competitivos de la literatura. En el caso de los enfoques del estado del arte más recientes, el modelo propuesto presenta resultados competitivos en cuanto a precisión, con una mejor clasificación del tipo de promotor. Cabe destacar que PromoterLCNN tiene una arquitectura más liviana que otros modelos, lo que genera un menor tiempo para los procesos de ajuste de hiperparámetros, entrenamiento y predicción, sin comprometer la calidad de la clasificación, una cualidad atractiva para los biólogos moleculares o sintéticos que trabajan con secuencias de nucleótidos a dia-

rio. Al utilizar parte de un genoma o una secuencia recién sintetizada como datos de entrada, PromoterLCNN podría ayudar a los investigadores y usuarios que trabajan en el campo de la genómica bacteriana, la biología molecular y la bioinformática a identificar promotores bacterianos y clasificarlos en cada una de las subclases σ validadas en este estudio.

6.1. Contribuciones

[1] Daryl Hernández, Nicolás Jara, Mauricio Araya, Roberto E. Durán y Carlos Buil-Aranda. PromoterLCNN: A Light CNN-Based Promoter Prediction and Classification Model. *Genes*, 13(7), 2022. ISSN: 2073-4425. DOI: 10.3390/genes13071126

Bibliografía

[1] Bertil Schmidt y Andreas Hildebrandt. Next-generation sequencing: big data meets high performance computing. *Drug Discovery Today*, 22(4):712-717, 2017. ISSN: 1359-6446. DOI: https://doi.org/10.1016/j.drudis.2017.01.014.

[2] Sam Behjati y Patrick S Tarpey. What is next generation sequencing? *Archives of Disease in Childhood - Education and Practice*, 98(6):236-238, 2013. ISSN: 1743-0585. DOI: 10.1136/archdischild-2013-304340. eprint: https://ep.bmj.com/content/98/6/236.full.pdf.

[3] Jun Zhang, Rod Chiodini, Ahmed Badr y Genfa Zhang. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, 38(3):95-109, 2011. ISSN: 1673-8527. DOI: https://doi.org/10.1016/j.jgg.2011.02.003.

[4] René Dreos, Giovanna Ambrosini, Rouayda Cavin Périer y Philipp Bucher. EPD and EPDnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic Acids Research*, 41(D1):D157-D164, noviembre de 2012. ISSN: 0305-1048. DOI: 10.1093/nar/gks1233. eprint: https://academic.oup.com/nar/article-pdf/41/D1/D157/18788951/gks1233.pdf.

[5] A.S. Nikitina, E.I. Sharova, S.A. Danilenko, O.V. Selezneva, T.B. Butusova, A.O. Vasiliev, A.V. Govorov, E.A. Prilepskaya, D.Y. Pushkar y E.S. Kostryukova. Datasets for next-generation sequencing of DNA and RNA from urine and

plasma of patients with prostate cancer. *Data in Brief*, 10:369-372, 2017. ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2016.12.016.

[6] John Dahl y Philippe Collas. A rapid micro chromatin immunoprecipitation assay (ChIP). *Nature Protocols*, 3:1032-1045, mayo de 2008. DOI: 10.1038/nprot.2008.68.

[7] Hiroto Matsumine, Yasuhiro Yamamura, Nobutaka Hattori, Tomonori Kobayashi, Tohru Kitada, Asako Yoritaka y Yoshikuni Mizuno. A Microdeletion of D6S305 in a Family of Autosomal Recessive Juvenile Parkinsonism (PARK2). *Genomics*, 49(1):143-146, 1998. ISSN: 0888-7543. DOI: https://doi.org/10.1006/geno.1997.5196.

[8] Junwen Wang y Sridhar Hannenhalli. A mammalian promoter model links cis elements to genetic networks. *Biochemical and Biophysical Research Communications*, 347(1):166-177, 2006. ISSN: 0006-291X. DOI: https://doi.org/10.1016/j.bbrc.2006.06.062.

[9] Ramzan Kh. Umarov y Victor V. Solovyev. Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. *PLOS ONE*, 12(2):1-12, febrero de 2017. DOI: 10.1371/journal.pone.0171410. URL: https://doi.org/10.1371/journal.pone.0171410.

[10] Y. Qian, Y. Zhang, B. Guo, S. Ye, Y. Wu y J. Zhang. An Improved Promoter Recognition Model Using Convolutional Neural Network. *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 01:471-476, julio de 2018. ISSN: 0730-3157. DOI: 10.1109/COMPSAC.2018.00072.

[11] Mhaned Oubounyt, Zakaria Louadi, Hilal Tayara y Kil To Chong. DeePromoter: Robust Promoter Predictor Using Deep Learning. *Frontiers in Genetics*, 10:286, 2019. ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00286.

[12] Ruhul Amin, Chowdhury Rafeed Rahman, Sajid Ahmed, Md Habibur Rahman Sifat, Md Nazmul Khan Liton, Md Moshiur Rahman, Md Zahid Hossain Khan

y Swakkhar Shatabda. iPromoter-BnCNN: a novel branched CNN-based predictor for identifying and classifying sigma promoters. *Bioinformatics*, 36(19):4869-4875, julio de 2020. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa609. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/19/4869/34806226/btaa609.pdf.

[13] Tanja M. Gruber y Carol A. Gross. Multiple sigma subunits and the partitioning of bacterial transcription space. *Annu. Rev. Microbiol.*, 57(1):441-466, 2003.

[14] Murilo Henrique Anzolini Cassiano y Rafael Silva-Rocha. Benchmarking bacterial promoter prediction tools: Potentialities and limitations. *mSystems*, 5(4), 2020.

[15] John D. Helmann. Where to begin? Sigma factors and the selectivity of transcription initiation in bacteria. *Mol. Microbiol.*, 112(2):335-347, agosto de 2019.

[16] Indra Bervoets, Maarten Van Brempt, Katleen Van Nerom, Bob Van Hove, Jo Maertens, Marjan De Mey y Daniel Charlier. A sigma factor toolbox for orthogonal gene expression in *Escherichia coli*. *Nucleic Acids Res.*, 46(4):2133-2144, febrero de 2018.

[17] Jennie E Mitchell, Dongling Zheng, Stephen J W Busby y Stephen D Minchin. Identification and analysis of 'extended -10' promoters in Escherichia coli. *Nucleic Acids Res.*, 31(16):4689-4695, 2003.

[18] Athanasios Typas y Regine Hengge. Differential ability of sigma(s) and sigma70 of Escherichia coli to utilize promoters containing half or full UP-element sites. *Mol. Microbiol.*, 55(1):250-260, 2005.

[19] Ana G. Abril, Jose Luis R. Rama, A. Sánchez-Pérez y Tomás G. Villa. Prokaryotic sigma factors and their transcriptional counterparts in Archaea and Eukarya. *Appl. Microbiol. Biotechnol.*, 104(10):4289-4302, mayo de 2020.

[20] Daryl Hernández, Nicolás Jara, Mauricio Araya, Roberto E. Durán y Carlos Buil-Aranda. PromoterLCNN: A Light CNN-Based Promoter Prediction and Classification Model. *Genes*, 13(7), 2022. ISSN: 2073-4425. DOI: 10.3390/genes13071126.

[21] Marco Di Salvo, Eva Pinatel, Adelfia Talà, Marco Fondi, Clelia Peano y Pietro Alifano. G4PromFinder: an algorithm for predicting transcription promoters in GC-rich bacterial genomes based on AT-rich elements and G-quadruplex motifs. *BMC Bioinformatics*, 19(1), 2018.

[22] Akira Ishihama, Tomohiro Shimada y Yukiko Yamazaki. Transcription profile of *Escherichia coli*: genomic SELEX search for regulatory targets of transcription factors. *Nucleic Acids Res.*, 44(5):2058-2074, 2016.

[23] Victor Solovyev y Asaf Salamov. *Automatic annotation of microbial genomes and metagenomic sequences*. En *Metagenomics and its applications in agriculture, biomedicine and environmental studies*. Robert W. Li, edición. Nova Science Publishers, Inc., 2011, páginas 61-78.

[24] Richard Münch, Karsten Hiller, Andreas Grote, Maurice Scheer, Johannes Klein, Max Schobert y Dieter Jahn. Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics*, 21(22):4187-4189, 2005.

[25] Wenying He, Cangzhi Jia, Yucong Duan y Quan Zou. 70ProPred: a predictor for discovering sigma70 promoters based on combining multiple features. *BMC Syst. Biol.*, 12(S4), 2018.

[26] Md Siddiqur Rahman, Usma Aktar, Md Rafsan Jani y Swakkhar Shatabda. iPro70-FMWin: identifying Sigma70 promoters using multiple windowing and minimal features. *Mol. Genet. Genomics*, 294(1):69-84, 2019.

[27] Muhammad Shujaat, Abdul Wahab, Hilal Tayara y Kil To Chong. pcPromoter-CNN: A CNN-Based Prediction and Classification of Promoters. *Genes*, 11(12),

2020. ISSN: 2073-4425. DOI: 10.3390/genes11121529. URL: https://www.mdpi.com/2073-4425/11/12/1529.

[28] D. Pribnow. Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. *Proceedings of the National Academy of Sciences*, 72(3):784-788, 1975. DOI: 10.1073/pnas.72.3.784. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.72.3.784. URL: https://www.pnas.org/doi/abs/10.1073/pnas.72.3.784.

[29] Meng Zhang, Fuyi Li, Tatiana T Marquez-Lago, André Leier, Cunshuo Fan, Chee Keong Kwoh, Kuo-Chen Chou, Jiangning Song y Cangzhi Jia. MULTiPly: a novel multi-layer predictor for discovering general and specific types of promoters. *Bioinformatics*, 35(17):2957-2965, enero de 2019. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz016. URL: https://doi.org/10.1093/bioinformatics/btz016.

[30] Pengmian Feng, Hui Ding, Hui Yang, Wei Chen, Hao Lin y Kuo-Chen Chou. iRNA-PseColl: identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. *Molecular Therapy-Nucleic Acids*, 7:155-163, 2017.

[31] Kuo-Chen Chou. Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.*, 273(1):236-247, marzo de 2011.

[32] Socorro Gama-Castro, Heladia Salgado, Alberto Santos-Zavaleta, Daniela Ledezma-Tejeida, Luis Muñiz-Rascado, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Irma Martínez-Flores, Lucia Pannier, Jaime Abraham Castro-Mondragón, Alejandra Medina-Rivera, Hilda Solano-Lira, César Bonavides-Martínez, Ernesto Pérez-Rueda, Shirley Alquicira-Hernández, Liliana Porrón-Sotelo, Alejandra López-Fuentes, Anastasia Hernández-Koutoucheva, Víctor Del Moral-Chávez, Fabio Rinaldi y Julio Collado-Vides. RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Res.*, 44(D1):D133-43, enero de 2016.

[33] Alberto Santos-Zavaleta, Heladia Salgado, Socorro Gama-Castro, Mishael Sánchez-Pérez, Laura Gómez-Romero, Daniela Ledezma-Tejeida, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Luis José Muñiz-Rascado, Pablo Peña-Loredo, Cecilia Ishida-Gutiérrez, David A Velázquez-Ramírez, Víctor Del Moral-Chávez, César Bonavides-Martínez, Carlos-Francisco Méndez-Cruz, James Galagan y Julio Collado-Vides. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* K-12. *Nucleic Acids Res.*, 47(D1):D212-D220, 2019.

[34] Francis Crick. Central Dogma of Molecular Biology. *Nature*, 227(5258):561-563, agosto de 1970. ISSN: 1476-4687. DOI: 10.1038/227561a0. URL: https://doi.org/10.1038/227561a0.