UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Diseño e implementación de modelos de lenguaje para información genómica asociada a enfermedades raras mediante inferencia gramatical

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Gómez Ruiz, Iván

Tutor/a: Sempere Luna, José María

CURSO ACADÉMICO: 2023/2024

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de modelos de lenguaje para información genómica asociada a enfermedades raras mediante inferencia gramatical

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Iván Gómez Ruiz

Tutor: Jose María Sempere Luna

Curso 2023-2024

Resumen

El presente Trabajo de Fin de Grado aborda el diseño e implementación de modelos de lenguaje para la información genómica asociada a enfermedades raras, específicamente la retinosis pigmentaria, utilizando algoritmos de inferencia gramatical k-testables. La investigación se centra en desarrollar modelos de autómatas finitos deterministas (AFD) que puedan identificar patrones en secuencias genéticas, diferenciando entre muestras de ADN mutado y no mutado.

El estudio comenzó con la recopilación de datos genómicos a partir de archivos VCF proporcionados por el Instituto de La Fe de Valencia. Estos datos fueron procesados para generar muestras de secuencias genéticas, las cuales se utilizaron para entrenar y evaluar los modelos. Se implementaron dos tipos de modelos: uno con secuencias mutadas y otro con secuencias no mutadas. Cada modelo fue optimizado variando el parámetro k, que define el tamaño de la ventana de contexto del autómata.

Los resultados del estudio muestran que los modelos entrenados con secuencias mutadas y no mutadas presentan diferentes niveles de precisión, especificidad y recall. La investigación demuestra el potencial de los AFD basados en algoritmos k-testables para el análisis genómico y la identificación de mutaciones, aportando valor al campo de la bioinformática y el diagnóstico de enfermedades raras.

Palabras clave: Modelos de lenguaje, Enfermedades raras, Retinosis pigmentaria, Inferencia gramatical, Autómatas finitos deterministas, Secuencias genéticas, Mutaciones, Bioinformática, Diagnóstico genético

Abstract

This Final Degree Project addresses the design and implementation of language models for genomic information associated with rare diseases, specifically retinitis pigmentosa, using k-testable grammatical inference algorithms. The research focuses on developing deterministic finite automata (DFA) models capable of identifying patterns in genetic sequences, differentiating between mutated and non-mutated DNA samples.

The study began with the collection of genomic data from VCF files provided by the Instituto de La Fe de Valencia. These data were processed to generate samples of genetic sequences, which were used to train and evaluate the models. Two types of models were implemented: one with mutated sequences and another with non-mutated sequences. Each model was optimized by varying the k parameter, which defines the context window size of the automaton.

The study results show that models trained with mutated and non-mutated sequences exhibit different levels of precision, specificity, and recall. The research demonstrates the potential of DFA based on k-testable algorithms for genomic analysis and mutation identification, contributing to the field of bioinformatics and the diagnosis of rare diseases.

Key words: Language models, Rare diseases, Retinitis pigmentosa, Grammatical inference, Deterministic finite automata, Genetic sequences, Mutations, Bioinformatics, Genetic diagnosis

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

1.1 Conceptos Básicos

1.1.1. Enfermedades Raras y la Inteligencia Artificial

La inteligencia artificial (IA) ha emergido como una herramienta revolucionaria en el campo de la medicina. Su capacidad para procesar grandes volúmenes de datos y reconocer patrones complejos la convierte en una herramienta muy poderosa para los investigadores y médicos. [1]

Las enfermedades raras, aquellas que afectan a una pequeña parte de la población, presentan desafíos únicos debido a su baja prevalencia y la falta de investigación extensiva. Aquí es donde la IA puede desempeñar un papel crucial, ayudando a llenar los vacíos de conocimiento al analizar datos genéticos y biomédicos dispersos para descubrir correlaciones y causas subyacentes. [1]

Las distrofias hereditarias de retina son un conjunto de enfermedades raras que afectan la retina, la capa de tejido sensible a la luz en la parte posterior del ojo. Estas enfermedades son genéticas y provocan una pérdida progresiva de la visión, que puede variar desde una disminución leve hasta la ceguera total. [2]

El diagnóstico y tratamiento de estas afecciones son complejos debido a su naturaleza genética y a la diversidad de tipos que existen. [2]

1.1.2. Distrofias retinianas hereditarias

El ojo y la visión

El ojo actúa como un instrumento capaz de captar la luz del ambiente, proyectando una imagen invertida en la retina. Las células sensibles a la luz convierten la imagen capturada en señales eléctricas, que son enviadas al cerebro por medio del nervio óptico. En la zona posterior del cerebro, estas señales son procesadas e interpretadas, permitiéndonos así comprender lo que vemos. [3]

La luz, al entrar en el ojo, inicia su viaje hacia la corteza cerebral siendo captada por la retina, gracias a una capa de fotoreceptores (bastones y conos) que convierten la luz en señales cerebrales. Los bastones, sensibles a la luz de baja intensidad, son cruciales para la visión nocturna, mientras que los conos, concentrados en la fovea centralis, permiten la agudeza visual y la diferenciación de colores. [4]

Existen tres tipos de conos (rojos, verdes y azules) cada uno sensible a diferentes longitudes de onda. Estos fotoreceptores contienen proteínas (opsinas) en sus segmentos

externos, que al reaccionar con la luz, activan la fototransducción, transformando la luz en señales químicas. Estas señales son luego transmitidas al cerebro a través de las células bipolares y el nervio óptico, culminando su recorrido en la corteza visual, donde se procesa la percepción visual. [4]

La retinosis pigmentaria afecta a la capacidad del ojo para convertir la luz en señales cerebrales de manera eficiente, llevando a una pérdida progresiva de la visión que puede comenzar desde la infancia hasta la edad adulta. Este trastorno se caracteriza por síntomas como la ceguera nocturna y una reducción gradual del campo visual, eventualmente afectando la visión central. [5]

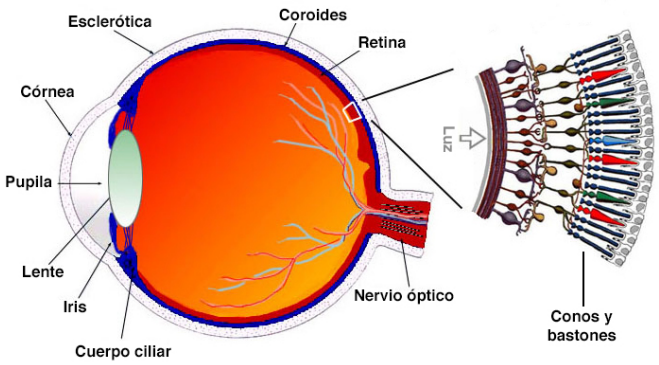


Figura 1.1: Diagrama de la estructura ocular que muestra una vista sagital del ojo humano con sus componentes principales, así como una representación detallada de las células de la retina, destacando los fotorreceptores de conos y bastones. [4]

1.1.3. Inteligencia Artificial

La inteligencia artificial (IA) es la capacidad de las máquinas para realizar tareas que normalmente requieren inteligencia humana. Esto abarca actividades como el aprendizaje, la comprensión del lenguaje natural, el reconocimiento visual y de voz, la toma de decisiones, y la capacidad de adaptarse a diferentes tipos de escenarios y datos. [6]

A nivel técnico, la inteligencia artificial utiliza algoritmos avanzados para procesar y aprender de grandes cantidades de datos, con el fin de simular procesos cognitivos humanos. [6]

Las máquinas basadas en inteligencia artificial son capaces de realizar tareas con una eficiencia y precisión que a menudo superan las capacidades humanas, como el procesamiento y análisis de datos a gran escala, la optimización de procesos y la automatización de decisiones.[6]

Uno de los elementos fundamentales de la inteligencia artificial es el aprendizaje automático (Machine Learning), un subconjunto de la IA que implica la creación de algoritmos que permiten a los sistemas aprender y mejorar a partir de la experiencia sin ser explícitamente programados. [6]

El aprendizaje profundo (Deep Learning), otra rama importante, utiliza redes neuronales con muchas capas para analizar varios tipos de datos, permitiendo así aplicaciones avanzadas en reconocimiento de imágenes, procesamiento del lenguaje natural, y más. [6]

1.1.4. Situación actual de la tecnología

1.1.4.1. Aprendizaje automático (Machine Learning)

Aprendizaje supervisado

El aprendizaje supervisado es una rama del aprendizaje automático que implica entrenar un modelo en un conjunto de datos etiquetados, donde cada entrada de datos tiene una salida correspondiente. Este método permite al modelo utilizar los patrones observados durante el entrenamiento para predecir la salida para nuevas entradas. El modelo principalmente crea una función que mapea entradas y salidas para optimizar el rendimiento a través de iteraciones. [7]

Características: El aprendizaje supervisado entrena algoritmos que pueden generalizarse a partir de estos ejemplos para predecir resultados en datos no vistos utilizando pares de datos de entrada-salida etiquetados. Problemas como la clasificación y la regresión utilizan esta metodología. [7] [8]

Aplicaciones: En el campo de la salud, los algoritmos de aprendizaje supervisado pueden utilizarse para diagnosticar enfermedades utilizando imágenes médicas o historiales clínicos para predecir el curso de una enfermedad. [8]

Aprendizaje no supervisado

El aprendizaje no supervisado, por otro lado, utiliza datos no etiquetados. El objetivo principal es identificar estructuras o patrones subyacentes en los datos sin un objetivo específico en mente. Esta sección es útil para encontrar agrupaciones ocultas, reducir dimensiones y encontrar anomalías en conjuntos de datos de gran tamaño. [7]

Características: Los algoritmos en el aprendizaje no supervisado buscan patrones en los datos sin referencias a salidas conocidas. El análisis de grupos y la reducción de dimensionalidad son métodos comunes. [7] [8]

Aplicaciones: La categorización de grupos de pacientes con síntomas similares para identificar una causa común sería un uso notable en la sanidad. [8]

Aprendizaje profundo (Deep Learning)

El aprendizaje profundo (DL, por sus siglas en inglés) es una clase de algoritmos que aprende utilizando una extensa colección de procesos conectados y multicapa, exponiendo estos procesadores a un vasto conjunto de ejemplos. El DL se ha establecido como uno de los métodos predominantes en la inteligencia artificial (IA) en la actualidad, impulsando mejoras en áreas como el reconocimiento de imágenes y el reconocimiento de voz. [8]

Características: El aprendizaje profundo se distingue por su capacidad para procesar y aprender de grandes volúmenes de datos de manera autónoma. Sus redes neuronales, compuestas por múltiples capas, permiten que el sistema identifique patrones complejos y extraiga características esenciales sin intervención humana directa. Esta técnica es altamente adaptable y puede mejorar su rendimiento a medida que se expone a más datos. Además, su flexibilidad le permite abordar una amplia gama de problemas con una precisión notable, desde el análisis de textos hasta la predicción de comportamientos. [8]

Aplicaciones: En el campo de la medicina, el aprendizaje profundo ha revolucionado numerosos aspectos, desde el diagnóstico hasta el tratamiento. Por ejemplo, en la radiología, los algoritmos de DL pueden analizar imágenes médicas, como radiografías y resonancias magnéticas, para detectar anomalías con una precisión comparable o incluso superior a la de los especialistas humanos. Esto es especialmente útil en la detección

temprana de enfermedades como el cáncer. En la genómica, el DL ayuda a interpretar secuencias genéticas para identificar mutaciones asociadas con trastornos hereditarios. [9]

1.1.4.2. Procesamiento del lenguaje natural (NLP)

El Procesamiento del Lenguaje Natural (PLN), conocido también por sus siglas en inglés NLP (Natural Language Processing), es una rama de la inteligencia artificial que se centra en la interacción entre las computadoras y los lenguajes humanos. El objetivo principal del PLN es permitir que las máquinas comprendan, interpreten y produzcan el lenguaje humano de una manera que sea valiosa y útil. A diferencia de los simples dispositivos que procesan texto o audio sin entender su significado, el PLN implica una comprensión más profunda de los contextos y las estructuras lingüísticas.

El procesamiento del lenguaje natural no se limita simplemente a reconocer palabras o frases, sino que se extiende a la comprensión del contexto, la semántica y la intención detrás de las palabras. Esto implica una serie de tareas complejas como la tokenización (división del texto en unidades manejables como palabras o frases), el análisis sintáctico (comprensión de la estructura gramatical), el análisis semántico (interpretación del significado) y la generación de lenguaje natural (producción de texto o habla coherente y relevante).[10]

Las aplicaciones del PLN son vastas y variadas, abarcando múltiples industrias y aspectos de la vida cotidiana. A continuación, se describen algunas de las principales aplicaciones del PLN:

1. **Sistemas de Búsqueda y Recuperación de Información:** Los motores de búsqueda avanzados utilizan técnicas de PLN para entender las consultas de los usuarios y proporcionar resultados relevantes. A diferencia de los sistemas básicos que buscan coincidencias exactas de palabras clave, los sistemas basados en PLN pueden entender la intención detrás de una consulta y devolver información que es semánticamente relevante, aunque no contenga las palabras exactas utilizadas en la búsqueda.[10]

2. **Traducción Automática:** La traducción de texto de un idioma a otro ha sido una de las áreas más visibles del PLN. Herramientas como Google Translate utilizan modelos avanzados de PLN para traducir textos y páginas web en tiempo real, permitiendo la comunicación entre personas que hablan diferentes idiomas.[10]

3. **Asistentes Virtuales y Chatbots:** Asistentes como Siri, Alexa y Google Assistant son productos del PLN. Estos sistemas pueden comprender comandos hablados, responder preguntas, realizar tareas y mantener conversaciones, mejorando continuamente a través del aprendizaje automático y la acumulación de datos.[11]

El Procesamiento del Lenguaje Natural es una disciplina vital en el campo de la inteligencia artificial que tiene aplicaciones profundas y amplias en diversas industrias. Su capacidad para entender y manipular el lenguaje humano abre nuevas posibilidades para la automatización, la eficiencia y la mejora de la toma de decisiones.

1.1.4.3. Visión por computadora

La visión por computadora es un campo de la inteligencia artificial que se enfoca en la capacidad de las máquinas para interpretar y comprender imágenes del mundo real. Utilizando algoritmos y modelos matemáticos, estos sistemas pueden procesar y analizar imágenes de forma automática para extraer información útil, detectar objetos o interpretar imágenes de manera similar a como lo haría un ser humano. Esta disciplina combina técnicas de procesamiento de imágenes y aprendizaje automático para lograr una comprensión avanzada de los datos visuales, facilitando tareas complejas como el reconocimiento de patrones, la detección de anomalías y la clasificación de objetos.[12]

La visión por computadora tiene una amplia gama de aplicaciones en diversas industrias. Algunas de las más destacadas incluyen:

1. **Seguridad y Vigilancia:** Sistemas de cámaras que utilizan algoritmos de visión por computadora para detectar movimientos sospechosos, identificar rostros y monitorear actividades en tiempo real.
2. **Automóviles Autónomos:** Los vehículos autónomos emplean visión por computadora para identificar señales de tráfico, peatones y otros vehículos, permitiendo una conducción segura y autónoma.
3. **Agricultura de Precisión:** Drones y robots agrícolas utilizan visión por computadora para monitorear el crecimiento de cultivos, detectar plagas y optimizar el uso de fertilizantes y agua.
4. **Manufactura y Control de Calidad:** En la industria manufacturera, la visión por computadora se usa para inspeccionar productos, detectar defectos y asegurar la calidad de los bienes producidos.

La visión por computadora ha encontrado aplicaciones significativas en el campo de la medicina, mejorando la precisión y la eficiencia en el diagnóstico y tratamiento de enfermedades. Algunas de las aplicaciones más importantes son:

1. **Detección de Cáncer en Imágenes de Tejidos:** Modelos de visión por computadora analizan imágenes histopatológicas para detectar regiones con posibles anomalías que podrían indicar la presencia de cáncer. Esta técnica permite a los patólogos identificar células y tejidos enfermos con mayor precisión y rapidez.[12]
2. **Análisis Cerebral en Resonancias Magnéticas:** Las resonancias magnéticas (RM) generan imágenes tridimensionales del cerebro que pueden ser analizadas por sistemas de visión por computadora para identificar enfermedades neurológicas, como tumores cerebrales y esclerosis múltiple. [12]
3. **Clasificación de Enfermedades en Radiografías de Tórax:** Las radiografías de tórax son utilizadas comúnmente para diagnosticar enfermedades pulmonares y del sistema respiratorio. Modelos de redes neuronales convolucionales se entrenan con datos de radiografías etiquetadas por expertos para identificar condiciones como neumonía, tuberculosis y cáncer de pulmón. [12]
4. **Monitorización de Pacientes y Bioseguridad:** Durante la pandemia del COVID-19, se utilizaron cámaras equipadas con visión por computadora para detectar automáticamente personas con temperatura elevada y verificar el uso de mascarillas, contribuyendo a los esfuerzos de bioseguridad y control de la propagación del virus.[12]

5. **Aplicaciones Didácticas:** La visión por computadora también se utiliza en la educación médica, proporcionando plataformas didácticas donde los estudiantes pueden aprender a interpretar imágenes médicas y entender cómo los algoritmos de inteligencia artificial detectan anomalías. [12]

La visión por computadora representa un avance significativo en la capacidad de las máquinas para interpretar datos visuales, con aplicaciones que abarcan desde la seguridad hasta la medicina. En el ámbito médico, esta tecnología no solo mejora el diagnóstico y tratamiento de enfermedades, sino que también facilita la educación y formación de nuevos profesionales. La integración de la visión por computadora en la medicina promete seguir revolucionando la atención sanitaria, ofreciendo herramientas poderosas para enfrentar los desafíos del diagnóstico y tratamiento en el futuro inmediato.

1.1.5. Inteligencia Artificial aplicada a la Medicina

La aplicación de IA en el ámbito médico tiene el potencial de revolucionar tanto la práctica clínica como la investigación biomédica, proporcionando herramientas avanzadas para mejorar el diagnóstico, tratamiento y seguimiento de enfermedades. Las capacidades de la IA para procesar grandes volúmenes de datos, identificar patrones complejos y realizar predicciones precisas permiten a los profesionales de la salud tomar decisiones más informadas y personalizadas. Esta sección explora las diversas aplicaciones y beneficios de la IA en la medicina.

1.1.6. Aplicaciones de la IA en Diagnóstico y Tratamiento

La IA se ha integrado en el diagnóstico y tratamiento médico a través de múltiples enfoques que optimizan la precisión y eficiencia de las intervenciones clínicas. A continuación, se desglosan las principales aplicaciones de la IA en estas áreas.

1.1.6.1. Diagnóstico asistido por IA

Análisis de imágenes médicas: La IA ha mostrado un rendimiento excepcional en el análisis de imágenes médicas, incluyendo radiografías, tomografías computarizadas (TC), resonancias magnéticas (RM) y mamografías. Los algoritmos de aprendizaje profundo, especialmente las redes neuronales convolucionales (CNN), pueden detectar anomalías con una precisión comparable, y en algunos casos superior, a la de los radiólogos humanos. Estas herramientas son capaces de identificar patrones sutiles en las imágenes que podrían pasar desapercibidos para el ojo humano, facilitando un diagnóstico temprano y más preciso de diversas patologías, como cánceres, enfermedades cardiovasculares y trastornos neurológicos. [12] [13]

Reconocimiento de patrones en datos genéticos: La IA también desempeña un papel crucial en el análisis de datos genéticos. Los algoritmos de aprendizaje automático pueden identificar variantes genéticas asociadas con enfermedades específicas, permitiendo un diagnóstico más preciso y personalizado. Este reconocimiento de patrones es esencial en el campo de la medicina de precisión, donde el objetivo es adaptar los tratamientos a las características genéticas individuales de cada paciente. La secuenciación del genoma y el análisis de grandes conjuntos de datos genómicos han sido facilitados por técnicas de IA, proporcionando nuevas vías para la identificación de predisposiciones genéticas y la personalización de terapias.[14]

1.1.6.2. Sistemas de soporte para decisiones clínicas

Los sistemas de soporte para decisiones clínicas (CDSS, por sus siglas en inglés) utilizan IA para asistir a los médicos en la toma de decisiones basadas en datos. Estos sistemas integran información clínica del paciente, guías médicas, bases de datos de enfermedades y tratamientos, y otros recursos relevantes para proporcionar recomendaciones basadas en la evidencia.

Estos sistemas pueden mejorar significativamente la calidad del cuidado del paciente al sugerir diagnósticos diferenciales, recomendaciones de tratamiento y alertas sobre posibles interacciones medicamentosas. Además, los CDSS pueden ayudar a estandarizar la práctica clínica y reducir la variabilidad en el cuidado de los pacientes, garantizando que se sigan las mejores prácticas clínicas basadas en la evidencia más reciente. [15]

1.1.6.3. Modelado predictivo para diagnósticos y pronósticos

El modelado predictivo mediante IA implica la utilización de algoritmos para predecir el curso de una enfermedad o la respuesta a un tratamiento basado en datos históricos y características específicas del paciente. Estos modelos pueden analizar grandes cantidades de datos clínicos y extraer patrones que no son evidentes para los médicos, proporcionando predicciones precisas sobre la progresión de la enfermedad, las tasas de supervivencia y la eficacia de los tratamientos.[16]

Aplicaciones del modelado predictivo:

- **Diagnósticos:** Los modelos predictivos pueden ayudar a identificar pacientes en riesgo de desarrollar ciertas condiciones antes de que los síntomas se manifiesten, permitiendo intervenciones tempranas que pueden mejorar los resultados de salud.
- **Pronósticos:** Estos modelos pueden prever la evolución de enfermedades crónicas, como la diabetes o enfermedades cardíacas, y predecir la probabilidad de complicaciones, lo que permite a los médicos ajustar los planes de tratamiento de manera proactiva.
- **Eficacia del tratamiento:** La IA puede predecir la respuesta de un paciente a tratamientos específicos, facilitando la personalización de terapias y mejorando la efectividad del tratamiento.

1.2 Planteamiento del problema y motivación

El campo de la medicina ha experimentado un cambio significativo con la introducción de la inteligencia artificial (IA) y el aprendizaje automático (Machine Learning). Una de las áreas donde estos avances tecnológicos tienen un impacto notable es en el análisis de datos genéticos y la identificación de variantes genéticas asociadas con enfermedades específicas. En este contexto, se presenta un problema complejo: la detección precisa y eficiente de mutaciones genéticas asociadas a enfermedades raras, como la retinosis pigmentaria, utilizando modelos de lenguaje y algoritmos de inferencia gramatical.

Como se ha descrito antes, la retinosis pigmentaria es una enfermedad rara que afecta a la retina, la capa de tejido sensible a la luz en la parte posterior del ojo, y se caracteriza por una pérdida progresiva de la visión. Esta afección presenta un desafío significativo para la investigación y el diagnóstico debido a su baja prevalencia y la variabilidad genética entre los individuos afectados . Los avances en la secuenciación del genoma han permitido recopilar volúmenes de datos genómicos, pero la interpretación de estos datos para identificar variantes genéticas específicas sigue siendo un desafío considerable .

La motivación principal de este estudio radica en la necesidad de desarrollar modelos más efectivos para analizar secuencias genéticas y detectar mutaciones asociadas con enfermedades raras. Los modelos basados en autómatas finitos deterministas (AFD) y creados a partir de algoritmos de inferencia gramatical k-testables ofrecen una prometedora solución a este problema. Estos modelos pueden procesar grandes cantidades de datos y reconocer patrones complejos en secuencias genéticas, lo que es esencial para la identificación precisa de mutaciones genéticas.

1.3 Objetivos

El objetivo principal de este trabajo es diseñar e implementar modelos de lenguaje para la información genómica asociada a enfermedades raras mediante el uso de algoritmos de inferencia gramatical. Este objetivo se desglosa en varios subobjetivos específicos que guiarán el desarrollo y la evaluación de los modelos propuestos:

1. **Desarrollo de Modelos de Lenguaje Basados en AFD:**

   Crear y entrenar modelos de lenguaje utilizando algoritmos de inferencia gramatical k-testables. Estos modelos deben ser capaces de procesar y analizar secuencias genéticas, identificando patrones que indiquen la presencia de mutaciones específicas relacionadas con la retinosis pigmentaria.

2. **Evaluación del Rendimiento del Modelo:**

   Realizar una evaluación exhaustiva del rendimiento de los modelos desarrollados, utilizando métricas como la precisión, el recall, la especificidad y la precisión global (accuracy). Esta evaluación debe considerar tanto secuencias genéticas mutadas como no mutadas para asegurar una comprensión completa del desempeño del modelo.

3. **Optimización del Proceso de Entrenamiento:**

   Implementar técnicas de optimización para mejorar la eficiencia del proceso de entrenamiento. Esto incluye el uso de paralelización de procesos mediante hilos y estructuras de datos eficientes que permitan reducir el tiempo de cómputo y manejar conjuntos de datos más grandes.

4. **Comparación de Modelos:**

   Comparar el rendimiento de modelos entrenados con secuencias genéticas mutadas frente a modelos entrenados con secuencias no mutadas. Esta comparación debe proporcionar información sobre la capacidad de cada tipo de modelo para identificar correctamente las mutaciones genéticas y diferenciar entre secuencias mutadas y no mutadas.

Este trabajo tiene como objetivo contribuir al campo de la bioinformática y la genética mediante el desarrollo de modelos avanzados de análisis de secuencias genéticas. A través de la implementación y optimización de algoritmos de inferencia gramatical, se espera mejorar la capacidad de identificar mutaciones genéticas y apoyar el diagnóstico y tratamiento de enfermedades raras como la retinosis pigmentaria.

CAPÍTULO 2
Marco Teórico

2.1 Lenguajes K-testables

2.1.1. Definición y características de los lenguajes k-testables

Un lenguaje, en el contexto de autómatas y lenguajes formales, es un conjunto de cadenas de caracteres que están formadas a partir de un alfabeto específico. Estas cadenas deben de cumplir ciertas reglas o patrones que definen cierto lenguaje.[17]

Un autómata finito determinista es una máquina teórica compuesta por un conjunto de estados y un conjunto de transiciones. Estas son una función que, dado un estado del conjunto de estados y un símbolo de un alfabeto, produce otro estado.[17]

Un lenguaje es considerado k-testable si puede ser reconocido por un autómata que únicamente requiere una memoria de tamaño k para realizar el análisis. [18]

En términos formales, esto significa que se cumple lo siguiente para cualquier k:

$$\forall l, l' \in \Sigma^\star, \forall w \in \Sigma^{\geq k}, lw \in L \iff l'w \in L.$$

Para reconocer lenguajes k-testables, se utilizan máquinas específicas conocidas como máquinas k-testables (k-TSS), las cuales son definidas formalmente como una 5-tupla $Z_k = (\Sigma, I, F, T, C)$ donde $k > 0$: [18]

- Σ es un alfabeto finito.
- $I, F \subseteq \Sigma^{k-1}$ son conjuntos de prefijos y sufijos de longitud $k-1$, respectivamente.
- $C \subseteq \Sigma^{<k}$ es un conjunto de cadenas cortas (de longitud menor que k)
- $T \subseteq \Sigma^{k}$ es el conjunto de segmentos permitidos de longitud k

Dada una máquina k-testable, el lenguaje k-testable reconocido por $Z_k = \langle \Sigma, I, F, T, C \rangle$ es $L(Z_k) = \left(I\Sigma^\star \cap \Sigma^\star F - \Sigma^\star(\Sigma^k - T)\Sigma^\star\right) \cup C$. [18]

Esto significa que las únicas cadenas admisibles son aquellas que corresponden exactamente a cadenas que se encuentran en C, o aquellas cuyo prefijo de longitud $k-1$ está en I, cuyo sufijo de longitud $k-1$ está en F y donde todas las subcadenas de longitud k pertenecen a T. Es decir, hay dos tipos de cadenas: aquellas de longitud menor que k y definidas exactamente así en C, y aquellas más largas que k para las cuales $\Sigma^k - T$ define los segmentos prohibidos.

2.1.2. Importancia en la investigación actual

La importancia de los lenguajes k-testables en la investigación actual es notable, especialmente en el contexto de la bioinformática y la genética. Estos lenguajes pueden ser útiles en esta área debido a su capacidad para analizar secuencias genéticas y detectar patrones que pueden estar asociados con la presencia de mutaciones patógenas. [18]

Los lenguajes k-testables permiten la identificación de secuencias específicas de nucleótidos que pueden estar presentes en genomas afectados por ciertas mutaciones. Dado que estos lenguajes son eficaces para manejar grandes cantidades de datos y pueden ser implementados de manera eficiente, son herramientas valiosas para el análisis de secuencias genéticas.

Los algoritmos basados en lenguajes k-testables pueden ayudar a filtrar y focalizar la búsqueda en regiones específicas del genoma, facilitando la identificación de genes patógenos y mejorando la comprensión de la base genética de la enfermedad.

2.2 Algoritmo de inferencia de lenguajes k-Explorables en sentido Estricto

2.2.1. Descripción del Algoritmo

El algoritmo empleado en este proyecto se fundamenta en lenguajes k-explorables en sentido estricto. Los lenguajes k-EE permiten la identificación de patrones dentro de secuencias de símbolos, donde k define la longitud de los fragmentos considerados para la exploración. Estos lenguajes son de particular interés en el análisis de secuencias de ADN porque permiten la construcción de modelos que capturan propiedades estructurales de los datos observados.[17]

2.2.2. Implementación del algoritmo

Para implementar el algoritmo, hemos seguido los siguientes pasos: partiendo de un conjunto de datos primitivos S, se define:

- Σ es el conjunto de símbolos contenidos en las palabras del conjunto S.
- $I_k(S) = \{u \mid uv \in S, |u| = k-1, v \in \Sigma(S)^*\} \cup \{x \in S \mid |x| < k-1\}$
- $F_k(S) = \{v \mid uv \in S, |v| = k-1, v \in \Sigma(S)^*\} \cup \{x \in S \mid |x| < k-1\}$
- $T_k(S) = \{v \mid uvw \in S, |v| = k, u, w \in \Sigma(S)^*\}$

2.2.2.1. Algoritmo de inferencia e lenguajes k-EE

Utilizando los conjuntos inferidos, definimos el siguiente algoritmo de inferencia de lenguajes k-explorables en sentido estricto (k-EE):

Entrada: $k \geq 2, S$
Salida: AFD $A_k = (Q, \Sigma, \delta, q_0, Q_f)$
$(\Sigma, I, F, T) = (\Sigma(S), I_k(S), F_k(S), T_k(S))$;
$Q = \{\lambda\}$; $\delta = \emptyset$; $q_0 = \lambda$;
for all $a_1 \dots a_m \in I$ **do**
  for $j = 1$ **to** m **do**
    $Q = Q \cup \{a_1 \dots a_j\}$;
    $\delta = \delta \cup \{(a_1 \dots a_{j-1}, a_j, a_1 \dots a_j)\}$;
  end for
end for
for all $a_1 \dots a_k \in T$ **do**
  $Q = Q \cup \{a_2 \dots a_k\}$;
  $\delta = \delta \cup \{(a_1 \dots a_{k-1}, a_k, a_2 \dots a_k)\}$;
end for
$Q_f = F$;
$A_k = (Q, \Sigma, \delta, q_0, Q_f)$

Figura 2.1: Algoritmo de inferencia de lenguajes k-EE

2.2.2.2. Ejemplos de aplicación

En primer lugar, procederemos a aplicar el algoritmo descrito en la Figura 2.1 utilizando los parámetros proporcionados. Consideraremos el conjunto de cadenas $S = \{abba, aaabba, bbaaa, bba\}$ con un valor de $k = 2$

1. **Obtención de conjuntos**: En primer lugar, se determinan los conjuntos $\{\Sigma, I_3(S), F_3(S), T_3(S)\}$:
   - $\Sigma = \{a, b\}$
   - $I_3(S) = \{a, b\}$
   - $F_3(S) = \{ab, bb, ba, aa\}$
   - $T_3(S) = \{a\}$
2. **Aplicación del algoritmo**: Con los conjuntos obtenidos, se aplica el algoritmo de la Figura 2.1, obteniendo así el autómata finito determinista (AFD) del lenguaje inferido a partir de la muestra S.

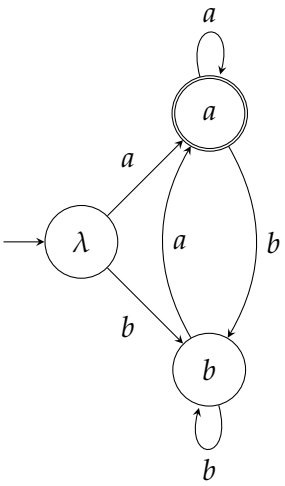


Figura 2.2: Autómata Finito Determinista generado a partir de la muestra S con $k = 2$

A continuación, aplicaremos el mismo algoritmo descrito en la Figura 2.1, pero en esta ocasión modificaremos el valor del parámetro $k = 3$.

1. **Obtención de conjuntos**: Los conjuntos obtenidos para este caso son:
   - $\Sigma = \{a, b\}$
   - $I_3(S) = \{ab, aa, bb\}$
   - $F_3(S) = \{abb, bba, aaa, aab, bba, baa\}$
   - $T_3(S) = \{ba, aa\}$
2. **Aplicación del algoritmo**: Al aplicar el algoritmo nuevamente con los nuevos conjuntos obtenidos obtenemos el siguiente AFD:

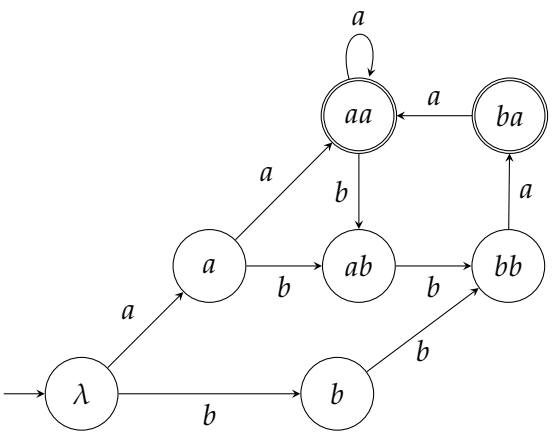


Figura 2.3: Autómata Finito Determinista generado a partir de la muestra S con $k = 3$

A partir de las Figuras 2.2 y 2.3, se pueden extraer varias conclusiones respecto al impacto del valor de k en la construcción de autómatas finitos deterministas (AFD).

Cuando el valor de k es menor, el AFD generado presenta una menor discriminación entre las cadenas. Esto se debe a que el automáta tiene menos estados y transiciones, permitiendo que un mayor número de cadenas compartan estados comunes. Por lo tanto, es más general y menos preciso en su aceptación de cadenas.

Sin embargo, con un valor de k mayor, el AFD generado tiene una mayor capacidad de discriminación. Esto se observa en la mayor cantidad de estados y transiciones específicas, lo que permite al automáta diferenciar mejor entre cadenas similares. Esta mayor precisión resulta en una mayor discriminación, haciendo que el AFD sea más específico en la aceptación de cadenas.

El análisis de los AFDs generados con diferentes valores de k indica que un valor menor de k produce un autómata más general y menos discriminante, mientras que un valor mayor de k resulta en un autómata más específico y estricto en la aceptación de nuevas cadenas.

Esta relación es muy importante para el análisis de secuencias de ADN en este proyecto. Por lo tanto, realizará una experimentación exhaustiva con distintos valores de k para determinar el valor óptimo que balancee adecuadamente la sensibilidad y la especificidad.

CAPÍTULO 3
Preprocesado de datos

3.1 Obtención de muestras S

Para la inferencia de un autómata finito determinista (AFD) a partir de datos genómicos y utilizando el algoritmo basado en k-explorables de la Figura 2.1, es fundamental definir y obtener muestras específicas que permitan entrenarlo.

3.1.1. Definición de las muestras necesarias

Las muestras necesarias se derivan de secuencias de ADN ubicadas en posiciones específicas de los cromosomas, las cuales han sido seleccionadas por su relevancia en la enfermedad rara que se está investigando.

El fichero VCF proporcionado por el *Instituto de Investigación de La Fe de Valencia* contiene mutaciones de pacientes reales que han sido relacionadas con la Retinosis Pigmentaria. Estas secuencias deben tener una longitud de contexto N, lo que implica que cada muestra abarca una región del ADN de tamaño N alrededor de la posición de interés.

3.1.2. Procedimiento de obtención

El procedimiento de obtención de las muestras se llevó a cabo en colaboración con el Instituto de La Fe de Valencia, que proporcionó un archivo VCF con mutaciones directamente relacionadas con la Retinosis Pigmentaria. A partir de este archivo se realizaron los siguientes pasos:

1. **Identificación de Posiciones:** Se identificaron las posiciones específicas en los cromosomas que son relevantes para esta enfermedad. Estas posiciones fueron determinadas en base a estudios previos y a la información proporcionada por el Instituto de La Fe de Valencia.

2. **Extracción de Secuencias:** Para cada posición identificada, se extrajo una secuencia de ADN con una longitud de contexto N. Esto significa que se tomó una región del ADN que incluye la posición de interés y una extensión de nucleótidos a ambos lados de dicha posición, formando una muestra de longitud $N + 1 + N$.

3. **Almacenamiento de Muestras:** Las secuencias extraídas se guardaron en un único fichero. En este fichero se incluyó una muestra para cada posición cromosómica en la que se detectó una mutación, pero sin aplicar la mutación. Estas muestras representan la secuencia de ADN original y tienen una longitud de $N + 1 + N$.

4. **Aplicación de la Mutación:** Para cada mutación identificada en el archivo VCF, se generó una cadena de ADN de longitud N+1+N aplicando dicha mutación (igual que en el paso anterior pero con la mutación aplicada a la cadena) y se almacenó en otro fichero.

5. **Conjunto S para Entrenamiento:** Una vez obtenidos los dos ficheros, se conforman dos conjuntos S diferenciados: uno compuesto por las secuencias originales sin mutaciones y otro por las secuencias con las mutaciones aplicadas. Ambos conjuntos serán empleados para el entrenamiento y testeo del algoritmo de inferencia de AFD, permitiendo evaluar la eficacia del algoritmo tanto en condiciones normales como en presencia de mutaciones.

3.1.3. Ejemplos y casos prácticos

Para ilustrar el proceso de creación del conjunto S, se tomará como ejemplo una posición específica del cromosoma donde se ha detectado una mutación relevante para la Retinosis Pigmentaria, con un contexto $N = 5$.

Supongamos que la posición de interés en el cromosoma es la base 100 (indicada como P).

1. **Identificación de Posiciones:**
   - **Posición 100 (P):** AAGCT**P**AGCTA
2. **Extracción de Secuencias:**
   - **Secuencia Original (sin mutación):** Se extraen 5 nucleótidos antes de P y 5 nucleótidos después de P, formando una secuencia de longitud $N + 1 + N$ = 11 nucleótidos.

     Secuencia Extraída: AAGCT**C**AGCTA
   - **Secuencia con mutación:** Se aplica la mutación en la posición P (el tamaño de la muestra mutada puede variar si la mutación elimina un nucleótido o añade más de un nucleótido)

     Secuencia con Mutación: AAGCT**T**AGCTA
3. **Almacenamiento de Muestras:**
   - **Fichero Original:** Se guarda la secuencia AAGCTCAGCTA en un fichero.
   - **Fichero Con Mutación:** Se guarda la secuencia AAGCTTAGCTA en otro fichero.
4. **Conjunto S para Entrenamiento y Testeo:**
   - **Conjunto S Original:** Contiene la secuencia AAGCTCAGCTA.
   - **Conjunto S con Mutación:** Contiene la secuencia AAGCTTAGCTA.

3.2 Entrenamiento del Modelo

El proceso de entrenamiento del modelo es un paso crucial en la inferencia de un autómata finito determinista (AFD) que pueda reconocer patrones específicos en las secuencias de ADN.

En este estudio, se generaron dos tipos de conjuntos de muestras: uno con las secuencias originales (sin mutaciones) y otro con las secuencias mutadas. Estos conjuntos fueron utilizados para entrenar dos modelos distintos. A continuación, se detalla el procedimiento y las consideraciones llevadas a cabo durante el entrenamiento del modelo.

3.2.1. Uso de muestras mutadas y sin mutar

Para asegurar una evaluación exhaustiva y precisa, se decidió entrenar dos modelos distintos. Uno de los modelos se entrenó con las muestras originales (sin mutaciones), mientras que el otro se entrenó con las muestras mutadas. Este enfoque permite que cada modelo se especialice en reconocer y aceptar (o rechazar) secuencias específicas de acuerdo a su estado mutacional.

1. **Modelo 1: Muestras Originales**
   - Este modelo se entrenó utilizando el conjunto S de secuencias originales sin mutaciones. El objetivo del modelo es reconocer patrones genéticos presentes en el ADN sin alteraciones.
   - Se experimentó con diferentes longitudes de muestra para optimizar el rendimiento del modelo. Inicialmente, se utilizaron longitudes muy extensas, pero se observó que el algoritmo tardaba mucho en ejecutarse debido a la limitada capacidad de procesamiento disponible.
   - Se adoptó una longitud de muestra de 25 nucleótidos antes y 25 nucleótidos después de la posición de interés, $N = 25 + 1 + 25 = 51$.
   - Se realizaron pruebas con distintos valores de k (parámetro que define el tamaño de la ventana de contexto del modelo) para encontrar la configuración óptima. La influencia de este parametro en el modelo se ha definido en el apartado 2.2.2.2.
2. **Modelo 2: Muestras Mutadas**
   - Este modelo se entrenó utilizando el conjunto S de secuencias mutadas. El objetivo es que el modelo reconozca y gestione las secuencias de ADN que contienen mutaciones específicas relacionadas con la Retinosis Pigmentaria.
   - Similar al primer modelo, se probó con distintas longitudes de muestra y valores de k. La configuración final también adoptó una longitud de muestra de $N = 25 + 1 + 25 = 51$.

Finalmente, se han obtenido varias versiones de cada modelo, variando el valor de k, para evaluar y optimizar el rendimiento en función de cada configuración.

3.2.2. Fuente de datos: Archivo VCF del Instituto de La Fe de Valencia

Para el desarrollo de este estudio, la fuente principal de datos fue un archivo en formato VCF proporcionado por el Instituto de La Fe de Valencia.

Este archivo contiene un conjunto exhaustivo de mutaciones genéticas directamente relacionadas con la Retinosis Pigmentaria, recopiladas de pacientes reales.

A continuación, se presenta una descripción detallada del contenido y la estructura de este archivo, así como su relevancia en el contexto del estudio.

3.2.2.1. Formato y Estructura del Archivo VCF

El archivo VCF sigue el estándar VCF versión 4.1, como se especifica en su encabezado. Este formato es ampliamente utilizado en estudios genómicos para almacenar y comunicar variantes genéticas. El archivo contiene metadatos en su encabezado, seguidos por las variantes genéticas anotadas. A continuación, se describen los principales componentes del archivo.

1. **Encabezado del Archivo VCF:** El encabezado del archivo incluye varias líneas que comienzan con '##' y una línea que comienza con '#'. Estas líneas proporcionan información interesante sobre el formato del archivo y las variantes que contiene:
    - **##fileformat=VCFv4.1:** Indica que el archivo sigue el estándar VCF versión 4.1.
    - **##fileDate=2023-12-17:** La fecha de creación del archivo.
    - **##source=ClinVar:** La fuente de las variantes, en este caso, la base de datos *ClinVar*.
    - **##reference=GRCh38:** La referencia del genoma humano utilizada para alinear y anotar las variantes, en este caso, GRCh38.
    - **##ID=<Description=ClinVar Variation ID>:** Descripción del ID de la variación en *ClinVar*.
2. **Campos Principales del Archivo VCF:** La última línea del encabezado que comienza con '#' describe los campos principales de cada variante en el archivo:
    - **CHROM:** El cromosoma en el que se encuentra la variante.
    - **POS:** El cromosoma en el que se encuentra la variante.
    - **ID:** Identificador único de la variante.
    - **REF:** Alelo de referencia.
    - **ALT:** Alelo alternativo (mutado).
    - **QUAL:** Calidad de la variante.
    - **FILTER:** Filtros aplicados a la variante.

Para finalizar este apartado, se presenta una tabla que ilustra cinco ejemplos de mutaciones presentes en un archivo VCF:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER
1	69134	2205837	A	G	.	.
3	136255988	511286	ATGG	A	.	.
5	141485494	1687021	GT	G	.	.
15	90809213	405304	G	A	.	.
X	129589971	2059109	T	G	.	.

Tabla 3.1: Ejemplos de mutaciones en el archivo VCF

3.3 Optimización del Entrenamiento

El proceso de entrenamiento en algoritmos de inferencia de lenguajes k-explorables en sentido estricto (k-EE) (Figura 2.1) puede enfrentarse a diversos desafíos, siendo uno de los más significativos el cuello de botella en el conjunto T .

Este problema surge debido a la complejidad y el tamaño del conjunto T , que contiene todas las posibles concatenaciones de los símbolos en las palabras del conjunto S hasta una longitud k. Este cuello de botella impacta directamente en el rendimiento del algoritmo y, por ende, en la eficiencia del proceso de entrenamiento.

Durante el entrenamiento del algoritmo, uno de los principales problemas es la gestión eficiente de los conjuntos inferidos $I_k(S)$, $F_k(S)$, y $T_k(S)$.

El conjunto $T_k(S)$, en particular, puede crecer exponencialmente con el tamaño del alfabeto y la longitud de las palabras, haciendo que las operaciones de búsqueda y agregación se vuelvan prohibitivamente costosas en términos de tiempo de cómputo y memoria. Esto crea un cuello de botella que ralentiza significativamente el proceso de entrenamiento y limita la escalabilidad del algoritmo.

Para calcular $T_k(S)$, se toma cada palabra contenida en el conjunto S y se divide en todas las posibles combinaciones de tres partes. Luego, se agregan a $T_k(S)$ aquellas divisiones donde la segunda parte tiene una longitud k y las primeras y terceras partes pertenecen al alfabeto $\Sigma(S)^*$.

Necesidad de optimización

Para abordar estos desafíos, es esencial implementar técnicas de optimización que permitan mejorar el rendimiento del algoritmo. La optimización no solo ayuda a reducir el tiempo de cómputo, sino que también permite manejar conjuntos de datos más grandes, lo que se traduce en modelos más precisos y robustos. Entre las herramientas disponibles para la optimización se encuentran la paralelización de procesos y el uso de estructuras de datos eficientes.

La justificación para elegir estas técnicas se basa en sus beneficios comprobados en términos de rendimiento y eficiencia. La paralelización mediante hilos permite distribuir la carga de trabajo en múltiples núcleos del procesador, aprovechando al máximo los recursos de hardware disponibles. Por otro lado, el uso de estructuras de datos eficientes, como conjuntos y listas en Python, optimiza las operaciones de búsqueda y agregación, reduciendo significativamente el tiempo de ejecución.

3.3.1. Paralelización de procesos mediante hilos

Introducción a los hilos

Un hilo de ejecución es la unidad más pequeña de procesamiento que puede ser manejada de manera independiente por un sistema operativo. Los hilos permiten la ejecución concurrente de múltiples tareas dentro de un mismo programa, lo que puede mejorar significativamente el rendimiento, especialmente en sistemas multicore. [19]

Justificación del Uso de Hilos

En el contexto del algoritmo de inferencia de lenguajes k-EE, la generación del conjunto $T_k(S)$ es una tarea que puede ser paralelizada. Esto se debe a que $T_k(S)$ se puede dividir en múltiples subconjuntos independientes, que luego pueden ser procesados en paralelo y unidos posteriormente. La paralelización de este proceso permite distribuir la carga de trabajo en los diferentes núcleos del procesador, reduciendo así el tiempo total de cómputo. [20]

Implementar hilos de ejecución en la generación del conjunto $T_k(S)$ ha demostrado ser una estrategia eficaz para acelerar el entrenamiento del algoritmo. Al dividir el trabajo en n hilos, donde n es el número de núcleos disponibles en el procesador, se logra una aceleración significativa en el tiempo de procesamiento. Esto es especialmente beneficioso en sistemas con múltiples núcleos, donde la computación paralela puede ser aprovechada al máximo.

3.3.2. Estructuras de datos eficientes

Importancia de las Estructuras de Datos

El uso de estructuras de datos eficientes es crucial para optimizar el rendimiento del algoritmo. En Python, las estructuras de datos como los conjuntos (set) y las listas (list) ofrecen operaciones de búsqueda y agregación rápidas, respectivamente. Los conjuntos permiten búsquedas en tiempo constante, mientras que las listas son eficientes para agregar elementos.

Uso de Conjuntos para Búsquedas Rápidas

En el algoritmo de inferencia de lenguajes k-EE, los conjuntos pueden utilizarse para almacenar $I_k(S)$, $F_k(S)$, y $T_k(S)$. Esto es especialmente útil cuando se necesita verificar la pertenencia de un elemento a uno de estos conjuntos. [21]

Uso de Listas para Agregaciones Rápidas

Las listas son ideales para operaciones de agregación debido a su eficiencia en la adición de elementos. En el contexto del algoritmo, se pueden utilizar listas para construir transiciones y estados de manera eficiente.[22]

Gracias a la aplicación de técnicas de optimización como la paralelización mediante hilos y el uso de estructuras de datos eficientes, se ha logrado mejorar significativamente el rendimiento del algoritmo de inferencia de lenguajes k-EE.

Estas optimizaciones permiten manejar conjuntos de datos más grandes y complejos, acelerando el proceso de entrenamiento y produciendo resultados más precisos y significativos. La combinación de computación paralela y estructuras de datos adecuadas ha sido clave para desarrollar el algoritmo de la manera más eficiente posible.

CAPÍTULO 4
Evaluación del Modelo

4.1 Proceso de Testing

En el desarrollo de mi Trabajo de Fin de Grado, se ha explorado el uso de algoritmos basados en k-explorables para inferir autómatas finitos deterministas (AFD) a partir de conjuntos de muestras genómicas. El algoritmo utilizado es el descrito en la Figura 2.1.

Las muestras han sido obtenidas gracias a un archivo VCF (Variant Call Format) que recoge todas las mutaciones relacionadas con la enfermedad rara conocida como retinitis pigmentaria. Como se ha explicado en apartados anteriores, este archivo VCF proporciona una base de datos detallada de las variaciones genéticas relacionadas con la enfermedad que estamos estudiando.

Para llevar a cabo el proceso de testing, se han desarrollado dos modelos distintos: uno basado en muestras sin mutar y otro en muestras mutadas. La experimentación se ha centrado en observar cómo varía el rendimiento del modelo al modificar el parámetro k del algoritmo de k-explorables, que determina el grado de exploración del autómata generado. Este enfoque permite ajustar el balance entre generalización y especialización del modelo, optimizando su capacidad para discriminar entre muestras mutadas y no mutadas.

A continuación, se detalla el procedimiento de testing, las métricas utilizadas para evaluar el rendimiento del modelo, y el cálculo de la precisión (accuracy) del mismo.

4.1.1. Procedimiento de testing

Para el modelo de muestras sin mutar, se han utilizado muestras originales, y para el modelo de muestras mutadas, se han utilizado las muestras mutadas. Estas muestras se han sometido a análisis mediante el algoritmo de k-explorables. Este algoritmo permite generar un autómata finito determinista (AFD) que puede generalizar o especializar en función del valor de k. Se han probado diferentes valores de k para observar cómo afecta a la capacidad del modelo para aceptar o rechazar muestras, incluyendo las mutadas.

Se han utilizado muestras de tamaño $N = 25 + 1 + 25$. En total, se han entrenado 48 modelos (AFDs), variando el valor de k desde 2 hasta 25 y utilizando ambos conjuntos de datos (muestras mutadas y sin mutar) para observar la evolución de los resultados. Además, para facilitar la visualización de estos resultados, se ha desarrollado una aplicación gráfica que muestra estos resultados de manera interactiva.

Para la evaluación de los modelos, se ha seguido el siguiente procedimiento: se partió de dos conjuntos de datos, uno sin mutar y otro mutado. De cada conjunto, se seleccionaron 500,000 muestras para test, haciendo un total de 1,000,000 de muestras para el conjunto de test. Estas muestras no fueron vistas por el modelo durante el entrenamiento, asegurando así una evaluación imparcial y robusta de su rendimiento.

A continuación, se muestra una imagen explicativa del proceso, donde se ilustran los pasos seguidos y la estructura del procedimiento de testing (modelo mutado):

Figura 4.1: Proceso de testing y evaluación del modelo

4.1.2. Métricas utilizadas

En el análisis del rendimiento de un modelo de clasificación, como el desarrollado en este trabajo, es esencial utilizar diversas métricas que permitan evaluar su eficacia y precisión de manera exhaustiva.

Las principales métricas empleadas en este estudio son la precisión (precision), el recall (sensibilidad), la especificidad y la precisión global (accuracy). Para comprender adecuadamente estas métricas, es fundamental definir los conceptos como verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos.[23]

- **Verdaderos Positivos (TP):** Un verdadero positivo se refiere a aquellas muestras que el modelo ha identificado correctamente como pertenecientes a la clase positiva (en el caso de la Figura 4.1, muestras mutadas). Por ejemplo, si el modelo predice que una muestra está mutada y efectivamente esta muestra está mutada, se considera un verdadero positivo.[23]
- **Falsos Positivos (FP):** Un falso positivo ocurre cuando el modelo clasifica incorrectamente una muestra como perteneciente a la clase positiva cuando, en realidad, pertenece a la clase negativa (en el caso de la Figura 4.1, muestras no mutadas). Por ejemplo, si el modelo predice que una muestra está mutada pero en realidad no lo está, se considera un falso positivo. Este tipo de error es crítico porque puede llevar a falsas alarmas y tratamientos innecesarios.[23]
- **Verdaderos Negativos (TN):** Un verdadero negativo se refiere a aquellas muestras que el modelo ha identificado correctamente como pertenecientes a la clase negativa (en el caso de la Figura 4.1, muestras no mutadas). Por ejemplo, si el modelo predice que una muestra no está mutada y efectivamente esta muestra no está mutada, se considera un verdadero negativo.[23]
- **Falsos Negativos (FN):** Un falso negativo ocurre cuando el modelo clasifica incorrectamente una muestra como perteneciente a la clase negativa cuando, en reali-

dad, pertenece a la clase positiva (en el caso de la Figura 4.1, muestras mutadas). Por ejemplo, si el modelo predice que una muestra no está mutada pero en realidad sí lo está, se considera un falso negativo. Este tipo de error es especialmente preocupante en diagnósticos médicos, ya que puede llevar a la falta de tratamiento adecuado.[23]

Estas definiciones permiten calcular las siguientes métricas:

1. **Precision (Precisión):** La precisión mide la proporción de verdaderos positivos entre todas las muestras que el modelo ha clasificado como positivas. Se calcula de la siguiente manera:
$$Precision = \frac{TP}{TP + FP}$$
Una alta precisión indica que el modelo tiene una baja tasa de falsos positivos.[23]

2. **Recall (Sensibilidad):** El recall, o sensibilidad, mide la proporción de verdaderos positivos entre todas las muestras que son realmente positivas. Se calcula de la siguiente manera:
$$Recall = \frac{TP}{TP + FN}$$
Un alto recall indica que el modelo tiene una baja tasa de falsos negativos, lo cual es crucial en contextos donde es importante detectar todos los casos positivos.[23]

3. **Specifity (Especifidad):** La especificidad mide la proporción de verdaderos negativos entre todas las muestras que son realmente negativas. Se calcula de la siguiente manera:
$$Specifity = \frac{TN}{TN + FP}$$
Una alta especificidad indica que el modelo tiene una baja tasa de falsos positivos.[23]

4. **Precisión Global (Accuracy):** La precisión global mide la proporción de todas las muestras correctamente clasificadas (tanto positivas como negativas) entre el total de muestras [23]. Se calcula de la siguiente manera:
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Estas métricas ofrecen una visión completa del rendimiento del modelo, permitiendo identificar sus puntos fuertes y débiles. En el contexto de este estudio, donde se busca distinguir entre muestras genéticas mutadas y no mutadas, es esencial considerar todas estas métricas para evaluar adecuadamente la eficacia del modelo.

4.2 Resultados obtenidos

En este capítulo se presentan y analizan los resultados obtenidos del modelo de Autómata Finito Determinista (AFD) entrenado mediante un algoritmo basado en k-explorables. Se pretende comparar el desempeño del modelo en dos escenarios distintos: con muestras de ADN mutado y no mutado. La evaluación se centrará en cómo el modelo maneja y clasifica estos fragmentos de ADN, permitiendo observar las diferencias y la efectividad del AFD en la identificación de mutaciones relacionadas con la enfermedad genética retinosis pigmentaria.

La importancia de los resultados en este capítulo radica en su capacidad para validar la eficacia del AFD desarrollado. Dado que las muestras de ADN utilizadas provienen de posiciones específicas asociadas a mutaciones relacionadas con la retinosis pigmentaria, los resultados proporcionan una visión crítica sobre el potencial del modelo para identificar correctamente estas mutaciones. Esto no solo tiene implicaciones directas para el diagnóstico y estudio de la retinosis pigmentaria, sino que también aporta valor al campo de la bioinformática al demostrar la aplicabilidad de algoritmos basados en k-explorables en el análisis de secuencias genéticas. Así, los hallazgos de este capítulo contribuirán a establecer la robustez y precisión del modelo propuesto, facilitando futuras investigaciones y aplicaciones en el ámbito de las enfermedades genéticas.

4.2.1. Comparación entre modelos (mutado vs no mutado)

Como se ha mencionado anteriormente, para asegurar una evaluación exhaustiva y precisa de los resultados obtenidos, se ha decidido entrenar y comparar dos modelos distintos: uno con muestras de ADN sin mutaciones y otro con muestras de ADN mutadas.

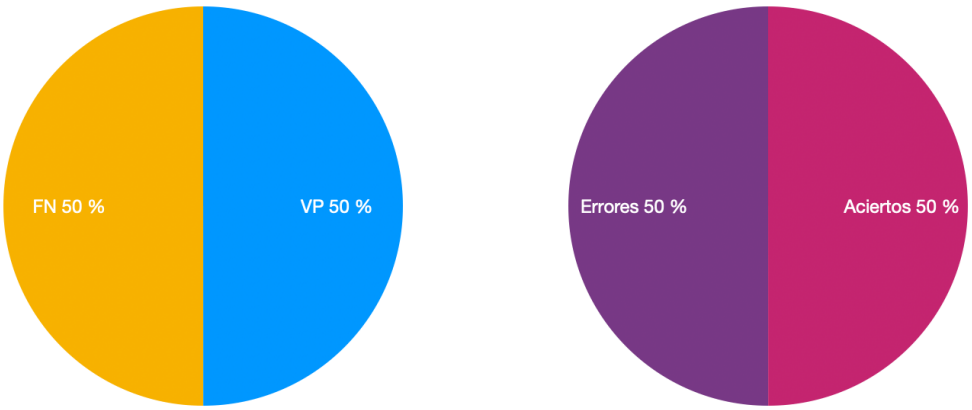
El modelo no mutado se entrenó utilizando el conjunto de secuencias originales sin mutaciones. Su principal objetivo es identificar patrones genéticos presentes en el ADN sin alteraciones. Se adoptaron diversas estrategias para optimizar el rendimiento del modelo, incluyendo la experimentación con diferentes longitudes de muestra y valores de k, que es el parámetro que define el tamaño de la ventana de contexto del modelo.

El modelo mutado, por otro lado, se entrenó con secuencias de ADN que contienen mutaciones específicas relacionadas con la retinosis pigmentaria. Este modelo está diseñado para reconocer y gestionar estas mutaciones. Similar al modelo no mutado, se utilizaron distintas longitudes de muestra y valores de k para asegurar una configuración óptima.

Ambos modelos se evaluaron utilizando la misma longitud de muestra de 51 nucleótidos, comprendiendo 25 nucleótidos antes y 25 después de la posición de interés, más la posición en sí. Esto asegura una base común para la comparación de los resultados y permite evaluar la eficacia de cada modelo en su respectivo dominio de especialización.

4.2.1.1. Resultados obtenidos

Modelo entrenado con Muestras Mutadas



Modelo Mutado con k=5

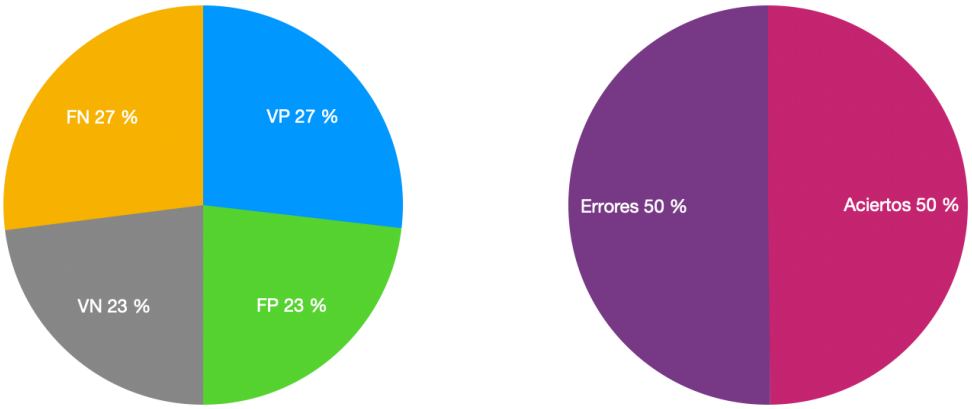
	Mutado	Sano
Mutado	499.901	99
Sano	499.993	7

Figura 4.2: Representación de los resultados obtenidos con el Modelo entrenado con Muestras Mutadas con $k = 5$

En el gráfico correspondiente al modelo entrenado con muestras mutadas con $k = 5$, se observa una distribución equitativa entre verdaderos positivos (VP) y falsos negativos (FN), ambos representando el 50 % de las predicciones. Esto indica que el modelo tiene una capacidad igual para identificar correctamente las secuencias mutadas y fallar en la detección de las mismas.

En términos de precisión general, el modelo presenta una tasa de aciertos del 50 % y una tasa de errores del 50 %, lo que sugiere que las predicciones correctas y las incorrectas están equilibradas. La tabla de confusión proporciona detalles adicionales: el modelo predice correctamente 499,901 secuencias mutadas como mutadas, mientras que clasifica incorrectamente 99 secuencias mutadas como sanas.

Por otro lado, predice erróneamente 499,993 secuencias sanas como mutadas y clasifica correctamente solo 7 secuencias sanas como sanas. Estos resultados revelan una alta tasa de falsos positivos, lo que indica que el modelo tiene dificultades significativas para diferenciar entre secuencias mutadas y no mutadas con precisión, a pesar de mantener un equilibrio general en la tasa de aciertos y errores.



Modelo Mutado con k=11

	Mutado	Sano
Mutado	268.425	231.575
Sano	270.132	229.868

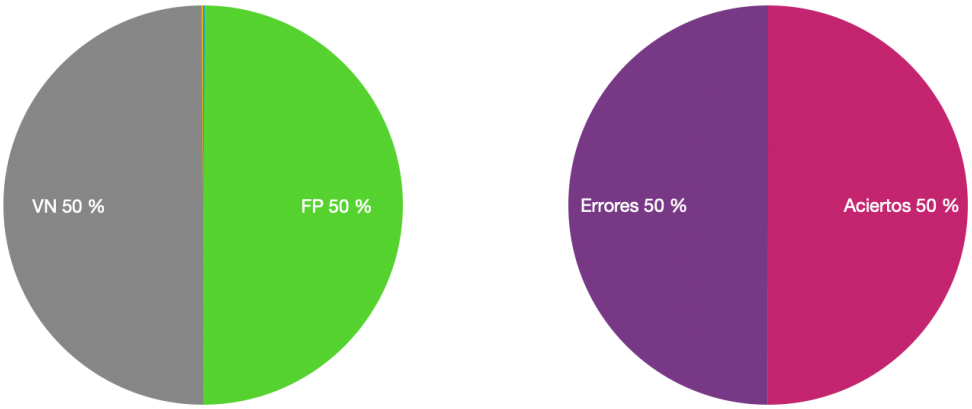
Figura 4.3: Representación de los resultados obtenidos con el Modelo entrenado con Muestras Mutadas con $k = 11$

En el gráfico correspondiente al modelo mutado con $k = 11$, se observa una distribución de predicciones donde los verdaderos positivos (VP) y los falsos negativos (FN) representan cada uno el 27 % de las predicciones, mientras que los verdaderos negativos (VN) y los falsos positivos (FP) representan cada uno el 23 %.

Esto sugiere que el modelo tiene una distribución más equilibrada en todas las categorías de predicción en comparación con $k = 5$, aunque sigue mostrando una significativa tasa de error. La precisión general del modelo es del 50 %, con una tasa de aciertos y errores igualmente dividida.

La tabla de confusión muestra que el modelo predice correctamente 268,425 secuencias mutadas como mutadas, pero clasifica incorrectamente 231,575 secuencias mutadas como sanas. Asimismo, clasifica incorrectamente 270,132 secuencias sanas como mutadas y predice correctamente 229,868 secuencias sanas como sanas.

Estos resultados indican que, aunque la distribución de las predicciones es más balanceada, el modelo sigue enfrentando desafíos importantes en la correcta identificación de secuencias mutadas y no mutadas, reflejando una tasa de falsos positivos y falsos negativos que aún necesita ser mejorada para lograr una mayor precisión.



Modelo Mutado con k=14

	Mutado	Sano
Mutado	1.441	498.559
Sano	1.271	498.729

Figura 4.4: Representación de los resultados obtenidos con el Modelo entrenado con Muestras Mutadas con $k = 14$

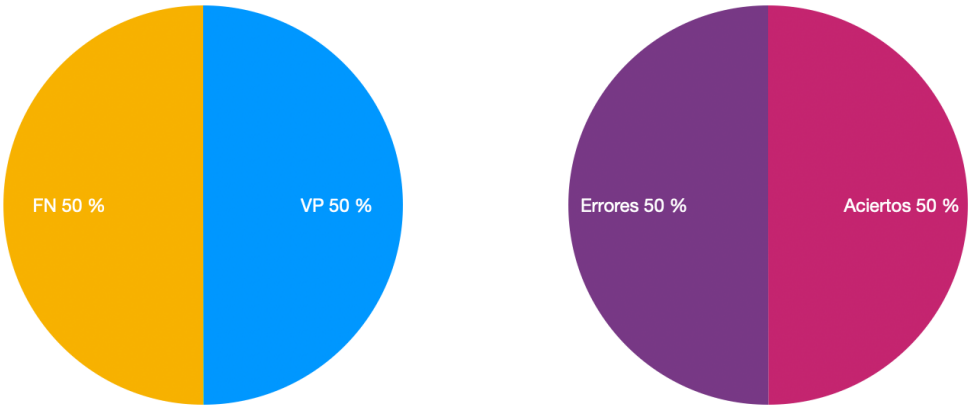
En el gráfico correspondiente al modelo mutado con $k = 14$ (Figura 4.4), se observa una distribución de predicciones donde los verdaderos negativos (VN) y los falsos positivos (FP) representan cada uno el 50 %.

Esto indica que el modelo tiene dificultades para diferenciar correctamente entre secuencias mutadas y sanas. La tasa de aciertos y errores está equilibrada al 50 %, mostrando que el modelo tiene una capacidad igual para realizar predicciones correctas e incorrectas.

La tabla de confusión revela más detalles: el modelo predijo correctamente 1,441 secuencias mutadas como mutadas y 498,729 secuencias sanas como sanas. Sin embargo, clasificó incorrectamente 498,559 secuencias sanas como mutadas y 1,271 secuencias mutadas como sanas.

Estos resultados indican una baja precisión en la identificación de secuencias mutadas, con una alta proporción de falsos positivos. Esto sugiere que el modelo con $k = 13$ tiene una significativa dificultad para distinguir entre secuencias mutadas y sanas, resultando en una alta tasa de error en la clasificación de estas secuencias.

Modelo entrenado con Muestras Sin Mutar



Modelo Mutado con k=5

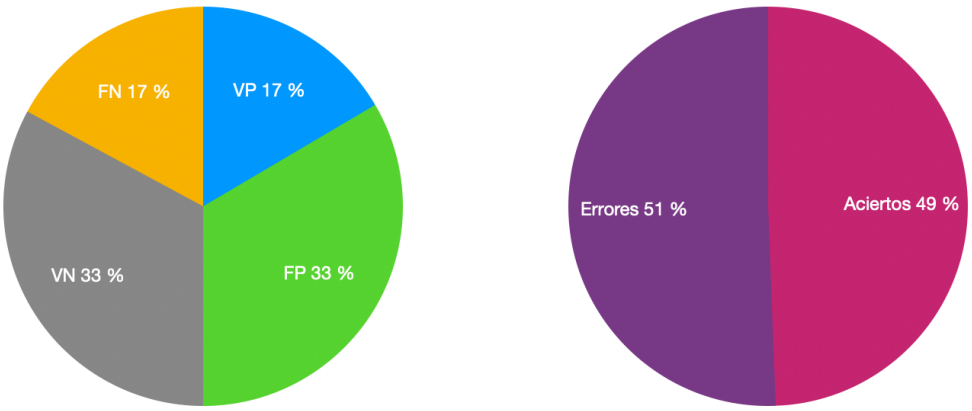
	Sano	Mutado
Sano	499.597	403
Mutado	499.993	7

Figura 4.5: Representación de los resultados obtenidos con el Modelo entrenado con Muestras Sin Mutar con $k = 5$

En el gráfico correspondiente al modelo mutado con $k = 5$ (Figura 4.5), se observa una distribución equitativa entre verdaderos positivos (VP) y falsos negativos (FN), cada uno representando el 50 % de las predicciones. Este equilibrio indica que el modelo tiene una capacidad igual para identificar correctamente las secuencias sanas y para detectar una secuencia sana como mutada. La tasa de aciertos y errores también está equilibrada al 50 %, reflejando que el modelo es igualmente probable de realizar predicciones correctas e incorrectas.

La tabla de confusión proporciona más detalles sobre el rendimiento del modelo: se predijeron correctamente 7 secuencias mutadas como mutadas, mientras que 403 secuencias sanas fueron incorrectamente clasificadas como mutadas. Por otro lado, 499,597 secuencias sanas fueron correctamente identificadas, y 499,993 secuencias mutadas fueron incorrectamente clasificadas como sanas.

Estos resultados indican que el modelo tiene una precisión extremadamente alta en la identificación de secuencias sanas, pero falla significativamente en la detección de secuencias mutadas, lo que se refleja en el elevado número de falsos negativos y falsos positivos.



Modelo Mutado con k=11

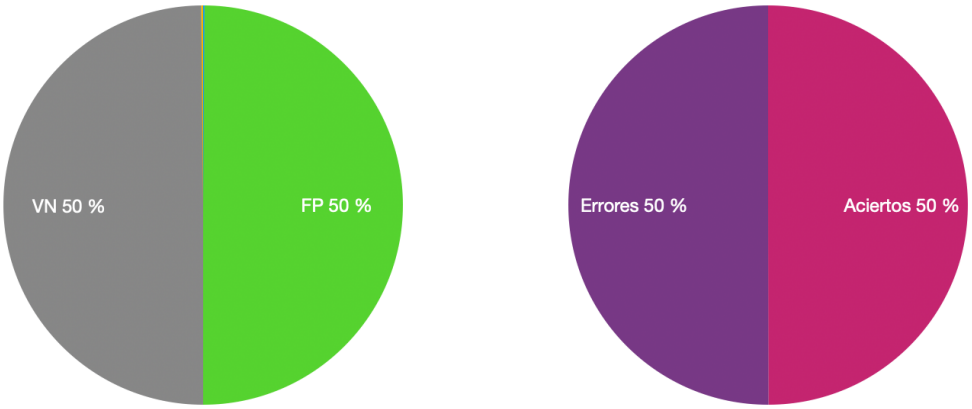
	Sano	Mutado
Sano	165.156	334.844
Mutado	171.254	328.746

Figura 4.6: Representación de los resultados obtenidos con el Modelo entrenado con Muestras Sin Mutar con $k = 14$

En el gráfico correspondiente al modelo mutado con $k = 11$ (Figura 4.6), se observa una distribución de predicciones donde los verdaderos positivos (VP) y los falsos negativos (FN) representan cada uno el 17 % de las predicciones, mientras que los verdaderos negativos (VN) y los falsos positivos (FP) representan cada uno el 33 %. Esto indica que el modelo tiene una mejor capacidad para identificar secuencias sanas (VN) en comparación con secuencias mutadas (VP), pero aún tiene un equilibrio significativo entre las predicciones correctas e incorrectas.

La tasa de aciertos y errores está casi equilibrada, con un 49 % de aciertos y un 51 % de errores, lo que refleja un ligero sesgo hacia las predicciones incorrectas. La tabla de confusión proporciona más detalles sobre el rendimiento del modelo: se predijeron correctamente 328,746 secuencias mutadas como mutadas y 165,156 secuencias sanas como sanas. Sin embargo, 334,844 secuencias sanas fueron incorrectamente clasificadas como mutadas, y 171,254 secuencias mutadas fueron incorrectamente clasificadas como sanas.

Estos resultados indican que el modelo tiene una precisión moderada en la identificación de secuencias mutadas y sanas, pero todavía presenta una alta tasa de falsos positivos y falsos negativos.



Modelo Mutado con k=14

	Sano	Mutado
Sano	1.149	498.851
Mutado	1.579	498.421

Figura 4.7: Representación de los resultados obtenidos con el Modelo entrenado con Muestras Sin Mutar con $k = 11$

En el gráfico correspondiente al modelo mutado con $k = 14$ (Figura 4.7), se observa una distribución de predicciones donde los verdaderos negativos (VN) y los falsos positivos (FP) representan cada uno el 50 %. Esto indica que el modelo tiene dificultades significativas para diferenciar correctamente entre secuencias sanas y mutadas, ya que predice un número igual de secuencias sanas correctamente y de secuencias sanas incorrectamente como mutadas. La tasa de aciertos y errores está equilibrada al 50 %, lo que refleja que el modelo tiene una capacidad igual para realizar predicciones correctas e incorrectas.

La tabla de confusión proporciona más detalles sobre el rendimiento del modelo: se predijeron correctamente 498,421 secuencias mutadas como mutadas y 1,149 secuencias sanas como sanas. Sin embargo, 498,851 secuencias sanas fueron incorrectamente clasificadas como mutadas, y 1,579 secuencias mutadas fueron incorrectamente clasificadas como sanas.

Estos resultados indican que el modelo tiene una precisión extremadamente baja en la identificación de secuencias sanas, lo que se refleja en el elevado número de falsos positivos y verdaderos negativos.

4.2.2. Análisis de los resultados

4.2.2.1. Aplicación de las métricas (Modelo entrenado con Muestras Mutadas)

Métrica	k=5	k=11	k=13
Precisión	0.999802	0.536850	0.002882
Recall	0.499954	0.498415	0.531342
Especificidad	0.066038	0.498150	0.500085
Precisión Global	0.499908	0.498293	0.500170

Tabla 4.1: Medidas de rendimiento de los modelos entrenados con las muestras mutadas con diferentes valores de k

Los resultados obtenidos a partir de los modelos mutados con diferentes valores de k muestran una relación inversa entre precisión y especificidad a medida que aumenta el valor de k. Para $k = 5$, el modelo presenta una alta precisión (0.999802), lo que indica una baja tasa de falsos positivos. Sin embargo, la especificidad es extremadamente baja (0.066038), lo que sugiere una alta tasa de falsos negativos y una pobre capacidad del modelo para identificar correctamente las secuencias sanas.

A medida que el valor de k incrementa, observamos una disminución significativa en la precisión del modelo. Para $k = 11$, la precisión cae a 0.536850 y para $k = 13$ a 0.002882. Esta disminución drástica en la precisión indica que el modelo es menos confiable en la identificación correcta de las secuencias mutadas, incrementando la proporción de falsos positivos.

En contraste, la especificidad mejora notablemente con valores más altos de k. Para $k = 11$, la especificidad aumenta a 0.498150 y para $k = 13$ alcanza 0.500085, lo que implica una mejor capacidad del modelo para distinguir correctamente las secuencias sanas. Este aumento en la especificidad con valores más altos de k sugiere que el modelo es más eficaz en la reducción de falsos positivos, aunque esto se logra a expensas de una menor precisión general.

El recall, que mide la proporción de verdaderos positivos correctamente identificados, se mantiene relativamente constante alrededor del 50 % para $k = 5$ y $k = 11$, pero mejora ligeramente a 0.531342 para $k = 13$. Esto indica que el modelo con $k = 13$ es ligeramente mejor en la detección de secuencias mutadas, aunque con una precisión significativamente reducida.

La precisión global, que refleja la proporción de todas las muestras correctamente clasificadas, se mantiene cercana al 50 % para todos los valores de k. Esto indica que, en términos generales, la capacidad del modelo para realizar predicciones correctas e incorrectas se mantiene constante independientemente del valor de k.

La elección del valor de k tiene un impacto significativo en el rendimiento del modelo. Un k más bajo ofrece una alta precisión pero muy baja especificidad, lo que puede ser útil en contextos donde la minimización de falsos positivos es crucial. Por otro lado, un k más alto mejora la especificidad y la capacidad del modelo para identificar correctamente las secuencias sanas, pero con una precisión reducida. La selección del valor óptimo de k debe equilibrar la necesidad de detectar mutaciones con precisión y la necesidad de minimizar los falsos positivos, dependiendo del contexto específico y los objetivos del análisis genético. Este balance es crucial para aplicaciones prácticas en el diagnóstico y estudio de enfermedades genéticas como la retinosis pigmentaria, donde la precisión y la especificidad tienen implicaciones directas en la efectividad del diagnóstico y la investigación.

4.2.2.2. Aplicación de las métricas (Modelo entrenado con Muestras Sin Mutar)

Métrica	k=5	k=11	k=14
Precisión	0.999194	0.330912	0.002298
Recall	0.49998	0.490937	0.421188
Especificidad	0.01707	0.495405	0.499784
Precisión Global	0.499604	0.493902	0.49957

Tabla 4.2: Medidas de rendimiento de los modelos entrenados con las muestras no mutadas con diferentes valores de k

El rendimiento del modelo entrenado con muestras sin mutar varía significativamente con diferentes valores de k. Para $k = 5$, el modelo muestra una alta precisión pero baja especificidad, lo que indica una buena capacidad para evitar falsos positivos pero una alta tasa de falsos negativos. A medida que k aumenta, la precisión disminuye drásticamente mientras que la especificidad mejora, alcanzando un balance más equilibrado en $k = 11$ y $k = 14$. La precisión global se mantiene cerca del 50 % para todos los valores de k, sugiriendo un desempeño equilibrado en términos de aciertos y errores. La elección del valor de k debe equilibrar la precisión y la especificidad según los objetivos del análisis genético.

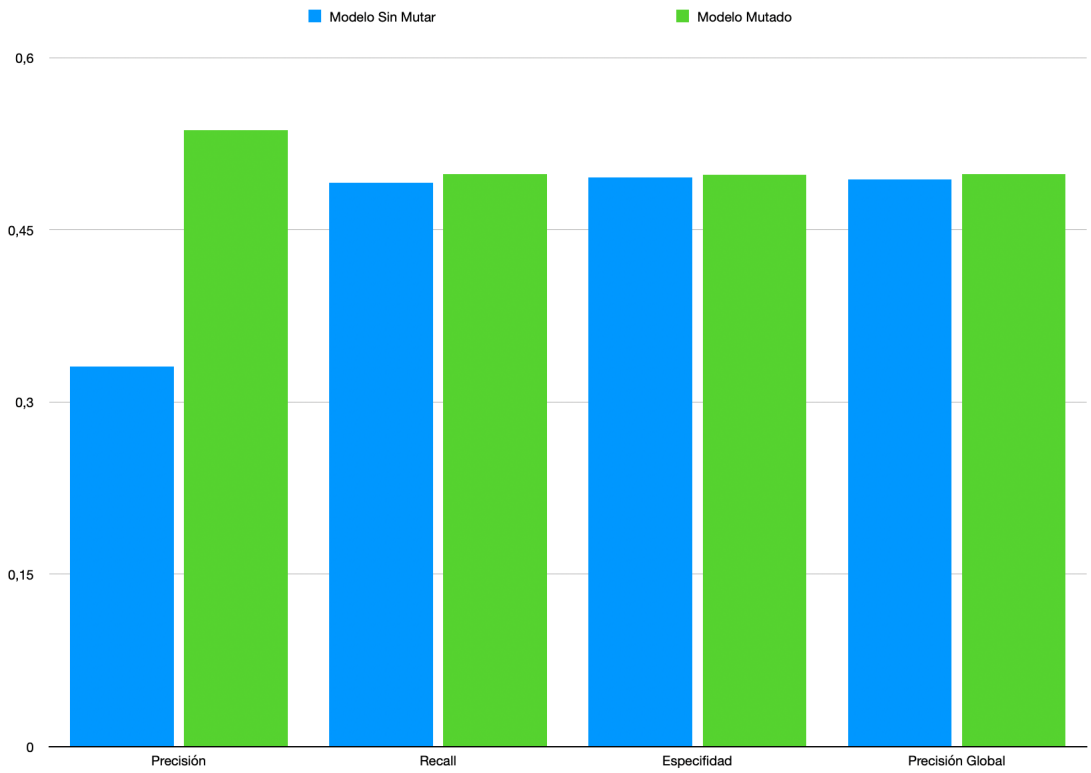


Figura 4.8: Comparación de métricas de rendimiento entre el modelo entrenado con muestras sin mutar y el modelo entrenado con muestras mutadas. Se muestran las métricas de precisión, recall, especificidad y precisión global para cada modelo con un valor de $k = 11$.

El gráfico comparativo muestra el rendimiento de los modelos entrenados con muestras sin mutar y mutadas en términos de cuatro métricas clave: precisión, recall, especificidad y precisión global. En términos de precisión, el modelo entrenado con muestras mutadas supera al modelo entrenado con muestras sin mutar, indicando que tiene una mayor capacidad para clasificar correctamente las muestras positivas. Este resultado sugiere que el modelo mutado es más efectivo en la reducción de falsos positivos.

Por otro lado, el recall del modelo mutado también es superior, lo que implica que este modelo es mejor en la identificación de verdaderos positivos, reduciendo así los falsos negativos. En cuanto a la especificidad, ambos modelos muestran un rendimiento casi idéntico, lo que indica que tienen una capacidad similar para identificar correctamente las muestras negativas. Finalmente, la precisión global de ambos modelos es comparable, lo que refleja un equilibrio general en su capacidad de clasificación.

CAPÍTULO 5
Conclusiones

5.1 Conclusiones generales del estudio

Este estudio ha explorado la utilización de modelos basados en autómatas finitos deterministas (AFD) creados a partir de algoritmos basados en k-explorables para identificar mutaciones genéticas asociadas con la retinosis pigmentaria. A pesar de que los resultados obtenidos no fueron óptimos, este trabajo ha sido valioso para investigar nuevas formas de abordar el problema. Se desarrollaron y compararon dos modelos: uno entrenado con muestras de ADN sin mutaciones y otro con muestras mutadas. Los resultados indican que el modelo entrenado con muestras mutadas tuvo un desempeño relativamente mejor en términos de precisión y recall. Sin embargo, ambos modelos presentaron tasas significativas de falsos positivos y falsos negativos, lo que subraya la necesidad de mejorar las técnicas de modelado para obtener resultados más fiables.

A lo largo del estudio, se observó que la precisión del modelo variaba considerablemente con el valor de k, el parámetro que define el tamaño de la ventana de contexto. Valores de k más bajos mejoraron la precisión pero redujeron la especificidad, mientras que valores más altos de k tuvieron el efecto contrario. Esto sugiere que el ajuste del parámetro k es crucial para equilibrar la precisión y la especificidad del modelo, lo que es fundamental para su aplicabilidad en la detección de mutaciones genéticas. Aunque los resultados no fueron excepcionales, el estudio proporciona una base importante para futuras investigaciones y aplicaciones en el campo de la bioinformática y la genética.

5.2 Limitaciones del estudio

El estudio presenta varias limitaciones que deben ser abordadas en futuras investigaciones. En primer lugar, una limitación importante fue la tasa de falsos positivos y falsos negativos observada en los resultados. Los modelos mostraron dificultades para diferenciar correctamente entre secuencias mutadas y no mutadas, resultando en una cantidad significativa de errores de clasificación. Estos errores son críticos en aplicaciones médicas, donde un falso negativo podría llevar a la falta de tratamiento adecuado y un falso positivo podría causar ansiedad innecesaria y tratamientos inapropiados. Esta tasa de error sugiere que los modelos necesitan mejoras sustanciales para ser viable en un entorno clínico.

Además, debido a limitaciones técnicas y de recursos, no se pudieron entrenar modelos más grandes ni experimentar con valores de k superiores. Esta restricción impidió evaluar si un aumento en el tamaño del modelo o en el parámetro k podría haber mejorado significativamente los resultados.

Los resultados obtenidos no fueron óptimos, el estudio proporciona una base valiosa para investigar nuevas formas de abordar la identificación de mutaciones genéticas, destacando áreas clave que requieren atención en futuras investigaciones

5.3 Recomendaciones para futuras investigaciones

Para futuras investigaciones, se recomienda explorar varias direcciones para mejorar el rendimiento y la aplicabilidad de los modelos AFD en la identificación de mutaciones genéticas.

En primer lugar, es esencial ampliar la diversidad de las muestras de entrenamiento para incluir una mayor variedad de mutaciones y contextos genéticos. Esto podría mejorar significativamente la capacidad de generalización del modelo. También sería beneficioso entrenar modelos más grandes utilizando conjuntos de datos más extensos, lo que permitiría capturar una mayor complejidad y variabilidad genética.

Además, sería interesante desarrollar enfoques híbridos que combinen modelos AFD con otras técnicas de machine learning, como redes neuronales profundas, para mejorar la precisión y reducir la tasa de falsos positivos y negativos.

Finalmente, se recomienda realizar estudios longitudinales que evalúen la efectividad de los modelos en el diagnóstico y seguimiento de pacientes con enfermedades genéticas a lo largo del tiempo. Estos estudios podrían proporcionar información valiosa sobre cómo los modelos pueden ser mejorados y adaptados para uso clínico, asegurando que las herramientas desarrolladas sean prácticas y beneficiosas para los pacientes y profesionales médicos.

APÉNDICE A
Anexo 1

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	**Alto**	**Medio**	**Bajo**	**No Procede**
ODS 1. *Fin de la pobreza.*				X
ODS 2. *Hambre cero.*				X
ODS 3. *Salud y bienestar.*	X			
ODS 4. *Educación de calidad.*		X		
ODS 5. *Igualdad de género.*				X
ODS 6. *Agua limpia y saneamiento.*				X
ODS 7. *Energía asequible y no contaminante.*				X
ODS 8. *Trabajo decente y crecimiento económico.*				X
ODS 9. *Industria, innovación e infraestructuras.*	X			
ODS 10. *Reducción de las desigualdades.*				X
ODS 11. *Ciudades y comunidades sostenibles.*				X
ODS 12. *Producción y consumo responsables.*				X
ODS 13. *Acción por el clima.*				X
ODS 14. *Vida submarina.*				X
ODS 15. *Vida de ecosistemas terrestres.*				X
ODS 16. *Paz, justicia e instituciones sólidas.*				X
ODS 17. *Alianzas para lograr objetivos.*	X			

APÉNDICE B
Anexo 2

B.1 Objetivos de Desarrollo Sostenible

El 25 de septiembre de 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.

1. Fin de la pobreza
2. Hambre Cero
3. Salud y Bienestar
4. Educación de Calidad
5. Igualdad de género
6. Agua limpia y saneamiento
7. Energía asequible y no contaminante
8. Trabajo decente y crecimiento económico
9. Industria, innovación e infraestructura
10. Reducción de las desigualdades
11. Ciudades y comunidades sostenibles
12. Producción y consumos responsables
13. Acción por el clima
14. Vida submarina
15. Vida de ecosistemas terrestres
16. Paz, justicia e instituciones sólidas
17. Alianzas para lograr los objetivos

B.2 Relación del Trabajo de Fin de Grado con los Objetivos de Desarrollo Sostenible

De los objetivos de desarrollo sostenible mencionados, el presente Trabajo de Fin de Grado está relacionado principalmente con:

B.2.1. Salud y Bienestar

El proyecto de diseño e implementación de modelos de lenguaje para información genómica asociada a enfermedades raras, específicamente la retinosis pigmentaria, contribuye significativamente a la salud y bienestar. Este trabajo busca mejorar la capacidad de diagnóstico y el entendimiento de estas enfermedades, lo que a largo plazo podría resultar en tratamientos más efectivos y personalizados, mejorando así la calidad de vida de los pacientes.

B.2.2. Educación de Calidad

El desarrollo de modelos de inferencia gramatical k-testables para analizar secuencias genéticas también tiene un impacto positivo en la educación de calidad. Los resultados y métodos de este TFG pueden ser utilizados como material didáctico y de investigación en cursos de bioinformática, genética y ciencias de la computación, promoviendo una educación basada en proyectos y en la aplicación práctica de teorías avanzadas.

B.2.3. Industria, Innovación e Infraestructura

La implementación de tecnologías avanzadas como la inteligencia artificial y el aprendizaje automático en el campo de la bioinformática fomenta la innovación y la creación de infraestructuras tecnológicas robustas. Este proyecto no solo avanza en el conocimiento científico, sino que también establece una base para futuras investigaciones y desarrollos en el análisis genómico, fortaleciendo así la industria tecnológica y científica.

B.2.4. Alianzas para lograr los objetivos

La colaboración con el Instituto de La Fe de Valencia para la obtención de datos genómicos es un claro ejemplo de cómo las alianzas pueden impulsar la investigación y el desarrollo. Este TFG destaca la importancia de trabajar conjuntamente entre instituciones académicas y centros de investigación para lograr avances significativos en el conocimiento y tratamiento de enfermedades raras.

B.3 Conclusión

El presente trabajo no solo avanza en el conocimiento técnico y científico en el área de la bioinformática y la genética, sino que también contribuye de manera significativa a varios Objetivos de Desarrollo Sostenible, demostrando que la investigación y la innovación son esenciales para alcanzar un desarrollo sostenible y mejorar la calidad de vida a nivel global.

Bibliografía

[1] César Paz et al. «La inteligencia artificial en medicina general y en genómica». En: *Metro Ciencia* 31.2 (2023), págs. 81-86.

[2] Irene Perea-Romero et al. «Distrofias Hereditarias de Retina en España: tres décadas de estudio epidemiológico, clínico y genético». En: *Anales de la Real Academia Nacional de Medicina*. Vol. 139. 03. Real Academia Nacional de Medicina. 2022, pág. 274.

[3] *Las partes del ojo humano y sus funciones*. 2020. URL: https://www.masvision.es/blog/las-partes-del-ojo-humano-y-sus-funciones.

[4] *El ojo*. BCM Families Foundation. URL: https://www.blueconemonochromacy.org/es/how-the-eye-functions/.

[5] Delgado-Pelayo Sarai. «Retinosis Pigmentaria». En: *Revista Médica MD* 3.3 (2012), págs. 163-166.

[6] Lasse Rouhiainen. «Inteligencia artificial». En: *Madrid: Alienta Editorial* (2018), págs. 20-21.

[7] Esperanza Manrique Rojas. «Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo». En: *Revista Ibérica de Sistemas e Tecnologias de Informação* E28 (2020), págs. 586-599.

[8] Junaid Bajwa et al. «Artificial intelligence in healthcare: transforming the practice of medicine». En: *Future Healthcare Journal* 8.2 (2021), e188-e194. ISSN: 2514-6645. DOI: 10.7861/fhj.2021-0095. eprint: https://www.rcpjournals.org/content/8/2/e188.full.pdf. URL: https://www.rcpjournals.org/content/8/2/e188.

[9] Tianwei Yue et al. «Deep learning for genomics: A concise overview». En: *arXiv preprint arXiv:1802.00810* (2018).

[10] Alexander Gelbukh. «Procesamiento de lenguaje natural y sus aplicaciones». En: *Komputer Sapiens* 1 (2010), págs. 6-11.

[11] Mónica María Echeverri Torres y Roberto Manjarrés-Betancur. «Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural». En: *Revista Politécnica* 16.31 (2020), págs. 85-96.

[12] Franklin Estuardo Velásquez Fuentes. «Aplicación de visión por computadora para detección de enfermedades en radiografías de tórax». Tesis doct. Universidad de San Carlos de Guatemala, 2021.

[13] S Kevin Zhou, Hayit Greenspan y Dinggang Shen. *Deep learning for medical image analysis*. Academic Press, 2023.

[14] Yee Wen Choon et al. «Artificial intelligence and database for NGS-based diagnosis in rare disease». En: *Frontiers in Genetics* 14 (2024). ISSN: 1664-8021. DOI: 10.3389/fgene.2023.1258083. URL: https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2023.1258083.

[15] Marianela Mejías, Yeisy Cristina Guarate Coronado y Ana Lucía Jiménez Peralta. «Inteligencia artificial en el campo de la enfermería. Implicaciones en la asistencia, administración y educación». En: *Salud, Ciencia Y Tecnología* 2 (2022), págs. 88-88.

[16] John J Sprockel et al. «Ensamble de redes neuronales artificiales ponderado mediante características operativas para el pronóstico de la insuficiencia cardiaca aguda». En: *Revista Colombiana de Cardiología* 30.5 (2023), págs. 235-242.

[17] Alejandro Granados Bañuls. «Diseño e implementación de sistemas de anotación genómica basados en computación biomolecular y biocelular y técnicas de machine learning». Trabajo de Fin de Grado. Universitat Politècnica de València, 2021.

[18] Colin De la Higuera. *Grammatical inference: learning automata and grammars*. Cambridge University Press, 2010.

[19] *Threads y Procesos*. URL: https://codigofacilito.com/articulos/threads-procesos.

[20] J Aguilar, E Leiss et al. «Introducción a la computación paralela». En: *Editorial Venezolana, Universidad de Los Andes, Mérida* (2004).

[21] j2logo. *Set python - conjuntos python. El Tipo de Dato set y operaciones básicas*. Ene. de 2022. URL: https://j2logo.com/python/tutorial/tipo-set-python/.

[22] j2logo. *List Python - Listas Python. El Tipo de Dato List. Operaciones Sobre Listas*. Ene. de 2022. URL: https://j2logo.com/python/tutorial/tipo-list-python/.

[23] Davide Chicco y Giuseppe Jurman. «The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation». En: *BMC genomics* 21 (2020), págs. 1-13.