Pontificia Universidad Católica del Ecuador

LIMPIEZA DE LECTURAS DE SECUENCIACIÓN DE ADN DE *THEOBROMA CACAO* UTILIZANDO REDES NEURONALES

Trabajo de titulación previo a obtener el título de MSc. en Biología Computacional

Ana María Acosta Irigoyen

Director: Laura Gonzales

Quito - Ecuador

2025

Dedicatoria

Para Dios y toda mi familia.

Agradecimientos

A Dios, por todos los dones recibidos, que me han permitido llegar a esta etapa de mi vida junto a mis seres más amados.

A mis padres, Rocío y Patricio, por todo el apoyo inquebrantable y amor infinito que me han proporcionado en cada etapa de mi vida, quienes me han enseñado a luchar por lo que me proponga con voluntad, esfuerzo y lo más importante, por servicio a Dios.

A mi hermano, Sebastián, por ser mi roca y mi modelo a seguir personal y profesionalmente, quien no solo me ha brindado su apoyo y conocimientos, sino que me ha alentado en los momentos más complicados de esta etapa.

A mi enamorado, Victor, por todo el amor, cariño y apoyo constantes que han sido fundamentales para concluir este camino de aprendizaje.

A Laura, mi tutora, por su acompañamiento y conocimientos brindados a lo largo del presente trabajo.

Índice general

Índice de tablas

Índice de gráficos

Índice de anexos

Índice de apéndices

Resumen

Theobroma cacao, comúnmente conocido como Cacao, es un fruto indispensable para la producción del chocolate. Su cultivo enfrenta retos significativos debido a una alta susceptibilidad a enfermedades, lo cual puede causar pérdidas económicas. Contar con información genómica precisa es crucial para entender las posibles enfermedades y, por tanto, colaborar eventualmente con el desarrollo de variedades de cacao resistentes a enfermedades. Actualmente, las secuencias de lectura crudas de *Theobroma cacao* contienen lecturas contaminadas y ruido, lo cual complica el subsecuente análisis, afectando la confiabilidad de los estudios genómicos.

El propósito de este tema de titulación se centró en limpiar las secuencias de lectura crudas de *Theobroma cacao* que están contaminadas y/o contienen ruido, a través de la utilización de redes neuronales profundas.

Esta investigación tomó ventaja de las técnicas avanzadas de aprendizaje de máquina, mediante el entrenamiento de una red neuronal, alimentándola con grandes conjuntos de datos de lecturas crudas de secuenciación de Illumina de *Theobroma cacao*.

El entrenamiento de los datos se enfocó en distinguir entre secuencias correctas de *Theobroma cacao* y secuencias que no lo son. La eficiencia de esta red neuronal fue contrastada con otros métodos de limpieza de lecturas de secuenciación para determinar la eficiencia de este modelo de red neuronal, constituyéndose como la hipótesis motora de la presente investigación.

La relevancia que propone este tema de investigación es de gran importancia para la industria del cacao y, a su vez, para el estudio genómico de plantas en general.

Abstract

Theobroma cacao, commonly known as cocoa, is an indispensable fruit for chocolate production. Its cultivation faces significant challenges due to its high susceptibility to diseases, which can lead to economic losses. It is crucial to account for precise genomic information to understand and study cocoa diseases and, therefore, contribute to the development of disease-resistant cocoa variants. Nowadays, *Theobroma cacao* raw read sequences contain contaminated reads, which could lead to inaccurate analyses, affecting the reliability of genomic studies.

The purpose of the current thesis is to clean *Theobroma cacao* raw read sequences that contain contaminants through the development of a deep neural network.

This investigation took advantage of advanced machine learning techniques by training a neural network that was fed with large sets of *Illumina* raw read sequences of *Theobroma cacao*.

The data training was mainly focused on classifying real cocoa sequences and those which are not. The efficiency of this thesis was contrasted with other clean-up methods to determine the performance of the neural network, which constitutes the central hypothesis of this research.

The importance of this thesis is crucial for the cocoa industry and for plant genomic research.

1. Introducción

La presente investigación surge desde las limitaciones de las actuales lecturas de secuenciación, intentando plantear una mejora en su depuramiento y, así, aportar en el análisis genómico de la especie *Theobroma cacao*. Para el efecto, este trabajo incorporó el uso específico del aprendizaje de máquina, especialmente a través de redes neuronales, constituyéndose como una de las principales innovaciones metodológicas y conceptuales para esta labor. Con dichos antecedentes, es importante contextualizar que, hasta el presente, este proceso se había venido suscitando únicamente a través de métodos tradicionales de depuración genómica con algoritmos complejos, como las técnicas de alineación y filtrado por calidad. Por lo tanto, esta investigación aprovechó esta oportunidad de mejora y planteó su hipótesis alrededor de estas posibilidades.

Adicionalmente, y a nivel metodológico, se ha tomado en consideración los aportes de Mock, Kriese y Marz (2022), cuyo proceso, aunque con alcances diferentes, podría suponer una guía para resolver los problemas biológicos que podrían haberse suscitado. En cuanto al nivel teórico, se contemplaron las posibilidades infinitas de la aplicación del aprendizaje de máquina como apoyo en las áreas de bioinformática en general.

Para entender la problemática central se resalta la importancia, tanto a nivel económico como productivo, del *Theobroma cacao* para el Ecuador, ya que constituye la base para la elaboración de chocolate. Sin embargo, entre estas oportunidades productivas se barajan nuevas investigaciones y tecnologías que podrían suponer importantes avances en otras áreas como la medicina. Un ejemplo de este punto es la investigación realizada por científicos de Malasia, India y Bangladesh (Huq et. al. 2024), quienes han relacionado ciertos componentes propios de *Theobroma cacao*, como la catequina, y su posible impacto como un candidato terapéutico para mitigar enfermedades como el dengue que, actualmente, no posee una medicación retroviral específica y que, en muchos casos, resulta fatal. Así, las

oportunidades económicas y de avance médico-científico relacionadas a este producto poseen inconmensurables oportunidades.

Adicionalmente, el cacao se ha convertido últimamente en un motivo de orgullo e identidad nacional, ya que ciertas investigaciones han apuntado a que el origen de su domesticación sería en nuestro territorio, contraponiéndose a las visiones clásicas de su origen centroamericano. Curiosamente, dicha investigación también se basó en indicadores genéticos que rastrearon un ancestro común del cacao actual a través de sus componentes como la teobromina, la cafeína y la teofilina. Dicha investigación tiene su base en 326 muestras arqueológicas ubicando un origen de hace por lo menos 5300 años en la Amazonía ecuatoriana (Lanaud et al. 2024).

Estos descubrimientos tuvieron repercusiones a nivel gubernamental ya que promueven sentimientos identitarios y abren la puerta al desarrollo de planes y oportunidades en áreas relativas a ciencias sociales, diseño, merchandising y sobre todo, el turismo. Sobre todo en lo referente a este último, los esfuerzos y planificaciones que giran en torno a esta especie han sido claves para establecer las 16 rutas turísticas que destacan por sus actividades recreativas y demostrativas sobre esta planta y la cultura que genera. Cabe destacar también que muchas de estas rutas tienen como enfoque principal el turismo comunitario, afirmando el punto anterior sobre el apoyo al crecimiento económico del país. (Salazar y Espinoza, 2022)

Por otra parte, es importante entender que esta especie es susceptible a contraer enfermedades, de modo que, su análisis genómico es importante para poder plantear vías para contrarrestarlas al crear nuevas variedades más resistentes. Entre dichas enfermedades, se destacan la enfermedad de la vaina o mazorca negra, la escoba de bruja y la moniliasis (Adeniyi, 2019). Ya que dichas enfermedades constituyen una problemática muy frecuente, para poder mitigarlas, el uso de pesticidas químicos peligrosos ha sido muy

difundido y extendido, como afirma la investigación de Boakye, Stanley y White (2023). En muchos casos, dichos agentes químicos externos pueden incurrir en otras problemáticas adicionales, por lo que, análisis como el presente y sus posibilidades referentes a la mejora genética de variantes, podrían suponer una reducción de dichas prácticas y por tanto mejorar las condiciones del producto y medioambientales. Cabe destacar, como punto adicional, que este estudio sobre pesticidas también resalta a esta especie como el sustento para más de 50 millones de personas a nivel mundial, ratificando su importancia.

Además de las enfermedades, pueden existir otras afectaciones como las malezas, entre la que destaca la lengua de vaca, (*Dieftenbachia sanguine*) o las plagas como el gusano falso medidor (*Stenoma* sp), los chinches, los piojillos o incluso mamíferos como las ardillas o ciertas especies de pájaros. Si bien es cierto que son organismos externos, las soluciones más comunes involucran también el uso de elementos químicos como los que se revisaron previamente y que podrían afectar directamente a la pureza genómica de esta especie (Córdova 2009).

Adicionalmente, y en el contexto ecuatoriano, en donde la sequía ha venido permeando desde hace un tiempo, las amenazas no solo incluyen a las enfermedades, sino también a la falta de agua. Con dichas consideraciones, un grupo de investigadores compatriotas ha planteado la importancia del estudio genómico para hallar variaciones adaptables al estrés ocasionado por la falta de agua, planteando soluciones tangibles a la productividad de este cultivo (Montenegro et al. 2023). Todos estos estudios demuestran cómo la información genómica puede ayudar a la mejora de cultivos y a una calidad mayor del producto final (Argout et al., 2011).

Con todos estos antecedentes, la presente investigación pretende mejorar los procesos de análisis genómico al otorgar alternativas innovadoras que podrían obtener resultados más limpios de dichas secuencias que, por lo general, poseen información

genómica de otras especies u otros errores. Estos resultados podrían contrarrestar enfermedades, plagas, sequías o el uso extensivo de agentes químicos externos. En ese sentido, esta propuesta pretende aprovechar las cualidades positivas y destacables que posee la tecnología de aprendizaje de máquina, entrenando su red neuronal con grandes sets de datos con etiquetas que permitan reconocer entre contaminados y lecturas de *Theobroma cacao*.

El valor de estas lecturas más puras es clave, ya que, investigaciones como la realizada por Winters NP, Wafula EK, Knollenberg BJ, et al. (2024), hablan sobre la importancia de la cría selectiva en el mundo botánico, ejemplificando con el caso del patógeno *Phytophthora palmivora* hacia *Theobroma Cacao*. El estudio destaca que las plantas poseen fuertes adaptaciones evolutivas para contrarrestar amenazas y enfermedades, pero al perder la diversidad genética a través de procesos controlados como la agricultura, el humano es quien debería tomar el rol de potenciar esas defensas falentes a través de la selección genética.

Así, se planteó como objetivo principal de esta investigación: Desarrollar una red neuronal para limpiar y filtrar lecturas de secuenciación de *Theobroma cacao*, con el fin de intentar mejorar la eficiencia de resultados contra técnicas tradicionales. En cuanto a los objetivos específicos, se planteó lo siguiente:

- Implementar una red neuronal que ayude a la clasificación taxonómica de secuencias de ADN en una lectura cruda de secuenciación de Illumina de la especie *Theobroma cacao*.

- Generar un conjunto de datos limpio y de alta calidad de las secuencias de *Theobroma cacao* para su posterior ensamblaje y uso en estudios genómicos.

- Comparar el rendimiento de la red neuronal versus técnicas de limpieza tradicionales en términos de eficiencia, en la medida de lo posible.

El alcance de la presente investigación estará delimitado por las fuentes primarias y secundarias a las que se ha logrado acceso. Entre las primeras, se resalta la red neuronal basada en BERTax que clasifica las lecturas crudas de secuenciación, que estará disponible en GitHub y los resultados propios de este trabajo. Entre las segundas, se listan los datos crudos de secuenciación de *T. cacao*, datos de genomas de referencia y bases de datos taxonómicas.

2. Revisión de la literatura

La evolución que ha tenido el proceso de secuenciación de ADN se remonta desde la secuenciación de Sanger en 1977 hasta el presente, con tecnologías de secuenciación de próxima generación, también conocidas como NGS, por sus siglas en inglés (Next-Generation Sequencing). Dichas tecnologías han contribuido a mejorar la precisión de los resultados a un menor costo (Mardis, 2017).

Sin embargo, algunas de las tecnologías que se usan hoy día, como Illumina, proporcionan lecturas crudas contaminadas, debido a diversas fuentes como sesgos de amplificación de los fragmentos de ADN por PCR, errores de las máquinas de secuenciación y/o contaminantes ambientales (Tremblay et al., 2015). Estos errores pueden afectar gravemente la calidad de los ensamblajes genómicos, lo que, a su vez, puede ocasionar interpretaciones biológicas erróneas para otras investigaciones.

Existen métodos tradicionales que tienen el objetivo de limpiar dichas lecturas crudas de secuenciación como el recorte de calidad, el filtrado basado en k-mers y la eliminación de adaptadores (Bolger, Lohse, & Usadel, 2014). Estos métodos han resultado bastante efectivos, aunque dependen de umbrales predefinidos y reglas heurísticas que pueden no capturar la complejidad de los datos de secuenciación, especialmente cuando se trata de regiones del genoma altamente repetitivas o de baja complejidad (Del Fabbro et al., 2013).

En los últimos años, el aprendizaje automático, y específicamente, el aprendizaje profundo, han surgido como herramientas poderosas para abordar problemas complejos en la genómica. Las redes neuronales se han aplicado con éxito a una variedad de tareas, incluyendo la clasificación de secuencias, el descubrimiento de motivos y la anotación del genoma (Zou, Huss, & Abidin, 2019).

En este punto, es importante también entender la importancia de la taxonomía como una poderosa herramienta de clasificación de los organismos y de las muestras de

secuencias genómicas a tratarse. Investigadores, encabezados por Florian Mock (2022) han realizado una investigación en donde utilizan el *deep learning* para decodificar el lenguaje del ADN y, posteriormente, poder descubrir los lenguajes de cada grupo taxonómico y así identificarlos. Para el efecto, utilizaron la misma red planteada para este proyecto, es decir BERTax. Entre las novedades, se destaca que esta red trabaja en base al lenguaje natural y pudo traspasar las clásicas limitaciones de redes similares, especialmente las relacionadas a codificaciones por grupos. Adicionalmente, al utilizar el mecanismo de *self-attention*, no procesa los datos en un orden preestablecido, logrando reconocer relaciones complejas no aparentes en un menor periodo de tiempo. Así, aunque el enfoque es algo distinto, se pudo corroborar que esta red neuronal puede otorgar resultados muy positivos y también se puede tomar en consideración algunas de las configuraciones de entrenamiento propuestas por este grupo, como se puede revisar en la sección de metodología.

2.1. ***Theobroma cacao***

Theobroma cacao, comúnmente conocido como cacao, es un fruto indispensable para la producción del chocolate. Es altamente cultivado en regiones tropicales como en las de Ecuador, Brasil, Costa de Marfil, Ghana, entre otros (Cilas & Bastide, 2020). La producción de esta planta se ve favorecida gracias a condiciones climáticas cálidas y también por prácticas de cultivación que buscan gestionar de mejor forma las enfermedades agrícolas que las acechan, como lo mencionan Cilas & Bastide en su investigación de "*Challenges to Cocoa Production in the Face of Climate Change and the Spread of Pests and Diseases"* y también Adeniyi en su investigación "*Diversity of Cacao Pathogens and Impact on Yield and Global Production"*. Tristemente, el cacao enfrenta varios riesgos que afectan su producción, debido a factores como cambios climáticos que suelen alterar los patrones de lluvia y que, a su vez, las vuelven más vulnerables a aumentar plagas y enfermedades (Cilas & Bastide, 2020). Esto lo corrobora Adeniyi (2019), indicando que el cacao es susceptible a varias enfermedades que pueden afectar gravemente su producción. Entre las más comunes, están la enfermedad

de la vaina o mazorca negra, causada por *Phytophthora* spp, la escoba de bruja y la moniliasis.

El cultivo de cacao resulta ser de gran importancia económica para las regiones antes mencionadas, pero su producción puede enfrentar retos significativos debido a su potencial riesgo a enfermedades (Aneja et al., 2015). Por todo lo mencionado anteriormente, resulta crucial que la información genómica de estas plantas sea precisa para entender dichas enfermedades y así, facilitar vías de investigación para el desarrollo eventual de variedades de cacao resistentes a ellas, como lo han hecho Gutiérrez, Campbell, y Phillips-Mora, en 2016.

Con respecto al genoma de *Theobroma cacao*, dependiendo del genotipo, su tamaño varía entre 324 y 445 megabases (Mb). La variedad de cacao criollo, tiene un tamaño de 324.7 Mb, mientras que la variedad Matina 1-6 posee un tamaño aproximado de 445 Mb, siendo este el más cultivado en el mundo (Ricaño, Ramos, Cocoletzi & Hipólito, 2018). El *Theobroma cacao* es una especie diploide con un número de $2n = 2x = 20$, es decir, posee 10 pares de cromosomas, a excepción de una anomalía en una planta de cacao que tiene 19 cromosomas y que tiene hojas arrugadas y muy pequeñas (Muñoz O, 1948). En cuanto a la variedad de cacao criollo, este dispone de 10 cromosomas más el material genético del cloroplasto. Dichos orgánulos tienen su propio ADN circular, separado del genoma nuclear. Este ADN plastídico codifica genes esenciales para funciones como la fotosíntesis y otras actividades metabólicas específicas de las plantas (Wicke et al., 2011).

2.2. ***Lecturas crudas de secuenciación***

El análisis y estudios de la información genómica pueden verse altamente comprometidos debido a que las lecturas crudas de secuenciación están contaminadas por diversos motivos, conteniendo lecturas contaminadas y con ruido (Sangiovanni et al., 2019). Las lecturas crudas de secuenciación se refieren a los datos no procesados generados por tecnologías de secuenciación de alto rendimiento, como la secuenciación Illumina (Dunning,

2018). Tal como lo describe Dunning, estas lecturas contienen secuencias cortas de los nucleótidos Adenina (A), Timina (T), Citosina (C) y Guanina (G) que representan fragmentos de la molécula de ADN original. Si bien las tecnologías de secuenciación han revolucionado la genómica al permitir la secuenciación rápida y rentable de genomas completos, las lecturas crudas que se producen, a menudo están contaminadas, incluyendo errores de diversos tipos como llamadas de bases incorrectas, regiones de baja calidad, contaminación por adaptadores y contaminantes ambientales (Goodwin, McPherson, & McCombie, 2016). Es importante resaltar que, según Stoler y Nekrutenko (2021), la secuenciación Illumina suele presentar una tasa típica de error del 0.1% al 1%. Así mismo, para entender la problemática de la contaminación presente en secuenciaciones con Illumina, es importante saber las tasas típicas de contaminación para esta tecnología en plantas. Aunque no se encontró esta información concreta, a través del análisis taxonómico en SRA browser de NCBI de algunas muestras de *Theobroma cacao,* como las presentadas en la Tabla 1, se encontró una tasa de contaminación desde 9.8% al 17.63% para lecturas con Illumina NovaSeq 6000. Dicha contaminación está compuesta por bacterias, arqueas, virus y otras familias de eucariotas no relacionadas a *Theobroma cacao*.

La falta de una limpieza y procesamiento adecuados de las lecturas crudas puede llevar a análisis genómicos inexactos, lo que podría comprometer los resultados de las investigaciones y programas de mejoramiento (Bolger, Lohse, & Usadel, 2014). Debido a esto, existen algunas herramientas bioinformáticas que colaboran con la detección y eliminación de la contaminación en las lecturas. Una de las más conocidas es Kraken, el cual utiliza algoritmos de búsqueda con índices basados en k-mers; lamentablemente, esta herramienta consume mucha capacidad de memoria de acuerdo a Wood, Lu & Langmead (2019), alrededor de 72 GB, resultando en limitaciones para algunas aplicaciones. Debido a eso, se desarrolló Kraken2 con un algoritmo similar pero mejorado (Wood, Lu & Langmead, 2019). Esta búsqueda por mejorar las herramientas en términos de precisión y eficiencia en las lecturas, es continua. Otros ejemplos que demuestran que este avance no se ha detenido

son DecontaMiner, desarollado por Sangiovanni et al. (2019), que buscaba mejorar la precisión y DeconSeq, desarrollado por Schmieder (2011), cuya meta era lograr una mayor eficiencia en secuencias mayores a 150 bp.

2.3. ***Aprendizaje automático***

Los recientes avances en el aprendizaje automático, particularmente en el aprendizaje profundo, han demostrado tener un gran potencial en diversas áreas de la bioinformática (Li et al., 2020). Las redes neuronales profundas, conocidas como Deep Neural Network (DNN), son una clase de algoritmos de aprendizaje automático profundo particularmente adecuadas para manejar grandes conjuntos de datos complejos, como los datos generados en genómica. Estas redes están compuestas por múltiples capas de neuronas interconectadas entre capa y capa, las cuales pueden aprender representaciones jerárquicas de los datos. Las DNN y, en particular las redes neuronales convolucionales, conocidas como CNN, se han aplicado ampliamente en diversas tareas de bioinformática, incluyendo la clasificación de secuencias, el descubrimiento de motivos y la corrección de errores en datos de secuenciación (LeCun, Bengio, & Hinton, 2015). La capacidad de las DNN para aprender y reconocer automáticamente patrones en los datos sin la necesidad de ingeniería manual las convierte en una herramienta potencialmente poderosa para mejorar la precisión del preprocesamiento de datos de secuenciación (Gulati, 2024). Estudios recientes han demostrado que las DNN pueden superar a las herramientas tradicionales de bioinformática en tareas como la detección de variantes y el ensamblaje de genomas, ofreciendo un nuevo paradigma para el análisis de datos de secuenciación de alto rendimiento (Poplin et al., 2018).

Otro gran subcampo del aprendizaje de máquina es el procesamiento de lenguaje natural (Natural Language Processing NLP) que se enfoca en que la máquina sea capaz de procesar, interpretar y generar lenguaje humano (Caelen & Blete, 2024). Según Caelen y Blete, los objetivos de los NLPs son: la clasificación de texto, la traducción automática, la respuesta a preguntas y la generación de texto. Todas ellas son características que resultan

de gran ayuda para la clasificación de secuencias genómicas, como se comprobó en el 2022 en la investigación de Mock, Kriese, y Marz, quienes desarrollaron una red neuronal profunda basada en NLP para clasificar taxonómicamente secuencias de ADN.

Con todo esto expuesto, se intentó desarrollar una red neuronal que mejore la limpieza de las secuencias crudas, utilizando NLP u otras arquitecturas modernas de DNN que faciliten el análisis de secuencias de ADN.

2.4. ***Marco Conceptual***

Theobroma cacao: Es un árbol tropical perenne de la familia Malvaceae, originario de las selvas bajas de las cuencas del Amazonas y Orinoco. Sus semillas, conocidas como granos de cacao, son procesadas para producir cacao en polvo, manteca de cacao y chocolate. Su nombre científico significa "alimento de los dioses" en griego (Britannica, 2024).

Illumina Sequencing: La secuenciación Illumina es una tecnología de secuenciación de alto rendimiento (NGS) que utiliza métodos de secuenciación por síntesis para obtener grandes volúmenes de datos genéticos con alta precisión. Es ampliamente utilizada en genómica y transcriptómica (Illumina, 2023).

Fastq: Es un formato de archivo de texto que almacena secuencias de nucleótidos (A, T, C, G) junto con sus calidades de lectura. Cada secuencia está representada en cuatro líneas: identificación de la secuencia, lectura de la secuencia, símbolo separador (+) y puntaje de calidad. (Dunning, 2018)

WGS (Whole Genome Sequencing): Es la secuenciación de genoma completo. Esta es una técnica que permite la obtención de la secuencia completa de ADN de un organismo, proporcionando una visión integral de su genoma (Mardis, 2017).

Aprendizaje automático: El aprendizaje automático o también conocido como Machine Learning, es un campo de la inteligencia artificial que utiliza algoritmos y modelos estadísticos para que los sistemas informáticos lleven a cabo tareas específicas sin ser programados explícitamente para ello. (Mitchell, 1997).

Aprendizaje profundo: También conocido por su nombre en inglés, Deep Learning, es un subcampo del aprendizaje automático que se basa en redes neuronales artificiales de múltiples capas para modelar patrones complejos en grandes conjuntos de datos (LeCun, Bengio, & Hinton, 2015).

Red neuronal profunda: Es un tipo de red neuronal artificial, compuesta de distintas capas que, a su vez, conforman una serie de neuronas artificiales. Las capas pueden ser de entrada, de salida u ocultas. Se utiliza para modelar relaciones complejas no lineales (Schmidhuber, 2015).

NLP (Natural Language Processing): El procesamiento de lenguaje natural es un campo de la inteligencia artificial que se centra en el procesamiento, interpretación y generación de lenguaje humano entre las computadoras y un usuario final. (Caelen & Blete, 2024)

Tasa de alineamiento: Es el porcentaje de lecturas de secuenciación que logran alinearse contra un genoma de referencia. (Langmead & Salzberg, 2012)

3. Metodología

Se trabajó la metodología desde un paradigma cuantitativo compuesto de dos métodos, uno de observación y otro experimental. El primero contempla una guía de observación y medición. El segundo contempla netamente la experimentación que permita encontrar el mejor modelo de aprendizaje profundo, basándose en la arquitectura BERTax propuesta por Mock, Kriese y Marz en 2022. Así, se detallan los hitos propios de este proceso:

3.1. **Preparación y Preprocesamiento del Dataset**

3.1.1. Preparación de los datos

Como primer paso, se descargaron satisfactoriamente las lecturas crudas de secuenciación de ADN detalladas en la Tabla 1.

Tabla 1

Datos de *WGS* de *Illumina NovaSeq* 6000 del tejido de hoja de *Theobroma cacao*

# de Bases	**Tamaño (Gb)**	**Accession**	**Origen**
21.9 G	6.6 Gb	SRR21562212	NCBI*/ENA**
64 G	19.1 Gb	SRR14022547	NCBI*/ENA**
4.6 G	2.5 Gb	SRR377719	NCBI*/ENA**

* *National Center for Biotechnology Information*
** European Nucleotide Archive

También, se descargaron satisfactoriamente los genomas de referencia citados a continuación:

Tabla 2

Genomas de referencia utilizados y disponibles en NCBI

ID de Genoma de Referencia	Nombre de la muestra	Acceso GCA
1	Genome assembly Criollo_cocoa_genome_V2	GCA_000208745.2
2	Genome assembly *Theobroma_cacao*_20110822 Matina 1-6	GCA_000403535.1
3	ASM3589663v1	GCA_035896635.1

3.1.2. **Validación de los datos**

En una segunda instancia, se corrió *Fastqc* con el fin de validar la calidad de las lecturas crudas de secuenciación de ADN descargadas (SRR21562212, SRR14022547, SRR377719). Este punto fue clave para determinar la validez de estas lecturas o, incluso, la necesidad de tomar otra muestra.

Los resultados obtenidos fueron satisfactorios para todas las muestras ya que la valoración de calidad *Phred* fue de 30 o más, como se puede ver en la Tabla 3.

Tabla 3

Resultados de de *Fastqc* en varias lecturas crudas de secuenciación de ADN de *Theobroma cacao*

	Lecturas totales	Bases totales	Lecturas marcadas como poca calidad	%GC
SRR377719_1*	28.794.335	2.3 Gbp	0	33
SRR377719_2*	28.794.335	2.3 Gbp	0	33
SRR21562212_1**	72.717.892	10.9 Gbp	0	36
SRR21562212_2**	72.717.892	10.9 Gbp	0	36
SRR14022547_1***	211.825.883	31.9 Gbp	0	38
SRR14022547_2***	211.825.883	31.9 Gbp	0	36

* Anexo 1, 2, 3 y 4
** Anexo 5, 6, 7 y 8
*** Anexo 9, 10, 11 y 12

3.2. ***Clasificación de los datos***

En la siguiente etapa, el proceso clasificatorio se realizó a través de *Kraken2* de manera local, con el fin de permitir la utilización de la base de datos taxonómica de *Kraken2*. Sin embargo, este proceso presentó una serie de inconvenientes y errores que tomó alrededor de 10 intentos para solventarse, y solo de manera parcial. Este proceso, evidenció la necesidad de un nuevo algoritmo para clasificar de manera eficiente las lecturas. Los primeros errores tuvieron que ver con la falta de espacio en disco duro; aunque las indicaciones originales de la documentación de *Kraken2* hablan de un espacio requerido de 100 *Gb*, realmente se necesitaron 280 *Gb* en total. Para descubrir y lograr este resultado, estos primeros 4 intentos fueron configurados a través de una máquina virtual de *Ubuntu*, en la que se asignaron 100, 200, 250 y 300 *Gb* correspondientemente.

Como segunda problemática, las fallas adujeron que la capacidad de *RAM* no era suficiente. En todas las configuraciones de máquinas virtuales, la asignación de 14 *Gbs* de *RAM* no cumplió los requerimientos. Así, se decidió cambiar de enfoque y realizar este proceso con *WSL* en *Windows*, instalando directamente *Ubuntu* dentro del mismo. Gracias a esto, se pudo utilizar completamente las capacidades totales de la máquina principal de *Windows*, es decir 800 *Gb* en *SSD* y 32 *Gb* de *RAM* disponibles. Con esta configuración se realizaron 5 intentos nuevos, en donde la capacidad de la *RAM* siguió presentando problemas al no permitir la asignación necesaria de memoria al construir la base de datos con el comando *build.* Para sortear este problema, cada uno de los intentos se configuró con un *K-mer size* diferente, arrancando desde el *default* (35 a 45) hasta llegar a uno de 55.

Este último intento con *K-mer* size de 55 finalmente funcionó, pero únicamente, de forma parcial. El paso final de la construcción de la base de datos de 250 *Gb* fue el que falló, ya que solicitó aún más capacidades en *RAM* (Apéndice 1). Este error parece es conocido

dentro de la comunidad de *Kraken2* y muchos blogs hablan sobre una necesidad real de al menos 80 gbs en *RAM* (Wood, 2018) (Anand, 2024)

Con todos estos antecedentes y con el fin de conseguir lecturas «clasificadas» y «no clasificadas» para el input de la red neuronal, se decidió utilizar *Kraken2* en *Galaxy*. Esta plataforma web tiene algunos servidores distribuidos en zonas como Europa, Estados Unidos y Australia. Inicialmente, se utilizó el servidor de Estados Unidos, pero en este *Galaxy*, la herramienta de *Kraken2* no disponía de ninguna base de datos útil que clasifique directamente las secuencias de cacao, de modo que se llevaron a cabo diversas corridas para filtrar las secuencias que pertenecían a otros reinos biológicos. Eventualmente, se encontró la base de datos requerida en el servidor de Europa, logrando al fin la clasificación taxonómica del *ID* 3641 (*Theobroma cacao*).

A raíz de dicha clasificación, se trabajó inicialmente con el set de datos correspondiente a la accesión SRR377719 y para corroborar la limpieza de estos datos se corrieron diversos alineamientos con *Bowtie2*. Así, se tomaron 3 diferentes genomas referenciales y se realizó una evaluación de alineamientos, buscando las mejores coincidencias, como se puede apreciar en la Tabla 4.

Tabla 4

Evaluación de alineamientos de tres genomas referenciales de *Theobroma cacao*.

	Accesión	***Tasa global de alineamiento***
Criollo_cocoa_genome_V2	GCA_000208745.2	89.72%
y Theobroma_cacao_20110822	GCA_000403535.1	89.64%
ASM3589663v1	GCA_035896635.1	88.03%

Dado que el mejor alineamiento se logró con el genoma de referencia GCA_000208745.2, se procedió a utilizarlo para los análisis posteriores.

La clasificación antes mencionada, realizada en *Kraken2*, arrojó resultados «clasificados» y «no clasificados» contra la base de datos Genbank. Del primer grupo, «clasificados», surgieron varios *Tax Ids* identificados; entre ellos el *ID* 3641 correspondiente a *Theobroma cacao*, y otros más. A raíz de este hallazgo, se procedió a filtrar dentro de estas secuencias a todas las que poseían y no poseían el *ID* 3641 para así, en un posterior análisis, lograr determinar la limpieza de los mismos y poder determinar la contaminación real de las lecturas «clasificadas diferentes a 3641» y las lecturas «no clasificadas» posterior a la ejecución de *Kraken2*. En la Tabla 5, se puede observar este proceso y los porcentajes de alineamiento con *Bowtie2*.

Tabla 5

Resultados del alineamiento del genoma GCA_000208745.2

	GCA_000208745.2
Lectura cruda de SRR377719.1 y SRR377719.2	89.72%
SRR377719 «clasificadas» contra GenBank bdd filtradas por tax id = 3641	96.12%
SRR377719 «clasificadas diferentes a 3641» contra GenBank	80.70%
SRR377719 «no clasificadas» contra GenBank bdd	89.57%
SRR377719 «clasificadas» contra SILVA bdd	53.09%
SRR377719 «clasificadas» contra Standard dbb*	79.29%

**(archaea, bacteria, viral, plasmid, human1, UniVec_Core)*

Dado que los resultados no mostraron consistencia frente al genoma de referencia con el cual fueron alineados, el enfoque tuvo que ser cambiado. Entre estas inconsistencias se resaltan lecturas «clasificadas» taxonómicamente contra bases de datos como SILVA o Standard, que pertenecen a reinos, familias y órdenes biológicos diferentes al del cacao, pero aún así, la tasa de alineamiento con *Bowtie2* llegó a porcentajes altos como 53.09% o 79.29%. Otra inconsistencia tuvo que ver con las secuencias «no clasificadas» contra la base de datos

de *Genbank* que, igualmente, tuvo una tasa de alineamiento muy alta del 89.57%. Con dichos resultados se concluyó que, posiblemente, el genoma de referencia usado, GCA_000208745.2, no mostraba fidelidad con el análisis y, por tanto, se procedió a filtrar el genoma de referencia usando *SamlTools faidx* como herramienta (Anexo 2). En este nuevo proceso, se decidió filtrar únicamente por los 10 cromosomas + cloroplasto, propios del cacao (National Center for Biotechnology Information, 2025) y así, evitar incluir *contigs* que puedan estar causando el ruido previamente detallado.

Los resultados de este análisis con el genoma de referencia filtrado se muestran en la Tabla 6.

Tabla 6

Resultados del alineamiento del genoma GCA_000208745.2 filtrado

	GCA_000208745.2 filtrado
SRR377719 «clasificadas» contra GenBank bdd filtradas por tax id = 3641	95.32%
SRR377719 «clasificadas diferentes a 3641» contra GenBank	79.58%
SRR377719 «no clasificadas» contra GenBank bdd	86.47%
SRR377719 «clasificadas» contra SILVA bdd	46.16%
SRR377719 «clasificadas» contra Standard dbb*	66.92%
**(archaea, bacteria, viral, plasmid, human1, UniVec_Core)*	

Incluso después de haber filtrado el genoma de referencia, los resultados de las tasas de alineamiento no mejoraron mucho. Contra la base de datos *SILVA*, la tasa de alineamiento solo bajó un 6.93% mientras que contra la base de datos Standard, bajó un 12.37%.

Tomando en cuenta aquello, se decidió a proceder el análisis cambiando a la secuencia SRR21562212 que, a diferencia de la anterior, es una muestra *WGS*, en lugar de *ultra-barcoding*. Esta decisión planteó un desafío adicional, ya que la nueva secuencia contenía una cantidad de información mucho mayor y, por tanto, necesita de una capacidad

computacional mucho mayor; sin embargo, el proceso recorrido hasta este momento lo requirió.

Adicionalmente y, debido a que en la muestra anterior, SRR377719, el porcentaje de alineamiento de las lecturas «clasificadas diferentes a 3641» con otros organismos de otros reinos biológicos fue bastante alto, se decidió también proceder a clasificar con *Kraken2* a las siguientes familias:

- *Theobroma* (*tax id*: 3640) y sus hijos -> padre de *Theobroma cacao*
- *Byttnerioideae* (*tax id*: 214909) y sus hijos -> abuelo de *Theobroma cacao*

Esta decisión se planteó con el propósito de mejorar la clasificación y también para que el alineamiento con Bowtie2 obtenga un porcentaje más cercano al 0% con los «no clasificados» y más cercano al 100% con los «clasificados».

Además, se propuso llevar a cabo la alineación contra el genoma de referencia filtrado por:

- 10 cromosomas + cloroplasto
- 10 cromosomas

Esto, con el fin de identificar si alguno de estos genomas de referencia podría presentar mejores resultados frente al otro. En la tabla 7 se pueden ver los resultados de estos alineamientos.

Tabla 7

Resultados del alineamiento del genoma GCA_000208745.2 filtrado

Genoma referencia	***SRR21562212 Byttnerioideae e hijos***	***SRR21562212 no Byttnerioideae ni hijos***	***SRR21562212 Theobroma e hijos***	***SRR21562212 no Theobroma ni hijos***
Criollo filtrado 10 cromosomas + cloroplasto	93.81%	71.00%	93.70%	73.84%
Criollo filtrado 10 cromosomas	90.82%	64.25%	90.57%	67.74%

Como se puede apreciar en los resultados, tampoco se presentó una mejoría significativa en general, y tampoco hubo una gran diferencia entre los dos genomas de referencia filtrados. Así, se decidió que, en adelante, solo se tomará en cuenta el genoma de referencia filtrado con los 10 cromosomas + cloroplasto.

De esta manera, se ejecutó nuevamente *Kraken2* con parámetros diferentes a los configurados por defecto. Según Wood, Lu y Langmead (2025), el parámetro conocido como –*confidence* o *Confidence Score*, permite ajustar la precisión de las clasificaciones. Si se elige un umbral bajo se asignan etiquetas más específicas, pero con un mayor riesgo de error. Por el contrario, si se elige un umbral alto, se pueden perder clasificaciones específicas y dejar más secuencias sin clasificar. Otro parámetro, *--minimum-hit-groups* o *hit group threshold*, permite establecer un umbral mínimo de grupos de coincidencias o *hit groups*, tomándose como un requisito para que una secuencia sea clasificada. Esto podría ser útil para evitar clasificaciones basadas en coincidencias débiles o aleatorias, especialmente cuando se trabaja con bases de datos personalizadas y se quiere verificar si una secuencia realmente pertenece a un genoma específico (Wood, Lu y Langmead,2025). Tomando esto en cuenta, los parámetros se cambiaron a los siguientes valores, para intentar mejorar los resultados de clasificación de *T. cacao*, como se puede ver en la Tabla 8.

Tabla 8

Parámetros usados para mejorar la clasificación de *T. cacao* con Kraken2

	Valor anterior	***Valor nuevo***
Confidence score	0.0	0.2
Hit Group threshold	2	3

Así, se ejecutaron nuevamente 3 filtrados en base a esta clasificación con el fin de extraer únicamente las secuencias correspondientes a:

- *Theobroma cacao* (*tax id*: 3641),
- *Theobroma* (*tax id*: 3640) y sus hijos -> padre de *Theobroma cacao*
- *Byttnerioideae* (*tax id*: 214909) y sus hijos -> abuelo de *Theobroma cacao*

Todo esto se realizó con el propósito de conseguir el porcentaje de alineación más alto comparado con el genoma de referencia filtrado con 10 cromosomas + cloroplasto y para tomar las secuencias alineadas como las lecturas «clasificadas» que serán usadas para la red neuronal. Los resultados pueden apreciarse en la Tabla 9:

Tabla 9

Tasas de alineamiento sobre distintos resultados filtrados

	Bowtie2 overall alignment rate
Theobroma cacao	97.61%
Theobroma y sus hijos	93.57%
Byttnerioideae y sus hijos	94.36%

Dado que el porcentaje de alineación más alto se consiguió con las lecturas filtradas correspondientes únicamente a *Theobroma cacao*, fueron estas las secuencias utilizadas como el conjunto de datos clasificados como cacao para la red neuronal.

Simultáneamente, se ejecutaron mejoras a la clasificación de contaminantes con *Kraken2* utilizando otros valores en los parámetros antes mencionados. Primero, se ejecutó una clasificación contra la base de datos Standard (*Refeq archaea, bacteria, viral, plasmid, human1, UniVec_Core*) + *protozoa & fungi*, con un *confidence score* de 0.7 y un hit group threshold de 5. En esta ocasión los resultados fueron bastante favorables para identificar los contaminantes de una manera confiable. Los resultados mostraron un alineamiento bajo del 15.97%, usando *Bowtie2* con el genoma de referencia filtrado. Por esta razón,se escogieron las lecturas no alineadas de esta ejecución para usarse como el primer conjunto de datos contaminados que se usarán en la red neuronal posteriormente.

Como siguiente paso, se decidió conseguir más datos contaminados. Para el efecto,el conjunto de datos no clasificados en la ejecución anterior de *Kraken2* fueron usados como data de entrada para una nueva ejecución en *Kraken2, pero esta vez,* contra la base de datos *SILVA.* Esta base de datos es conocida por contener información taxonómica de bacterias, arqueas y eucariotas. También, en esta nueva ejecución el *confidence score* se configuró con

un valor de 0.8 y un *hit group threshold* de 4. De esta nueva ejecución, los datos clasificados obtuvieron un alineamiento del 44.74% que, a pesar de no ser tan bajo como planteaba la hipótesis inicial, si permitió tomar las secuencias no alineadas para ser usadas como el segundo conjunto de datos contaminados para la red neuronal. Los resultados finales de todo este proceso se presentan en la Tabla 10 a continuación:

Tabla 10

Resultados finales de las ejecuciones en Kraken

	Secuencias totales	***Porcentaje***
SRR21562212	72.717.892	100%
<<clasificadas>>	10.489.596	14.43%
<<no clasificadas>>	402.755 (*pluspf*) + 85.252 (*SILVA*) = 488.007	0.67%
Lecturas desconocidas	61.740.289	84.90%

Es entonces que salta a la vista que las lecturas desconocidas mostradas en la Tabla 10, representan un gran porcentaje de la muestra inicial SRR21562212. Así, se decidió proceder a un contraste adicional para entender la realidad de dicha secuencia, es decir, saber cuántos datos, que si correspondían a *Theobroma cacao,* no pudieron ser clasificados con éxito por *Kraken2*. Entonces, se optó por realizar un *Bowtie2* contra el genoma de referencia de *Theobroma cacao*. El resultado de este proceso arrojó una tasa de alineamiento del 89.80%, demostrando parcialmente la ineficacia de *Kraken2* al no haber podido clasificar una inmensa cantidad de las secuencias que efectivamente correspondían a *Theobroma cacao*.

Todo este proceso dejó abiertas nuevas hipótesis sobre el porqué de la ineficacia de *Kraken2*, dudas que se plantean como conclusiones en su respectivo capítulo. Sin embargo, con fines pragmáticos para la consecución del presente trabajo, se decidió proceder

únicamente con las secuencias efectivamente «clasificadas» y «no clasificadas», independientemente de sus porcentajes.

Finalmente, en las Tablas 11, 12 y 13 se muestran los recursos computacionales usados para la clasificación exitosa con Kraken2.

Tabla 11

Recursos computacionales usados para la obtención de secuencias «clasificadas» con *Kraken2.*

Tiempo de uso de CPU	3 horas y 23 minutos
Uso máximo de memoria registrado	289.9 GB
Núcleos asignados	2
RAM asignada	70 GB
	Apéndice 2 para referencia

Tabla 12

Recursos computacionales usados para la obtención de secuencias «no clasificadas» con *Kraken2* y base de datos pluspf.

Tiempo de uso de CPU	2 horas y 54 minutos
Uso máximo de memoria registrado	151.6 GB
Núcleos asignados	2
RAM asignada	70 GB
	Apéndice 3 para referencia

Tabla 13

Recursos computacionales usados para la obtención de secuencias «no clasificadas» con *Kraken2* y base de datos SILVA.

Tiempo de uso de CPU	1 hora y 10 minutos
Uso máximo de memoria registrado	70 GB
Núcleos asignados	2
RAM asignada	70 GB
	Apéndice 4 para referencia

3.3. ***Preprocesamiento de datos***

Para poder tener una mejor comprensión sobre esta sección es importante mencionar algunas consideraciones relativas a la elaboración de los modelos de redes neuronales, cuyo desarrollo a profundidad se presenta en el punto 3.2. de la presente investigación.

Entre dichas consideraciones, se resalta que el conjunto de datos obtenidos anteriormente como «clasificados» y «no clasificados», fueron divididos de tal forma que el 80% de datos se utilizaron para entrenamiento, mientras que el 20% restante, para la validación final con *BERTax Cacao*.

Adicionalmente, de dicho 80% de datos, se usó el 10% para determinar el ajuste de hiperparámetros más óptimo, mientras que el 90% restante para el pre-entrenamiento y *fine-tunning*. Esta división de datos y su planificación por etapas puede apreciarse con más detalle en la Tabla 14. Esta información requirió ser adelantada pues, es clave comprenderla para la correspondiente preparación y división de datos.

Tabla 14

División de datos para cada etapa de la creación de la red neuronal

	División de datos		**# Secuencias**
Ajuste de Hiperparámetros	80%	8%	878.208,24 ≈ **878.208**
Pre-entrenamiento con *BERTax Cacao*		72%	7.903.874,16 ≈ **7.903.874**
Fine tuning* con *BERTax Cacao			
***BERTax Cacao* validación final**	20%	20%	2.195.520,6 ≈ **2.195.521**
TOTAL	100%		**10.977.603**

También se recalca que los datos deben estar distribuidos proporcionalmente en cada etapa o época del entrenamiento de la red neuronal para asegurar un aprendizaje coherente. Dado que el número total de secuencias entre «clasificadas» y «no clasificadas» es 10.977.603 (Tabla 9), la distribución de datos para «clasificados» corresponde al 95.55%

mientras la de «no clasificados» representa el 4.45%. Con estos porcentajes establecidos, se decidió que, en cada etapa del entrenamiento de la red neuronal los datos deben estar distribuidos como se muestra en la Tabla 15.

Tabla 15

Distribución de datos para cada etapa de la creación de la red neuronal

	Total secuencias por etapa	***# Secuencias «clasificadas» (95.55%)***	***# Secuencias «no clasificadas» (4.45%)***
Ajuste de Hiperparámetros	878.208	839.127,74 ≈ **839.128**	39.080,26 ≈ **39.080**
Pre-entrenamiento con BERTax Cacao	7.903.874	7.552.151,71 ≈ **7.552.152**	351.722,39 ≈ **351.722**
Fine tuning con BERTax Cacao			
BERTax Cacao validación final	2.195.521	2.097.820,32 ≈ **2.097.820**	97.700,68 ≈ **97.701**

Una vez establecida la división de datos para cada etapa, fue necesaria la conversión de la totalidad de los datos a un formato *fasta* para su posterior procesamiento con el script: *fasta2fragments.py*, proporcionado por el repositorio *BERTax-training*. Este tipo de formato es necesario para el pre-entrenamiento con *BERTax.*

Para esta etapa también se crearon 2 scripts con el fin de extraer y combinar los fragmentos: *extract_fragments.py* y *combine_fragments.py*, de modo que el directorio que sirve de *input* para la red neuronal tenga la estructura requerida, así:

fragments_dir:

→ Cacao_fragments.json

→ *Cacao_species_picked.txt*

→ *NotCacao_fragments.json*

→ *NotCacao_species_picked.txt*

A continuación, se muestran los comandos requeridos en el preprocesamiento de los datos para cada etapa de la red neuronal, junto con sus *entradas* y *salidas*(Tablas 16-19).

1. Uso de *extract_fragments.py* para extraer secuencias de cacao:

```
python extract_fragments.py
"clasificados/output_cacao_fragments.json"
"clasificados/output_cacao_species_picked.txt"
Cacao_fragments.json Cacao_species_picked.txt
```

Tabla 16

Resultados de extracción de fragmentos de cacao para *BERTax Cacao.*

Etapa	***Entrada***	***Salida***
Ajuste de hiperparámetros	*Total sequences available:* ***10489596*** *Enter the number of sequences to extract:* ***839128***	*Successfully extracted* ***839128*** *sequences to Cacao_fragments.json and Cacao_species_picked.txt Remaining sequences in master files:* ***9650468***
Pre-entrenamiento Fine-tunning	*Total sequences available:* ***9650468*** *Enter the number of sequences to extract:* ***7552152***	*Successfully extracted* ***7552152*** *sequences to Cacao_fragments.json and Cacao_species_picked.txt Remaining sequences in master files:* ***2098316***

2. Uso de *extract_fragments.py* para extraer secuencias contaminadas de la *BDD pluspf:*

```
python extract_fragments.py
"no_clasificados/output_plusf_fragments.json"
"no_clasificados/output_plusf_species_picked.txt"
NotCacao_fragments1.json NotCacao_species_picked1.txt
```

Tabla 17

Resultados de extracción de fragmentos de contaminantes *pluspf* para *BERTax Cacao.*

Etapa	***Entrada***	***Salida***
Ajuste de hiperparámetros	*Total sequences available:* ***402755*** *Enter the number of sequences to extract:* ***32253***	*Successfully extracted* ***32253*** *sequences to NotCacao_fragments1.json and NotCacao_species_picked1.txt Remaining sequences in master files:* ***370502***
Pre-entrenamiento Fine-tunning	*Total sequences available:* ***370502*** *Enter the number of sequences to extract:* ***290280***	*Successfully extracted* ***290280*** *sequences to NotCacao_fragments1.json and NotCacao_species_picked1.txt Remaining sequences in master files:* ***80222***

3. Uso de *extract_fragments.py* para extraer secuencias contaminadas de la *BDD SILVA:*

```
python extract_fragments.py
"no_clasificados/output_silva_fragments.json"
"no_clasificados/output_silva_species_picked.txt"
NotCacao_fragments2.json NotCacao_species_picked2.txt
```

Tabla 18

Resultados de extracción de fragmentos de contaminantes Silva para *BERTax Cacao.*

Etapa	***Entrada***	***Salida***

Ajuste de hiperparámetros	*Total sequences available:* ***85252*** *Enter the number of sequences to extract:* ***6827***	*Successfully extracted* ***6827*** *sequences to NotCacao_fragments2.json and NotCacao_species_picked2.txt Remaining sequences in master files:* ***78425***
Pre-entrenamiento Fine-tunning	*Total sequences available:* ***78425*** *Enter the number of sequences to extract:* ***61442***	*Successfully extracted* ***61442*** *sequences to NotCacao_fragments2.json and NotCacao_species_picked2.txt Remaining sequences in master files:* ***16983***

4. Uso de *combine_fragments.py* para combinar las secuencias de *pluspf* y *SILVA*:

```
python combine_fragments.py
"no_clasificados/NotCacao_fragments1.json"
"no_clasificados/NotCacao_species_picked1.txt"
"no_clasificados/NotCacao_fragments2.json"
"no_clasificados/NotCacao_species_picked2.txt"
NotCacao_fragments.json NotCacao_species_picked.txt
```

Tabla 19

Resultados de combinación de contaminantes *pluspf* y silva para *BERTax Cacao.*

Etapa	***Salida***
Ajuste de hiperparámetros	*Successfully combined files into NotCacao_fragments.json and NotCacao_species_picked.txt* *Total sequences in combined file:* ***39080***
Pre-entrenamiento Fine-tunning	*Successfully combined files into NotCacao_fragments.json and NotCacao_species_picked.txt*

	Total sequences in combined file: ***351722***
Validación final	*Successfully combined files into NotCacao_fragments.json and NotCacao_species_picked.txt* *Total sequences in combined file:* **97701**

3.4. ***Entrenamiento y validación de red neuronal***

Se decidió usar como punto de partida la arquitectura *BERTax,* realizada por F. Mock, F. Kretschmer, A. Kriese, S. Böcker y M. Marz (2021), debido a su alta similitud con lo propuesto en la presente investigación, siendo usada para la clasificación taxonómica con entradas de secuencias de nucleótidos y enfocada a la clasificación por reinos: Virus, *Archaea*, *Bacteria*, *Eukaryota*.

En este proceso, al intentar modificar el código para adaptar la arquitectura a una clasificación binaria de *Cacao* y *NotCacao*, se descubrió que esta arquitectura tenía una limitante de la versión de *tensorflow*, ya que fue construída hace 4 años. Es decir, *BERTax* funciona correctamente sólo con la versión 2.2.0 de *tensorflow,* lanzada en 2020*,* y con todas las versiones compatibles de los paquetes requeridos para *BERTax*, que tampoco son vigentes. Esta diferencia de versiones se muestra en la Tabla 20:

Tabla 20

Versiones antiguas y actuales de paquetes para *BERTax.*

Paquete	***Versión requerida para BERTax 2021***	***Última versión estable***
tensorflow	2.2.0	2.18.0
python	3.8	3.11
scikit-learn	1.0.2	1.6.1
SciPy	1.4.1	1.15.2
keras-preprocessing	1.0.1	1.1.2
CUDA	10.1	12.8.1
cuDNN	7.6	8.9.7
NumPy	1.19.5	2.2.4
pandas	1.1.5	2.2.3
Matplotlib	3.3.4	3.10.1

Según NVIDIA (s.f.), es recomendable realizar el entrenamiento de redes neuronales con GPU debido a la alta demanda de millones de operaciones matemáticas en paralelo, hacerlo con CPU es también posible pero su procesamiento sería mucho más lento. Esto constituyó un reto a abordar en la presente investigación; así, se buscó mejorar el tiempo de procesamiento implementando el modelo con GPU en lugar de CPU. Para el efecto, se requería utilizar las versiones de *CUDA* y *cuDNN* compatibles con *tensorflow 2.2.0*, pero, durante este procedimiento se presentó un problema bastante conocido en la comunidad, mencionado en algunos foros como el de smoreira00 (2023). Dicho foro indica que se debe actualizar la versión de *tensorflow* a 2.8.0 y superiores como la solución definitiva.

Para abordar este problema se procedió a actualizar el código correspondiente a *bertax_training* y *bertax*, de modo que funcione con las nuevas versiones y se pueda procesar con GPU.

Todos los modelos corregidos y actualizados para la elaboración del presente trabajo y que, desde ahora se llamarán *BERTax Cacao*, están disponibles en el siguiente repositorio en GitHub:

- https://github.com/mariana2323/bertax_cacao_training
- https://github.com/mariana2323/bertax_cacao

3.4.1. **Ajuste de Hiperparámetros (Tuning) – Fase Preliminar**

Comenzando con esta fase, fue necesario llevar a cabo el pre-entrenamiento del 8% de los datos previstos para la fase de ajuste, ya que el pre-entrenamiento con *BERTax Cacao* es necesario para *tokenizar* los fragmentos, o en otras palabras, para aprender representaciones del lenguaje (secuencias de nucleótidos de *Cacao* y *NotCacao*). Entonces, se previó realizar este proceso en 10 épocas y con un margen de paciencia de 5 épocas. Sin embargo, durante esta ejecución y debido al margen de paciencia, el proceso se detuvo en 8 épocas con 878.208 fragmentos procesados, evitando un sobreajuste. (Gráfico 1) (Tabla 21).

Gráfico 1

Pérdida de entrenamiento versus pérdida de validación en la etapa de pre-entrenamiento con *BERTax Cacao* en 8 épocas para el 8% de los datos.

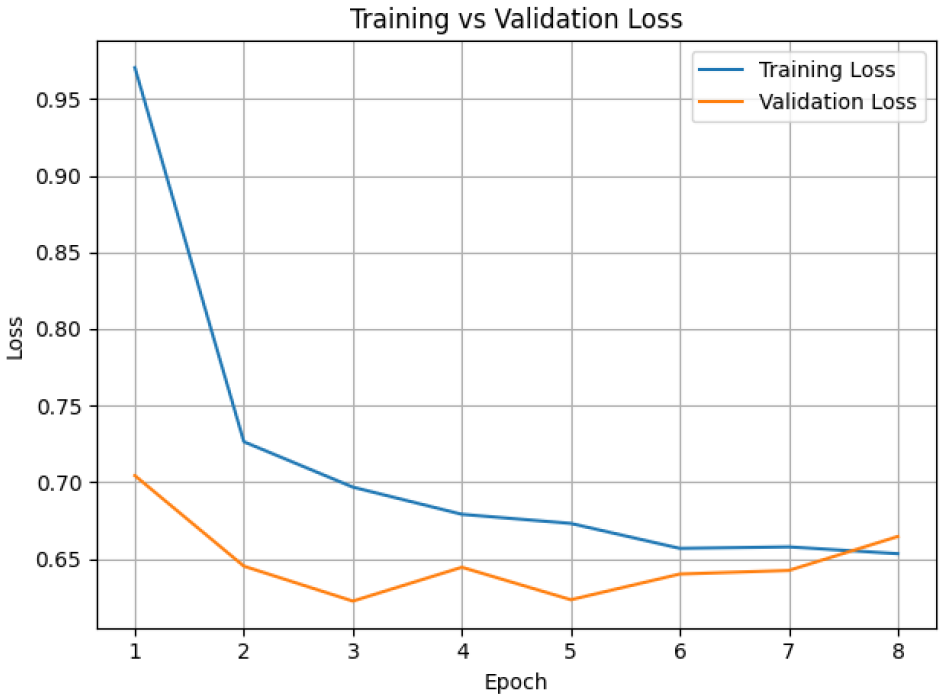


Tabla 21

Pérdida de entrenamiento y validación del pre-entrenamiento con *BERTax Cacao* en 8 épocas para el 8% de los datos.

epoch	loss	val_loss

1	0.970411121845245	0.704327583312988
2	0.72640323638916	0.645249605178833
3	0.696801722049713	0.622456789016723
4	0.679075539112091	0.644529163837432
5	0.673120558261871	0.623348176479339
6	0.656844615936279	0.640112042427063
7	0.657833874225616	0.642464876174926
8	0.653397142887115	0.664599418640136

Aunque la pérdida de validación, *val_loss,* no podría ser considerada ideal en un entrenamiento común, el objetivo principal en esta etapa no giraba en torno a aquello. El objetivo real era lograr que el modelo aprenda las representaciones del lenguaje, *tokenice* secuencias y pueda continuar con su aprendizaje en las siguientes etapas. Estas decisiones se justifican gracias a criterios como los planteados por J. Devlin, M. Chang, K. Lee y K. Toutanova (2018). Adicionalmente, mientras *val_loss* sea menor a 1, no se está rebasando el umbral permitido para entrenamiento.

Posteriormente, este modelo se guardó en el archivo *bert_nc_trained.keras*, que fue cargado para así proceder a la fase de *fine-tunning*. Aquí se ajustaron los resultados con *keras-tuner*, el cuál buscó aleatoriamente la mejor combinación de hiperparámetros, siendo estos, en este caso específico, *optimizer* y *learning rate*.

Se configuró el tuner de keras con 20 iteraciones de búsqueda y 5 épocas para cada una de ellas, con *learning_rates* aleatorios y 3 diferentes tipos de *Optimizers:* Adam, RMSprop *y* seSGD, todos seleccionados aleatoriamente por el *tuner*. A continuación se pueden apreciar los mejores hiperparámetros encontrados por el tuner en la Tabla 22.

Tabla 22

Mejores hiperparámetros encontrados en fase de ajuste.

Learning rate	5.0973e-05
Optimizer	Adam
Mejor val_accuracy	0.94595456

Posteriormente, se ejecutó el modelo en un *runtime* de *Google Colab Pro L4* y los resultados de recursos computacionales usados para este procesamiento se presentan en la Tabla 23.

Tabla 23

Recursos computacionales usados en fase de ajuste de hiperparámetros.

Unidades de cómputo por hora	2.09 compute units/ hora
RAM del sistema	13.9 GB usadas de 53 GB disponibles
GPU RAM	4.5 GB usadas de 22.5 GB disponibles
Uso de disco	40.1 GB usadas de 235.7 GB disponibles
Tiempo de ejecución	5 horas 51 minutos

Apéndice 5 para referencia

3.4.2. **Pre-entrenamiento con *BERTax Cacao***

Continuando con esta fase, se ejecutaron los fragmentos correspondientes al 72% de los datos totales, en 10 épocas. Después de dicha ejecución y como se puede ver en el Gráfico 2 y la Tabla 24, se constata que el modelo efectivamente está aprendiendo. Esto se demuestra ya que, tanto la pérdida de entrenamiento como la de validación disminuyen de forma sostenida. Aún así, como se mencionó anteriormente, el propósito fundamental de esta etapa sigue siendo que el modelo entienda el lenguaje para su posterior afinamiento y validación final a través de *BERT*. Adicionalmente, cabe resaltar que, en la época 10, *val_loss* tiene un valor bastante cercano a cero (0.3152), por lo que resulta conveniente para la siguiente etapa de *fine-tuning.*

Gráfico 2

Pérdida de entrenamiento versus pérdida de validación en la etapa de pre-entrenamiento con *BERTax Cacao* en 10 épocas para el 72% de los datos.

Tabla 24

Pérdida de entrenamiento y validación del pre-entrenamiento con *BERTax Cacao* en 10 épocas para el 72% de los datos.

epoch	***loss***	***val_loss***
1	0.692544639110565	0.613662421703338
2	0.616087555885314	0.559797465801239
3	0.565764665603637	0.511409878730773
4	0.516050398349762	0.482971549034118
5	0.473462909460067	0.431691259145736
6	0.437427520751953	0.457461506128311
7	0.402801394462585	0.35775750875473
8	0.374114423990249	0.33594411611557
9	0.348317176103591	0.315245926380157
10	0.325877457857131	0.286323577165603

Posteriormente, se ejecutó el modelo en un *runtime* de *Google Colab Pro L4* y los resultados de recursos computacionales usados para este procesamiento se presentan en la Tabla 25.

Tabla 25

Recursos computacionales usados en fase de pre-entrenamiento del 72% de los datos.

Unidades de cómputo por hora	2.09 compute units/ hora
RAM del sistema	7.9 GB usadas de 53 GB disponibles
GPU RAM	8.7 GB usadas de 22.5 GB disponibles
Uso de disco	40.1 GB usadas de 235.7 GB disponibles
Tiempo de ejecución	4 horas 8 minutos
	Apéndice 6 para referencia

3.4.3. *Fine-tuning* con *BERTax Cacao*

Posteriormente, en esta fase se ejecutaron los fragmentos correspondientes al 72% de los datos totales con los hiperparámetros señalados en la Tabla 26.

Tabla 26

Hiperparámetros usados para la fase de fine-tunning.

Learning rate	5.0973e-05
Optimizer	Adam
# épocas	10
Márgen de paciencia	2 epochs

El modelo logró un buen rendimiento tanto en entrenamiento como en validación, cómo se puede constatar en el Gráfico 3 y la Tabla 27. Se puede apreciar las mejores en exactitud con cada época, tanto para entrenamiento como validación, además de que la curva de validación se mantiene bastante cerca de la de entrenamiento, evitando un posible sobreajuste; esto, dado al márgen de paciencia de 2 épocas que se configuró en el modelo. Se puede apreciar también que la exactitud de validación subió y bajó 2 veces, algo bastante normal debido a los lotes de validación entre época y época. Se demuestra un buen aprendizaje, con el mejor valor de exactitud logrado, siendo de 98.51%.

Gráfico 3

Exactitud de entrenamiento versus exactitud de validación en la etapa de *fine-tuning* con *BERTax Cacao* en 8 épocas para el 72% de los datos.

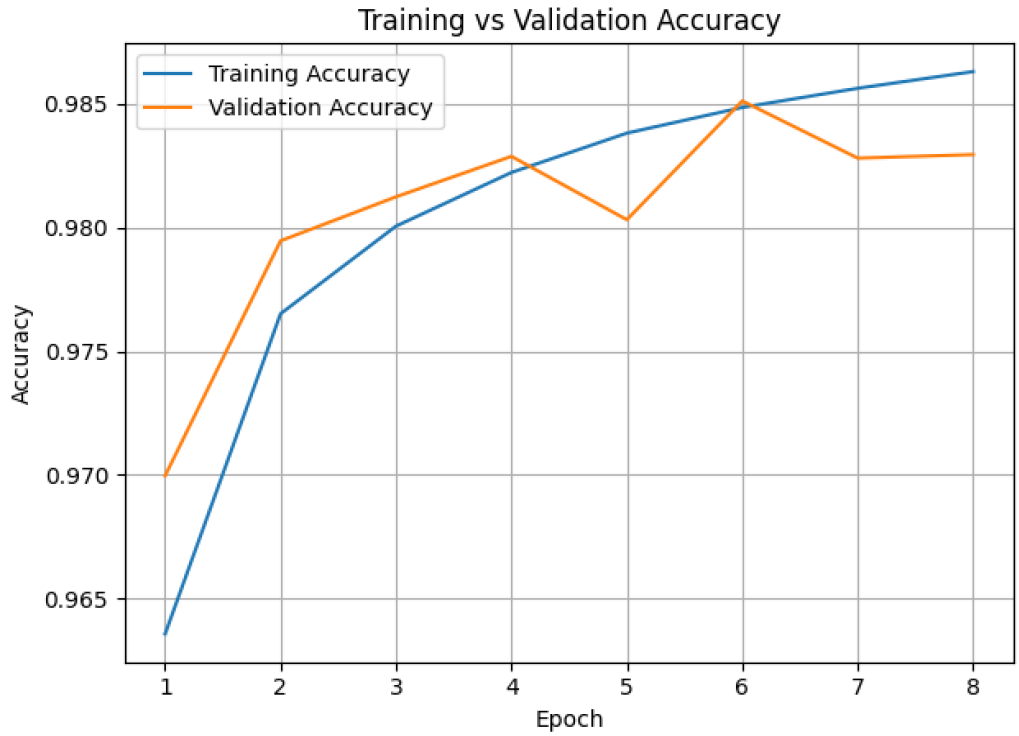


Tabla 27

Exactitud de entrenamiento y validación del *fine-tuning* con *BERTax Cacao* en 8 épocas para el 72% de los datos.

epoch	***accuracy***	***val_accuracy***
1	0.963579535484314	0.969969451427459
2	0.976512134075164	0.979458391666412
3	0.980051159858703	0.981235325336456
4	0.982219040393829	0.982870161533355
5	0.983808994293212	0.980311334133148
6	0.984847068786621	0.985109090805053
7	0.985619604587554	0.982799053192138
8	0.986294865608215	0.982941210269928

Adicionalmente, se obtuvo un *balanced accuracy* o exactitud balanceada de 98.56%, que representa la media entre la especificidad de (*Not Cacao*) y la sensibilidad de (*Cacao*) y

cuyo resultado es bastante positivo. En la Tabla 28 se puede revisar la matriz de confusión resultante, en donde se detalla información requerida para calcular dicho valor.

Tabla 28

Matriz de confusión resultante de la etapa de *fine-tuning* con *BERTax Cacao*.

	Predicho (Not Cacao)	***Predicho (Cacao)***	***Total reales***
Realmente (Not Cacao)	69790 aciertos	555 errores	70345
Realmente (Cacao)	1469 errores	68875 aciertos	70344
Total	71259	69430	140689

- Verdaderos negativos (TN): 69.790,
- Falsos positivos (FP): 555.
    - Que obtienen una especificidad del 99.21%
- Verdaderos positivos (TP): 68.875,
- Falsos negativos (FN): 1.469
    - Que obtienen una sensibilidad de 97.92%

Finalmente, se ejecutó el modelo en un *runtime* de *Google Colab Pro L4* y los resultados de recursos computacionales usados para este procesamiento se presentan en la tabla 29.

Tabla 29

Recursos computacionales usados en fase de *fine-tuning* con *BERTax Cacao* del 72% de los datos.

Unidades de cómputo por hora	4.18 compute units/ hora
RAM del sistema	6.8 GB usadas de 53 GB disponibles
GPU RAM	4.5 GB usadas de 22.5 GB disponibles
Uso de disco	39 GB usadas de 235.7 GB disponibles
Tiempo de ejecución	1 horas 57 minutos
	Apéndice 7 para referencia

3.4.4. **Validación del modelo final**

Finalmente, con el modelo ya entrenado, con el proceso explicado en la sección metodológica, se procedió a hacer la validación final. Para el efecto, se utilizó el 20% de datos reservados desde el principio, es decir, 2.195.521 secuencias. Esta muestra se compone del 95.55% de secuencias «clasificadas» (2.097.820 aproximadamente) y 4.45% de secuencias «no clasificadas» (97.701 aproximadamente).

También, para esta última etapa se utilizó el repositorio *https://github.com/rnajena/bertax* (Kretschmer et al., 2022), sin embargo, se realizaron algunas modificaciones en su código con el fin de asegurar la validación de la predicción binaria de las etiquetas *Cacao* y *NotCacao.* Adicionalmente, el input de esta fase requería un archivo *fasta* y el modelo *BERTAX Cacao,* previamente guardado en la fase de *fine-tuning*.

Posteriormente, y como primer intento, se ejecutó la predicción con el 20% de los datos. Sin embargo, este proceso tardó más de 12 horas y no mostraba señales de concluir. Con ese antecedente, se decidió detener dicha ejecución para analizar las capacidades computacionales y, así, entender cómo poder mejorar el rendimiento de dicho procesamiento. Cabe destacar que esta ejecución se estaba llevando a cabo en un *runtime* de tipo *L4 GPU* en *Google Colab*, en el cual, apenas se usaban 0.4 GB de las 22.5 GB de GPU RAM disponibles, según se descubrió en el análisis planteado, indicando un manejo muy ineficiente de la predicción.

Tras un posterior análisis más profundo, se logró identificar que el parámetro *batch_size* del método *model.predict()* estaba siendo usado incorrectamente, provocando que se ejecute “secuencia por secuencia” y excluyendo el comportamiento *batch_size*, el cuál debería aprovechar las capacidades de procesamiento en paralelo para GPU. Ya identificado el problema, se procedió a refactorizar todo el código, prácticamente en su totalidad, y aprovechar al máximo el uso de la GPU para un nuevo procesamiento eficiente y final.

Una vez afinado el procesamiento con GPU, utilizando un *batch_size* dinámico y habiendo limpiado la memoria RAM y GPU RAM, se logró llevar a cabo una ejecución total de las predicciones en solo 6 minutos. En la Tabla 30 se puede revisar el detalle de los recursos computacionales usados para esta fase en un *runtime* de tipo A100, el cuál dispone de mucha más memoria de RAM del sistema y GPU RAM, comparado con *L4 GPU*.

Tabla 30

Recursos computacionales usados en fase de *validación* con *BERTax Cacao* con el 20% de los datos.

Unidades de cómputo por hora	7.62 compute units/ hora
RAM del sistema	15 GB usadas de 83.5 GB disponibles
GPU RAM	32.5 GB usadas de 40 GB disponibles
Uso de disco	37.8 GB usadas de 235.7 GB disponibles
Tiempo de ejecución	26 minutos
	Apéndice 8 para referencia

5. Resultados

En la Tabla 31 se presentan los resultados referentes a la predicción del 20% de datos restantes, es decir, 2.195.521 secuencias. De dicha totalidad, 2.097.820 son de *Cacao* y 97.701 son de *NotCacao*.

Tabla 31

Resultados de predicción en la fase de validación con el 20% de los datos.

Etiqueta	**# predicciones**
Cacao	2.162.285
NotCacao	33.236

Adicionalmente, se obtuvieron los siguientes resultados en cuanto a la probabilidad predicha para ambas etiquetas, como se muestra en la Tabla 32.

Tabla 32

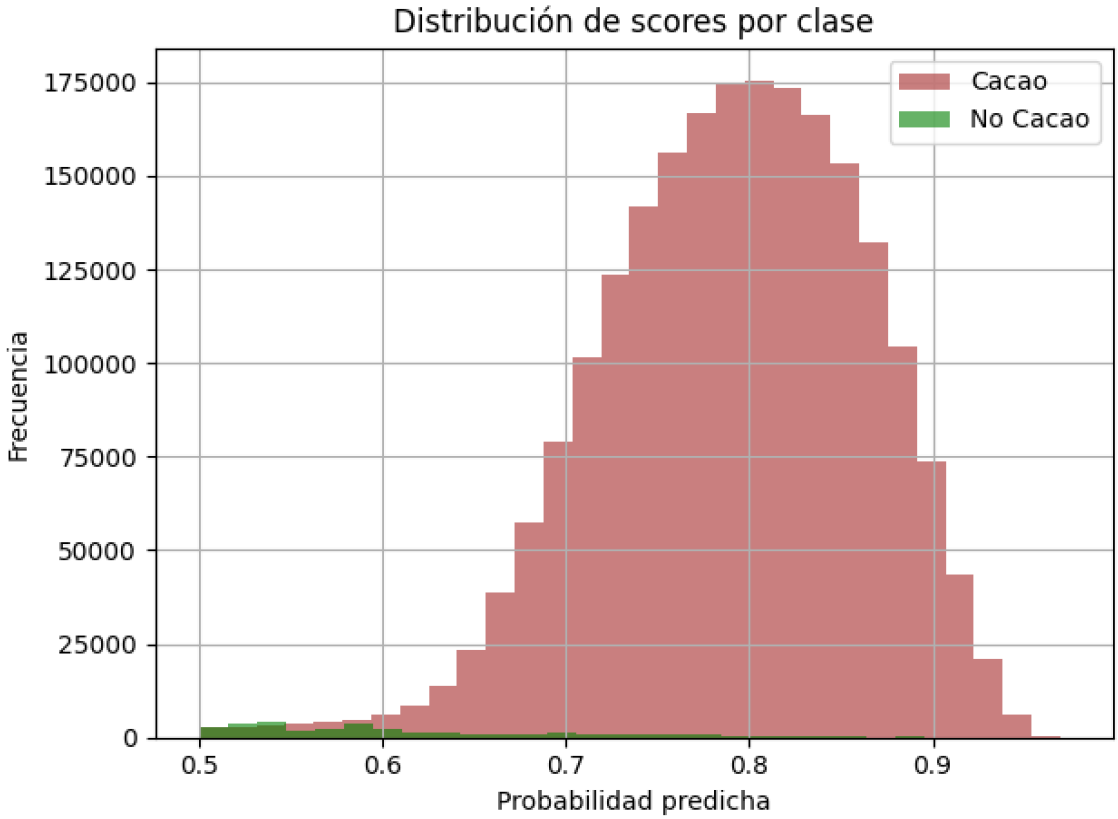
Resultados de promedio de probabilidad predicha en la fase de validación con el 20% de los datos.

Etiqueta	**Promedio de confiabilidad de predicción**
Cacao	0.7904
NotCacao	0.6145

A continuación, en el Gráfico 4, se puede apreciar la frecuencia de clasificación versus la probabilidad predicha o nivel de confianza de la predicción para *Cacao* y *NotCacao*.

Gráfico 4

Gráfico de frecuencias vs probabilidad predicha para *Cacao* y *NotCacao*.



6. Análisis de resultados

La Tabla 31 se puede interpretar como una sobrecalificación de la etiqueta de *Cacao*, ya que el modelo clasificó 64.465 falsos positivos. A partir de esta información, se aprecia también que el modelo predijo como *Cacao* al 98% de las secuencias totales. Adicionalmente, resalta un bajo *recall* para la etiqueta *NotCacao*, habiendo clasificado correctamente 33.236 secuencias de 97.701 reales.

En la Tabla 32 se puede percibir que el modelo posee una confianza alta (79.04%) a la hora de etiquetar *Cacao* y una medianamente alta (61.45%) para etiquetar *NotCacao.*

Finalmente, en el Gráfico 4 se puede interpretar un mayor sesgo de clasificación hacia la etiqueta *Cacao*, lo cual se consideraría correcto debido a la distribución usada, tanto en las fases de entrenamiento como en la fase final de validación. Entonces, la representación de *Cacao* en los datos fue de 95.55% versus un 4.45% de contaminantes, respondiendo a la índole de la problemática que nos concierne, es decir, la limpieza de contaminantes de las secuencias crudas de *T. cacao.*

Para corroborar la tendencia de los datos, se procedió a filtrar las secuencias clasificadas como Cacao en un nuevo archivo fasta y corroborar con una alineación en Bowtie2. Tras llevar a cabo esta ejecución se obtuvo una tasa de alineación del 95.86%, lo cual indica que efectivamente muchas de las secuencias clasificadas como Cacao, efectivamente lo son y poseen una tasa alta que confirma la confiabilidad de BERTax Cacao para la predicción de secuencias de Cacao.

7. Conclusiones y Recomendaciones

- Se implementó una red neuronal que predice la clasificación de *Theobroma cacao* en una lectura cruda de secuenciación de Illumina para la limpieza de contaminantes de la misma, con un porcentaje de confianza del 79.04%.
- Se generó un conjunto de datos a partir de la predicción final, el cuál se alineó con Bowtie2 y se obtuvo una tasa de alineación de 95.86%, lo cual indica que el dicho conjunto de datos si resulta ser de cacao en su mayoría pero también que puede mejorar su precisión.
- La clasificación de *Theobroma cacao* con técnicas tradicionales dió como resultado un procesamiento de 7 horas y 27 minutos con el uso de 70 GB de RAM y un máximo de memoria registrado de 289.9 GB, en donde sólo se clasificó exitosamente el 14.43% de los datos. Por otro lado, el procesamiento clasificatorio con *BERTax Cacao* se dió en 7 minutos solo para un 20% del mismo set de datos. A pesar de haber usado distintos entornos computacionales, se demuestra que el rendimiento de la red neuronal es mucho más eficiente que la clasificación con técnicas de limpieza tradicionales.
- Se generaron dos hipótesis con respecto a la precisión de clasificación de Kraken2, debido a que *Kraken2* no funcionó correctamente al no clasificar el 84.90% de los datos. Es importante recordar que, debido a las limitaciones computacionales propias de esta investigación, mencionadas con anterioridad, este proceso se realizó únicamente a través de Galaxy y su base de datos Genbank. La primera hipótesis podría suponer que el malfuncionamiento tuvo que ver con el hecho de que, posiblemente, la base de datos de Genbank, disponible en Galaxy, era menor que la base de datos taxonómica completa que se suele usar para *Kraken2* con sus capacidades computacionales completas. Sin embargo, cabe resaltar que *Kraken2* dentro de Galaxy, aun con todas estas posibles limitaciones, si pudo clasificar correctamente un 14.43% de las secuencias de *Theobroma cacao*. Entonces, esta

aparente contradicción abre la puerta a la segunda hipótesis que plantearía que, simplemente, *Kraken2* no es suficientemente preciso para este proceso clasificatorio.

- Se recomienda disponer de altos recursos computacionales, específicamente una RAM de al menos 128 GB, para asegurar una ejecución completa de *Kraken2* con la base taxonómica completa.
- Se recomienda utilizar un *batch_size* dinámico que empiece en 1028 para las fases de ajuste, pre-entrenamiento y fine-tunning con un *runtime type A100* con el fin de incrementar aún más el rendimiento.
- Se recomienda agregar más iteraciones, con al menos 10 épocas por iteración, en el *tuner search* para la fase de ajuste de hiperparámetros, con el fin de obtener un *val_accuracy* aún mayor al 94.59% obtenido en este trabajo.
- Se recomienda repetir el entrenamiento desde las fases iniciales con un enfoque diferente respecto a los datos. Se recuerda que, dado que la intención de esta red neuronal era limpiar las secuencias crudas de *T. cacao*, se procedió a utilizar un conjunto de datos donde la distribución entre datos de *T. cacao* es mayoritaria a la de contaminantes. Así, aunque el resultado sea positivo, no es totalmente preciso. Con dicho antecedente, se recomienda que un nuevo enfoque, en un futuro proyecto, realice el entrenamiento con datos distribuidos equitativamente, tanto para *Cacao* como para *NotCacao*.
- Se recomienda elaborar una plataforma web especializada en limpieza de secuencias crudas de *T. cacao* enfocada a la comunidad científica, investigadores y/o compañías especializadas en el estudio del cacao.
- Se recomienda ampliar el alcance de esta red neuronal para clasificar otras especies diferentes al cacao, que también podrían necesitar frecuentemente una limpieza genómica de sus muestras.
- Se recomienda elaborar una plataforma web que clasifique secuencias en base a los *Super Kingdoms*, como lo plantea la arquitectura *BERTax*. Cabe destacar que dicha arquitectura se encuentra desactualizada y no funciona con GPU, pero el presente

proyecto, *BERTax Cacao*, mejoró el modelo base, actualizándolo y otorgándole más eficiencia. Así, las posibilidades de nuevas plataformas, como la recomendada, podrían ser muy útiles para la comunidad científica.

8. Referencias

Adeniyi, Dele. (2019). Diversity of cacao pathogens and impact on yield and global production. In I. S. Manickavasagan & C. Thangavel (Eds.), Cocoa (pp. 55-82). IntechOpen. https://doi.org/10.5772/intechopen.83590

Anand, C. (2024, julio 28). *Mastering Kraken2 - Part 2 - Performance optimisation*. Avil Page. https://avilpage.com/2024/07/mastering-kraken2-performance-optimisation.html

Aneja, M., Gianfagna, T., T. Shewfelt, R., Luthria, D. L., & Luthria, D. L. (2015). Cacao (Theobroma cacao L.)—A Source of Health Enhancing Compounds: A Review. Critical Reviews in Food Science and Nutrition, 55(10), 1457-1469. https://doi.org/10.1080/10408398.2012.706855

Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. Molecular Systems Biology, 12(7), 878. https://doi.org/10.15252/msb.20156651

Argout, X., et al. (2011). The genome of Theobroma cacao. *Nature Genetics, 43*, 101-108. https://doi.org/10.1038/ng.736

Boakye, R. G., Stanley, D. A., & White, B. (2023). Honey contamination from plant protection products approved for cocoa (*Theobroma cacao*) cultivation: A systematic review of existing research and methods. *PLoS ONE, 18*(10), 1–23. https://doi.org/10.1371/journal.pone.0280175

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics, 30(15), 2114–2120. https://doi.org/10.1093/bioinformatics/btu170

Britannica. (2024). Cacao (Theobroma cacao). Recuperado de https://www.britannica.com/plant/cacao

Caelen, O., & Blete, M.-A. (2024). Developing apps with GPT-4 and ChatGPT (2nd ed.). O'Reilly Media, Inc. Chapter 1

Cilas, C., & Bastide, P. (2020). Challenges to cocoa production in the face of climate change and the spread of pests and diseases. Agronomy, 10(9), 1232. https://doi.org/10.3390/agronomy10091232

Córdova-Ávalos, V. (2009). Factores que afectan la producción de cacao (*Theobroma cacao* L.) en el ejido Francisco I. Madero del plan Chontalpa, Tabasco, México. *Universidad y Ciencia, 17*(34). Villahermosa, México: Universidad Juárez Autónoma de Tabasco. https://elibro.net/es/ereader/utiec/7965?page=6

Del Fabbro, C., Scalabrin, S., Morgante, M., & Giorgi, F. M. (2013). An extensive evaluation of read trimming effects on Illumina NGS data analysis. PLoS ONE, 8(12), e85024. https://doi.org/10.1371/journal.pone.0085024

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [Preprint]. arXiv. https://arxiv.org/abs/1810.04805

Dunning, M. (2018, January 21). Understanding sequencing reads: Introduction. Recuperado de https://bioinformatics-core-shared-training.github.io/cruk-summer-school-2017/Day1/Session4-seqIntro.html

Goodwin, S., McPherson, J. D., & McCombie, W. R. (2016). Coming of age: Ten years of next-generation sequencing technologies. Nature Reviews Genetics, 17(6), 333–351. https://doi.org/10.1038/nrg.2016.49

Gulati, A. (2024). Pattern recognition explained [Updated 2024]. KnowledgeHut. Recuperado de https://www.knowledgehut.com/blog/data-science/pattern-recognition

Gutiérrez, O.A., Campbell, A.S., Phillips-Mora, W. (2016). Breeding for Disease Resistance in Cacao. In: Bailey, B., Meinhardt, L. (eds) Cacao Diseases. Springer, Cham. https://doi.org/10.1007/978-3-319-24789-2_18

Hernández León, R. A. (2011). El proceso de investigación científica: (ed.). Editorial Universitaria. https://elibro.net/es/ereader/utiec/71435?page=64

Huq, A. K. M. M., Roney, M., Dubey, A., et al. (2024). Phenolic compounds of *Theobroma cacao* L. show potential against dengue RdRp protease enzyme inhibition by in-silico docking, DFT study, MD simulation and MMGBSA calculation. *PLoS ONE, 19*(3), 1–20. 10.1371/journal.pone.0299238

Illumina. (2023). Illumina Sequencing Technology. Recuperado de https://www.illumina.com/technology/next-generation-sequencing.html

Kretschmer, F., Mock, F., & Kriese, A. (2022). *BERTax: Taxonomic classification of DNA sequences* [Repositorio GitHub]. GitHub. https://github.com/rnajena/bertax

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4), 357–359. https://doi.org/10.1038/nmeth.1923

Lanaud, C., Vignes, H., Utge, J., et al. (2024). A revisited history of cacao domestication in pre-Columbian times revealed by archaeogenomic approaches. *Scientific Reports, 14*, 2972. https://doi.org/10.1038/s41598-024-53010-6

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Li, H., Tian, S., Li, Y., Fang, Q., Tan, R., Pan, Y., Huang, C., Xu, Y., & Gao, X. (2020). Modern deep learning in bioinformatics. Journal of Molecular Cell Biology, 12(11), 823–827. https://doi.org/10.1093/jmcb/mjaa030

Lindgreen, S. (2012). AdapterRemoval: Easy cleaning of next-generation sequencing reads. BMC Research Notes, 5(1), 337. https://doi.org/10.1186/1756-0500-5-337

MacManes, M. D. (2014). On the optimal trimming of high-throughput mRNA sequence data. Frontiers in Genetics, 5, 13. https://doi.org/10.3389/fgene.2014.00013

Mardis, E. R. (2017). DNA sequencing technologies: 2006–2016. Nature Protocols, 12(2), 213–218. https://doi.org/10.1038/nprot.2016.182

Martínez Ruiz, H. (2012). Metodología de la investigación: (ed.). Cengage Learning. https://elibro.net/es/ereader/utiec/39957?page=1

Mitchell, T. M. (1997). Machine learning. New York: McGraw-Hill.

Mock, F., Kriese, A., & Marz, M. (2022). Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks. Proceedings of the National Academy of Sciences, 119(6), e2122636119. https://doi.org/10.1073/pnas.2122636119

Montenegro, J., Morante, J., Acosta, M., et al. (2023). Molecular response of cocoa (*Theobroma cacao*) to water deficit conditions. *JAPS: Journal of Animal & Plant Sciences, 33*(6), 1314–1321. https://doi.org/10.36899/JAPS.2023.6.0671

Mullis, K., & Faloona, F. (1987). Specific synthesis of DNA in vitro via a polymerase-catalyzed chain reaction. Methods in Enzymology, 155, 335-350.

Muñoz O, J. M. (1948). *Estudios cromosómicos en el género Theobroma L.* [Tesis de maestría, Instituto Interamericano de Ciencias Agrícolas (IICA)]. Repositorio CATIE. https://repositorio.catie.ac.cr/handle/11554/1875

National Center for Biotechnology Information. (2025). *Theobroma cacao* genome assembly Criollo_cocoa_genome_V2. National Institutes of Health. https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000208745.1/

NVIDIA. (s.f.). *Deep learning*. NVIDIA Developer. https://developer.nvidia.com/deep-learning

Poplin, R., Chang, P. C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., … DePristo, M. A. (2018). A universal SNP and small-indel variant caller using deep neural networks. Nature Biotechnology, 36(10), 983–987. https://doi.org/10.1038/nbt.4235

Raschka, S. (2016). Model evaluation, model selection, and algorithm selection in machine learning. Recuperado de: https://sebastianraschka.com/blog/2016/modelevaluation-selection-part3.html

Ricaño-Rodríguez, J., Ramos-Prado, J. M., Cocoletzi-Vásquez, E., & Hipólito-Romero, E. (2018). *El estudio genómico del cacao (Theobroma cacao L.); breve recopilación de sus bases conceptuales*. Agroproductividad, 11(9), 29–35. https://www.revista-agroproductividad.org/index.php/agroproductividad/article/view/1211

Salazar Duque, D., & Espinoza Muñoz, D. (2022). Análisis de competitividad del destino turístico y el desarrollo de las rutas del cacao ecuatoriano. *Revista Turismo y Patrimonio, 18*, 95–112. https://doi.org/10.24265/turpatrim.2022.n18.05

Sangiovanni, M., Granata, I., Thind, A., & et al. (2019). From trash to treasure: detecting unexpected contamination in unmapped NGS data. BMC Bioinformatics, 20(Suppl 4), 168. https://doi.org/10.1186/s12859-019-2684-x

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. Neural Networks, 61, 85-117. doi:10.1016/j.neunet.2014.09.003

Schmieder, R., & Edwards, R. (2011). Fast identification and removal of sequence contamination from genomic and metagenomic datasets. PLoS ONE, 6(3), e17288. https://doi.org/10.1371/journal.pone.0017288

smoreira00. (2023, enero 23). ERROR: failed to run cuBLAS routine: CUBLAS_STATUS_EXECUTION_FAILED [Mensaje en un foro en línea]. *GitHub*. https://github.com/bmild/nerf/issues/174

Stoler, N., & Nekrutenko, A. (2021). *Sequencing error profiles of Illumina sequencing instruments*. *NAR Genomics and Bioinformatics*, 3(1), lqab019. https://doi.org/10.1093/nargab/lqab019

Tremblay, E. D., Yergeau, É., Fortin, N., & Greer, C. W. (2015). Preprocessing of high-throughput bacterial rRNA gene sequencing data. BMC Bioinformatics, 16, 37. https://doi.org/10.1186/s12859-015-0487-1

Wicke, S., Schneeweiss, G. M., dePamphilis, C. W., Müller, K. F., & Quandt, D. (2011). The evolution of the plastid chromosome in land plants: gene content, gene order, gene function. Plant molecular biology, 76(3-5), 273–297. https://doi.org/10.1007/s11103-011-9762-4

Winters, N. P., Wafula, E. K., Knollenberg, B. J., Hämälä, T., Timilsena, P. R., Perryman, M., Zhang, D., Sheaffer, L. L., Praul, C. A., Ralph, P. E., Prewitt, S., Leandro-Muñoz, M. E., Delgadillo-Duran, D. A., Altman, N. S., Tiffin, P., Maximova, S. N., dePamphilis, C. W., Marden, J. H., & Guiltinan, M. J. (2024). A combination of conserved and diverged responses underlies *Theobroma cacao*'s defense response to *Phytophthora palmivora*. *BMC Biology, 22*(1), 1–24. https://doi.org/10.1186/s12915-024-01831-2

Wood, D. E., Lu, J., & Langmead, B. (2019). Improved metagenomic analysis with Kraken 2. *Genome Biology*, 20(1), 1-13. https://doi.org/10.1186/s13059-019-1891-0

Wood, D. (2018, octubre 23). *xargs: cat: terminated by signal 13 when building library with kraken2* [Comentario en un issue]. GitHub. https://github.com/DerrickWood/kraken2/issues/58

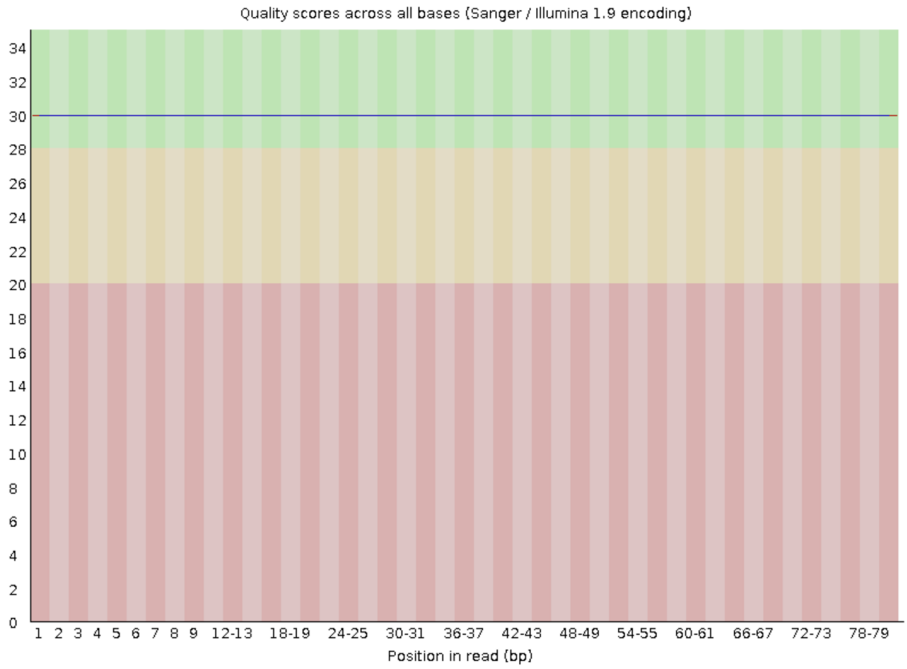
Zou, J., Huss, M., Abidin, F., & Mathee, K. (2019). A primer on deep learning in genomics. Nature Genetics, 51(1), 12-18. https://doi.org/10.1038/s41588-018-0295-5

9. Anexos

Anexo 1

Resultados de validación de datos SRR377719_1.fastq

Per base sequence quality



Anexo 2

Estadísticas resultantes de validación de datos SRR377719_1.fastq

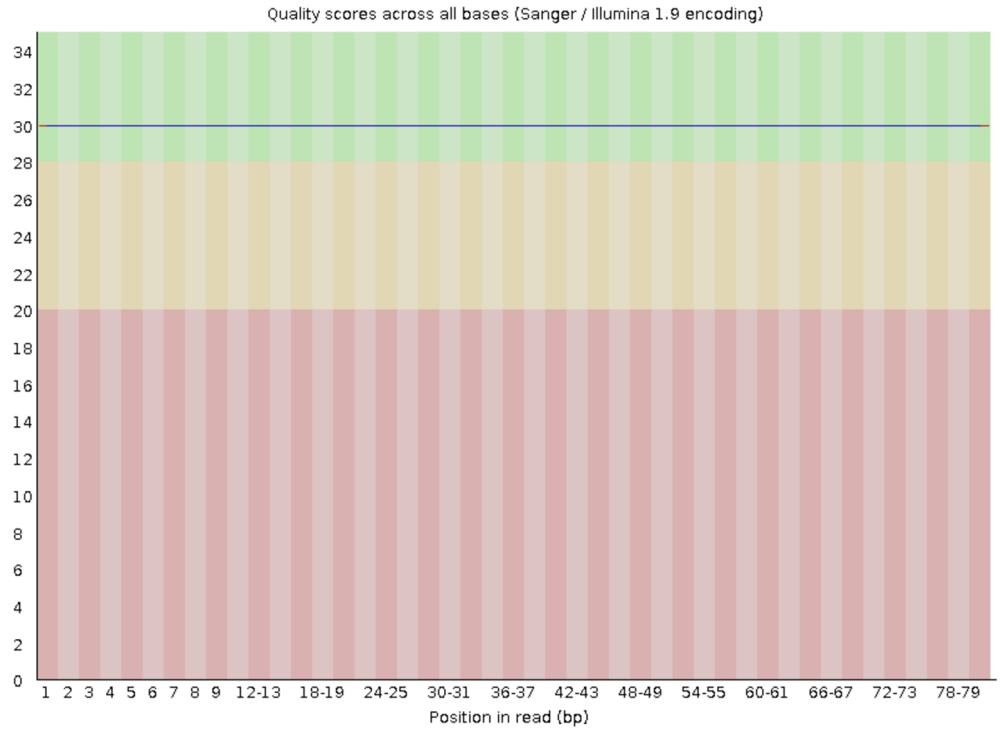
Basic Statistics

Measure	Value
Filename	SRR377719_1_fastq_gz.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	28794335
Total Bases	2.3 Gbp
Sequences flagged as poor quality	0
Sequence length	80
%GC	33

Anexo 3

Resultados de validación de datos SRR377719_2.fastq

Per base sequence quality



Anexo 4

Estadísticas resultantes de validación de datos SRR377719_2.fastq

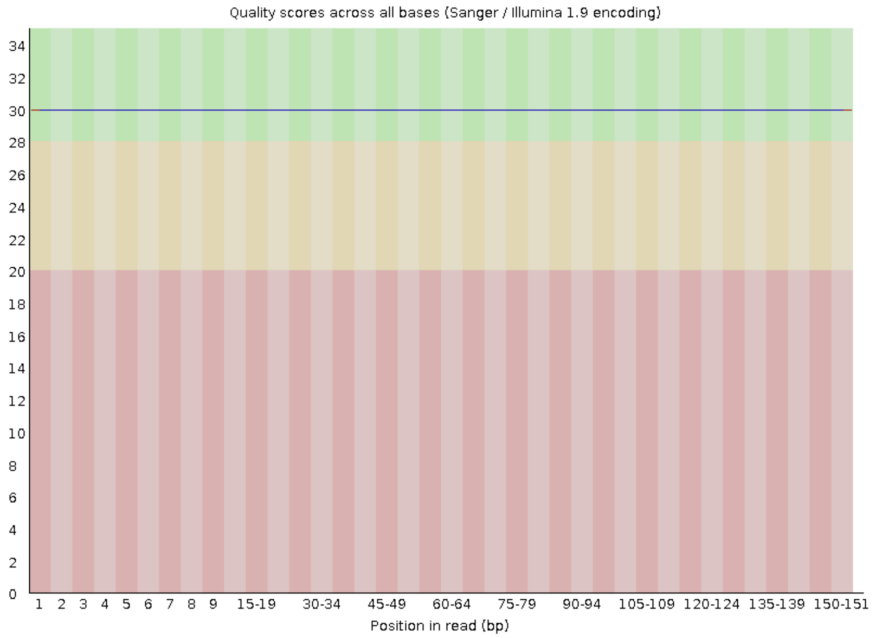
Basic Statistics

Measure	Value
Filename	SRR377719_2_fastq_gz.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	28794335
Total Bases	2.3 Gbp
Sequences flagged as poor quality	0
Sequence length	80
%GC	33

Anexo 5

Resultados de validación de datos SRR21562212_1.fastq

Per base sequence quality



Anexo 6

Estadísticas resultantes de validación de datos SRR21562212_1.fastq

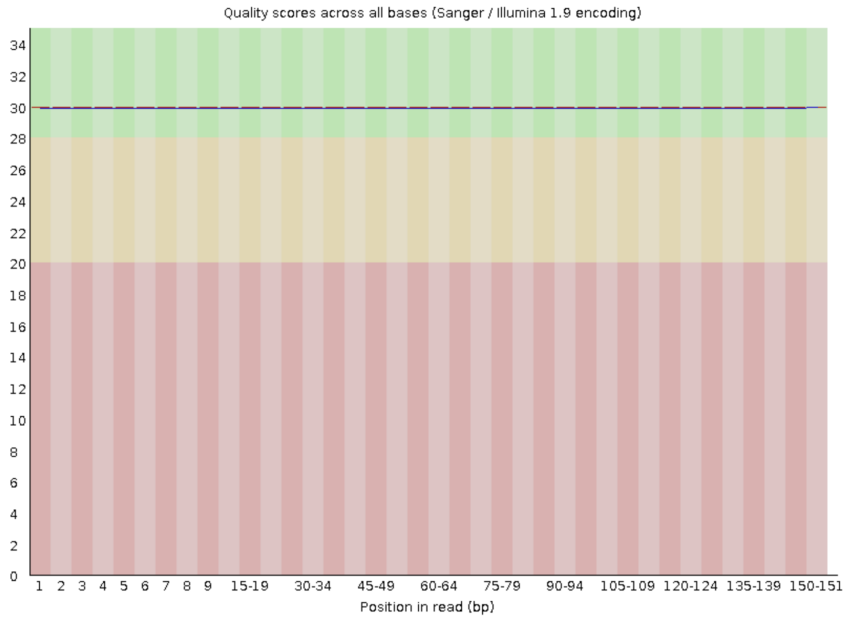
Basic Statistics

Measure	Value
Filename	SRR21562212_1_fastq_gz.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	72717892
Total Bases	10.9 Gbp
Sequences flagged as poor quality	0
Sequence length	30-151
%GC	36

Anexo 7

Resultados de validación de datos SRR21562212_2.fastq

Per base sequence quality



Anexo 8

Estadísticas resultantes de validación de datos SRR21562212_2.fastq

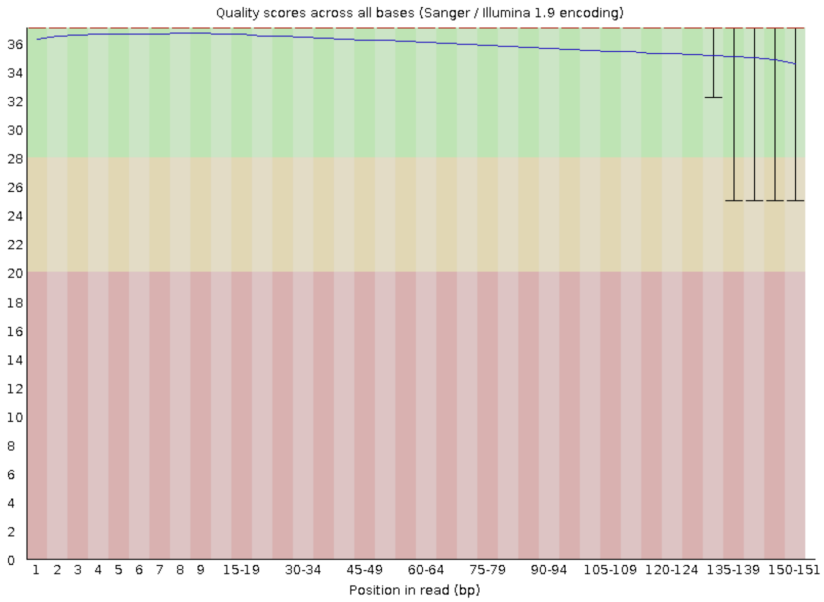
Basic Statistics

Measure	Value
Filename	SRR21562212_2_fastq_gz.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	72717892
Total Bases	10.9 Gbp
Sequences flagged as poor quality	0
Sequence length	30-151
%GC	36

Anexo 9

Resultados de validación de datos SRR14022547_1.fastq

Per base sequence quality



Anexo 10

Estadísticas resultantes de validación de datos SRR14022547_1.fastq

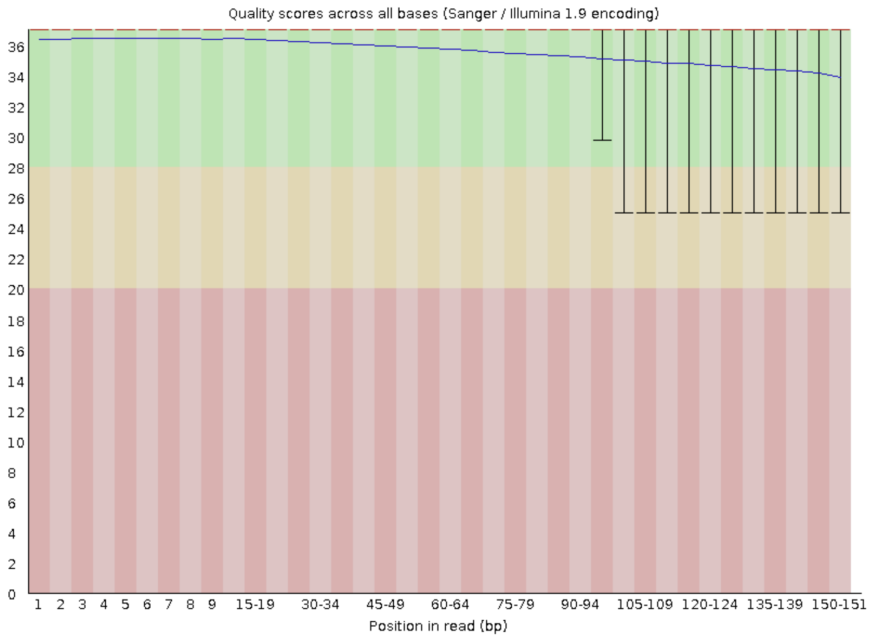
Basic Statistics

Measure	Value
Filename	SRR14022547_1_fastq_gz.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	211825883
Total Bases	31.9 Gbp
Sequences flagged as poor quality	0
Sequence length	151
%GC	38

Anexo 11

Resultados de validación de datos SRR14022547_2.fastq

Per base sequence quality



Anexo 12

Estadísticas resultantes de validación de datos SRR14022547_2.fastq

Basic Statistics

Measure	Value
Filename	SRR14022547_2_fastq_gz.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	211825883
Total Bases	31.9 Gbp
Sequences flagged as poor quality	0
Sequence length	151
%GC	36

10. Apéndices

Apéndice 1

Error en Clasificación de datos por falta de memoria RAM

```
anax@Ana:~/database$ ./kraken2-build --build --kmer-len 55 --threads 1 --db ana-db
Creating sequence ID to taxonomy ID map (step 1)...
Sequence ID to taxonomy ID map already present, skipping map creation.
Estimating required capacity (step 2)...
Estimated hash table requirement: 19979271312 bytes
Capacity estimation complete. [1h5m49.020s]
Building database files (step 3)...
Taxonomy parsed and converted.
xargs: cat: terminated by signal 13
/home/anax/database/build_kraken2_db.sh: line 143:  1973 Done                    list_sequence_files
      1974 Exit 125               | xargs -0 cat
      1975 Killed                 | build_db -k $KRAKEN2_KMER_LEN -l $KRAKEN2_MINIMIZER_LEN -S $KRAKEN2_SEED_TEMPLATE $
KRAKEN2XFLAG -H hash.k2d.tmp -t taxo.k2d.tmp -o opts.k2d.tmp -n taxonomy/ -m $seqid2taxid_map_file -c $required_capacity
 -p $KRAKEN2_THREAD_CT $max_db_flag -B $KRAKEN2_BLOCK_SIZE -b $KRAKEN2_SUBBLOCK_SIZE -r $KRAKEN2_MIN_TAXID_BITS $fast_bu
ild_flag
```

```
anax@Ana:~/database$ ./kraken2-build --build --kmer-len 55 --threads 10 --db ana-db
Creating sequence ID to taxonomy ID map (step 1)...
Sequence ID to taxonomy ID map already present, skipping map creation.
Estimating required capacity (step 2)...
Estimated hash table requirement: 19979271312 bytes
Capacity estimation complete. [18m47.217s]
Building database files (step 3)...
Taxonomy parsed and converted.
xargs: cat: terminated by signal 13
/home/anax/database/build_kraken2_db.sh: line 143:  1903 Done                    list_sequence_files
      1904 Exit 125               | xargs -0 cat
      1905 Killed                 | build_db -k $KRAKEN2_KMER_LEN -l $KRAKEN2_MINIMIZER_LEN -S $KRAKEN2_SEED_TEMPLATE $
KRAKEN2XFLAG -H hash.k2d.tmp -t taxo.k2d.tmp -o opts.k2d.tmp -n taxonomy/ -m $seqid2taxid_map_file -c $required_capacity
 -p $KRAKEN2_THREAD_CT $max_db_flag -B $KRAKEN2_BLOCK_SIZE -b $KRAKEN2_SUBBLOCK_SIZE -r $KRAKEN2_MIN_TAXID_BITS $fast_bu
ild_flag
```

Apéndice 2

Recursos computacionales usados para la obtención de secuencias «clasificadas» con *Kraken2.*

Job Metrics

cgroup

CPU usage time	3 hours and 23 minutes
CPU user time	3 hours and 2 minutes
CPU system time	20 minutes
Number of processes belonging to this cgroup killed by any kind of OOM killer	0
Max memory usage recorded	289.9 GB

core

Container ID	/cvmfs/singularity.galaxyproject.org/all/kraken2:2.1.3--pl5321h077b44d_4
Container Type	singularity
Cores Allocated	2
Memory Allocated (MB)	71680
Job Start Time	2025-03-02 00:55:25
Job End Time	2025-03-02 03:02:48
Job Runtime (Wall Clock)	2 hours and 7 minutes

hostname

hostname	vgcnbwc-worker-c36m975-0002.novalocal

Apéndice 3

Recursos computacionales usados para la obtención de secuencias «no clasificadas» con *Kraken2* y base de datos pluspf.

Job Metrics

cgroup

CPU usage time	2 hours and 54 minutes
CPU user time	2 hours and 35 minutes
CPU system time	18 minutes
Number of processes belonging to this cgroup killed by any kind of OOM killer	0
Max memory usage recorded	151.6 GB

core

Container ID	/cvmfs/singularity.galaxyproject.org/all/kraken2:2.1.3--pl5321h077b44d_4
Container Type	singularity
Cores Allocated	2
Memory Allocated (MB)	71680
Job Start Time	2025-03-02 23:14:42
Job End Time	2025-03-03 01:00:44
Job Runtime (Wall Clock)	1 hour and 46 minutes

hostname

hostname	vgcnbwc-worker-c36m975-0003.novalocal

Apéndice 4

Recursos computacionales usados para la obtención de secuencias «no clasificadas» con *Kraken2* y base de datos SILVA.

Job Metrics

cgroup

CPU usage time	1 hour and 10 minutes
CPU user time	1 hour and 0 minutes
CPU system time	9 minutes
Number of processes belonging to this cgroup killed by any kind of OOM killer	0
Max memory usage recorded	70.0 GB

core

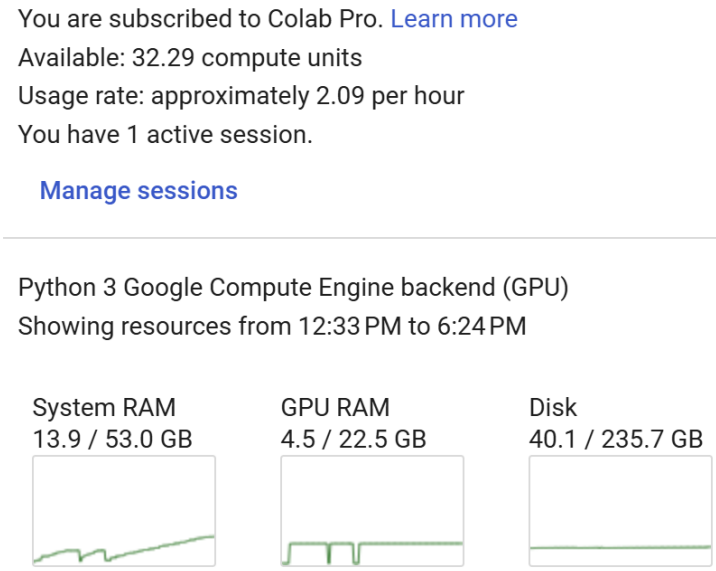
Container ID	/cvmfs/singularity.galaxyproject.org/all/kraken2:2.1.3--pl5321h077b44d_4
Container Type	singularity
Cores Allocated	2
Memory Allocated (MB)	71680
Job Start Time	2025-03-03 01:57:20
Job End Time	2025-03-03 02:52:00
Job Runtime (Wall Clock)	54 minutes

hostname

hostname	vgcnbwc-worker-c36m900-0000.novalocal

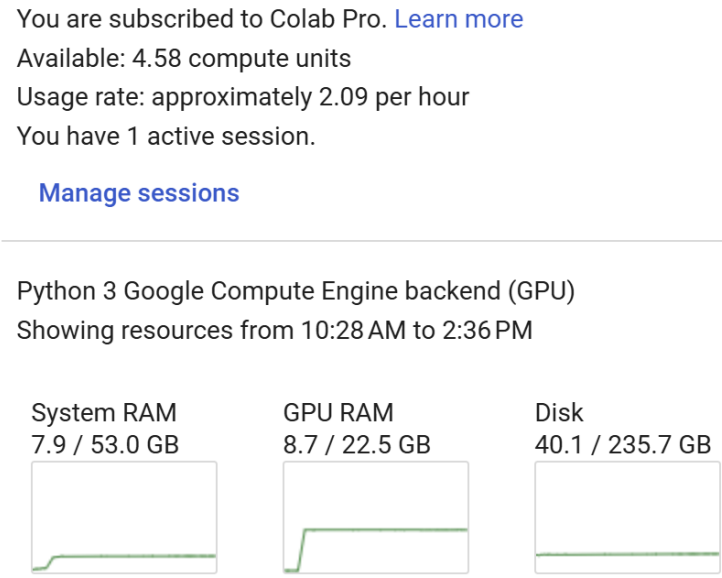
Apéndice 5

Recursos computacionales usados en fase de ajuste de hiperparámetros.



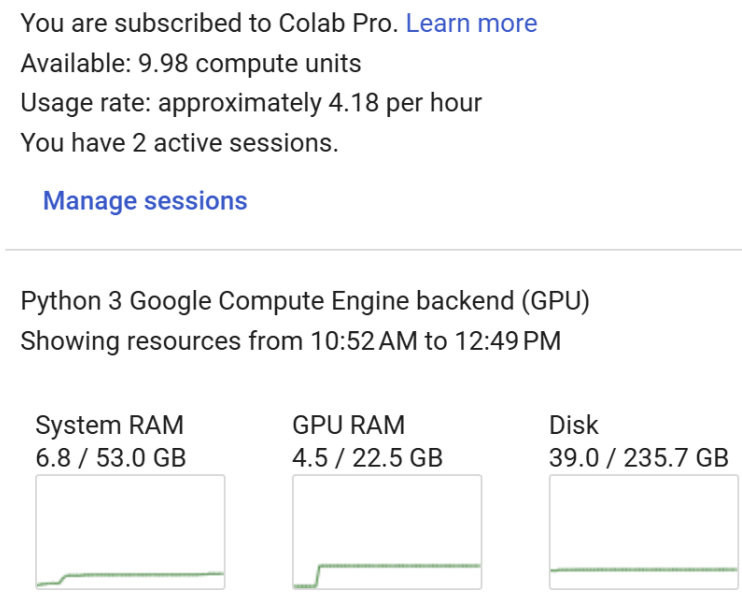
Apéndice 6

Recursos computacionales usados en fase de pre-entrenamiento del 72% de los datos.



Apéndice 7

Recursos computacionales usados en fase de pre-entrenamiento del 72% de los datos.



Apéndice 8

Recursos computacionales usados en fase de validación con el 20% de los datos.