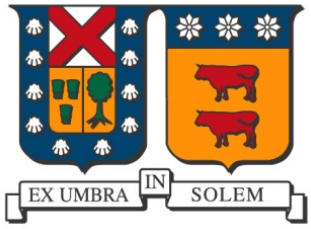
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE TELEMÁTICA
VALPARAÍSO - CHILE



“CARACTERIZACIÓN DE MICROORGANISMOS DE INTERÉS BIOTECNOLÓGICO MEDIANTE EL USO DE DATOS GENÓMICOS Y RECONOCIMIENTO DE PATRONES”

FABIÁN GUERRERO MAUREIRA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN TELEMÁTICA

Profesor Guía: Mauricio Araya
Profesor Correferente: Nicolas Jara Carvallo

Diciembre - 2021

DEDICATORIA

Con gratitud hacia aquellos que me brindaron su apoyo inquebrantable y con dedicación incansable, este trabajo es un tributo a la perseverancia y el esfuerzo. A todos aquellos que creyeron en mí y compartieron este viaje, mi mas sincera gratitud.

AGRADECIMIENTOS

”La gratitud es la memoria del corazón.” - Jean Baptiste Massieu

A continuación mencionaré a las personas que más impactaron en el desarrollo de este trabajo, a cada una de ellas les agradezco profundamente.

Para mi familia, quienes desde que entré a la Universidad me hicieron saber lo orgullosos que se sentían de mi. Gracias por el constante apoyo en especial a mi mamá, mi abuela y mis hermanos, quienes son de las personas más importantes de mi vida.

Para mi segunda familia, los que me instaron a tomar el camino de la ingeniería porque vieron en mí el potencial que ni yo sabía que tenía. Gracias a todos los Ampuero!

Para Roberto y Jesus, quienes sin obligación de ayudarme estuvieron hasta el final usando de su tiempo para que este informe quedara lo mejor posible.

Por último y más importante, para mi querida esposa Katherine, mi compañera desde los 14 años, la persona que me hace ver que todo mi esfuerzo vale la pena y una de las grandes responsables de que sea la persona que soy hoy en día.

RESUMEN

Resumen— La caracterización de microorganismos y enzimas mediante herramientas bioinformáticas ha acortado la brecha entre la ciencia y el desarrollo tecnológico. Sin embargo, aun existen problemáticas asociadas a la calidad de los datasets para su utilización en herramientas de aprendizaje automático. La presente memoria describe el proyecto de desarrollo de un clasificador binario para identificar enzimas degradadoras de contaminantes aromáticos en secuencias genómicas. Se emplean técnicas de aprendizaje automático como SVM con kernel RBF, KNeighbors y Random Forest. Se elaboran seis datasets con diferentes características de balance y longitud de secuencia, sobre los cuales se entrenan y prueban los modelos. A pesar de los buenos resultados iniciales como curvas ROC, accuracy, F1-score y AUC, la aplicación en genomas de *Escherichia coli* y *Paraburkholderia xenovorans LB400* revela una alta incidencia de falsos positivos, lo que indica la necesidad de mejorar la representatividad de los datasets y la metodología de clasificación. Esta memoria resalta la importancia de la validación robusta y la potencial aplicación de aprendizaje profundo para futuras investigaciones.

Palabras Clave— Clasificador Binario, Grupo COG (Clusters of Orthologous Groups), Homología, Oxigenasa, Falsos Positivos, ML, Random Forest, SVM, KNeighbors, FASTA

ABSTRACT

Abstract— The characterization of microorganisms and enzymes using bioinformatics tools has bridged the gap between science and technological development. However, there are still challenges associated with the quality of datasets for their use in machine learning tools. The present work describes the project of developing a binary classifier to identify enzymes degrading aromatic contaminants in genomic sequences. Machine learning techniques such as SVM with RBF kernel, KNeighbors, and Random Forest are employed. Six datasets with different balance and sequence length characteristics are created, on which the models are trained and tested. Despite the promising initial results, such as ROC curves, accuracy, F1-score, and AUC, the application to genomes of *Escherichia coli* and *Paraburkholderia xenovorans LB400* reveals a high incidence of false positives, indicating the need to improve dataset representativeness and classification methodology. This work highlights the importance of robust validation and the potential application of deep learning for future research.

Keywords— Binary Classifier, COG Group(Clusters of Orthologous Groups), Homology, Oxygenase, False Positives, ML, Random Forest, SVM, KNeighbors, FASTA

GLOSARIO

Clasificador Binario: Un modelo de aprendizaje automático que predice uno de dos posibles resultados para una entrada dada, utilizado en contextos donde las respuestas son dicotómicas, como 'sí' o 'no'.

Grupo COG (Clusters of Orthologous Groups): Un sistema que clasifica proteínas de genomas completos en grupos homólogos, lo que puede reflejar la funcionalidad de una enzima o su participación en ciertos procesos biológicos.

Homología: La existencia de un ancestro común entre un par de estructuras o genes en diferentes especies, a menudo determinada mediante alineación de secuencias y utilizada para predecir similitudes funcionales.

Oxigenasa: Una enzima que cataliza la incorporación de oxígeno del oxígeno molecular (O2) en sustratos orgánicos, un proceso crítico para la degradación de compuestos aromáticos.

Falsos Positivos: En el contexto de la clasificación, son identificaciones incorrectas donde un resultado no objetivo es clasificado incorrectamente como objetivo, como una enzima no degradadora siendo mal clasificada como degradadora.

ML (Machine Learning): Machine Learning es una rama de la inteligencia artificial que permite a las máquinas aprender de datos y mejorar su desempeño en tareas sin programación explícita.

Random Forest: Random Forest es un algoritmo de aprendizaje automático que combina múltiples árboles de decisión para tomar decisiones más precisas.

SVM (Support Vector Machine): SVM es un algoritmo de aprendizaje supervisado que busca encontrar un hiperplano óptimo para separar datos en dos categorías.

KNeighbors: KNeighbors es un algoritmo que clasifica o predice valores basándose en la mayoría de los k puntos de datos más cercanos en un conjunto de entrenamiento.

FASTA: FASTA es un formato de archivo utilizado en bioinformática para representar secuencias de ADN, ARN y proteínas, con un formato que facilita el análisis de datos biológicos.

ÍNDICE DE CONTENIDOS

RESUMEN iv

ABSTRACT iv

GLOSARIO v

ÍNDICE DE FIGURAS viii

ÍNDICE DE TABLAS ix

INTRODUCCIÓN 1

CAPÍTULO 1: MARCO TEÓRICO Y ESTADO DEL ARTE 3
1.1 ¿Qué es la biotecnología? . . . 3
1.1.1 Tipos de biotecnología . . . 3
1.2 ¿Qué es el genoma? . . . 3
1.3 ¿Qué es una secuencia? . . . 4
1.4 ¿Qué es la homología? . . . 4
1.5 ¿Cómo representar las secuencias? . . . 5
1.5.1 El formato FASTA . . . 5
1.5.2 El formato GBK . . . 5
1.6 Alineamiento de secuencias . . . 6
1.6.1 ¿Qué es un alineamiento? . . . 6
1.6.2 Alineamientos de pares . . . 7
1.7 BLAST: un software basado alineamiento de secuencias . . . 9
1.7.1 Cómo funciona BLAST . . . 10
1.7.2 La heurística de BLAST . . . 11
1.7.3 Ventajas y desventajas de su uso . . . 12
1.8 Herramientas de identificación automáticas utilizando genomas completos . 12
1.8.1 AntiSmash . . . 13
1.9 Herramientas basadas en *Machine learning* en identificación de homólogos . 14
1.9.1 DeepNOG . . . 14
1.9.2 Cómo funciona DeepNog . . . 14
1.10 Enzimas degradadoras . . . 16
1.10.1 Oxigenasas Hidroxilantes de Anillo . . . 16
1.11 Bases de datos . . . 17
1.11.1 KEGG . . . 17
1.11.2 RHOBase . . . 17
1.11.3 AromaDeg . . . 19
1.11.4 UniProt . . . 20

CAPÍTULO 2: Identificación del Problema y Propuestas de Mejora 21
2.1 Comprensión del problema y análisis de impacto . . . 21

2.2 Propuesta de mejora: ADEC 22
2.3 Implementación de la solución 22
2.3.1 Creación de datasets 24
2.3.2 Dataset basado en AromaDEG 24
2.3.3 Dataset basado en KEGG 27
2.3.4 Dataset basado en RHOBase para Clasificador de Función Enzimática 33
2.3.5 Creación de dataset para Clasificador Binario 36
2.3.6 Embedding: Vectorización de la secuencia 39
2.3.7 Clasificador Binario 40

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN **50**
3.1 *Escherichia coli*, modelo general de fisiología bacteriana 50
3.2 *Paraburkholderia xenovorans* LB400, modelo de degradación de compuestos aromáticos 51

CAPÍTULO 4: CONCLUSIONES **53**

REFERENCIAS BIBLIOGRÁFICAS **54**

ÍNDICE DE FIGURAS

1 [Fuente: Richardson Silva Lima [7], *O formato FASTA*. Accedido el 2022-04-01.] Archivo en formato FASTA: A cada linea que comienza con un símbolo ``>'' le siguen los datos de la enzima que representa, como el identificador, el nombre del organismo, el gen al que pertenece, etc. Luego de esta linea siempre le sigue la secuencia asociada a la enzima en cuestión. Esta seguidilla de caracteres se mantendrá hasta que se encuentre una nueva linea con un símbolo ``>'' en donde comenzará la información de una nueva secuencia. . . 5

2 [Elaboración propia, 2021.] Perfil Oculto de Markov. 9

3 [Fuente: antiSMASH [8], Antismash pipeline. Accedido el 2021-09-01.] Pipeline del análisis de metabolitos secundarios. 14

4 Fuente: DeepNOG[13], Accedido el 2021-09-01] Arquitectura de la red DeepNOG. 15

5 [Elaboración propia, 2021.] Diagrama de interacción entre los componentes de una RHO. [21] 16

6 [Elaboración propia, 2021.] Diagrama de la arquitectura de ADEC. En rojo los módulos en los que se trabaja a lo largo de esta memoria. 23

7 Diagrama de la arquitectura del módulo de clasificación. En rojo los módulos que se trabajan a lo largo de esta memoria. 24

8 [Elaboración propia, 2024.] Diagrama de flujo de script para agregar enzimas desde un archivo en formato FASTA a un archivo .csv. 25

9 [Elaboración propia, 2024.] Uso de la herramienta DeepNOG para predecir el grupo COG de una Extradiol dioxygenasa. Notar que se genera un archivo de salida llamado "prediction.csv". 26

10 [Elaboración propia, 2024.] Diagrama de flujo para crear un archivo FASTA a partir de un .csv. 28

11 [Elaboración propia, 2024.] Diagrama de flujo para filtrar y separar los COG de KEGG. 29

12 [Elaboración propia, 2024.] Panel de configuración para la alineación de BLASTP. 30

13 [Elaboración propia, 2024.] Carga de las secuencias a analizar en BLASTP. . . 31

14 [Elaboración propia, 2024.] Panel de selección de bases de datos para comparar entradas. 31

15 [Elaboración propia, 2024.] Panel de selección de programa. 31

16 [Elaboración propia, 2024.] Resultados de la alineación de BLAST en su sección de información sobre la secuencia alineada. 32

17 [Elaboración propia, 2024.] Resultados de la alineación de BLAST en su sección de información sobre la secuencia alineada. 32

18 [Elaboración propia, 2024.] Diagrama paso a paso del proceso de creación del dataset basado en KEGG. 33

19 [Elaboración propia, 2024.] Diagrama de flujo para separar archivos el dataset RHOBase en .fasta. 34

20 [Elaboración propia, 2024.] Filtros para datos encontrados en BLASTP de RHO. 35

21 [Elaboración propia, 2024.] Resultados de una vectorización, cada fila representa una secuencia de una bacteria y cada columna corresponde al embedding de la misma. 40
22 Resultados al evaluar los distintos modelos usando el dataset con el criterio de largo entre 350 y 460 caracteres 44

ÍNDICE DE TABLAS

1 Contenido e identificadores de la base de datos de KEGG. [19] 18
2 Clasificaciones de DeepNOG para la base de datos AromaDEG. 27
3 Resumen de resultados para bacteria Escherichia coli. 51

INTRODUCCIÓN

Los microorganismos están involucrados ampliamente en procesos de transformación que pueden ser de utilidad industrial o beneficiosos para las personas y el medio ambiente. Su uso data desde hace miles de años en la fermentación de alimentos como el queso, vino y cerveza, sin embargo, el creciente desarrollo de la biotecnología ha permitido ampliar las aplicaciones de su uso a industrias complejas como la farmacéutica, cosmética y química, permitiendo incluso el tratamiento para la eliminación de desechos orgánicos y degradación de contaminantes en el medio ambiente.

Cientos de bacterias y hongos son aislados año a año para buscar nuevas aplicaciones, pero la estandarización de muchos de sus parámetros fisiológicos como crecimiento, descubrimiento de capacidad metabólica y resistencia a estresores ambientales, son lentos y requieren un profundo trabajo experimental, lo que ha limitado el avance en esta materia. La secuenciación de ADN permitió acelerar la caracterización, pero aún se requieren herramientas para automatizar este proceso.

La secuenciación de ADN es un conjunto de métodos que permiten determinar el orden de los nucleótidos que forman la macromolécula. De esta manera, contar con la secuencia de nucleótidos es posible determinar los genes de un segmento específico de ADN, así como también las posible proteínas que codifican y sus funciones celulares asociadas, incluyendo en términos más generales la herencia en la propensión a enfermedades, la respuesta a las influencias ambientales o en la predicción de mecanismos moleculares específicos.

Para determinar la función de las proteínas de un microorganismo, se realiza una comparación genómica y/o de secuencias de amino ácidos respecto a organismos ya secuenciados y con caracterización experimental. Un criterio importante es la homología, concepto que permite confirmar la relación evolutiva entre dos secuencias nucleotídicas o aminoacídicas mediante comparación de su secuencia. Al obtener una alta similaridad entre las secuencias u observar conservación de segmentos relevantes para su función, se confirman que son homólogas, que poseen un mismo origen evolutivo y posiblemente ambas proteínas pueden tener la misma función.

La búsqueda de homólogos se realiza de forma convencional mediante el alineamiento de secuencias, procedimiento que utiliza distintos tipos de algoritmos que identifican patrones, deleciones o mutaciones en ellas y entregan porcentajes de identidad, junto con las secciones de las secuencias asociadas al análisis. Esta información debe ser corroborada por el investigador para asegurar la homología, incluyendo información importante asociada a cada secuencia como dominios o residuos proteícos importantes para la función, de modo

que este método aún se encuentra en mejora constante debido a que no se ha encontrado una herramienta óptima en términos de precisión, tiempo de procesamiento de información y confiabilidad de los resultados.

El grupo de Bioinformática UTFSM-Valparaíso posee un Proyecto Interno Multidisciplinario, ”Aplicación de aprendizaje de máquinas y minería de texto para el análisis de datos genómicos y desarrollo de software bioinformático” financiado por la Dirección General de Investigación, Innovación y Emprendimiento UTFSM, el cual busca desarrollar un software o herramienta que pueda facilitar la caracterización de aislados bacterianos utilizando minería de datos o aprendizaje de máquinas, entregando respuestas automáticas que no requieran un posterior análisis experto. Esta memoria significaría un salto de innovación en investigaciones que analicen la información genética de microorganismos, lo que permitiría facilitar su estudio y su uso tanto industrial como científico. Así es como nace ADEC, un clasificador de enzimas degradadoras de compuestos aromáticos (Aromatic compound Degradation Enzyme Classifier) que logra hacer una caracterización rápida y eficaz de los microorganismos de interés, pero para desarrollar esta herramienta completamente hace falta llevar a cabo bastante trabajo en cuanto a la información que se maneja y como se presenta, es por esto que a lo largo de este trabajo se sentarán las bases para lograr desarrollar un software que ayude a la caracterización de microorganismos en base a su información genética.

CAPÍTULO 1
MARCO TEÓRICO Y ESTADO DEL ARTE

1.1. ¿Qué es la biotecnología?

La Biotecnología se define como un área multidisciplinaria, que emplea la biología, química y procesos varios, con gran uso en agricultura, farmacia, ciencia de los alimentos, ciencias forestales y medicina. Probablemente, el primero que usó este término fue el ingeniero húngaro Karl Ereky, en 1919.

Una definición de biotecnología aceptada internacionalmente es la siguiente *"La biotecnología se refiere a toda aplicación tecnológica que utilice sistemas biológicos y organismos vivos o sus derivados para la creación o modificación de productos o procesos para usos específicos."* (Secretaría del Convenio de Diversidad Biológica, 1992). [1]

1.1.1. Tipos de biotecnología

- **Médica:** Es aquella que emplea el uso de células vivas y otros elementos celulares, para lograr mejoras en la salud de las personas.
- **Agrícola:** Es aquella que se emplea para la mejora de cultivos y plantas. En general se centra en el desarrollo e investigación de las plantas genéticamente modificadas, con la finalidad de mejorar sus características, haciéndolas resistentes a otras plantas, el clima, plagas, entre otras.
- **Industrial:** Es aquella que contribuye a la creación de elementos industriales basados en la microbiología, o bien, al reemplazo de otros, por unos menos contaminantes.

1.2. ¿Qué es el genoma?

Es el conjunto de genes organizados en cromosomas, lo que puede interpretarse como la totalidad del material genético que posee un organismo o una especie en particular. El genoma en los organismos eucariontes comprende el ADN contenido en el núcleo, organizado en cromosomas y el genoma de orgánulos celulares, como las mitocondrias y los plastos. Mientras que en las bacterias y arqueas toda la información genética esencial está contenida en una única molécula de ácido desoxirribonucleico (ADN) de doble cadena, generalmente circular y cerrado. Dicha molécula se denomina cromosoma bacteriano [6].

Muchas bacterias poseen además ADN extracromosómico llamado plásmido, generalemente circular y cerrado, denominado ADN plasmídico o plasmidial por estar contenido en los plásmidos. Éstos, portan información génica para muchas funciones que no son esenciales para la célula en condiciones normales de crecimiento. [6]

El término genoma fue acuñado en 1920 por Hans Winkler, profesor de Botánica en la Universidad de Hamburgo, Alemania, como un acrónimo de las palabras *gene* y *cromosoma*.

1.3. ¿Qué es una secuencia?

Es una línea de caracteres que tiene una dirección (3' a 5' o 5' a 3'), un orden, una composición, un largo y cuyos caracteres representan una unidad básica. Esta puede representar un gen, un transcrito, un fragmento de un gen o transcrito, una estructura o una función. Ejemplos de esto son:

- ADN.
- ARN, incluyendo a los ARN mensajeros (ARNm).
- Proteínas, incluyendo enzimas.

Para el caso del ADN, ARN y ARNm, las unidades básicas son los nucleótidos. Cada uno de ellos está formado a su vez por una base nitrogenada, un azúcar (la desoxirribosa) y un grupo fosfato, donde los carácteres que se anotan en la secuencia representan las bases nitrogenadas, que para el ADN son adenina (A), timina (T), citosina (C) y guanina (G); y para el ácido ribonucleico (ARN) son en lugar de timina, el uracilo (U). Así, una cadena o hebra de ácido nucleico, tendrá una estructura primaria determinada por la secuencia de las bases que la componen.

Por otro lado, en el caso de proteínas y enzimas, las unidades básicas son los aminoácidos: un conjunto de 20 tipos de moléculas que a su vez se agrupan en cadenas llamadas polipéptidos. La secuencia de la cadena de aminoácidos determinará cómo se pliega tridimensionalmente el polipéptido, pues la forma que adquiera es muy importante para que sea biológicamente activo. De forma general, la secuencia de aminoácidos que forma una proteína está codificada en un gen.

1.4. ¿Qué es la homología?

La homología es la relación evolutiva que existe entre dos secuencias de dos organismos distintos, implicando que ambos determinantes genéticos tienen el mismo origen evolutivo.

De modo que esta es una cualidad, es decir, una proteína es o no es homóloga con otra, no existiendo el concepto de homología parcial. Una manera de identificar la homología es el alineamiento de secuencias. Proteínas o genes homólogos poseen un mismo origen ancestral y, en general, tienen una estructura y función similar, sin embargo, esto no implica que deben tener exactamente la misma función.

1.5. ¿Cómo representar las secuencias?

1.5.1. El formato FASTA

El formato FASTA es la forma más común de representar computacionalmente una secuencia de aminoácidos para proteínas y de nucleótidos para ADN y ARN. Este tipo de archivos hace uso de una serie de caracteres en donde cada uno tiene como significado una unidad básica (aminoácido y nucleótido según corresponda). Estos ficheros siguen un estándar bien definido para almacenar secuencias (i.e. genomas completos, proteínas) y así, evitar complejidad a la hora de procesar el texto. A continuación, se puede apreciar la estructura de un archivo FASTA.

```
>FOSB_MOUSE Protein fosB. 338 bp
MFQAFPGDYDSGSRCSSSPSAESQYLSSVDSFGSPPTAAASQECAGLGEMPGSFVPTVTA
ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGTSYSTPGLSAYSTGGASGS
GGPSTSTTTSGPVSARPARARPRRPREETLTPEEEEKRRVRRERNKLAAAKCRNRRRELT
DRLQAETDQLEEEKAELESEIAELQKEKERLEFVLVAHKPGCKIPYEEGPGPGPLAEVRD
LPGSTSAKEDGFGWLLPPPPPPPLPFQSSRDAPPNLTASLFTHSEVQVLGDPFPVVSPSY
TSSFVLTCPEVSAFAGAQRTSGSEQPSDPLNSPSLLAL
```

Figura 1: [Fuente: Richardson Silva Lima [7], *O formato FASTA*. Accedido el 2022-04-01.] Archivo en formato FASTA: A cada linea que comienza con un símbolo “>” le siguen los datos de la enzima que representa, como el identificador, el nombre del organismo, el gen al que pertenece, etc. Luego de esta linea siempre le sigue la secuencia asociada a la enzima en cuestión. Esta seguidilla de caracteres se mantendrá hasta que se encuentre una nueva linea con un símbolo “>” en donde comenzará la información de una nueva secuencia.

1.5.2. El formato GBK

Este tipo de formato difiere al mencionado anteriormente, ya que GBK se refiere a un archivo de base de datos genéticos de secuencias asociado a GenBanK [9]. Este formato no solo

guarda identificadores y las secuencias de un genoma, sino que contiene información certera referente a la anotación del genoma, las enzimas y proteínas propiamente tal, como a la información del autor y la publicación que respalda la existencia de este microorganismos y características varias.[1]

1.6. Alineamiento de secuencias

1.6.1. ¿Qué es un alineamiento?

Un alineamiento de secuencias en bioinformática es una forma de representar y comparar dos o más secuencias o cadenas de ADN, ARN, o estructuras primarias proteicas para resaltar sus zonas de similitud, que podrían indicar relaciones funcionales o evolutivas entre los genes o proteínas consultados. Las secuencias alineadas se escriben con las letras (representando aminoácidos o nucleótidos) en filas de una matriz en las que, si es necesario, se insertan espacios para que las zonas con idéntica o similar estructura se alineen.

Los alineamientos son útiles para:

- Asegurar que dos secuencias son similares y cuantificar su similitud.
- Encontrar dominios funcionales.
- Comparar un gen y su producto.
- Identificar homología entre las secuencias.

Tipos de Alineamientos

Locales o bloque: Se buscan porciones de la secuencia con el mayor grado de similitud posible. Los método de búsquedas proporcionan una o más regiones de alta similitud.

Globales: En este caso se comparan las secuencias completas, usando la totalidad de bases o aminoácidos que sean posibles, además se incluyen ambos extremos de la secuencias.

Estos alineamientos se recomiendan en los siguientes casos:

[1] Para información más detallada sobre los archivos .gbk puede consultar en `https://www.ncbi.nlm.nih.gov/genbank/samplerecord/`

- Se pretende realizar comparaciones en las que se quiera identificar homología de la secuencia.
- Se comparan secuencias que son muy similares entre sí.
- Las secuencias tienen aproximadamente la misma longitud.

1.6.2. Alineamientos de pares

Esta técnica es utilizada para encontrar la mejor coincidencia en bloque o alineamiento global de dos secuencias. El alineamiento solo puede realizarse con dos entradas a la vez, pero son eficientes de calcular y son utilizados a menudo en métodos que no requieren precisión extrema, como la búsqueda en bases de datos de secuencias con alta homología de secuencia con respecto a una petición.

Los tres métodos principales de generar alineamientos de pares son *matriz de puntos*, *programación dinámica y búsqueda de palabra corta*.

Método de matriz de puntos Es un método gráfico de recurrencia para comparar dos secuencias biológicas e identificar regiones de estrecha similitud tras la alineación de ambas.

Los gráficos de puntos comparan las secuencias organizando una cadena en el eje x, y otra en el eje y, de un gráfico. Cuando los residuos de ambas secuencias coinciden en el mismo lugar del gráfico, se dibuja un punto en la posición correspondiente.

Programación dinámica

1. **Algoritmo de Needleman-Wunsch**
   Sirve para realizar alineamientos globales de dos secuencias. Fue propuesto por primera vez en 1970, por Saul Needleman y Christian Wunsch. Se trata de un ejemplo típico de programación dinámica. El algoritmo funciona del mismo modo independientemente de la complejidad o longitud de las secuencias y garantiza la obtención del mejor alineamiento[10].
2. **Algoritmo de Smith-Waterman**
   El algoritmo SW fue propuesto por Temple Smith y Michael Waterman en 1981. Está basado en el uso de algoritmos de programación dinámica, de tal forma que tiene la deseable propiedad de garantizar que el alineamiento local encontrado es óptimo con respecto a uno de los posibles sistemas de puntajes que a utilizar[11].
3. **Búsqueda de palabra corta**
   Los métodos de palabra corta, también conocidos como métodos de k-tuplas (*k-mers*),

son soluciones heurísticas que no garantizan encontrar un resultado de alineamiento óptimo, pero son significativamente más eficientes que la programación dinámica. Estas técnicas son especialmente útiles en búsquedas sobre bases de datos a gran escala, ya que se asume que una larga proporción de las secuencias candidatas no tendrán coincidencias significativas con la aquella que fue ingresada en primera instancia[12].

Estos métodos identifican en la entrada una serie de subsecuencias cortas que no se solapan, estas denominadas ”palabras”, las cuales se contrastan con las secuencias de la base de datos.

Alineamientos múltiples El alineamiento múltiple de secuencias es una extensión del alineamiento de pares, este incorpora más de dos secuencias al mismo tiempo. Los métodos de este tipo de alineamiento son usados a menudos en la identificación de regiones conservadas en un grupo de secuencias que hipotéticamente están relacionadas evolutivamente.

Estos son computacionalmente difíciles de producir y la mayoría de las formulaciones del planteamiento conducen a problemas de optimización combinatorial NP-completos[2]. Sin embargo, la utilidad de estos alineamientos en la bioinformática ha dado lugar al desarrollo de una variedad de métodos adecuados para su realización.

Métodos para alineamientos múltiples

1. **Construcción progresiva**
   La técnica progresiva, también conocido como método jerárquico o de árbol, construye un alineamiento múltiple final realizando primero una serie de alineamientos de pares sobre secuencias sucesivamente menos emparentadas.

   Este método comienza alineando las dos secuencias más cercanamente relacionadas, luego de realizar este proceso continúa el alineamiento con aquellas secuencias que están más relacionadas con el alineamiento anterior hasta terminar con todo el espacio de búsqueda.

   Una limitación importante de los métodos progresivos es su fuerte dependencia de la asignación inicial del parentesco entre las secuencias, así como de la calidad del alineamiento inicial. De este modo, los métodos son sensibles también a la distribución de las secuencias en el conjunto problema.

2. **Métodos iterativos**
   El funcionamiento de este tipo de métodos es similar al anterior pero cada vez que alinea dos secuencias realínea las secuencias iniciales, lo que permite eliminar gran cantidad de los errores que se producen al utilizar un método progresivo.

[2]Un problema NP-completo es una categoría de problemas de decisión para los cuales no se conoce ningún algoritmo eficiente que los resuelva

Modelos ocultos de Markov

Una forma de búsqueda de hómologos menos estricta y que considera más secuencias son perfiles basados en modelos ocultos de Markov, modelos probabilísticos que asignan una probabilidad a todas los posibles estados de un aminoácido en un alineamiento de secuencias multiples, es decir, espacios, coincidencias o diferencias.

Los HMM (*Hidden Markov models*) pueden producir una salida única con la mayor puntuación, pero también pueden generar una familia de alineamientos posibles que puedan ser evaluados en su significancia biológica. Debido a que la técnica utiliza métodos probabilísticos, no producen la misma solución cada vez que se ejecutan sobre el mismo conjunto de datos, de esta forma, no pueden garantizar converger al alineamiento óptimo.

¿Cómo funcionan los perfiles ocultos de Markov?

Es una variación de la cadena de Markov oculta en la que la posición de un alineamiento de secuencias múltiples se convierte en los estados del modelo; la matriz de transición es la probabilidad de pasar de un estado/posición al siguiente.

De este modo, se introduce la probabilidad de emisión para cada estado, y por tanto la probabilidad de tener un determinado nucleótido o aminoácido en ese estado de la Cadena de Markov.

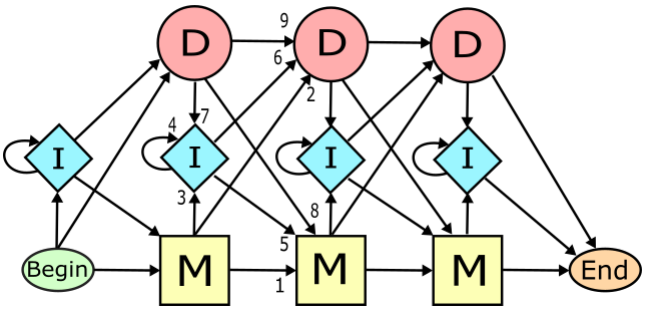


Figura 2: [Elaboración propia, 2021.] Perfil Oculto de Markov.

1.7. BLAST: un software basado alineamiento de secuencias

BLAST (Basic Local Alignment Search Tool) es un programa computacional de alineamiento de secuencias de tipo local, ya sea de ADN, ARN o de proteínas, desarrollado por los Institutos Nacionales de Salud del gobierno de EE. UU., al cual es posible acceder gratuitamente desde el servidor web del Centro Nacional para la Información Biotecnológica (NCBI). BLAST es capaz de comparar rápidamente una secuencia problema ingresada al programa

(también denominada como secuencia *query*) contra la numerosa cantidad de secuencias que se encuentran en la base de datos de NCBI, lo que la convierte en una herramienta fundamental en la investigación biológica, genética y genómica en curso. En efecto, el artículo inicial que describe el programa, publicado en el Journal of Molecular Biology y titulado ”Basic Local Alignment Search Tool”, fue la publicación más citada de la década de 1990 [4].

En los últimos años, el desarrollo paralelo de proyectos de secuenciación a gran escala y herramientas bioinformáticas como BLAST, ha permitido a los científicos estudiar el modelo genético de la vida en muchas especies, y también ha ayudado a conectar la biología y la informática en el campo de la bioinformática en proceso de maduración.

BLAST es ampliamente utilizado para encontrar posibles genes homólogos: Cuando una nueva secuencia es obtenida, a través de BLAST es posible compararla con las secuencias que han sido previamente caracterizadas, para así poder inferir su función, de modo que esta herramienta es la más usada para la anotación y predicción funcional de genes o secuencias proteicas.

1.7.1. Cómo funciona BLAST

La entrada de búsqueda de la herramienta BLAST permite cargar archivos de secuencia tipo FASTA, o bien, ingresar código de acceso a la base de datos de NCBI. Antes de inciar la búsqueda, el usuario puede seleccionar la base de datos a utilizar, el tipo de organismo de búsqueda, filtrar por Modeos (XM/XP),RefSeq proteins (WP) no redundantes, y excluir secuencias de muestras sin cultivar, o bien, ambientales. Además, permite seleccionar el tipo de algoritmo a utilizar y filtrar parámetros para investigadores más avanzados. Los algoritmos disponibles para la alineación de secuencias de proteínas son los siguientes:

- QuickBLASTP: Es una versión acelerada de BLASTP que es muy rápida y funciona mejor si el porcentaje de identidad objetivo es 50 % o más.
- BlastP: simplemente compara una consulta de proteínas con una base de datos de proteínas.
- PSI-BLAST: permite al usuario construir una PSSM (matriz de puntuación específica de la posición) utilizando los resultados de la primera ejecución de BlastP.
- PHI-BLAST: realiza la búsqueda pero limita las alineaciones a aquellas que coinciden con un patrón en la consulta.
- DELTA-BLAST: construye un PSSM utilizando los resultados de una búsqueda en la base de datos de dominio conservado y busca en una base de datos de secuencias.

Antes de BLAST, los programas de alineamiento utilizaban algoritmos de programación dinámica, como los algoritmos Needleman-Wunsch y Smith-Waterman, que requerían largos tiempos de procesamiento y el uso de supercomputadoras o procesadores de computadora paralelos [15], [16].

Aunque este tipo de programación dinámica hizo un trabajo completo al comparar cada residuo de una secuencia con cada residuo de una segunda secuencia y mantuvo un registro de qué tan bien se alineaban las secuencias en cada paso, estos algoritmos requerían una cantidad considerable de memoria y tiempo de procesamiento, de modo que estos programas requerían un hardware informático potente que era costoso, poco común y, en última instancia, poco práctico para la mayoría de los científicos y laboratorios.

Con el fin de aumentar la velocidad de alineación, el algoritmo BLAST se diseñó para aproximar los resultados de un algoritmo de alineación creado por Smith y Waterman (1981), pero para hacerlo sin comparar cada residuo con todos los demás residuos [17]. Por lo tanto, BLAST es de naturaleza heurística, lo que significa que tiene “atajos inteligentes” que le permiten ejecutarse más rápidamente [18]. Sin embargo, en esta compensación por una mayor velocidad, la precisión del algoritmo se reduce ligeramente.

1.7.2. La heurística de BLAST

BLAST aumenta la velocidad de alineación al disminuir el espacio de búsqueda o el número de comparaciones que realiza. Específicamente, en lugar de comparar cada residuo entre sí, BLAST usa segmentos cortos de ”palabra”(w) para crear ”semillas”de alineación. BLAST está diseñado para crear una lista de palabras a partir de la secuencia de consulta con palabras de una longitud específica, según lo defina el usuario. Exigir que solo coincidan tres residuos para sembrar una alineación, significa que es necesario comparar menos regiones de secuencia. Los tamaños de palabras más grandes generalmente significan que hay incluso menos regiones para evaluar.

Una vez que se siembra un alineamiento, BLAST la extiende de acuerdo con un umbral (T) establecido por el usuario. Al realizar una consulta, la computadora extiende las palabras con una puntuación de vecindad mayor que T. Se usa una puntuación de corte (S) para seleccionar alineamientos sobre el corte, lo que significa que las secuencias comparten similitud significativa. Si se detecta un acierto, el algoritmo verifica si w está contenido dentro de un par de segmentos alineados más largos que tiene una puntuación de corte mayor o igual que S [17]. Cuando una puntuación de alineamiento comienza a disminuir más allá de una puntuación de umbral más baja (X), la alineación se termina. Estas y otras variables se pueden ajustar para aumentar la velocidad del algoritmo o enfatizar su sensibilidad.

Una de las innovaciones más notables de BLAST es que el programa calcula la significación estadística para cada resultado. Esto se conoce como valor esperado (valor E) o valor de probabilidad (valor P) y se calcula para cada alineación. El valor E describe cuántos resultados puede esperar ver por casualidad al buscar en una base de datos de cierto tamaño, mientras que el valor P describe la probabilidad que el alineamiento que está observando se deba al azar. En general, cuanto más bajo sea el valor E o P, más probable es que una alineación sea significativa. Por debajo de la puntuación común de 10-5, P y E son aproximadamente equivalentes. [18]

1.7.3. Ventajas y desventajas de su uso

Algunas ventajas de usar el servidor del NCBI son que el usuario no tiene que mantener ni actualizar las bases de datos y que la búsqueda se hace en un cluster de computadoras, lo que otorga rapidez. Las desventajas son que no se permiten hacer búsquedas masivas, debido a que es un recurso compartido, la rapidez de respuesta disminuye considerablemente en la medida que se sensibiliza la búsqueda y además, las secuencias son enviadas al servidor del NCBI sin ningún tipo de cifrado, lo que puede ser un problema para quienes quieran mantener sus secuencias privadas.

Es importante destacar que BLAST usa un algoritmo heurístico para aumentar su rapidez, por lo que no puede garantizar que ha encontrado la solución correcta. Sin embargo, es capaz de calcular la significación de sus resultados, por lo que provee al usuario de un parámetro para juzgar los resultados que se obtienen.La aplicación local de BLAST tiene la ventaja de que permite manejar varios parámetros que en las búsquedas de NCBI están estandarizados, por lo que provee una mayor flexibilidad para los usuarios avanzados.

1.8. Herramientas de identificación automáticas utilizando genomas completos

La identificación del potencial biotecnológico de una bacteria en base a su genoma ha permitido apresurar procesos de selección y caracterización de cepas para su uso industrial. Tradicionalmente, esto se realiza mediante el análisis de todas las secuencias codificantes identificadas en el genoma utilizando software basado en alineamientos de secuencias para la identificación de homólogos con evidencia experimental. Algunos ejemplos de herramientas exitosas son nombrados a continuación.

1.8.1. AntiSmash

Es una plataforma web que permite la rápida identificación, anotación y análisis de todo el genoma, de grupos de genes de biosíntesis de metabolitos secundarios en genomas bacterianos y fúngicos.

Cómo funciona AntiSmash

La entrada del front-end del servidor web antiSMASH permite cargar archivos de secuencia de varios tipos (archivos FASTA, GBK o EMBL). Como alternativa, se puede proporcionar un número de acceso a GenBank/RefSeq, que el servidor web utiliza para obtener automáticamente el archivo GenBank asociado. Si el usuario opta por utilizar un archivo de entrada FASTA, la predicción de genes la realiza Glimmer3 utilizando su herramienta long-hole para construir un modelo de genes basado en la propia secuencia de entrada, o GlimmerHMM cuando se envían datos de entrada eucariotas.

Antes de iniciar la ejecución del análisis antiSMASH, el usuario puede seleccionar los tipos de agrupamientos de genes que desea buscar. Además, puede seleccionar qué módulos de análisis posteriores quiere incluir. Para los usuarios que, por ejemplo, trabajan con sus propios datos, existe una versión independiente con una interfaz gráfica de usuario en Java con las mismas opciones de entrada que la versión web. Por último, los usuarios expertos pueden optar por ejecutar el programa pipeline basado en Python directamente desde la línea de comandos para analizar por lotes un mayor número de entradas.

Para analizar la capacidad del genoma completo en la producción de metabolitos secundarios, los genes se extraen o predicen a partir de la secuencia de nucleótidos de entrada, y los grupos de genes se identifican con los pHMMs de los genes de firma. Posteriormente, se puede efectuar: análisis y anotación de dominios importantes para enzimas productoras de metabolitos, predicción de la estructura química central de los compuestos predichos, análisis comparativo de agrupamientos de genes usando ClusterBlast y análisis de familias de proteínas del metabolismo secundario usando la anotación smCOG. El resultado se visualiza en una página web XHTML interactiva, y todos los detalles se almacenan en un archivo EMBL para el análisis adicional y la edición en un navegador del genoma. También se genera un archivo de Microsoft Excel con un resumen de todos los grupos de genes detectados y sus detalles. El procedimiento antes descrito se aprecia en la Figura (3).

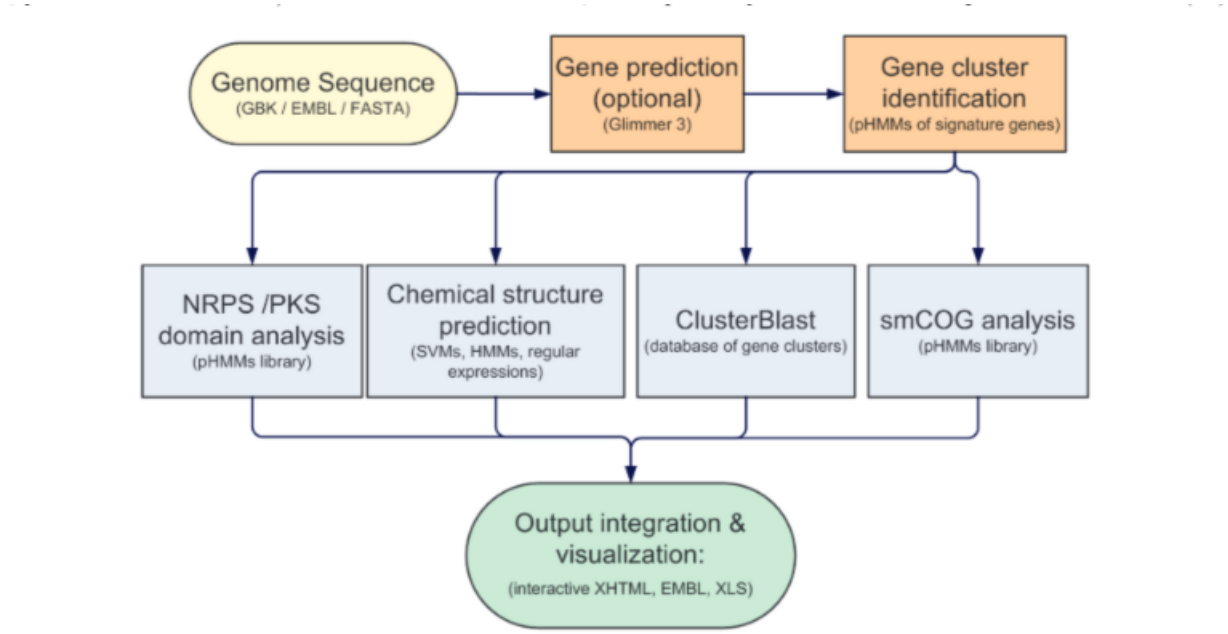


Figura 3: [Fuente: antiSMASH [8], Antismash pipeline. Accedido el 2021-09-01.] Pipeline del análisis de metabolitos secundarios.

1.9. Herramientas basadas en *Machine learning* en identificación de homólogos

1.9.1. DeepNOG

Es un método de asignación de grupos ortólogos *(OG*[3]*)* para proteínas, libre de algoritmos de alineamiento y basado en una red neuronal convolucional. Se caracteriza por ser muy rápida y eficiente computacionalmente además de tener una precisión comparable con métodos de alineación como son HMMER (basado en perfiles ocultos de Markov) o DIAMOND. [4] [13]

1.9.2. Cómo funciona DeepNog

La arquitectura de DeepNog está completamente basada en su predecesor DeepFAM [14], mejorando limitaciones como la restricción a proteínas con secuencias de longitud fija y la aplicabilidad para el uso de la base de datos de grupos ortólogos supervisados y no supervi-

[3]Los grupos ortólogos se definen como el conjuntos de genes que descienden de un único ancestro común dentro de un rango taxonómico de interés.

[4]Para más información, revisar `https://github.com/bbuchfink/diamond`

sados eggNOG[5]. A continuación se presenta un diagrama sobre la arquitectura que presenta DeepNog.

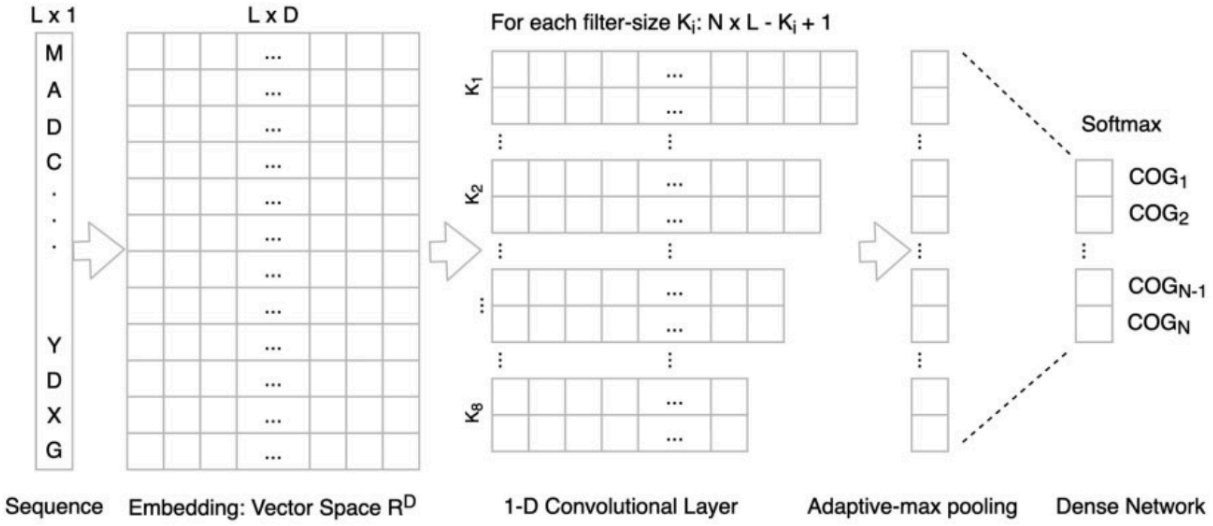


Figura 4: Fuente: DeepNOG[13], Accedido el 2021-09-01] Arquitectura de la red DeepNOG.

A grandes rasgos, una secuencia de entrada de un largo D es codificada por una capa llamada *Embedding layer*, en donde se vectoriza la secuencia formando un vector de dimensión $\mathbb{R}^D$, estas representaciones vectoriales también se entrenan conjuntamente con la red, por efectos prácticos cada aminoácido se representa como un vector de dimensión 10. Luego las secuencias vectorizadas se pasan a la extracción de características en la capa de convolución. Esta capa de convolución tiene una dimensión 1-D usando una función de activación llamada SELU (Scaled Exponencial Linear Unit). Al ser una capa convolucional 1-D, cada dimensión en el espacio de la secuencia vectorizada $\mathbb{R}^D$ es tratada por este filtro como un vector de característica de entrada independiente, a cada una de estas características se le cuantifica positiva o negativamente un peso para luego ser ponderados y eventualmente asignar un grupo COG (Cluster of Orthologous Group). La siguiente capa es una *Adaptative-max pooling*, que en palabras simples extrae el máximo valor de una fila de entrada en la capa convolucional, pero con la particularidad de que puede manejar vectores de tamaños variables a diferencia de la red de DeepFam. Por último, se agrega una capa de tipo *softmax* en donde se calcula la probabilidad de cada característica de la secuencia y se asigna la categoría (COG) con mayor probabilidad.

[5]Para más información revisar `http://eggnog5.embl.de/#/app/home`

1.10. Enzimas degradadoras

1.10.1. Oxigenasas Hidroxilantes de Anillo

Las Oxigenasas Hidroxilantes de Anillo, RHO por sus siglas en inglés (*Ring Hydroxilating Oxigenases*), son un sistema de enzimas multicomponente implicado en la degradación de diversos compuestos aromáticos en el medio ambiente, que incluyen aromáticos unidos y fusionados, olefinas alifáticas y aromáticos altamente sustituidos, así como muchos compuestos tóxicos y/o cancerígenos como los bifenilos policlorados (PCB) y policíclicos e hidrocarburos aromáticos (PAH). Todos los miembros de la familia RHO tienen una o dos proteínas de transporte de electrones solubles (ET), a saber: ferredoxinas y reductasas, que tienen flavin y/o centros Fe-S, que transfieren electrones de nucleótidos reducidos, NAD(P)H, al componente oxigenasa terminal (Figura 5). La oxigenasa terminal se compone principalmente de dos subunidades separadas, una gran subunidad catalítica (α) con un centro catalítico de Fe-S tipo Rieske y una pequeña subunidad estructural (β) en forma hetero-multimérica, $\alpha\beta$n. Sin embargo, ciertas dioxigenasas están desprovistas de subunidad β y existen en forma homo-multimérica, αn.

Las RHO inician el catabolismo de varios compuestos recalcitrantes al atacar el núcleo aromático inerte haciéndolos propensos a una mayor transformación y mineralización. Por lo tanto, estas enzimas son cruciales para estudiar la degradación de diferentes contaminantes orgánicos. Además, la naturaleza estereoespecífica de la catálisis por estas enzimas, las convierte en candidatas ideales para la modificación de productos naturales desde el punto de vista de los químicos farmacéuticos y sintéticos.[21]

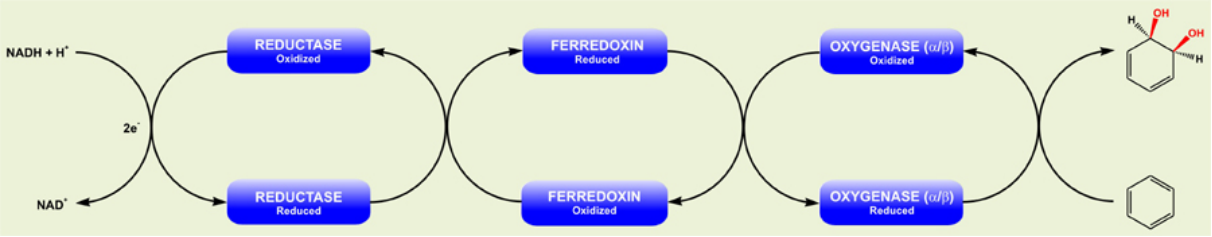


Figura 5: [Elaboración propia, 2021.] Diagrama de interacción entre los componentes de una RHO. [21]

1.11. Bases de datos

1.11.1. KEGG

La Enciclopedia de genes y genomas de Kioto, KEGG, por sus siglas en inglés (*Kyoto Encyclopedia of Genes and Genomes*) es una base de datos integrada que desarrolla ”modelos”de sistemas biológicos, en forma de mapas de rutas creadas manualmente mediante la captura de conocimientos de la literatura publicada. Los modelos KEGG se pueden utilizar en análisis biológicos de macrodatos, por ejemplo, para descubrir funciones sistémicas de un organismo, codificado en la secuencia de su genoma.[19]

La base de datos KEGG, permite descubrir funciones a nivel celular y de organismo a partir de secuencias de genomas y otros conjuntos de datos moleculares, mediante el procedimiento de mapeo, que se guía por el concepto de ortólogos funcionales (homología). Los mapas de la ruta KEGG, así como las clasificaciones de la familia de proteínas BRITE y los módulos KEGG, se crean como redes de identificadores, incluidos los metabolitos mapeados en rutas metabólicas, fármacos y enfermedades mapeados en clasificaciones jerárquicas BRITE, y organismos celulares y virus mapeados en la taxonomía NCBI.

KEGG consta de 16 bases de datos en cuatro categorías, como se muestra en la Tabla 1.11.1 bases de datos, excluyendo GENES y ENZYME, son bases de datos originales creadas manualmente por los inveestigadores del *Institute for Chemical Research*, de la universidad de Kioto. Los datos de secuencia en GENES son extraídos de RefSeq, GenBank y otras bases de datos de secuencias públicas, y se les da una anotación original de las funciones de genes/proteínas representadas por un código interno (KO). Los números de clase de enzima (EC) en ENZYME fueron extraídos de ExplorEnz [20], la base de datos oficial de Nomenclatura de enzimas, y se les proporciona una anotación de enlaces de datos de secuencias de enzimas.

1.11.2. RHOBase

La base de datos Ring-Hydroxylates Oxygenase (RHO), abreviada como “RHObase”, es una base de datos web, curada manualmente y con capacidad de búsqueda que proporciona información completa sobre todas las RHO bacterianas de tipo Rieske estudiadas y caracterizadas bioquímicamente. La versión actual de la base de datos compila alrededor de 1000 entradas que incluyen 196 subunidades alpha de oxigenasa, 153 subunidades beta de oxigenasas, 92 ferredoxinas y 110 reductasas, distribuidas entre 131 cepas bacterianas diferentes. Las proteínas están unidas a las estructuras PDB disponibles y los correspondientes dominios conservados (y motivos). La base de datos también incluye información sobre más de

Category	**Database**	**Content**	**ID KEGG**
Información del sistema	PATHWAY	Mapa de ruta KEGG	Número de mapa
	BRITE	Jerarquías y tablas Brite	Número br/ko
	MODULE	Módulos KEGG	Número M
		Módulos de reacción	Número RM
Información genómica	KO	Grupos KO para ortólogos funcionales	Número K
	GENES	Genes y proteínas	<org>:<entry>
	GENOME	Organismos y virus KEGG	Número T, gn:<org>
Información química	COMPOUND	Moléculas pequeñas	Número C
	GLYCAN	Glicanos	Número G
	REACTION	Reacciones bioquímicas	Número R
	RCLASS	Clase de Reacción	Número RC
	ENZYME	Nomenclatura de enzima	Ec:<entry>
Información de salud	NETWORK	Mapas de variación de red	Número nt
		Elementos de la red relacionados con enfermedades	Número N
	VARIANT	Variantes de genes humanos	hsa_var:<entry>
	DISEASE	Enfermedades humanas	Número H
	DRUG	Medicamentos	Número D
	DGROUP	Grupos de medicamentos	Número DG

Tabla 1: Contenido e identificadores de la base de datos de KEGG. [19]

cien compuestos aromáticos y los mecanismos de oxigenación seguidos por diferentes RHO que implementan un total de 318 reacciones de oxigenación. Además de la recuperación de datos, también hay herramientas analíticas integradas mediante las cuales los usuarios también pueden realizar búsquedas rápidas en la base de datos para la predicción de sustratos putativos para sus secuencias de oxigenasa de consulta o pueden buscar RHO candidatos potenciales capaces de transformar un compuesto deseado.[21]

1.11.3. AromaDeg

La base de datos de degradación de hidrocarburos aromáticos (AromaDeg) es un recurso web dirigido a la degradación aeróbica de aromáticos que comprende bases de datos actualizadas a 2013, seleccionadas y construidas manualmente considerando un enfoque filogenómico. Basado en análisis filogenéticos de secuencias de de familias de proteínas catabólicas clave y de función documentada, el programa permite consultas y minería de datos de nuevos conjuntos de datos genómicos, metagenómicos o metatranscriptómicos. Cada secuencia de consulta que coincide con una familia de proteínas de AromaDeg, se asocia a un grupo específico de un árbol filogenético dado, y la anotación de función adicional y/o la especificidad del sustrato, pueden inferirse de los miembros del grupo vecino con función validada experimentalmente. Esto permite una caracterización detallada de superfamilias de proteínas individuales, así como clasificaciones funcionales de alto rendimiento. Por lo tanto, AromaDeg aborda las deficiencias de la predicción de la función de proteínas basada en homología, combinando la construcción de árboles filogenéticos y la integración de datos experimentales para obtener anotaciones más precisas de nuevos datos biológicos relacionados con las vías de biodegradación aromática aeróbica. [22]

La base de datos de AromaDeg cuenta con 3605 secuencias, clasificadas en las siguientes familias:

- Rieske non heme iron oxygenases
  - Phthalate oxygenases
  - Biphenyl oxygenases
  - Benzoate oxygenases
  - Salicylate oxygenases
- Extradiol oxygenases
  - Viccinal chelate superfamily

    EXDO miscellaneous

    EXDO monocyclic substrates

    EXDO bicyclic substrates

- LigB superfamilily

  Homoprotocatechuate

  Protocatechuate

  Cupin superfamily

  Gentisate

1.11.4. UniProt

Universal Protein Resource (UniProt) es un recurso completo de secuencias de proteínas y datos de anotación.
UniProt es una colaboración entre el Instituto Europeo de Bioinformática (EMBL-EBI), el Instituto Suizo de Bioinformática SIB y el Protein Information Resource (PIR). En los tres institutos, más de 100 personas participan a través de diferentes tareas, como la conservación de bases de datos, el desarrollo y el soporte de software.

Las bases de datos de UniProt permiten secundar la investigación biológica y biomédica al proporcionar un compendio completo de todos los datos de secuencias de proteínas conocidas vinculadas a un resumen de la información funcional verificada experimentalmente o predicha computacionalmente sobre esa proteína. La base de conocimiento de UniProt (UniProtKB) combina las entradas revisadas de UniProtKB/Swiss-Prot, las cuales incluyen diversos datos asociados a su información genética y reactividad bioquímica; con las entradas de UniProtKB/TrEMBL no revisadas que están anotadas por sistemas automatizados. UniProt además integra, interpreta y estandariza datos de múltiples recursos seleccionados para agregar conocimiento biológico y metadatos asociados a los registros de proteínas y actúa como un centro desde el cual los usuarios pueden conectarse a otros 180 recursos [23].

CAPÍTULO 2
Identificación del Problema y Propuestas de Mejora

2.1. Comprensión del problema y análisis de impacto

Actualmente, las técnicas de caracterización de microorganismos son ejecutadas a través de estudios físicos, químicos y experimentales; o bien, utilizan la información genética para analizar las capacidades del aislado bacteriano. De los procesos señalados, el estudio genético es el más reciente y eficiente, pero, aún así, requiere de un largo periodo de ejecución y personal altamente calificado, tanto en conocimiento técnico, como en experiencia práctica para el análisis de resultados. Específicamente en este contexto, el área tecnológica ha desarrollado pocos avances para aportar en el procesos de caracterización de microorganismos, una de las razones es que hay una falta de bases de datos que guarden información específica según determinadas funciones enzimáticas, además no hay una estandarización para identificar cualidades relevantes de enzimas entre los distintos datasets del mundo. Ahora bien, herramientas como los alineadores múltiples de secuencias, búsqueda de homólogos basados en Cadenas Ocultas de Markov u otros algoritmos ayudan al proceso de caracterización pero no son del todo útiles cuando los genomas contienen miles de secuencia debido a la complejidad algorítmicas que estos presentan[6], además de que una alineamiento sin análisis no provee de información concluyente ni certezas en cuanto a las funciones enzimáticas de una proteína. La tecnología en sí misma puede ahorrar meses de trabajo a investigadores en los procesos bioquímicos[5] y agilizar el proceso de reconocimiento de enzimas que puedan ser de utilidad para resolver problemáticas latentes, como por ejemplo el cambio climático, la recuperación de matrices ambientales contaminadas y utilización en procesos biotecnológicos.

Una de las aplicaciones que aporta esta memoria es la identificación de enzimas que cumplan con funciones de degradación de contaminantes aromáticos. Actualmente los derrames de hidrocarburos aromáticos tales como el diésel o el petróleo crudo causan estragos en el medioambiente y en Chile[7]. De hecho, no solo es un problema medioambiental, sino también económico, existiendo estimados que calculan pérdidas de 2.9 billones de dolares (equivalente al 3.3 % del Producto Interno Bruto mundial)[3]. A raíz de lo anterior, es imperativo para el desarrollo sustentable encontrar formas de degradar estos compuestos químicos de los suelos y los océanos, pero la búsqueda de organismos y enzimas con este tipo de funciones es lenta por el trabajo experimental necesario, además de lo engorroso que es verificar

[6] Recordar que estos algoritmos son de tipo NP-completo

[7] Informe Medioambiental del año 2020 del Instituto Nacional de Estadística (INE), página 155, capítulo 16, *16.2 Derrames de contaminantes*: `https://www.ine.cl/docs/default-source/variables-basicas-ambientales/publicaciones-y-anuarios/informe-anual-de-medio-ambiente/informe-anual-de-medio-ambiente-2020-(versi%C3%B3n-actualizada-al-25-de-febrero-de-2021).pdf?sfvrsn=a6ddf6f1_2`

la homología y la función de una enzima, sumado a la expertiz que hay que tener para validar la información obtenida.

2.2. Propuesta de mejora: ADEC

Hoy en día una de las plataformas que más se acercan al objetivo de este trabajo es antiSmash (Sección 1.8.1), tanto la interfaz de usuario como la funcionalidad tienen elementos útiles que replicar en el contexto de caracterizar microorganismos y enzimas asociadas a degradación. Como solución se propone la plataforma "Aromatic compounds Degradation Enzyme Classifier"(ADEC), una herramienta basada en Machine Learning enfocado en el reconocimiento de patrones de enzimas asociadas a degradación para una caracterización automática de enzimas. Un gran problema incluye la confección de datasets necesarios para confeccionar una herramienta como ADEC por lo que antes de realizar cualquier modelo para predecir las funciones enzimáticas de un microorganismo se debe recopilar, limpiar y estandarizar información que hoy en día está dispersa en distintas bases de datos públicas y a eso está destinado el presente trabajo. En especial este trabajo se enfocará en el desarrollo de datasets para la clasificación binaria de enzimas degradadores y no degradadores, como también el desarrollo de modelos basados en ML y validación en genomas bacterianos.

ADEC es una aplicación web que permite al usuario identificar si una secuencia problema corresponde a una enzima degradadora de compuestos aromáticos, y clasificarla de acuerdo al tipo de enzima al que corresponde y el sustrato primario que degrada.

El uso de aprendizaje de máquinas para la caracterización, permite que la respuesta sea considerablemente más rápida en comparación con las alternativas actuales de búsqueda de homólogos, además de que esta no depende de la cantidad de secuencias que ingrese el usuario, ni requiere de un análisis posterior, ya que no realiza una alineación de secuencias. De esta manera, ADEC es una alternativa innovadora que permite reducir considerablemente los tiempos de procesamiento y análisis de datos, representando un salto importante en la caracterización de microorganismos.

2.3. Implementación de la solución

ADEC consta de 4 módulos principales que interactúan entre si para poder caracterizar las enzimas de un genoma completo. Estos módulos son la Interfaz de usuario, el Visualizador de contexto genómico, el Clasificador y la Base de datos, a lo largo de esta propuesta nos centraremos en los últimos dos. A continuación en la Figura 6 se puede ver un diagrama que representa la interacción que tiene cada módulo, sus entradas y salidas.

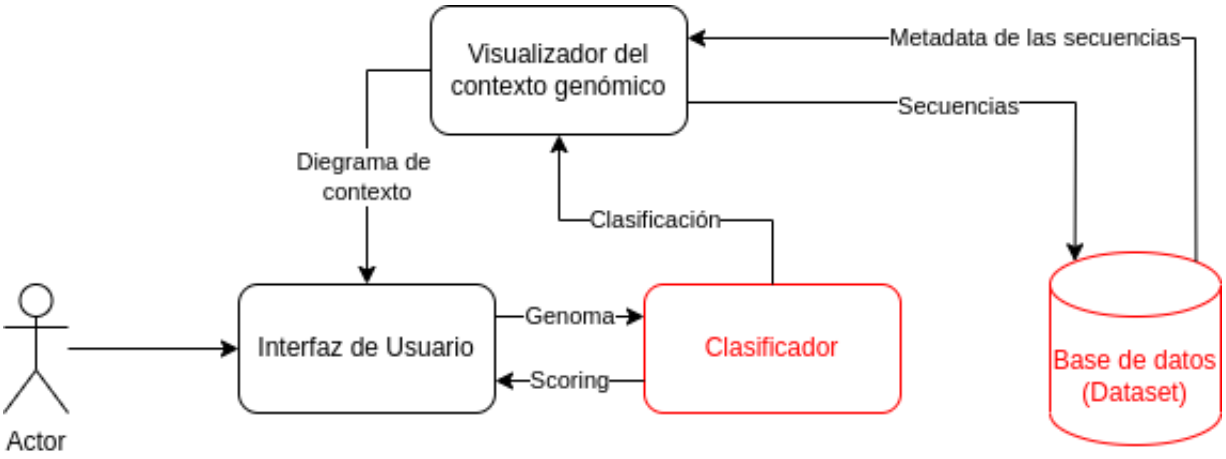


Figura 6: [Elaboración propia, 2021.] Diagrama de la arquitectura de ADEC. En rojo los módulos en los que se trabaja a lo largo de esta memoria.

El módulo de clasificación por su parte consta de 3 submódulos que interactúan como se muestra en la Figura 7. Aquí las proteínas de un genoma anotado de un microorganismo en formato .fasta o similar entra para primero ser vectorizado, de esta forma los modelos implementados pueden entender como trabajar con las secuencias. El segundo paso es clasificar las enzimas entre degradadoras y no degradadoras, si la enzima no es degradadora entonces se etiquetará como tal y no pasarán al siguiente módulo y si por el contrario se clasifica como degradadora de contaminantes aromáticos pasa al último submódulo. El submódulo de clasificador enzimática es el encargado de encontrar la función que cumple la enzima de entrada, categorizándola según su Familia, Sustrato Primario, Estructura Primaria, Polaridad, etc. Este submódulo queda fuera del alcance de este trabajo por lo que no se detallará en profundidad. De hecho a lo largo de esta memoria el autor solo se adentra en detalles de los módulos marcados en las figuras.

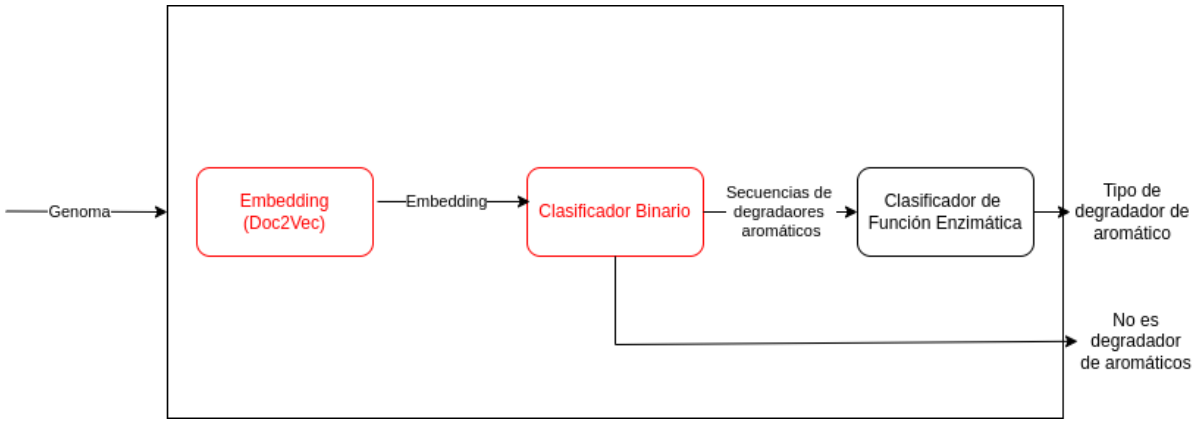


Figura 7: Diagrama de la arquitectura del módulo de clasificación. En rojo los módulos que se trabajan a lo largo de esta memoria.

2.3.1. Creación de datasets

Por supuesto antes de crear un clasificador de cualquier tipo es necesario reunir mucha información, limpiarla y estandarizarla para que sea consumida más fácilmente. El objetivo es implementar un dataset con la información más depurada posible con la que eventualmente un clasificador de función enzimática pueda realizar predicciones entregando la mayor cantidad de información útil al investigador.

2.3.2. Dataset basado en AromaDEG

Para comenzar a armar una base de datos sobre enzimas degradadoras de contaminantes aromáticos se usa AromaDEG, uno de los dataset más conocidos y respaldados para encontrar este tipo de proteínas[8]. En base a lo que nos señalan los creadores de esta base de datos *"AromaDeg aborda las deficiencias de la predicción de la función de proteínas basada en homología, combinando la construcción de árboles filogenéticos y la integración de datos experimentales para obtener anotaciones más precisas de nuevos datos biológicos relacionados con las vías de biodegradación aromática aeróbica."*[22] podemos confiar en la veracidad de estos datos y usarlos para el primer propósito: Crear un dataset de enzimas degradadoras de contaminantes aromáticos.

El primer paso es reunir toda la información que provee AromaDEG complementándola con datos que los expertos en el área puedan aportar, por lo que para cumplir este cometido se crea un script en Python que pueda transformar todos los ficheros .fasta en un solo archivo de tipo .csv en donde se guardan principalmente la secuencia junto con toda la linea de

[8]Para más información visitar `http://aromadeg.siona.helmholtz-hzi.de/`

cabecera para cada enzima. Este script recibe como entrada un archivo .fasta y agrega la información extraída a un *DataFrame*, este proceso se repite hasta que todos los archivos obtenidos desde el repositorio de AromaDEG estén agregados al fichero .csv.

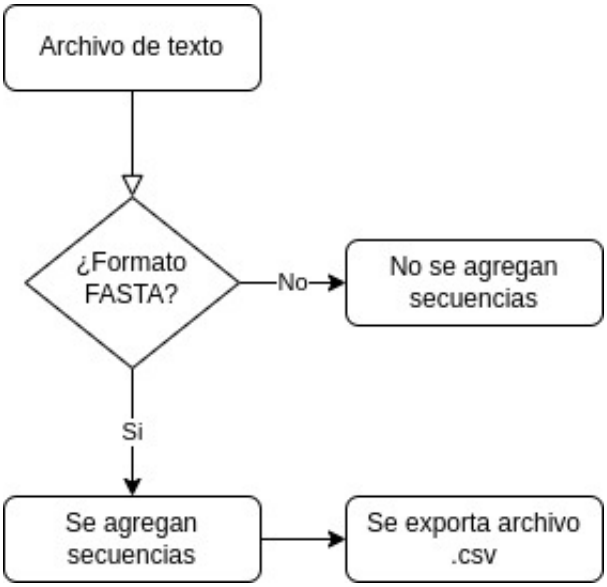


Figura 8: [Elaboración propia, 2024.] Diagrama de flujo de script para agregar enzimas desde un archivo en formato FASTA a un archivo .csv.

Finalmente los expertos en el área de biodegradación analizan este archivo final y lo complementan con su conocimiento, como se mencionó anteriormente. Al concentrar toda la información en un archivo se cuenta con un poco más de 3000 entradas.

Por supuesto, los datos provistos por AromaDEG no son suficientes para crear un dataset robusto y voluminoso en datos el cual pueda ser usado para entrenar una red, es por esto que se debe complementar el trabajo con una herramienta pionera en el uso del Machine Learning para reconocimiento de patrones en datos genómicos. La herramienta mencionada es DeepNOG [13] y como se muestra en capítulos anteriores DeepNOG nos ofrece una rápida y precisa forma de clasificar proteínas en Clusters de Grupos Ortologos (COG) a partir de la secuencia que las representa. A raíz de lo anterior es que se decide usar esta herramienta para poder encontrar los grupos COG principales donde se alojen enzimas que entre sus funciones se encuentre la de degradar contaminantes aromáticos, luego sabiendo que entre las proteínas de un mismo grupo COG existe homología es posible extraer más secuencias que aumenten el tamaño del dataset recién creado. A continuación se muestra un ejemplo del uso de DeepNOG con los datos obtenidos por AromaDEG y una tabla con los principales grupos COG obtenidos.

```
> deepnog infer EXDO_bicyclic_substrates.fasta -o prediction.csv
[2021-12-13 12:45:14] deepnog.client.client - INFO - Starting deepnog
[2021-12-13 12:45:15] deepnog.client.client - INFO - Loading NN-parameters from
/home/de4dbeef/deepnog_data/eggNOG5/2/deepnog.pth ...
[2021-12-13 12:45:16] deepnog.client.client - INFO - Accessing dataset from EXDO
_bicyclic_substrates.fasta ...
[2021-12-13 12:45:16] deepnog.client.client - INFO - Starting protein sequence g
roup/family inference ...
[2021-12-13 12:45:16] deepnog.learning.inference - INFO - Inference device: cuda
deepnog inference: 440seq [00:00, 3.01kseq/s]
[2021-12-13 12:45:16] deepnog.learning.inference - INFO - Inference complete.
[2021-12-13 12:45:16] deepnog.client.client - INFO - Writing prediction to predi
ction.csv
[2021-12-13 12:45:16] deepnog.client.client - INFO - All done.
>
```

Figura 9: [Elaboración propia, 2024.] Uso de la herramienta DeepNOG para predecir el grupo COG de una Extradiol dioxygenasa. Notar que se genera un archivo de salida llamado ”prediction.csv”.

Este proceso se lleva a cabo con todos los fastas descargados desde AromaDEG, luego los resultados obtenidos nos muestran las clasificaciones de cada enzima en un grupo COG específico junto con un porcentaje de exactitud de la predicción, todo queda guardado en un archivo en formato .csv llamado *prediction.csv*[9]. Ya con todos los archivos de predicción que se obtienen de la aplicación de DeepNOG en AromaDEG es momento de unirlos en un solo fichero para analizarlos. El análisis final nos muestra varios grupos COG que se repiten dentro de las predicciones, en la tabla 2.3.2 presentan los más importantes y con mayor exactitud. Desde aquí podemos notar fácilmente que los grupos más interesantes son:

- **COG3384: Aromatic ring-opening dioxygenase, catalytic subunit, LigB family**.
- **COG0346: Catechol 2,3-dioxygenase or related enzyme, vicinal oxygen chelate (VOC) family**
- **COG4638: Phenylpropionate dioxygenase or related ring-hydroxylating dioxygenase, large terminal subunit** .

Toda esta información fue facilitada para el grupo de biodegradación que posteriormente buscará enzimas homologas que puedan ser agregadas al dataset final.

[9] La instalación y otras funciones de DeepNOG se pueden hallar en su repositorio oficial `https://github.com/univieCUBE/deepnog`

Clasificación AromaDeg	Code COG	Cantidad de secuencias	Exactitud
protocatechuate	COG3384	462	0.99286
protocatechuate	COG3885	6	0.89103
protocatechuate	COG1355	4	0.92066
monocyclic	COG0346	495	0.97980
monocyclic	COG2105	3	0.70806
miscellaneous	COG0346	506	0.97916
miscellaneous	COG2514	4	0.87857
homoprotocatechuate	COG2078	17	0.71798
homoprotocatechuate	COG3153	4	0.90715
bycyclic	COG0346	419	0.97801
bycyclic	COG3153	2	0.99165
NonHemeIronOxy	COG4638	137	0.99998
PhthalateO	COG4638	232	0.99416
BenzoateO	COG4638	303	0.99998

Tabla 2: Clasificaciones de DeepNOG para la base de datos AromaDEG.

2.3.3. Dataset basado en KEGG

Para ampliar aún más el espectro de enzimas degradadoras de contaminantes aromáticos es que se buscan nuevas bases de datos de las que descargar más información valiosa para construir un dataset. Por recomendación del cliente, la búsqueda de nuevos datos se realizó en la base de datos KEGG, que como se describe en capítulos anteriores, provee de mucha información relacionada a bases de datos de genomas, rutas enzimáticas y sustratos químicos.

La base de datos específica de la que se extrajo información fue **KEGG Pathway**[10] en donde es posible encontrar rutas enzimáticas de múltiples procesos, en particular nos centramos en la ruta de degradación enzimática de compuestos aromáticos [11] a la cual se le hizo un proceso de WebScrapping para obtener toda la información de las familias KEGG que participan en dicho proceso obteniendo alrededor de 205 secuencias extras con su metadata incluída en donde destacan datos como el Organismo, nombre de la enzima, Gen asociado y uno de los más importantes, el Sustrato Primario.

Hasta este punto, si bien se dispone de una buena cantidad de datos proveídos por AromaDEG, se pudo notar que solamente la base de datos KEGG nos proporciona información vital para poder construir un clasificador de función enzimática ya que además de solo la secuencia, también se pueden obtener características de interés bioquímico muy importantes para poder predecir información valiosa como la ruta de degradación en la que se ve envuelta la enzima analizada o el contexto genómico de la misma.

[10] `https://www.genome.jp/kegg/pathway.html`
[11] `https://www.genome.jp/pathway/map01220`

Con estos antecedentes es que se decide dar comienzo a un proceso de *Data Aumentation* para los pocos datos obtenidos de KEGG. Para lograr esto es que se hace uso de una herramienta que al día de hoy es muy conocida en el mundo de la Bioinformática, BLAST: un software basado alineamiento de secuencias, en específico su versión dedicada a proteínas **BLASTP**[12]. Para utilizar la herramienta de alineación es imprescindible disponer de un archivo en formato FASTA, ya que como se describe en los capítulos anteriores la alineación múltiple de secuencias se ejecuta en base a una secuencia de aminoácidos. Es por esta razón que se vuelve necesario crear un *script* que pueda convertir el fichero (.csv, .xlsx, etc) que guarda los datos a un archivo en formato FASTA. El diagrama de flujo que representa dicho script es el siguiente.

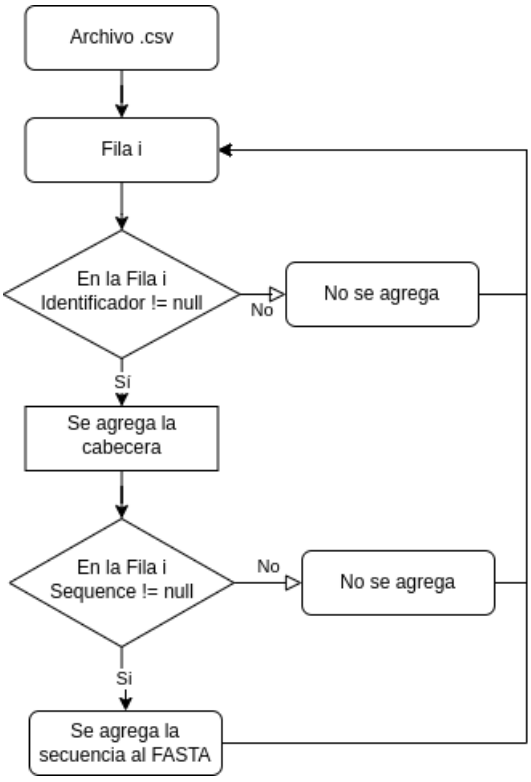


Figura 10: [Elaboración propia, 2024.] Diagrama de flujo para crear un archivo FASTA a partir de un .csv.

Amplificando datos de KEGG

Este proceso en total consta de 4 pasos específicos de procesamiento de datos y 1 de análisis, los cuales se muestran a continuación.

1. **CSV a FASTA:** Como se detalla en la Figura 10 es posible obtener un archivo .fasta a

[12]Este programa usa como entrada una secuencia de nucléotidos y la compara con una base de datos solo de proteínas.

partir de un dataset en formato .csv o similar. Este paso es importante porque eventualmente se hace uso de la herramienta DeepNOG en donde obligatoriamente se debe de utilizar un fichero con formato .fasta para ejecutar las predicciones.

2. **Paso por DeepNOG:** Con el fin de poder identificar los grupos COG más interesantes es usada nuevamente la herramienta DeepNOG sobre el .fasta que representa al dataset de KEGG.

3. **Script de filtrado y separación:** Con el output de DeepNOG se procede a analizar y hacer un filtrado por la exactitud de las predicciones, en específico se decide mantener las predicciones con exactitud por sobre el 75 %. Además por cada COG que cumpla con la condición anterior se le genera un archivo .fasta en donde se guardarán todas las secuencias que se clasifiquen como dicho COG, de esta forma se segregarán las secuencias de KEGG según el grupo COG al que pertenecen.

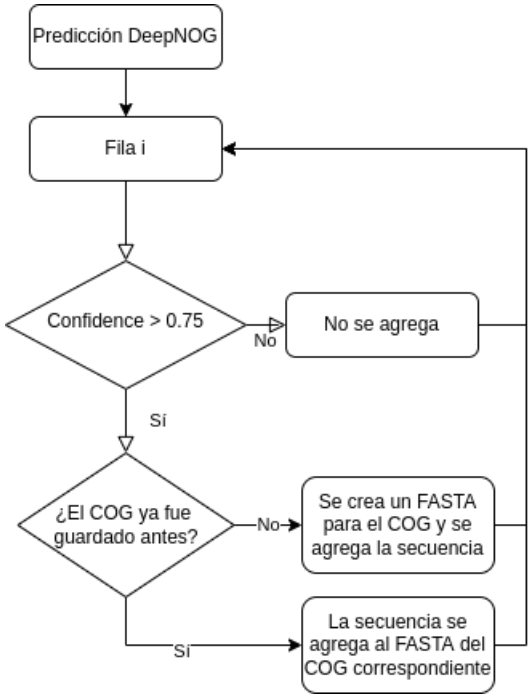


Figura 11: [Elaboración propia, 2024.] Diagrama de flujo para filtrar y separar los COG de KEGG.

4. **Alineación múltiple con BLASTP:** El último paso se trata de hacer uso de la herramienta BLASTP, como se menciona en la sección 1.7 este software toma una secuencia de entrada y la compara con secuencias de enzimas guardadas en su base de datos. Posteriormente entrega una lista de todas las enzimas que coinciden con la secuencia de entrada con los parámetros de cada comparación. A continuación se muestra un ejemplo de la configuración que se elige para hacer la búsqueda de nuevas proteínas.

En la página de BLAST se debe buscar la opción de BLASTP[13] en donde se desplegará un panel como el siguiente.

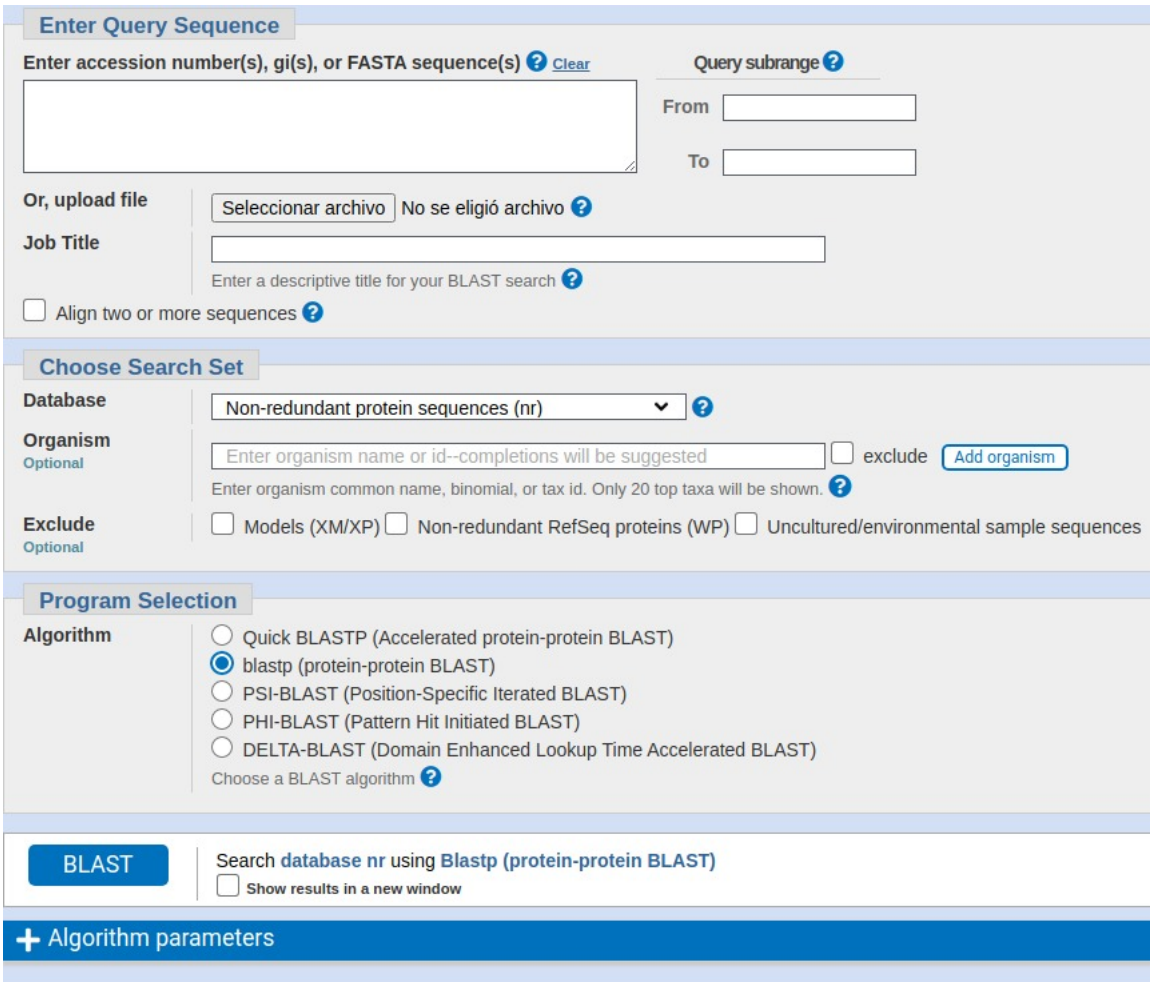


Figura 12: [Elaboración propia, 2024.] Panel de configuración para la alineación de BLASTP.

A continuación se va a desglosar la Figura 12 para explicar cada configuración. En primer lugar está la sección para agregar la secuencia a comparar, también es posible subir un archivo en formato .fasta o similar. En la subsección *Query subrange* se puede configurar la posición desde donde se quiere comenzar a comparar la secuencia de entrada hasta un índice final. En la Figura 13 se muestra una carga del archivo "AAK62353.txt" donde el formato que tiene sigue el patrón de un archivo .fasta.

[13] `https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome`

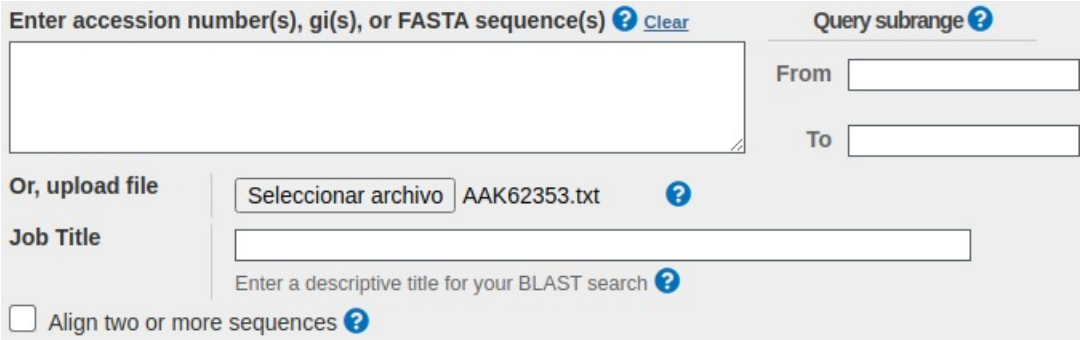


Figura 13: [Elaboración propia, 2024.] Carga de las secuencias a analizar en BLASTP.

Luego se debe seleccionar la base de datos sobre la cual se hace la alineación múltiple de secuencias. En este apartado se selecciona la Base de Datos “Non-redundant protein sequences (nr)” que como bien dice su nombre son secuencias de proteínas no redundantes, eso quiere decir que cada secuencia a comparar se encuentra una única vez dentro de toda la base de datos. En cuanto a la sección de *Organismo* se deja la opción “Bacteria (taxis:2)” para limitar la búsqueda solo a secuencias que correspondan a bacterias.

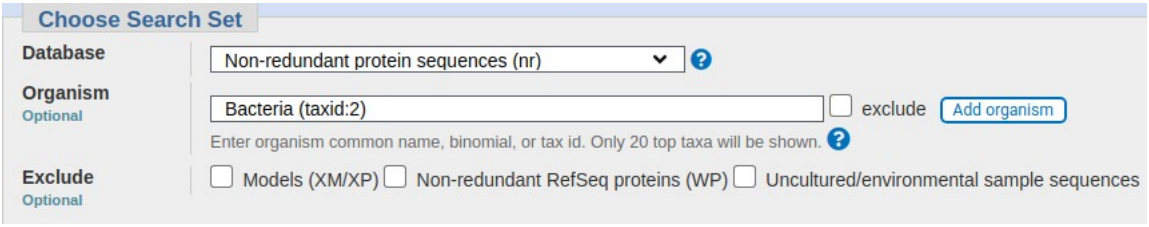


Figura 14: [Elaboración propia, 2024.] Panel de selección de bases de datos para comparar entradas.

En tercer lugar, está la sección de selección del programa que hace la alineación, por supuesto debe estar seleccionada “BLASTP” que es el alineador para secuencias de proteínas.

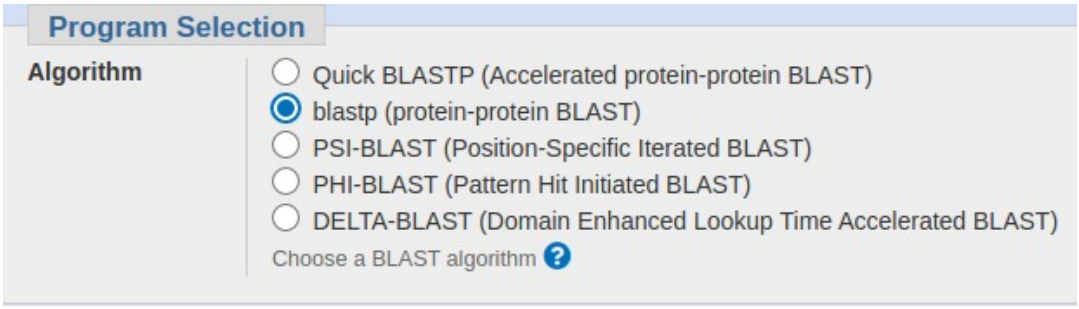


Figura 15: [Elaboración propia, 2024.] Panel de selección de programa.

Ya con todas las configuraciones realizadas se puede comenzar con la alineación, dependiendo de la cantidad de secuencias entregadas a BLAST el tiempo de ejecución varía, cabe recordar que BLAST es un alineador múltiple de secuencias el cual en el contexto de complejidad de algoritmos es un problema de tipo NP-Completo. Una vez que la alineación termina de ejecutarse se despliegan las siguientes secciones.

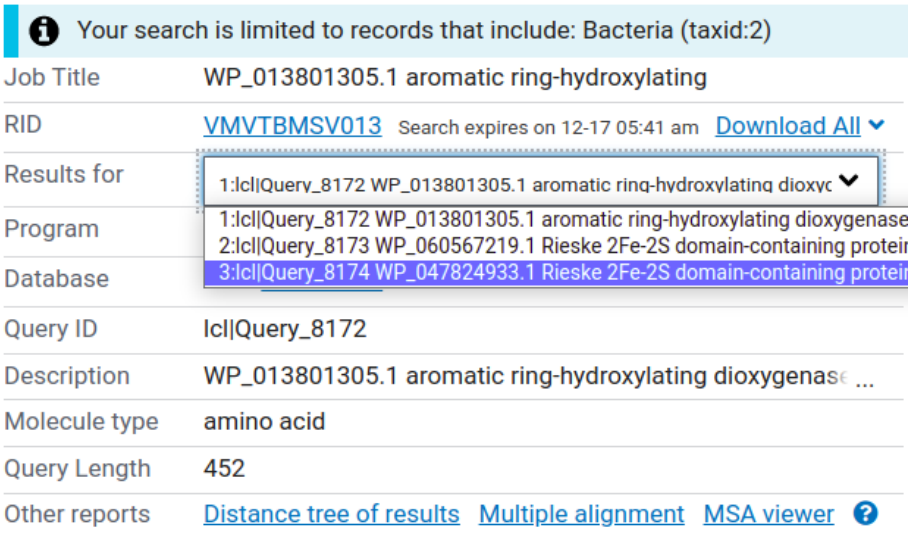


Figura 16: [Elaboración propia, 2024.] Resultados de la alineación de BLAST en su sección de información sobre la secuencia alineada.

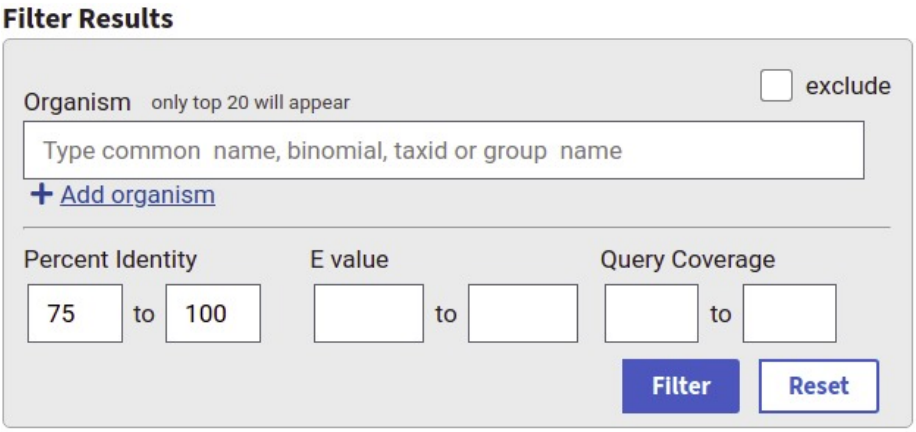


Figura 17: [Elaboración propia, 2024.] Resultados de la alineación de BLAST en su sección de información sobre la secuencia alineada.

En la Figura 16 se muestra la información rescatada desde el archivo .fasta entregando una descripción e información varia de cada enzima que trae el fichero subido. Además, cabe destacar que en el apartado “Results for” se enlistan todas las enzimas que fueron analizadas y alineadas, en este caso el archivo original tiene 3 secuencias para

analizar y por ello hay tres alineaciones distintas para seleccionar. También como se ve en la Figura 17 se cuenta con un filtro para clasificar según porcentaje de similitud, E-valores y cobertura de la consulta. Por recomendación de la contraparte se usaron los valores de similitud entre 75 % y 100 %, luego se descargan todas las secuencias filtradas y se guardan los resultados en archivos .txt separados identificados por el identificador de la secuencia de entrada. Es importante notar que por cada filtrado se debe descargar una lista de secuencias similares, por lo que de una sola secuencia que ingresemos a BLAST es posible obtener más de 500 secuencias extras (en los mejores casos), así es que aproximadamente el proceso termina con más de 5.000 secuencias para ser analizadas.

A modo de síntesis se puede revisar la Figura 18 en donde se ilustra el proceso de creación del dataset basado en los datos de KEGG.

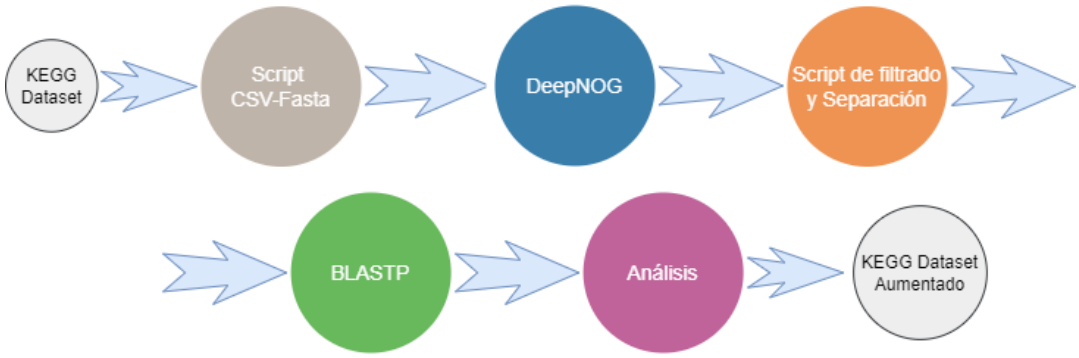


Figura 18: [Elaboración propia, 2024.] Diagrama paso a paso del proceso de creación del dataset basado en KEGG.

Finalmente toda la información obtenida se le entrega a los expertos en bioquímica para su posterior análisis ya que si bien hay altos porcentajes de similitud no siempre se encontrarán enzimas degradadoras de contaminantes aromáticos.

2.3.4. Dataset basado en RHOBase para Clasificador de Función Enzimática

Si bien hasta el momento se crearon 2 datasets con mucha información para analizar, el proceso es muy laborioso para el equipo de biodegradación ya que el objetivo es crear un dataset que primero cumpla con altos estándares de calidad de datos, es decir con respaldo científico acerca de la capacidad de degradar contaminantes aromáticos y en segundo lugar, que cuente con propiedades interesantes de las enzimas almacenadas. A raíz de esto es que se decide acotar un poco más el problema enfocándose en enzimas del tipo RHO y aprovechando una base de datos como la de RHOBase (Sección 1.11.2) que provee proteínas de este tipo. Así es como el equipo de biodegradación forma un subdataset

con secuencias que representan enzimas de tipo RHO degradadoras de contaminantes aromáticos con respaldo científico y entregando las propiedades más importantes de cada una. Adicionalmente el cliente recomienda guardar solo las secuencias que se clasifiquen en el grupo COG4638[14] para tener más seguridad de su capacidad degradadora.

El subdataset creado de RHOBase fue procesado de una forma similar al dataset formado apartir de KEGG, cabe destacar que esta nueva colección de datos tiene una calidad mucho mayor que las anteriores ya que cada una de las 173 enzimas que fueron agregadas se revisaron manualmente de forma analítica y respaldadas científicamente por publicaciones en donde se verifica sus funciones degradadoras, además de pasarlas por la red DeepNOG para verificar la su pertenencia al grupo COG correspondiente. A continuación se detallan los pasos a seguir para procesar y aumentar el volumen de estos datos.

1. **Creación de archivos .fasta:** Si bien este paso ya es común para poder aumentar los datos de un dataset, para este caso en particular hay una excepción. Debido a que las secuencias de estas enzimas RHO son mucho más largas (entre 380-430 aminoácidos) que las secuencias obtenidas en KEGG se hace imposible entregar el dataset completo en formato .fasta a la herramienta BLASTP ya que la capacidad de procesamiento de la plataforma en su versión gratuita se hace insuficiente. Es por esta razón que se deben separar las secuencias en grupos de 5 por cada archivo .fasta. En la Figura 19 se muestra un diagrama de flujo para el script que cumple esta tarea de separación.

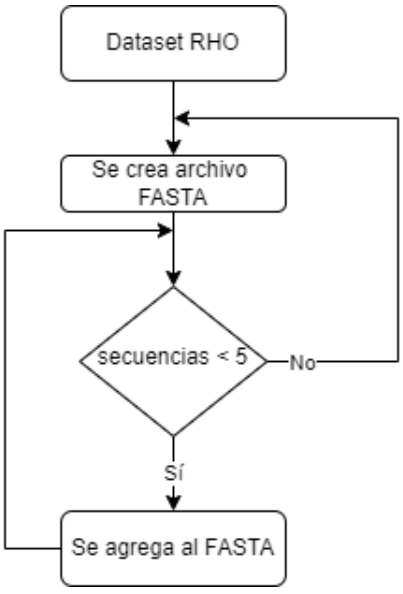


Figura 19: [Elaboración propia, 2024.] Diagrama de flujo para separar archivos el dataset RHOBase en .fasta.

En total se formaron 35 archivos .fasta para ser subidos a la plataforma de BLAST.

[14]Repositorio para COG4638 `https://www.ncbi.nlm.nih.gov/research/cog/cog/COG4638/`

2. **Alineación múltiple con BLAST:** Este proceso se asemeja mucho a lo realizado con KEGG en la subsección anterior, por lo que la configuración previa de la herramienta se considera exactamente igual. Lo que diferencia esta alienación de la anterior es el proceso posterior, ya que conversando con el cliente a partir de la experiencia anterior con BLASTP se decide cambiar parámetros de porcentaje de similitud y hacer uso del filtrado por *"Query Coverage"*. La configuración queda de la siguiente manera.

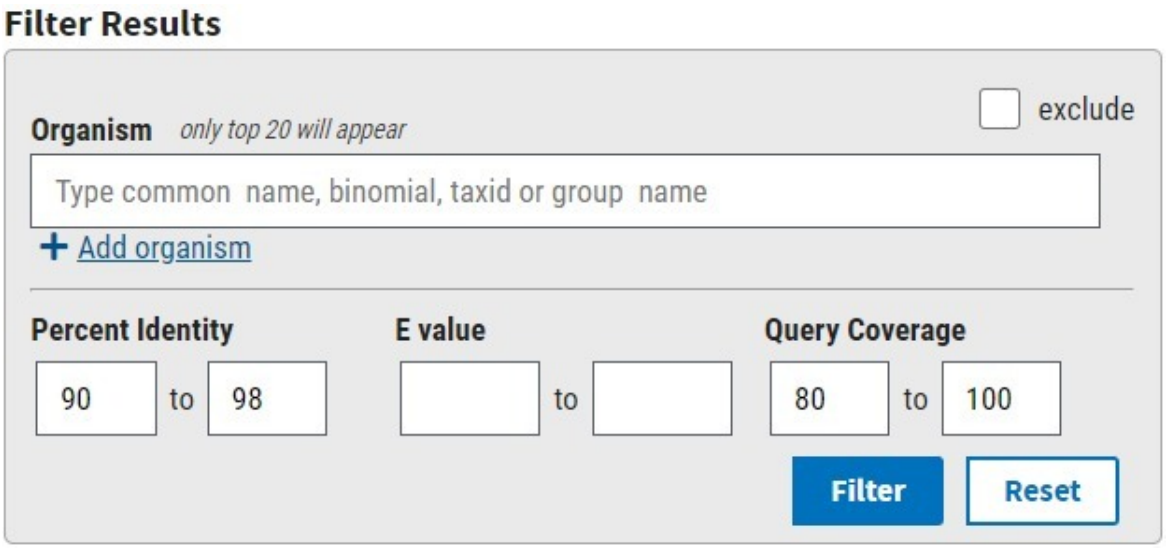


Figura 20: [Elaboración propia, 2024.] Filtros para datos encontrados en BLASTP de RHO.

La razón de los valores del porcentaje de similitud (90 %-98 %) es debido a que las secuencias que son entre un 99 % y 100 % similares solo variaciones de entre 1 o 3 caracteres y en los peores casos es exactamente la misma secuencia con un nombre distinto. Por supuesto, la similitud no puede ser baja ya que se necesita mantener la calidad de los datos que hay guardados hasta el momento, es por esto que se llega al acuerdo de usar un 90 % de similitud como mínimo en un primer filtrado, si es que este filtro resultara en pocas secuencias (entre 1 y 5) es posible relajar el filtro hasta un piso de 80 % para obtener al menos unas 20 secuencias extras, si aún así el el número de enzimas extras sigue siendo bajo entonces solo se guardan las encontradas.

Por otra parte el filtrado por cobertura de la consulta tiene la función de comprar los largos de las secuencias comparadas, por ende dejar los valores entre 80 % y 100 % quiere decir que solo se enlisten las secuencias encontradas que tengan al menos un 80 % de del largo de la secuencia de entrada. Este filtrado se decide agregar debido a que en alineaciones anteriores se descargaron varios datos de enzimas cuyas secuencias eran una serie de caracteres de largo no mayor a 100, esto por supuesto no es lo óptimo ya que por muy similar que sea con la secuencia de entrada no es totalmente representativo en cuanto a homología y función enzimática.

3. **Revisión y limpieza del dataset aumentado:** Una vez realizado el proceso de aumento

de datos se procede formar un nuevo dataset con los archivo .fasta obtenidos desde BLAST, haciendo uso del script representado en la Figura 8 se obtiene un nuevo archivo .csv con aproximadamente 5.000 entradas. Por supuesto, no es posible hacer uso de toda la información descargada ya que debe ser validada por los expertos en el área, además para una mayor validación de la calidad de los datos se hace uso nuevamente de la herramienta DeepNOG para revisar la clasificación obtenida para este nuevo dataset.

Los resultados obtenidos por DeepNOG son alentadores ya que en su gran mayoría, las secuencias obtenidas por BLAST mostraban que un 98.6 % de estas correspondían al COG4638 (grupo COG de interés para formar el dataset para el clasificador de función enzimática) y en total solo 34 enzimas descargadas son de grupos ajenos a COG4638. Además de lo anterior, muchas enzimas se repetían en más de un oportunidad por lo que se crea un pequeño script que borre las secuencias repetidas. Todo este procesado y limpieza de datos realizados tanto por la parte técnica como por la parte biológica deja como resultado un dataset completamente curado y validado con 3789 entradas en donde se detallan las siguientes propiedades.

- Número de acceso a NCBI.
- Nombre de la enzima.
- Gen asociado a la secuencia.
- Nombre del organismo en el que fue encontrada.
- Sustrato primario que degrada.
- Sustratos secundarios registrados (de existir evidencia).
- Publicación que respalda la información registrada.
- Secuencia aminoacídica de la enzima.

2.3.5. Creación de dataset para Clasificador Binario

El objetivo principal del clasificador binario, como se verá en el capítulo 2.3.7, es saber a priori si una secuencia es degradadora de contaminantes aromáticos antes de dar paso al clasificador de función enzimática, de esta forma no se procesan datos en vano. A raíz de la simplicidad de este módulo el entrenamiento de la red y la creación del dataset también no es trivial.

En primer lugar se deben juntar todos los datos que durante este proceso se respaldan como degradadores de contaminantes aromáticos, luego etiquetarlos y agregarlos a un nuevo dataset. Es importante destacar que todos los datos agregados como degradadores fueron validados por el equipo de biodegradación, dejando de lado incluso datos obtenidos desde

BLAST que no pudieron ser respaldados con una publicación científica o simplemente validados a través de análisis de su secuencia. Los dataset de degradadores agregados son los siguientes.

- RHOBase Aumentado (versión 1) y Subdataset RHOBase: El dataset RHOBase Aumentado en su primera versión aporta con 4.859 secuencias, solo considerando las pertenecientes a COG4638, además dos subdatasets de RHOBase generados por el equipo de bioquímica aporta con 173 y 34secuencias más. En total se agregan 5.067 secuencias.
- Subdataset KEGG: Este dataset corresponde al descargado desde la página web de KEGG y que fue validado por los expertos en la materia, no fueron usadas las secuencias descargadas desde BLAST ya que hasta ese momento no se habían validado. Se suman 206 secuencias nuevas.
- AromaDEG: Se agregaron todas las secuencias descargadas desde la base de datos de AromaDEG. En total suman 3.335 secuencias aminoacídicas nuevas.

Luego de concatenar cada uno de los datasets anteriores es necesario limpiar el conjunto de datos final (que cuenta con 8574 entradas) ya que puede existir duplicación en los datos y eso sería fatal para construir un buen clasificador. Por lo tanto, una vez revisado y limpiado el dataset final queda con un total de 5.916 secuencias en donde la más corta tiene un largo de 351 caracteres y la más larga con 455.

Obviamente el dataset debe presentar un buen balance entre enzimas degradadoras y no degradadoras, es por esto que se buscaron enzimas que tengan poca relación con la biodegradación ya que a este primer módulo de clasificación llegará un genoma completo de algún microorganismo en donde las secuencias pueden cumplir con cualquier función enzimática. En esta primera instancia de formación del dataset el cliente recomendó usar principalmente 2 fuentes de datos para enzimas no degradadoras.

- **BacMet**: Es una base de datos curada a mano con evidencia experimental sobre enzimas de biosidas antibacterianos y genes de resistencia a metales[26], cabe señalar que la secuencia más corta tiene solamente 156 caracteres y la más larga 306. El dataset descargado corresponde a *"BacMet2 Experimentally confirmed resistence genes"* en su versión 2 (11 Marzo de 2018). [15]
- **MiBig**: Base de datos curada por el equipo de MiBIG donde se almacena información de genes y enzimas que participan en rutas de biosíntesis[27]. Este dataset contiene 33.173 secuencias [16] de tamaños variados. La información de enzimas que cumplan

[15] Repositorio de BacMet `http://bacmet.biomedicine.gu.se/download_temporary.html`

[16] Archivo .fasta de MiBIG `https://dl.secondarymetabolites.org/mibig/mibig_prot_seqs_2.0.fasta`

una función de biosíntesis es de mucha importancia para el clasificador binario debido a que las secuencias de biodegradación y de biosíntesis comparten similitudes que pueden llegar provocar una mala predicción.

Todas las secuencias agregadas desde estas dos bases de datos fueron etiquetadas como no degradadores así que en total el dataset final queda compuesto por 39.842 secuencias, a las cuales se les aplica una limpieza similar a la anterior para eliminar secuencias repetidas e inválidas, resultando en un dataset de 38.011 filas. Hasta aquí ya se cuenta con un dataset repleto de información pero completamente desbalanceado, además que hay muchos datos inservibles para el clasificador. En este punto se propone la construcción de múltiples datasets cada uno con características distintas para encontrar el que entregue los mejores resultados a la hora de entrenar el clasificador binario y así también aprovechar de validar el trabajo realizado en la confección de la base de datos RHOBase, luego el cliente propuso dejar secuencias de largos entre 150 y 460 aminoácidos, ya que la mayoría de las secuencias del dataset para el clasificador de función enzimática mantienen estos largos, en consecuencia se crean los siguientes datasets.

1. **Sin criterio (balanceado):** Este dataset contiene todos los datos catalogados como degradadores de contaminantes aromáticos y se le anexaron suficientes no degradadores hasta dejarlo completamente balanceado, para esta colección no se consideraron los criterios propuestos. En total contiene 11.832 muestras.

2. **Sin criterio (desbalanceado):** Este dataset contiene las mismas secuancias degradadores que el primero pero está completamente desbalanceado, es decir se anexaron todas las secuencias no degradadoras. En total hay 38.013 muestras, de las cuales 5.916 son secuencias de degradadores y 32.097 son no degradadores.

3. **Cadenas de aminoácidos de un largo mayor a 350 y menor a 460 caracteres:** Dentro de la base de datos generada solo hay 3.139 secuencias que cumplen este criterio por lo que se genera un subdataset balanceado de 6.278 entradas, lamentablemente para este subdataset no entran secuencias de BacMet debido a su largo.

4. **Cadenas de aminoácidos de un largo mayor a 150 y menor a 460 caracteres:** Para este criterio existen 5.888 secuencias degradadoras por lo que se forman los siguientes subdatasets.

    a) **Dataset tamaño grande:** Subdataset hecho con la totalidad de las secuencias disponibles y perfectamente balanceado, es decir, 10.714 entradas

    b) **Dataset tamaño mediano:** Subdataset hecho con la misma cantidad de datos que el subdataset con secuencias de largo mayor a 350 caracteres, esto se hace con el fin de poder comparar si existe una mejora real al limitar el tamaño de la secuencia

c) **Dataset sin RHOBase:** Este subdataset se genera sin la Base de datos creada por el equipo para comprobar si es que realmente es un aporte dentro del contexto de identificación de degradadores aromáticos, el largo de este subdataset es de 6.508 secuencias donde la mitad corresponden a degradadoras de contaminantes aromáticos

2.3.6. Embedding: Vectorización de la secuencia

Como se ve en la Figura anterior, este submódulo es el primer paso antes de comenzar con cualquier proceso de clasificación, aquí la secuencia pasa de ser una serie de caracteres a un vector numérico con el cual es más facil trabajar para el los clasificadores. Para llevar a cabo este proceso de vectorización se hace uso de una librería de Python desarrollada justamente con el propósito de vectorizar una secuencia para eventualmente procesar dichos vectores y obtener una salida desde el clasificador. La librería en cuestión es **doc2vec**[17]. Para hacer el entrenamiento de este modelo se hace uso de un dataset proveniente de Swiss-Prot [28], el cual contiene secuencias de varios tipos de enzimas evitando abarcar solo las degradadoras, ya que es importante que en primera instancia el clasificador entienda el contexto dentro de una secuencia, cualquiera sea esta.
En palabras simples el proceso es el siguiente, se entrena un modelo K-means siendo K=4, el modelo forma palabras desde la secuencia donde cada palabra corresponde a 4 aminoácidos adyacentes. Al tener formadas todas las palabras de la secuencia se hace uso de una *"ventana deslizante"* de ancho 5, para que las palabras se solapen entre sí y así se logre una mayor comprensión del dominio de la secuencia.
Para efectos prácticos y ejemplificar un poco como es que se vectoriza una secuencia se tiene la siguiente figura.

[17]Documentación oficial `https://radimrehurek.com/gensim/models/doc2vec.html`

```
In [179]: X_train
```

Out[179]:

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57
1817	-0.171886	-0.052814	0.083427	-0.174908	-0.083743	-0.239680	-0.043483	0.018934	-0.070752	-0.010858	...	-0.008860	0.069377	0.133097	-0.193222
2233	-0.001944	0.041636	-0.158793	-0.064647	-0.065224	0.127920	0.069829	-0.013468	-0.062151	-0.094494	...	-0.107781	0.061972	0.115510	0.156529
733	0.029623	0.097839	0.047237	0.043758	-0.009071	-0.239638	-0.015736	-0.043563	-0.019240	-0.075629	...	-0.060820	-0.076081	0.095143	0.133136
317	0.339756	-0.086443	-0.253591	0.157971	0.189678	0.054730	0.021137	-0.018196	-0.047885	-0.037261	...	-0.048827	0.047821	0.262862	0.079975
1580	0.211969	-0.184777	0.060186	0.046756	0.027398	-0.008095	-0.160981	-0.242073	-0.185327	-0.152641	...	-0.122142	0.286872	0.133730	-0.035802
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3606	0.059169	0.070429	0.015313	-0.217242	0.085746	0.157179	0.002468	-0.008461	-0.071862	0.011332	...	0.023189	0.294957	-0.236145	-0.126932
1608	0.129106	0.021288	-0.020403	0.073328	0.105104	-0.342510	0.091669	0.007784	0.106085	-0.009628	...	-0.115182	0.167939	-0.098125	-0.082191
2541	-0.018352	-0.020498	-0.219564	0.198986	-0.059924	-0.042698	0.043772	0.028017	-0.340233	0.036369	...	-0.136021	-0.000333	-0.089491	0.137707
2575	0.187304	0.024290	-0.105813	0.140855	0.000502	0.035532	0.119320	0.091413	-0.028541	0.051706	...	0.114297	-0.187667	-0.096851	-0.001171
3240	-0.046377	-0.176783	-0.026975	0.269031	0.177330	-0.121703	0.094578	-0.125568	-0.171323	0.193480	...	-0.087903	-0.038870	0.046016	-0.139264

2841 rows × 64 columns

Figura 21: [Elaboración propia, 2024.] Resultados de una vectorización, cada fila representa una secuencia de una bacteria y cada columna corresponde al embedding de la misma.

2.3.7. Clasificador Binario

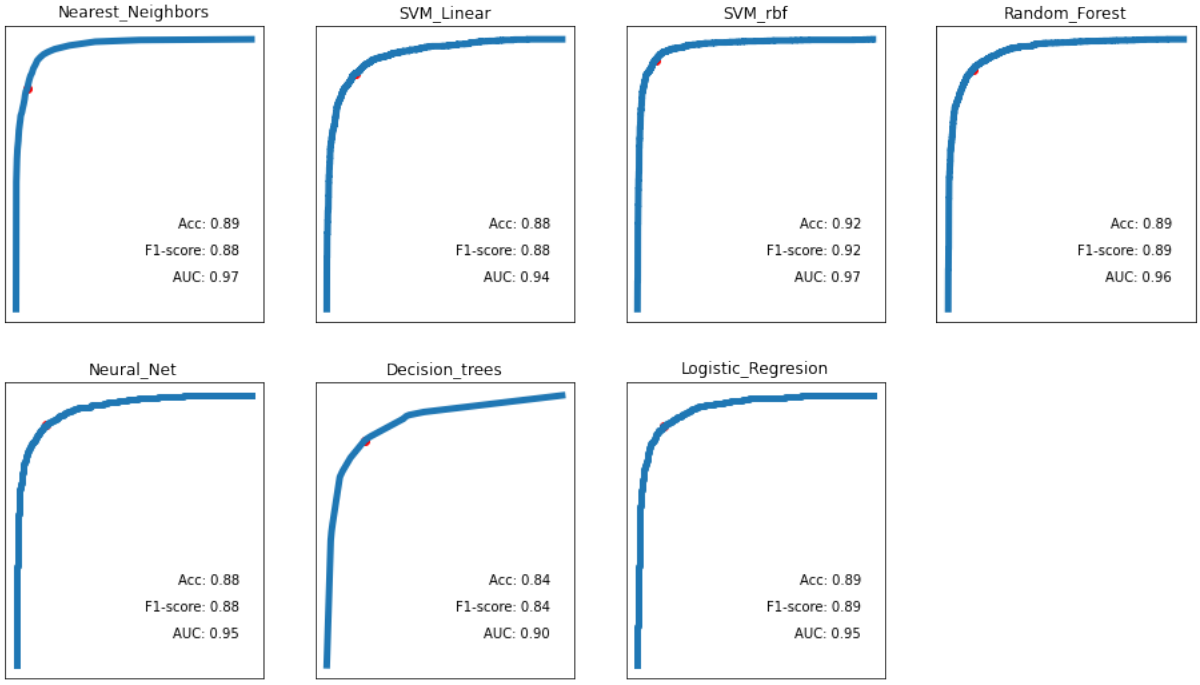
Para esta etapa del módulo se debe implementar un clasificador que pueda reconocer una secuencia degradadora de contaminantes aromáticos. En el caso que la secuencia no corresponda a un degradador entonces se debe pasar por alto y etiquetarla inmediatamente como **NO** degradadora.

En primera instancia se implementa una serie de modelos para ser comparados con el objetivo de elegir el que presente las mejores métricas. Los modelos a evaluar son:

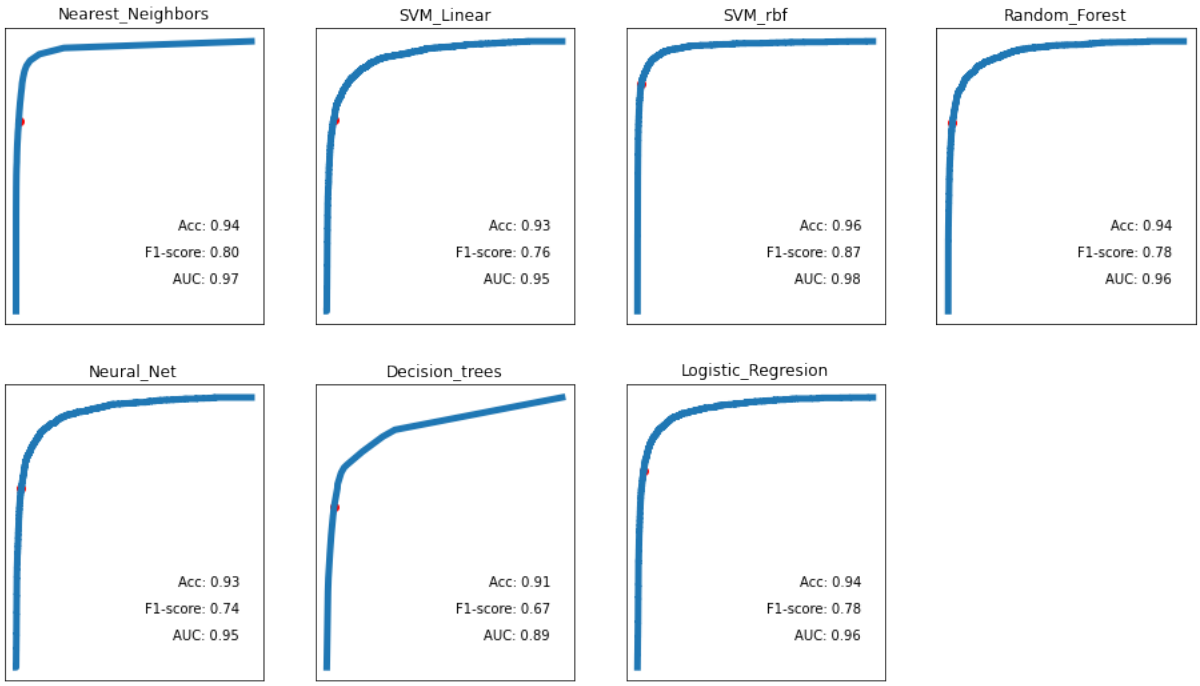
- **Nearest Neighbors**
- **SVM (kernel Lineal)**
- **SVM (kernel RBF)**
- **Random Forest**
- **Red Neuronal**
- **Decision Trees**
- **Regresión Logística**

Cada uno de estos modelos fueron entrenados con los distintos datasets que se generaron en la sección anterior y usando la vectorización de secuencia descrita en la subsección previa, de esta forma se busca el par Modelo-Dataset que presente las mejores métricas. Así se obtuvo los siguientes resultados.

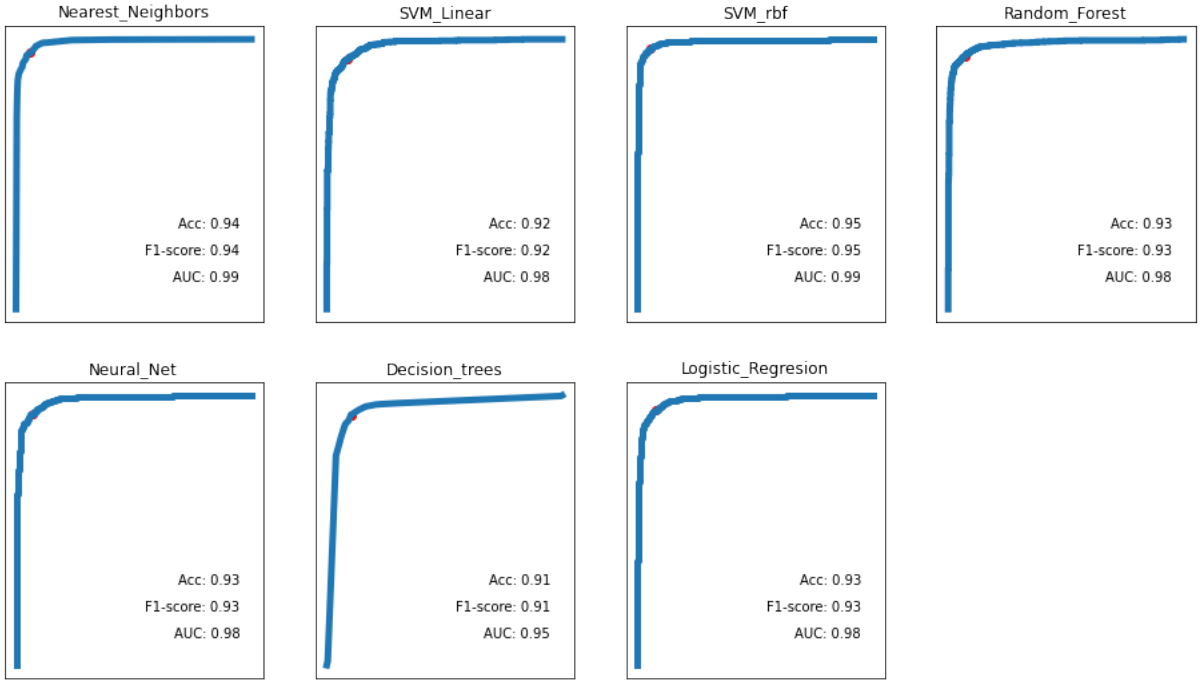
- **Dataset sin criterios y balanceado:**



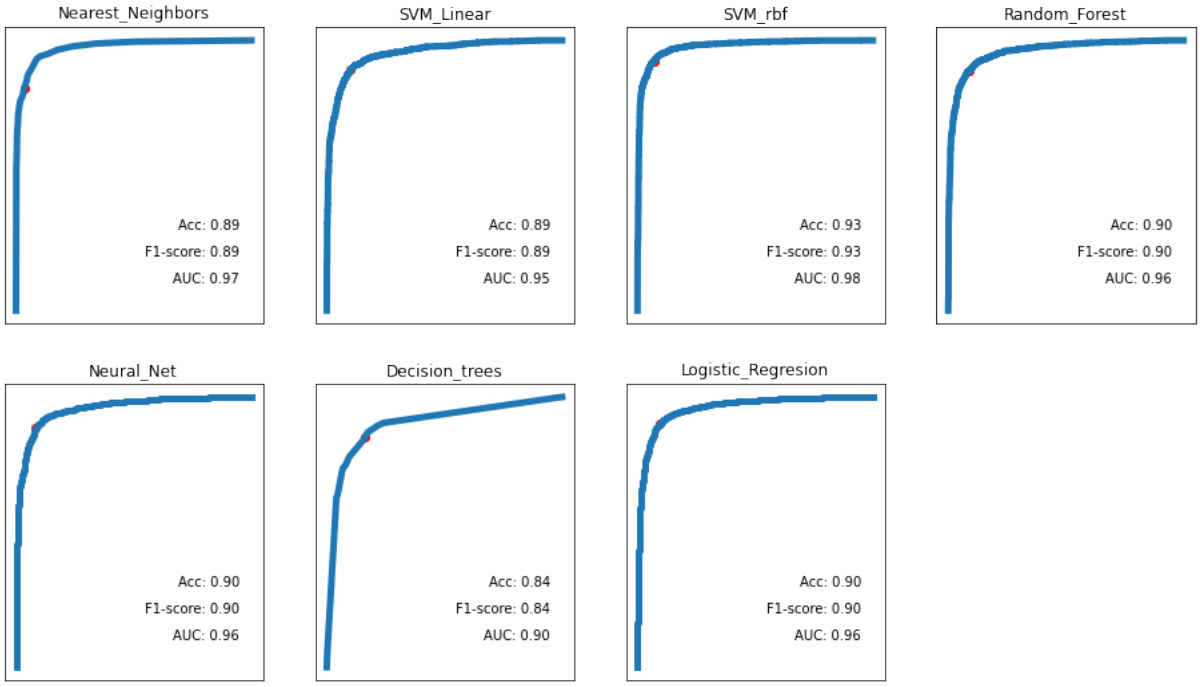
- **Dataset sin criterios y desbalanceado:**



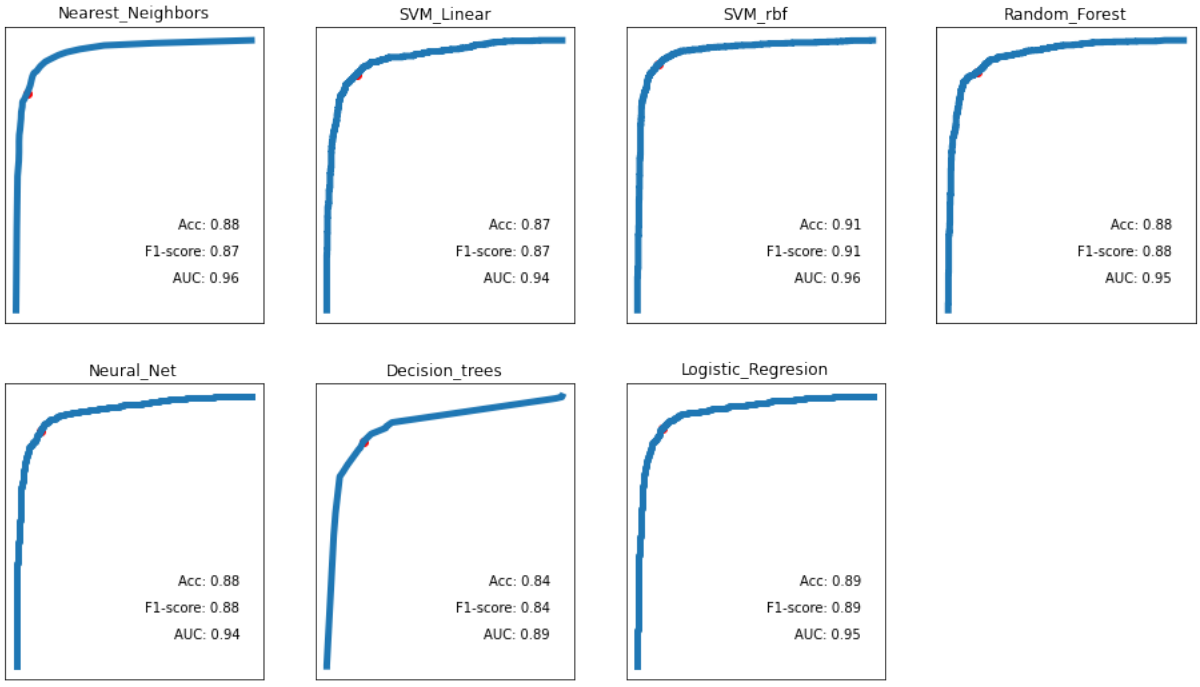
- **Dataset con cadenas de aminoácidos de un largo mayor a 350 y menor a 460 caracteres:**



- **Dataset con cadenas de aminoácidos de un largo mayor a 150 y menor a 460 caracteres (grande):**



- **Dataset con cadenas de aminoácidos de un largo mayor a 150 y menor a 460 caracteres (mediano):**



- **Dataset con cadenas de aminoácidos de un largo mayor a 350 y menor a 460 carac-**

teres (sin RHOBase):

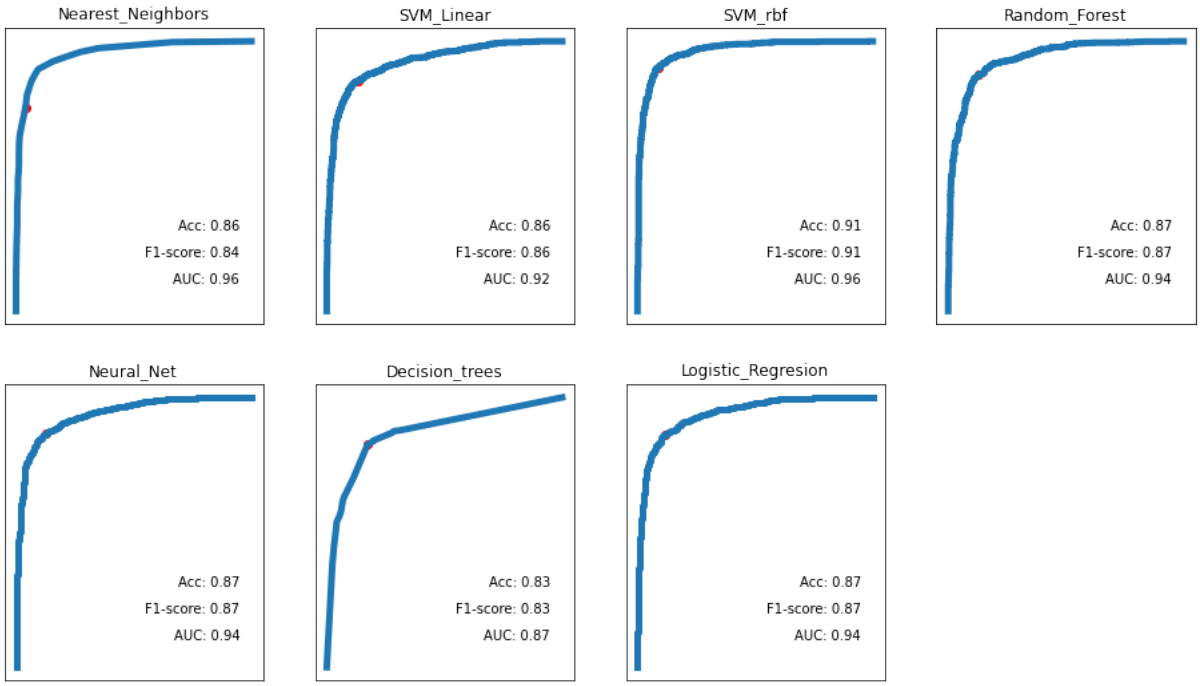


Figura 22: Resultados al evaluar los distintos modelos usando el dataset con el criterio de largo entre 350 y 460 caracteres

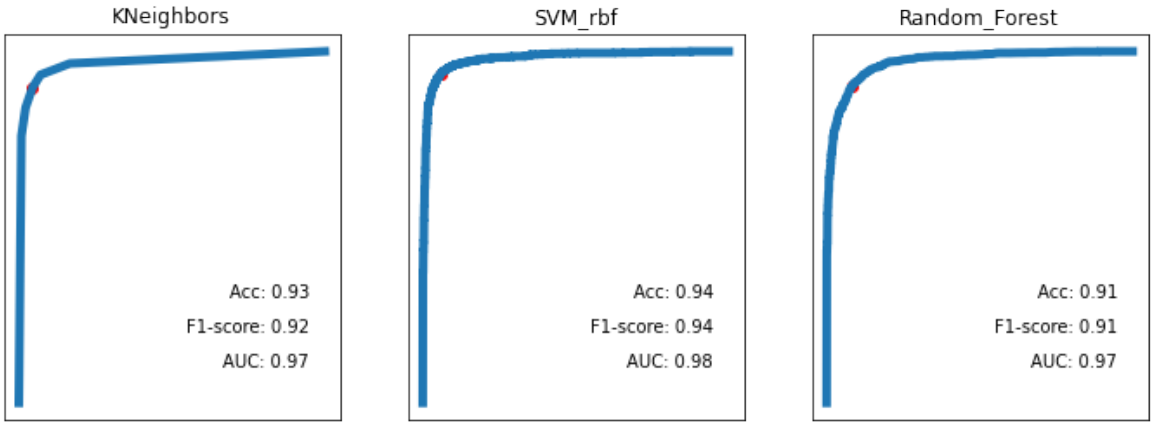
Los resultados anteriores muestran las métricas obtenidas para cada caso, para la mayoría de los datasets se destacan los métodos de **Nearest Neighbors**, **Support Vector Machine con kernel RBF** y **Random Forest**. El siguiente paso es usar *GridSearch* para encontrar las mejores métricas posibles para cada modelo aplicándolo a los 3 mejores resultados de cada caso. Los parámetros usados para cada optimización en GridSearch son los siguientes:

- **Support Vector Machine (SVM)**
  - Parámetro *C*: [0.0001 - 10000], avanzando en 1 orden de magnitud para valores <0 y avanzando en 50 unidades para valores [0 - 1000] y luego avanzando en 500 unidades para valores [1000 - 10000]
  - Parámetro *gamma*: [1 - 1e-16], retrocediendo en medio orden de magnitud
  - Kernel: Radial Basis Function (RBF)
- **Random Forest**
  - Parámetro *Max Depth*: [5 - 100], avanzando de 5 en 5 unidades
  - Parámetro *N° Estimators*: [5 - 300], avanzando de 5 en 5 unidades
- **KNeighbors**
  - Parámetro *N° Neighbors*: [5 - 100], avanzando de 5 en 5 unidades

Finalmente las versiones optimizadas de los modelos para cada dataset son las siguientes:

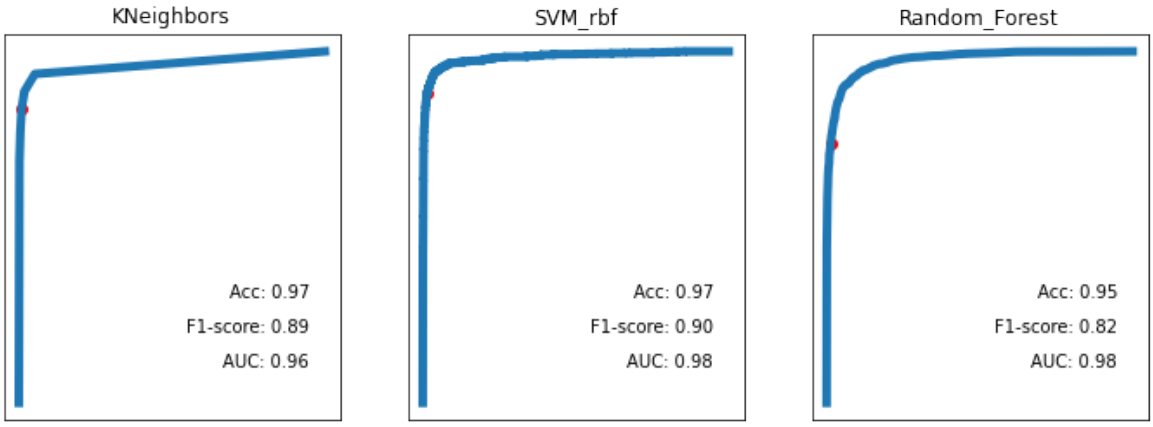
- **Dataset sin criterios y balanceado:**

  - **KNeighbors**
    - Parámetro *N° Neighbors*: 5
  - **Support Vector Machine (SVM)**
    - Parámetro *C*: 50
    - Parámetro *gamma*: 1
    - Kernel: Radial Basis Function (RBF)
  - **Random Forest**
    - Parámetro *Max Depth*: 40
    - Parámetro *N° Estimators*: 160



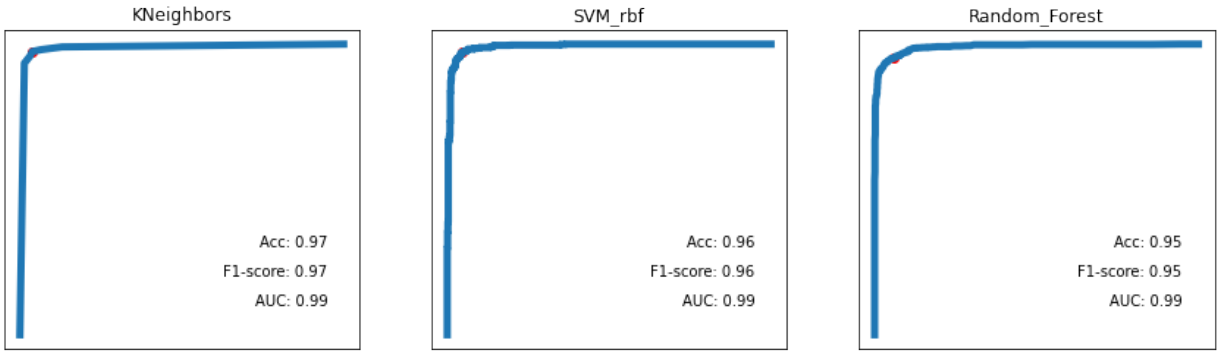
- **Dataset sin criterios y desbalanceado:**

  - **KNeighbors**
    - Parámetro *N° Neighbors*: 5
  - **Support Vector Machine (SVM)**
    - Parámetro *C*: 50
    - Parámetro *gamma*: 1
    - Kernel: Radial Basis Function (RBF)
  - **Random Forest**
    - Parámetro *Max Depth*: 55
    - Parámetro *N° Estimators*: 195



- **Dataset con cadenas de aminoácidos de un largo mayor a 350 y menor a 460 caracteres:**

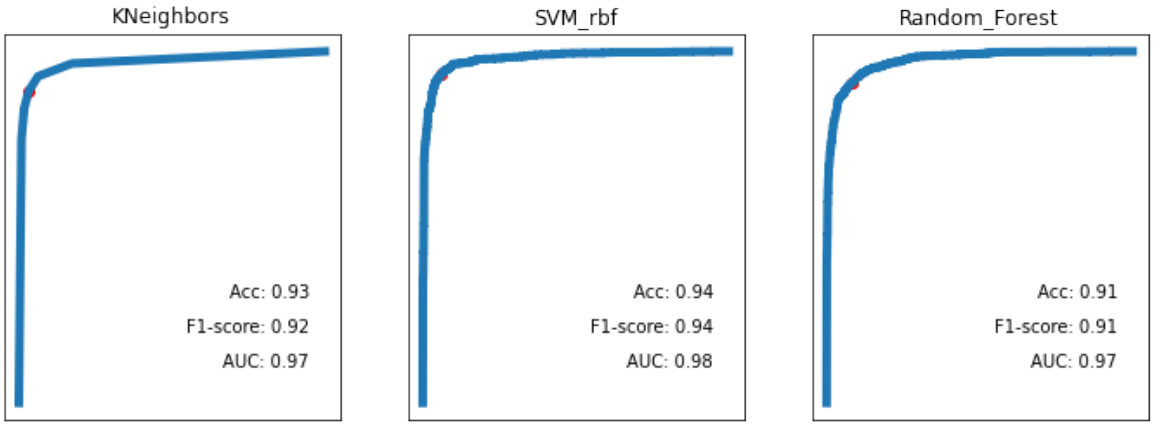
  - **KNeighbors**
    - Parámetro *N° Neighbors*: 5
  - **Support Vector Machine (SVM)**
    - Parámetro *C*: 50
    - Parámetro *gamma*: 1
    - Kernel: Radial Basis Function (RBF)
  - **Random Forest**
    - Parámetro *Max Depth*: 80
    - Parámetro *N° Estimators*: 155



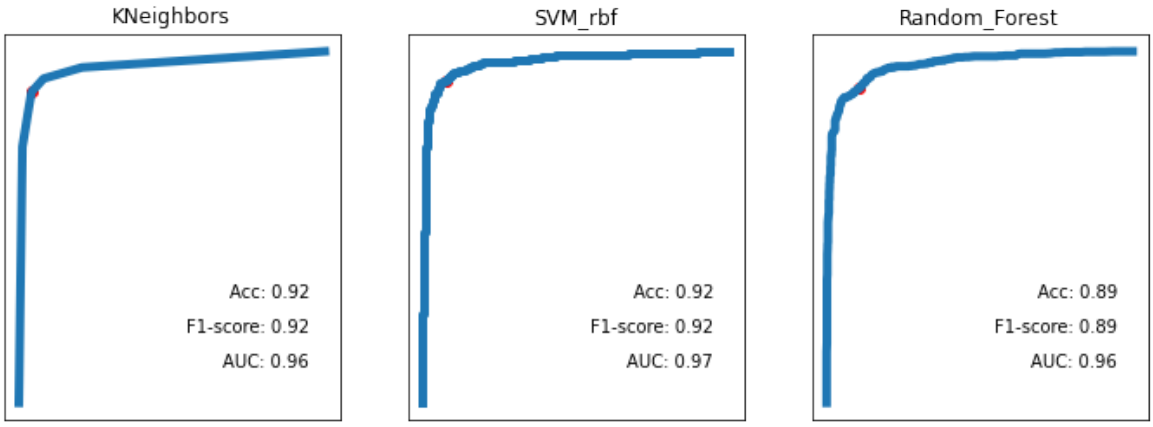
- **Dataset con cadenas de aminoácidos de un largo mayor a 150 y menor a 460 caracteres (grande):**

  - **KNeighbors**
    - Parámetro *N° Neighbors*: 5

- **Support Vector Machine (SVM)**
  - Parámetro *C*: 50
  - Parámetro *gamma*: 1
  - Kernel: Radial Basis Function (RBF)
- **Random Forest**
  - Parámetro *Max Depth*: 25
  - Parámetro *N° Estimators*: 210

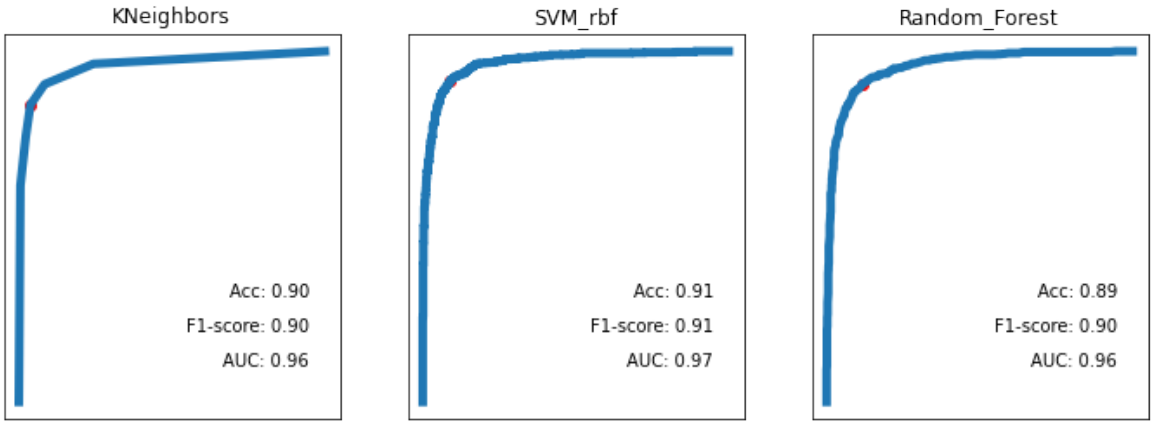


- **Dataset con cadenas de aminoácidos de un largo mayor a 150 y menor a 460 caracteres (mediano):**
  - **KNeighbors**
    - Parámetro *N° Neighbors*: 5
  - **Support Vector Machine (SVM)**
    - Parámetro *C*: 50
    - Parámetro *gamma*: 1
    - Kernel: Radial Basis Function (RBF)
  - **Random Forest**
    - Parámetro *Max Depth*: 45
    - Parámetro *N° Estimators*: 280



- **Dataset con cadenas de aminoácidos de un largo mayor a 350 y menor a 460 caracteres (sin RHOBase):**

  - **KNeighbors**
    - Parámetro *N° Neighbors*: 5
  - **Support Vector Machine (SVM)**
    - Parámetro *C*: 50
    - Parámetro *gamma*: 1
    - Kernel: Radial Basis Function (RBF)
  - **Random Forest**
    - Parámetro *Max Depth*: 75
    - Parámetro *N° Estimators*: 295



Analizando los resultados, el primer punto importante a mencionar es que en los parámetros óptimos para cada caso siempre varían para el método de Random Forest, mientras que para

KNeighbors y SVM se mantuvieron constantes en todos los escenarios (N° Neighbors = 5 y C = 5, Gamma = 1, respectivamente). En segundo lugar también se aprecia que tanto KNeighbors como SVM siempre mantuvieron los más altos números en cuanto a F1-Score y Accuracy y lo común es que el segundo de ellos es el que entrega mejores resultados. En tercer lugar la mejor curva ROC fue utilizando el Dataset con cadenas de aminoácidos de un largo mayor a 350 y menos a 460 caracteres, este resultado es esperable ya que son los largos de secuencia de las enzimas del dataset curado por el equipo de biodegradación. En concreto el clasificador que mejor se comporta es **KNeighbors** seguido muy de cerca por SVM y solo un poco más abajo Random Forest, ahora bien este resultado si bien es excelente no entrega garantías de que sirva para toda secuencia ya que dentro de un genoma pueden existir otras secuencias con tamaños variables asociadas a degradación, con contextos genómicos distintos o que representen enzimas que no sean oxigenasas del dataset curado.

CAPÍTULO 3
VALIDACIÓN DE LA SOLUCIÓN

Para validar el trabajo realizado en la sección anterior se seleccionan dos genomas de microorganismos conocidos, el primero es el de la bacteria modelo *Escherichia Coli* y el segundo es de la bacteria *Paraburkholderia xenovorans* LB400, una cepa bacteriana modelo para la degradación de policlorobifenilos y otros compuestos aromáticos. En el caso de *E. coli*, modelo de fisiología bacteriana, las funciones enzimáticas de cada proteína codificada en su genoma han sido altamente estudiadas, por lo que ya se conocen la mayoría de las secuencias que pueden clasificarse como degradadoras de contaminantes aromáticos. En el caso de *P. xenovorans* LB400, muchas rutas y enzimas asociadas a degradación de compuestos aromáticos han sido caracterizadas, por lo que sabemos a priori la clasificación de muchas de sus enzimas. De esta manera los resultados obtenidos de cada clasificador podrán ser validados rápidamente viendo la salida que generan. Utilizando las versiones optimizadas de los modelos entrenados para cada tipo de dataset se obtienen los siguientes resultados.

3.1. *Escherichia coli*, modelo general de fisiología bacteriana

Este genoma consta de un archivo fasta que contiene 4.284 secuencias de largos variables (varían entre secuencias con 20 caracteres de largo y otras con secuencias de más de 600 caracteres) por lo que esto puede seer un problema para la clasificación. Después de correr cada uno de los clasificadores con el genoma propuesto se obtuvieron varios ficheros .csv con las secuencias clasificadas como degradadoras, estas cantidades están reflejadas en la tabla 3.1.

Analizando los resultados se puede notar que el intervalo de la cantidad de secuencias degradadoras obtenidas por cada uno de los clasificadores es bastante acotado, no superando las 90 secuencias de diferencia en la mayoría de los casos. Lamentablemente, al analizar en profundidad todas las secuencias clasificadas como degradadoras no se obtuvieron los resultados esperados, ya que en la mayoría de los casos la predicción entrega falsos positivos, más allá de que acierta en algunas secuencias, la mayoría de las predicciones son erróneas. *Escherichia coli* es una bacteria comúnmente encontrada en el microbioma intestinal no especializada en la degradación de compuestos aromáticos, que en su genoma solo posee una enzima clasificada como RHO asociada a degradación. El uso de un clasificador binario especialmente entrenado con secuencias curadas de RHO podría llevar al aumento de falsos positivos. De igual manera al hacer una intersección entre todas las salidas de cada uno de los clasificadores se obtuvo una lista con 25 secuencias positivas.

Dataset	KNeighbors Nº degradadoras	SVM (RBF) Nº degradadoras	Random Forest Nº degradadoras
Sin criterio (balanceado)	209	296	269
Sin criterio (desbalanceado)	82	89	69
Con criterio $350 \leq secuencia \leq 460$	251	272	227
Grande con criterio $150 \leq secuencia \leq 460$	270	278	276
Mediano con criterio $150 \leq secuencia \leq 460$	229	313	294
Sin RHOBase con criterio $350 \leq secuencia \leq 460$	265	335	324

Tabla 3: Resumen de resultados para bacteria Escherichia coli.

3.2. *Paraburkholderia xenovorans* LB400, modelo de degradación de compuestos aromáticos

Para complementar la validación realizada anteriormente con *Escherichia coli*, se opta por probar uno de los clasificadores anteriores con el genoma de *Paraburkholderia xenovorans* LB400, modelo de degradación de compuestos aromáticos aislada desde un vertedero contaminado con policlorobifenilos (PCBs) que efectivamente posee varias RHO codificadas en su genoma.

Para realizar esta clasificación se escoge el clasificador entrenado el dataset formado a partir de secuencias de largo entre 350 y 460 caracteres y que a su vez usa el método de Support Vector Machine con kernel RBF, debido a que el genoma en su mayoría tiene secuencias cercana a estos largos y a que este clasificador fue el que obtuvo las métricas más altas. Al finalizar la clasificación se obtuvieron un total de 538 secuencias clasificadas como degradadoras de un total de 8631. Por supuesto se deben validar los resultados obtenidos a través de la revisión de un experto en el área, ya que este genoma en particular no está tan detalladamente estudiado a diferencia de *E. coli*.

Finalmente, los resultados obtenidos confirman lo observado en *E. coli*, donde se es posible corroborar que 70 (13.01 %) de ellas han sido asociadas a rutas de degradación de contaminantes aromáticos en estudios previos, donde en su mayoría se tratan de oxigenasas. Cabe mencionar que la cepa LB400 no se encuentra completamente estudiada, por lo que varias de las enzimas no corroboradas (468) podrían estar involucradas en degradación de compuestos aromáticos. El equipo experto informa que en los casos de un grupo de más de 5 enzimas diferentes asociadas a la degradación de un mismo compuesto, siempre se identificó las proteínas asociadas a oxigenesas hidroxilantes de anillo aromático, confirmando el

sesgo que existe de este tipo de enzimas en el clasificador binario.

Al comparar los resultados obtenidos en la clasificación de los genomas anteriores se muestran las capacidades y limitaciones de los métodos computacionales en la identificación de secuencias genómicas asociadas a la degradación de contaminantes aromáticos. En el caso de E. coli, un modelo general de fisiología bacteriana, se evidencia que, a pesar de utilizar múltiples enfoques de clasificación, los resultados mostraron un alto número de falsos positivos, reflejando la necesidad de afinar los criterios de clasificación y validación, especialmente en organismos no especializados en la degradación de estos compuestos. Por otro lado, el análisis en P. xenovorans LB400, que naturalmente participa en la degradación de contaminantes aromáticos, mostró un mayor éxito en la clasificación con una proporción significativa de secuencias identificadas correctamente, aunque aún con un considerable número de secuencias por validar. Esto denota la importancia de la selección de organismo y las características específicas del genoma en el éxito de las herramientas bioinformáticas. En conclusión, mientras que estos clasificadores ofrecen una valiosa primera aproximación para identificar potenciales enzimas degradadoras en genomas complejos, también resaltan la necesidad de integrar estos métodos con análisis experimentales y conocimiento experto para su validación y mejora continua.

CAPÍTULO 4
CONCLUSIONES

En resumen, este estudio presentó un enfoque de clasificación binaria para identificar enzimas implicadas en la degradación de contaminantes aromáticos a partir de secuencias genómicas completas. Utilizando algoritmos como KNeighbors, SVM con kernel RBF y Random Forest, se desarrollaron clasificadores basados en varios datasets con distinto balance y rangos de longitud de secuencias. Los resultados indicaron una precisión y puntuación F1 prometedores en los entornos de prueba, con SVM con kernel RBF mostrando un rendimiento superior en varios escenarios.

Sin embargo, la aplicación práctica de estos clasificadores en genomas reales de *Escherichia coli* y *Paraburkholderia xenovorans* LB400 reveló un sesgo en la identificación de estas enzimas, además de una cantidad significativa de falsos positivos en las predicciones de función enzimática, lo que destaca la dificultad de clasificar funciones biológicas complejas a partir de secuencias genéticas. A pesar de los esfuerzos de optimización y validación en el desarrollo de datasets curados, las limitaciones actuales enfatizan la necesidad de enriquecer los datasets con otras familias de enzimas, mejorar el feature engineering y considerar técnicas avanzadas de aprendizaje automático para mejorar la especificidad y la generalización de los modelos de clasificación en bioinformática.

Este estudio subraya la importancia de una validación exhaustiva de los modelos de aprendizaje automático en bioinformática y proporciona una base sólida para futuras investigaciones que podrían expandir el conocimiento y la aplicación de herramientas computacionales en la biodegradación de contaminantes, clasificación de función enzimática, y predicción de capacidades metabólicas en base a genomas completos. Los pasos futuros incluirán la incorporación de características más sofisticadas, el enriquecimiento de los datos de entrenamiento y la exploración de modelos de aprendizaje profundo para abordar los desafíos encontrados y mejorar la fiabilidad de las predicciones de funciones enzimáticas.

REFERENCIAS BIBLIOGRÁFICAS

[1] Glowka, L. (1996). Guía del convenio sobre la diversidad biológica (No. 30). Iucn.

[2] Castro Varela, G. (2007). Informe final diseño monitoreo frente derrames de hidrocarburos. Quillota, Bogotá.

[3] Toxic air: The price of fossil fuels - es | greenpeace españa. (n.d.). Retrieved December 1, 2021, from https://es.greenpeace.org/es/wp-content/uploads/sites/3/2020/02/TOXIC-AIR-Report-110220.pdf.

[4] Taubs, G. Sense from sequences: Stephen F. Altschul on bettering BLAST. Science Watch 11, 3–4 (2000)

[5] Verma, S., Kour, S., & Pathak, R. K. (2022). In Silico Approaches in Bioremediation Research and Advancements. In Bioremediation of Environmental Pollutants (pp. 221-238). Springer, Cham.

[6] L. Betancor, M. Gadea, K. Flores. (2008). Genética bacteriana. Oct, 2021, de INSTITUTO DE HIGIENE UNIVERSIDAD DE LA REPÚBLICA DE URUGUAY Sitio web: http://www.higiene.edu.uy/cefa/2008/GeneticaBacteriana.pdf

[7] Lima, Richardson. Sistema multiagente para anotação manual em projetos de seqüenciamento de genomas. https://www.researchgate.net/publication/41009654_Sistema_multiagente_para_anotacao_manual_em_projetos_de_sequenciamento_de_genomas

[8] antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. Nucleic acids research. 39. W339-46. 10.1093/nar/gkr466.

[9] National Center for Biotechnology Information. (2021, 26 abril). GenBank Overview. https://www.ncbi.nlm.nih.gov/genbank/. Recuperado 8 de diciembre de 2021, de https://www.ncbi.nlm.nih.gov/genbank/

[10] Saul B. Needleman & Christian D. Wunsch. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology, Volume 48, Issue 3.

[11] Smith, T.F. and Waterman, M.S. (1981) Identification of Common Molecular Subsequences. Journal of Molecular Biology, 147, 195-197. http://dx.doi.org/10.1016/0022-2836(81)90087-5

[12] Prunella N, Liuni S, Attimonelli M, Pesole G. FASTPAT: a fast and efficient algorithm for string searching in DNA sequences. Comput Appl Biosci. 1993 Oct;9(5):541-5. doi: 10.1093/bioinformatics/9.5.541. PMID: 8293327.

[13] Feldbauer, R., Gosch, L., Lüftinger, L., Hyden, P., Flexer, A., & Rattei, T. (2020). DeepNOG: fast and accurate protein orthologous group assignment. Bioinformatics, 36(22-23), 5304-5312.

[14] Seo, S., Oh, M., Park, Y., & Kim, S. (2018). DeepFam: deep learning based alignment-free method for protein family modeling and prediction. Bioinformatics, 34(13), i254-i262.

[15] Collins, J. F., & Coulson, A. F. Applications of parallel processing algorithms for DNA sequence analysis. Nucleic Acids Research 12, 181–192 (1984)

[16] Smith, T. F., & Waterman, M.S. Identification of common molecular subsequences. Journal of Molecular Biology 147, 195–197 (1981) doi:10.1016/0022-2836(81)90087-5

[17] Altschul, S. F., et al. Gapped Blast and PSI-Blast: A new generation of protein database search programs. Nucleic Acids Research 25, 3389–3402 (1997)

[18] Madden, T. The BLAST sequence analysis tool. In NCBI Handbook, ed. J. McEntyre and J. Ostell (National Library of Medicine, Bethesda, MD, 2005)

[19] Kanehisa M, Sato Y, Kawashima M. KEGG mapping tools for uncovering hidden features in biological data. Protein Science. 2021;1–7. https://doi.org/10.1002/ pro.4172

[20] McDonald AG, Tipton KF. Fifty-five years of enzyme classification: advances and difficulties. FEBS J. 2014;281:583–592.

[21] Chakraborty J, Jana T, Saha S & Dutta TK (2014). Ring-Hydroxylating Oxygenase database: a database of bacterial aromatic ring-hydroxylating oxygenases in the management of bioremediation and biocatalysis of aromatic compounds. Environ. Microbiol. Rep. 6(5):519-523. [PMID: 25646545]

[22] Márcia Duarte, Ruy Jauregui, Ramiro Vilchez-Vargas, Howard Junca, Dietmar H. Pieper, AromaDeg, a novel database for phylogenomics of aerobic bacterial degradation of aromatics, Database, Volume 2014, 2014, bau118, https://doi.org/10.1093/database/bau118

[23] The UniProt Consortium, UniProt: the universal protein knowledgebase in 2021, Nucleic Acids Research, Volume 49, Issue D1, 8 January 2021, Pages D480–D489, https://doi.org/10.1093/nar/gkaa1100

[24] The UniProt Consortium, UniProt: the universal protein knowledgebase in 2021, Nucleic Acids Research, Volume 49, Issue D1, 8 January 2021, Pages D480–D489, https://doi.org/10.1093/nar/gkaa1100

[25] J. Nielsen, Üsing paper prototypes in home-page design,ïn IEEE Software, vol. 12, no. 4, pp. 88-89, July 1995, doi: 10.1109/52.391840.

[26] Pal, C., Bengtsson-Palme, J., Rensing, C., Kristiansson, E., & Larsson, D. J. (2014). BacMet: antibacterial biocide and metal resistance genes database. Nucleic acids research, 42(D1), D737-D743.

[27] Kautsar, S. A., Blin, K., Shaw, S., Navarro-Muñoz, J. C., Terlouw, B. R., van der Hooft, J. J., ... & Medema, M. H. (2020). MIBiG 2.0: a repository for biosynthetic gene clusters of known function. Nucleic acids research, 48(D1), D454-D458.

[28] Yang, X., Yang, S., Li, Q., Wuchty, S., & Zhang, Z. (2020). Prediction of human-virus protein-protein interactions through a sequence embedding-based machine learning method. Computational and structural biotechnology journal, 18, 153-161.