EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Ambiente de desarrollo para la alineación de secuencias genómicas con inteligencia artificial

presentada por
Ing. Raul Magdaleno Peñaloza

como requisito para la obtención del grado de
Maestría en Ciencias de la Computación

Directora de tesis
Dra. Andrea Magadan Salazar

Codirector de tesis
Dr. Gerardo Reyes Salgado

Cuernavaca, Morelos, México. Agosto de 2023.

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

Ambiente de desarrollo para la alineación de secuencias genómicas con inteligencia artificial

Tesis para obtener el grado de Maestro en Ciencias de la Computación

Maestría en Ciencias de la Computación

Especialidad: Inteligencia Artificial

Director de tesis

Dra. Andrea Magadan Salazar

Codirector de tesis

Dr. Gerardo Reyes Salgado

Centro Nacional de Investigación y Desarrollo Tecnológico

Departamento de Ciencias de la Computación
"Maestría en Ciencias de la Computación"

Cuernavaca, Morelos, 7 de junio de 2023

Cuernavaca, Mor., 07/agosto/2023

OFICIO No. DCC/154/2023

Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de RAÚL MAGDALENO PEÑALOZA, con número de control M20CE062, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"AMBIENTE DE DESARROLLO PARA LA ALINEACIÓN DE SECUENCIAS GENÓMICAS CON INTELIGENCIA ARTIFICIAL"**. y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

EDUCACIÓN
14 AGO 2023
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SERVICIOS ESCOLARES
RECIBIDO

Andrea Magadán Salazar
Directora de tesis

Raúl Pinto Elías
Revisor 1

Jonathan Villanueva Tavira
Revisor 2

SEP TecNM CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
RECIBIDO
14 AGO 2023
SUBDIRECCIÓN ACADÉMICA

EDUCACIÓN
14 AGO 2023
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SERVICIOS ESCOLARES
RECIBIDO

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante

Cuernavaca, Mor., 14/agosto/2023
No. De Oficio: SAC/132/2023
Asunto: Autorización de impresión de tesis

RAÚL MAGDALENO PEÑALOZA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"AMBIENTE DE DESARROLLO PARA LA ALINEACIÓN DE SECUENCIAS GENÓMICAS CON INTELIGENCIA ARTIFICIAL"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/LMZ

Agradecimientos

- Agradezco al CONACYT por el apoyo económico brindado para la realización de mis estudios.
- Agradezco al TecNM/CENIDET por todo el apoyo brindado mediante las opciones de estudio en línea debido a pandemia durante toda mi estancia.
- Al Dr. Manuel Mejía Lavalle, siendo mi asesor con quien empecé este proyecto, le agradezco enseñarme y compartirme sus conocimientos, siendo de una manera muy estricta, rígida y motivadora, dejando una huella en mí, para seguir aprendiendo y llegar lejos. Me habría gustado mucho el poder aprender más de Ud. Descanse en Paz.
- Agradezco a la Dr. Andrea Magadan Salazar, por su apoyo y estímulo al ser quien continúo como mi asesora en este proyecto hasta poder terminarlo.
- Agradezco al Dr. Gerardo Reyes Salgado, por toda su ayuda y dirección brindada durante el proyecto.
- Agradezco a mis padres Raul Magdaleno Gómez, Amelia Peñaloza Bustos y mi hermana Mariana Magdaleno Peñaloza por haberme alentado tanto en este proyecto.
- Agradezco a mi novia Bianca Naomi Flores Corrales, mi gran compañera de vida por haberme acompañado, cuestionado, alentado y ayudado con cada comentario que pudiera surgir de este proyecto.
- Agradezco a los compañeros en los seminarios, aportando su apoyo, sus dudas y comentarios para hacer de este un gran proyecto.

Resumen

En la actualidad se utilizan y crean Ambientes de Desarrollo computacional (Frameworks) útiles para distintas áreas. El área de las ciencias genómica no es la excepción.

Este proyecto propone un Framework desarrollado para la alineación de secuencias genómicas, que es una herramienta básica, que permite extraer la información funcional, estructural y evolutiva contenida en secuencias biológicas, cuyo principal objetivo, es el determinar la relación entre diferentes especies y cuantificar el grado de similitud que hay entre ellas, conservadas a través de los años.

Este proyecto consiste en desarrollar un Framework que contiene métodos de alineación genómica mediante algoritmos clásicos de alineamiento y algoritmos de inteligencia artificial propuestos. Los algoritmos utilizados para este proyecto fueron Needleman-Wunsch, Smith-Waterman, Colonia de Abejas Artificiales y Algoritmo de Coste Uniforme. Estos algoritmos fueron seleccionados debido a las características para leer y generar resultados mediante la obtención de datos numéricos. Con la investigación de estos algoritmos, se planteó la hipótesis para su experimentación y de ser posible, su implementación, con el objetivo de encontrar diversas combinaciones, resultados y soluciones.

Con este proyecto, se busca crear un Ambiente que facilite el proceso de alineamiento genómico para encontrar adaptaciones y encontrar diferentes similitudes en el alineamiento de secuencias a partir del algoritmo básico propuesto por Needleman-Wunsh e implementar de una manera óptima, métodos de Inteligencia Artificial.

Abstract

At present, useful Computational Development Environments (Frameworks) are used and created for different areas. The area of genomic sciences is no exception.

This project proposes a Framework developed for the alignment of genomic sequences, which is a basic tool that allows extracting the functional, structural and evolutionary information contained in biological sequences, whose main objective is to determine the relationship between different species and quantify the degree of similarity between them, preserved through the years.

This project consists of developing a Framework that contains genomic alignment methods using classical alignment algorithms and proposed artificial intelligence algorithms. The algorithms used for this project were Needleman-Wunsch, Smith-Waterman, Artificial Bee Colony and Uniform Cost Algorithm. These algorithms were selected due to the characteristics to read and generate results by obtaining numerical data. With the investigation of these algorithms, the hypothesis was raised for its experimentation and, if possible, its implementation, with the aim of finding various combinations, results and solutions.

With this project, we seek to create an Environment that facilitates the genomic alignment process to find adaptations and find different similarities in the sequence alignment from the basic algorithm proposed by Needleman-Wunsh and optimally implement Artificial Intelligence methods.

Contenido

Índice de Figuras

Capítulo I

Capítulo II

Capítulo III

Capítulo IV

Capítulo V

Trabajos académicos Adicionales

Índice de Tablas

Capítulo III

Capítulo V

Capítulo I

Introducción

Con este proyecto, se busca desarrollar un ambiente que facilite el proceso de alineamiento genómico para encontrar nuevas y distintas adaptaciones, al igual que se busca encontrar nuevas alineaciones con diferentes similitudes entre ellas, partiendo de la implementación del algoritmo clásico Needleman-Wunsh [1].

El proyecto, también incorpora los métodos de inteligencia artificial: coste uniforme y colonia artificial de abejas para la optimización de los algoritmos clásicos, aportando nuevos métodos de solución con nuevas y diferentes posibilidades a las que podría generar un algoritmo clásico. A continuación, se enuncian los objetivos y alcances y limitaciones del presente proyecto.

1.1 Problema

El problema es, generar un Framework para generar Alineaciones Genómicas, empleando técnicas de Inteligencia Artificial.

1.2 Objetivos

General:

Desarrollar una aplicación Framework de manejo sencillo para usuarios con conocimientos básicos en alineación de secuencias genómicas, empleando métodos de inteligencia artificial que permitan encontrar, proporcionar y mostrar las mediciones de dichos alineamientos.

Específicos:

- Leer secuencias genómicas.
- Implementar métodos de alineación clásicos (Needleman-wunsh, Smith-Waterman).
- Implementar métodos de inteligencia artificial (Coste uniforme, Colonia Artificial de Abejas).
- Desarrollo de una interfaz comprensible para usuarios no especialistas.

1.3 Alcances:

- Alinear secuencias genómicas sin limitación en el tamaño de las mismas.
- Implementar algoritmos de inteligencia artificial para el alineamiento genómico.
- Implementar y optimizar los algoritmos clásicos de alineamiento Needleman-wunsh y Smith-Waterman.
- Proporcionar porcentaje de similitud entre alineaciones de secuencias mediante el score de valores utilizando métricas matriciales.

1.4 Propuesta de solución

Para la propuesta de solución, se comenzó por investigar métodos de alineación clásicos, para poder tener una base para empezar a trabajar. De esta manera, se propuso un Framework de uso sencillo que incorpora algoritmos clásicos de alineamiento y alineamiento de secuencias utilizando algoritmos de inteligencia artificial. Al investigar dónde obtener las secuencias genómicas para la generación de alineamientos, se encontró que estas secuencias pueden ser descargadas del banco de datos National Center of Biomedical Information (NCBI) [8] y leídas por la computadora personal, como si fuera un archivo de texto con extensión .txt.

Se decidió elegir el lenguaje de programación Python [2], tanto por su facilidad para leer archivos de texto como por su popularidad en el ámbito de la inteligencia artificial. En este lenguaje se procesan los datos de las secuencias, aplicando los métodos clásicos de alineación Needleman [1] y Smith-Waterman [15], junto con un método propuesto de alineación, utilizando el algoritmo clásico de inteligencia artificial de Coste Uniforme y concluyendo con el método de optimización Colonia Artificial de Abejas incorporado al algoritmo Smith-Waterman.

La metodología final propuesta, se muestra estructurada en la figura 1.1

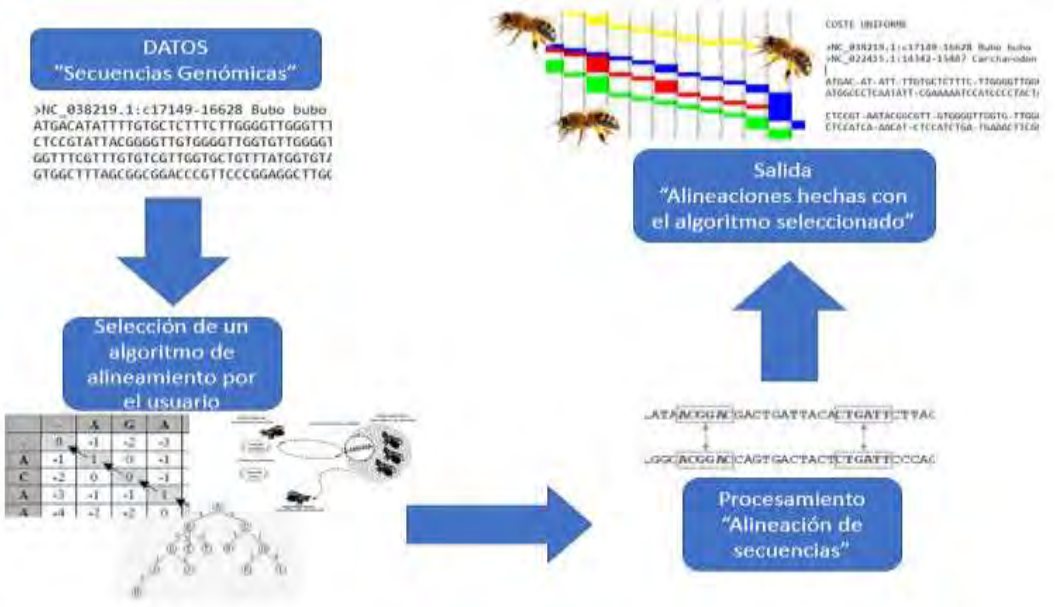


Figura 1. 1 Diagrama de flujo de metodología final empleada

1 Adquisición de las secuencias genómicas: las secuencias genómicas, fueron descargadas de los bancos de datos internacionales “National Center for Biotechnology Information NCBI” [8], “Dna DataBase of Japan DDBJ” [9] y “The European Bioinformatics Institute EBI” [10]. Se escogieron estas tres bases de datos, debido a que las tres forman la International Nucleotide Sequence Database Collaboration, la cual fue creada para cubrir el espectro de colectar bases de datos que contengan información sobre secuencias de DNA y RNA. Estas bases de datos internacionales son actualizadas cada tres meses.

2 Procesamiento: las secuencias genómicas son descargadas en un formato llamado “fasta”, este formato, puede leerse y usarse con un archivo con extensión “.txt”. Para el procesamiento de las secuencias descargadas, se optó por leerlas con el lenguaje de programación Python, debido a su versatilidad para leer documentos con ambas extensiones.

3 Algoritmos: en esta sección, el usuario puede utilizar entre cuatro algoritmos: “Needleman-Wunsh”, “Smith-Waterman”, “Colonia de Abejas Artificiales” y “Coste Uniforme”. Dependiendo del algoritmo utilizado, se generan matrices o árboles de decisión para generar las soluciones mediante lo indicado por el algoritmo.

4 Resultado: el resultado dependerá del algoritmo seleccionado con anterioridad. El resultado del algoritmo de “Coste Uniforme”, es una alineación global de las secuencias genómicas mediante la interpretación de el mejor resultado posible mediante puntuación. El algoritmo de “Colonia de Abejas Artificiales” genera las distintas posibilidades de alineación que pueden encontrarse como alternativa dentro del algoritmo “Smith-Waterman”, donde solo una alineación será la misma que el algoritmo clásico.

1.5 Organización de la tesis

En el presente documento se detalla el marco teórico en el Capítulo 2, listando los conceptos de secuencias genómicas, alineación de secuencias, bases de datos de internacionales de secuencias genómicas, algoritmos clásicos de alineación y algoritmos de inteligencia artificial. En el Capítulo 3 se presenta el Estado del Arte, con los temas relaciondos a la alineación de secuencias, donde también se presentan los trabajos de tesis antecedentes realizados en el TecNM/CENIDET. El Capítulo 4 menciona la Metodología de solución, donde se encuentra la propuesta de alineamiento, el método de Smith-Waterman optimizado con el algoritmo enjambre de abejas junto con el algoritmo de coste uniforme. El Capítulo 5 muestra las pruebas y resultados hechos con la experimentación de los algoritmos, terminando con el análisis de los resultados obtenidos. Y finalmente, en el Capítulo 6 se presentan las conclusiones, aportaciones y trabajos futuros.

Capítulo II
MARCO TEÓRICO

2.1 Secuencias Genómicas

Los ácidos Nucleicos son las biomoléculas portadoras de la información genética, biopolímeros de elevado peso molecular, formadas por otras subunidades estructurales o monómeros, denominados Nucleótidos [3].

Desde el punto de vista químico, los ácidos nucleicos son macromoléculas formadas por polímeros lineales de nucleótidos, unidos por enlaces éster de fosfato, sin periodicidad aparente.

De acuerdo con la composición química, los ácidos nucleicos se clasifican en Ácidos Desoxirribonucleicos (ADN) que se encuentran residiendo en el núcleo celular y algunos organelos, y en Ácidos Ribonucleicos (ARN) que actúan en el citoplasma.

De la misma manera que las proteínas son polímeros lineales aperiódicos de aminoácidos, los ácidos nucleicos lo son de nucleótidos. La aperiodicidad de la secuencia de nucleótidos implica la existencia de información.

Existe el significado biológico, determinado por [4], quien menciona, que las secuencias de ADN contienen la información genética en todos los seres vivos. Mientras dos secuencias contengan mayores rasgos en común, tenderán a ser más similares las funciones de las proteínas codificadas por ellas y las secuencias de un mismo gen en un conjunto de especies, serán más distintas las especies comparadas, cuando se encuentren más alejadas filogenéticamente.

Se conoce que el ADN es una cadena finita construida a partir de un alfabeto N = {A, C, G, T} de nucleótidos y el GENOMA es un conjunto de todas las secuencias de ADN asociadas a un organismo [5].

Dos secuencias tienden una alta similitud al ser homólogas cuando comparten un ancestro en común. A diferencia de la similitud, los genes al tener ancestros reducen posibilidades de que las secuencias puedan ser o no ser homólogas, debido a que la mayoría de las moléculas existentes no vienen del mismo ancestro. A partir de la similitud de las secuencias se infiere la homología. Con esto entendemos que el ADN sufre mutaciones a través de los años y a través de sus descendientes, lo que es la causa de que las secuencias de un mismo gen en dos especies distintas no sean idénticas. Cuanto más tiempo pase desde el último antecesor, más diferentes serán las secuencias [5].

2.2 Alineación de secuencias

La alineación de secuencias genómicas es una herramienta básica que permite extraer la información funcional, estructural y evolutiva contenida en secuencias biológicas.

Cuando se comparan dos o más secuencias, los principales objetivos son:

- Determinar y cuantificar el grado de similitud que hay entre ellas
- Determinar si existe algún tipo de relación entre ellas o si el parecido es simplemente fruto de la casualidad
- Detectar la presencia de motivos estructurales y/o funcionales conservados a través de sus descendientes.
- Construir árboles filogenéticos que reflejen sus relaciones evolutivas

La propuesta de [5] menciona, que, para poder encontrar el grado de similitud en dos secuencias, lo primero que debe de hacerse, es alinearlas. Consiste en escribirlas una arriba de otra, de modo que los símbolos que coincidan en una misma posición sea el máximo. A estas coincidencias se les conoce como MATCH, tal como lo muestra la Figura 2.1

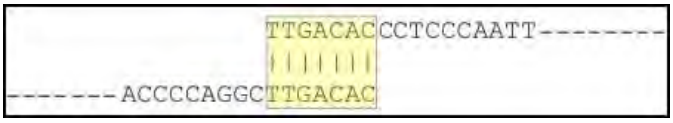


Figura 2. 1 Ejemplo de secuencias. [6]

De ser necesario, se pueden introducir huecos en cualquiera de las secuencias. Estos huecos también se denominan *gaps* e *indels* (insertion/deletion), los cuales son introducidos en las secuencias y son considerados como la inserción de un residuo en una de las secuencias, dando así la existencia de alguna inserción o borrado de un ancestro en otra.

En una alineación, cuando no existe coincidencia y para no mover toda la secuencia, se deja en el lugar más cercano a ella, nombrándolo NoMATCH, como lo muestra figura 2.2

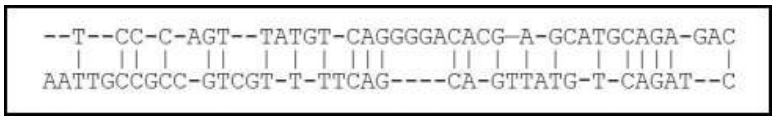


Figura 2. 2Alineación Match NoMatch Gap/Indel [6]

[7] menciona que, en cada posición del alineamiento, existen dos caracteres, donde pueden ser iguales (MATCH), diferentes (NoMATCH) o un carácter alineado con un hueco (Indel/gap). Lo que no puede ocurrir, es la existencia de dos huecos alineados, debido a que de manera natura, ambas alineaciones, no existiría una separación de nucleótidos, que permitieran dicho suceso.

Dos secuencias se pueden alinear de diferentes maneras. Para determinar cuál es el mejor alineamiento se utiliza un sistema de puntuación que otorga a cada pareja de caracteres un valor distinto en función de que sean iguales, distintos o que exista algún hueco. La puntuación de un alineamiento se calcula sumando la puntuación de cada una de las posiciones y nos ayuda a determinar si las secuencias están realmente relacionadas o si su parecido es fruto de la casualidad.

El alineamiento que obtiene la mayor puntuación se denomina alineamiento óptimo, mostrado en la figura 2.3

T	T	G	T	C	-	A	G	A	
T	T	G	T	C	G	A	G	G	score
+1	+1	+1	+1	+1	-2	+1	+1	-1	4

Figura 2. 3 Suma de puntos [6]

Al comparar dos secuencias, puede existir la posibilidad, de que ambas sean idénticas, siendo lo más probable de que exista alguna descienda directamente de la otra por mecanismos hereditarios.

También existe la posibilidad de que las secuencias sean parecidas. En este caso, el parecido puede deberse a que ambas secuencias descienden de un ancestro en común, tornando la alineación en homologa o puede tratarse de un caso de evolución convergente.

2.3 Bases de datos internacionales de secuencias genómicas

Hasta ahora, se ha hablado de las secuencias genómicas y de cómo es la alineación de las mismas. Sin embargo, existe la pregunta, ¿De dónde se obtienen las secuencias genómicas?

Existen distintos bancos de datos genómicos en el mundo. En esta sección se presentarán las más importantes y él porque es bueno utilizar la información de estos lugares.

2.3.1 National Center of Biomedical Information (NCBI) [8]

La Librería Nacional de Medicina (National Library of Medicine - NLM) [33] en Estados Unidos de America, tiene una base de datos de secuencias genéticas y colección de anotaciones de ADN.

El Banco de Genomas conocido como GenBank, es parte de una colaboración internacional de Bases de Datos de secuencias de nucleocitos, los cuales albergan a las siguientes instituciones:

- Dna DataBank of Japan (DDBJ)
- European Nucleotide Archive (ENA)
- National Center of Biomedical Information (NCBI)

El NCBI actualiza su banco de datos de genomas cada dos meses. Los nucleótidos e información biomédica están disponibles en su plataforma de manera pública, la cual proporciona información detallada sobre los cambios en genomas, al igual que nuevas investigaciones realizadas.

El acceso a esta plataforma contiene los siguientes datos:

- Identificadores de secuencias.
- Alineación de secuencias.
- Buscar, vincular y descargar secuencias.
- Formatos ASN1 y Flatfile disponibles en servidor FTP anónimo de NCBI.

El Centro Nacional de Información Biomédica (NCBI) tiene como objetivo promover la ciencia y la salud brindando acceso a la información biomédica de manera pública.

MISIÓN DEL NCBI

- Descubrir nuevos conocimientos.
- Comprende el lenguaje de la naturaleza y de las células vivas de una manera elegante mediante un alfabeto de cuatro letras, representando las sub unidades químicas de donde surge una sintaxis de procesos virtuales cuya expresión más completa es la del ser humano.

El volumen de datos moleculares ha llevado a una necesidad absoluta de herramientas de análisis de base de datos computarizada. El desafío consiste en encontrar nuevos enfoques para hacer frente al volumen y a la complejidad de los datos, al igual que el proporcionar un mejor acceso a las herramientas y análisis de bases de datos computarizadas.

Investigación Básica

El NCBI se ha encargado de crear sistemas automatizados para almacenar y analizar conocimientos sobre biología molecular, bioquímica y genética, facilitando el uso de bases de datos y softwares por parte de la comunidad biomédica para recopilar información sobre métodos avanzados de procesamiento de información por computadora para analizar la estructura y función de moléculas biológicamente importantes.

Responsabilidades del NCBI

- Investigaciones sobre problemas biomédicos fundamentales a nivel molecular utilizando métodos matemáticos.
- Colaboración con la Biblioteca Nacional de la Salud.
- Desarrollar, distribuir, apoyar y coordinar el acceso a una variedad de bases de datos y softwares para la comunidad científica médica.
- Desarrolla y promueve estándares para la base de datos, depositar e intercambiar nomenclaturas biológicas.

2.3.2 DNA Database Bank of Japan – DDBJ [9]

DNA Database Bank of Japan (DDBJ), es el banco de bases de datos de ADNs más importante de oriente, recopilando información de investigadores orientales, teniendo como actividad principal el recopilar datos de secuencias de nucleótidos como miembro de la "Colaboración Internacional de Bases de Datos de Secuencias de Nucleótidos" (INSDC) proporcionando secuencias de nucleótidos y un sistema de supercomputadoras gratuitamente para la investigación en ciencias de la vida.

Misión DDBJ

- Mejorar la calidad de INSDC como dominio público, esforzándose por describir la información sobre los datos de la manera más rica posible.
- Registrar la evolución orgánica de las secuencias de nucleótidos de una manera más directamente que otros materiales biológicos.

La DDBJ está oficialmente certificada para recolectar secuencias de nucleótidos de investigadores y emitir el número de acceso reconocido internacionalmente a los remitentes de los datos. Dado que la DDBJ intercambia los datos publicados con ENA/EBI y NCBI a diario, los tres centros de datos comparten prácticamente los mismos datos en cualquier momento.

La DDBJ recopila datos de secuencia principalmente de investigadores japoneses, al igual que también acepta datos y emite los números de acceso a investigadores de otros países. El 99% de los datos del INSD de los investigadores japoneses se envían a través del DDBJ.

Gestión de bases de datos biológicas:

- Proporciona bases de datos mantenidas por DDBJ y otros a través de servicios web
- Puede descargarse bases de datos colectivamente desde el sitio FTP.
- Proporcionar herramientas de software para análisis biológicos desarrollados por DDBJ y otros a través de servicios web o en la supercomputadora NIG.

2.3.3 EBI – European Biotecnologic Institute [10]

El European Biotecnologic Institute (EBI) colabora con científicos e ingenieros de todo el mundo y proporcionan la infraestructura necesaria para compartir datos abiertamente en las ciencias de la vida.

Su principal misión es:

"Comprender cómo la genética afecta la salud de los seres humanos, las plantas y los animales, siendo fundamental para los avances en la prevención de enfermedades, la seguridad alimentaria y la biodiversidad."

EBI desarrolla bases de datos de ADN al igual que herramientas y software que permiten su respectiva alineación. Verifica y visualizar los diversos datos producidos en investigaciones haciendo que esa información esté disponible gratuitamente para todo el mundo.

Siendo parte del Laboratorio Europeo de Biología Molecular (EMBL), una organización de investigación internacional, innovadora e interdisciplinaria financiada por más de 20 estados, siendo miembros asociados potenciales.

El EBI proporciona datos y servicios bioinformáticos de libre acceso a la comunidad científica, manteniendo la gama más completa del mundo de recursos de datos moleculares disponibles gratuitamente, desarrolladas en colaboración por el tratado INSDC, las bases de datos y herramientas que ayudan a los científicos a compartir datos de manera eficiente, realizando consultas complejas y analizar los resultados de diferentes maneras.

Apoyando la coordinación del suministro de datos biológicos en toda Europa, el EMBL-EBI es un socio fundamental en varias de las infraestructuras de investigación emergentes de Europa, incluida la infraestructura ELIXIR para información biológica.

El Instituto Europeo de Bioinformática (EMBL-EBI) mantiene la gama más completa del mundo de recursos de datos moleculares actualizados y disponibles gratuitamente.

Desarrolla en colaboración con colegas de todo el mundo, permitiendo compartir datos, realizar consultas complejas y analizar resultados de diferentes formas. Pudiendo trabajar localmente descargando los datos y software o utilizarlos servicios de web para acceder a recursos mediante programación.

Los datos y herramientas de EBI están disponibles gratuitamente. Siendo la única excepción, la información genética humana potencialmente identificable, cuyo acceso depende de acuerdos de consentimiento de investigación.

2.3.4 International Neucleotide Sequence Database Colaboration – INSDC [11]

La Colaboración Internacional de Base de Datos de Secuencias de Nucleótidos (INSDC) es una iniciativa fundamental de larga data que opera entre DDBJ, EMBL-EBI y NCBI. INSDC cubre el espectro de lecturas sin procesar de datos, desde alineaciones y ensamblajes hasta anotaciones funcionales, enriquecidas con información contextual relacionada con muestras y configuraciones experimentales.

Esta colaboración debe cubrir con los siguientes puntos garantizando el acceso a las diferentes bases de datos.

- El INSDC tiene una política uniforme de acceso libre y sin restricciones a todos los registros de datos que contienen sus bases de datos. Pudiendo acceder a estos registros para planificar experimentos o publicar cualquier análisis, otorgando el crédito apropiado citando la presentación original.

- INSDC no adjunta declaraciones a los registros que restrinjan el acceso a los datos, limiten el uso de la información en estos registros o prohíban ciertos tipos de publicaciones basadas en estos registros. Específicamente, no se incluirán restricciones de uso o requisitos de licencia en ningún registro de datos de secuencia, y ninguna de las partes impondrá restricciones o tarifas de licencia a la redistribución o uso de la base de datos.

- Todos los registros de la base de datos enviados al INSDC permanecen accesibles permanentemente como parte del registro científico.

- Se advierte a los remitentes que la información que se muestra en los sitios web mantenidos por el INSDC se divulga completamente al público.

- Más allá del control editorial limitado y algunas verificaciones internas de integridad, la calidad y precisión del registro son responsabilidad del autor que envía, no de la base de datos. Las bases de datos trabajarán con los remitentes y los usuarios de la base de datos para lograr el recurso de la mejor calidad posible.

2.4 - Algoritmos clásicos de alineamiento genómico

Existen diferentes tipos de algoritmos de alineamiento. En esta sección se detallarán los métodos clásicos. Entre ellos se encuentra el algoritmo de alineación Dot Plot, también conocido como Algoritmo de fuerza bruta. El algoritmo clásico Needleman y el algoritmo Smith-Waterman, ambos llamados así por sus respectivos desarrolladores.

2.4.1 - Dot-Plot, Algoritmo de fuerza bruta [6]

Para este algoritmo, las secuencias son leídas como una cadena de texto, siendo procesadas y divididas carácter por carácter. Cada uno de estos caracteres es ingresado dentro de las posiciones de un vector, que más adelante permitirá llevar a cabo el alineamiento de una manera más fácil.

Los caracteres son leídos y procesados uno por uno, dando unidades de distancia entre caracteres para identificar por puntuación entre caracteres, cual es el mejor puntaje. Los caracteres son desplazados o alineados según la heurística. De esta manera, se hace el alineamiento global de las secuencias.

2.4.2 - Algoritmo Needleman – Wunch [1]

Saul B. Needleman y Christian D. Wunch introdujeron en 1970 un enfoque para calcular la alineación global óptima de dos secuencias [12]. A diferencia de los métodos convencionales anteriormente mencionados Needleman – Wunch [1] es una manera de reducir de manera masiva el número de posibilidades a considerar para encontrar la mejor solución.

De acuerdo con [12] [13], bajo el supuesto de que ambas secuencias de entrada “a” y “b” proceden del mismo origen, una alineación global trata de identificar las partes que coinciden y los cambios necesarios para transferir una secuencia a la otra. Los cambios se anotan y se identifica un conjunto óptimo de cambios, lo que define una alineación.

El enfoque de programación dinámica tabula las sub-soluciones óptimas en la matriz D [1]. Donde se observa una representación de una matriz 2D.

Este algoritmo consiste en tres pasos.

- Iniciar la matriz Score
- Calcular la puntuación y rellenar la matriz posterior
- Deducir el alineamiento de la matriz posterior

Dando un ejemplo de dos palabras: SEND Y AND.

Para determinar cuál es el mejor alineamiento se utiliza un sistema de puntuación que otorga a cada pareja de caracteres un valor distinto en función de que exista una alineación con mismo carácter (MATCH), una alineación con diferente carácter (NoMATCH) o en su defecto un hueco entre caracteres (GAP/INDEL). Dando puntuaciones, Match de 1+, NoMATCH de -1, INDEL de -2.

SEND
-AND score: +1
A-ND score: +3 ← Esta fue la mejor puntuación encontrada.
AN-D score: -3
AND- score: -8

Ahora, para poder crear una matriz Needleman-Wunch [1], se usa el siguiente procedimiento, que es: alinear la primera secuencia de forma horizontal en la parte superior y la segunda secuencia de manera vertical, pegada a la izquierda. En la figura 2.4 se presenta la manera de ordenar las secuencias, donde la puntuación de cada celda será el máximo.

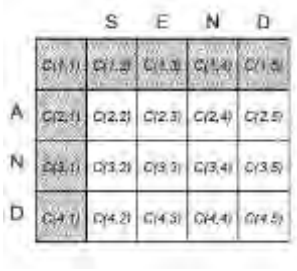


Figura 2. 4 Matriz Needleman [14]

En [3] se presenta el llenado de cada celda, se asigna el valor dependiendo si se genera un MATCH, un NoMatch o un GAP/INDEL.

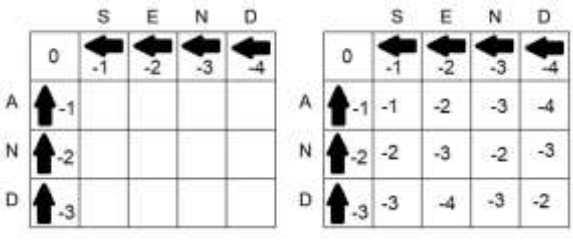


Figura 2. 5 Alineación por SCORE [14]

La figura 2.5 muestra cómo se deben de llenar las matrices con *Score,* entonces se podrá seguir de la manera siguiente: comenzando de la esquina inferior derecha hasta la esquina superior izquierda y siguiendo los mejores *SCORES*, como se muestra en la figura 2.6.

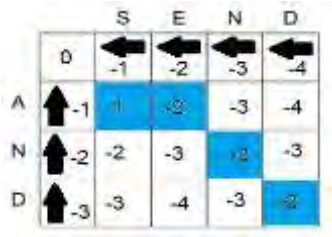


Figura 2. 6 Seguimiento de Score en matriz [14]

2.4.3 - Algoritmo Smith-Waterman [15]

El enfoque de programación dinámica de Temple F. Smith y Michael S. Waterman (1981) explicado de acuerdo a [15], calcula las alineaciones locales óptimas de dos secuencias. Identifica las dos secuencias que mejor se conservan, su alineación muestra la máxima puntuación de similitud.

En [16] se argumenta que este algoritmo está designado para encontrar el óptimo alineamiento local entre dos secuencias, basado en la computación de alineación de matrices de **Needleman.** El número de filas y columnas está dado por el número de caracteres en una secuencia.

Dos secuencias moleculares siendo A = [A1, A2, A3] & B = [B1, B2, B3] dicho por [17], su similitud es dada entre la secuencia de los elementos *A & B*. La eliminación de longitudes es dada por anchura, para encontrar pares de segmentos con altos grados de similitud.

Para encontrar pares de segmentos con altos grados de similitud, se configura una matriz *H*. El primer conjunto se ve expresado en la ecuación 1.

$$HkO = Hot = O\ for\ O < k < n\ \ and\ O < 1\ m$$

Ecuación 1 Expresión Matemática para la Matriz H [17]

Los valores preliminares de *H* tienen la interpretación donde H_{ij} es la máxima similitud de dos segmentos terminando en "a_i" y en "b_j" respectivamente. Se obtienen estos valores por la relación que expresa la ecuación 2:

$$H_{ij} = max\{H_{i-1,j-1} + g(a_i, b_j), max\{H_{i-k,j} - W_k\}, max\{H_{i,j-1} - W_1\}, 0\}$$

$$K<=1$$

Ecuación 2 Expresión Matemática [17]

En [4] se propone el ejemplo donde alinea las palabras "COELACANTH" con" PELICAN". La matriz se construye de la misma manera que **Needleman** [15]. La diferencia es que amplía con un caso

adicional '0'. Este límite inferior en la puntuación de similitud excluye las alineaciones "demasiado malas" que eventualmente "no son similares" (puntuación < 0).

Siendo que en la fila y la columna son 0 en lugar de los números negativos (-1, -2, -3, …) Los valores de los **MATCH & NoMATCH** siguen siendo los mismos. Ahora, al hacer el conteo, cualquier número que, de negativo, se pondrá en 0. Aquí se sigue el **SCORE** del más alto obtenido hasta llegar a 0. Este método se puede observar en la figura 2.7.

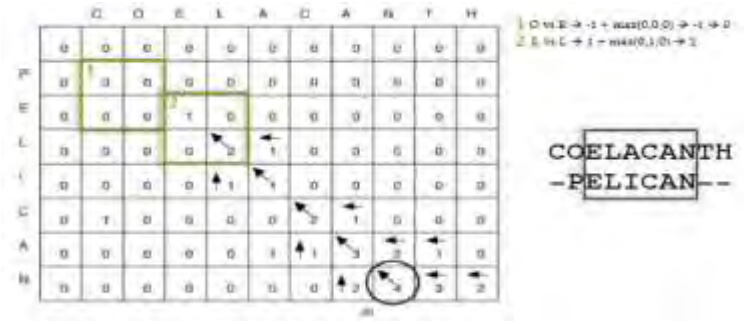


Figura 2. 7 Matriz Smith-Waterman [3]

2.5 - Algoritmos de Inteligencia artificial

2.5.1 - Algoritmo Coste Uniforme [18]

Este algoritmo permite realizar una búsqueda en el espacio de estados, teniendo en cuenta el factor coste [18]. En esta búsqueda, se genera un árbol de decisiones, donde cada nodo tiene un coste y este se empieza a recorrer desde el primer nodo generado hasta el último evaluado, siendo el coste final, la suma de cada uno de los costes en los nodos recorridos.

El algoritmo de Coste Uniforme utiliza una cola con prioridad, la cual se mantiene ordenada, que permanece de dicha manera al consultar la prioridad de los nodos, siendo posible ordenarla en cada paso al igual que alternativamente extrayendo el menor o mayor de los valores, dependiendo el coste que desea buscarse [19].

Este algoritmo es considerado óptimo, debido a que selecciona el nodo deseado con mayor o menor coste. Al encontrarse con una solución, siendo seguro, que no existía mejor solución más que la determinada por el algoritmo, como se muestra en el pseudo código de la figura 2.8.

```
Pseudo código:
        -Estado inicial
       -Almacenar
        -While True:
                generar posibilidades
                asignar costo a cada posibilidad
                determinar la mejor posibilidad
                almacenar mejor posibilidad

                SI NODO ACTUAL == SOLUCION:   Terminar
```

Figura 2. 8 Algoritmo Coste Uniforme [18]

2.5.2 - Colonia Artificial de Abejas

El algoritmo Colonia Artificial de Abejas define un conjunto de operaciones que asemeja a las características del comportamiento de las abejas [20]. Propuesto por Dervis Karaboga en 2005 [21] está basado en el comportamiento de las abejas y diseñado originalmente para problemas de optimización numérica [22].

El proceso de búsqueda numérica se define como néctar, siendo parte de las abejas un proceso de optimización, donde el comportamiento de éstas se modela como una heurística de optimización basada en el modelo biológico que consta de los siguientes elementos:

- **Fuentes de alimento:** Valor numérico que indica su potencial.
- **Abejas recolectoras empleadas:** estas abejas explotan la fuente de alimento, al igual que son las encargadas de comunicar su ubicación y rentabilidad a las abejas observadoras.
- **Abejas recolectoras desempleadas:** etas abejas se encuentran buscando fuentes de alimento para explotar. Se dividen en dos tipos: las exploradoras que se encargan de buscar nuevas fuentes de alimento y las observadoras que esperan en la colmena para elegir alguna de las fuentes de alimento que se encuentran en un proceso de explotación por las abejas empleadas.

En este algoritmo, la posición de las fuentes de alimento, representan una posible solución al optimizar el problema y dicho alimento corresponde a la calidad de la asociación. El número de abejas empleadas es igual al número de soluciones posibles [21].

Las abejas empleadas seleccionan al azar un conjunto de posiciones de las fuentes de alimento y determinan sus cantidades de néctar. Comparten la información de las fuentes de alimento. Después de compartir la información, cada abeja empleada retorna a la zona de la fuente de alimento visitada previamente y a continuación elige una nueva fuente de alimento por medio de la información visual en el vecindario al evaluar su cantidad de néctar. Si la cantidad de néctar de una fuente de alimento aumenta, entonces también aumenta la probabilidad con la que la fuente de alimento es elegida. Después que la abeja observadora llega a la zona seleccionada, elige una nueva fuente de alimento en el vecindario dependiendo de la información obtenida [22]. En la figura 2.9 se observa un ejemplo en pseudo código:

Pseudo código: [21]

```
-Inicializar población
-Evaluar población
-Ciclos=n
-While Ciclos <= N:
      -Producir solución
      -Seleccionar néctar
      -Calcular valor para la solución
      -Producir nueva solución a partir del néctar recolectado
      -Archivar la solución
      -Ciclo + = 1
```

Figura 2. 9 Pseudo código Colonia Artificial de Abejas

Capítulo III
Estado del Arte

En este capítulo, se analizan diferentes temas y trabajos donde se han utilizado diferentes algoritmos de alineación genética tales como los mencionados en el capítulo anterior. En los artículos, se reportan investigaciones sobre el desarrollo de Frameworks y ejemplos de cómo se ha utilizado el algoritmo de Colonia Artificial de Abejas.

3.1 Antecedentes

Se presenta el resumen de dos tesis elaboradas en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET).

3 .1 .1 Alineamiento Genómico basado en el algoritmo Best First Search [23]

La alineación genómica es la predicción de las relaciones de evolución en un nivel nucleico entre dos o más genomas, diciendo en donde el correspondiente son encontrados los segmentos correspondientes en diferentes genomas.

Es utilizado **Needleman-Wunsch** que requiere dos secuencias como entrada. Poniendo en práctica este algoritmo, es posible obtener la mejor puntuación debido a la matriz y el seguimiento que se obtiene. Cuando el algoritmo termina, el programa genera automáticamente el mejor resultado de alineamiento.

Empleando un algoritmo de búsqueda, es posible buscar a través de cada solución viable para encontrar una secuencia para un problema específico. Este algoritmo es propuesto para realizar un alineamiento de genomas y nombrado como: “LA PRIMERA MEJOR BUSQUEDA” (THE BEST FIRST SEARCH).

Cualquier nodo generado por él es considerado. Utilizando heurística para determinar cuál es el mejor nodo que debería de ser explorado, por medio de los siguientes pasos.

1- Crea una secuencia vacía
2- Inserta un estado inicial
3- Explora el nodo, agrega posibilidades, marca las posibilidades, elimina posibilidades ya exploradas, reorganiza las posibilidades.

El principal problema es que se requiere una vasta cantidad de espacio y memoria para que el algoritmo se pueda poner en práctica.

Lo que se propone, es hacer el algoritmo por partes, utilizando alguna sección principal, con el objetivo de reducir tamaño y tiempo.

La complejidad de este algoritmo en el peor de los casos es donde cada carácter de la secuencia es el número de nodos y se visita cada uno de ellos para lograr una solución.

3.1.2 Análisis de datos genómicos para el diagnóstico temprano de osteosarcoma [24]

El cáncer se caracteriza por la presencia de células anormales, las cuales se multiplican de manera descontrolada e invaden otros órganos del cuerpo. El osteosarcoma, es un cáncer primario que se origina en los huesos, se define como un tumor maligno las cuales producen tejido óseo patológico que se forma alrededor de las articulaciones, y se presenta como una matriz no mineralizada. Uno de los grandes misterios del osteosarcoma, es la alta predisposición para desarrollar metástasis en pulmones, la falta de conocimiento al respecto es debido a que aún no se entiende el mecanismo por el cual las células de este cáncer migran a los pulmones y que propiedades del microambiente de los pulmones es ideal para el desarrollo y proliferación.

La bioinformática es una ciencia que surge de la necesidad de interpretar la información contenida en las secuencias de ADN, ARN y proteínas, a través de la implementación de técnicas computacionales como lo es la inteligencia artificial (IA). Debido a la problemática dada en fenómenos tan complejos de la genética, como la simulación del efecto de medicinas, la predicción de enfermedades. Todas estas situaciones manejan gran cantidad de información y variables, de allí emerge la necesidad de apoyarse con la IA. En este artículo se presentan conceptos importantes para el desarrollo de sistemas de reconocimiento. A continuación, se detallan los que se consideran más importantes.

Aprendizaje supervisado: El objetivo del aprendizaje supervisado es lograr que la computadora aprenda a clasificar. Implica proporcionar información para entrenar al algoritmo a reaccionar con los ejemplos que se le den, así aplicará lo que aprenda para dar respuestas a situaciones completamente nuevas que no ha visto antes. Ejemplos de técnicas tradicionales del aprendizaje supervisado son:

- Máquinas de vector soporte (SVM por sus siglas en inglés)
- Redes neuronales
- Clasificador Naïve Bayes
- Árboles de decisión
- Vecinos más cercanos (kNN)

Análisis de componentes independientes: se utiliza para la separación de señales multivariadas en subcomponentes aditivos, surge de la técnica conocida por su sigla BSS, o *Blind Sepparation Source*, que intenta obtener las fuentes independientes a partir de combinaciones de las mismas.

Random Forest: El concepto de Bosques Aleatorios (Random forest), es una combinación de árboles predictivos, el cual crea ejemplos separados del conjunto de entrenamiento y genera un clasificador para cada ejemplo. El resultado de estos clasificadores se muestra de manera gráfica en la figura 3.1. La estrategia es que cada ejemplo del conjunto de entrenamiento es diferente, entonces, cada clasificador o árbol entrenado tiene un diferente enfoque y perspectiva del problema.

En el algoritmo, cada árbol depende de los valores de un vector aleatorio de la muestra de manera independiente y con la misma distribución de todos los árboles en el bosque. La generalización de error para los bosques converge a un límite en cuanto el número de árboles en el bosque es grande. El error de generalización de un bosque de árboles de clasificación depende de la fuerza de los árboles individuales en el bosque y la correlación entre ellos.

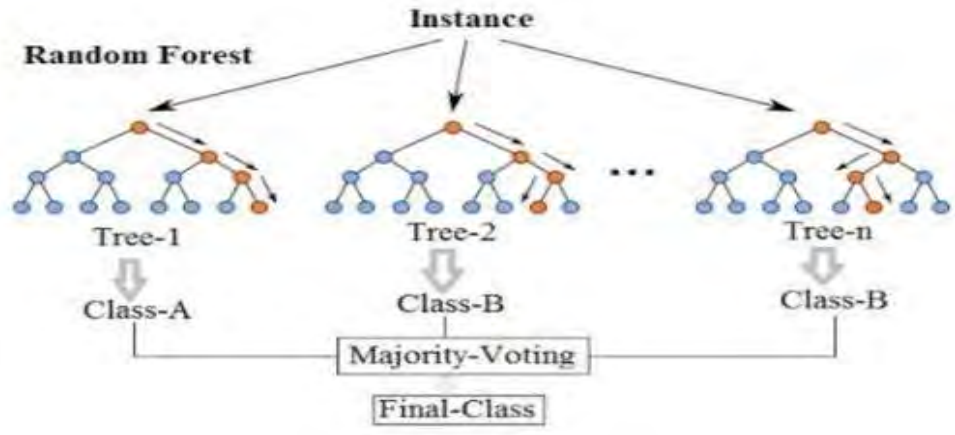


Figura 3. 1 Funcionamiento de Random Forest [24].

XGBoost: *eXtreme Gradient Boosting*. Implementación de árboles de decisión con *Gradient boosting* diseñada para minimizar la velocidad de ejecución y maximizar el rendimiento. Pertenece a una familia de algoritmos Boosting que convierten al aprendizaje débil en aprendizaje fuerte. Un aprendiz débil es uno que es ligeramente mejor que adivinar al azar. Boosting es un proceso secuencial; es decir, los árboles se cultivan utilizando la información de un árbol previamente crecido uno tras otro. Este proceso aprende lentamente de los datos e intenta mejorar su predicción en iteraciones posteriores para reducir el error de clasificación como lo muestra la figura 3.2.

Box= clasificador
D=división o split

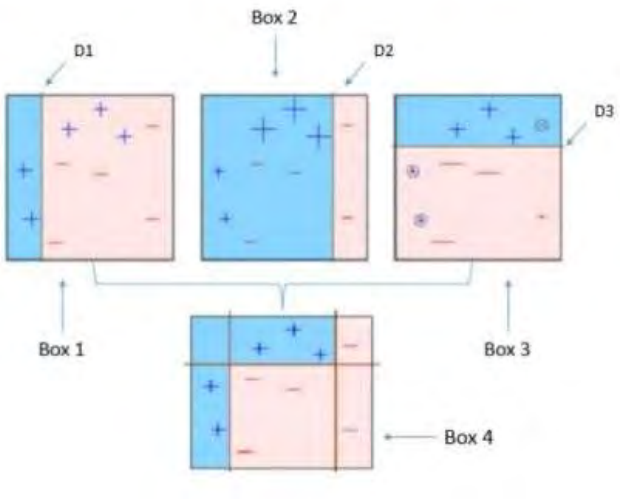


Figura 3. 2 Funcionamiento XGBoost [24]

Reconocimiento de patrones: Un patrón se representa por un vector numérico de dimensión '*n*'. De esta forma, un patrón es un punto en un espacio n-dimensional (características).

El reconocimiento de patrones se integra de dos fases: entrenamiento y reconocimiento. En el entrenamiento, se diseña el extractor o selector de características para representar los patrones de entrada y se entrena al clasificador con un conjunto de datos de ejemplo de forma que el número de patrones mal identificados se minimice. En la etapa de reconocimiento, el clasificador ya entrenado toma como entrada el vector de características de un patrón desconocido y lo asigna a una de las clases o categorías.

Metodología de solución: Para el desarrollo del proyecto se utilizó R versión 1.1.463.
Para cumplir los objetivos propuestos, siguió la metodología que se muestra en la figura 3.3.

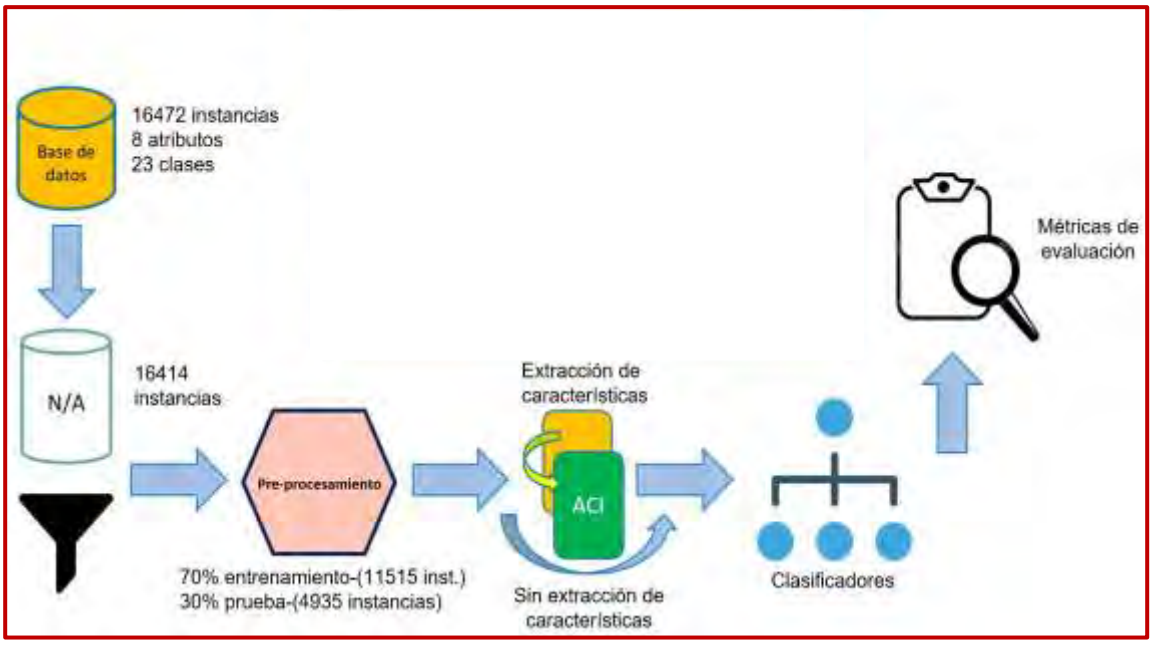


Figura 3. 3 Metodología de Solución [24]

La base de datos fue proporcionada por el Laboratorio de Biología de Sistemas y Medicina Traslacional (BSMT) de la Universidad Autónoma del Estado de Morelos (UAEM) proveniente de la secuenciación de ADN, la cual consta de 8 atributos:

- gene symbol: característica de tipo cadena, ejemplo: *ABCA5, ABCB1 ABCC4.*
- function/ phenotype: característica de tipo cadena, ejemplo: *Actins are highly conserved proteins that are involved in various types of cell motility and are ubiquitously expressed in all eukaryotic cells.* Esta característica puede llegar a tener 4942.
- clinical significance: esta característica es de tipo cualitativo, ejemplo: *Benign drug response, Pathogenic.*
- chromosome, esta característica es de tipo numérico, ejemplo: 17, 7, 13 • dbSNP: característica de tipo nominal, ejemplo: 199753304, 2032582, 1045642, 1128501, 1751034.
- association, esta característica es de tipo cadena, ejemplo: *Familial adenomatous polyposis 1; Hereditary cancer-predisposing síndrome.* de igual forma que la segunda característica, esta puede contener más de 1000 caracteres.
- ancestral allele: característica de tipo cualitativo, ejemplo: C, A, T, G.
- alternate allele, característica de tipo cualitativo, ejemplo: A, C, T, G.

Implementación del preprocesamiento

Se analizó el banco de datos y se encontraron datos grandes, la cual muestra que en una sola celda puede haber 4,942 caracteres. Se realizó la limpieza de los datos, aquellos ausentes de valores (NA) o *missing values.* Posteriormente se pasó a la codificación únicamente del atributo Clase, que es de tipo cadena, a tipo nominal.
A partir del análisis de los datos, se realizaron las siguientes adecuaciones para su limpieza y mejor manejo:

- Valores de N/A (ausencia de valor) se eliminaron ya que no aportan información.
- Transformación del atributo clase (23 clases). se transformó a tipo nominal con el fin de reducir el tiempo de entrenamiento por medio de una codificación.

Extracción de características

Debido a que el vector de características se integra de pocos atributos, se eligió utilizar el análisis de componentes independientes para tener el mismo número de variables.

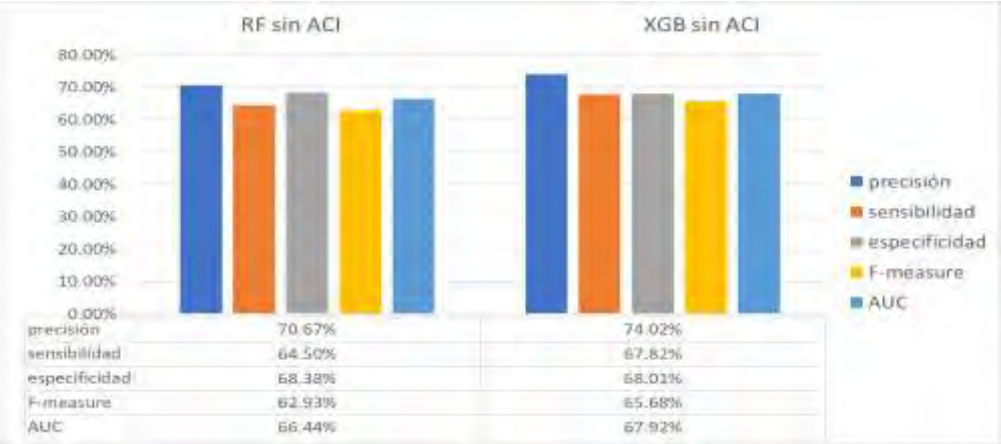
Clasificación

Los algoritmos de Random Forest y XGBoost son los utilizados para este trabajo, debido a que permiten procesar múltiples variables de tipo cadena, cualitativos y cuantitativos. A diferencia de otros clasificadores como, Naive Bayes, máquina de soporte vectorial, K vecinos cercanos, Redes neuronales, regresión logística, que son algoritmos de clasificación más utilizados, no son capaces de procesar la naturaleza de las variables ya mencionadas.

Tabla 3. 1 Resultados del experimento

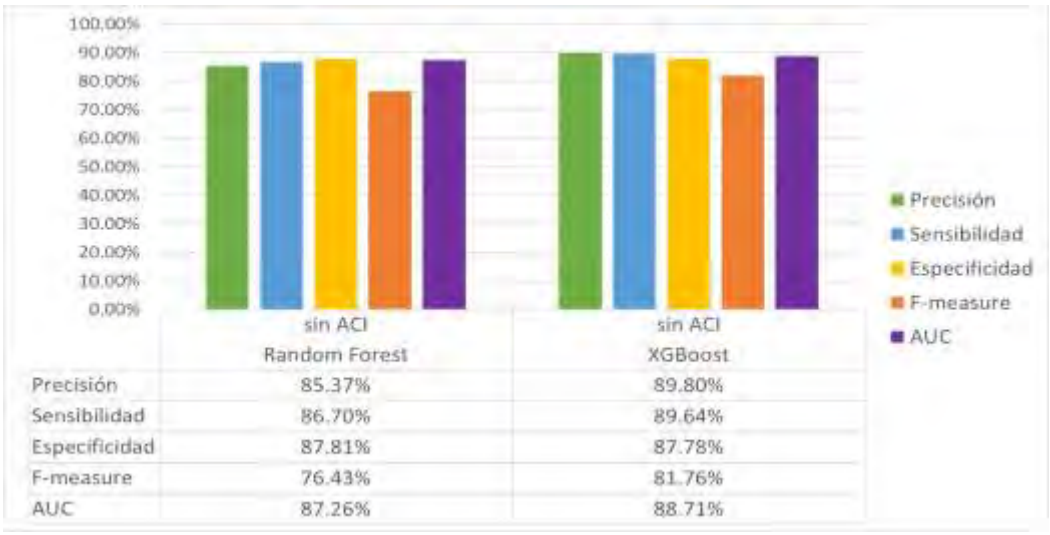
Algoritmo	Técnica	Precisión	Sensibilidad	Especificidad	F-measure	AUC	No. Arboles	Profundidad	Tasa de aprendizaje	Peso	No. Prueba
Random Forest	sin ACI	69.09%	56.02%	64.12%	61.69%	60.07%	100	3	0.2	1	1
	ACI	68.17%	62.20%	66.13%	58.96%	64.17%					
XGBoost	sin ACI	71.01%	61.06%	63.40%	62.33%	62.23%					
	ACI	70.43%	63.10%	63.75%	62.07%	63.43%					
Random Forest	sin ACI	69.36%	63.10%	66.20%	58.69%	64.65%	200	3	0.2	1	2
	ACI	69.80%	64.59%	67.35%	58.43%	66.97%					
XGBoost	sin ACI	72.60%	62.48%	68.85%	57.15%	66.67%					
	ACI	73.20%	66.35%	67.41%	66.23%	66.38%					
Random Forest	sin ACI	69.36%	63.10%	66.20%	64.29%	64.65%	600	3	0.2	1	3
	ACI	69.80%	64.59%	67.35%	66.99%	66.97%					
XGBoost	sin ACI	72.60%	62.48%	67.85%	64.19%	66.17%					
	ACI	73.20%	66.35%	67.41%	57.00%	66.38%					
Random Forest	sin ACI	72.10%	69.18%	72.40%	65.64%	70.79%	1000	3	0.2	1	4
	ACI	74.56%	61.25%	71.43%	60.97%	66.34%					
XGBoost	sin ACI	74.00%	73.80%	68.13%	73.09%	70.97%					
	ACI	75.10%	70.47%	73.20%	68.02%	71.84%					
Random Forest	sin ACI	73.43%	71.10%	73.00%	64.46%	72.05%	2000	3	0.2	1	5
	ACI	76.40%	66.37%	74.16%	61.16%	69.77%					
XGBoost	sin ACI	79.87%	79.30%	71.80%	71.61%	75.55%					
	ACI	80.98%	68.90%	79.10%	59.23%	74.00%					

En la Tabla 3.1 se presentan los resultados obtenidos del experimento 1 con cinco métricas (precisión, sensibilidad, especificidad, F-measure y AUC), donde se hizo un promedio de los resultados. Se compararon los resultados alcanzados con los detallados en la Figura 3.4, donde la sensibilidad promedio RF sin ACI fue de 64.50%, un 68.38% en especificidad, un 62.93% y el promedio de la precisión con RF con ACI fue de 71.75% donde el menor valor fue de 68.17%.



precisión	70.67%	74.02%
sensibilidad	64.50%	67.82%
especificidad	68.38%	68.01%
F-measure	62.93%	65.68%
AUC	66.44%	67.92%

Figura 3. 4 Promedio de experimento 1 sin ACI [24]



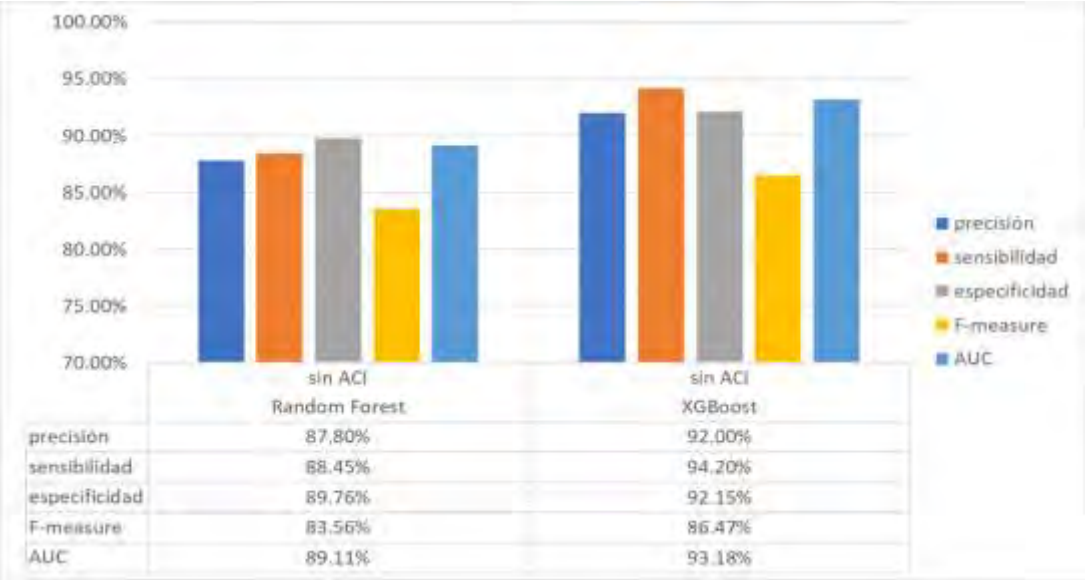
	sin ACI Random Forest	sin ACI XGBoost
Precisión	85.37%	89.80%
Sensibilidad	86.70%	89.64%
Especificidad	87.81%	87.78%
F-measure	76.43%	81.76%
AUC	87.26%	88.71%

Figura 3. 5 Comparación de experimento 3 sin ACI [24]

En la figura 3.5 muestra que la precisión de RF sin ACI fue de 85.37%, una sensibilidad de 86.70%, especificidad de 87.81%, F-measure de 76.43% y un AUC de 87.26%. XGBoost sin ACI obtuvo una precisión de 89.80%, una sensibilidad de 89.64%, una especificidad de 87.81%, F-measure de 81.76% y un AUC de 88.71%.

Experimentación: Tasa de aprendizaje

En este experimento se fue aumentando el número de árboles partiendo de los parámetros del tercer experimento, también se modificó la tasa de aprendizaje para una rápida conversión en el entrenamiento, dando como resultado los siguientes parámetros y resultados (ver figura 3.6 y figura 3.7).



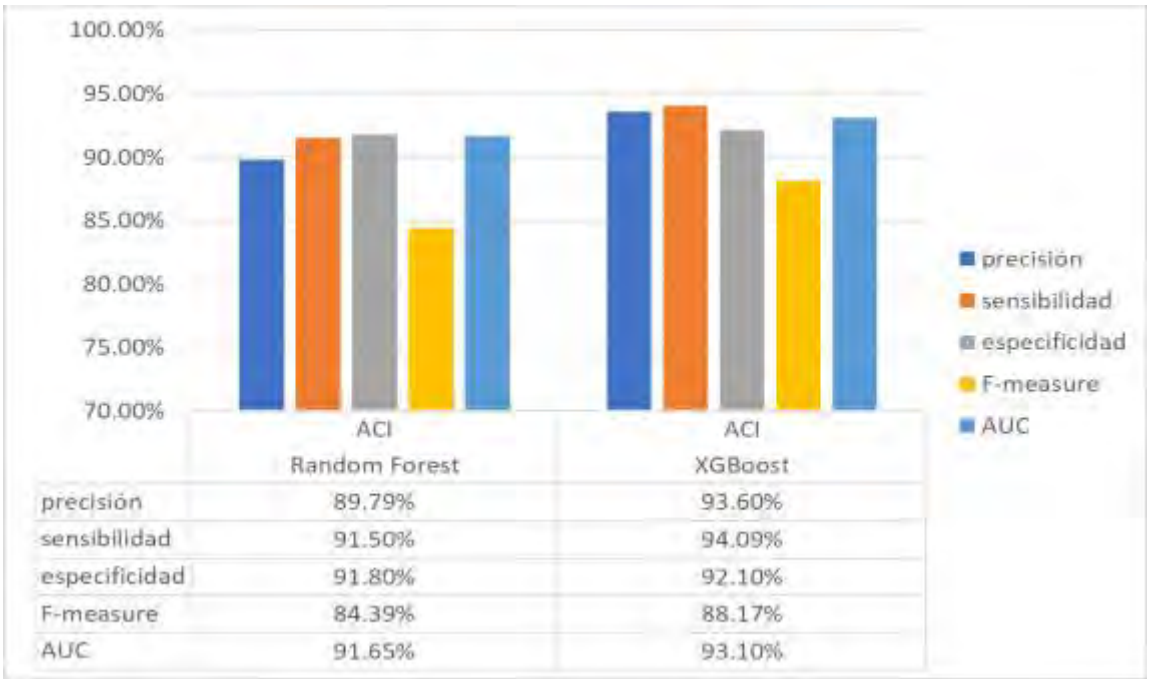
	sin ACI Random Forest	sin ACI XGBoost
precisión	87.80%	92.00%
sensibilidad	88.45%	94.20%
especificidad	89.76%	92.15%
F-measure	83.56%	86.47%
AUC	89.11%	93.18%

Figura 3. 6 Comparación de resultados experimento 4 sin ACI [24].

Parámetros: número de árboles = 2,800 RF y 2,200 XGB, Profundidad = 10 RF y 11 XGB, Tasa aprendizaje = 0.75, Peso = 1.

La precisión de RF sin ACI fue de 87.80%, una sensibilidad de 88.45%, especificidad de 89.76%, F-measure de 83.56% y un AUC de 89.11%.

XGBoost sin ACI obtuvo una precisión de 92%, una sensibilidad de 94.20%, una especificidad de 92.15%, F-measure de 86.47% y un AUC de 93.18%.



	ACI Random Forest	ACI XGBoost
precisión	89.79%	93.60%
sensibilidad	91.50%	94.09%
especificidad	91.80%	92.10%
F-measure	84.39%	88.17%
AUC	91.65%	93.10%

Figura 3. 7 Comparación de resultados experimento 4 con ACI [24].

La precisión de RF con ACI fue de 89.79%, una sensibilidad de 91.50%, especificidad de 91.80%, F-measure de 84.39% y un AUC de 91.65%.

XGBoost con ACI obtuvo una precisión de 93.60%, una sensibilidad de 94.09%, una especificidad de 92.10%, F-measure de 88.17% y un AUC de 93.10%.

Análisis de resultados

Durante las experimentaciones se fueron modificando los parámetros para tener los mejores valores en las métricas de evaluación, por lo que se dio a la tarea de hacer el cálculo de la precisión por clase como se muestra en la figura 3.8, se utilizó el experimento 4 para dicho cálculo ya que es el experimento donde mejores resultados se obtuvo.

Figura 3. 8 Precisión por clase [24]

Tabla 3. 2 Precisión por cada clase

	ACI XGBoost	ACI RF
Clase 1	88.62%	81.66%
Clase 2	97.39%	90.60%
Clase 3	89.91%	84.21%
Clase 4	95.05%	92.13%
Clase 5	85.60%	81.46%
Clase 6	87.12%	84.51%
Clase 7	94.87%	92.02%
Clase 8	92.71%	86.43%
Clase 9	87.94%	83.45%
Clase 10	90.70%	91.98%
Clase 11	91.25%	87.51%
Clase 12	97.79%	94.36%
Clase 13	94.67%	91.82%
Clase 14	92.82%	89.96%
Clase 15	97.88%	92.65%
Clase 16	98.40%	94.05%
Clase 17	94.61%	91.77%
Clase 18	95.00%	89.90%
Clase 19	96.13%	95.00%
Clase 20	93.23%	93.47%
Clase 21	96.78%	90.26%
Clase 22	96.84%	93.15%
Clase 23	97.57%	92.90%
Precisión final	93.60%	89.79%

Una vez ya calculado la precisión por clase, se dio a la tarea de conocer el número de ejemplos por clase (ver Ilustración 21 y Tabla 3), para conocer si los números de ejemplos correspondía al valor de la precisión.

Tabla 3. 3 Número de objetos por cada clase

Objetos por clase	
Clase 11	787
Clase 12	1022
Clase 13	286
Clase 14	267
Clase 15	2442
Clase 16	1571
Clase 17	349

Objetos por clase	
Clase 18	311
Clase 19	363
Clase 20	238
Clase 21	451
Clase 22	661
Clase 23	647

En figura 3.9 se muestra la distribución del número de objetos o ejemplos por clase, donde la clase 2, 15 y 16 muestran el mayor número de ejemplos y la clase 9 muestra el menor número de ejemplos. Al analizar los resultados se puede determinar que la clase 9 es la que cuenta con el valor más bajo de precisión en XGBoost obtuvo 87.94% y en Random Forest un 83.45%. Por otra parte, la clase 15 que es la cuenta con la mayor cantidad de ejemplos (2442) alcanza un 97.88% en XGBoost y un 92.65% en Random Forest, pero otro ejemplo es la clase 19 que alcanza el 96.13% en XGBoost y un 95% en Random Forest y cuenta con 363 ejemplos, por lo que se puede concluir que el valor de la precisión no es proporcional a la cantidad de ejemplos u observaciones por clase.

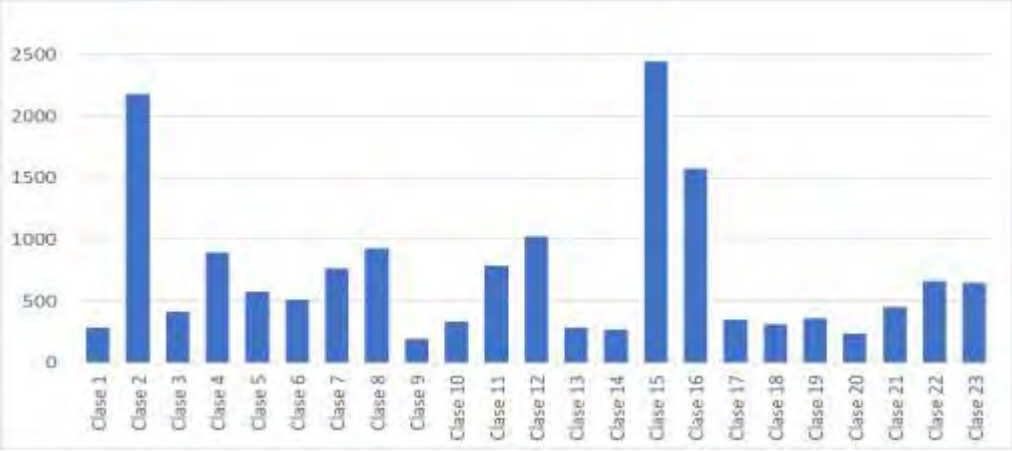


Figura 3. 9 Numero de objetos por clase [24]

Las evaluaciones efectuadas a los clasificadores en este trabajo permiten destacar que la combinación ACI-XGBoost del experimento 4 (ver figura 3.10) tuvo mejor desempeño respecto al tiempo, precisión, sensibilidad, especificidad, F-measure y AUC.

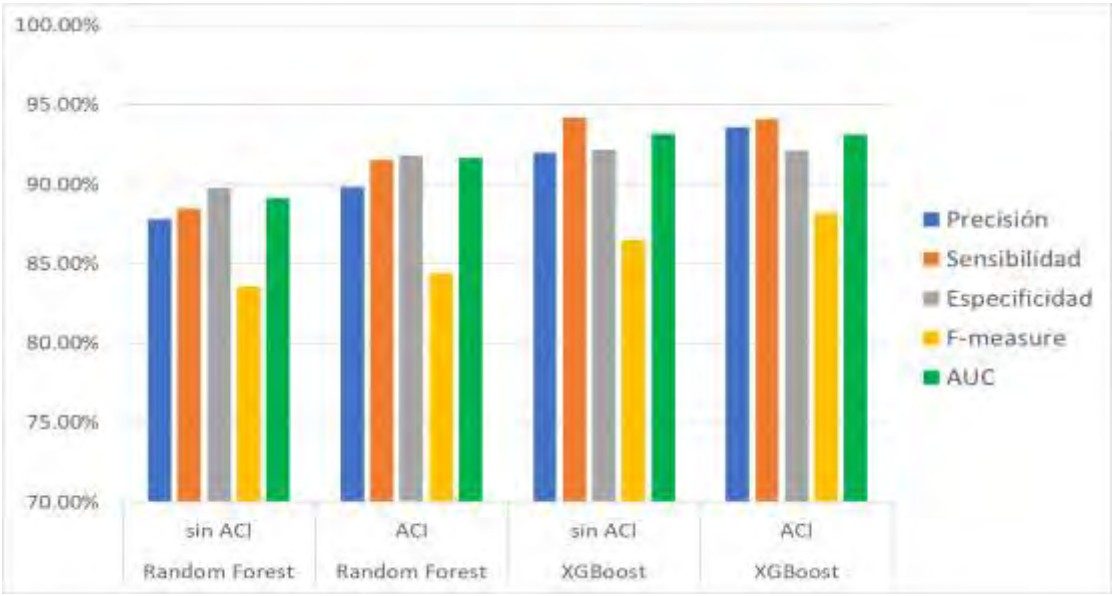


Figura 3. 10 Comparación de los clasificadores [24]

Una vez realizada la evaluación, esta permite identificar los genes con mayor peso en la clasificación. Estos genes podrían ser los marcadores seleccionados con un grado de relevancia como biomarcadores de cáncer de hueso. Es importante destacar que se requiere de la validación biológica y asignación de valores de riesgo para su utilización en la clínica, así como lo muestra la figura 3.11.

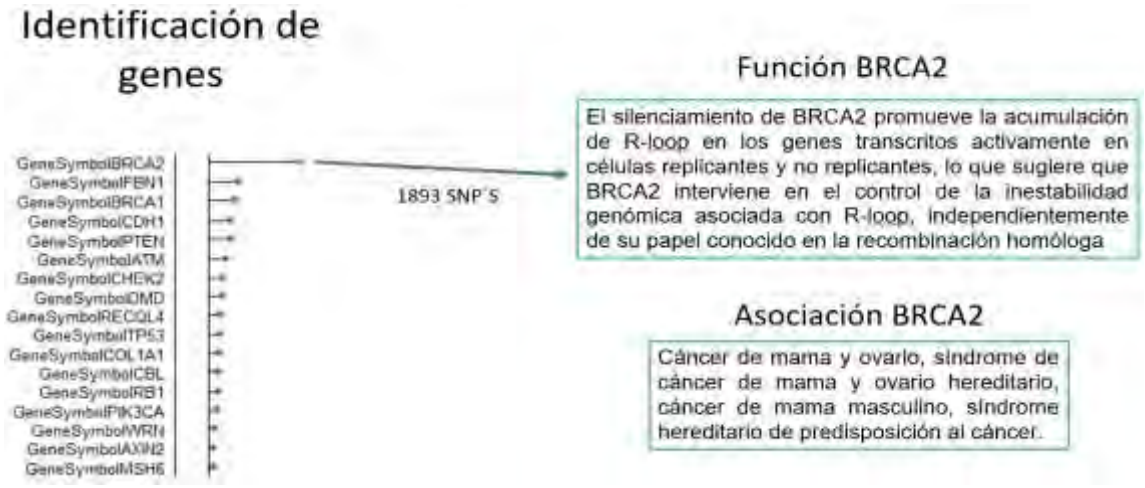


Figura 3. 11 Identificación de genes [24]

Conclusión del reporte

- Se cumplió con los objetivos general y específicos.
- Se obtuvo mejores resultados con XGBoost- ACI, superando en un 3.81% a Random Forest- ACI.
- Se optimizó el proceso del entrenamiento a través de la paralelización, logrando una reducción de 4.03 horas (29.3%) en XGBoost con ACI y 5.23 horas (30.9%) en Random Forest con ACI.

Estos métodos se tomaron en cuenta para el procesamiento de alineación genómico, y generación de los arboles de decisión por medio de coste uniforme.

3.2 Algoritmo de alineación de secuencias para enfermedades del sistema nervioso central [25]

El trabajo muestra un algoritmo que permite la comparación de secuencias de ADN, en enfermedades del sistema nervioso central a partir de tres métodos:

- Alineamiento global
- Alineamiento local
- Alineamiento por triples

Conforma una base de datos que contiene la información necesaria acerca de las enfermedades consultadas. La base de datos se integra de información sobre genomas de varias especies almacenadas en bases de datos como **Genbank, EMBL y CCBJ.**

El alineamiento de secuencias de ADN, es utilizado para procesos como búsqueda de genes o predicción de enfermedades, con el objetivo de hacer diagnósticos, conocer más sobre ellas y encontrar la cura más adecuada. "Esta investigación trata las patologías que afectan el sistema nervioso".

Base de Datos: Para el desarrollo de algoritmos, fue necesario realizar un proceso de recolección de información que permite desarrollar una base de datos genómica.

Para fines del desarrollo del algoritmo, en la figura 3.12, se presenta un modelo "entidad-relación" de la base de datos propuesta que contiene los datos más relevantes para la realización del alineamiento. Esta base de datos se realiza con el motor MySQL

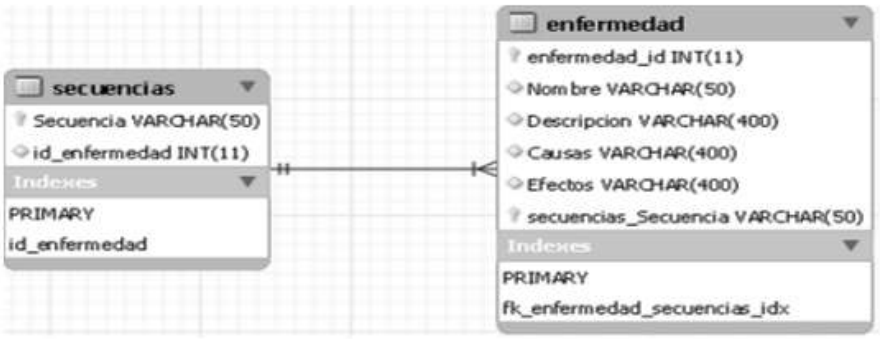


Figura 3. 12 Modelo "Entidad-Relación" [25]

Dentro del algoritmo, se hace una recolección de información, la cual tiene que ver con una secuencia que ingresa el usuario a través de una interfaz gráfica. Dicha frecuencia, ingresa al sistema a través de la interfaz para hacer los alineamientos "**GLOBAL", "LOCAL" y "TRIPLETES".** En la figura 3.13, muestra la interfaz.

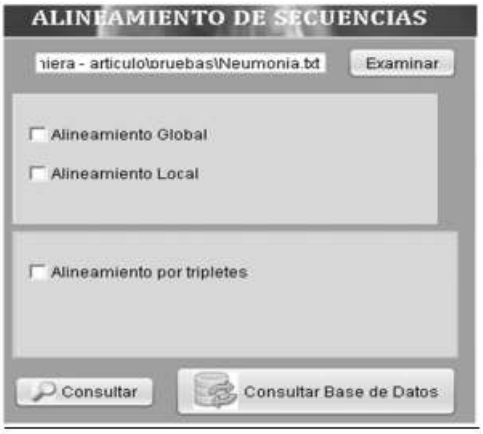


Figura 3. 13 Interfaz gráfica [25]

Alineamiento local y global: La secuencia es leída como una cadena de texto, la cual es procesada y dividida carácter por carácter y cada uno de ellos es ingresado dentro de las posiciones de un vector, lo cual llevará a cabo el alineamiento de una manera sencilla. La secuencia se comparará con las que se encuentran en la base de datos, con el fin de averiguar cuál de ellas tiene mayor similitud con la actual ingresada.

Alineamiento por Triples: En este caso, la comparación se hace igual que el alineamiento **local y global**, sin embargo, las secuencias en este caso se dividen en conjuntos de tres caracteres.

Resultados: En la figura 3.14, se observa cómo los resultados son mostrados por una interfaz, dando a conocer la enfermedad con mayor grado de similitud, sus características y el puntaje obtenido en el alineamiento. Mostrando resultados con el puntaje asignado a cada una de las comparaciones y mostrando cero de puntaje cuando no se presenten coincidencias.

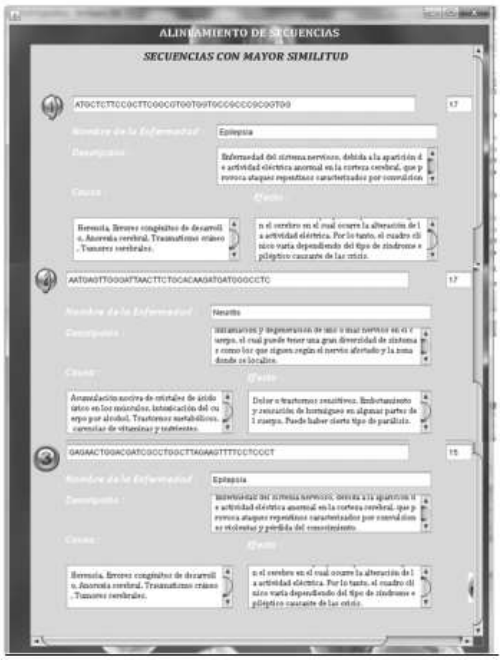


Figura 3. 14 Muestra de resultados [25]

Sin embargo, en los puntajes realizados por comparaciones por **TRIPLES** los resultados pueden mostrarse completamente diferentes, debido al manejo de caracteres y se especifica mucho más la comparación.

3.3 Implementación y Análisis de Algoritmos de alineación para datos next Generation Sequencing (NGS) [26]

El coste del proceso de secuenciación de los genomas de los seres vivos, se ha reducido en gran cantidad en los últimos diez años debido a la aplicación de nuevas técnicas de secuenciación denominadas Next Generation Sequencing (NGS). Esta situación ha propiciado la aparición de muchos alineadores que permiten, dada una serie de secuencias provenientes de la secuenciación NGS, conseguir hallar la posición en el genoma del que proceden usando un genoma de referencia. Sin embargo, es difícil escoger cuál es el alineador que mejor se puede adaptar a cada problema, dada la dificultad de encontrar comparaciones justas entre alineadores en términos de efectividad en la alineación y coste computacional.

El objetivo es el análisis teórico e implementación de tres alineadores para su posterior comparación, determinando en qué casos es mejor optar por la utilización de unos u otros. Adicionalmente, se proporciona un análisis de la influencia de sus metas parámetros en el rendimiento de cada alineador.

Una de las principales vías de investigación en este campo incluye la secuenciación, mapeo y ensamblado de los genomas. Gracias a estas técnicas, se puede determinar la información genética que forma la base de todos los seres vivos.

Este trabajo se centra en el problema de mapeo o alineación. Éste consiste en comparar cada una las lecturas provenientes del proceso de secuenciación con una secuencia de referencia, como un genoma, previamente conocida.

Objetivo:

Arrojar cierto grado de luz a los algoritmos, centrada en tratar de explica el funcionamiento de los algoritmos de estas herramientas y luego llevar a cabo una implementación de cada una de ellas para poder compararlas en términos de efectividad en la alineación de diferentes tipos de lecturas.

Los alineadores actualmente poseen principalmente dos problemas. Por un lado, un sesgo derivado de la autoría de cada herramienta y por otro, la falta de transparencia en la implementación, lo que hace difícil realizar comparaciones justas o realmente cuantificar cuál de los algoritmos es más eficiente si, más allá de la implementación utilizada.

Metodología:

Alineadores analizados:

- **BOWTIE**: Herramientas diseñadas para buscar secuencias cortas de (35-100bp) en el genoma
- **BWA (Burrows Wheeler Alignment):** Es una herramienta de alineamiento del genoma desarrollada por Heng Li y Richard Durbin en 2009. Su principal cualidad es que permite

alinear secuencias cortas frente a un genoma de referencia, pero permitiendo que se puedan producir tanto errores como *gaps* (huecos) en la secuencia.

- **BWT-SW (Burrows Wheeler Alignment – SmithWaterman):** Herramienta que explota las posibilidades de la BWT unida a la programación dinámica, creando una herramienta capaz encontrar todas las alineaciones locales de secuencias largas.
- **FM-Index:** Estructura de datos que permite el indexado y la búsqueda en cadenas de caracteres ocupando un espacio proporcional a su entropía.

Conclusiones:

Este trabajo ha permitido la creación de una herramienta que implementa tres algoritmos de alineación de secuencias NGS basados en el FM-Index. Se ha seguido un desarrollo por etapas, en el que cada etapa se ha centrado en cada uno de los alineadores, teniendo su fase de implementación y su fase de pruebas.

Todo el proyecto se ha realizado en C++, pero la representación de resultados se ha hecho a partir de scripts de *shell* y *awk*, que recogían la salida del programa y la procesaban. Después se ha usado R, en concreto la librería ggplot2 para la generación de gráficas y la librería seaborn de Python para la creación de los mapas de calor (heatmaps).

Trabajos Futuros:

Añadir nuevos alineadores a la comparativa: En este trabajo solo se han tratado tres alineadores, basados todos en el FM-Index. Un trabajo futuro sería incorporar otros alineadores que estén basados en otras estructuras o métodos como tablas hash o alineadores probabilísticos.

Mejora de los algoritmos implementados: Tanto Bowtie como BWA, tienen diferentes formas de ejecutar sus rutinas. La que se ha implementado en este trabajo, revisa todo el genoma para encontrar la mejor alineación. Se podría añadir el resto de las funcionalidades para unas versiones más completas.

Incorporación de diversas métricas de comparación: En este trabajo las comparaciones se han realizado en base a simulaciones de secuencias **NextGenerationSequence**. Sin embargo, en la práctica, no se dispone de una "realidad" sobre la que comparar y, por tanto, las medidas de los verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos no son triviales. Una posible línea de trabajo futuro sería incorporar algunas de estas métricas a la herramienta desarrollada con el fin de poder hacer comparaciones entre alineadores sin necesidad de recurrir a simulaciones.

3.4 Genómica comparada de dos dianas moleculares en modelos animales de hipersensibilidad [27]

La bioinformática es aplicable al diseño de medicamentos, la simulación de efectos biológicos y en la comparación inter-especies de las moléculas implicadas en diversos fenómenos y enfermedades.

Las alergias son un importante y creciente problema de salud, que ha escalado en su magnitud hasta ubicarse incluso entre las primeras causas de muerte.

La selección del bio-modelo para el problema a estudiar depende de múltiples factores e influye luego en la posibilidad de extrapolación al humano de los resultados obtenidos.

Las tecnologías óhmicas pueden ser de utilidad para escoger el animal a usar, en la modelación del evento en cuestión y en la mejor interpretación y comprensión de los datos resultantes. La bioinformática no solo es aplicable al diseño de medicamentos y la simulación de efectos biológicos, sino también en la comparación inter-especies de las moléculas implicadas en los fenómenos estudiados.

OBJETIVOS

General: Comparar dos moléculas claves en los trastornos alérgicos, por medio de herramientas bioinformáticas, entre el hombre y otras especies animales.

Específicos:

- Determinar la utilidad de una metodología basada en herramientas bioinformáticas para la selección de los modelos animales para el estudio de fenómenos alérgicos.
- Describir las similitudes y diferencias a nivel genómico entre el humano y tres modelos animales para las moléculas seleccionadas.

Para este trabajo se utilizaron diferentes herramientas, las cuales son:

- NCBI (https://www.ncbi.nlm.nih.gov/gene). Obtención de datos de los genes
- Ensembl (http://ensembl.org/). Comparación a nivel genómico.
- MUSCLE (http://www.ebi.ac.uk/Tools/msa/muscle/). Alineación Múltiple.
- Clustal2.1. Generación de matrices de identidad
- UCSC Genome Browser (https://genome.ucsc.edu/). Representación gráfica de las alineaciones de secuencia y la existencia de polimorfismos.

RESULTADOS

En el caso del gen de la *IL-4*, se encontró una mayor similitud en términos de composición y ubicación de las secuencias codificadoras entre el hombre y el conejo.

El gen de la molécula humana FcεR1a ha sido ubicado en el brazo largo del cromosoma 1, específicamente en 1q23.2. Si bien en el conejo es menor el número de inserciones y no se aprecian inversiones. Una perspectiva más gráfica de la similitud entre especies se obtuvo para la IL-4 con UCSC Genome Browser. También es de destacar que los polimorfismos mono-nucleotídicos (SNP, de single nucleotide polymorphism) predominan en áreas no codificadoras.

Para dilucidar mejor las similitudes de secuencia entre todas las especies estudiadas, se ejecutaron alineaciones múltiples para ambas moléculas, mostrando resultados en la tabla 3.4.

Tabla 3. 4 Comparación de genes [27]

Especie	IL-4				FcεR1a			
	Conejo	Hombre	Ratón	Rata	Conejo	Hombre	Ratón	Rata
Conejo	100	64,89	55,14	54,15	100	71,14	60,34	59,35
Hombre	64,89	100	61,37	62,97	71,14	100	62,70	63,23
Ratón	55,14	61,37	100	83,27	60,34	62,70	100	81,90
Rata	54,15	62,97	83,27	100	59,35	63,23	81,90	100

Los resultados presentados apoyan la utilidad del conejo como modelo de enfermedades humanas relacionadas con las alergias, a partir de las similitudes para ambas especies de organismos entre dos de las moléculas centrales en los fenómenos de hipersensibilidad.

3.5 Documentación y análisis de los principales Frameworks de arquitectura de software en aplicaciones empresariales [28]

La necesidad imperante del manejo de la información ha permitido el desarrollo de nuevas tecnologías que sean aplicables y adaptables a entornos laborales.

Este trabajo se enfoca en un tema común hoy en día, el cual es la arquitectura del software y su aplicabilidad a través de proyectos de Frameworks de arquitectura de software, siendo más usados en el desarrollo de aplicaciones empresariales.

Este tema, abarca los siguientes conocimientos.

- Uso de arquitectura de software en proyectos de desarrollo.
- Diseño de arquitectura de software
- Aplicaciones empresariales
- Framework de arquitectura de software y su aplicabilidad.

Modelos de semántica:

- **Modelo Estructural:** Compuesto por componentes, conexiones entre ellos y así también la configuración, estilos, restricciones, semántica, análisis, propiedades, racionalizaciones, requerimientos, necesidades de los participantes.
- **Modelo de Framework:** Estructura coherente del sistema completo. Dominio de problemas específicos.
- **Modelo Dinámico:** Destaca la cualidad conductual de los sistemas dinámicos, “cambios en la configuración del sistema o dinámica involucrada”
- **Modelo de Proceso:** Se concentra en la construcción de arquitectura pasos y procesos involucrados en una construcción.
- **Modelos Funcionales:** Conjunto de componentes funcionales organizados en capas que proporcionan servicios hacia arriba, “Visión de un Framework particular”.

Aspectos que lo constituyen:

- Unidad arquitectónica
- Vista/Perspectiva
- Estilos arquitectónicos
- Abstracción
- Proceso arquitectónico

Tipos de arquitectura de software IT

Arquitectura en Capas:

- Interacción entre capas vecinas.
- Puede alojarse en una misma máquina fija.
- Componentes de las capas pueden comunicarse con otras por interfaces bien definidas.
- Pirámide invertida: Cada capa agrega responsabilidades y abstracción a la que se encuentra sobre ella.

Cliente servidor:

- Solo cuenta con dos capas
  - Aplicaciones
  - Bases de Datos
- Tareas repartidas entre los proveedores de recursos, servicios y demandantes.

Tres capas.

- Capa de presentación. Interacción, Usuario y Aplicación.
- Capa de regla de negocio. Comprende la lógica de ejecución.
- Capa de datos. Lógica de comunicación para realizar operaciones en la aplicación.

CONCLUSIONES

La motivación de este trabajo nace de la necesidad de conocer sobre un punto de gran importancia en el ámbito de la informática como es la arquitectura de software, considerando el hecho o la realidad de las empresas u organizaciones en la actualidad, donde todas ellas dependen de un software (sistema empresarial) que le permita realizar la gestión empresarial.

Todo comienza en la definición de una arquitectura que le permita englobar de una forma significante todos los requerimientos y además se encuentra enfocado a las reglas propias del negocio. Es por ello que el punto inicial en un proyecto es la definición de la arquitectura que se empleará, uno de los objetivos que se busca con ello, es obtener como producto una aplicación empresarial que le facilite la gestión con los clientes, pero a su vez le brinde la posibilidad de optimizar sus recursos internos y satisfacer las necesidades del negocio.

En este sentido es importante resaltar la importancia de especificar los requerimientos y realizar el modelado integrado que proporcione los métodos y la tecnología que sea requerida para el diseño e implementación de la arquitectura adecuada.

El éxito de un desarrollo óptimo de una aplicación empresarial depende de la disposición de cada miembro del equipo encargado de la planeación, diseño, desarrollo e implementación de la misma, esta investigación sirve de referencia para la evaluación de los diferentes estilos arquitectónicos que pueden ser utilizados en los desarrollos empresariales.

Se puede concluir, que esta investigación proporciona una guía inicial al lector, sobre el uso de Frameworks de arquitectura de software en proyectos de ingeniería de software empresarial. A partir de los conocimientos adquiridos en esta investigación, el arquitecto de software o ingeniero de software podrá complementar sus conocimientos y aplicar técnicas como procedimientos descritos en esta investigación.

3.6 Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos [29]

La cantidad de datos biológicos disponibles para su análisis se ha multiplicado exponencialmente a lo largo de la última década. Ahora se dispone de datos de naturaleza muy variada.

A medida que crecen en número y variedad la información disponible, esta va dificultando el proceso para conseguir extraer información útil, lo cual hace necesario recurrir a procedimientos automatizados que intenten ayudar en la tarea de analizar los datos.

Objetivos:

- Desarrollar series de análisis sobre los datos de HamMap para detectar las características de estos datos.
- Realizar informe de resultados.
- Familiarizarse con el proyecto de HapMap y sus datos.
- Aplicar varios algoritmos de Machine Learning a los datos preparados, seleccionados entre todas las pruebas realizadas para obtener mejor resultado.
- Realizar una comparación del rendimiento de los modelos generados.
- Recoger los resultados obtenidos en informe.
- Probar la generalización de los análisis implementados a otros datos de estructura similar.

Una parte relevante del trabajo es desarrollar código reutilizable, que facilite la repetición del estudio comparativo descrito en este documento.

Para este trabajo se utilizó el Lenguaje Python, es un lenguaje de programación interpretado multiparadigma. Administrado por la Python Software Fundación con una licencia de código abierto.

Motivos por los que se eligió Python

1- Lenguaje muy utilizado en Machine Learning
2- Lenguaje Multiplataforma
3- Lenguaje más utilizado en el área de Bioinformática
4- Lenguaje muy empleado en el procesamiento de grandes cantidades de datos.

Conclusiones:

La metodología inicial se pudo seguir sin problemas. Se consiguió trabajar y obtener buenos modelos de "Machine Learning" para trabajar con datos de HapMap, aunque se encontraron dificultades en los tiempos de proceso.

3.7 GAIA: Framework Annotation of Genomic Sequence [30]

Este es un prototipo de arquitectura de software que implementa varios elementos para la anotación del marco, constando de una base de datos de anotaciones y un bus de datos del cual define tres conceptos.

1. Entradas (secuencias genómicas)
2. Características (información de interés biológico)
3. Experimentación

La secuencia del Genoma Humano se actualiza 2Mb cada día, debido a esto, los mecanismos utilizados para la anotación deben de ser capaces de mantener un alto rendimiento de secuencia. Un problema que se presenta es que la anotación relacionada es más propensa a errores, pero inicia razonablemente con predicciones que se pueden obtener mediante el uso de algoritmos de coincidencia de patrones. Tales como **GenScan** y **Grail.** Un problema también presentado es agregar nuevos datos al cambiar información subyacente a través del tiempo o por corrección de los métodos analíticos. "La anotación se desvanece y actualiza".

"Cada tipo de anotación debe ser tan completa como sea posible, para que los usuarios tengan acceso a toda la información que pueda producir de manera fiable al igual que poder sacar conclusiones basadas en la ausencia de anotaciones"

Es fundamental indicar claramente la fuente de datos para todos los resultados, de modo que los usuarios puedan seleccionar información útil y evaluar su validez. Esta información debe registrarse en diferentes instancias de datos y secuencias, con esta semántica pueden definirse con mayor detalle, simplificando el proceso de comparar diferentes datos permitiendo a los usuarios generalizar la experiencia de un caso a otro.

Definir estos modelos es el aspecto más difícil de la automatización, de lo cual depende de los diferentes métodos analíticos para filtrar los datos correctamente, requiriendo pruebas cuidadosas en un punto de referencia.

Framework GAIA: Destinado a proporcionar una estructura dentro de los cuales los enfoques pueden implementarse y probarse, siendo rastreados para evaluar el proceso de anotación sobre el tiempo, desarrollando interfaces para proporcionar mejores visualizaciones y recuperación de datos.

Conformado por una base de datos, "CARTA" es un conjunto de sensores autónomos que realizan análisis automáticos para aserciones de resultados en la base de datos. La figura 3.15 muestra una forma más visible de cómo es y cómo trabaja este Framework.

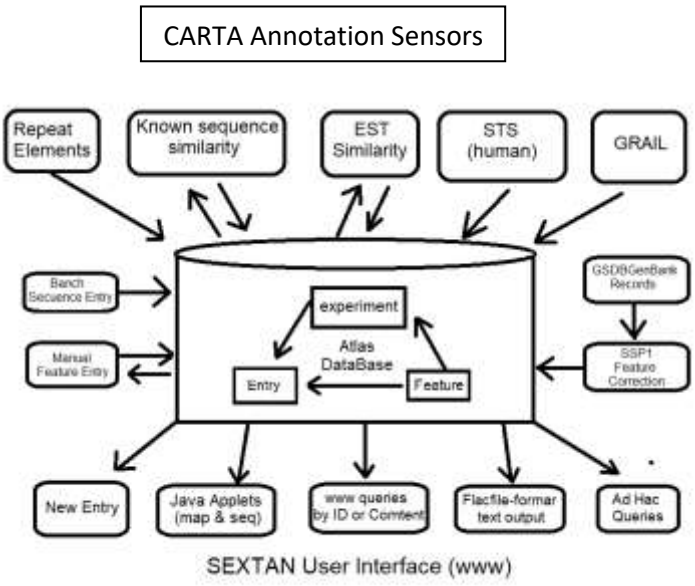


Figura 3. 15 Framework GAIA [30]

CARTA: Responsable de la automatización. Comprende un conjunto de sensores, los cuales realizan análisis específicos y registra sus resultados a través de ATLAS como registros de características y experimentos construidos.

ATLAS: Sistema de gestión de datos. Funciona como un archivo y bus del cual se comunica con otros componentes por medio de un API. Al ser desarrollado para diferentes combinaciones de sensores, los componentes están aislados de los formatos de datos específicos de otros componentes.

SEXTAN: Esta es la interfaz principal de GAIA. Maneja ambas consultas contra los datos actuales y la presentación de nuevas entradas de secuencia, ofreciendo actualmente acceso de solo lectura a la base de datos.

Métodos:

Recursos computacionales. Anotaciones en Bases de Datos. Entradas de secuencias. Intérpretes, Generadores de secuencias. Creadores de repeticiones. Determinadores de patrones. Son algunos de pasos y métodos que utiliza este Frameworks.

Procedimiento.

Se obtienen varias secuencias genómicas obtenidas de bases de datos públicas, integrando en su totalidad secuencias terminadas. Los datos fueron proporcionados por los sensores CARTA y por las bases de datos. Es interesante notar que los registros generados por CARTA cuentan para la mayoría de las funciones como entrada de bases de datos y como nuevas secuencias.

Al comparar las anotaciones con el mapa de transcripción para obtener genes, se encuentra que el sistema automatizado encontró ocho genes. Sin embargo, no es claro si sus resultados representan partes de genes codificadores de proteínas producidas en una región de transcripción activa.

Como se informó anteriormente, la anotación demostró un grupo de tecnologías ecológicamente racionales que no superponen una transcripción conocida. Este grupo proporcionó un modelo de estructura del exón, incluido el empalme alternativo potencial, que se amplió mediante la secuenciación adicional de clones de ADNc asociados con las tecnologías racionales. La búsqueda de similitud de aminoácidos produjo una fuerte alineación a partir de la cual una predicción funcional podría hacerse.

3.8 ALINEAMIENTO DE SECUENCIAS USANDO CLUSTALX [31]

Los científicos analizan el ADN de las especies que recolectan para obtener un "código de barras de ADN" que se usa para identificar a dicha especie. Un ejemplo es el gen mitocondrial de la subunidad I del citocromo oxidasa (COI), el cual codifica parte de una enzima que es importante para la respiración celular, y otro es la NADH deshidrogenasa subunidad 2 (ND2).

COI o ND2 son útiles porque en general muestran poca variación entre los miembros de la misma especie, y la variación de su especie para distinguir a miembros de especies diferentes.

Hay diferentes formatos comúnmente utilizados para representar secuencias de ADN. El formato FASTA empieza con un ">," seguido de la información acerca de la secuencia en la primera línea, seguido de la secuencia de ADN.

¿Qué programa usar para el alineamiento genómico?

- ClustalX es intuitivo, y a su vez es una herramienta excelente para ilustrar el concepto y el proceso del alineamiento de secuencias. ClustalX es gratuito y se puede instalar en el disco duro, lo que es una ventaja (no depende del internet) y también una desventaja (requiere instalación del programa) para su uso en el salón de clases. El algoritmo es un poco anticuado, hay otros programas que son mejores para generar filogenias; sin embargo, ClustalX es apropiado para demostrar cómo generar árboles filogenéticos en base a secuencias de ADN. La filogenia generada requiere de otro programa que también está disponible de forma gratuita, NJplot, para imprimir o para ver los árboles.

- www.Phylogeny.fr es una herramienta en internet para generar filogenias. Usando los parámetros estándar, "phylogeny.fr" es simple de usar, y utiliza un generador de alineamientos llamado MUSCLE. El sitio genera una filogenia que puede ser guardada en diferentes formatos. Sin embargo, la forma en que se muestra el alineamiento de las secuencias no es tan intuitivo como lo es en ClustalX.

Se utilizó ClustalX para comparar las secuencias de ADN. Para interpretar las secuencias alineadas use las secuencias de prueba del archivo "test.txt", el cual contiene secuencias cortas de ADN (test1, test2, y test3) como se muestra en la figura 3.16

“>test1
AAGGAAGGAAGGAAGGAAGGAAGG
>test2
AAGGAAGGAATGGAAGGAAGGAAGG
>test3
AAGGAACGGAATGGTAGGAAGGAAGG”

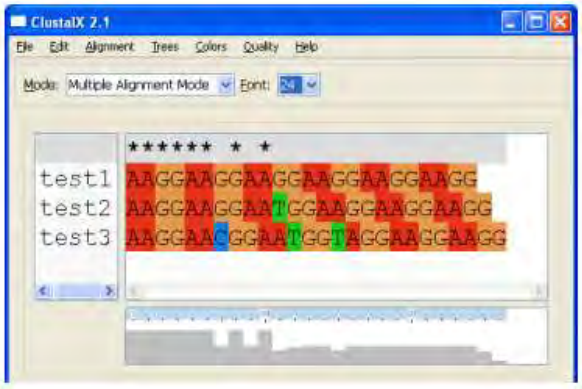


Figura 3. 16 Alineación mediante ClustalX [31]

3.9 Alineamiento gráfico de secuencias a través de programación paralela: un enfoque desde la era post genómica [32]

Una de las tareas comunes en la bioinformática es el alineamiento de secuencias que consiste en una forma de comparar dos o más cadenas de ADN con el fin de mostrar las zonas de similitud para indicar relaciones entre genes.

Los alineamientos son una técnica que se encarga de comparar regiones individuales y globales de secuencias generando como resultado una imagen, denominada “dot-plot” lo que corresponde a un arreglo en dos dimensiones en el que se ubica una secuencia sobre el eje horizontal y verticalmente, en los cuales se asignan puntos que representan el grado de similitud de las secciones de las regiones que se están comparando.

Los *dot plots* se utilizan para evaluar repetitividad en una sola secuencia contra ella misma y las regiones que comparten similitudes aparecerán como líneas fuera de la diagonal principal.

Uno de los softwares tradicionalmente utilizados para la generación de *dot plots* es DOTTER (1995) ya que posee una serie de herramientas que permiten modificar el resultado sin tener que recalcular el alineamiento.

GEPARD (2007) utiliza el método de sufijos para la comparación, logrando mejorar el consumo de tiempo y recursos computacionales, además cuenta con una interfaz gráfica para ingresar, visualizar y realizar anotación de la información.

En el presente artículo, se propone una estrategia que permite analizar y procesar grandes cantidades de datos de forma rápida y eficiente utilizando lenguajes de programación de alto nivel como Python.

Este tipo de estrategias resultan de gran utilidad en las ciencias biológicas ya que permiten de manera visual determinar zonas repetitivas, reorganizaciones y mutaciones, además facilitan la comparación de genomas entre especies diferentes para distinguir rasgos comunes.

Se utilizó **Python** versión 3 como lenguaje de programación, utilizando las librerías **Numpy, Matplotlib, Sys, Time, Getopt y Multiprocessing**, las cuales son esenciales para el manejo de arreglos, gráficos, lectura de archivos, toma de tiempos, lectura de parámetros en línea de comandos y trabajo sobre múltiples procesadores.

El algoritmo de alineamiento gráfico implementado recibe como parámetros dos archivos a analizar en formato FASTA. Inicialmente se unen todas las secuencias presentes en cada archivo para generar una única secuencia, luego se realiza el alineamiento dividiendo la secuencia en cadenas más pequeñas de tamaño definido por el usuario.

El algoritmo utiliza solo el sistema de puntuación de ***match*** **y** ***mismatch*** para comparar localmente las ventanas, asignando un puntaje positivo cuando el par de bases son iguales y negativo cuando el par de bases es negativo. El algoritmo tiene definido el puntaje de **match como +5** y el de **mismatch como -4** como lo sugiere **DOTTER**. La figura 3.17, muestra el pseudocódigo del algoritmo.

Pseudocódigo 1. Función del cálculo del puntaje de alineamiento por *match* y *mismatch*.

```
Enteros:  N1 (Longitud de la secuencia1)
          N2 (Longitud de la secuencia2)
          match_puntaje=5
          mismatch_puntaje=-4
Vectores: V_mismatch = []
Function score (secuencia1, secuencia2)
si   N1 > N2 entonces
     longitud ← N1
sino
     longitud ← N2
fin_si
pos=0
    para i ← 0 hasta longitud; hacer
      si secuencia1[i]! = secuencia2[i]
      entonces
            v_mismatch[pos]← 1
            pos++
    fin_si
fin_para
num_mis ← sumatoria(v_mismatch) + absoluto(N1-N2)
mismatch_valor ← num_mis * mismatch_puntaje
    match_valor ←(longitud - sumatoria(v_mis
    match)) * match_puntaje
    puntaje_total ← mismatch_valor + match valor
Retornar puntaje_total
```

Figura 3. 17 Pseudocodigo [32]

Luego de obtener los resultados de cada alineamiento y guardarlo en una matriz, se mapean los puntajes obtenidos a intensidades entre 0 y 255 siendo el valor más alto 255 y 0 el más bajo. Por

último, se le aplican a la matriz de puntos filtros de reducción de ruido para mejorar la calidad y el detalle del alineamiento realizado.

Los experimentos realizados durante esta investigación incluyen la implementación de un algoritmo usando estrategias paralelas. El algoritmo inicialmente se ejecutó con un procesador y tomando como entrada el cromosoma 21 del Homo sapiens para compararlo contra él mismo, se registró el tiempo de ejecución.

Posteriormente, se ejecutó el algoritmo incrementado el número de procesadores de 2 hasta 64 analizando el cromosoma 21 contra él mismo y registrando los tiempos de ejecución. Finalmente, se calculó el tiempo promedio de ejecución, desviación estándar y la aceleración obtenida para cada número de procesadores; los resultados se encuentran en la tabla 3.5.

Tabla 3. 5 Registro de tiempos del algoritmo

	Tiempo por número de ejecución [S]								
CPUs	*1*	*2*	*3*	*4*	*5*	*Media [S]*	*tiempo [hh:mm:ss]*	*Desviación Estándar*	*Aceleración*
1	6.611	6.696	6.696	6.641	6.639	6.657	1:50:57	33,87	1,000
2	3.340	3.342	3.347	3.338	3.331	3.340	0:55:40	5,24	1,993
4	1.686	1.666	1.684	1.671	1.672	1.676	0:27:56	7,81	3,972
8	869	859	870	869	866	867	0:14:27	4,03	7,681
12	600	619	608	606	613	609	0:10:09	6,43	10,927
16	476	470	470	471	486	475	0:07:55	6,12	14,026
20	399	378	383	381	377	384	0:06:24	7,99	17,353
24	324	328	324	323	323	324	0:05:24	1,85	20,520
32	284	282	276	284	277	281	0:04:41	3,44	23,723
48	270	257	264	262	264	263	0:04:23	4,18	25,272
64	237	238	242	237	239	239	0:03:59	1,85	27,899

De acuerdo con los datos anteriores registrados en la tabla 3.5, se muestra el gráfico de los tiempos promedios obtenidos para los diferentes números de CPUs. De la misma manera, se muestran las aceleraciones del algoritmo que se obtuvieron al dividir el tiempo promedio para cada número de procesadores sobre el tiempo obtenido con un procesador. Adicionalmente, se presenta el comportamiento de los tiempos tomados para cada número de procesadores con un gráfico de caja y bigote en el que está contenida la desviación estándar de los tiempos de ejecución.

Los resultados obtenidos en esta investigación permiten concluir que es posible calcular el alineamiento gráfico dividiendo el *dot plot* en sub-matrices más pequeñas, que son menos complejas de calcular para cada procesador y finalmente, unir estos procesos para generar la matriz de puntajes correspondiente. Se logró disminuir considerablemente los tiempos de ejecución, de horas a minutos, logrando una aceleración de hasta 27.9 veces.

3.10 Múltiple alineamiento de secuencias con los programas en serie Clustal [33]

Las secuencias pueden ser alineadas por su longitud o sus regiones. Los programas más utilizados para las alineaciones son de la serie Clustal, que fueron escritos por Des Higgins en 1988, diseñados específicamente para trabajar eficientemente en computadoras personales.

Estos programas combinan una memoria dinámica eficiente de un algoritmo de programación, con el alineamiento estratégico progresivo desarrollado por Feng y Doolittle y escrito por Willie Taylor. El alineamiento múltiple, es construido por las series de pares de alineamientos, seguidos de ordenar ramificaciones en un árbol guía.

En 1992 un nuevo programa fue lanzado con el nombre ClustalV, el cual incorpora alineamientos dentro de alineamientos existentes, que facilita la generación de árboles a partir de alineamientos múltiples usando el algoritmo Neightbour-Join.

La siguiente generación, llamada ClustalW en 1994, incorpora mejoras de algoritmos de alineación, incluyendo, ponderación de secuencia, penalizaciones de espacio específicas de la posición y la elección automática de una matriz de comparación de residuos adecuada en cada etapa en la alineación múltiple.

Una colaboración entre biólogos y científicos computacionales ha sido la principal razón para el triunfo y continuar generalizando el uso del programa Clustal, llevando a cabo el surgimiento de desarrollos como lo es, con la última versión ClustalX.

Existen otros desarrollos como lo son ClustalNet y DbClustal, los cuales fueron programados para el alineamiento de secuencias conectados a una base de datos que usan información de alineamiento local para anclar alineaciones globales múltiples.

ClustalWWW WEB SERVER: es una interfaz que provee de ayuda e introducción a múltiples alineamientos, para nuevos usuarios. Un factor importante para obtener una alineación de alta calidad es la habilidad para cambiar los parámetros numéricos de alineamientos disponibles en ClustalW. Al igual que los parámetros han sido optimizados para trabajar en la mayoría de los casos, por lo que las secuencias deben de estar en uno de los siete diferentes formatos de lectura (GCG, FASTA, EMBL, GenBank, PIR, NBRF, Phylip or SWISS-PROT).

El resultado de alineamientos múltiples puede ser mostrado con textos de diferentes colores o solo en blanco y negro. La alineación consiste en cuatro dominios de unión de oxidorreductasa NAD. La coloración de residuos se realiza según criterios fisicoquímicos resaltando las posiciones conservadas en las secuencias. También se muestra una línea de consenso debajo de la alineación con los siguientes símbolos que indican el grado de conservación observado en cada columna: ‘*’ (residuos idénticos en todas secuencias), ':' (columna altamente conservada), '.' (columna débilmente conservada).

Tanto ClustalW como ClustalX son activamente mantenidos y actualizados, los cuales han incluido la posibilidad de guardar alineaciones y arboles genéticos en un formato NEXUS para la compatibilidad entre programas.

La última versión del programa contiene cuatro principales modificaciones, siendo la primera, el guardar alineamientos múltiples en un formato *fasta*. Otro es el proveer una matriz de identificación porcentual. La tercera, siendo una nueva opción de posibilidades de guardado en el rango de residuos del archivo de salida al guardar un rango especifico de usuario de alineación.

3.11 El diagrama, un método para comparar secuencias [34]

El método consiste en comparar las secuencias en pares. El resultado de la comparación de dos secuencias, son escritas en una matriz rectangular. Existen varias formas de testeo, de las cuales se utilizaron dos:

- La frecuencia de longitud de todas las diagonales intactas donde se reconocen los “Matchs” mediante la comparación calculada directamente.
- El número total de “matchs” en cada diagonal encontrada en el diagrama.

En la figura 3.18 se muestran tres de los diagramas obtenidos mediante la alineación del Cytochrome Humano con el cytocrome del Mono, Pez y la bacteria Rhodospirillum rubrum. El mono y el pez se alinean con el humano de diferentes maneras, pero lo hacen solo cuando se lleva a cabo la alineación completa. Por otro lado, la alineación entre el humano y rumbru, tienen diferentes partes de una secuencia que no representan a otra.

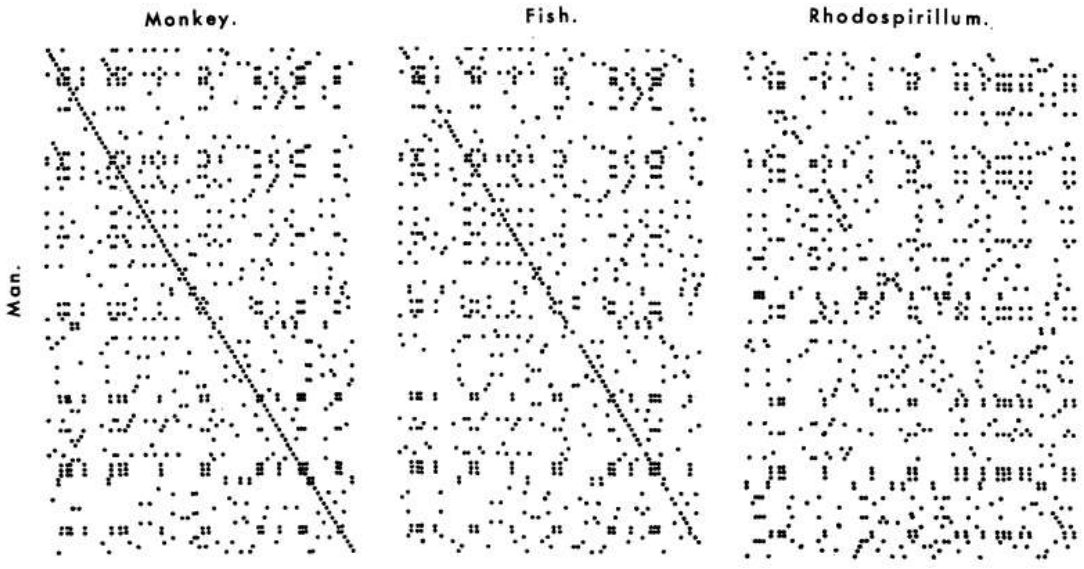


Figura 3. 18 Diagrama de alineación, Humano, Mono, Pez y Rhodospirillum [34]

La tabla 3.6, muestra el número de “Matchs” en la diagonal principal y la tabla 3.7 muestra el total de números de “Matchs” y 35 alineaciones adyacentes diagonales encontradas en el diagrama, listadas en la tabla 3.7. La línea homologa, puede estar en más de una diagonal, indicando nuevas partes de la secuencia que no están presentes en la original.

Tabla 3. 6 Matchs en diagonal principal [34]

Other cytochromes	Run length																							
	0	1				5					10									20		30		50
Monkey	680	36	2	2																			1	1
Rattlesnake	673	43	3	3		1					1						1					1		
Dog	692	36	7	3			1			1			1		1			1			1			
Whale	684	41	5	2			1			1				1				1			1			
Kangaroo	684	45	3	2	1					2				1		1						1		
Chicken	687	33	5	1	1		3													1		1		
Bullfrog	683	41	4	3	1		4			1										1				
Fish	636	44	4	3	3	1				1									1					
Silkworm	641	35	4	1	2	2			1	1				1										
Wheat	651	40	2	1	1	1				1	1	1												
Saccharomyces	672	49	7	1	2	1		1			1													
Rhodospirillum	729	47	6	1	1																			
Pseudomonas	482	21	2	1																				
Random*	738	58	5	0.4																				

Tabla 3. 7 Matchs en diagonales adyacentes

Table 2. *Sums of matches in the principal and adjacent diagonals of diagrams comparing human cytochrome c with other cytochromes*

The principal diagonal is marked X. In each diagram the human cytochrome c is along the left margin of the diagram, and the other along the upper margin

									Man						X											
9	12	4	6	4	3	5	7	7	4	9	7	6	7	9	103	8	7	6	7	8	4	7	7	6	3	Monkey
8	10	4	6	5	5	5	6	8	4	7	6	6	7	8	90	10	10	8	6	8	3	7	8	4	4	Rattlesnake
8	13	5	6	3	3	7	7	8	4	9	9	6	6	7	93	8	5	7	8	9	4	6	8	5	5	Dog
8	12	5	7	4	3	7	7	8	4	9	9	6	6	7	94	9	6	6	7	9	4	6	8	6	4	Whale
8	12	5	7	4	3	7	7	7	4	8	8	6	7	7	94	9	7	5	7	9	4	6	8	5	4	Kangaroo
8	12	5	7	4	3	6	8	7	5	9	7	6	6	7	91	7	6	6	8	9	4	7	8	7	4	Chicken
9	12	6	6	4	3	6	7	8	5	9	7	6	6	7	86	8	6	7	8	10	5	5	9	5	4	Bullfrog
8	12	5	6	4	5	8	6	7	4	6	7	5	6	7	83	7	5	6	6	9	5	7	10	4	2	Fish
8	10	11	3	4	6	5	5	7	4	6	9	4	4	10	39	6	5	5	45	5	6	9	5	4	3	Silkworm
5	3	7	5	7	4	3	6	7	9	4	5	5	1	8	7	11	6	6	6	6	8	7	69	8	8	Wheat
5	6	6	5	7	6	9	6	5	8	4	6	9	11	7	8	5	8	7	11	64	7	6	4	6	7	*Saccharomyces*
8	8	6	8	6	12	8	3	5	4	9	9	9	8	8	21	8	8	12	9	8	10	6	5	6	9	*Rhodospirillum*
4	2	5	5	5	5	5	4	1	8	12	5	7	7	3	4	8	6	6	3	4	5	3	2	6	11	*Pseudomonas*
7·2	7·2	7·3	7·4	7·5	7·6	7·6	7·7	7·8	7·9	8·0	8·0	8·1	8·2	8·3	8·4	8·3	8·2	8·1	8·0	8·0	7·9	7·8	7·7	7·6	7·6	Random*

Se hicieron dos pruebas para proveer coeficientes similares que puedan ser usados para clasificaciones:

a) Un índice se derivó de cada diagrama de las diagonales obtenidas a partir de los Matchs y de las diagonales esperadas si las secuencias habían tenido o no similitud. Esto fue calculado considerando todos los pares comparados. La máxima puntuación obtenida fue de 1.997 puntos entre el pato y el pollo y siendo la mínima puntuación de -0.0826 entre el mono y la vaca.
b) El índice diagonal de similitud se obtuvo de cada diagrama mediante el cálculo del total del número de Matchs en cada diagonal observada en las matrices. Los demás Matchs en el diagrama fueron resultados aleatorios, que se obtuvieron mediante el desarrollo del diagrama de una secuencia escogida aleatoriamente y comparada consigo misma.

El índice de diagonales máximo obtenido fue de 0.9692 entre los citocromos de pollo y pingüino, y el mínimo – 0.0254 entre pollo y *P*, fluorescentes citocromos. La distancia entre cada par de citocromos se calculó como 1.2000-diagonales

Las dos clasificaciones son similares; sin embargo, existen sus diferencias:

a) Las clasificaciones muestran la estrecha similitud entre serpiente de cascabel y citocromos de primates, esto debe hacer que uno se cuestione seriamente el valor de interpretar clasificaciones de este tipo estrictamente en términos de filogenia.
b) Las relaciones entre el atún, el gusano de seda y los citocromos del gusano barrenador son diferentes en las dos clasificaciones. En la clasificación del "índice de diagonales", el gusano barrenador se clasifica con los peces y los vertebrados, mientras que el gusano de seda se clasifica con los demás no vertebrados, mientras que en la clasificación del "índice de carreras" los dos citocromos de insectos se clasifican juntos y grupo con los no vertebrados.

Las relaciones entre pollo, pato y los citocromos de los pingüinos también son diferentes en las dos clasificaciones por las mismas razones. Los dos índices de similitud que han utilizado podrían tal vez combinarse para dar un solo índice de similitud para su uso en la clasificación.

Repeticiones dentro de una Secuencia: Homología Interna

Desde Smithies, Connell y Dixon mostraron que la 2a-haptoglobina humana tiene una secuencia, las dos mitades de las cuales son casi idénticas entre sí (figura 3.19) y también con la secuencia de Inhaptoglobina. Se ha puesto de moda examinar el aminoácido en secuencias de proteínas para secuencias repetidas y, ahora hay muchos informes de tales repeticiones.

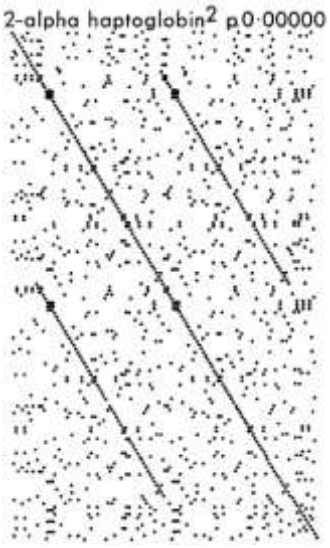


Figura 3. 19 El diagrama obtenido comparando la 2-alfa haptoglobina consigo misma [34]

El método del diagrama se puede utilizar para comparar una secuencia consigo mismo, y por lo tanto detectará repeticiones; para estos se mostrarán como series de coincidencias en diagonales distintas de la diagonal principal.

El método del diagrama confirma lo informado: repeticiones en las ferredoxinas bacterianas. Además, sugiere que puede haber repeticiones en la globina de lamprea. El método del diagrama no detectó repeticiones reportadas en las cadenas pesada y ligera de inmunoglobulina, clupeínaz humana, ni tampoco detecta repeticiones de secuencias de otras 16 diferentes proteínas.

Los diagramas obtenidos al comparar entre sí los citocromos de Neurospora y Rhodospirillum y la hemoglobina alfa humana muestran que la parte de las secuencias que se afirma que muestran homología no dan carreras diagonales más obvias de coincidencias que otras partes de las secuencias.

La figura 3.20 muestra el diagrama obtenido con la secuencia reportada recientemente por Adams, Jeppesen, Sanger y Barril. La secuencia se registra a lo largo del borde izquierdo del diagrama iJhe, la secuencia complementaria, con el término 5' de cada secuencia junto a la esquina superior izquierda de la matriz. Las filas diagonales de coincidencias indican secuencias en común. Marcado en el diagrama de la figura 3.20 es la línea bien definida de posibles bases-*ptirings* anotadas por Adams y esto indica cómo el diagrama puede ser interpretado.

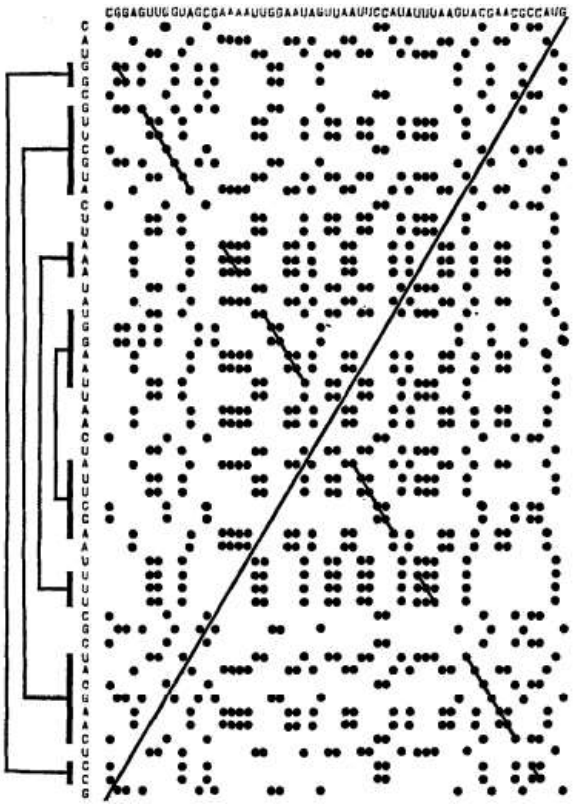


Figura 3. 20 diagrama secuencia reportada por Adams [34]

En la posición del espejo, en la otra mitad del diagrama, un par de coincide con la secuencia complementaria La siguiente secuencia de nucleótidos marcada es opuesta la secuencia -G-U-U-C-G-U-A, y en el espejo la posición es la serie de partidos opuestos a la secuencia complementaria -C-A-A-G-C-A-UK. Estos y las otras secuencias complementarias anotadas se indican entre paréntesis a la izquierda lado del diagrama y está claro que, aunque los emparejamientos de bases son posibles entre otras partes de la secuencia, la línea marcada es, con mucho, la más completa y es también simétrica, permitiendo así una estructura helicoidal.

Este uso del método del diagrama es, creemos, un complemento útil para el algo más complejo métodos para determinar la posible posición de apareamiento de bases en ácidos nucleicos sugerido.

3.12 Colonia de Abejas Artificiales (ABC) Algoritmo de optimización para resolver problemas de optimización con restricciones [35]

El problema considerado es la optimización de dos funciones, donde su objetivo es generar problemas de optimización restringida para poder encontrar una variable “X”.

Karaboga ha descrito el algoritmo de Colonia de Abejas Artificiales basado en el comportamiento de búsqueda de alimento de abejas para la optimización de problemas numéricos, el cual extiende la posibilidad de solución de problemas de optimización restringida.

Colonia de abejas artificiales:

El algoritmo consiste en tres tipos de abejas: Abejas empleadas, abejas desempleadas y exploradoras. Para cada alimento existe una abeja empleada.

En este algoritmo, la posición de la comida representa una posible solución para la optimización de problemas y la cantidad de alimento corresponde a la calidad asociada a la solución. El número de abejas empleadas y desempleadas es igual al número de soluciones en la población.

En la primera parte, el algoritmo genera una distribución aleatoria con una población inicial de soluciones, donde se denota la población. Cada población es un vector D-dimensional. Después de la inicialización de las soluciones, el algoritmo este sujeto a repetir ciclos de búsqueda con las abejas generadas.

Las abejas empleadas producen modificación en la posición en su respectiva memoria, dependiendo de la información local y probando la cantidad de alimento en nuevas fuentes.

Después de que todas las abejas completen el proceso de búsqueda, comparten la información de la cantidad de alimento y la posición en la que se encuentran con las abejas desempleadas, quienes evalúan la información y eligen mediante probabilidades, la mejor fuente de alimento.

En el algoritmo, existen cuatro parámetros de control utilizados.

1- Número de fuentes de alimento
2- Número de abejas, empleadas y desempleadas
3- Valores límites de las fuentes de alimento
4- Número máximo de ciclos en el algoritmo

A continuación, se muestra el pseudocódigo del algoritmo ABC.

1- Inicializar el número de abejas
2- Evaluar la población
3- Ciclos = “n”
4- LOOP
5- Búsqueda de soluciones con las abejas empleadas y desempleadas
6- Aplicación de proceso de selección

7- Calcular probabilidad de mejor fuente de alimento
8- Abejas desempleadas producen nuevas soluciones
9- Aplicación de proceso de selección nuevamente
10- Determinar si se abandonará una solución o si será reemplazada
11- Memorizar la mejor solución
12- Ciclo = Ciclo+=1
13- Repetir hasta que Ciclo == Abejas

3.13 Optimización mediante el algoritmo de colonia de abejas artificial [36]

El Algoritmo de la colonia artificial de abejas (ABC) es uno de los algoritmos más recientes en el dominio de la inteligencia colectiva. Propuesto por Dervis Karaboga en 2005, basado en el comportamiento de forrajeo de las abejas.

ABC es un algoritmo de optimización inspirado en poblaciones, donde las soluciones del problema de optimización, llamadas fuentes de alimento, son modificadas por las abejas artificiales, que se desempeñan como operadores de variación. El objetivo de estas abejas es descubrir las fuentes de alimento con mayor néctar.

El proceso de búsqueda es un proceso de optimización, y el comportamiento de estas se modeló como una heurística de optimización basada en el modelo biológico.

- Fuente de alimento: el valor de una fuente de alimento depende de muchos factores, como su proximidad a la colmena, riqueza o la concentración de la energía y la facilidad de extracción de esta energía. Es resumido en un valor numérico que indica su potencial.
- Abejas recolectoras empleadas: están asociadas a una fuente de alimento. Llevan con ellas información sobre esa fuente en particular, su distancia, ubicación y rentabilidad para compartirla a las abejas observadoras.
- Abejas recolectoras desempleadas: este tipo de abejas se encuentran buscando fuentes de alimento para explotar. Hay dos tipos:
- Exploradoras: se encargan de buscar nuevas fuentes de alimento en el ambiente que rodea a la colmena. Es decir, lleva información sobre una fuente específica y la comparte con otras abejas esperando en la colmena. La información incluye la distancia, la dirección y el néctar de la fuente de alimento.
- Observadoras: Con la información compartida por las empleadas o por otras exploradoras en el nido, buscan una fuente de alimento.

El intercambio de información es el suceso más importante en la formación del conocimiento colectivo, mediante esta interacción, las abejas, decidirán el comportamiento que debe llevar la colmena.

La figura 3.21 muestra a las abejas desempleadas asignadas a una fuente de alimento. Se observa la comunicación de información de las fuentes de alimento, donde las abejas observadoras visitan

las fuentes de alimento más prometedoras. Y por último las abejas exploradoras buscan nuevas fuentes.

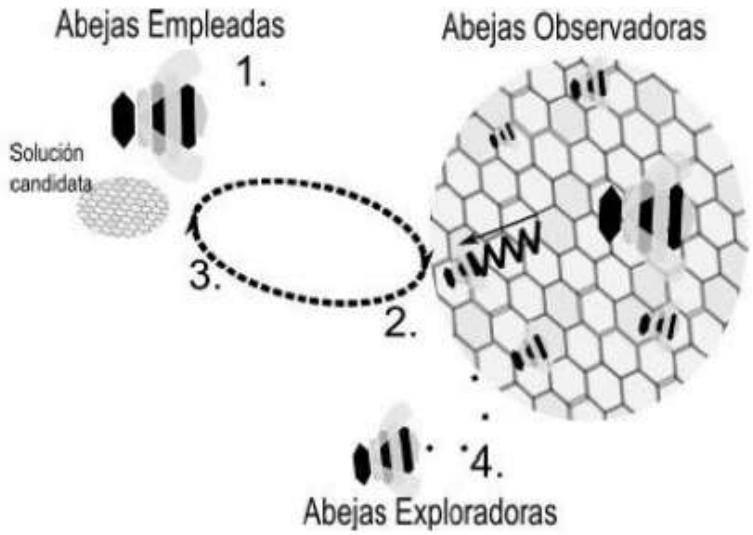


Figura 3. 21 Comportamiento de la colmena de abejas [36]

Comportamiento artificial

Generación de las fuentes de alimento: Se lleva a cabo de manera aleatoria, con base en los límites de cada variable. Una fuente de alimento es una solución al problema de optimización.

Abejas empleadas: Su número es proporcional al número de fuentes de alimento. Su función es evaluar y modificar las soluciones actuales para mejorarlas. Si la nueva posición no es mejor entonces se mantiene la posición actual.

Abejas observadoras: Su número es proporcional al número de fuentes de alimento. Estas abejas seleccionarán una fuente de alimento, de acuerdo con la información que comparten las abejas empleadas.

Abejas exploradoras: Estas abejas crean una nueva fuente de alimento de manera aleatoria, para reemplazar fuentes existentes que no han sido mejoradas.

Límite: Determina el número máximo de ciclos que una fuente de alimento puede persistir sin mejorar antes de ser reemplazada. El límite se incrementa a partir de que una fuente que no es modificada por las abejas, ya sean empleadas u observadoras, hasta obtener su valor máximo permitido.

A continuación, se presenta el pseudocódigo del algoritmo ABC

1- Inicializar la población
2- Loop
3- Colocar abejas empleadas en el alimento
4- Colocar abejas observadoras en fuentes de alimento dependiendo de la cantidad de néctar
5- Enviar abejas exploradoras a zona de búsqueda

6- Memorizar la mejor fuente de alimento encontrada
7- Repetir, hasta cumplir requisitos de terminación

En el algoritmo ABC cada ciclo de la búsqueda consiste en tres pasos:

- El envío de las abejas empleadas a las fuentes de alimento y evaluar sus cantidades de néctar.
- Selección de las fuentes de alimento por parte de las observadoras después de compartir la información de las abejas empleadas y determinar la cantidad de néctar de las fuentes de alimento.
- La determinación de las abejas exploradoras y luego enviarlas a posibles fuentes de alimento en forma aleatoria.

En la fase de inicialización, las abejas empleadas seleccionan al azar un conjunto de posiciones de las fuentes de alimento y se determinan sus cantidades de néctar. Estas abejas comparten la información de néctar de las fuentes de alimento con las abejas observadoras. Después de compartir la información, cada abeja empleada retorna a la zona de la fuente de alimento visitada previamente, continuación elige una nueva fuente de alimento por medio de la información.

Una abeja observadora elige una fuente de alimento dependiendo de la información distribuida por las abejas empleadas. Si la cantidad de néctar de una fuente de alimento aumenta, entonces también aumenta la probabilidad con la que la fuente de alimento es elegida. Después que la abeja observadora llega a la zona seleccionada, elige una nueva fuente de alimento. Esta selección se realiza comparando las posiciones de las mismas en forma visual. Cuando una fuente de alimento es abandonada por las abejas, una abeja exploradora determina una nueva fuente de alimento en forma aleatoria sustituyendo a la abandonada. En cada ciclo una exploradora sale a buscar una nueva fuente de alimento como máximo.

El número de las abejas empleadas o de las abejas observadoras es igual al número de soluciones en la población. Una abeja empleada u observadora produce probabilísticamente una modificación de la posición en su memoria para encontrar una nueva fuente de alimento. La producción de nuevas fuentes de alimento se basa en un proceso de comparación de las fuentes de alimento. la producción de una nueva posición de la fuente de alimento se basa en un proceso de comparación de las posiciones de las fuentes de alimento.

Después de que todas las abejas empleadas completan el proceso de búsqueda, comparten la información del néctar de las fuentes de alimento donde la abeja observadora evalúa esa información de todas las abejas empleadas y elige una fuente de alimento con una probabilidad relacionada con su valor de néctar. Las fuentes de alimento representan a cada solución como un vector n-dimensional.

Los parámetros del algoritmo son los siguientes:

- Fuentes de alimento
- Número total de ciclos que se ejecutaran

- Número de ciclos que será conservada una solución sin mejorar antes de ser reemplazada por una nueva solución

Variante de Colonia de Abejas Artificiales

La propuesta se basa en introducir una variante al algoritmo original donde se modifica este mecanismo de selección que realizan las abejas observadoras, consiste en utilizar un mecanismo de selección por torneo.

Existen dos versiones de selección mediante torneo:

- Determinista
- Probabilística

En la primera versión se selecciona al azar un número de *t* de fuentes de alimento, por lo que generalmente se escoge *t=2* (torneo binario). Luego entre las fuentes de alimento seleccionadas se elige la de mayor fitness como para utilizar como fuente de alimento

En la versión probabilística, la diferencia radica en el paso de selección de la fuente de alimento ganadora del torneo. En lugar de escoger siempre la mejor, se genera un número aleatorio en el intervalo [0,...,1], si es mayor que un parámetro se escoge la fuente de alimento de mayor néctar y en caso contrario, la de menor cantidad de néctar.

Variando la cantidad de soluciones que participan en cada torneo se puede modificar la presión selectiva. Un caso particular se trata de un torneo en el que participan todos los individuos de la población con lo cual la selección se vuelve totalmente determinística. Cuando el tamaño del torneo es reducido, la presión selectiva disminuye y las peores soluciones tienen más oportunidades de ser seleccionadas.

Descripción de la variante del algoritmo ABC

El pseudocódigo del ABC en este trabajo se muestra en el siguiente pseudocódigo.

1- Lectura de parámetros
2- Etapa de inicialización
3- Ciclos
4- Loop
5- Generación de abejas empleadas
6- Proceso de abeja observadora
7- Proceso de abeja exploradora
8- Evaluar y memorizar las fuentes de alimento actuales y mejores
9- Ciclo= Ciclo+=1
10- Máximo ciclo

Procedimiento de Generación de abejas empleadas Pseudocódigo

1- Loop
2- Posiciones de la fuente de alimento
3- Criterios del manejo de los limites
4- Evaluar limites

5- Hasta – No. De fuentes de alimento
6- Actualizar posiciones
7- End

Procedimiento de Proceso de abeja observadora

Aquí se producen nuevas soluciones por las abejas observadoras bajo uno de los siguientes métodos de selección, torneo y ruleta.

Pseudocódigo de abeja observadora

1- Loop
2- Elegir fuente de alimento siguiendo los métodos de selección (torneo/ruleta)
3- Buscar en posiciones de la fuente actual de alimento la solución actual
4- Aplicar criterios para limites
5- Evaluar posiciones de alimento
6- Hasta = Numero de fuentes de alimento
7- Actualizar fuentes de alimento
8- End

Pseudocódigo de abeja exploradora

1- Loop
2- Si el límite permitido es superior
3- Inicializar limite en forma aleatoria
4- Evaluar posiciones de alimento
5- Hasta = número de fuentes de alimento
6- End

3.14 Colonia de abejas artificiales y optimización por enjambre de partículas para la estimación de parámetros de regresión no lineal [37]

El método heurístico Colonia de Abejas Artificiales (ABC) como el método de Optimización por Enjambre de Partículas (PSO), se consideran dentro de los métodos de inteligencia de enjambre.

El objetivo de este trabajo es presentar las heurísticas ABC y PSO para resolver el problema de encontrar los valores de los parámetros en el problema de regresión no lineal utilizando el criterio de mínimos cuadrados.

La regresión no lineal se denota por $x = (x_1, x_2,...,x_m)$ a las variables explicativas e *y* a la variable a explicar, todas observadas sobre *n* objetos, $y = f(x, \theta) + $ " es la relación funcional entre *x* e *y*, donde *f* es en general una función no lineal y $\theta = (\theta 1, \theta 2,...,\theta n)$ es el vector de parámetros. Se quiere minimizar la ecuación 3.1 en la norma euclidiana usual:

$$S(\theta) = |\, y - f(x, \theta)|^2 = \sum_i [y_i - f(x_i, \theta]^2$$

Ecuación 3.1 – ecuación a minimizar [37]

Se denota D como D = " i[θminj , θmaxj], θminj < θmaxj , i = 1, 2 ...,p, y θ∗ = argminθ∈DS(θ) el óptimo buscado.

PSO se basa en el uso de un conjunto de partículas o agentes que corresponden a estados de un problema de optimización, los agentes se comunican entre sí, y entonces el agente con una mejor posición, influye en los demás atrayéndolos hacia él.

La población se inicializa asignando a las variables una posición y una velocidad de manera aleatoria. En cada iteración, la velocidad de cada partícula es aleatoriamente acelerada hacia su mejor posición y a través de las mejores posiciones de sus vecinos.

Se propone utilizar PSO con un manejo dinámico de las partículas, lo que permite romper ciclos y diversificar la búsqueda. En la figura 3.22 se proporciona el seudocódigo de PSO

1. Crear una población de partículas distribuidas en el espacio factible.
2. Evalúe cada posición de las partículas de acuerdo a una función objetivo
3. Si la posición actual de una partícula es mejor que las previas, actualícela.
4. Determine la mejor partícula
5. Actualice las velocidades de las partículas $j = 1, 2, \ldots, r$.
6. Mueva las partículas a sus nuevas posiciones
7. Vaya al paso 2 hasta que el criterio de finalización se satisfaga.

Figura 3. 22 Pseudo código de PSO [37].

Los resultados de los algoritmos propuestos se obtuvieron mediante la cantidad del número de dígitos duplicados (cuando se compararon con los resultados certificados) proporcionados en NIST (2001), los cuales se encontraron utilizando algoritmos deterministas iterativos. El número de dígitos duplicados denotados por λ puede ser calculado vía el logaritmo del error relativo, calculado en la ecuación 3.2:

$$\lambda = \begin{cases} 0 & \text{si } \frac{|w-c|}{|c|} \geq 1, \\ 1 & \text{si } \frac{|w-c|}{|c|} < 1 \times 10^{-11}, \\ -\log_{10}(\frac{|w-c|}{|c|}) & \text{de otra forma.} \end{cases}$$

Ecuación 3.2 – logaritmo de error relativo [37]

Donde *c* denota el valor certificado y *w* denota el valor estimado por el algoritmo propuesto. De acuerdo con NIST (2001).

Un buen procedimiento por mínimos cuadrados no lineal es el que permite duplicar 4 o 5 dígitos de los valores certificados. En este trabajo se presentan los resultados considerando el valor de λ obtenido mediante la ecuación 3.3:

$$\lambda = -\log_{10}(\frac{|w-c|}{|c|}).$$

Ecuación 3.3 – logaritmo de error [37]

Conclusiones

En general, la utilización de ABC no proporciona tan buenos resultados como PSO debido a que PSO tiene la ventaja de poder salir con mayor facilidad de regiones sub optímales, sin embargo, ABC proporciona intervalos más compactos. La gran ventaja tanto de ABC como de PSO, es su fácil implementación y sus cortos tiempos de ejecución.

3.15 El algoritmo “Artificial Bee Colony” (ABC) y su uso en el Procesamiento digital de Imágenes [38]

Durante la última década se ha presentado un crecimiento sostenido en el campo de los algoritmos bio inspirados de cómputo evolutivo para la búsqueda y optimización.

Los algoritmos bio-inspirados consideran el fenómeno de inteligencia en enjambre como fuente de inspiración.

Karaboga presentó en 2005 un algoritmo de enjambre de abejas para resolver problemas numéricos de optimización conocido como el método “artificial bee colony”. Inspirado por el comportamiento biológico de las colonias de abejas en su búsqueda por alimento.

En este trabajo se presenta un enfoque alternativo de segmentación basado en el algoritmo de optimización ABC. El histograma de una imagen es aproximado a través de la mezcla de un conjunto de funciones gaussianas, las cuales representan a cada una de las clases.

Algoritmo artificial bee colony (ABC)

Define un conjunto de operaciones que asemejan algunas características del comportamiento de las abejas. Cada solución dentro del espacio de búsqueda incluye un conjunto de parámetros que representan las posiciones de las fuentes de alimento. El valor de “afinidad” hace referencia a la calidad de la fuente de alimento. El proceso de optimización imita la búsqueda de las abejas por fuentes importantes de alimento dando como resultado un proceso análogo a encontrar soluciones óptimas.

Inicialización de la población

El algoritmo comienza inicializando Np fuentes de alimento para las abejas obreras; cada fuente de alimento simboliza un vector de D elementos que representa las variables de decisión, las cuales son aleatoriamente determinados entre los límites inferiores *low j x y* superiores *high j x* previamente definidos en la ecuación 3.4.

$$x_{j,i} = x_j^{low} + \text{rand}(0,1) \cdot (x_j^{high} - x_j^{low});\ j = 1,2,\ldots,D;\quad i = 1,2,\ldots,N_p$$

Ecuación 3.4 – Inicialización de población de abejas [38]

Siendo *j* e *i* los índices del parámetro y población respectivamente. Por lo tanto, j i, x es el j-ésimo parámetro del i-ésimo individuo.

Enviar abejas obreras

En esta operación cada abeja obrera genera una nueva fuente de alimento en la vecindad de su posición actual descrita en la ecuación 3.5:

$$v_{j,i} = x_{j,i} + \phi_{j,i}(x_{j,i} - x_{j,k});\ k \in \{1,2,\ldots,N_p\};\ j \in \{1,2,\ldots,D\}$$

Ecuación 3.5 – Envío de abejas obreras [38]

Donde $x_{j,i}$ es un parámetro *j* seleccionado aleatoriamente del i-ésimo individuo y *k* es una de las Np fuentes de alimento, satisfaciendo la condición $i \neq k$. Si un parámetro dado de la solución candidata

$v_{j,i}$ excede sus límites predeterminados, ese parámetro debe ser ajustado de manera tal que se encuentre en el rango definido. El factor de escalamiento $\phi_{j,i}$ es un número aleatorio entre [-1 1]. Una vez que una nueva solución ha sido generada, se calcula su calidad mediante una función objetivo. La calidad fit_i de una solución candidata $v_{,i}$ en el contexto de ABC para un problema de minimización se asigna a través de la ecuación 3.6:

$$fit_i = \begin{cases} \dfrac{1}{1+J(v_i)} & \text{if } J(v_i) \geq 0 \\ 1+\left|J(v_i)\right| & \text{if } J(v_i) < 0 \end{cases}$$

Ecuación 3.6 Función de calidad [38]

$J(v_i)\cdot$ es la función objetivo a ser minimizada. Si la cantidad de néctar (calidad de la solución) de v_i es mayor, entonces la solución x_i es reemplazada por v_i; en otro caso x_i permanece.

Selección de fuentes de alimento por abejas espectadoras

Cada abeja espectadora selecciona una de las fuentes de alimento propuestas dependiendo de su calidad. La probabilidad de que una fuente de alimento sea seleccionada se obtiene a partir de la ecuación 3.7:

$$Prob_i = \frac{fit_i}{\sum_{i=1}^{N_p} fit_i}$$

Ecuación 3.7 Probabilidad de fuentes de alimento [38]

donde "fit_i" es el valor de calidad de la fuente de alimento $_i$. La probabilidad de que una fuente de alimento sea seleccionada por una abeja espectadora incrementa con un aumento en el valor de calidad de la fuente de alimento. Después, las abejas espectadoras irán a la posición seleccionada y determinarán una nueva fuente de alimento dentro de la vecindad de la fuente seleccionada. En caso de que la calidad de la nueva solución sea mejor que antes, dicha posición es mantenida; en otro caso la última solución se remplaza.

Determinar abejas exploradoras

Si una fuente de alimento *i* no puede ser mejorada a lo largo de un número predeterminado de *L* intentos. La fuente de alimento se abandona y la abeja correspondiente se convierte en una exploradora. Para verificar si una solución candidata ha alcanzado el límite *L* predeterminado, un contador A_i es asignado a cada fuente de alimento *i*. Dicho contador es incrementado como consecuencia de que una operación de abeja falle en mejorar la calidad de una solución.

Resultados de la segmentación

En el primer caso se considera la imagen "The Camera-man" con su correspondiente histograma. El objetivo es segmentar la imagen en tres diferentes clases. De esta manera el algoritmo ABC ajusta nueve diferentes parámetros, siguiendo el procedimiento de minimización dictado por la función

objetivo. Para este caso ABC fue configurado con una población de 40 abejas, con 20 abejas obreras y 20 espectadoras.

Los parámetros son inicializados aleatoriamente, pero asumiendo algunas restricciones para cada parámetro. Después de 200 iteraciones el algoritmo ABC llegó a un mínimo global. Estos procesos son mostrados en las figuras 3.23, 3.24 y 3.25

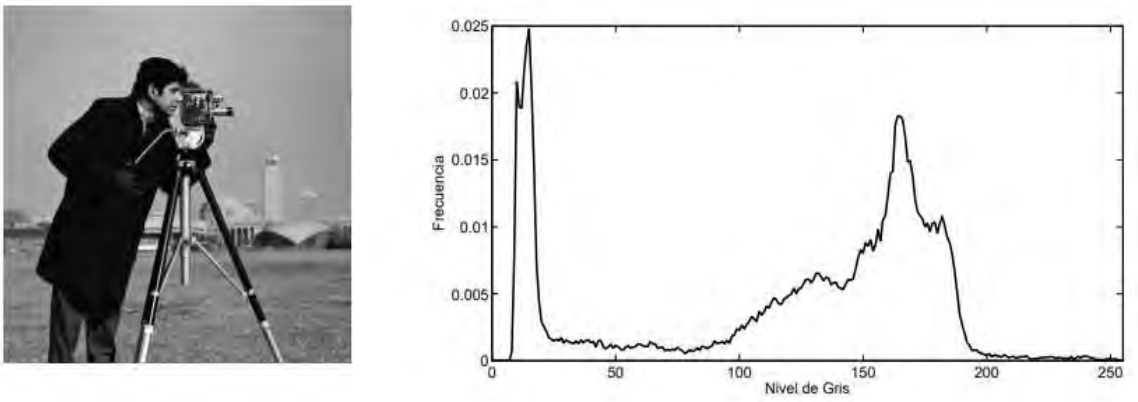


Figura 3. 23 (a) imagen original "The Cameraman", y (b) su histograma correspondiente [38]

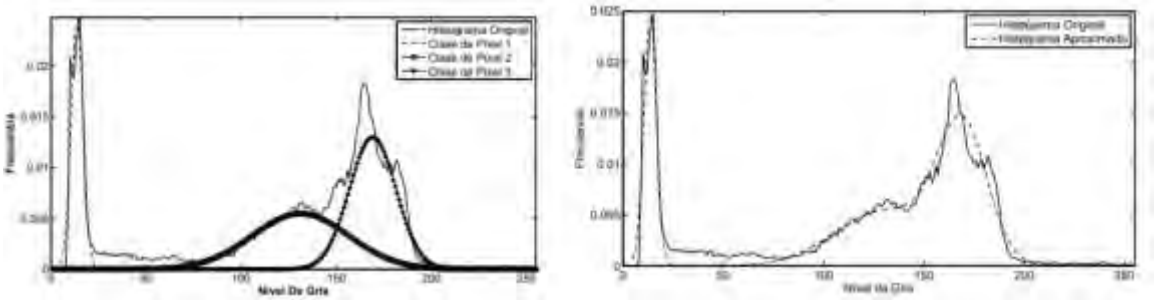


Figura 3. 24 Aplicación del algoritmo ABC para 3 clases y sus resultados: (a) funciones gaussianas de cada clase y (b) aproximación final [38]

Figura 3. 25 Imagen segmentada considerando solo tres clases [38]

También comparan el desempeño de la propuesta ABC contra otros enfoques tales como el Expectación Maximización y el Levenberg-Marquadt, los cuales son comúnmente usados para determinar los parámetros de mezclas gaussianas.

La comparación entre los métodos se concentra en los aspectos: la sensibilidad a las condiciones iníciales y el costo computacional.

La Tabla 3.8 resume los resultados obtenidos en el experimento, donde se muestra los valores promedio obtenidos durante 30 diferentes ejecuciones.

Tabla 3. 8 comparación entre los algoritmos EM, LM y ABC, considerando diferentes valores iníciales.

	Valor	E			Valor	E		
	Inicial (1)	EM	LM	ABC	inicial (2)	EM	LM	ABC
μ_1	10	6.15	5.22	1.21	60	22.78	14.21	1.14
μ_2	81	9.41	8.04	2.54	150	30.14	20.07	1.87
μ_3	185	8.57	7.25	2.14	220	27.24	18.54	1.74
σ_1	8	0.3	0.24	0.14	15	4.51	3.11	0.16
σ_2	10	0.21	0.25	0.17	20	5.12	2.45	0.11
σ_3	6	0.14	0.10	0.05	10	3.9	1.78	0.08
P_1	0.025	1×10^{-3}	0.88×10^{-3}	0.11×10^{-3}	0.060	3×10^{-3}	2.7×10^{-3}	0.10×10^{-3}
P_2	0.025	0.9×10^{-3}	0.54×10^{-3}	0.08×10^{-3}	0.040	4.1×10^{-3}	3.2×10^{-3}	0.07×10^{-3}
P_3	0.025	1.1×10^{-3}	0.41×10^{-3}	0.04×10^{-3}	0.070	3.8×10^{-3}	2.7×10^{-3}	0.03×10^{-3}

Del análisis de la Tabla 3.8 se nota como los algoritmos EM y LM presentan sensibilidad a la elección de sus valores iníciales. Tal hecho se debe a la incapacidad por parte de los algoritmos de evitar quedar atrapados en un mínimo local.

Costo computacional.

En la comparación se usó el histograma, como la mezcla a aproximar, considerando 3 clases. La Tabla 3.9 resume los resultados obtenidos, mostrándose los valores promedio obtenidos considerando 30 diferentes ejecuciones. Resulta evidente que el algoritmo EM utiliza mucho mayor número de iteraciones para asegurar la convergencia, mientras que el método LM es el que invierte mayor tiempo en encontrar la solución. Por otro lado, ABC, resulta el enfoque con la mejor relación de desempeño.

Tabla 3. 9 Comparación entre los algoritmos EM, LM y ABC, considerando el número de iteraciones y tiempo computacional.

Iteraciones	
Tiempo	4(b)
	1865
EM	7.81s
	975
LM	5.04s
	389
ABC	0.67s

Comparación del rendimiento de detector ABC

Con el objetivo de analizar el desempeño del detector circular, el enfoque propuesto fue comparado con el detector basado en algoritmos genéticos (GA) y el algoritmo basado en la optimización de búsqueda de alimento de bacterias (BFOA).

Las imágenes difícilmente contienen círculos perfectos. Por consiguiente, para medir la precisión en la identificación, se coteja el resultado obtenido por la detección automática con el círculo “ideal”, determinado manualmente por una persona sobre la imagen de bordes. De esta manera representa los parámetros del círculo determinado manualmente.

En la comparación los tres métodos GA, BFAO y ABC fueron ejecutados considerando el conjunto experimental de las imágenes. La Tabla 3.8 resume los resultados obtenidos en la comparación, considerando los índices de razón de éxito (RE), el promedio del error de detección (Es), los cuales fueron calculados considerando 35 diferentes ejecuciones sobre la misma imagen. Los mejores resultados fueron resaltados en la tabla 3.10, donde revela que el método ABC es capaz de alcanzar los mejores índices de desempeño al presentar la más alta razón de éxito y el error de detección más pequeño.

Tabla 3. 10 Resultados de los índices de desempeño, razón de éxito (RE) y error de detección (Es) de los algoritmos GA, BFOA y ABC

Imagen	Razón de éxito (*RE*) (%)			Promedio de *Es* ± desviación estándar		
	GA	**BFOA**	**ABC**	**GA**	**BFOA**	**ABC**
(a)	98	**100**	**100**	0.45±(0.022)	0.30±(0.033)	**0.20±(0.021)**
(b)	98	98	**100**	0.61±(0.022)	0.41±(0.034)	**0.19±(0.035)**
(c)	75	90	**100**	0.56±(0.029)	0.46±(0.051)	**0.21±(0.012)**
(d)	90	**98**	**100**	0.62±(0.021)	0.53±(0.018)	**0.28±(0.018)**
(e)	92	96	**100**	0.42±(0.019)	0.37±(0.011)	**0.12±(0.039)**
(f)	95	98	**100**	0.87±(0.056)	0.81±(0.021)	**0.39±(0.027)**

Para validar estadísticamente que el método ABC posee un mejor desempeño, se realizó un análisis no paramétrico sobre el error de detección producido por las pruebas 35 ejecuciones, mostradas en las tablas anteriores. Dicho análisis realiza una prueba basada en la jerarquía, la cual coteja la diferencia en resultados entre dos métodos distintos.

Como hipótesis nula existe diferencia entre los dos métodos bajo análisis. Los valores que arroja son los valores-p, los cuales si son más pequeños que 0.05 ofrecen evidencia suficiente para indicar que la hipótesis nula es incorrecta, de tal manera que los dos métodos bajo prueba deben de ser considerados como diferentes. La Tabla 3.10 reporta los valores-p producidos por el análisis para las pruebas ABC vs. GA y ABC vs. BFOA.

Conclusiones

En este artículo se propuso el uso del algoritmo de optimización “Artificial Bee Colony” (ABC) en el área de procesamiento digital de imágenes. Los problemas abordados fueron dos, segmentación automática y detección de círculos en imágenes digitales.

En la detección de círculos en imágenes se utiliza una combinación de tres puntos borde para la codificación de círculos candidatos. Utilizando las evaluaciones de una función objetivo el algoritmo ABC realiza una búsqueda eficiente hasta encontrar el círculo que mejor se aproxime a aquel contenido en la imagen.

Resultados experimentales validados por el análisis estadístico demostraron que el algoritmo ABC detecta círculos de una manera más robusta y precisa que sus contrapartes.

De la utilización del algoritmo ABC a los problemas planteados se puede concluir que los métodos tradicionales de procesamiento de imagen presentan diferentes dificultades, al momento de ser usados en imágenes que poseen ruido considerable y distorsiones. Bajo tales condiciones, el uso de ABC presenta un mejor rendimiento.

3.16 Tabla de artículos del Estado del Arte

En la tabla 3.11, se muestran de manera resumida cuatro puntos criterios analizados de cada uno de los artículos y antecedentes considerados, los cuales aportan objetivos, técnicas, resultados y utilidades para el desarrollo de la tesis.

Tabla 3.11: Tabla general de artículos

Artículo	Objetivo	Técnicas	Resultados	Utilidad para la Tesis
Alineamiento Genómico usando el algoritmo basado en Best First Search. [Aranda. 2020]	Llevar a cabo el alineamientos e inserción de *indels* similares a los obtenidos por Needleman-Wunsch con un menor coste computacional.	Considera cualquiera de los nodos como el nodo siguiente. La idea es utilizar una heurística que decida cuál de los nodos en la agenda es el más prometedor a ser explorado [Pearl, 1984].	Aplicando este algoritmo se obtiene un puntaje en score y cantidad de *indel* insertados cerca de lo obtenido por Needleman-Wunsch, con menor coste computacional.	Implementación del algoritmo Needlema-Wunch. Variantes en el algoritmo para mejora en *scores*.
Análisis de datos genómicos para el diagnóstico temprano de osteosarcoma [Moncada, 2019]	Investigar, aplicar y evaluar técnicas de aprendizaje automático artificial para el manejo de datos genómicos relacionados con el diagnóstico temprano de cáncer de hueso en humanos.	Banco de datos. Extracción de características. Análisis de componentes principales e independientes. Clasificación. Random Forest. CI, búsqueda a profundidad. XGBoost.	Se aplicaron Random Forest y XGBoost para grandes cantidades de datos, como lo son las muestras genéticas y se realizó la paralelización. Se evaluaron los algoritmos con las siguientes cinco métricas: Precisión, Sensibilidad, Especificidad, Fmeasure y AUC. XGBoost se pueden identificar aquellos patrones o genes más significativos	Banco de datos. Clasificación. Random Forest. ACI, búsqueda a profundidad.
Algoritmo de alineación de secuencias para enfermedades del sistema nervioso central. [Higuera, *et all*. 2017]	El estudio de las enfermedades genéticas que afectan el sistema nervioso central y construir una base de datos que contenga la más relevante información acerca de estas.	Lectura de secuencia carácter por carácter. Alineamiento global. Algoritmo de agrupamiento por tripletes. Lectura y escritura a través de una interfaz.	Muestra las secuencias con mayor similitud en puntaje. Resultado por triples es diferente a los globales, debido a ser más específico.	Lectura de secuencias. Algoritmo global y de agrupamiento. Lectura y escritura a través de interfaz.

Artículo	Objetivo	Técnicas	Resultados	Utilidad para la Tesis
Implementación y análisis de Algoritmos de alineación para datos de next generation sequencing (NGS) [Gago. 2017]	El principal objetivo de este trabajo es escoger tres herramientas basadas en una estructura común, FM-Index. Explicar el funcionamiento de los algoritmos y luego llevar a cabo una implementación de cada uno de ellos para poder compararlos en términos de su efectividad.	Descripción del estado del arte en algoritmos de alineación basados en secuencias. FM-index, donde se describe esta estructura de datos y la implementación realizada para ser usada como base de los alineadores estudiados. Alineadores estudiados son: Bowtie, BWA y por último BWT-SW.	Permite la creación de una herramienta que implementa tres algoritmos de alineación de secuencias basados en el FM-Index.	Realizó la creación de una herramienta que implementa tres algoritmos de alineación de secuencias basados en el FM-Index. Comprensión de diferentes métodos de alineación y el conjunto en un solo lugar de varias de ellas.
Genómica comparada de dos dianas moleculares en modelos animales de hipersensibilidad [Serrano, et all 2017]	Determinar la utilidad de una metodología basada en herramientas bioinformáticas para la selección de los modelos animales para el estudio de fenómenos alérgicos. Describir las similitudes y diferencias a nivel genómico entre el humano y tres modelos animales para las moléculas seleccionadas.	Seleccionar moléculas: hombre (Homo sapiens) ratón (Mus musculus) rata (Rattus norvegicus) conejo (Oryctolagus cuniculus) obteniéndose de la base de datos Gene www.ncbi.nlm.nih.gov/gene/ Comparación a nivel genómico se hizo por medio de Ensembl www.ensembl.org Alineación múltiple de las secuencias herramienta MUSCLE www.ebi.ac.uk/Tools/msa/muscle/ Matrices de identidad Clustal 2.1. Representación gráfica de las alineaciones UCSC Genome Browser http://genome.ucsc.edu	Se encontró una mayor similitud en términos de composición y ubicación de las secuencias codificadoras entre el hombre y el conejo.	Base de datos Gene comparación a nivel genómico Ensemble Alineaciones múltiples MUSCLE Matrices de identidad.

Artículo	Objetivo	Técnicas	Resultados	Utilidad para la Tesis
Documentación y análisis de los principales Frameworks de arquitectura de software en aplicaciones empresariales. [Giménez, 2016]	Obtener como producto una aplicación empresarial que le facilite la gestión con los clientes, pero a su vez le brinde la posibilidad de optimizar sus recursos internos y satisfacer las necesidades del negocio.	Arquitectura de software Diseño de Arquitectura Atributos de calidad Aplicaciones empresariales	Desarrollo de una buena aplicación empresarial El éxito de un desarrollo óptimo de una aplicación empresarial depende de cada área. Planeación, diseño, desarrollo e implementación de la misma.	Arquitectura de Softwares empresariales. Diseño de Arquitectura Evaluación de Arquitecturas Características de Softwares Empresariales Frameworks de arquitectura
Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos. [Sarasty. 2015]	Desarrollar series de análisis sobre los datos de HamMap para detectar las características de estos datos. Realizar una comparativa de comportamiento de diferentes algoritmos de machine learning. Realizar informe de resultados.	Implementación de diferentes algoritmos de Machine Learning en el lenguaje de programación Python. Aprendizaje supervisado: Aprendizaje no supervisado: regresión lógica: Super Vector Machine: Arboles de decisión: Random Forest: K-nn: K-means:	La metodología inicial se pudo seguir sin problemas. Se consiguió trabajar y obtener buenos modelos de "Machine Learning" para trabajar con datos de HapMap, aunque se encontraron dificultades en los tiempos de proceso. K-NN – 91% LR - 98% SVM - 78% TREE - 87% FOREST – 85%	Diferentes técnicas de "Machine Learning" Aprendizaje supervisado: Aprendizaje no supervisado: regresión lógica: Super Vector Machine: Arboles de decisión: Random Forest: K-nn: K-means:
GAIA: Framework Annotation of Genomic Sequence [Charles, *et all*. 1997]	Hacer fácilmente secuencias disponibles para cualquier genoma de una especie para uso de reactivo primario de una investigación de interés para problemas biológicos en particular.	Esquema de bases de datos y anotaciones. Datos funcionales, análisis de genes, datos de mapeo, ubicación de elementos repetidos, descriptores de riesgo, valores actuales manuales y automáticos	Eficaz para identificar secuencias a través de largas distancias. Proporciona datos útiles para predecir la función de un gen.	Mapeo. Ubicación de elementos repetidos. Descriptores de riesgo. Valores actuales manuales y automáticos
Alineación de **secuencias usando CLUSTALX** [BioInteractive. 2014]	Conocimientos básicos en alineamiento genómico implementado mediante los programas ClustalX y Phylogeny	Formato Fasta ClustalX y Phylogeny	Alineaciones genómicas utilizando herramientas básicas de alineamiento	Alineación genómica Conocimiento de herramientas y Frameworks básicos

Artículo	Objetivo	Técnicas	Resultados	Utilidad para la Tesis
Alineamiento gráfico de secuencias a través de programación paralela: un enfoque desde la era postgenómica []	Desarrollar un algoritmo de alineación genómica, basado en los algoritmos DOTTER GEPARD con bajo coste computacional.	Alineación Smith Waterman, Needleman, Dotter y Gepard. Python, Numpy, Matplotlib, Sys, Time, Multiprocessiong, Getopt, Anaconda.	Alineaciones exitosas. Lecturas concretas Velocidad de alineación, aumentada 27.9 veces.	Métodos de alineación. Desarrollo de programa mediante lenguaje de programación Python. Velocidad en alineamientos
múltiple alineamiento de secuencias con los programas en serie Clustal []	Explicar y describir el funcionamiento de los diferentes programas de alineación Clustal	Clustal V, ClustalW, ClustalX, DbClustal, ClustalNet. Clustal WWW web server	Guardado de alineaciones múltiples Guardado con formato NEXUS compatible con otros programas	Desarrollo de programa para alineación de secuencias. Formatos compatibles para distintas plataformas
El diagrama, un método para comparar secuencias.	Ventajas de métodos de alineación mediante diagramas basados en la matriz Needleman	Alineación Gibbs Alineación Kinns, Dale y McKenzie. Diagrama de Fitch		Escritura y Desarrollo de diagramas de alineación mediante matrices
Colonia de Abejas Artificiales (ABC) Algoritmo de optimización para resolver problemas de optimización con restricciones	Resolver problemas de algoritmos de optimización numéricos	Algoritmo basado en el comportamiento de Abejas Artificial Bee Colony	Cada población de abeja es un resultado nuevo en su aplicación “N” numero de abejas programadas equivale a “N” número de nuevos resultados encontrados Solo puede ser utilizado en problemas numéricos	Búsqueda de nuevas alineaciones incorporación en tablas generadas por algoritmo Smith-Waterman (Tabla con datos numéricos)
Colonia de Abejas Artificiales (ABC) Algoritmo de optimización para resolver problemas de optimización con restricciones	Presentar principios básicos de inteligencia colectiva Desarrollar e implementar algoritmo ABC Analizar influencia de cantidades de alimento del algoritmo Proponer un nuevo método de obtención de alimento para el algoritmo ABC	Algoritmo Colonia Artificial de Abejas ABC Comportamiento biológico Inteligencia colectiva	múltiples resultados en búsqueda de alimento Algoritmos propuestos desarrollados implementando nuevas obtenciones de alimento	Desarrollo de algoritmo Colonia Artificial de Abejas Métodos de programación de algoritmo ABC
colonia de abejas artificiales y optimización por enjambre de ´ partículas para la estimación de parámetros de regresión no lineal	Presentar las heurísticas de los algoritmos Colonia de Abejas Artificiales y Optimización por enjambre de partículas para resolver problemas de encontrar parámetros en problemas de regresión no lineal usando mínimos cuadráticos	Algoritmo de Colonia Artificial de Abejas Algoritmo de Optimización por Enjambre de Partículas	la utilización de ABC no proporciona tan buenos resultados como PSO ABC proporciona intervalos más compactos PSO casi duplica el valor promedio obtenido por ABC	Implementación de Algoritmo de Colonia Artificial de Abejas.

Capítulo IV
Metodología de solución

En este capítulo se presenta la implementación de los algoritmos mostrados en el capítulo II y la propuesta de este trabajo. Además de las librerías de Python que se utilizan para el desarrollo de la solución y las secuencias utilizadas.

4.1 – Diseño del sistema

En el capítulo II se habló de los algoritmos clásicos de alineamiento, el algoritmo colonia de abejas, coste uniforme. En esta sección se detallará de la implementación y mejoras en el diseño de cada uno de ellos, los cuales se incorporarán en una sola aplicación.

La figura 4.1 muestra un diagrama de flujo simple indicando el procedimiento que realiza la aplicación.

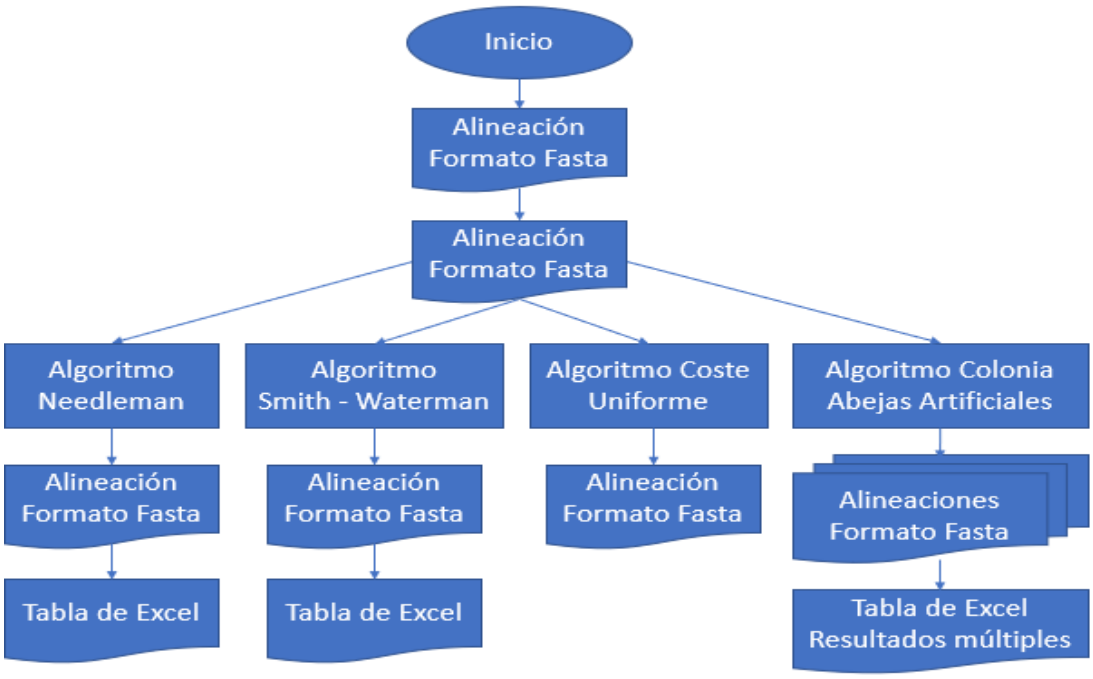


Figura 4. 1 Diagrama de flujo básico para la interfaz

Para esta fase del proyecto, se utilizó el lenguaje de programación Python versión 3.9 [2] para el procesamiento de las alineaciones, debido a que es uno de los lenguajes de programación más populares actualmente en el desarrollo de proyectos de inteligencia artificial. Complementado con su versatilidad, sencillez y buen procesamiento en la lectura y generación de documentos.

Las librerías utilizadas de este lenguaje son:

Openpyxl [39].**:** Es una librería para escribir, leer y procesar archivos del programa Excel con extensión las extensiones .xlsx, .xls y .csv. Se utilizó esta librería para poder procesar los datos en un espacio donde al final del procesamiento, los datos puedan ser visibles para el operador del Framework.

Os: Interfaces misceláneas del sistema operativo [40]. Esta librería, genera las carpetas correspondientes al momento de utilizar un algoritmo. De esta manera, genera una ruta para guardar los archivos generados por los algoritmos, proporcionando orden en los proyectos del operador.

Math: Librería que proporciona funciones matemáticas definidas por el lenguaje C [41]. Fue utilizada, para la generación de ecuaciones y procesos matemáticos en los algoritmos, junto con la implementación de lecturas en las matrices generadas de los mismos.

Tkinter: Generadora de la interfaz de Python [42]. Utilizada para la generación de la interfaz que manda a traer y contendrá en su interior, los algoritmos de alineación.

4.2 - Procesamiento de los datos

Para la lectura de las secuencias, se utiliza una línea de código proporcionada por el lenguaje de programación Python (**open()**), donde dentro de sus paréntesis se escribe la ruta de ubicación del archivo. Esta línea de código le permite leer línea a línea y carácter por carácter, cualquier tipo de formato de texto que se asemeje a la extensión ".txt". El formato FASTA tiene en la primera línea de texto, la información sobre el genoma y se empieza a trabajar con las líneas de texto que continúan después de la primera línea. Con estos datos, Python puede leer carácter por carácter las letras de las secuencias genómicas.

Para el uso de estos algoritmos, se genera la interfaz con la librería Tkinter [42] del lenguaje Python. El Framework incorpora los algoritmos, que pueden ser desplegados mediante una barra de menú. Los botones permiten la selección de los genomas de manera individual, donde al ser seleccionados, la barra de menú permite la selección de los algoritmos para su respectivo alineamiento. La figura 4.2 muestra la interfaz.

Figura 4. 2 interfaz generada con Python

4.3 – Implementación de algoritmos

Con este proyecto, se busca desarrollar un ambiente de desarrollo, que contenga los algoritmos básicos Needleman y Smith Waterman, al igual que algoritmos de inteligencia artificial, y que facilite el proceso de alineamiento genómico para encontrar nuevas y distintas adaptaciones, al igual que se busca encontrar nuevas alineaciones con diferentes similitudes.

Se propone, que el algoritmo de inteligencia artificial *Coste Uniforme*, genere alineaciones genómicas, mediante la generación de árboles de decisión con su respectivo coste generado a partir de la puntuación obtenida mediante la selección de caracteres de las secuencias genómicas.

De igual manera se propone como principal aportación la implementación de la combinación algoritmo de optimización *Colonia de Abejas Artificiales en el método clásico de alineamiento genómico Smith-Waterman*, con el fin de encontrar diferentes tipos de alineaciones en una misma tabla generada por un método clásico.

4.3.1 Método 1: Alineación con algoritmo Needleman

En el proyecto se desarrolla el algoritmo clásico Needleman, como lo muestra la figura 4.3.

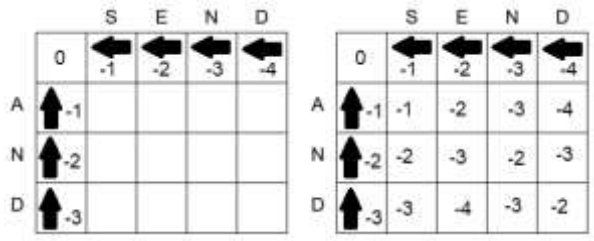


Figura 4. 3 Alineación por SCORE [14]

A continuación, se explica mediante un ejemplo utilizando los genomas del Búho y del Rinoceronte para generar las tablas mostradas a continuación.

Para el proceso de la escritura de la matriz, se leen ambas alineaciones de forma individual, para que, al leer carácter por carácter, se escriben en una tabla de Excel generada por el programa. La primera secuencia, se colocarán carácter por carácter en cada columna en la parte superior y la segunda secuencia, se pondrá en cada fila de la tabla, pegado a la izquierda. Las secuencias genómicas cuentan con una determinada cantidad de líneas de texto, el número de tablas generadas en Excel dependerá del número de líneas de texto que contenga el archivo FASTA. Se puede observar en la figura 4.4 como quedaría generada la primera parte de la matriz.

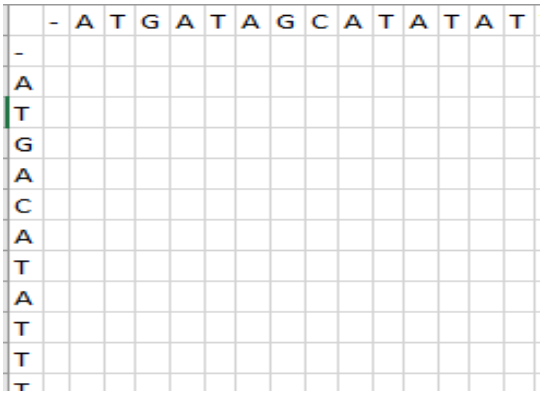


Figura 4. 4 Matriz Búho y Rinoceronte

Continuando con el proceso, la tabla se llena con sus respectivos números, sumando y asignando valores a cada cuadro de la tabla hasta llenarla por completo, donde al terminar, partiendo de la esquina inferior derecha, se buscará la ruta para el alineamiento de las secuencias mediante la selección de los mejores datos existentes en la tabla, donde su objetivo es terminar lo más cercano a la esquina superior izquierda. La figura 4.5 muestra el resultado del llenado y búsqueda de la mejor alineación.

	-	A	T	G	A	T	A	G	C	A	T	A	T	A	T	T
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15
A	-1	1	0	-1	0	-1	0	-1	-2	-1	-2	-1	-2	-1	-2	-3
T	-2	0	2	1	0	1	0	-1	-2	-2	0	-1	0	-1	0	1
G	-3	-1	1	3	2	1	0	1	0	-1	-1	-1	-1	-1	-1	0
A	-4	0	0	2	4	3	4	3	2	3	2	3	2	3	2	1
C	-5	-1	-1	1	3	3	3	3	4	3	2	2	2	2	2	1
A	-6	0	-1	0	4	3	4	3	3	5	4	5	4	5	4	3
T	-7	-1	1	0	3	5	4	3	2	4	6	5	6	5	6	7
A	-8	0	0	0	4	4	6	5	4	5	5	7	6	7	6	6
T	-9	-1	1	0	3	5	5	5	4	4	6	6	8	7	8	9
T	-10	-2	2	1	2	6	5	4	4	3	7	6	9	8	9	10
T	-11	-3	3	2	1	7	6	5	4	3	8	7	10	9	10	11
T	-12	-4	4	3	2	8	7	6	5	4	9	8	11	10	11	12

Figura 4. 5 Matriz Búho y Rinoceronte expresando alineación

4.3.2 Método 2: Alineación con algoritmo Smith-Waterman

El desarrollo del algoritmo de alineación Smith-Waterman, es muy similar al algoritmo Needleman, debido a que la lectura de los genomas, al igual que la escritura de la matriz es la misma. La diferencia se ejecuta al momento de llenar los valores de las tablas, donde su proceso empieza de la misma manera que Needleman. Sin embargo, al comparar carácter con carácter, al llegar al valor de cero, ese valor, no cambia a un valor negativo, se mantiene hasta encontrar valores similares entre los genomas. Al completarse la tabla, se busca la alineación de la misma manera en que lo hace el método Needleman. La figura 4.6 muestra el resultado de la alineación mediante el método Smith-Waterman.

	-	A	T	G	A	T	A	G	C	A	T	A	T	A	T
-	0														
A		1	0	0	1	0	1	0	0	1	0	1	0	1	0
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
G		0	0	1	0	0	0	1	0	0	0	0	0	0	0
A		1	0	0	1	0	1	0	0	1	0	1	0	1	0
C		0	0	0	0	0	0	0	1	0	0	0	0	0	0
A		1	0	0	1	0	1	0	0	1	0	1	0	1	0
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
A		1	0	0	1	0	1	0	0	1	0	1	0	1	0
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
G		0	0	1	0	0	0	1	0	0	0	0	0	0	0
T		0	1	0	0	1	0	0	0	0	1	0	1	0	1
G		0	0	1	0	0	0	1	0	0	0	0	0	0	0
C		0	0	0	0	0	0	0	1	0	0	0	0	0	0

Figura 4. 6 Matriz Búho y Rinoceronte Smith-Waterman

4.3.3 Método 3. Alineación mediante algoritmo de Coste Uniforme

El algoritmo de Coste Uniforme es utilizado para el alineamiento de secuencias genómicas, mediante la generación de posibilidades de alineamiento y la selección de los mejores resultados obtenidos.

El algoritmo empieza leyendo la primera línea de secuencia de ambos genes, al tener el registro, almacenará los tres primeros caracteres de cada línea, con ellos generará un árbol de posibilidades mediante el uso de las normas de puntuación **Match, NoMatch y Gap** mencionadas en el algoritmo de alineación Dot-Plot. La figura 4.7 muestra la generación de posibilidades con sus respectivos puntajes.

Búho **CTC**CGTATTACGGGGTTGTGGGGTTGGTGTTGGGGTCAGTTTGTGGTTGTGGATGGTTGTTGAGTTTGGG

Rinoceronte **CTT**CACCTATTTACGGGGGTTTAGTGTTGATTATGAGTGGTGGATTTGGTTGTGGTATTGTGATGAGTTT

Caracteres		Posibilidades			Puntaje		
CTC **CTT**	Puntaje +1	CTC- -CTT	CTC- C-TC	CTC- CT-C	**-4**	**-2**	**0**
		-CTC CTT-	C-TC CTT-	CT-C CTT-	**-2**	**0**	**0**

Figura 4. 7 Generación de posibilidades

De la misma manera, la puntuación se toma contando tres caracteres; de esta manera, al mejorar la alineación, selecciona dicha posibilidad y la guarda el primer carácter de cada posibilidad en un registro, luego moviendo el lugar de la lectura, tomando los próximos tres caracteres desplazándose un solo carácter, recorre cada carácter generando posibilidades y registrando los mejores alineamientos de manera detallada. La figura 4.8 muestra el completado de la alineación.

```
CTCCGTAACGG-GGTTGTGGGGTTGGTG-TTGGG-GTGAGTGTG-TGGTT-GTGGATGGTTGTTGAGTT
CTTCACCTTTACGGGGGTTTAG-TGTTGATTATGAGT-GGT-GGAT--TTGGTTG-TGGTA-TTGTGA-
```

Figura 4. 8 Alineación Búho y Rinoceronte implementada con Coste uniforme

A continuación, la figura 4.9 muestra el diagrama de flujo del algoritmo, mientras que la figura 4.10 muestra el seudocódigo del mismo:

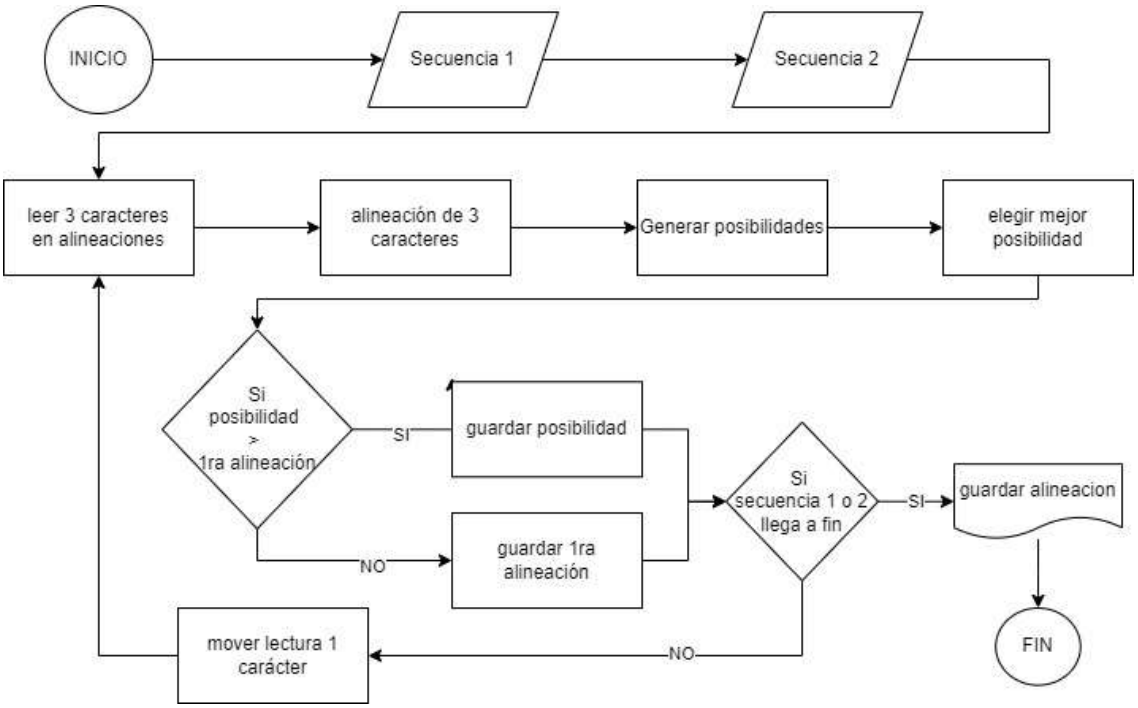


Figura 4. 9 Diagrama de Flujo Algoritmo Coste Uniforme

Pseudocodigo – AI Coste uniforme

Leer secuencia 1
Leer secuencia 2
secuencia más larga = secuencia1 <|> secuencia2
For: Hasta Secuencia más larga
    tomar 3 caracteres de secuencias1
    tomar 3 caracteres de secuencias2
    generar puntuación de posibilidad actual
    GENERAR POSIBILIDADES DE POSICION
    SI existe mejor posibilidad
        Guardar nueva posición
    ELSE:
        Guardar posición actual
    mover lectura un espacio
Leer secuencia final
Contabilizar puntos
Generar porcentaje

Figura 4. 10 Seudocódigo Algoritmo Coste Uniforme

4.3.4 Método Smith-Waterman optimizado con Colonia de Abejas Artificiales

Esta propuesta comienza por buscar posibilidades múltiples en el método de alineación, partiendo del desarrollo de la misma mediante el algoritmo clásico Smith-Waterman, programando una abeja que buscará néctar de manera inversa a como lo hace el algoritmo clásico. La figura 4.11 muestra el resultado original del algoritmo clásico con el color amarillo y el color verde, es la primera abeja haciendo el proceso de alineación de manera inversa.

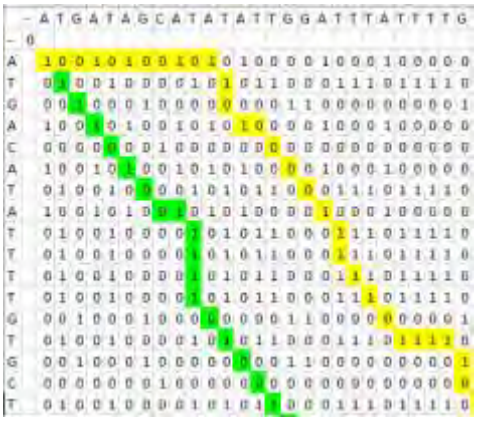


Figura 4. 11 Alineación Búho y Rinoceronte con abeja artificial

En seguida, se programaron otras cuatro abejas para su respectiva búsqueda de nuevos alineamientos, dos que buscaran alimento en la dirección original que marca el algoritmo clásico y las otras dos de manera inversa a como lo hizo la abeja mostrada en la figura 4.11. Considerando lo especificado en el algoritmo de optimización de Colonia de Abejas Artificiales [21], el néctar que buscan las abejas en estas tablas son las posibilidades de mejores valores numéricos, lo que generará nuevas rutas de néctar donde el resultado será la generación de distintas alineaciones genómicas en una sola matriz.

La figura 4.12, muestra el camino que tomaron las abejas programadas, siendo las abejas de color Cyan y Magenta, las que buscan néctar en el camino básico de Smith-Waterman marcado con el color amarillo y las abejas de color Rojo y Azul, buscando néctar de manera inversa como lo hizo la abeja Verde.

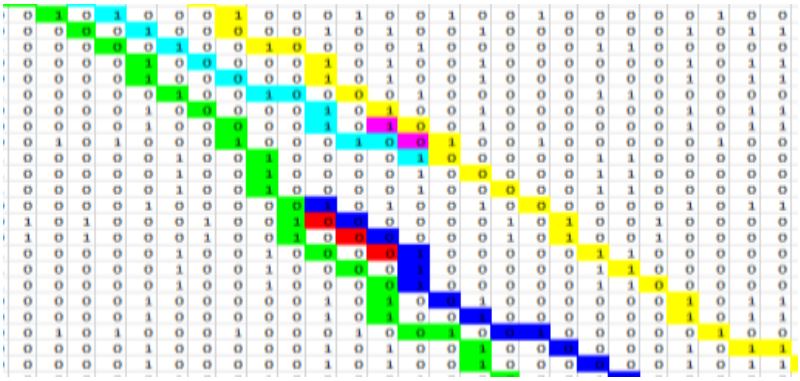


Figura 4. 12 Alineación Búho y Rinoceronte con abejas artificiales

A continuación, la figura 4.13 muestra un diagrama de flujo para este algoritmo, al igual que la figura 4.14 muestra el pseudocódigo.

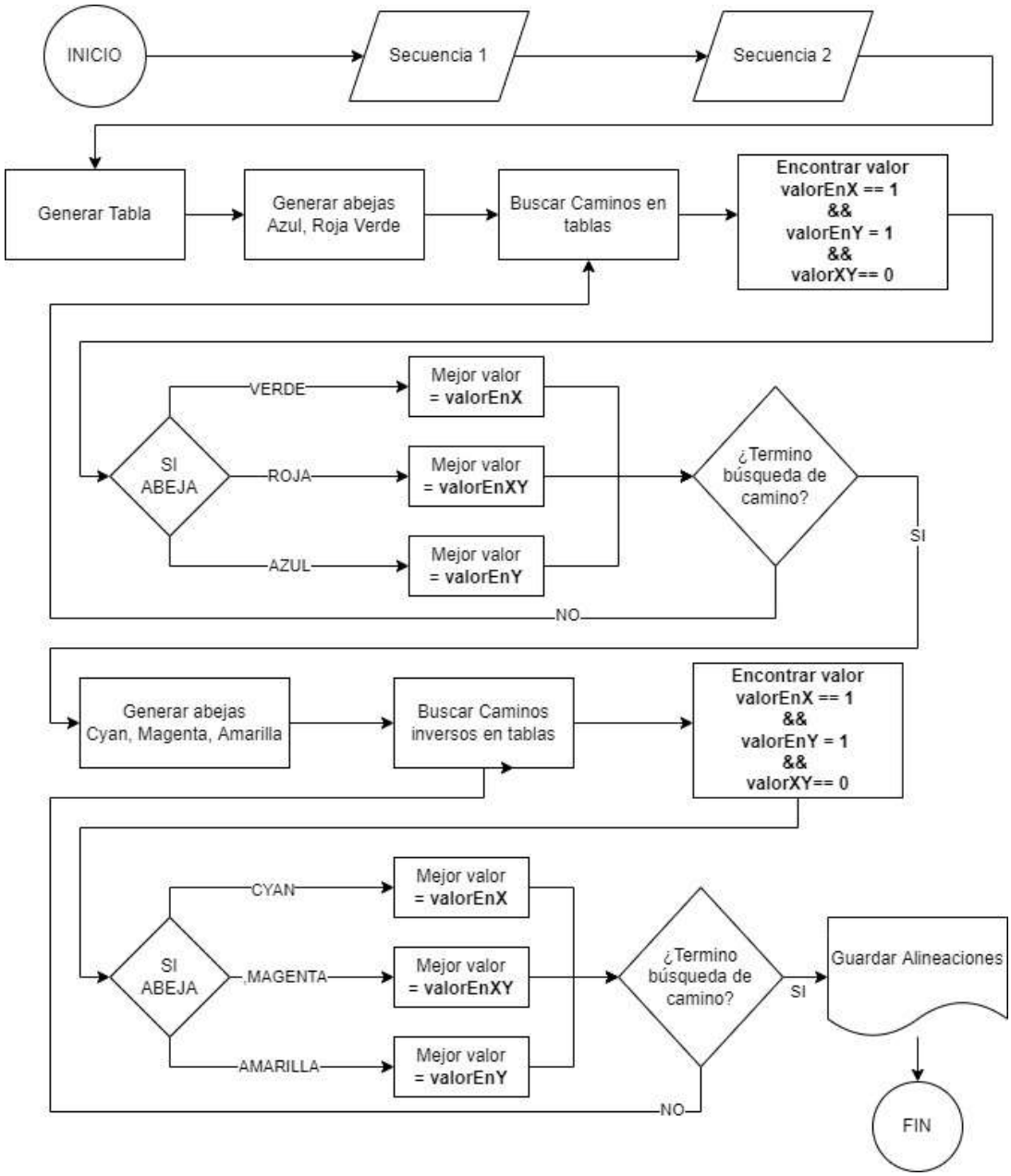


Figura 4. 13 diagrama de flujo Smith Waterman – ABC

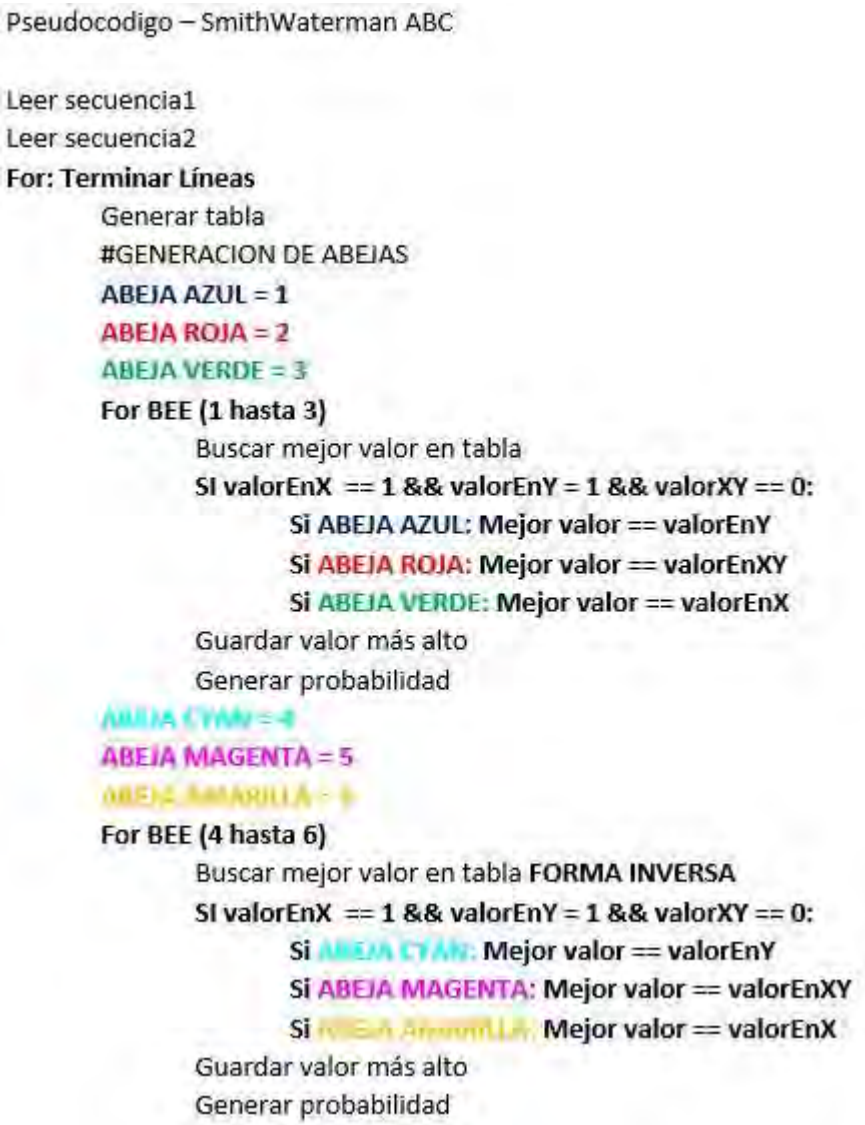


Figura 4. 14 diagrama de flujo Smith Waterman – ABC

Al incorporar el algoritmo de Colonia Artificial de Abejas al lgoritmo clásico Smith-Waterman, se pretende buscar diferentes opciones de alineación genómica dentro una sola matriz. De esta manera se tendrán diferentes resultados ya establecidos por las métricas del algoritmo clásico.

Capítulo V
Pruebas y Resultados

En esta sección se muestran los experimentos realizados para probar el Framework implementando cada uno de los algoritmos investigados.

La experimentación consiste en evaluar el desempeño de los algoritmos clásicos junto con los algoritmos propuestos.

Para cada uno de los experimentos, se lograron hacer 21 alineaciones en cada algoritmo utilizando los genomas de siete animales distintos.

Estos archivos, fueron descargados de el "National Center of Biomedical Institute" NCBI [8], con el formato FASTA, que, en bioinformática, es un formato de texto, utilizado para representar las secuencias de ácidos nucleicos usando códigos de letras.

En el formato FASTA su primera línea de texto comienza con una descripción de la secuencia, dando nombre del genoma. A partir de la segunda línea de texto, se muestra la secuencia genómica con el código de caracteres, siendo una palabra con una longitud de 71 caracteres. La continuación de la secuencia se da en las líneas siguientes, llegando a la última línea de la secuencia, cuya longitud puede ser de 71 caracteres o menor.

La figura 5.1 muestra como se ve el formato FASTA al abrir el archivo.

```
>NC_038219.1:c17149-16628 Bubo bubo isolate D166 mitochondrion, complete genome
ATGACATATTTTGTGCTCTTTCTTGGGGTTGGGTTTGTTTTGGGGGGGTTGGGAGTGGCGTCGAATCCTT
CTCCGTATTACGGGGTTGTGGGGTTGGTGTTGGGGTCAGTTTGTGGTTGTGGATGGTTGTTGAGTTTGGG
GGTTTCGTTTGTGTCGTTGGTGCTGTTTATGGTGTATTTAGGGGGAATGTTGGTGGTGTTTGTGTATTCG
GTGGCTTTAGCGGCGGACCCGTTCCCGGAGGCTTGGGGGGATTGGCGTGTTGTAGGGCGCGGTGCGGGTT
TAGTCGTGGTACTTGTGGCGGGGGTGGTTGGGTGGGGGTTAATTGGCGGTTCTGGGTTTGTGGTGGATGC
GGTAGATAGTGCGGGTACGTTTTTTGTTCGGTTTGATTTTAGTGGGGTTTCTCTTTTCTATTCGTGGGGG
GTAGGAATGTTTTTGGTGGCGGGGTGGGGGTTGCTGTTGACTTTGTTTGTTGTTCTGGAGGTTGTGCGGG
GGTTGTCTCGGGGGGCTATTCGGGCCGTTTAG
```

Figura 5. 1 Ejemplo de formato fasta con secuencia de animal Búho (Bubo Bubo)

Las secuencias de animales utilizadas para la experimentación fueron las siguientes [8]:

- Búho (Bubo Bubo) – 8 líneas – 71 caracteres
- Gato (Felis Catus) – 252 líneas – 71 caracteres
- Kiwi (Apterix) – 228 líneas - 71 caracteres
- Panda (Ailuropoda Melanoleuca) – 119 líneas – 71 caracteres
- Perro (Canis Lupus Familiaris) – 1392 líneas – 71 caracteres
- Rinoceronte (Ceratotherium simum) – 8 líneas – 71 caracteres
- Tiburón (Carcharodon carcharias) – 18 líneas – 71 caracteres

El formato básico cuenta con un encabezado precedido por el carácter “Mayor que” (>) donde se proporciona información sobre el genoma y tras un salto de línea, se presenta la secuencia de ADN o aminoácidos [43].

Al utilizarse estas siete especies, permiten un máximo de 21 alineaciones entre ellas. Por cada secuencia, se esperan un total de nueve alineaciones diferentes por cada par de genomas, dando un total de 189 alineaciones utilizando los cuatro algoritmos con cada una de las secuencias.

a) Objetivos:

En la experimentación tenemos diferentes objetivos dependiendo de cada tipo de algoritmo utilizado.

Objetivo General:

El objetivo general en la experimentación es lograr una alineación global por cada secuencia genómica, dando como salida, las respectivas matrices y formatos FASTA mostrando la respectiva alineación entre secuencias.

Objetivos específicos:

Existen diferentes objetivos específicos al utilizar los diferentes algoritmos. Los cuales se mencionan a continuación:

b) Caso 1: alineación Needleman-Wunsh y Smith-Waterman:

1. Lectura de secuencias genómicas
2. Generar matriz con valores numéricos
3. Obtener alineación como se especifica en el algoritmo
4. Generar formato FASTA mostrando la alineación obtenida

c) Caso 2: alineación de Coste Uniforme:

1. Lectura de secuencias genómicas
2. Generar posibilidades mediante la lectura de tres caracteres
3. Guardar mejor posibilidad con el mejor coste
4. Generar formato FASTA mostrando la alineación obtenida

d) Caso 3: alineación con Smith-Waterman optimizado con Colonia de Abejas Artificiales:

1. Lectura de secuencias genómicas
2. Generar matriz con valores numéricos
3. Obtener alineación como se especifica en el algoritmo Smith-Waterman
4. Enviar abejas a recorrer el primer camino generado
5. Obtener alineación con primeras abejas
6. Obtener alineación como se especifica en el algoritmo Smith-Waterman de manera inversa
7. Enviar abejas a recorrer el camino inverso generado
8. Obtener alineación con abejas de manera inversa
9. Generar formato FASTA mostrando la alineación obtenida

5.1 Experimentación con el Algoritmo Needleman

En el capítulo II, se mencionan los pasos a seguir de cada uno de los algoritmos. En esta sección, se pone en práctica los pasos a seguir del algoritmo clásico Needleman-Wunsh[12] para lograr el objetivo específico de alineación con este algoritmo.

En las figuras 5.2 a la 5.11, se muestran la generación de cinco tablas de alineación genómica hechas con el algoritmo clásico, mostrando el camino de alineación en color verde.

De la figura 5.2 a la figura 5.6, muestran las matrices generadas por los genomas de los animales Kiwi, Búho, Perro, Tiburón, Rinoceronte y Panda. De igual manera, de la figura 5.7 a 5.11, muestran una sección de el alineamiento obtenido por el algoritmo clásico, mostrado en formato FASTA.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		-	A	T	G	A	C	A	T	A	T	T	T	T	G	T	G	C
2	-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16
3	C	-1	-1	-2	-3	-4	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-12
4	G	-2	-2	-2	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-9	-10	-9	-10
5	G	-3	-3	-3	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-8	-9	-8	-9
6	C	-4	-4	-4	-1	-1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-9	-7
7	A	-5	-3	-4	-2	0	-1	1	0	1	0	-1	-2	-3	-4	-5	-6	-7
8	C	-6	-4	-4	-3	-1	1	0	0	0	0	-1	-2	-3	-4	-5	-6	-5
9	C	-7	-5	-5	-4	-2	2	1	0	-1	-1	-1	-2	-3	-4	-5	-6	-4
10	G	-8	-6	-6	-3	-3	1	1	0	-1	-2	-2	-2	-3	-2	-3	-2	-3
11	C	-9	-7	-7	-4	-4	2	1	0	-1	-2	-3	-3	-3	-3	-3	-3	-1
12	C	-10	-8	-8	-5	-5	3	2	1	0	-1	-2	-3	-4	-4	-4	-4	0
13	C	-11	-9	-9	-6	-6	4	3	2	1	0	-1	-2	-3	-4	-5	-5	1
14	C	-12	-10	-10	-7	-7	5	4	3	2	1	0	-1	-2	-3	-4	-5	2
15	A	-13	-9	-10	-8	-6	4	6	5	6	5	4	3	2	1	0	-1	1
16	G	-14	-10	-10	-7	-7	3	5	5	5	5	4	3	2	3	2	3	2
17	T	-15	-11	-9	-8	-8	2	4	6	5	6	7	8	9	8	9	8	7
18	T	-16	-12	-8	-9	-9	1	3	7	6	7	8	9	10	9	10	9	8
19	G	-17	-13	-9	-7	-8	0	2	6	6	6	7	8	9	11	10	11	10
20	T	-18	-14	-8	-8	-8	-1	1	7	6	7	8	9	10	10	12	11	10
21	T	-19	-15	-7	-8	-9	-2	0	8	7	8	9	10	11	10	13	12	11
22	C	-20	-16	-8	-8	-9	-1	-1	7	7	7	8	9	10	10	12	12	13
23	C	-21	-17	-9	-9	-9	0	-1	6	6	6	7	8	9	9	11	11	14
24	A	-22	-16	-10	-10	-8	-1	1	5	7	6	6	7	8	8	10	10	13
25	C	-23	-17	-11	-11	-9	0	0	4	6	6	5	6	7	7	9	9	14
26	C	-24	-18	-12	-12	-10	1	0	3	5	5	5	5	6	6	8	8	15
27	G	-25	-19	-13	-11	-11	0	0	2	4	4	4	4	5	7	7	9	14

Figura 5. 2 Alineación Kiwi – Búho

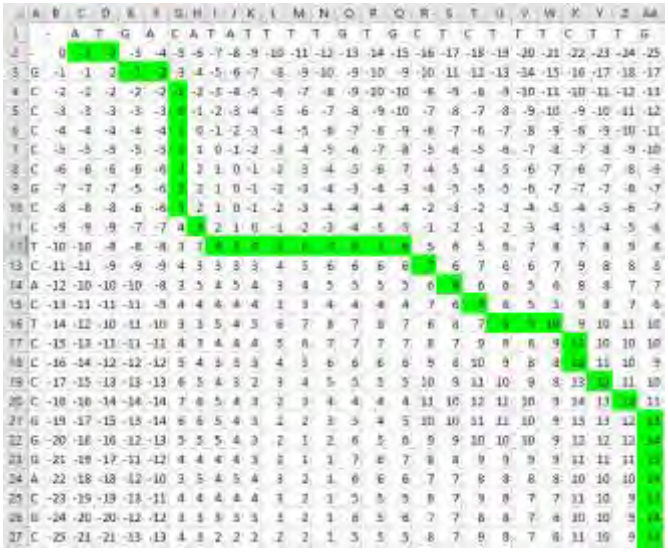


Figura 5. 3 Alineación Búho – Perro

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		-	A	T	G	G	C	C	C	T	C	A	A	T	A	T	T	C	G
2	-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15	-16	-17
3	G	-1	-1	-2	-1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-11
4	C	-2	-2	-2	-2	-1	1	2	3	2	3	2	1	0	-1	-2	-3	-2	-3
5	C	-3	-3	-3	-3	-2	2	3	4	3	4	3	2	1	0	-1	-2	-1	-2
6	C	-4	-4	-4	-4	-3	3	4	5	4	5	4	3	2	1	0	-1	0	-1
7	C	-5	-5	-5	-5	-4	4	5	6	5	6	5	4	3	2	1	0	1	0
8	C	-6	-6	-6	-6	-5	5	6	7	6	7	6	5	4	3	2	1	2	1
9	G	-7	-7	-7	-5	-4	4	5	6	6	6	6	5	4	3	2	1	1	3
10	C	-8	-8	-8	-6	-5	5	6	7	6	7	6	5	4	3	2	1	2	2
11	C	-9	-9	-9	-7	-6	6	7	8	7	8	7	6	5	4	3	2	3	2
12	T	-10	-10	-8	-8	-7	5	6	7	9	8	7	6	7	6	7	8	7	6
13	C	-11	-11	-9	-9	-8	6	7	8	8	10	9	8	7	6	6	7	9	8
14	A	-12	-10	-10	-10	-9	5	6	7	7	9	11	12	11	12	11	10	9	8
15	C	-13	-11	-11	-11	-10	6	7	8	7	10	10	11	11	11	11	10	11	10
16	T	-14	-12	-10	-11	-11	5	6	7	9	9	9	10	12	11	12	13	12	11
17	C	-15	-13	-11	-11	-12	6	7	8	8	10	9	9	11	11	11	12	14	13
18	C	-16	-14	-12	-12	-12	7	8	9	8	11	10	9	10	10	10	11	15	14
19	C	-17	-15	-13	-13	-13	8	9	10	9	12	11	10	9	9	9	10	16	15
20	C	-18	-16	-14	-14	-14	9	10	11	10	13	12	11	10	9	8	9	17	16
21	G	-19	-17	-15	-13	-12	8	9	10	10	12	12	11	10	9	8	8	16	18
22	G	-20	-18	-16	-12	-11	7	8	9	9	11	11	11	10	9	8	7	15	19
23	G	-21	-19	-17	-11	-10	6	7	8	8	10	10	10	10	9	8	7	14	20
24	A	-22	-18	-18	-12	-11	5	6	7	7	9	11	12	11	12	11	10	13	19
25	C	-23	-19	-19	-13	-12	6	7	8	7	10	10	11	11	11	11	10	14	18
26	G	-24	-20	-20	-12	-11	5	6	7	7	9	9	10	10	10	10	10	13	19
27	C	-25	-21	-21	-13	-12	6	7	8	7	10	9	9	9	9	9	9	14	18

Figura 5. 4 Alineación Perro – Tiburón

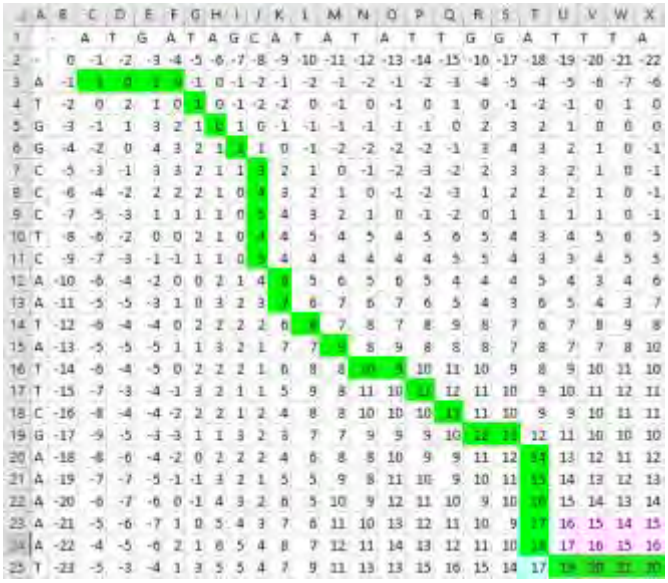


Figura 5. 5 Alineación Tiburón-Rinoceronte

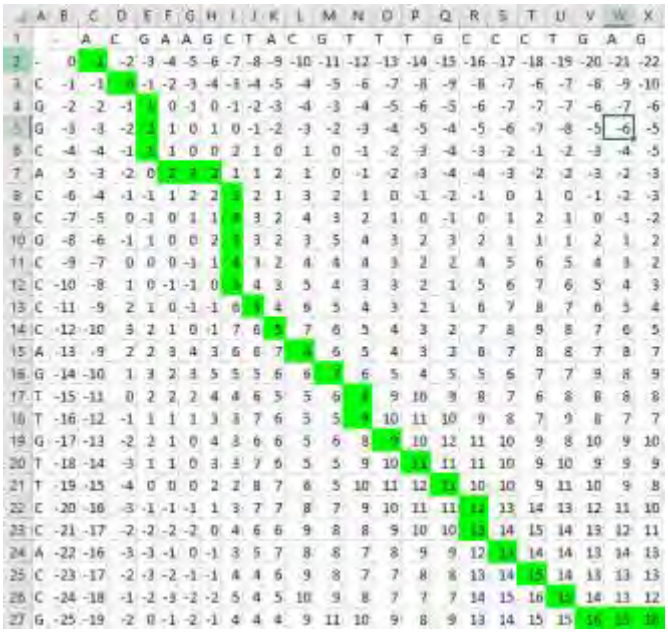


Figura 5. 6 Alineación Panda - Kiwi

Figura 5. 7 Alineación Kiwi – Búho

Figura 5. 8 Alineación Búho – Perro

```
ATTGC-----CCTCAATATTC---G---------AAAAATT-CATC-CCCTACTAAAAATTATAAACCAAAC------TC-TAATTGATTCTCCA-G
----CCCCCGCCTCA--CT-CCCCGGGACGCAGGA---AAACCGGGCGCCTGCGA----------GC-A-GCGGCTCCTCCT----G-GGC-CCAAG

-CTCCATCCAACATCTCCATCTGAA-GAAAA--TTC-G--GCT-CC--CTCTTAAGACTGTGTTTAATAAT---C-CAAAT------TGT
GAGC--CCC----GGC-CGT-TG-GTCGGGGCCT-CCGGGTCTTTTAAGTCT--T-ACT------CCCA-TCTTCCCAAATTCCCTTCTC

CACAA---GACTCTTCCTA-GCCATACATTACACCG---CAG--------ATAATACTATAG--CCTTCTTCTCA-G--TTACCCACAT-C---
-----GTGGAGGTTGC-GAAGGG------AAGCCCGAGGTAGGGCGCGCGA--A-GCT-GAGGGC-T---T--GATGGGGG-C--AAGTTCGCA

TT-------CCGTGACGTC---AATTAC--G-GCT-GACCTATTC---G---TTACATCCATTCCCACCGAGCCTCCTTATTCTTTG-----------TCT
--GAAGGGTC-G---GGTCCCCAGTCTCCCGCGCTTG-GG----CCCCGGGGGG--------A--G-AA-GGC---C-------TCGGAGAGGAGGGAGGGG

GC-A-TCTAATTC--CACATTGG--CCG----AG--------GA-CTTTA---TTAA----G----GCTCCTACCTCCAAAAAAAGACCTG-AAATATTTG
GCGAGTCTTTT-CGGCAAGT--TCTCCGGGGCAGTGGGGTGAGAACT--AGAGT--TCCCCGGGGGGC---------C-T---C----G-GGA------AC
```

Figura 5. 9 Alineación Perro – Tiburón

```
ATGAATGC----A-TATATTGGA----TTTATTTTGGATAAC----A-TA----TTTGTA-------AT-TAGGTTTGTTGGGTTTTCTTC-AAAAC
A---ATGCCCTCAATAT-TCG-AAAAAT--------TCA-ACCCCTACTAAAAAT-TATAAACCAAACTCTAAA----TT--AT---CTTCCA---G

C----TTCA-----C-CTTTTTACGGGGGTTTA--GT--G--TTGA-TTATGGATGGTGGA-TTTGGTTGT---GGTA-TTGTGA--T-GAGTTT
CTCCAT-CAAACATCTCCCT--CTG---AT-GAAACTTCGGCT-CACT-CTTTA-----GACT--G-T-GTTTAG-TAAT--CCAAATTG--T--

---TGGGGGT--TC-TTTTTTGGGGTTGA-TGGTTTTTTTA-ATTTA-TTTTGGTGGGA--T----GTTGGTGGT----TTTTGGTTA-------T-
CACAG-GACTCTTCCT----AG-----CAAT---------ACAT-TACAACC---CAGATATTACTATAG--CCTTCTCCTCAG-T-AACCCACATC

-AC-G--ACGGCTA-T-A-G-CTACCGAGCCAG-AT-CCTTAGGTATGGGTA---TCTTGGGC-A--G---C-TGTTTTAGGG-----TCGTTTG
TGCCGTGACG-TCAATTACGGC---CG---GACTATTC--C---------TAACATCCC-AGCCAACGGAGCCT-CTTTA--ATCTTTT--T-CT

---TTTTAGGGGTATTA-ATAAAGGTT--GTGTTGGTTTTAT-ATG--T-TTA--T--A-----AGAAT-GGTGAGGTGGAGGGTGTTTT
GCAT-CTA-AC--TCCACATTT----GCCG-AGGACT-TTATTACGGCTCCTACCTCTACAAAGAGACTTG---------AAAAATTT-G
```

Figura 5. 10 Alineación Tiburón-Rinoceronte

Figura 5. 11 Alineación Panda – Kiwi

Los resultados obtenidos, son la alineación global entre los distintos genomas. En este punto, se tiene una referencia de cómo se deben de ver las alineaciones futuras generadas con los demás algoritmos.

5.2 Experimentación Algoritmo Smith-Waterman

La implementación es similar a la realizada con el algoritmo Needleman, con la diferencia de que este algoritmo, no utiliza números negativos. Se espera que, con este algoritmo, los resultados no deben de ser muy distintos a los obtenidos con el algoritmo Needleman.

Para las siguientes alineaciones, se utilizaron los mismos genomas de los animales Buho, Tiburon, Rinoceronte, Perro, Kiwi, Panda y se agregó el genoma de Gato.

Las figuras muestran las matrices generadas de manera similar que en algoritmo Needleman. Se inicia con la figura 5.12 mostrando la primera matriz y terminando las matrices en la figura 5.16. A continuación, de la figura 5.17 a la figura 5.21 muestran el resultado de las alineaciones en formato FASTA.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	A	T	G	A	C	A	T	A	T	T	T	T	G	T	G	C	T	C	T	T	T	C	T	T
2	-	0																								
3	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
5	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
6	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
9	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
10	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
11	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
13	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
14	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
15	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
16	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
17	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
18	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
19	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
20	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
21	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1

Figura 5. 12 Alineación Panda – Búho

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
2	-	0																							
3	G		0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
4	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	G		0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
10	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	T		0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	0	0	0	1	1	1	0	1
13	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	A		1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	1	0
15	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	T		0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	0	0	0	1	1	1	0	1
17	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	C		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	G		0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Figura 5. 13 Alineación Perro – Rinoceronte

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	A	T	G	G	C	C	C	T	C	A	A	T	A	T	T	C	G	A	A	A	A	A	T	C
2	-	0																								
3	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	C		0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
5	C		0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
6	C		0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
7	C		0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
8	T		0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
9	T		0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
10	A		1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0
11	T		0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
12	A		1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0
13	T		0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
14	A		1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	1	1	1	0	0
15	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
16	T		0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
17	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
18	T		0	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0
19	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
20	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
21	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
22	G		0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Figura 5. 14 Alineación Tiburón – Gato

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	C	G	G	C	A	C	C	G	C	C	C	C	A	G	T	T	G	T	T	C	C	A	C	C
2	-	0																								
3	G		0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
4	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
5	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
6	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
7	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
8	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
9	G		0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
10	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
11	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
12	T		0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
13	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
14	A		0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
15	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
16	T		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0
17	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
18	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
19	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
20	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1
21	G		0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
22	G		0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
23	G		0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
24	A		0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
25	C		1	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1

Figura 5. 15 Alineación Kiwi – Perro

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	G	G	T	T	T	C	G	T	T	T	G	T	G	T	C	G	T	T	G	G	T	G	C	T
2	-	0																								
3	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
4	A		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
6	A		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	G		1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	1	0	1	0	0
8	G		1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	1	0	1	0	0
9	A		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
11	T		0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0	1
12	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
13	T		0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0	1
14	T		0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0	1
15	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
16	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
17	T		0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0	1
18	A		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	G		1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1	1	0	1	0	0
20	C		0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
21	A		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	A		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 5. 16 Alineación Búho – Tiburón

```
ATGGCCCCCTCCCAATTATTCGAAAAATCCCCCCATCCCCTACTTAAAAAAATTATAAAACCAAAAACTTTAAATTGATCTTCCAGG
GCCC--TAA--TAG-GTGGGGGGACGGG----TGCTCAGATC-TCCC--GGGGTG-CCCTGT--CTTGC-T-AGA-TCCCCCCCC-

CTCCATTCAAACATTCTCCCCATTTGATGAAACTTCGGCTCACTCTTAGGACTGGTGTTTTAGTTTAAATTCCAAATTGTT
CTGG-GTGATCC-CTC-C-AGAT-GCCTGAGCCGGGGGGGAAGGAAAAGCCA-GTA-CCGGG--GG-C-AGGGTTGGGG-

CACAGGACTCTTCCTAGCAATACATTACAACCCCCAGATATTACTAATAAGCCTTCTCCTTCCGTAAACCCACATCCCCCCCCCCC
TGGAGTGGGGGGGGGGCCAGGAAAAAG-GC-GG-GGCATCTAAA-AA-TGGGGGGATT-CTG-G-GTGGGTTTCC----------

TTCCGTTTACGTCAAATACGGGCTGACTTATTCGTAAACATCCATGCCAAAACGGAGGCCTCTTTATTCTTTTGTTCTTTTTT
G-GT-CC-TGAGGGT-TC-TGAGAGTTGGGGGCTG-CTTTCTGGGTT-T-TTTCT-ATTTTCTTGGTTT-CTC-TT-C----

GCCATCTACTTTTCCCACATTGCCCGGAGGACTTTATTACGGCTTCCTTACCTCTACAAAAAGAGACTTTAAATATTGGGGG
-TTCTGAA--AG-GTGAGGGTGAA-CAAGGGGCCAGGGGGGG-ACA-AGTGGGGCCC--CACCCCCCAG-TCTCGGGG---
```

Figura 5. 17 Alineación Tiburón – Gato

ATGATAGCATAATATTGGATTTATTTTGAGTATTCATATTTGTAATTAGTTTTGTGGGGTTTTCTTCAAAAAACCCCCCC
GCCCCCGCC-TCCAACTCCCCCCGGGACGCA-GAAACCCGGGCGCCTGCGAGCAGCCGGGCTCCCT--TGGGC------

CTTCACCTATTTACGGGGGGGTTTTTTTTTTTAGTGTTGATTATGAGTGGGTTTTGGATTTGGTTGTGGTATTGTGATGAGTTTTT
GAGCCCCCAAAGGGGC-G-T----G--C-CTCCGGGTCTTTAAAGTC-G---CTC-ATTCTTCCCAAATTCCCCCCTTC-CCC-

TGGGGGGGGGGTTCTTTTTTTTTTGGGGGTTGATGGTTTTTTTAAAAATTTTTTATTTTTGGGGGGGTTGGGATGTTGGTGGTTTTTGGTTATT
G----A-GTTGCGAA---AA-CCCC-AGGTAGGGCGCGCGAT---GA--GC-CG--GGG-GA---G-TCGCAAAAAAAAAAAAAAAAAAAAA-

ACGGACGGCCATAGCTACTGGAGCAGTATTCCCGAGGTATGGGTATCTTAAAAGTGCAGCTTTTTTTAGGTGCGTTTGGGGGGGGGG
G-AAAGGGT-CGGGGTCC-CCAGTCTT-CCGC-CGCTTGGTTTTTT-C---GGGGAGAAGG-G-CTTCGGAGAGGAA---------

TTTTAAGGGTGTATTTAATAGAGGTCGTGTTGGTTTTTATATTTTGTTTATAAGAATGGTGAGGTGGAGGGTTGTGTTTTTTTTTTTTTTTT
GCG-G-CTTCCT-GGCAAAAGTCCTCCCCCCGGG-GCAAA---GGGGTTGGAAGAACTAGAAAAAAAG-CCCCCCG-G------------

Figura 5. 18 Alineación Perro – Rinoceronte

ATGACATATTTTTGGGGTGCCCTTTCTTGGGGTTGGGGTTTTTTGTTTTGGGGGGGTTGGGAGTGGGCGGCGAATCCTTTTT
ATGGGCC-CTTT---TATTT-CGAAAAAATCCATC-CCT---AAAAATTTTATAAACCAAACT-TAATT-GGATCT--CA-

CTCCGTATTACGGGGGTTGTGGGGGGGGTTGGGTGTTGGGGGGCAGTTTTTGTGGTTGTGGAATGGTTGGTTGAGTTTGGGGGGGG
CTCCATCCCCAA--CATCT-CAT---ATG-AACTTTTCG-CT-CAC--TTAAGGGACCTG-GTTTTA-TAATCCCCCA--T--T-

GGTTTCGGTTTGTGTTGTTGGTGGCTGTTTATGGTTTGTAATTAGGGGGGAATGTTGGTTGGTGTTTGTGTATTTGGGGGG
CACAG-ACTCTTTCC-TAGCA-TACAAATTACA--GCAAG-AT-TTACTAT-AGCCC-TCTCCTTCAGTAACCC------

GGGGCTTTTTTAACGGCGGACCCCCCGTTCCCGGAGGCTTGGGGGGATTTGGGCGTGTTGTTAGGGCGCGGTTGGCGGGTTTT
T-GCC---AC-G-TTCAAT-A--GCCTGACTTATTCGTAACATCCAA-G-CAAACGGGG-GCCTCTTTAT-C-TTGT--TT-

TAGTCCGTTGGTACTTGTGGCCGGGGGGGGGTGGTTGGGTGGGGGGGTTAATTTGGCGGGTTCTGGGTTTGTGGGTGGGATGCC
GCC-TC-ACTTTCCACATT-GCCC----GGACTTTTTTA-TACG-CTCCT-CCTCTA-AAAAGAGACTTGG-AAT-TTGGGG-

Figura 5. 19 Alineación Búho – Tiburón

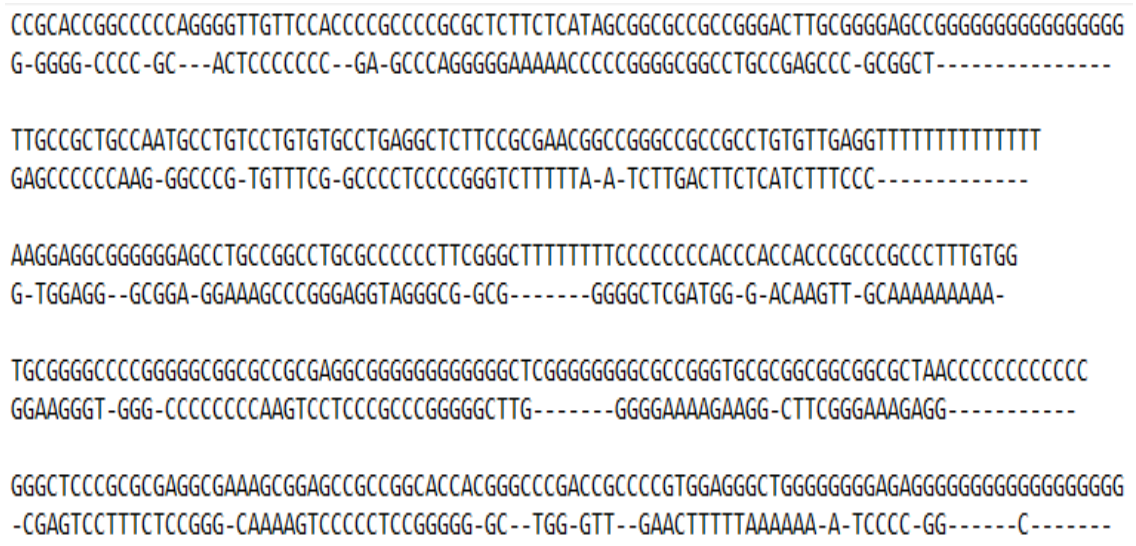


Figura 5. 20 Alineación Kiwi – Perro

Figura 5. 21 Alineación Panda – Búho

Se esperaba que los resultados obtenidos con los algoritmos Needleman y Smith-Waterman, fueran idénticos. Sin embargo, analizando los datos mostrados en las ilustraciones 5.2 a 5.21 y la forma en como Smith-Waterman genera la matriz, existe una ligera variante en los resultados.

Mencionado en el capítulo II, estos resultados no significan que sean erróneos, son otras posibilidades de alineación generada por otro algoritmo.

5.3 Experimentación Algoritmo Coste Uniforme

Este método se basa mediante la lectura de las secuencias genómicas y genera las alineaciones directamente.

El principal desarrollo de este algoritmo se basa en la lectura de los caracteres de las secuencias de tres en tres, de la misma manera como lo hace el "algoritmo de alineación de secuencias para enfermedades del sistema nervioso central "[25]. Al obtener los caracteres, se generan los árboles de posibilidades, dando como resultado la alineación por medio del mejor coste.

Las figuras 5.22 a 5.26, muestran los resultados obtenidos de las alineaciones de los genomas de los animales Rinoceronte, Gato, Panda, Tiburón, Búho y Kiwi.

```
ATG-A-TATCATATA-TGGATTTATTTTGA-GTATCATATTT-GTAATTAGTTGTGTGGTTTT-CTTC
GCCCCTTA-TATA-GGTGGGGGGACG-GGTCCT-GCTCAGATCCTCCTTGGGT-GGCCTGACTGC-TC

CTTCAGCT-ATTT-CGGGGGTTTAGTGATGATTATG-AGTGGTGGATTTGGTTGT-GGTATTGTGATGAG
CTGGT-GTGATCTTCTCAAGATGCCTG-AGCCGGGGAAGAAGC-CAGGTAGCCGCAGGTCTAGGGTTGGG

TGGG-GGTTCTTTTGTGG-GG-TTGATGGTTTTTTTAATTTATGTGGGTGGCATG-TTG-GTGGTTT--TT
TGAGTGGGGGGCCA--GGAGGGCTGG-GGCATCTACAATTGGG--GGATTC-CTGAGTGGGTTCCTGAGTT

ACGACGG-CTATCGCT-ACTGAGCAGTAT-CTGA-G-GTATGGGTATCTAGTGC-AGCTGTTTTAGGTGC
GTTCCTGAGT-T-CCTGAGAGTGGGGGCTTCTTCTGGGTCTTTTTCT-GATT-CTTGGT-TTCTCCTTCC

TTTTAGGTG-ATTAATAGAGGTCGTGGTGGTTTTAT-ATGTTTA-TAAG-AATGGTGA-GGTGGAGGTTG
CTTCTGATGGAGTGAGGGTGAACCAG--GGCCAGGGGGGTACAGAGTGGGCCATCACCCCAGTCTCGGGAC
```

Figura 5. 22 Alineación Rinoceronte – Gato

```
ACGAAGCCGT-TTGCCCTGAGAGAG-GCGGG-GCTGCTC-TTGTGGCCCTGGTCTCGAGAG-CATTCGG
ATGGCCCAATATT-CGAAAA-ATCCATCCCCTACTAAAAATTATAAACCAA-ACTCT-AATTGA-TCTT

CTCGCGA-GAATAGA-CGCCCTACTTG-TGCGCCCG-GGA-AACCCTGTT-GCTCCCT-TTCCGCTG-GCT
CTC-C-ATCA-AACATCTCCA-TC-TGATGAAACTTCGGCTCACTC--TTAGGACTGTGTTTA-GTAATCC

CGGAGGCCGC-ACCGTAAGGCGG-GCGCTCCGGTACGGACTGG-GCGCGCGGCGTCAGCGTCCG--ACTCC
CACAGGACTCTTC-CT-A-GCAATACATTACACCGCAG-ATATTACTATAGCCTTCTC-CTCAGTAAC-CC

GGGAG-CG-C-GTGTCTGGGGG-CTCCGGGGCGACTGC-GTGGG-GCTGCCCCGCCCCCTGGATCTG-C
CCGTGACGTCAATTAC-GGCTGACTTATTCGTAACATCCATGCCAACGGAGCCTCTTTATTCT-TTGTC

GT-GCGAGCGG-CAGGCCCGGACATG-CCTTGTCGAACGG-TCCGGGAACCAAGAATGGGCCAGGTGGGC
GCATCTA-CTTCCACAT-TGCCCGAGGACT-TTAT-ACGGCTCCTACCTCTACAA-AGAGACTTGAAATA
```

Figura 5. 23 Alineación Panda – Tiburón

```
C-GGCACCGCCCCAGTT-GTTCCA-CCGCCCGC-GCTCTT-CTCATA-GCGGCGCCGCCGGGACTTGCGG-
ATGGC-CCTCAATA-TTCGAAAAATCCATCCCCTACTAAAAATTATAAACCAAAC-TCTAATTGAT-CTTC

TTGCC-GCTGCC-CTGCCTTCTG-TGT-GCTGAGGCTCTTCGCGAACGG-CGGTCGCCGCCT-GTGTT
CT-CCATCAAACTCT-CC-TCTGATGAAACTTCGGCTC-AC-CTTA-GGACTG-TGTTTAGTAATCCA

AAGAGG-CGGGGCGCT-GCCGGCC-T-GCGCCCCCCTTCG-GCTCCCCCC-CCACACACACCCGCCGCCCT
CACAGGACTCTT-CCTAGCAATACATTACACCGCAGATATTACTATAGCCTTCTC-CTCAGTAACCCACAT

TGCGGGGCCCGGGGCG-GCGCCGGAGGCGGGGGGGGGGGGCTCGCGCCG-GGTGC-GCGCGGCTGCGCTA
TGCCGTGA-CGTCAATTACGGC-GACTTATTCGTAACATCCATGC-CAACGGAGCCTCTTTAT-TCTTTG

GGC-TC-CC-GCGCGAGG-CGAAGCGAG-CCGCCGGCACCACGGCGACGCCGTGGAGGGATGGGGGGAGA
-GCATCTACTTC-C-ACATTGC-CC-AGGACTTTATTACGGCTCCTACCTC-TACAAAG-AGACTTGAAA
```

Figura 5. 24 Alineación Kiwi – Tiburón

```
GCCCCCGACT-CACTCCCCGG-GACGGAGGACACCCGGGC-G--CCTGCGAGGAGCGG-CTGCT-CCTGGC
GCCCCTT-ATATAGTGTGGGGGGACG--GGT--CCTGCTCAGATCCTCCTTG-GGTGGCCT-GTGACT-GC

GAGCCCAGGCCGTTGTTCGTGC--CCTCCG--GGTCTTTAAAGTCTGGCTCTCATCTTCCCAAA--TTCCC
CTGGTGTGATCTCTC-TCAAGATGCCTGAGCCGGGGAAGA-AGCCAG-GTAGCCGCAGGTCTAGGGTTGGG

GGTGGAGGTTGCGAAGGAAAGCCCGAGGTAGG-GCGCGCGATG-CTGAGGG-CTCGATGG-GGACAAGTTC
TGAGTGGGGGGCCA-GGAGGGCTGG-GGCATCTACAATTGGGGGATTCCTGAGTGGGTTCCTGAGTTCCT-

GGAAGGGACGTGTCCCCAGTCTCCC-GCGCT-TGGTCC-CCGGGGG-AGAAGGTCTTCGGAGAGGAGGGAG
GTTCCTG-AG-TTCCTGAGAGTGGGGGCTGTCTTCTGGGTCTTTTTCTGATTC--TT-GGTTTCTCCTTCC

GCGAGTCTTCACG-GCAAGTGCTCCGG--G-G-CAGTGGGGTGAGA-ACTAGAGTCCCCCGGGG-G-TC
-CTTCTGATG-AGAGTGAGG-GTGAACCAGGGCCAGGGG-GT-ACAGAGTGG-G-CCATCACCCCACTC
```

Figura 5. 25 Alineación Perro – Gato

```
ATGAC-AT-ATT-TTGTGCTCTTTC-TTGGGGTTGGGTTTGTTTTGGGGGGG-TTGGGAGTG-GCGTCGA
ATGGCCCTCAATATT-CGAAAAAATCCATCCCCTACTAAAAAATTATAAACCAAACTCTAATTGATCTTCCA

CTCCGT-AATACGGCGTT-GTGGGGTTGGTG-TTGGGGTCAGT-TTGTGG-TTGTGGATGGTTGTTGAGTT
CTCCATCA-AACAT-CTCCATCTGA-TGAAACTTCGGCTCACTCTTA-GGACTGTGTTTAGTAATCCAAAT

GGTTGCGTTTGTGTCGTTGGTACTGT-TTATGGTGTATTTAGGGGGAATG--TTCGTGGTGTTTGTG-TA-
CACA-GGACTCT-TCCTAGCA-ATACATTACACCGCAGATATTACTATAGCCTT-CTCCTCAGTAACCCAC

TTGGCTTTACCGGCAGAC-CCG-TTCCCGGAGGCTTGGGGG-GATTGGCG-TGTTGTAGGGCGCTGTGCG-
-TGCCGTGA-CGTC-AATTACGGCTGACTTATTCGTAACATCCA-TGCCAACGGAGCCTCTTTA-TT-CTT

-TATTCGTGGTACTT-GTGGCGGG-GGTGGTTGTGTGGGGGTTTTGGCGGTTCTGGGT-TTGTGGTGTA
GCA-TC-TACTTCCACATTGCCCGAGGACTTTA-TTACGGCTCACCTCTACAAAGAGACTTGAAATA-T
```

Figura 5. 26 Alineación Búho – Tiburón

Con este algoritmo se genera la puntuación establecida por el algoritmo “Dot-Plot” [6], dando así, una nueva posibilidad de alineación generada por el algoritmo de inteligencia artificial “Coste Uniforme”, con un resultado diferente a las generadas por los algoritmos clásicos Needleman y Smith-Waterman, dando al usuario una nueva posibilidad de alineación.

5.4 Experimentación Algoritmo Smith-Waterman optimizado con Colonia Artificial de Abejas

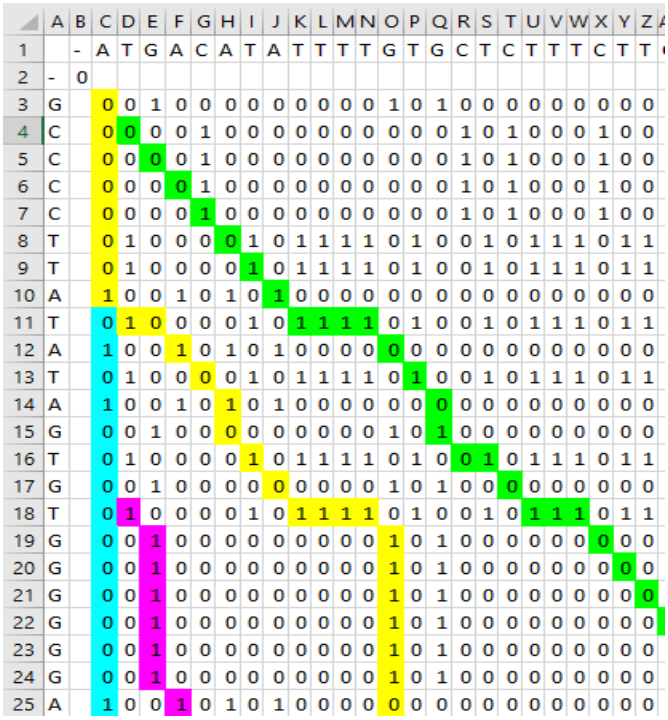
La experimentación con el algoritmo Smith-Waterman optimizado por Colonia Artificial de Abejas, comenzó inicialmente de manera manual para poder generar dos alineaciones diferentes, mediante la búsqueda de mejor puntuación en la tabla como lo hace el algoritmo tradicional y mediante la búsqueda de puntuación de manera inversa.

En la experimentación inicial se vio que, las alineaciones encontradas todas eran semejantes, por lo que se propuso la búsqueda de todos los caminos, evaluando todas las posibilidades mediante el algoritmo de abejas artificiales, tanto en la búsqueda original, como en la búsqueda inversa.

Los resultados obtenidos se muestran en la figura 5.27 y terminan con la figura 5.31, donde se muestra como las abejas encontraron nuevas posibilidades de alineamiento, siendo los colores Amarillo, Cian y Magenta, los resultados obtenidos en la búsqueda tradicional y los colores Verde, Rojo y Azul, siendo los resultados en la búsqueda de manera inversa.

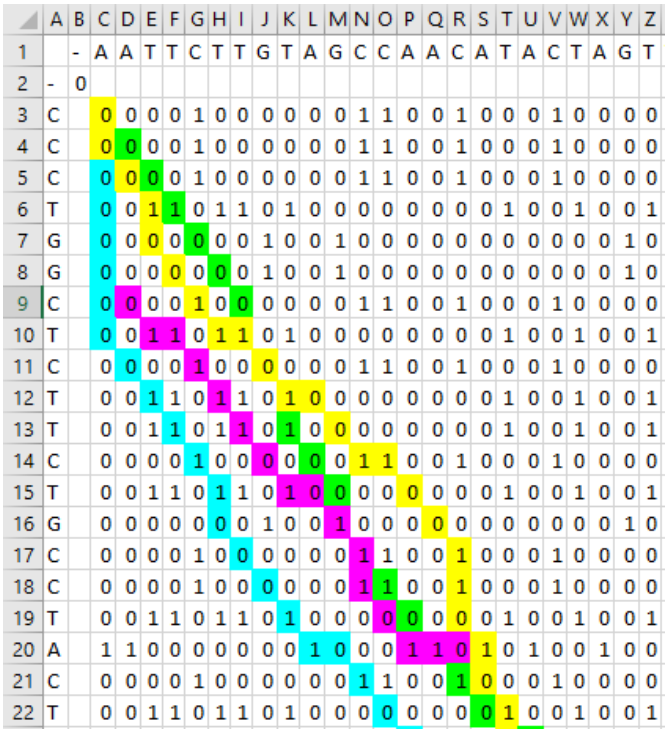
Se presentan las variantes de las alineaciones obtenidas por la optimización de Colonia de Abejas Artificiales, comenzando con la figura 5.32 y terminando en la figura 5.36, donde se muestran los caracteres alineados respecto al color de cada tabla.

Terminando con la figura 5.37, en donde se muestra el guardado de los archivos.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	A	T	G	A	C	A	T	A	T	T	T	T	G	T	G	C	T	C	T	T	T	C	T	T
2	-	0																								
3	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
4	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
5	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
6	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
7	C		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
8	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
9	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
10	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
12	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
14	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
16	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
17	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
18	T		0	1	0	0	0	0	1	0	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	1
19	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
20	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
21	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
22	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
23	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
24	G		0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
25	A		1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 5. 27 Colonia de abejas Búho – Gato



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	A	A	T	T	C	T	T	G	T	A	G	C	C	A	A	C	A	T	A	C	T	A	G	T
2	-	0																								
3	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
4	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
5	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
6	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
7	G		0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
8	G		0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
10	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
11	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
12	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
13	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
14	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
15	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
16	G		0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
17	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
18	C		0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0
19	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
20	A		1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1	0	0	1	0	0
21	C		0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0
22	T		0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1

Figura 5. 28 Colonia de abejas Tiburón – Gato

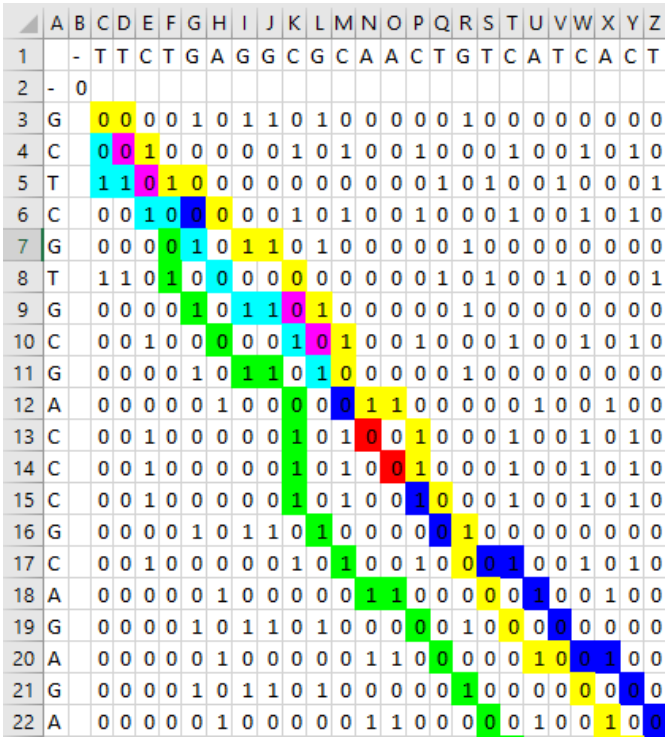


Figura 5. 29 Colonia de abejas Panda – Tiburón

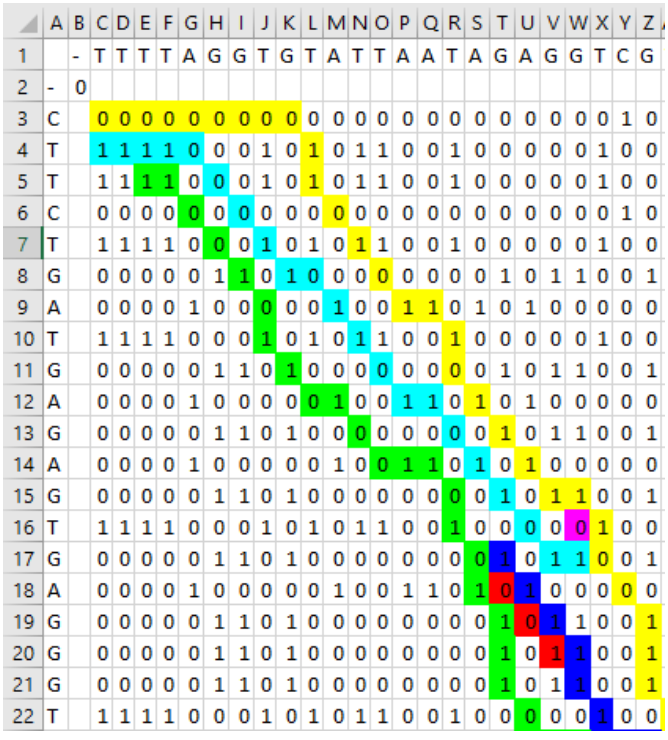


Figura 5. 30 Colonia de abejas Rinoceronte – Gato

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		-	G	A	A	T	C	T	G	A	G	G	T	G	G	C	T	T	C	T	C	A	G	T	A	G
2	-	0																								
3	A		0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
4	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
5	C		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
6	C		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
7	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
8	C		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
9	A		0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
10	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
11	A		0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
12	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0
13	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0
14	C		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
15	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
16	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
17	C		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
18	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
19	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
20	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1
21	C		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0
22	G		1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1

Figura 5. 31 Colonia de abejas Perro – Tiburón

Figura 5. 32 Colonia de abejas Panda – Rinoceronte

Figura 5. 33 Colonia de abejas Kiwi – Gato

Figura 5. 34 Colonia de abejas Búho – Gato

Figura 5. 35 Colonia de abejas Tiburón – Gato

Figura 5. 36 Colonia de abejas Panda – Gato

Con la implementación de la optimización del algoritmo Smith-Waterman con Colonia Artificial de Abejas, se obtuvieron seis resultados, de los cuales cinco son nuevos por completo, siendo el resultado marcado con el color verde, el resultado original con el algoritmo Smith-Waterman, el color rojo y azul sus abejas variantes y los colores amarillo, cyan y magenta, los resultados encontrados por las abejas, haciendo la alineación Smith-Waterman de manera inversa. Estas alineaciones, son nuevas propuestas generadas por el algoritmo de optimización. Los nuevos cinco resultados con colores amarillo, cyan, magenta, azul y rojo, no se habían obtenido con anterioridad, teniendo de esta manera, más opciones de alineamiento.

5.5 Resultados de similitud

Cada método cuenta con una métrica de alineación basado en puntajes obtenidos mediante el parecido de los caracteres. Recordando la puntuación Match, NoMatch y Gap. Debido a estas puntuaciones, se propuso un algoritmo simple para poder observar la puntuación en números de porcentaje, dando así un resultado visible de cuanto se parecen los genomas.

En el algoritmo de alineación Needleman [1], al generar los valores dentro de la tabla, existe un número que es el mayor dentro de la tabla, ese número representa el mayor puntaje alcanzado dentro de la alineación.

En el algoritmo propuesto con el método de Coste Uniforme [18], se obtiene un valor muy similar al obtenido en el algoritmo Needleman [1], sin embargo, en este caso, al buscar carácter por carácter, guarda los valores obtenidos dependiendo de cada nucleótido alineado.

En ambos algoritmos, Needleman [1] y Coste Uniforme [18] se obtiene un valor máximo por cada alineación. El algoritmo de porcentaje propuesto está basado en la mejor alineación y la peor alineación, donde la mejor alineación supone que todos los caracteres de la alineación sean iguales, dando un puntaje total al mismo número de caracteres de las alineaciones y siendo la peor alineación, en donde ningún carácter coincida, en cuyo caso sería el valor de cero.

Teniendo estos valores, se establece el método, donde la longitud más larga de las secuencias será el valor máximo dando el porcentaje de 100% y el mínimo siendo el cero o 0%. De esta manera el valor obtenido por los algoritmos se multiplica por el valor de 100 y se divide entre el valor máximo esperado, estos valores se guardan en el formato FASTA generado con la aplicación. Terminando las alineaciones, se escribe un apartado donde muestra, el valor esperado, el valor obtenido y el porcentaje final. De la tabla 5.1 hasta la tabla 5.3, se muestran los valores porcentuales obtenidos con el algoritmo Needleman [1] y de la tabla 5.4 hasta la tabla 5.6, se presentan los resultados porcentuales obtenidos con el algoritmo Coste Uniforme [18].

Tabla 5. 1 Valores obtenidos Needleman Panda – Kiwi

Porcentaje total	34.583 %
Puntos totales	5820
Puntos esperados	16829

Tabla 5. 2 Valores obtenidos Needleman Kiwi – Búho

Porcentaje total	37.067%
Puntos totales	407
Puntos esperados	1098

Tabla 5. 3 Valores obtenidos Needleman Búho – Gato

Porcentaje total	34.426 %
Puntos totales	378
Puntos esperados	1098

Tabla 5. 4 Valores obtenidos Coste Uniforme Perro – Tiburón

Porcentaje total	31.898%
Puntos totales	756
Puntos esperados	2370

Tabla 5. 5 Valores obtenidos Coste Uniforme Tiburón – Rinoceronte

Porcentaje total	32.065%
Puntos totales	354
Puntos esperados	1104

Tabla 5. 6 Valores obtenidos Coste Uniforme Rinoceronte – Gato

Porcentaje total	32.789%
Puntos totales	362
Puntos esperados	1104

En el algoritmo de alineación Smith-Waterman [17], la tabla se llena con ceros y unos, marcando un número cero cuando los nucleótidos no son similares y con el número uno cuando ambos son iguales. En este algoritmo se cuentan los números unos obtenidos al buscar la alineación. Cuando el algoritmo Smith-Waterman [17] se optimiza con el algoritmo Colonia Artificial de Abejas [21], se hace lo mismo, solo que se cuentan los números unos y se guardan los valores en variables de cada abeja.

Para el caso de Smith-Waterman [17] y Colonia de Abejas Artificiales [21], el algoritmo es el mismo, la matriz, genera una confusión al buscar los valores, debido a que se puede presentar que muchos números unos se encuentren de manera horizontal o vertical dentro de la matriz. Para resolver eso, solo se contaron los números uno, cuando son encontrados de manera diagonal, dando así el valor en puntos de las alineaciones, sin tener que modificar el valor máximo ni mínimo. La tabla 5.7 muestra el valor porcentual de Smith-Waterman [17] y de la tabla 5.8 a la tabla 5.12, muestra el porcentaje obtenido por el algoritmo Colonia de abejas Artificiales [21].

Tabla 5. 7 Smith Waterman Búho – Tiburón

Porcentaje total	53.531 %
Puntos totales	284
Puntos esperados	531

Tabla 5. 8 Abeja Azul Búho – Tiburón

Porcentaje total	50.758 %
Puntos totales	269
Puntos esperados	531

Tabla 5. 9 Abeja Roja Búho – Tiburón

Porcentaje total	48.636 %
Puntos totales	258
Puntos esperados	531

Tabla 5. 10 Abeja Verde Búho – Tiburón

Porcentaje total	52.655 %
Puntos totales	279
Puntos esperados	531

Tabla 5. 11 Abeja Cian Búho – Tiburón

Porcentaje total	51.939 %
Puntos totales	275
Puntos esperados	531

Tabla 5. 12 Abeja Magenta Búho – Tiburón

Porcentaje total	51.674 %
Puntos totales	274
Puntos esperados	531

5.6 Análisis de los resultados

En esta sección, se habla sobre las conclusiones del proyecto, hablando de manera general en cada uno de los resultados obtenidos por cada algoritmo.

A inicios de las experimentaciones, se creía que la lectura de las secuencias generaría un problema, sin embargo, los métodos utilizados para su lectura mostraron resultados de manera exitosa sin mostrar impedimentos para el trabajo de los algoritmos.

El algoritmo Coste Uniforme generó buenos resultados al poder alinear todos los caracteres de las secuencias sin pérdida de información, al igual que generando archivos de buena lectura en el formato FASTA.

En el proceso de escritura y lectura de tabla en los algoritmos Needleman y Smith-Waterman, se generaron correctamente mediante las librerías de Python, guardando los archivos con extensión .xlsx de manera correcta.

Al momento de la implementación del algoritmo de optimización Colonia de Abejas Artificiales en el algoritmo Smith-Waterman, las expectativas fueron superadas con resultados óptimos, dando posibilidades muy diferentes a las esperadas, dando como conclusión que el proceso de implementación fue exitoso, obteniendo un mínimo de tres alineaciones diferentes encontradas en una sola tabla.

Al comenzar las experimentaciones, se propuso que el algoritmo de optimización de Colonia de Abejas Artificiales se combinara con los algoritmos Needleman y Smith-Waterman. Sin embargo, se encontró una limitación en el algoritmo Needleman, debido a que, por su generación de tabla, obliga al proceso a seguir un solo camino para la alineación. Por esta razón, el algoritmo de optimización de Colonia de Abejas Artificiales, no pudo encontrar variantes dentro de la tabla, debido a que los números negativos provocaban que el proceso se alejara de una óptima posibilidad y, al hacerlo de manera inversa, el alineamiento no podía continuar con números negativo, siendo la mejor opción para el alineamiento, quedarse en la posición donde se encontraba, porque no existía mejor respuesta que el valor que le correspondía actual mente; razones que provocaban no encontrar ninguna alineación.

5.7 Comparación de resultados

Con cada algoritmo, se llegó a un resultado diferente, a excepción de dos, que fueron los algoritmos Smith-Waterman y Needleman.

Estos dos algoritmos se tomaron como punto de partida por ser los algoritmos clásicos. Sin embargo, los resultados obtenidos implementando los algoritmos de Coste Uniforme y Colonia Artificial de Abejas, mostraron resultados óptimos.

En la sección 5.5 se presentan las alineaciones y en la sección 5.6 se detallan los porcentajes de similitud, donde se observa como los valores pueden alejarse o acercarse unos de otros.

Los valores semejantes en las alineaciones fueron encontrados por los algoritmos Smith-Waterman y la optimización Colonia de Abejas Artificiales, muestreando que la abeja principal, dará el mismo resultado que el algoritmo clásico, mientras que las demás abejas, proponen otros caminos que se alejan o acercan al resultado clásico.

El algoritmo Coste Uniforme genera sus propias alineaciones. Los resultados obtenidos con este algoritmo pueden alejarse de los resultados obtenidos por los algoritmos clásicos y el optimizado, debido a que su desarrollo, no lo hace por base de matrices, si no, en base a árboles de decisión. Mientras que, para las Abejas en las matrices, la alineación genómica, puede estar muy alejada al encontrar una similitud de genes, este algoritmo, podría estar más cerca de encontrar un parentesco o viceversa.

Capítulo VI

CONCLUSIONES

6.1 Conclusiones Generales

El proyecto de tesis desarrollado cumple de manera satisfactorio los objetivos, general y específicos, como se puede ver en los capítulos anteriores se estudiaron e implementaron cada uno de los algoritmos considerados y con ello se logró con éxito, descrito en los siguientes puntos:

- Se encontraron nueve alineaciones diferentes al comparar dos secuencias genómicas, aumentando las posibilidades de alineación.
- Los algoritmos trabajaron de manera exitosa sin perder ningún solo dato al leer las secuencias.
- Se logro incluir la optimización de Colonia de Abejas Artificiales generando así múltiples opciones de alineamiento.
- El Framework se desarrolló completo y de manera sencilla

6.2 Cumplimiento de los Objetivos

Objetivo general:

"Desarrollar una aplicación Framework de manejo sencillo para usuarios con conocimientos básicos en alineación de secuencias genómicas, empleando métodos de inteligencia artificial que permitan encontrar, proporcionar y mostrar las mediciones de dichos alineamientos"

El objetivo general se cumplió, como se mostró en la experimentación al desarrollarse un Framework que realiza la alineación de las secuencias genómicas mediante dos métodos clásicos y dos métodos de inteligencia artificial.

Objetivos específicos:

• Leer secuencias genómicas	El objetivo se logró, al implementar un programa simple de lectura de archivos de texto utilizando el lenguaje Python. El sistema permite leer secuencias de tamaño indefinido.
• Implementar métodos de alineación clásicos (Needleman, Smith-Waterman)	La implementación de los algoritmos clásicos se logró generando un programa de archivos Excel con extensión .xlsx y su dicha escritura con la librería Openpyxl complementado con el programa de lectura de archivos de texto y su respectivo procesamiento de caracteres.

• **Implementar métodos de inteligencia artificial (Coste uniforme, Colonia Artificial de Abejas)**	La implementación de los dos algoritmos de inteligencia artificial se logró siendo el desarrollo del algoritmo de Coste Uniforme, generando los árboles de posibilidades con la lectura de los archivos de texto. De igual manera, la implementación del algoritmo Colonia de Abejas Artificiales, utilizando la librería Openpyxl para la escritura y lectura de las tablas de Excel, junto con el procesamiento de la librería Math de Python, para hacer los procesos matemáticos de desplazamiento y procesamiento matemático de los datos numéricos obtenidos por la tabla generada por el algoritmo Smith-Waterman.
• **Desarrollar una interfaz comprensible para cualquier usuario**	El Framework fue desarrollado con la librería Tkinter, teniendo en mente en que el manejo de este Framework debe ser sencillo y comprensible para cualquier usuario.

6.3 Aportaciones

- Desarrollo de un Framework sencillo y comprensible que permite realizar alineaciones con 4 algoritmos; siendo capaz de leer secuencias de tamaño indefinido.
- La herramienta es flexible y permite llevar a cabo la implementación de más algoritmos sin confundirse, ya que el Framework está debidamente estructurado.
- La implementación del algoritmo de Colonia de Abejas Artificiales [21] se implementó de manera óptima al algoritmo Smith-Waterman [17], encontrando diferentes alineaciones, logrando diferentes resultados óptimos.

6.4 - Trabajo Futuro

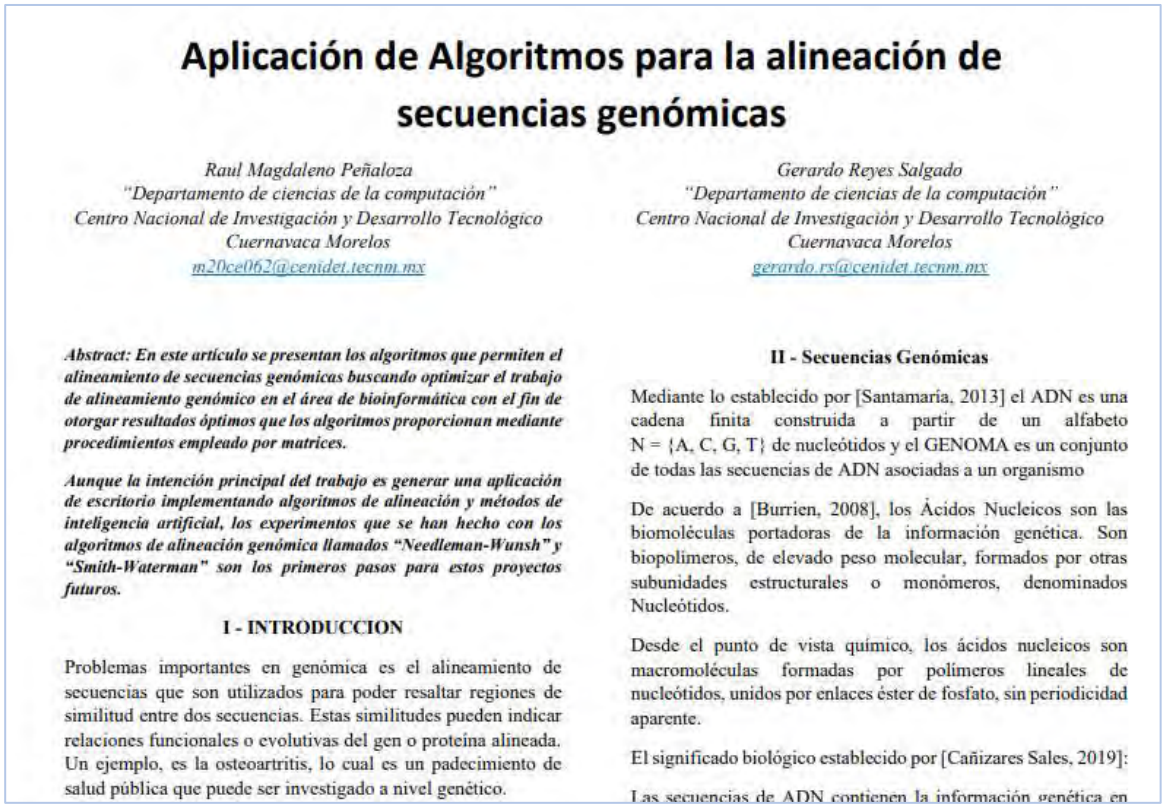
Al hacer las investigaciones de este proyecto, se propusieron nuevas ideas para optimizar el sistema, entre ellas el hacer uso de redes neuronales. El trabajo futuro consiste, en desarrollar una red neuronal multicapa que busque nuevas alineaciones detectando similitudes entre las secuencias genómicas.

Una segunda propuesta es, adaptar una pequeña red neuronal multicapa a cada una de las abejas, que podría darle una libertad mayor al explorar posibilidades.

6.5 Trabajos académicos adicionales

Con este proyecto, se publicaron tres artículos, ver figuras 6.1, 6.2 y 6.3:

- *"Aplicación de Algoritmos para la alineación de secuencias genómica"* Escuela de Inteligencia Computacional y Robótica 2021, 21 - 25 de junio 2021, Universidad Tecnológica Emiliano Zapata del Estado de Morelos. Temixco. Morelos.



Aplicación de Algoritmos para la alineación de secuencias genómicas

Raul Magdaleno Peñaloza
"Departamento de ciencias de la computación"
Centro Nacional de Investigación y Desarrollo Tecnológico
Cuernavaca Morelos
m20ce062@cenidet.tecnm.mx

Gerardo Reyes Salgado
"Departamento de ciencias de la computación"
Centro Nacional de Investigación y Desarrollo Tecnológico
Cuernavaca Morelos
gerardo.rs@cenidet.tecnm.mx

***Abstract:** En este artículo se presentan los algoritmos que permiten el alineamiento de secuencias genómicas buscando optimizar el trabajo de alineamiento genómico en el área de bioinformática con el fin de otorgar resultados óptimos que los algoritmos proporcionan mediante procedimientos empleado por matrices.*

Aunque la intención principal del trabajo es generar una aplicación de escritorio implementando algoritmos de alineación y métodos de inteligencia artificial, los experimentos que se han hecho con los algoritmos de alineación genómica llamados "Needleman-Wunsh" y "Smith-Waterman" son los primeros pasos para estos proyectos futuros.

I - INTRODUCCION

Problemas importantes en genómica es el alineamiento de secuencias que son utilizados para poder resaltar regiones de similitud entre dos secuencias. Estas similitudes pueden indicar relaciones funcionales o evolutivas del gen o proteína alineada. Un ejemplo, es la osteoartritis, lo cual es un padecimiento de salud pública que puede ser investigado a nivel genético.

II - Secuencias Genómicas

Mediante lo establecido por [Santamaría, 2013] el ADN es una cadena finita construida a partir de un alfabeto N = {A, C, G, T} de nucleótidos y el GENOMA es un conjunto de todas las secuencias de ADN asociadas a un organismo

De acuerdo a [Burrien, 2008], los Ácidos Nucleicos son las biomoléculas portadoras de la información genética. Son biopolímeros, de elevado peso molecular, formados por otras subunidades estructurales o monómeros, denominados Nucleótidos.

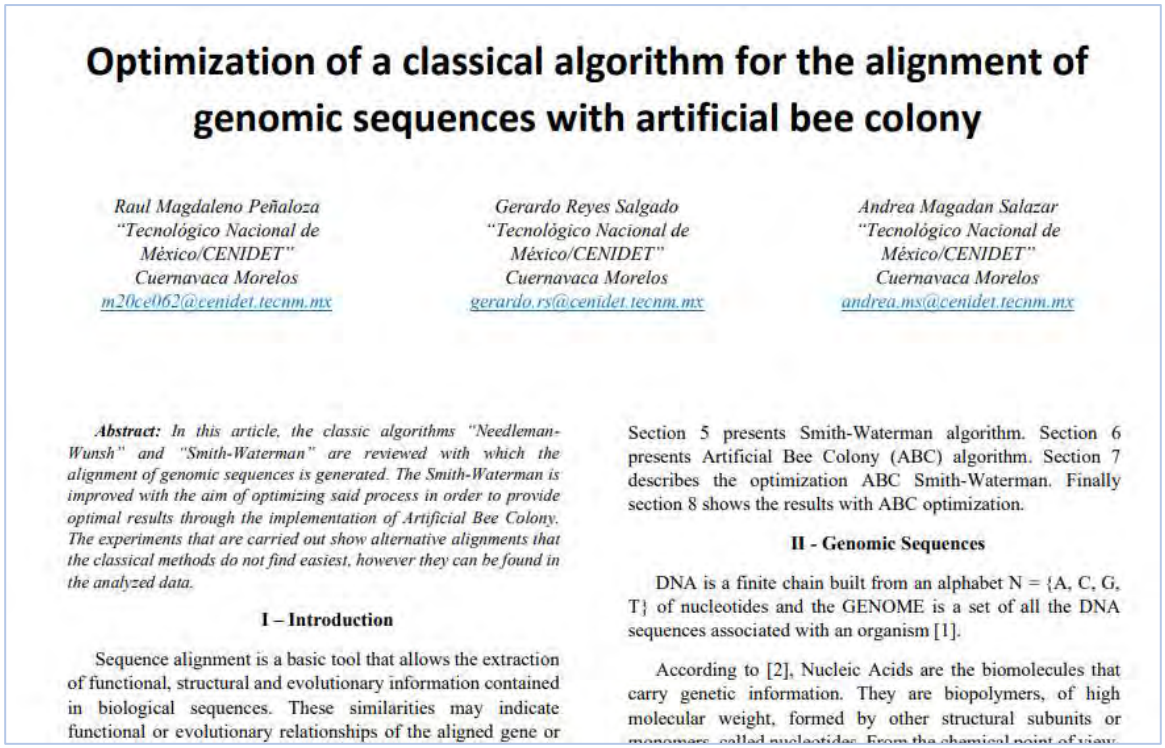
Desde el punto de vista químico, los ácidos nucleicos son macromoléculas formadas por polímeros lineales de nucleótidos, unidos por enlaces éster de fosfato, sin periodicidad aparente.

El significado biológico establecido por [Cañizares Sales, 2019]:

Las secuencias de ADN contienen la información genética en

Figura 6. 1 Artículo Publicado Escuela de Inteligencia Artificial, Zapata Morelos.

- “Optimization of a Classical Algorithm for the Alignment of Genomic Sequences with Artificial Bee Colony”. En el The 2021 International Conference on Computational Science and Computational Intelligence (CSCI’21), December 15-17, 2021, Luxor Hotel (MGM Property), 3900 Las Vegas Blvd. South, Las Vegas, 89109, USA. https://www.american-cse.org/csci2021/



Optimization of a classical algorithm for the alignment of genomic sequences with artificial bee colony

Raul Magdaleno Peñaloza
“Tecnológico Nacional de México/CENIDET”
Cuernavaca Morelos
m20ce062@cenidet.tecnm.mx

Gerardo Reyes Salgado
“Tecnológico Nacional de México/CENIDET”
Cuernavaca Morelos
gerardo.rs@cenidet.tecnm.mx

Andrea Magadan Salazar
“Tecnológico Nacional de México/CENIDET”
Cuernavaca Morelos
andrea.ms@cenidet.tecnm.mx

Abstract: *In this article, the classic algorithms “Needleman-Wunsh” and “Smith-Waterman” are reviewed with which the alignment of genomic sequences is generated. The Smith-Waterman is improved with the aim of optimizing said process in order to provide optimal results through the implementation of Artificial Bee Colony. The experiments that are carried out show alternative alignments that the classical methods do not find easiest, however they can be found in the analyzed data.*

I – Introduction

Sequence alignment is a basic tool that allows the extraction of functional, structural and evolutionary information contained in biological sequences. These similarities may indicate functional or evolutionary relationships of the aligned gene or

Section 5 presents Smith-Waterman algorithm. Section 6 presents Artificial Bee Colony (ABC) algorithm. Section 7 describes the optimization ABC Smith-Waterman. Finally section 8 shows the results with ABC optimization.

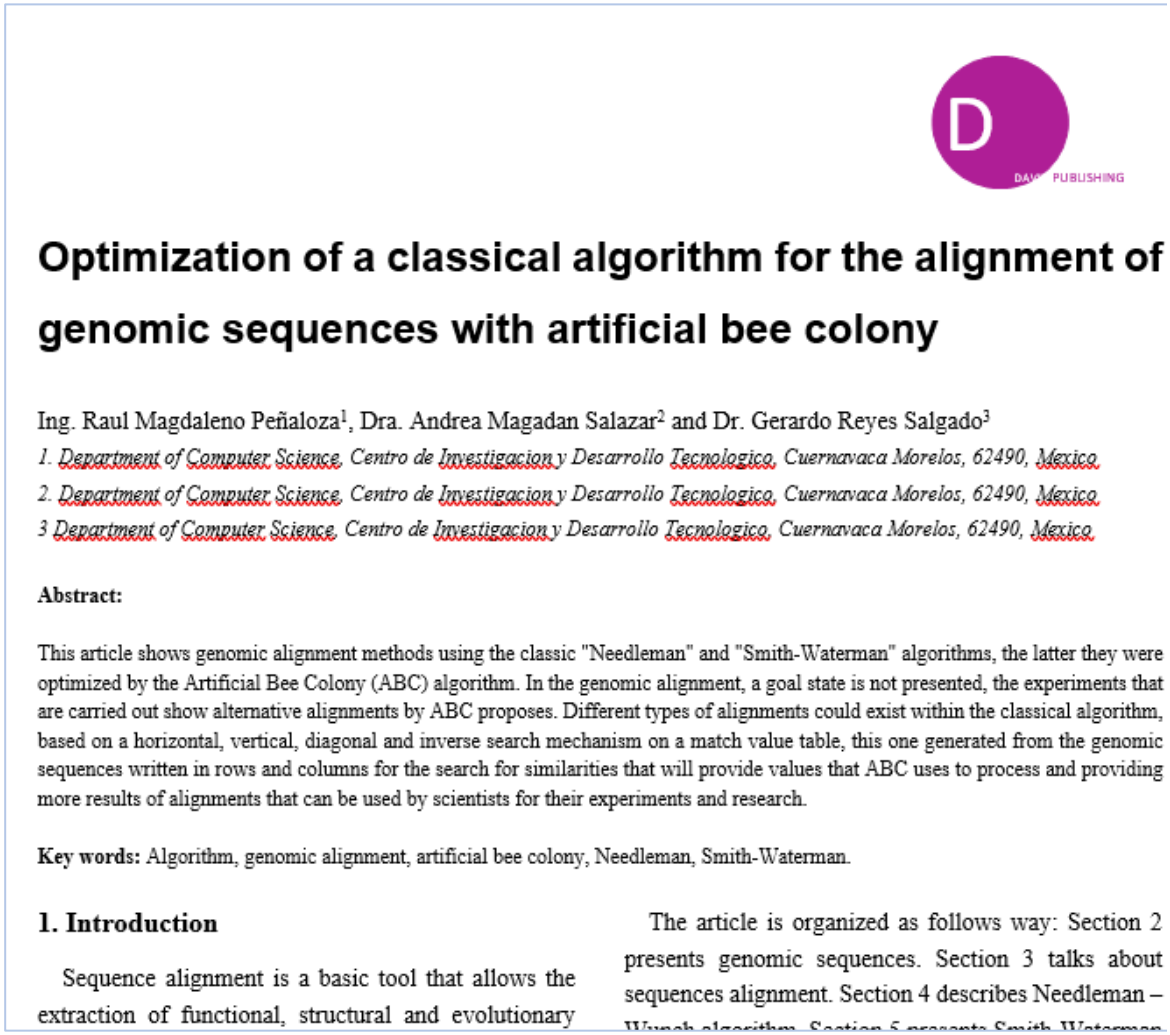
II - Genomic Sequences

DNA is a finite chain built from an alphabet N = {A, C, G, T} of nucleotides and the GENOME is a set of all the DNA sequences associated with an organism [1].

According to [2], Nucleic Acids are the biomolecules that carry genetic information. They are biopolymers, of high molecular weight, formed by other structural subunits or

Figura 6. 2 Artículo Publicado en CSCI – LAS VEGAS, NV

- "Optimization of a Classical Algorithm for the Alignment of Genomic Sequences with Artificial Bee Colony" David Publishing Company, 616 Corporate Way, Suite 2-4876, Valley Cottage, NY 10989, USA TEL: 323-984-7526, FAX: 323-984-7374, http://www.davidpublisher.com, order@davidpublishing.com



Optimization of a classical algorithm for the alignment of genomic sequences with artificial bee colony

Ing. Raul Magdaleno Peñaloza[1], Dra. Andrea Magadan Salazar[2] and Dr. Gerardo Reyes Salgado[3]

1. Department of Computer Science, Centro de Investigacion y Desarrollo Tecnologico, Cuernavaca Morelos, 62490, Mexico

2. Department of Computer Science, Centro de Investigacion y Desarrollo Tecnologico, Cuernavaca Morelos, 62490, Mexico

3 Department of Computer Science, Centro de Investigacion y Desarrollo Tecnologico, Cuernavaca Morelos, 62490, Mexico

Abstract:

This article shows genomic alignment methods using the classic "Needleman" and "Smith-Waterman" algorithms, the latter they were optimized by the Artificial Bee Colony (ABC) algorithm. In the genomic alignment, a goal state is not presented, the experiments that are carried out show alternative alignments by ABC proposes. Different types of alignments could exist within the classical algorithm, based on a horizontal, vertical, diagonal and inverse search mechanism on a match value table, this one generated from the genomic sequences written in rows and columns for the search for similarities that will provide values that ABC uses to process and providing more results of alignments that can be used by scientists for their experiments and research.

Key words: Algorithm, genomic alignment, artificial bee colony, Needleman, Smith-Waterman.

1. Introduction

Sequence alignment is a basic tool that allows the extraction of functional, structural and evolutionary

The article is organized as follows way: Section 2 presents genomic sequences. Section 3 talks about sequences alignment. Section 4 describes Needleman – Wunsch algorithm. Section 5 presents Smith-Waterman

Figura 6. 3 Artículo Publicado Libro Journal of Mechanics Engineering and Automation

REFERENCIAS

[1] – Saul B. Needleman and Christian D. Wunch. (1969). A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. Department of Biochemistry

[2] – Python. (2022, 24 marzo). Python.Org. *https://www.python.org/*

[3] - Verónica Burriel Coll. (16 de mayo de 2008). Estructura y propiedades de los ácidos nucleicos. Química Aplicada a la Ingeniería Biomédica Master en Ingeniería Biomédica (UV - UPV), 17.

[4] - Joaquín Cañizares Sales, José Blanca, Peio Ziarsolo(2019, Nov. 25)Alineamiento de secuencias. (s. f.-b). bioinf.comav.upv.es.

[5] –Rodrigo Santamaria. (2013, 10 abril). Alineamiento de pares de secuencias

[6] - Juan Manuel González Mañas. (2020). Comparación de secuencias. 7 diciembre 2020, de Universidad del País Vasco Sitio web: http://www.ehu.eus/biofisica/juanma/bioinf/pdf/0_intro.pdf

[7] - Verónica Burriel Coll. (16 de mayo de 2008). Estructura y propiedades de los ácidos nucleicos. Química Aplicada a la Ingeniería Biomédica (UV - UPV).
Recuperado:
https://www.researchgate.net/profile/Adrian_Gibbs/publication/229616013_The_Diagram_a_Method_for_Comparing_Sequences_Its_Use_with_Amino_Acid_and_Nucleotide_Sequences/links/5c58a285a6fdccd6b5e269bc/The-Diagram-a-Method-for-Comparing-Sequences-Its-Use-with-Amino-Acid-and-Nucleotide-Sequences.pdf

[8] – National Center for Biotechnology Information
Ncbi.nlm.nih.gov
URL: https://www.ncbi.nlm.nih.gov/

[9] – Dna DataBase of Japan
Ddbj.nig.ac.jp
URL: https://www.ddbj.nig.ac.jp/about/index-e.html

[10] – The European Bioinformatics Institute < EMBL-EBI
Ebi.ac.uk
URL: https://www.ebi.ac.uk/

[11] - International Nucleotide Sequence Database Collaboration | INSDC
Insdc.org
URL: http://www.insdc.org/

[12] - Backofen, R. (17 de May de 2011). Sequence Alignment Needleman-Wunsch. Obtenido de Uni Freiburg Bioinformatics.

[13] - ARIAS-LOPEZ, Mauricio and VELASCO-MEDINA, Jaime. Implementación hardware del algoritmo de Needleman-Wunsch modificado usando una arquitectura paralela. Rev. ing. biomed. [online]. 2018, vol.12, n.23, pp.53-62. ISSN 1909-9762.

[14] - VladimirLikic. (2016). TheNeedleman-Wunsch algorithm for sequence alignment. Molecular Science and Biotechnology Institute The University of Melbourne, 46.

[15] - Backofen, R. (2018) Teaching - Smith-Waterman. Obtenido de Uni Freiburg Bioinformatics. Recuperado: http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Smith-Waterman

[16] - Svetlin A Manavski and Giorgio Valle. (26 March 2008). CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. BioMed Central, -, 9

[17] - SAUL B. NEEDLEMAN AND CHRISTIAN D. WUNCH. (1969). A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. Department of Biochemistry.

[18] - García Serrano, A. (2016). Inteligencia artificial, fundamentos, práctica y aplicaciones: Vol. (2da edición revisada ed.). Alfaomega.

[19] – Marco Antonio Aceves Fernández. (2021). Inteligencia Artificial Para Programadores con Prisa. Las Vegas, NV: Amazon.

[20] – Adán Enrique Aguilar Justo. (2014). Un algoritmo basado en la colonia articial de abejas con búsqueda local para resolver problemas de optimización con restricciones. Xalapa Veracruz. México.: Universidad Veracruzana

[21] - Karaboga, D. (2010, 30 marzo). Artificial bee colony algorithm - Scholarpedia. Scholarpedia.Org. Recuperado 29 de octubre de 2021

[22] – Silvana Yanet García. (2020). Optimización mediante el algoritmo de colonia de abejas artificial. General Pico, La Pampa, Argentina: Universidad Nacional de La Pampa.

[23] - Alineación de secuencias genómicas utilizando el algoritmo de búsqueda Best First Search. A. Aranda-Díaz. Tecnológico Nacional de México / CENIDET, Tesis de maestría en Ciencias Computacionales en desarrollo. Cuernavaca, Morelos, México; Enero-2019

[24] - Carlos Alberto Moncada Vázquez. (junio del 2019). Análisis de datos genómicos para el diagnóstico temprano de osteosarcoma. Cuernavaca, Morelos, México. Tesis de maestría, en

Ciencias Computacionales. Tecnológico Nacional de México Centro Nacional de Investigación y Desarrollo Tecnológico.

[25] - ALGORITMO DE ALINEACION DE SECUENCIAS PARA ENFERMEDADES DEL SISTEMA NERVIOSO CENTRAL. Gustavo Adolfo Higuera Castro. Leidy Yolanda López Osorio. Andrea Yulieth Yara Rodríguez. Liliana Arévalo Tapias. 2017

[26] - Implementación y Análisis de Algoritmos de Alineación para datos de Next Generación Sequencing (NGS) Daniel Giménez Llorente, junio 2016

[27] - Genómica comparada de dos dianas moleculares en modelos animales de hipersensibilidad. Serrano-Barrera OR, Hernández-Betancourt JC 2017

[28] - Documentación y Análisis de los Principales Frameworks de Arquitectura de Software en Aplicaciones Empresariales. Ing. Hugo Fernando Sarasty España, octubre, 2015.

[29] - Uso de algoritmos de aprendizaje automático aplicado a bases de datos genéticos. Rosario Gago Utrera. Junio 2017

[30] - L. Charles Bailey, Jr.,1 Stephen Fischer, Jonathan Schug, Jonathan Crabtree, Mark Gibson, and G. Christian Overton. (December 5 1997). GAIA: Framework Annotation of Genomic Sequence. Genome Research, 1, 17.A

[31] - BioInteractive. (Marzo 2014). Alineamiento de Secuencias Usando ClustalX. free Resources of Sience Teachers and Students, https://www.biointeractive.org/, 1 - 4.

[32] - Johan S. Piña, Simon Orozco-Arias, Nicolás Tobón-Orozco, Mariana S. Candamil-Cortés, Reinel Tabares-Soto, Romain Guyot. (29 de julio de 2019). Alineamiento gráfico de secuencias a través de programación paralela: un enfoque desde la era postgenómica. Revista Ingeniería Biomédica, Volumen 13, 37-45.

[33] - Ramu Chenna, Hideaki Sugawara, Tadashi Koike, Rodrigo Lopez, Toby J. Gibson, Desmond G. Higgins and Julie D. Thompson. (March 4, 2003). Multiple sequence alignment with the Clustal series of programs. National Library of Medicine, Nucleic Acids Research, Vol. 31, 3497–3500.

[34] - Adrian J. GIBBS, George A. MCINTYR. (May 4, 2005). The Diagram, a Method for Comparing Sequences Its Use with Amino Acid and Nucleotide Sequences. European Journal of Biochemistry, Vol. 16, 1 - 11.

[35] - Dervis Karaboga and Bahriye Basturk. (January 2007). Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. Conference: Foundations of Fuzzy Logic and Soft Computing, 12th International Fuzzy Systems Association World Congress, 12, 789 -798.

[36] - Silvana Yanet García. (2017). Optimización mediante el algoritmo de colonia de abejas artificial. Universidad Nacional de La Pampa: General Pico, La Pampa, Argentina.

[37] - Sergio de los Cobos Silva, Miguel A. Gutierrez Andrade, Eric A. Rincon Garcia, Perdo Lara Velazquez, Miguel Aguilar Cornejo. (10 - Julio - 2013). Colonia de Abejas Artificiales y Optimización por Enjambre de Partículas para la Estimación de Parámetros de Regresión no Lineal. Revista de Matemática: Teoría y Aplicaciones, Vol. 21, 107–126.

[38] - Cuevas, Erik. (2015). El algoritmo "Artificial Bee Colony" (ABC) y su uso en el Procesamiento digital de Imágenes. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, Vol. 18 Num. 55, 1 - 19.

[39] – openpyxl. (2021, 22 septiembre). PyPI. https://pypi.org/project/openpyxl/

[40] – os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.4 https://docs.python.org/es/3.10/library/os.html

[41] – math — Mathematical functions — Python 3.10.4 documentation https://docs.python.org/3/library/math.html

[42] – tkinter — Interface de Python para Tcl/Tk — documentación de Python - 3.10.4 https://docs.python.org/es/3/library/tkinter.html

[43] - Alineamiento de secuencias. (s. f.). bioinf.comav.upv.es. Recuperado 10 de diciembre de 2020. Recuperado: https://bioinf.comav.upv.es/courses/intro_bioinf/alineamientos.html