Centro de Ciencias Básicas
Departamento de Sistemas de Información

Tesis

Clusterización de Proteínas Mediante Metaheurísticas Multiobjetivo en la Familia Coronaviridae

Presenta
Ing. Sergio Iván Galvis Motoa

Para Obtener El Grado De Maestro En Informática Y Tecnologías Computacionales

Tutora
Dra. Eunice Esther Ponce de León Sentí

Comité Tutoral
Dra. María Dolores Torres Soto

Dra. Aurora Torres Soto

Aguascalientes, Ags, 9 de junio de 2022

Autorizaciones.

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

CENTRO DE CIENCIAS BÁSICAS

CARTA DE VOTO APROBATORIO INDIVIDUAL

M. en C. Jorge Martín Alférez Chávez
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como ***TUTOR*** designado del ***SERGIO IVÁN GALVIS MOTOA*** con ID 279841 quien realizó *la tesis* titulado: ***CLUSTERIZACIÓN DE PROTEÍNAS MEDIANTE METAHEURÍSTICAS MULTIOBJETIVO EN LA FAMILIA CORONAVIRIDAE***, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a 10 días del mes de junio de 2022.

Dra. Eunice Esther Ponce de León Sentí
Tutor de *tesis*

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

Elaborado por: Depto. Apoyo al Posgrado.
Revisado por: Depto. Control Escolar/Depto. Gestión de Calidad.
Aprobado por: Depto. Control Escolar/ Depto. Apoyo al Posgrado.

Código: DO-SEE-FO-0
Actualización: 01
Emisión: 17/05/19

UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

CARTA DE VOTO APROBATORIO INDIVIDUAL

M. en C. Jorge Martín Alférez Chávez
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como ***ASESOR*** designado del ***SERGIO IVÁN GALVIS MOTOA*** con ID 279841 quien realizó *la tesis* titulado: ***CLUSTERIZACIÓN DE PROTEÍNAS MEDIANTE METAHEURÍSTICAS MULTIOBJETIVO EN LA FAMILIA CORONAVIRIDAE***, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a día 10 de junio de 2022.

Dra. Aurora Torres Soto
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

CARTA DE VOTO APROBATORIO INDIVIDUAL

M. en C. Jorge Martín Alférez Chávez
DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE

Por medio del presente como ***ASESOR*** designado del ***SERGIO IVÁN GALVIS MOTOA*** con ID 279841 quien realizó *la tesis* titulado: ***CLUSTERIZACIÓN DE PROTEÍNAS MEDIANTE METAHEURÍSTICAS MULTIOBJETIVO EN LA FAMILIA CORONAVIRIDAE***, un trabajo propio, innovador, relevante e inédito y con fundamento en el Artículo 175, Apartado II del Reglamento General de Docencia doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el **VOTO APROBATORIO**, para que él pueda proceder a imprimirla así como continuar con el procedimiento administrativo para la obtención del grado.

Pongo lo anterior a su digna consideración y sin otro particular por el momento, me permito enviarle un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., a 10 día de junio de 2022.

Dra. María Dolores Torres Soto
Asesor de tesis

c.c.p.- Interesado
c.c.p.- Secretaría Técnica del Programa de Posgrado

DICTAMEN DE LIBERACION ACADEMICA PARA INICIAR LOS TRAMITES DEL EXAMEN DE GRADO

Fecha de dictaminación dd/mm/aaaa: 10/06/22

NOMBRE: Sergio Iván Galvis Motoa **ID** 279841

PROGRAMA: Maestría en Informática y Tecnologías Computacionales **LGAC (del posgrado):** Ingeniería de sistemas decisionales para mejorar procesos organizacionales

TIPO DE TRABAJO: (X) Tesis ( ) Trabajo Práctico

TITULO: Clusterización de Proteínas Mediante Metaheurísticas Multiobjetivo en la Familia Coronaviridae

IMPACTO SOCIAL (señalar el impacto logrado): Este trabajo aporta información sobre organismos de la familia coronaviridae.

INDICAR SI NO N.A. (NO APLICA) SEGÚN CORRESPONDA:

	Elementos para la revisión académica del trabajo de tesis o trabajo práctico:
SI	El trabajo es congruente con las LGAC del programa de posgrado
SI	La problemática fue abordada desde un enfoque multidisciplinario
SI	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI	Los resultados presentados en el trabajo son de gran relevancia científica, tecnológica o profesional según el área
SI	El trabajo demuestra más de una aportación original al conocimiento de su área
SI	Las aportaciones responden a los problemas prioritarios del país
SI	Generó transferencia del conocimiento o tecnológica
SI	Cumple con la ética para la investigación (reporte de la herramienta antiplagio)
	El egresado cumple con lo siguiente:
SI	Cumple con lo señalado por el Reglamento General de Docencia
SI	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias,estancia, predoctoral, etc)
SI	Cuenta con los votos aprobatorios del comité tutoral, en caso de los posgrados profesionales si tiene solo tutorpodrá liberar solo el tutor
SI	Cuenta con la carta de satisfacción del Usuario
SI	Coincide con el título y objetivo registrado
SI	Tiene congruencia con cuerpos académicos
SI	Tiene el CVU del Conacyt actualizado
NO	Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)
	En caso de Tesis por artículos científicos publicados
N.A.	Aceptación o Publicación de los artículos según el nivel del programa
N.A.	El estudiante es el primer autor
N.A.	El autor de correspondencia es el Tutor del Núcleo Académico Básico
N.A.	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabajo de investigación.
N.A.	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
N.A.	La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado: SI X NO

FIRMAS

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCION: Dr. José Manuel Mora Tavarez

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO: MTIC Jorge Prestado Maclas Olivano

* En caso de conflicto de intereses, firmará un revisor asignado del NAB de la LGAC correspondiente distinto al tutor o miembro del comité tutoral, asignado por el Decano.

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO: Dra. Haydee Martínez Bovadilla

Autorizó:

NOMBRE Y FIRMA DEL DECANO: M. En C. Jorge Martín Alférez Chávez

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico ... Cuidar la eficiencia terminal del programa de posgrado y el Art. 109F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.

Agradecimientos.

A CONACyT, agradezco la beca que fue una ayuda determinante en este proceso académico.
A la Benemérita Universidad Autónoma de Aguascalientes por darme la oportunidad de sentirme un gallo en el camino de proyectarse en luz. A las personas que realizaron diversas gestiones administrativas permitiendo mi llegada al país. Sin lugar a duda agradecer a las y los excelentes docentes y su disposición para compartir y motivar a la construcción del conocimiento. Al ingeniero Eduardo Mauricio Martin Alvarez por su mano guía en el mundo de la Biología Computacional.
A la Dra. Eunice, por la paciencia, sus valores y calidez humana con la que transmite el gusto por la ciencia y la investigación además de la confianza depositada en mí.
A las Dras. Torres Soto por su ejemplo de profesionalismo.
A los estudiantes de pregrado a quienes tuve la oportunidad de acompañar en sus proyectos.

También un recuerdo de gratitud para esos compañeros del aula, que ahora son amigos para la vida.

Es importante reconocer y agradecer a quien con su confianza me acompaño a las puertas de la universidad y me recargo de motivos para no desfallecer.

A mis ejemplares tíos: Rocío y Luis Fernando que llenaron mi infancia de ciencia, tecnología y amor por el conocimiento.

Al ejemplo que me dio mi hermanote, Juan Sebastián, que con todos los miedos me demostró que si podemos abrir puertas y transitar nuevos caminos.

Finalmente, a quienes en su momento me ayudaron, opinaron, o simplemente escucharon de este proceso y no se identifiquen en las anteriores líneas, a ellas y ellos también gracias.

Dedicatorias.

Hasta el cielo a mi madre. Clemencia.
Tu amor siempre nos ilumina el camino.

Índice General

Índice de Figuras

Índice de Tablas

Índice de Ecuaciones

Resumen.

Actualmente la población mundial ha conocido sobre la existencia del coronavirus y las enfermedades que puede causar, así como las consecuencias que conlleva una pandemia. Sin embargo, no todas las personas conocen la existencia de la familia coronaviridae y el historial de los problemas que ha generado, debido a que la mayoría de estos virus usan animales como huéspedes, mayormente murciélagos. Este trabajo tiene por objetivo presentar un modelo que construya grupos de organismos basado en las similitudes de las proteínas que lo integran partiendo de la utilización del criterio de mejores aciertos bidireccionales (MAB o BBH). Se plantea como un problema de optimización combinatoria multiobjetivo y se aplican técnicas computacionales con algoritmos propuestos para el análisis de otros organismos. Durante el avance del proyecto, fue necesaria la construcción de piezas de software con el fin de aportar a la eficiencia de las metodologías usadas por medio de la automatización de la adquisición de los datos. Se ha trabajado con 68 organismos de los más de 200 reportados en la familia de los coronaviridae. Los proteomas fueron obtenidos de repositorios de secuencias que previamente han sido curados y dan confianza en la calidad de la información que contienen. En el primer capítulo se extiende un compendio de las temáticas de mayor relevancia para el desarrollo del proyecto. El contenido del segundo capítulo es la descripción del proceso de obtención de los datos, y la aportación de una herramienta que automatiza su preprocesamiento. Durante el capítulo tercero se presenta el modelo propuesto para resolver mediante una metaheurística el problema de optimización multiobjetivo de la búsqueda de grupos de proteínas y se relata la selección de una metaheurística de carácter poblacional como es el EDA o Algoritmo de Estimación de la Distribución, que ha sido la herramienta seleccionada para realizar la búsqueda de los grupos de datos luego de su alistamiento y procesamiento. En el mismo capítulo se presentan las funciones objetivo seleccionadas para la clusterización y subsecuentemente la salida obtenida de la ejecución de los experimentos. La sección de resultados muestra las agrupaciones propuestas por los clústeres generados con la ayuda de la metaheurística que agrupa respecto de las proteínas y las similitudes que éstas tienen. El trabajo desarrollado y explicado en este documento trata de realizar una contribución con la identificación de grupos de proteínas de los miembros de la familia coronaviridae usando una perspectiva desde el área proteómica. Esto conllevará a que los esfuerzos de entender estos organismos (también se puede aplicar a otros organismos), tengan en cuenta los elementos que son comunes entre ellos y se puedan tomar acciones que contrarresten su impacto en la humanidad a manera de familia y no de un solo organismo.

Abstract.

Nowadays the global population knows about coronavirus and the diseases that it can produce and, more important, the consequences in a lot of ways of a pandemic. But not all people know about the existence of the coronaviridae family, and the record of problems generated, because in a lot of this viruses the hosts is any animal, in a greater extent, bats. The aim of the work is to present a model to build a set of organisms based in the similarity of the proteins that make it up supported in the Bidirectional Best Hits (BBH) concept. The work developed and explained here try to contribute with the identification for a set of characteristics of the coronaviridae family members using a perspective in the proteomic area. Applying biocomputational techniques mixed with algorithms proposed for analyze other organisms. While this project was necessary develop a software piece aimed to support of the used methodologies. Has been used 68 organisms from over 200 reported in the coronaviridae family. The proteomes have been obtained from reviewed sequences repositories to have more confidence in the quality of the information contained. The first chapter confines an extended the more relevant syllabus needed for the project. In the second chapter it is described the acquisition of the proteomic data, with the developed tool for the automation of the data preprocessing. For the third chapter the aim is to relate the specific approach to the optimization problem. Also, this chapter contains the object functions selected for clustering and the metaheuristic selection process, the EDA (Estimation Distribution Algorithm) chosen for look the groups of data after the preprocess. Subsequently in the same chapter the output of the experiments performed will be presented. The results section of the document shows the sets proposed for the generated clusters based in the selected metaheuristic that make groups based in the proteins and their similarity. It will drive for an effort for understand these organisms contemplating the common elements between they and take actions to reduce their impact in the human spice as a family and not as a single organism.

1. Introducción.

El interés para la realización de este proyecto se enmarca en la pandemia mundial decretada en 2020 por la WHO (*World Health Organization*) por causa del virus SARS–CoV-2 (*Severe Acute Respiratory Syndrome Coronavirus Two*) y la enfermedad que produce: COVID-19. Esta situación demanda de toda la investigación, esfuerzo y dedicación para incrementar el conocimiento referente al virus y la enfermedad con la finalidad de alcanzar una cura, vacuna, tratamientos y todo tipo de medidas que ayuden a contrarrestar su propagación.

En las bases de datos del NCBI (*National Center for Biotechnology Information*) se logran encontrar alrededor de 200 Genomas reportados de la familia de los coronaviridae (*Genome List - Genome - NCBI*, n.d.-a). La mutación de alguno de estos especímenes puede suceder (Enjuanes Sánchez et al., 2011). Esto es suficiente motivo para profundizar el conocimiento que se tiene de esta familia de microrganismos no vivos que en tan poco tiempo han causado demasiada afectación en los seres humanos.

Las capacidades computacionales actuales, aunque son mucho mayores que al inicio de la década, aún tienen limitaciones de procesamiento y tiempos de ejecución para algoritmos que requieren manejar volúmenes de datos considerables, múltiples archivos, o tareas que se consideren parte de problemas intratables en las Ciencias de la Computación (Garey & Johnson, 1979). Estos problemas intratables son de alta complejidad computacional, pues se entiende que para llegar a la solución exacta se requiere de un tiempo que no se justifica, es decir, *el tiempo de espera se hace impráctico cuando el tamaño del problema crece*. Algunos de estos son problemas de tipo *NP-Completo* y una de las mejores formas de tratarlos es por medio de Metaheurísticas (Blum & Roli, 2003). Las metaheurísticas buscan generar soluciones cercanas a la solución exacta que se puedan tener en tiempos más cortos.

Hacer uso de estas herramientas en problemas de la vida real es una muestra de la flexibilidad y potencial que tienen estas técnicas. En la editorial de (Torres-Jiménez & Pavón, 2014) se muestran diferentes casos en los cuales una metaheurística ha sido llevada a un problema de la vida real como en (Pessoa Ferreira Lima, 2013) que se proponen métodos para organizar y determinar la cantidad de clusters en procesos de clasificación y que posteriormente se ha usado la misma técnica con sensores de gas que han logrado clasificar gases como el propano, butano, etano, metano y otros derivados del petróleo. Los anteriores escritos invitan a proponer soluciones a problemas relacionados con la generación de conocimiento de los diferentes especímenes de la familia de los coronaviridae y de poner a prueba el potencial

tanto de las herramientas computacionales como las metaheurísticas con algoritmos que ayuden a la optimización de soluciones, que tan necesarias resultan en los tiempos actuales.

En un corto periodo de tiempo luego del anuncio de la pandemia, los contagiados por esta enfermedad se han contado por millones y los fallecidos en decenas de miles. Cada esfuerzo que se pueda realizar puede incrementar el conocimiento sobre el virus al identificar características que den entendimiento a su forma de interactuar con el huésped, similitudes entre las diferentes mutaciones y el entendimiento de la enfermedad en pro de encontrar formas de prevención y tratamiento.

2. Justificación

La pandemia que se ha venido desarrollando en el marco del padecimiento y contagio de la enfermedad conocida como COVID-19, proveniente del microrganismo no vivo SARS-CoV-2 miembro de la familia de los coronaviridae, genera la oportunidad y necesidad de estudiar por medio de herramientas de biología computacional esta familia de microrganismos para aportar información y profundizar el conocimiento que se tiene.

Para los meses de mayo y junio del año 2020, se reportan en las bases de datos del CDD (*Conserved Domains Database (CDD) and Resources*, 2020) perteneciente al NCBI (*National Center of Biotechnology Information*) 198 genomas completos de los microrganismos de la familia de los coronaviridae. A pesar de ser a la fecha 207 diferentes integrantes de la familia coronaviridae en la misma base, se han registrado tan solo 68 de ellos con su información proteómica completamente curada (*RefSeq: NCBI Reference Sequence Database*, 2018). Lo anterior indica que se tiene menos de un 35% de información proteómica completamente curada sobre estos virus.

Con el modelo para la clusterización de proteínas que se propone en este documento se realizarán agrupaciones basadas en una agrupación multiobjetivo de las proteínas inter-especie en función de los mejores aciertos bidireccionales (BBH) dados por su información proteómica.

Proponer el modelo para la clusterización podrá ofrecer una agrupación para incrementar el conocimiento sobre esta familia de microrganismos y los grupos de proteínas que los componen, abriendo la puerta a futuras búsquedas de la interacción que puedan tener con proteínas del huésped, buscando ayudar a conocer aún más la estructura y las posibles mutaciones en la evolución de la enfermedad y las formas en que podría migrar entre especies como lo ha hecho en las últimas décadas (Letko et al., 2020).

La implementación de algoritmos metaheurísticos con enfoque multiobjetivo permite que la clusterización tenga diversos parámetros para la generación de los grupos de proteínas de los coronaviridae. Esta situación se puede considerar como un problema NP-Completo en el cual se requiere hacer análisis de tipo combinatorio, lo que propone una carga computacional alta. Lo anterior no excluye y al contrario se abre la puerta a utilizar el modelo para evaluar organismos de otras filogenias.

3. Objetivos.

3.1. Objetivo General:

Modelar una herramienta de Biología Computacional basada en algoritmos metaheurísticos multiobjetivo, que permita realizar la clusterización de proteínas de los microrganismos de la familia de los coronaviridae basándose en su información proteómica.

3.2. Objetivos Específicos:

- Identificar los algoritmos metaheurísticos apropiados para el desarrollo del modelo.
- Definir los objetivos del problema de optimización multiobjetivo.
- Realizar experimentos para la validación de los parámetros del modelo de clusterización.

Capitulo 1: Marco Teórico.

En este capítulo se exponen los diferentes conceptos teóricos y tecnológicos con los cuales se soporta la investigación haciendo un recorrido por temáticas como la optimización, las metaheurísticas, conceptos biológicos y la metodología para análisis de datos ómicos, y la aplicación en la computación.

1. Optimización.

La optimización según se ejemplifica en (Lange, 2013) es una de las ramas de la matemática con mayor edad que actualmente propone innumerables aplicaciones de tipo científico e ingenieril. Algunas situaciones o problemas que han sido optimizados a través de los tiempos son: El Problema de Heron, La Ley de Snell y otros más de tipo univariado, es decir de una sola variable a optimizar donde el cálculo diferencial y la geometría se han desarrollado en buena manera. A lo anterior se debe agregar que muchas situaciones y problemas del mundo real, pueden tener más de una variable a optimizar y por eso la optimización se debe considerar también de tipo multiobjetivo. Se entenderá por objetivo a la magnitud que se proponga optimizar.

1.1. Optimización Mono objetivo.

El campo de estudio de la optimización abarca diferentes áreas del conocimiento, desde la matemática, la investigación de operaciones, llegando a la biología que ha permitido e inspirado procesos de optimización, existen quienes consideran la optimización como un proceso propio de la evolución como se puede llegar a considerar basado en (Kuri, 2000) donde se conecta la obra de Charles Darwin (El origen de las especies) con la optimización en función de la supervivencia del individuo más apto, y se puede leer la inspiración que han generado diferentes especies en diferentes inventos.

Importante anotar que los problemas de optimización de tipo mono objetivo son aquellos en los que se requiere o considera un único valor a optimizar (Mirjalili, 1920).

En la Figura 1 cuya construcción se basa en (Fernández González, 2019) se muestra la generalización de un problema de optimización mono objetivo, con la que se procura minimizar (o maximizar) la función escalar

$$f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}, \Omega \neq \emptyset$$

$$\begin{array}{lll} \min & f(x_1 ... x_m) & m = 1, ...M \\ \text{s.t.} & g_j(x) \leq 0 & j = 1, ...J \\ & h_k(x) = 0 & k = 1, ...K \\ & x_i^L \leq x_i \leq x_i^U & i = 1, ...N \\ & x \in \Omega & \end{array}$$

Figura 1. Definición de problema de optimización mono objetivo. Fuente: (Fernández González, 2019)

A la anterior figura le corresponde como objetivo minimizar la función estando limitada por las desigualdades de g_j y la presencia de h_k igualdades y los límites que se expresen en x_i^L y x_i^U inferiores y superiores respectivamente.

1.2. Optimización Multiobjetivo.

De la misma forma en que se contempla la optimización para una sola función objetivo es necesario poner en el radar la optimización para múltiples variables donde se pretende mejorar el desempeño de cada función objetivo. Estas variables pueden estar en conflicto o son inversas al momento de optimizar una la(s) otra(s) pueden reducir su desempeño. Lograr balancear las variables tratando de mantener el mejor desempeño en todas las variables resulta ser el objetivo de optimización de múltiples variables.

En los problemas de optimización se busca obtener vectores de soluciones que logren satisfacer las funciones objetivo manteniendo el cumplimiento de las restricciones propias de la naturaleza del problema. Se muestra a continuación la forma de representar un problema de optimización basada en (Blank & Deb, 2020)

$$\begin{array}{lll} \min & f_m(x) & m = 1,..,M \\ \text{s.t.} & g_j(x) \leq 0 & j = 1,..,J \\ & h_k(x) = 0 & k = 1,..,K \\ & x_i^L \leq x_i \leq x_i^U & i = 1,..,N \\ & x \in \Omega & \end{array}$$

Figura 2. Definición de problema de optimización multiobjetivo. Fuente: (Blank & Deb, 2020)

Para la anterior definición el autor cita de (Deb, 2011) que el ideal es lograr un conjunto de soluciones (un vector de variables) que cumpla las diferentes restricciones y desigualdades, donde dicho problema (Figura 2) tendrá que x_i representa la variable a optimizar, con x_i^L y x_i^U como limites inferior y superior descritos para $\mathbf{N}$ variables, $\mathbf{M}$ funciones objetivo, sujeto a $\mathbf{J}$ desigualdades y $\mathbf{K}$ restricciones de igualdad.

La literatura respecto a la optimización multiobjetivo es bastante amplia y permite de igual forma dar una luz a los lectores sobre los usos y algoritmos que se han empleado en la solución de diferentes problemas, por ejemplo el problema de la planeación urbana es considerado como un multiobjetivo (J. P. Wang & Tong, 2009). Existen también propuestas de marcos de trabajo o frameworks para abarcar problemas de forma general (Blank & Deb, 2021), es decir el mismo método pueda ser aplicado con los menores ajustes a diferentes problemas.

2. Heurísticas y Metaheurísticas.

La aplicación de Heurísticas y Metaheurísticas resulta ser una opción importante de considerar al momento de abordar y proponer soluciones a problemas de tipo combinatorio. Aun mas cuando estos problemas requieren tener una solución especifica en un tiempo considerable y que permita ser tenida en cuenta como "optima".

La palabra Heurística según propone (Duarte Muñoz, 2007) es original del griego *heuriskein* que se traduce en descubrir o encontrar. También resulta muy valido apropiar la definición de la Real Academia Española (Española, 2014) que propone:

"En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos como por tanteo, reglas empíricas, etc. ".

Una de las limitaciones que se presentan en las heurísticas se generan al momento de encontrar soluciones locales que se logren considerar óptimas y no poder salir de estas en búsqueda de otras soluciones explorando otras posibilidades. Es este el punto en el cual se acuña por parte de Fred Glover (Glover, 1986) el termino Metaheurísticas, con el que se propone integrar aleatorización controlada, estrategias de aprendizaje, descomposición inducida y búsqueda tabú, enlazar a los métodos de optimización de problemas combinatorios cada una de las técnicas propuestas de forma que se logren explorar soluciones que van más allá de las óptimas locales.

Dentro de las metaheurísticas se encuentran algoritmos que al hibridar entre ellos se permite ser considerados como solución aproximada de algún problema de interés científico.

2.1. Algoritmos Heurísticos.

Se enuncian y describen brevemente algunos de los algoritmos heurísticos basados en la clasificación que propone (Duarte Muñoz, 2007).

- Métodos Constructivos: Buscan la construcción de una solución donde el resultado está altamente determinado por el algoritmo usado.
    - Algoritmos Voraces: Son algoritmos que en cada iteración agregan un elemento nuevo a la solución sin embargo, solo generan soluciones locales basados en sus elecciones cercanas sin tener en cuenta las soluciones del futuro.
    - Algoritmos de Descomposición: Proponen dividir el problema hasta el punto en que se tenga una solución del problema más pequeño o simple y recursivamente combinar las soluciones para así regresar hasta su parte más grande. En algunos casos se pueden tener de forma exacta como aproximada para las soluciones que se presentan.

  - Algoritmos de Reducción: Se deben localizar los atributos que debe tener una solución aceptable del problema que se está afrontando y se asume que otras soluciones también tendrán estos atributos. Con lo anterior se propone reducir el espacio de búsqueda.
  - Algoritmos de Manipulación: Se genera una simplificación del modelo del problema para obtener una solución lo más simple posible. Se extrapola este modelo con el fin de tener una solución aproximada. Se deben considerar técnicas de agrupación de variables, adición de restricciones y hasta linearización.
- Métodos de Búsqueda: Son los que tienen en su base de conocimiento o experiencia una solución factible y pretenden proponer una mejora a esta.
  - Búsqueda local (first improvement): Inicia la búsqueda en una solución factible que mejora progresivamente examinando la vecindad y tomando el primer elemento que causa una mejora a la solución actual.
  - Búsqueda local (best improvement): Inicia la búsqueda en una solución factible que mejora progresivamente al examinar la vecindad y tomando el mejor elemento de la vecindad para la solución.
  - Búsqueda Aleatoria: Se parte de una solución factible asociada a una vecindad asociada y se toman las soluciones de manera aleatoria.

Los algoritmos heurísticos presentan limitaciones que se deben tener en cuenta pues dependiendo del problema se debe evaluar la profundidad de su implementación. La dificultad de escapar de las soluciones óptimas locales es la más marcada de las limitaciones y esto infiere que no se logran soluciones de mayor optimada. Estos algoritmos no tienen funciones que les permitan explorar en otras vecindades soluciones que sean tenidas en cuenta.

El uso de heurísticas para realizar procesos de optimización es un reflejo de la capacidad científica y la integración de diferentes ramas del conocimiento que buscan aumentar las estrategias para la resolución de problemas tanto clásicos como específicos (Michel Fernández González et al., 2018)

2.2. Algoritmos Metaheurísticos.

Algunos autores proponen que las metaheurísticas son procesos iterativos que guían y modifican las heurísticas con la finalidad de generar soluciones de alta calidad. En (Duarte Muñoz, 2007, fig. 4.1) se consolidan las metaheurísticas en la combinación de ideas de diferentes áreas del conocimiento, técnicas para el diseño de algoritmos, algoritmos específicos, fuentes de inspiración y métodos estadísticos.

Las siguientes figuras muestran distintas metaheurísticas y una propuesta de clasificación para estas.

Paradigma	*Implementación*	*Inspiración*	*Multi-arranque*	*Búsqueda local*	*Solución inicial*	*Función objetivo*	*Niveles de vecindad*	*Vecindad*	*Memoria*	*Proceso aleatorios*	*Procesos adaptativos*
Optimización por colonias de hormigas - ACO	aco tradicional aco + demonios	si	no	no si	poblac.	estática dinámica	uno	estática	explícita	no	si
Equipos asíncronos- AT	AT	parcial	no	si	poblac.	estática	varios	estática	explícita	no	si
Algoritmos Culturales - CA	CA	si	no	no	poblac.	estática	varios	dinámica	explícita	si	si
Algoritmos de estimación de la distribución - EDA	EDA	parcial	no	no	poblac.	estática	varios	*	implícita	no	si
Búsqueda por entorno adaptativo borroso - FANS	FANS	no	no	si	trayec.	dinámica	varios	estática	no	no	si
Algoritmos Genéticos - GA	GA	si	no	no	poblac.	estática	varios	estática	implícita	si	si
Proc. aleatorizados y adaptativos de búsqueda voraz - GRASP	GRASP tradicion GRASP biased	no	si	si	trayect.	estática	varios	dinámica	no	si no no	no si si
Búsqueda local guiada - GLS	GLS	no	no	si	trayect.	dinámica	uno	estática	explícita	no	no
Concentración heurística - HC	HC	no	si	si	trayect.	estática	varios	estática	explícita	no	no
Búsqueda local iterativa - ILS	ILS tradicional ILS reactivo	no	no	si	trayect.	estática	varios	estática	no	si	no si
Algoritmos Meméticos - MA	MA	parcial	no	si	poblac.	estática	varios	estática	implícita	si	si
Métodos Multi-arranque - MSM	MSM basico AMS	no	si	si	trayect.	estática	uno	estática	no	si no	no si
Métodos ruidosos - NM	NM- ruido datos NM ruido f NM perturbación	no	no	si	trayect.	dinámica dinámica estática	uno	estática	no	si	no no no
Metah. de opt. parcial en condiciones. esp. de intensificacion - POPMUSIC	POPMUSIC	no	no	si	trayect.	estática	uno	estática	no	no	no
Reencadenamiento de trayectorias - PR	PR	no	*	si	poblac.	estática	uno	estática	implícita	no	si
Recocido simulado - SA	SA	si	no	si	trayect.	estatica	uno	estática	no	si	no
Inteligencia de enjambre - SI	SI	si	no	no	poblac.	dinámica	uno	dinámica	implícita	si	si
Búsqueda dispersa - SS	SS	no	no	si	poblac.	estática	varios	estática	explícita	no	si
Métodos de aceptación del umbral - TAM	TA	no	no	si	trayect.	estática	uno	estática	no	no	no
Búsqueda tabú - TS	TS tradicional TS aleatorio TS reactivo	no	no	si	trayect.	estática	uno	dinámica	explícita	no si no	no no si
Búsqueda en vecindad variable - VNS	VND RVNS BVNS	no	no	si no si	trayect.	estática	varios	estática	no	no si si	no

Figura 3. Listado de Metaheurísticas. Fuente: (Duarte Muñoz, 2007).

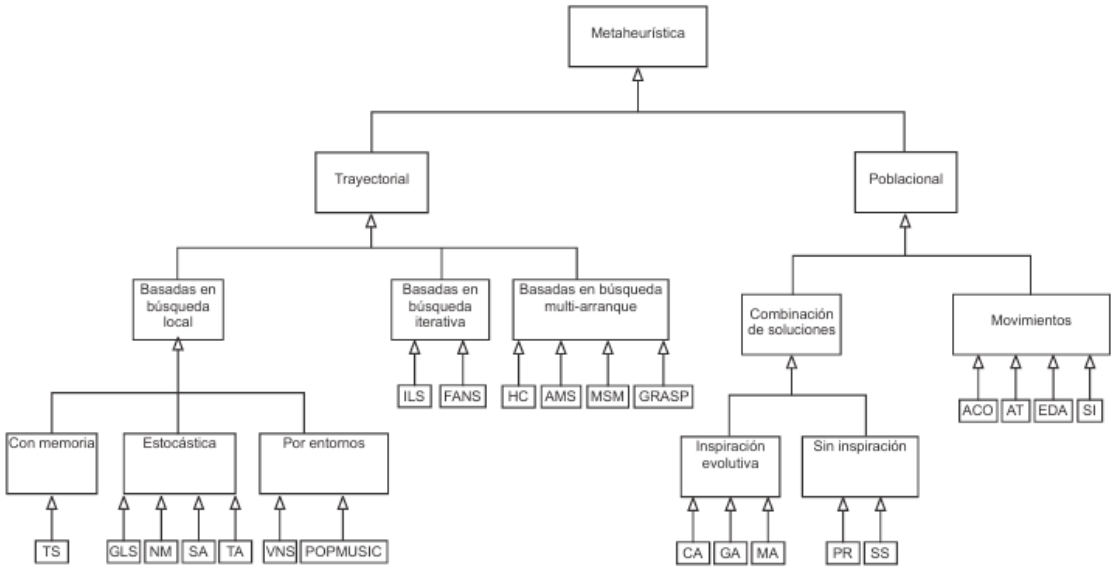


Figura 4. Clasificación de Metaheurísticas. Fuente: (Duarte Muñoz, 2007).

En (Rincón Miranda, 2016) se citan y dan a conocer a modo de lista las características o propiedades de las técnicas metaheurísticas:

- Es desconocido el resultado final y la calidad de este resultado.
- Es necesario establecer formas de finalizar la ejecución por cantidad de iteraciones o algún valor definido.
- No se garantiza la obtención de un resultado optimo, se espera un resultado confiable en tiempos razonables.
- Con la intensión de explorar nuevos espacios de solución permiten evaluar soluciones erradas.
- Se consideran generales por la capacidad que tienen de aplicarse en la resolución de casi todo tipo de problemas de optimización combinatoria.

En la literatura es posible encontrar una amplia gama de metodologías, marcos de trabajo y tutoriales que ofrecen realizar formulaciones de problemas reales de optimización y lograr soluciones eficientes con la implementación de algoritmos metaheurísticos (Osaba et al., 2021). De igual modo también se mantiene el estudio de problemas clásicos que se presentan en el mundo real como el enrutamiento de vehículos que ha tenido cerca de 300 publicaciones

en los últimos años en las que se muestran soluciones a las diferentes variantes de este problema incursionando en el campo de las metaheurísticas (Elshaer & Awad, 2020).

3. Aminoácidos y Proteínas.

Esta sección tiene como propósito realizar una introducción a los conceptos biológicos y metodológicos referentes al desarrollo del proyecto. La importancia de estos conceptos resulta valiosa toda vez que mostrará la forma de relacionar los conceptos y la metodología del análisis de datos ómicos. Para poder llegar al entendimiento de estos conceptos se presentarán algunas definiciones rescatadas de la literatura.

3.1. Biología Molecular.

Ya en los años de la década de 1970 se podían contar quizás un par de décadas en las que se consideraba el nacimiento de una rama de la biología que se proponía la búsqueda del fenómeno de la vida a nivel molecular y así mismo se definía como: "bioquímica mirando al gen" (Allende, 1972). Es posible que el interés por la biología molecular y el entendimiento de la forma en que se sintetizan las proteínas naciera de diversos sucesos y experimentos como el que sirviera para explicar la estructura del ADN en la famosa doble hélice (Watson & Crick, 1953) ayudados por los trabajos de cristalografía de Rosalind Franklin (Osman Elkin, 2003).

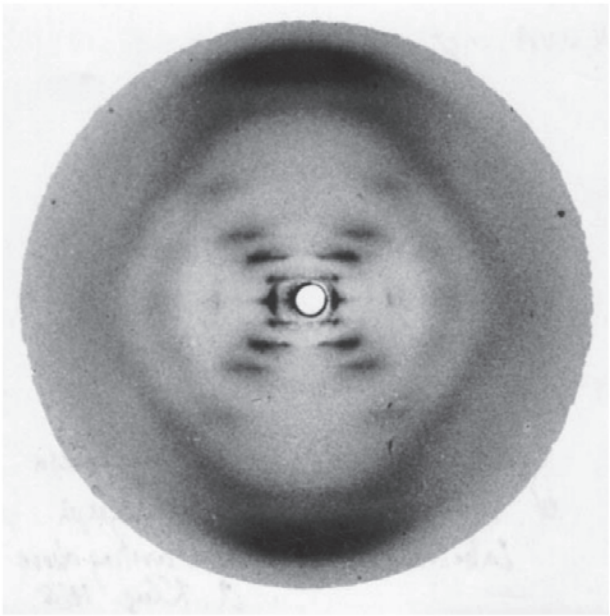


Figura 5. Fotografía 51. Difracción de rayos X para identificar la estructura del ADN. Fuente: (Osman Elkin, 2003).

Posteriormente se logró entender la existencia de un proceso de herencia en el que se transmitía un mensaje genético desde el ADN al ARN y la síntesis de proteínas, a esto se le ha

nombrado por diversos autores como el "dogma central" de la biología molecular (Roldán Martínez, 2015).

Figura 6. Descripción del dogma central de la Biología Molecular. Fuente: (Roldán Martínez, 2015).

En la descripción simple que se ve en la Figura 6 se muestra un esquema de la generación de proteínas desde una cadena de ADN que se transcribe en una cadena de ARN y que de allí se toman las respectivas bases que se agrupan y luego se traducen para formar las proteínas indicadas en el código genético. Esta es una forma de simplificar el proceso pues se pretende enunciarlo como elemento importante de la biología molecular, pero no resulta ser del alcance del proyecto.

3.2. Proteínas.

Las proteínas se pueden considerar como sustancias de animales y vegetales es decir en cualquier tejido biológico, las cuales se hacen necesarias para las células de dichos tejidos (Gómez-Moreno Calerra, 2000). En la literatura es posible encontrar una agrupación de las proteínas según la función que cumplen y según la estructura que tenga la proteína para generar relaciones con otras proteínas.

Algunos autores indican que el estudio de las proteínas puede comprender esfuerzos mayores que el estudio o trabajo con ADN; estudiar una agrupación de proteínas de un organismo se le conoce como PROTEÓMICA y propone hacer análisis de la estructura y las funciones de las proteínas en dicho organismo, pues conocer estas características de la proteína pueden ayudar a entender el comportamiento del organismo en términos físicos y químicos (Roldán Martínez, 2015).

Se ha encontrado que las proteínas se constituyen en todos los casos por Nitrógeno, Hidrogeno, Carbono y Oxigeno en la mayoría de los casos como indica (Hernández, 2009) se puede encontrar Azufre, Fosforo, Hierro, Zinc y otros elementos en menor medida. Los anteriores elementos se agrupan en bloques a los que se nombra y conoce como Aminoácidos con los cuales se forman sus unidades estructurales y según redacta el autor "La integración de estos bloques se les conoce como Estructuras Poliméricas. Estas estructuras se identifican

por contener un grupo carboxilo (-COOH) y uno amino ($-NH_2$) y un lateral con un radical (R) que se considera como el diferenciador con otras proteínas.

3.3. Aminoácidos.

Los aminoácidos se consideran la estructura base de las proteínas al ser tenidas en cuenta como polímeros lineales resultantes de la condensación de los aminoácidos (Gómez-Moreno Calerra, 2000) los cuales tienen propiedades físicas y químicas. Los aminoácidos cuentan con un grupo carboxilo libre y en su átomo de carbono un amino libre.

Actualmente los aminoácidos cuentan con un alfabeto de 20 aminoácidos (como se muestra en la Tabla 1) que se combinan para construir las diferentes proteínas de forma tal que las propiedades fisicoquímicas de los aminoácidos son de bastante relevancia para conocer las propiedades biológicas de las proteínas.

Aminoácido	Abreviatura
Glicina	Gly (G)
Alanina	Ala (A)
Valina	Val (V)
Leucina	Leu (L)
Isoleucina	Ile (I)
Metionina	Met (M)
Prolina	Pro (P)
Fenilalanina	Phe (F)
Tirosina	Tyr (Y)
Triptófano	Trp (W)
Serina	Ser (S)
Treonina	Thr (T)
Cisteína	Cys (C)
Asparagina	Asp (N)
Glutamina	Gln (Q)
Lisina	Lys (K)

Histidina	His (H)
Arginina	Arg (R)
Aspartato	Asp (D)
Glutamato	Glu (E)

Tabla 1. Alfabeto de Aminoácidos. Fuente: Construcción basada en (Gómez-Moreno Calerra, 2000)

La estructura general de los aminoácidos consta de un átomo de carbono (C), al cual se enlazan cuatro elementos, un grupo amino (-NH_2), un grupo carboxilo (-COOH), un átomo de hidrogeno (H) y una cadena lateral del aminoácido, la cual se representa con R pero dependiendo del aminoácido será una de las cadenas de la Tabla 1 y se puede representar como se muestra en la siguiente figura.

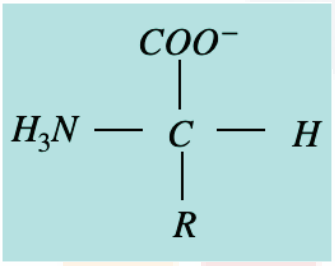


Figura 7. Estructura general de los aminoácidos. Fuente: Construcción basada en (Gómez-Moreno Calerra, 2000)

La estructura covalente de los aminoácidos se puede ver en la Figura 8 que muestra el equivalente de la cadena lateral R en cada uno de los casos de la Tabla 1. El color amarillo claro es para los aromáticos. Los de color verde representan los polares de carga neutra. El color azul para la carga positiva. El color rojo para la carga negativa.

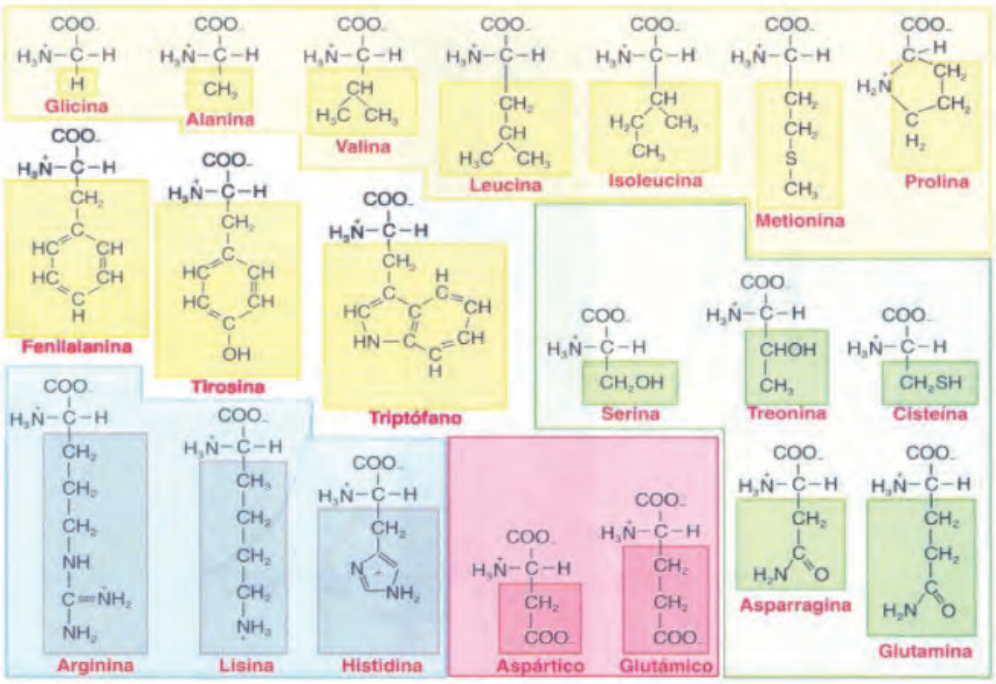


Figura 8. Estructura de los 20 aminoácidos. Fuente: (Gómez-Moreno Calerra, 2000)

3.4. Retos en el área de las proteínas.

El estudio de las proteínas ha permitido avances referentes a la salud por medio del reconocimiento de patrones y generando correcciones en las cadenas de proteínas y sin duda la importancia que han tomado algoritmos como BLAST (*Basic Local Alignment Search Tool*) propuesto en (Altschul et al., 1990) para los procesos de alineación que desde los años 90 han venido en incremento y sin embargo, como propone David Baker en (TED Talk, 2019) ya se hace necesario la construcción y diseño de nuevas proteínas, la expansión del alfabeto de aminoácidos con lo que se puede iniciar un nuevo camino a solucionar problemas desde las proteínas como construir un mayor conocimiento de estas áreas.
Para solventar dichas situaciones se requerirá poder conocer completamente las estructuras de las proteínas dado que su constante movimiento dificulta conocer su estructural tridimensional exacta. Algunas técnicas han tenido acercamientos como se ve en (Osguthorpe, 2000) donde se indican los elementos a tener en cuenta para idear la estructura de una proteína, su carga, su condición geométrica y la energía en la superficie. Al igual que algunas de las metodologías de la época para intentar conocer la estructura de plegado o "*Folding*".
Otros de los retos que se pueden tener en cuenta para buscar una aproximación a través de metaheurísticas podrían ser de manera computacional el "*folding*" e interacción entre diferentes proteínas (Adiyaman & McGuffin, 2019; Feig, 2017) ya que por medio de la Cristalografía en las proteínas y la Microscopia aún se hace difícil obtener una modelación y conocimiento 3D de las estructuras de las proteínas.

3.5. Metodología de análisis de datos Ómicos.

Lo primero en esta sección es declarar que un dato de tipo ómico es la unidad en la que se generaliza un genoma, un proteoma o un transcriptoma, este concepto permitirá abordar en las siguientes secciones otros conceptos propios de los análisis computacionales a través de la computación usando las herramientas y metodologías propuestas.

Ahora bien, los vertiginosos avances de la ciencia y la tecnología han causado impactos en la forma y estilo de vida de las personas propone (Oppenheimer, 2018) cuando introduce los incrementos en la longevidad y calidad de vida de las personas gracias a los procesos de automatización que caminan en las diferentes industrias.

Las técnicas y metodologías que se han generado para los procesos de secuenciación de ADN y proteínas han tenido grandes mejoras, introduce (Rincón Miranda, 2016) que son elementos que han generado un llamado a nuevos algoritmos para ordenar, analizar, comparar y almacenar grandes cantidades de secuencias donde un ejemplo impulsor de esta revolución es propuesto por (Altschul et al., 1990) el cual en la actualidad sigue siendo de amplio uso. Esto es uno de los llamados a la aparición de una nueva área de estudios: La bioinformática. En (Rincón Miranda, 2016) al citar a (Altamiranda et al., 2008) propone que la computación es una pieza importante por su robustez y eficiencia. Es aquí un campo para proponer y aplicar heurísticas y metaheurísticas que puedan generar nuevas soluciones aproximadas y óptimas con un grado de confiabilidad satisfactorio para los diferentes problemas que puedan ser requeridos sin dejar de lado que no bastará con las técnicas convencionales, será necesario realizar diseños e hibridaciones de las existentes (Rothlauf, 2011).
Un ejemplo de la biología computacional es la organización por medio de arboles filogenéticos que se ve en (H. Wang et al., 2009) en el que se propone organizar 82 hongos basados en su filogenia usando técnicas computacionales.

3.5.1. Descripción de la Metodología para análisis de datos ómicos.

En este proyecto se tendrá en cuenta la visión de la investigación en la aproximación de distancias para los árboles filogenéticos que se enuncia en (E. Ponce-de-Leon-Senti et al., 2017) que compara y propone diferentes formas de medir las distancias entre los especímenes y los expresa en arboles filogenéticos. Esta metodología de trabajo es de interés por la flexibilidad que provee en su implementación. Los conceptos claves de esta metodología como la forma en la que se miden las distancias, la definición de los mejores aciertos (en inglés Best Hit BHT), los mejores aciertos bidireccionales (Best Bidirectional Hit BBH) se mostrarán en las siguientes secciones. Esta metodología en principio se propone para construir arboles filogenéticos, luego se ha considerado aplicarla para la obtención de cliques de proteínas que se basan en el árbol filogenético previamente elaborado (Ponce de León Sentí et al., 2022). Ha esta metodología se le ha propuesto aplicar un proceso de automatización que sea previo a sus análisis que se describe en los capítulos siguientes de este trabajo; esa propuesta a la metodología ha sido reportada con la ejemplificación de un árbol filogenético elaborado usando la metodología y la aportación en (Galvis-Motoa et al., 2021). Esta adición a la metodología permite reducir errores al momento de la obtención de los datos de entrada, la

forma de renombrar los archivos y la automatización de la construcción y ejecución de las sentencias BLAST.

3.5.2. BLAST.

En anteriores secciones se ha hecho referencia o nombrado a BLAST (*Basic Local Alignment Search Tool*) que se traduce como Herramienta de Búsqueda Local de Alineamientos, la cual fue introducida en (Altschul et al., 1990) y es ampliamente usada y reconocida (Pérez Castillo et al., 2014). Se emplea con la finalidad de encontrar alineamientos de secuencias no conocidas en diferentes secuencias conocidas al contrastarlas entre si. Es posible hacer de uso de BLAST desde Internet, basta con visitar el sitio web "https://blast.ncbi.nlm.nih.gov/Blast.cgi" (*BLAST: Basic Local Alignment Search Tool*, 2016) una vez en el sitio web se puede tener acceso a las diferentes variaciones de BLAST.

- Blastp: Permite realizar comparaciones de secuencias de proteínas con otras secuencias de proteínas.
- Blastn: Permite realizar comparaciones de secuencias de nucleótidos con otras secuencias de nucleótidos.
- Blastx: Permite comparar secuencias de nucleótidos contra secuencias de proteínas.
- Tblastn: Permite realizar comparaciones de secuencias de proteínas contra secuencias de nucleótidos.

Las anteriores son algunas de las posibles comparaciones que se derivan de BLAST, y se pueden ejecutar en línea, sin la necesidad de instalaciones en la máquina de trabajo, pero esto debe dar a entender que los recursos se verán limitados a la cantidad de otros usuarios con los que se esté compartiendo anónimamente el uso de los servidores del NCBI. También al hacer uso del recurso por medio de Internet, se cuenta con la posibilidad de realizar comparaciones con infinidad de secuencias, lo cual sino se tiene contemplado como necesario puede llegar a generar procesamientos innecesarios al comparar la secuencia con otras que no son de interés y en algunos casos por ser recursos compartidos se tienen limitaciones en los volúmenes de datos que se pueden trabajar. Para poder tener un mayor control sobre la ejecución de BLAST (sea cualquiera de las formas de sus variaciones) es posible realizar una instalación en la maquina local (o un servidor propio) descargando el software (*Download BLAST Software and Databases Documentation*, 2008), esto aportará la ventaja de que se pueda trabajar con

volúmenes de datos mayores que los permitidos en la versión en línea y las velocidades de procesamiento serán tales como las que aporte la máquina que ejecuta (Camacho et al., 2019). En el capítulo de 2 se presentará la forma en la que se implementó de manera local una instalación de BLAST, el uso y la forma en que se automatizo el proceso con la herramienta.

Para usar BLAST además resulta importante el contar con los archivos de las secuencias que se requieren evaluar, cuando se hace en la versión WEB, solo se requiere la secuencia con la que se va a contrastar, y esta puede proporcionarse por medio de un archivo o en forma de una cadena de texto que representa bien sea las bases de los genes o los aminoácidos de las proteínas. Posiblemente la forma para mantener más organizado y limpio los procesos sea hacer uso de un archivo de texto.

Para poder hacer uso de las diferentes formas de BLAST instalado de manera local, es necesario contar inicialmente con algunos ficheros de entrada que permitirán construir la base de datos para los procedimientos de BLAST(Gertz, 2005; Madden, 2008). Estos procedimientos se explican con mayor detalle en el capítulo 2.

Los ficheros requeridos han de contener la secuencia biológica representada por caracteres de texto. Estos archivos o ficheros tienen un formato nombrado como FASTA se caracterizan estos tipos de archivos por incluir una cabecera inicial que se identifica por tener el carácter '>' precediendo el nombre de la secuencia y una descripción e identificación de esta (Roldán Martínez, 2015).

En algunas ocasiones se pueden encontrar ficheros con más de una secuencia y cada una de estas secuencias deberá iniciar con el identificador precedido con '>'.

Un ejemplo de estos formatos de archivo se puede encontrar en la Figura 16

3.5.3. BHT y BBH.

Uno de los métodos que se emplea en gran cantidad de las fuentes consultadas y que tendrá mucho renombre en el presente trabajo, se conoce con las siglas BBH (bidirectional best hit) reportado en (Overbeek et al., 1999) y ajustado para ser usado con proteínas como se muestra en (Ponce de León Sentí et al., 2015). Este método consiste en hallar los alineamientos que ofrezcan menos cantidad de cambios, es decir los que tengan mayor similitud entre las cadenas de proteínas de cada organismo. Se hace necesario entender un alineamiento como los cambios que se apliquen a una cadena o secuencia de aminoácidos que representan una proteína, para transformarse en otra secuencia de aminoácidos de una proteína. El proceso se conoce homologación de cadenas y consiste en realizar los ajustes insertando espacios o

cambiando caracteres para que se hagan lo más semejantes. La Figura 9 es un ejemplo de estas representaciones de secuencias.

Secuencia 1	G	L	A	M	S	C	P	K
	G	+	+	M	S	C	+	K
Secuencia 2	G	V	I	M	S	C	N	K

Figura 9. Ejemplo de homologación de secuencias. Fuente: Propia.

Como se ha indicado se pretende encontrar los alineamientos que tengan la menor cantidad de variación, mayor similitud entre cadenas se tendrá a menor cantidad de cambios donde se entenderán como idénticas las proteínas que sus cambios sean iguales a cero (0).

Cuando se mencionan los BBH (que en español se traducen como mejores aciertos bidireccionales) se requiere contemplar que una secuencia tiene homología desde el organismo con la secuencia 1 hacia el organismo con la secuencia 2 y luego desde la secuencia 2 hacia la secuencia 1. Para hacer una representación que genere más claridad se propone la Figura 10.

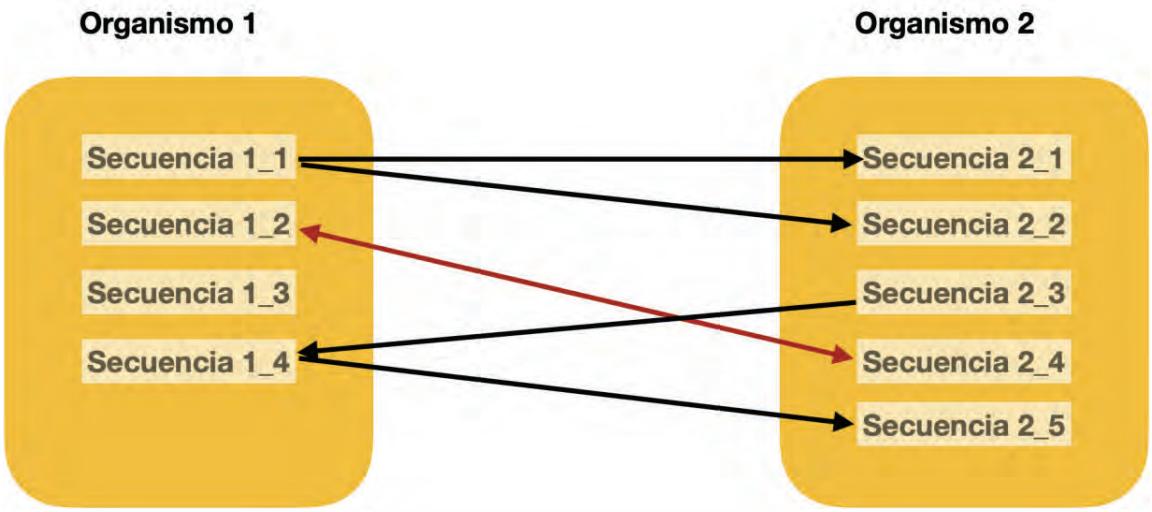


Figura 10. Representación BHT y de un BBH. Fuente: Propia.

Para el ejemplo en la Figura 10 se propone entender a los Organismos 1 y 2 bien sea como proteomas o genomas, de forma que las secuencias que los integran sean proteínas o genes según sea más cómoda la interpretación.

Se ve que la Secuencia 1_1 tiene hits con la Secuencia 2_1 y Secuencia 2_2 y estos aciertos tienen un sentido, es decir, se toma como base la secuencia en el Organismo 1 y el destino el Organismo 2. Ejemplo similar de estos hits se ven entre las Secuencias 2_3 con Secuencia 1_4

y desde esta hacia Secuencia 2_5. Con una flecha roja se ha señalado un BBH que se ha encontrado entre la Secuencia 1_2 y Secuencia 2_4. También se puede entender que la homología es bidireccional entre las secuencias indicadas. La existencia de estas similitudes entre secuencias puede ayudar a la interpretación de una relación evolutiva entre los organismos.

El conteo de BBH entre organismos propondrá una estimación en la similaridad entre los organismos estudiados (diferencias entre organismo a organismo) esta forma de medición se ha trabajado y documentado en diferentes proyectos como (Ponce de León Sentí et al., 2013; E. Ponce-de-Leon-Senti et al., 2017) con la siguiente ecuación:

$$\delta\left(G^i, G^j\right) = 1 - \frac{2\left|B_{i,j}\right|}{n_i + n_j}$$

Ecuación 1. Medición de la similaridad.

De la que $\left|B_{i,j}\right|$ será la totalidad de BBH entre los proteomas o genomas G^i y G^j. Teniendo que n_i y n_j son la cantidad de genes o proteínas de G^i y G^j respectivamente. Se aplica esta definición para poder estandarizar la medida dadas las diferencias de cantidad de genomas o proteomas del organismo.

En el 4. Alistamiento de Datos., se expondrán los resultados de la aplicación de este concepto y el paso a paso para lograr llegar a este procesamiento.

4. SARS-CoV-2

Esta es una enfermedad proveniente de un virus que para muchos puede resultar nueva, pero su antecesor el SARS-CoV, ya ha realizado apariciones en otros tiempos y ha causado decesos de una gran cantidad de personas por problemas de tipo respiratorio (considerada epidemia por la OMS) mostrando la necesidad de una vacuna para defendernos de los virus de la familia de los coronaviridae (Enjuanes Sánchez et al., 2011).

4.1. COVID-19.

En el año 2019 se reportó la aparición de la enfermedad COVID-19 por sus siglas en inglés (*COronaVIrus Disease*) la cual es generada por el virus SARS-CoV-2 que son las siglas para (*Severe Acute Respiratory Syndrome COronaVirus 2*). Enfermedad que ha desatado la necesidad de la declaración de una emergencia pandémica en todo el mundo a partir de marzo de 2020 (Xu & Li, 2020).

Esta enfermedad resulta ser parte de la familia de los coronavirus como el (*Middle East Respiratory Syndrome*) conocido como MERS. Esta asociada con síntomas de tipo respiratorio al igual que la mayor cantidad de coronaviridaes (Pascual-Iglesias et al., 2021).

A la fecha se encuentra en el repositorio del NCBI el CDD (*Conserved Domains Database (CDD) and Resources*, 2020) cerca de 210 miembros de la familia coronaviridae. Microrganismos no vivos que se componen de una alineación de proteínas de estructura simple (9 a 15 proteínas). Como se indica en el reporte de (Rabi et al., 2020) los virus del tipo coronaviridae han estado presentes en las emergencias médicas de las últimas décadas prendiendo las alarmas para unir esfuerzos en consolidar el mayor conocimiento y entendimiento de estos microrganismos dadas sus mutaciones a través de diferentes hospederos. Por ejemplo los murciélagos, los dromedarios y más recientemente los pangolines que han permitido que estos virus entren en la especie humana causando las difíciles situaciones de salud mundial.

Como se explica en (Rabi et al., 2020) este es un virus del orden de los nidovirales, los cuales se conocen porque al realizar su replicación en el citoplasma lo hacen por nidos o conjuntos de material genético. También se caracteriza el virus por tener un tamaño medio y considerarse como un virus "pesado" (*(87) COVID 19 CORONAVIRUS FISIOPATOLOGIA Parte 1 - YouTube*, 2020) el cual tiene una especie de espinas "spikes" que se encargan de romper o cortar la parte externa de la célula del hospedero y así permitir el ingreso del virus y su replicación. Como se ve en la Figura 11.

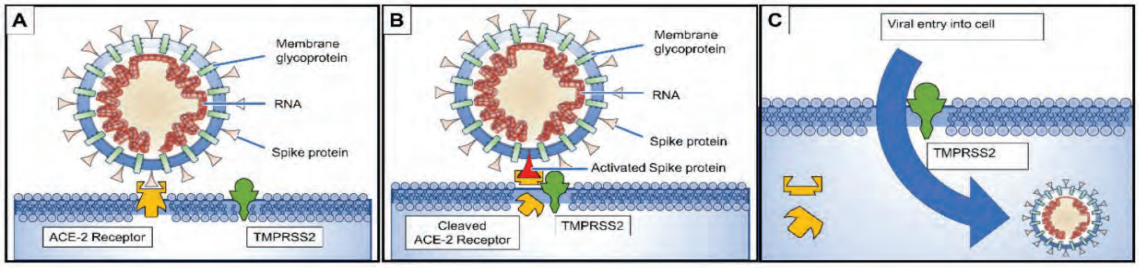


Figura 11. Proceso de penetración del virus. Fuente: (Rabi et al., 2020)

Las relaciones filogenéticas que presentan los especímenes coronaviridae pueden ser útiles para realizar entendimientos de sus proteínas y similitudes brindando conocimientos por sus relaciones que podrían considerarse desde un punto algorítmico como combinatorio. La importancia de conocer sobre estos especímenes se propone para poder buscar similitudes en los especímenes y las enfermedades con las cuales pone en dificultades a la raza humana Figura 12.

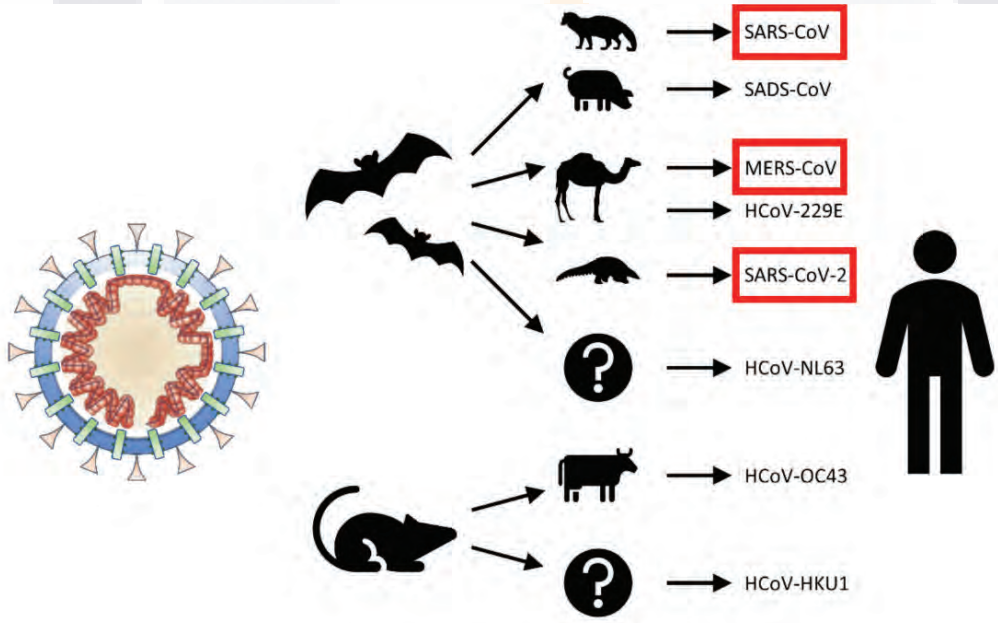


Figura 12. Orígenes animales de los coronavirus humanos. Fuente: (Rabi et al., 2020)

La importancia y necesidad de entender y describir los coronaviridaes y los virus que estos producen vienen creciendo desde posiblemente antes de la actual pandemia y ha permitido conocer que su hospedero natural son los murciélagos (Enjuanes Sánchez et al., 2011). Ahora bien, el previo conocimiento que se ha venido construyendo sobre el SARS-CoV (Nieto-Torres et al., 2014, 2015) ha permitido un rápido desarrollo de múltiples opciones de vacuna que ayuden a pronto superar esta enfermedad (Pascual-Iglesias et al., 2021).

Capitulo 2: Preparación de los Datos.

En este capítulo se presenta al lector el proceso para el alistamiento de los datos de las proteínas exponiendo el desarrollo de una herramienta para la automatización de la obtención y pre-procesamiento de los archivos que contienen los proteomas.

1. Origen de los datos.

El volumen de datos de tipo biológico que se genera diariamente derivado de procesos de secuenciación es descomunal y crece con el pasar del tiempo como se muestra en (Pelta, 2002). Estos datos conllevan a retos desde la computación como son su almacenamiento y distribución pero esto debe tomarse como oportunidades para implementar nuevos algoritmos y desarrollar herramientas que faciliten el trabajar con estos datos con el propósito de lograr mejores entendimientos de los fenómenos e interacciones biológicas como se enuncia en (Barreto Hernández, 2008).

El NCBI (por sus siglas en inglés: *National Center for Biotechnology Information*) se ha encargado de proporcionar herramientas de computación para el almacenamiento y análisis de información bioquímica, genética y de biología molecular a través de bases de datos y software desarrollados por grupos de investigación en medicina, biotecnología, computación y otras disciplinas (*Our Mission - NCBI*, 1988) para mejorar la salud y controlar la enfermedad. Los esfuerzos y propuestas para el cumplimiento de la misión que tiene el NCBI son constantes tratando de simplificar las labores de los investigadores proponiendo herramientas que permitan depurar los datos que en los diferentes recursos se almacenan, por ejemplo el NCBI BioSystems database que se propone por Lewis en (Geer et al., 2010) en el que se centralizan elementos de diferentes repositorios de información de NCBI, como lo son las publicaciones, anotaciones proteínas, genes, taxonomías y diversa información que pueda existir enlazándola en un solo espacio.

1.1 Consultar los recursos.

El NCBI tiene a disposición del público diferentes bases de datos que son usadas dependiendo de los objetivos de investigación y análisis que se tengan propuestos, para enunciar algunos de ellos se puede acceder a:

- PubMed[1], es el recurso para la búsqueda de literatura y material publicado dentro del contexto de la salud, las enfermedades, la biología molecular y la bioinformática.
- Genome[2], es el recurso que organiza la información en genomas incluyendo secuencias, mapas, cromosomas, ensambles y anotaciones.
- Protein[3], es la colección de secuencias de proteínas de muchas fuentes.

Las anteriores y otros más están disponibles en el sitio web respectivo https://www.ncbi.nlm.nih.gov/.

Para el desarrollo del actual proyecto se ha consultado el recurso de Genome, en el que se realiza una búsqueda por Organismo(Figura 13), pues esto muestra la información de los organismos que cuentan con información secuenciada y proporciona una lista muy detallada.

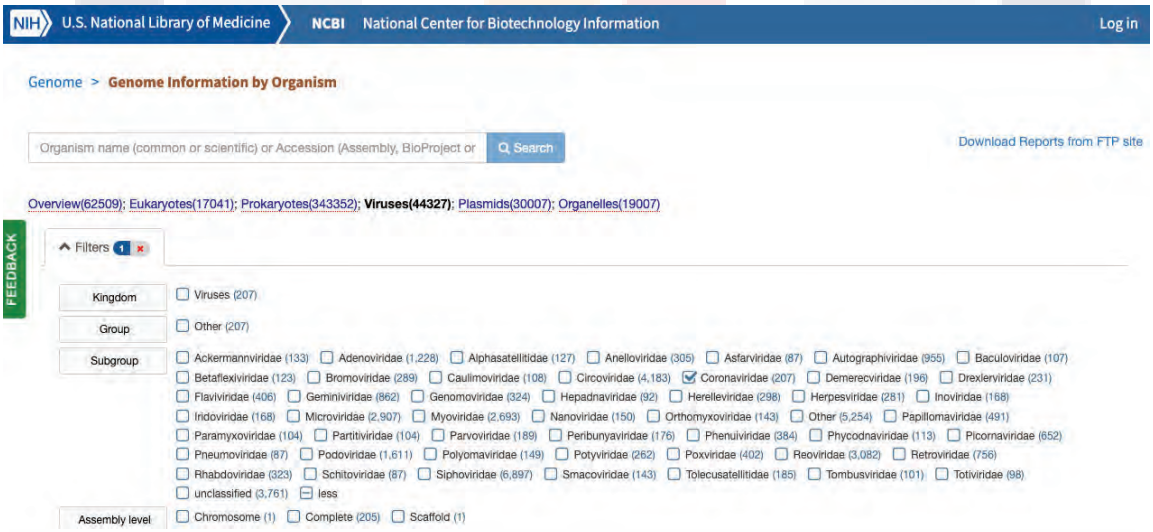


*Figura 13. Captura de Pantalla del Recurso Genome. Fuente: (*Genome List - Genome - NCBI, *n.d.-b)*

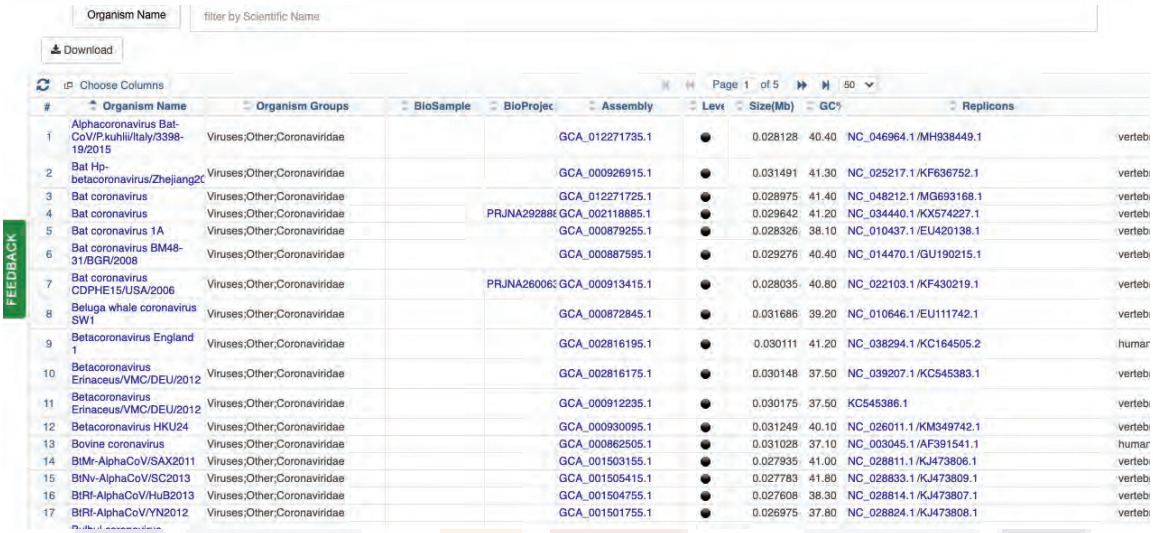
Como se observa en la Figura 13 se ha realizado un filtro en el campo de Reino, seleccionando los virus y en el subgrupo se ha seleccionado Coronaviridae, lo cual muestra un total de 207

[1] https://pubmed.ncbi.nlm.nih.gov/
[2] https://www.ncbi.nlm.nih.gov/genome/
[3] https://www.ncbi.nlm.nih.gov/protein/

coincidencias. Los resultados de aplicar estos criterios de búsqueda se pueden observar en la parte inferior del sitio web los cuales se visualizan como en la Figura 14.

Del resultado de la búsqueda que se percibe en la Figura 14 es importante remarcar un elemento muy valioso, a través del botón "Download" se da la posibilidad de descargar un archivo con la información generada, lo que abre la posibilidad de realizar otros procesos.

El archivo descargado en formato CSV (Comma-Separeted-Values) se muestra en la sección de anexos.



#	Organism Name	Organism Groups	BioSample	BioProject	Assembly	Level	Size(Mb)	GC%	Replicons	
1	Alphacoronavirus Bat-CoV/P.kuhlii/Italy/3398-19/2015	Viruses;Other;Coronaviridae			GCA_012271735.1		0.028128	40.40	NC_046964.1 /MH938449.1	
2	Bat Hp-betacoronavirus/Zhejiang2(	Viruses;Other;Coronaviridae			GCA_000926915.1		0.031491	41.30	NC_025217.1 /KF636752.1	
3	Bat coronavirus	Viruses;Other;Coronaviridae			GCA_012271725.1		0.028975	41.40	NC_048212.1 /MG693168.1	
4	Bat coronavirus	Viruses;Other;Coronaviridae		PRJNA29288(	GCA_002118885.1		0.029642	41.20	NC_034440.1 /KX574227.1	
5	Bat coronavirus 1A	Viruses;Other;Coronaviridae			GCA_000879255.1		0.028326	38.10	NC_010437.1 /EU420138.1	
6	Bat coronavirus BM48-31/BGR/2008	Viruses;Other;Coronaviridae			GCA_000887595.1		0.029276	40.40	NC_014470.1 /GU190215.1	
7	Bat coronavirus CDPHE15/USA/2006	Viruses;Other;Coronaviridae		PRJNA26006(	GCA_000913415.1		0.028035	40.80	NC_022103.1 /KF430219.1	
8	Beluga whale coronavirus SW1	Viruses;Other;Coronaviridae			GCA_000872845.1		0.031686	39.20	NC_010646.1 /EU111742.1	
9	Betacoronavirus England 1	Viruses;Other;Coronaviridae			GCA_002816195.1		0.030111	41.20	NC_038294.1 /KC164505.2	
10	Betacoronavirus Erinaceus/VMC/DEU/2012	Viruses;Other;Coronaviridae			GCA_002816175.1		0.030148	37.50	NC_039207.1 /KC545383.1	
11	Betacoronavirus Erinaceus/VMC/DEU/2012	Viruses;Other;Coronaviridae			GCA_000912235.1		0.030175	37.50	KC545386.1	
12	Betacoronavirus HKU24	Viruses;Other;Coronaviridae			GCA_000930095.1		0.031249	40.10	NC_026011.1 /KM349742.1	
13	Bovine coronavirus	Viruses;Other;Coronaviridae			GCA_000862505.1		0.031028	37.10	NC_003045.1 /AF391541.1	
14	BtMr-AlphaCoV/SAX2011	Viruses;Other;Coronaviridae			GCA_001503155.1		0.027935	41.00	NC_028811.1 /KJ473806.1	
15	BtNv-AlphaCoV/SC2013	Viruses;Other;Coronaviridae			GCA_001505415.1		0.027783	41.80	NC_028833.1 /KJ473809.1	
16	BtRf-AlphaCoV/HuB2013	Viruses;Other;Coronaviridae			GCA_001504755.1		0.027608	38.30	NC_028814.1 /KJ473807.1	
17	BtRf-AlphaCoV/YN2012	Viruses;Other;Coronaviridae			GCA_001501755.1		0.026975	37.80	NC_028824.1 /KJ473808.1	

*Figura 14. Captura de Pantalla, Resultado de la búsqueda. Fuente: (*Genome List - Genome - NCBI, *n.d.-b)*

En el archivo descargado se encuentran los datos referentes a 207 organismos coronaviridae y contiene las siguientes columnas (Tabla 2) para los organismos:

Columna	Descripción
Organism Name	Nombre del Organismo
Organism Groups	Agrupación del Organismo.
BioSample	Identificador del ejemplar en la base de datos BioSample.
BioProject	Identificador de la colección en el proyecto.
Assembly	Nombre del Ensamblaje.
Level	Nivel de Ensamblaje.

Size(Mb)	Tamaño del ensamblaje.
Host	Hospedero
CDS	Genes Codificadores de Proteínas.
Release Date	Fecha de Lanzamiento
GenBank FTP	Dirección del FTP en GenBank
RefSeq FTP	Dirección del FTP en RefSeq

*Tabla 2. Campos disponibles para descargar. Fuente: Propia basado en (*Genome List - Genome - NCBI, *n.d.-b)*

1.2 Adquisición de los datos.

Luego de obtener el archivo en formato CSV con el listado de los organismos y algunas de sus características este contiene dos columnas que apuntan a servidores donde se almacenan sus datos proteomicos. Estos archivos son conocidos como fasta. Los dos servidores a los que se puede tener acceso son recursos del NCBI que se agrupan en Protein.

El primero es conocido como GenBank (*GenBank Overview*, 2013) en el cual participan investigadores de diversos centros de investigación y proporcionan múltiples secuenciaciones. El segundo recurso es conocido como RefSeq (*RefSeq: NCBI Reference Sequence Database*, 2018) donde se consolidan secuencias genomicas y proteomicas de los organismos. Los registros de proteinas que se encuentran en este recurso son generados de la revisión computacional y curación manual de los que se encuentran reportados en GenBank.

En el presente proyecto se ha determinado utilizar los datos contenidos dentro del recurso RefSeq para aprovechar el beneficio de estar curados y clasificados a través de algoritmos computacionales y revisados por investigadores.

Se debe acceder a la dirección de FTP del organismo con el que se desea trabajar, para el caso se usa la columna RefSeq del archivo CSV.

Al acceder al FTP respectivo se debe observan diferentes archivos que contienen información genómica y proteómica (aunque esto depende de que tan completo o estudiado este el organismo). Los archivos que se requieren obtener son conocidos como fasta y se encuentran comprimidos en formato GZ dentro del FTP (Figura 15).

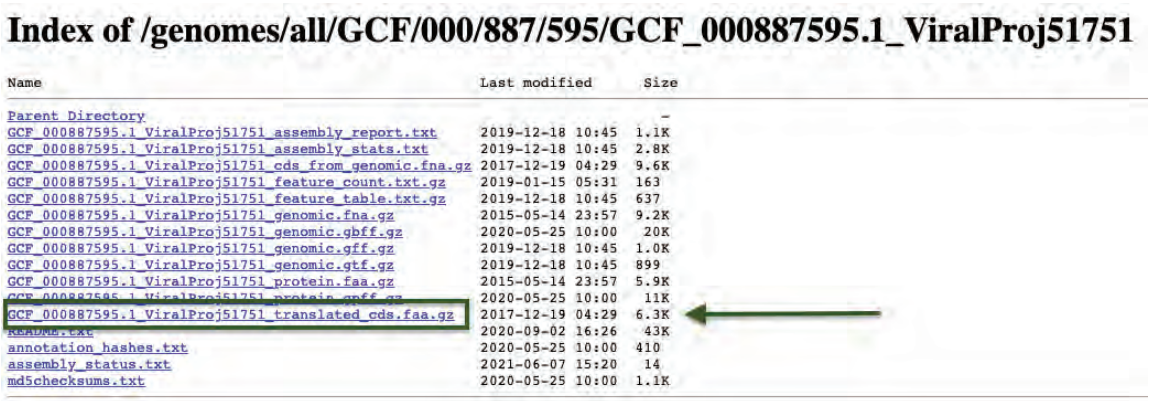


Figura 15. Captura de Pantalla, Ejemplo de FTP. Fuente: FTP RefSeq.

Es importante recordar que este proceso se debe hacer con cada uno de los registros que se encuentren en el mencionado archivo CSV que con anterioridad se obtuvo del filtro mostrado en la Figura 14.

1.3 Organización de los datos.

Luego de completar la descarga de todos los archivos comprimidos en el formato GZ con extensión (.faa.gz) tras navegar por las direcciones FTP contenidas en los registros del archivo CSV generado de los pasos indicados y ejemplificados en la Figura 13 y Figura 14. La primera parte del alistamiento consiste en realizar la descompresión de cada uno de los archivos descargados que se reconocen por la extensión igual que la del FTP (ver Figura 15).

El siguiente paso en el alistamiento es el proceso de renombrar los archivos descomprimidos para que sea sencilla la identificación de cada virus en futuros pasos. Este cambio de nombre es requerido ya que los archivos vienen nombrados con nombres de la secuencia y otros identificadores diferentes al nombre del espécimen (en este caso virus). En la Figura 16 se observa el archivo "GCF_000848685.1_ViralProj14739_protein.faa" que contiene la información proteómica del "Porcine epidemic diarrhea virus". Este archivo pasará a ser nombrado: *Porcine_epidemic_diarrhea_virus.faa*.

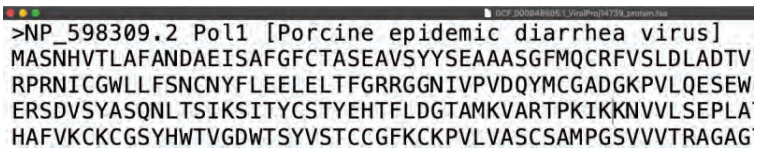


Figura 16. Captura de un archivo Fasta. Fuente: Propia.

Una vez todos los archivos se encuentren con los nombres de los virus que corresponden, se hace necesario realizar la construcción de las sentencias de BLAST (Altschul et al., 1990) y su respectiva ejecución. Una actividad necesaria es realizar una instalación de la herramienta BLAST (*Download BLAST Software and Databases Documentation*, 2008) para que se realicen las ejecuciones de forma local. Es importante resaltar que existen opciones de ejecución a través de internet sin requerir instalaciones, sin embargo, esta opción puede verse limitada por los extendidos tiempos de procesamiento que se puedan presentar, por eso se hace de manera local.

La labor de la ejecución de BLAST se enmarca en dos pasos sin incluir la instalación de BLAST:

1.3.1 Creación de las Sentencias.

Este es el proceso en el cual se construyen las sentencias que se ingresarán en el programa BLAST basadas en los archivos fasta. Se requiere crear dos diferentes tipos de sentencias basadas en los comandos BLAST como se muestran a continuación.

- Makebalstdb: Es el comando que construye una base de datos personalizada basada en los archivos fasta de las secuencias. Se debe ejecutar para cada una de las secuencias con las que se va a trabajar. Para su funcionamiento requiere ser invocado como mínimo con los argumentos de archivo de entrada(-in), tipo de base de datos(-dbtype), archivo de salida(-out).

  ```
  makeblastdb -in <fasta file> -dbtype <prot> -parse_seqids -out <output file>
  ```

  Un ejemplo del uso de la anterior sentencia sería como se muestra a continuación:

  ```
  makeblastdb -in Porcine_epidemic_diarrhea_virus.faa -dbtype prot -parse_seqids -out Porcine_epidemic_diarrhea_virus
  ```

- Blastp: Con este comando se puede realizar la comparación de una secuencia con una base de datos. Se debe ejecutar realizando la comparación de cada secuencia contra cada uno de los elementos de la base de datos creada con el comando "makeblastdb". Para su funcionamiento se deben invocar los argumentos referentes a la base de datos (-db), la secuencia a comparar (-query), el archivo de salida (-out) se debe concatenar

con la siguiente sentencia por medio del carácter '&' y luego invertir los valores del (-db), el (-query) y el sentido del (-out).

```
blastp -db <db file> -query <fasta file> -out <output file> & blastp -db <db file> -query <fasta file> -out <output file>
```

Una muestra de uso del anterior comando sería:

```
blastp -db file1 -query file2.faa -out file2_vs_file1.txt & blastp -db file2 -query file1.faa -out file1_vs_file2.txt
```

Es de aclarar que se concatenaran sentencias como secuencias se tengan.

Para cada uno de los comandos expuestos se crea un archivo de tipo script que contenga la totalidad de las sentencias requeridas. En el caso de makeblastdb se construye una sentencia por cada archivo fasta, mientras en el caso de blastp se construyen $\mathbf{n-1}$ bloques de sentencias concatenadas, donde $\mathbf{n}$ es igual a las secuencias a trabajar.
Dependiendo del sistema operativo en el cual se este trabajando, se deben poner las extensiones para los scripts, para Windows se debe usar *.bat* y para Unix (Linux o MacOS) *.sh*.

1.3.2 Ejecución de las Sentencias.

Es el segundo paso en el proceso de obtener los alineamientos resultantes de BLAST. El orden para realizar la ejecución es primero el archivo que contiene el comando makeblastdb en el mismo directorio en el cual se encuentran los archivos fasta(que previamente se descomprimieron y se renombraron). Cuando la ejecución del script termine, se debe ejecutar el archivo que contiene el comando blastp también dentro del directorio que contiene los archivos fasta y los generados por el primer script.
Una vez terminadas estas ejecuciones en el directorio es una buena práctica mover los archivos fasta en un directorio aparte para los posteriores procesos que puedan ser necesarios ya que no se requieren para las ejecuciones de los aciertos.
Los pasos enunciados en esta sección se han descrito para su realización de forma manual, lo que abre la posibilidad de errores humanos involuntarios que provoquen retrasos en los procesamientos, que excluyan datos y resten confiabilidad a los resultados. Por lo cual se debe tener bastante rigurosidad y tiempo para realizarlos y revisarlos.

2. Automatización del proceso.

En la sección inmediatamente anterior se ha explicado el proceso manual de obtención de los datos basada en el archivo CSV que contiene los datos de las direcciones FTP y su alistamiento para el posterior procesamiento. En esta sección se detalla una propuesta para automatizar los procesos de la sección anterior, con la que se pretende reducir las posibilidades de cometer algún error en el proceso que por ser repetitivo está abierto a que estos errores sucedan.
Esta automatización se desarrolló en lenguaje de programación Python por ser un lenguaje de propósito general y buscando lograr futuras integraciones con herramientas existentes y la creación de otras en el futuro del proyecto, al igual que su fácil ajuste y uso con diferentes organismos y sus respectivas secuencias.

El inicio del proceso de forma automatizado inicia de la misma forma que la obtención manual como se indica en las secciones 1.1 Consultar los recursos. y 1.2 Adquisición de los datos. con los cuales se pueden identificar y enlistar las secuencias de los organismos que se proponen analizar y se consolidan en el archivo .CSV que contendrá los datos que se muestran en la Tabla 2. El archivo mencionado será ingresado en la herramienta para el iniciar el proceso automatizado.

Para el caso de los coronaviridae se ha indicado que se logran por medio del buscador de NCBI encontrar 207 registros que se descargan en el archivo CSV.
Es con la información que se obtenga de este archivo que se van a realizar las diferentes tareas del presente trabajo.

Los pasos que se deben realizar en forma automatizada son los siguientes:

1. Recibir el archivo de extensión CSV.
2. Navegar en las direcciones FTP contenidas.
3. Descargar los archivos respectivos de cada una de las direcciones FTP.
4. Descomprimir los archivos descargados.
5. Renombrar los archivos. (Ver Figura 16)
6. Crear los archivos de tipo script con las sentencias *makeblastdb* y *blastp.*
7. Realizar la ejecución de los archivos script.
8. Separar los archivos fasta de los archivos de alineamiento.

Los pasos descritos se han condensado en la Figura 17 para facilitar el entendimiento del proceso y lo que sucede en el interior de la herramienta de automatización de la obtención de los datos.

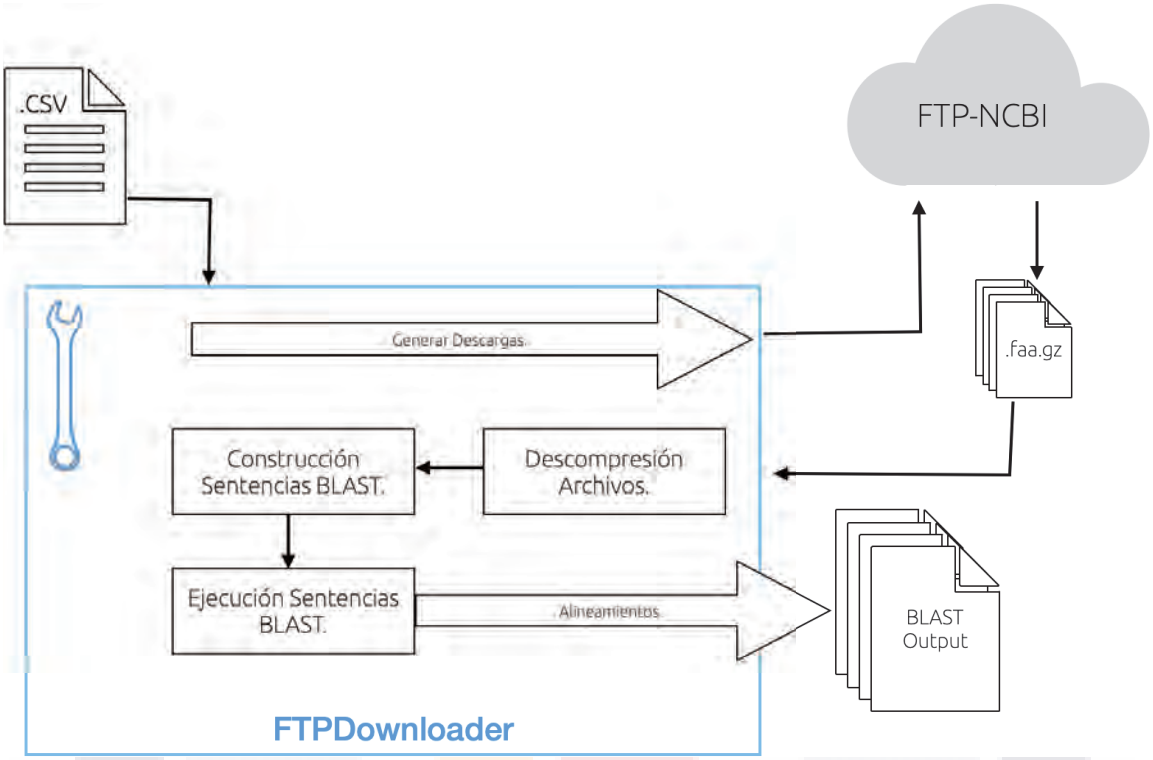


Figura 17. Descripción del Proceso de descarga. Fuente: Propia.

Esta herramienta es una forma de automatizar los primeros dos (2) pasos de la metodología propuesta para la obtención de cliques que se presenta en (E. E. Ponce-de-Leon-Senti et al., 2020) y en la medición de distancias para la construcción de los árboles filogenéticos que se presenta en (E. Ponce-de-Leon-Senti et al., 2017) La integración de la herramienta con las metodologías indicadas se ha publicado en el artículo arbitrado (Galvis-Motoa et al., 2021). De igual forma se propone la herramienta para ser utilizada con otros organismos diferentes a los virus. Durante el desarrollo de este proyecto, la herramienta se ha llevado a una versión Beta en modalidad WEB a través de una tesina de grado titulada: "Integración de los módulos de pre procesamiento en la metodología basada de mejores aciertos bidireccionales: Software UAA-PROT" con autoría del ahora Ingeniero en Computación Inteligente por la Universidad Autónoma de Aguascalientes, José Ramsés Moreno dirigida por Sergio Iván Galvis Motoa y co-dirigida por la Dra. Eunice Esther Ponce de León Sentí.

En los anexos se podrán consultar los enlaces a los repositorios en los que se almacenan los desarrollos generados en este documento.

3. Datos Obtenidos.

Luego de la ejecución de la herramienta que automatiza la obtención de los datos de los organismos a estudiar basados en los registros del archivo CSV, se puede consolidar la siguiente información en la Tabla 3.

Entrada	
viruses.csv	205 registros de virus de la familia coronaviridae.
Salida	
Archivos comprimidos.	68 archivos con extensión (.faa.gz)
Archivos Fasta.	68 archivos con extensión (.faa)
Scripts.	2 archivos generados para ejecutar comandos BLAST.
Base de Datos Local BLAST.	9 archivos generados por cada archivo fasta ingresado al comando makeblastdb. 612 archivos con las extensiones. (.pdb, .phr, .pin, .pog, .pos, .pot, .psq, .ptf, .pto)
Ejecución BLAST	4556 archivos generados de la sentencia blastp de la comparación de las secuencias de proteínas con la base de datos local.

Tabla 3. Entrada y Salidas de la obtención de datos automatizada. Fuente: Propia.

Necesariamente se debe hacer una aclaración sobre la cantidad de secuencias descargadas comparadas con la cantidad de registros que contiene el archivo CSV. En un primer momento se esperaría contar con 205 archivos fasta descargados como propone el listado, sin embargo, se ha indicado que se decidió hacer uso del recurso RefSeq (*RefSeq: NCBI Reference Sequence Database*, 2018) el cual contiene los datos que han sido validados y curados provenientes del recurso GenBank (*GenBank Overview*, 2013). Esta característica es la que ha dado razón a la selección del recurso, dando a entender que revisados y curados se pueden encontrar las secuencias de 68 virus. Los virus que se han obtenido por medio de la herramienta para el desarrollo se consignan en la Tabla 4.

Nombre del virus.	
1	Betacoronavirus_Erinaceus_VMC_DEU_2012_v1
2	Miniopterus_bat_coronavirus_HKU8
3	SARS_coronavirus_Tor2
4	Infectious_bronchitis_virus
5	Thrush_coronavirus_HKU12_600
6	Feline_infectious_peritonitis_virus
7	Duck_coronavirus
8	Camel_alphacoronavirus
9	Munia_coronavirus_HKU13_3514
10	Bat_coronavirus_CDPHE15_USA_2006
11	Bat_coronavirus_1A
12	Human_coronavirus_229E
13	Wencheng_Sm_shrew_coronavirus_v1
14	Betacoronavirus_Erinaceus_VMC_DEU_2012
15	BtRf_AlphaCoV_YN2012
16	Alphacoronavirus_Bat_CoV_P_kuhlii_Italy_3398_19_2015
17	Murine_hepatitis_virus_v1
18	Porcine_epidemic_diarrhea_virus
19	Scotophilus_bat_coronavirus_512
20	Human_coronavirus_OC43
21	Transmissible_gastroenteritis_virus
22	Infectious_bronchitis_virus_v1
23	Pipistrellus_bat_coronavirus_HKU5
24	Rousettus_bat_coronavirus_HKU9
25	Betacoronavirus_England_1
26	Bulbul_coronavirus_HKU11_934
27	Rat_coronavirus_Parker
28	Porcine_coronavirus_HKU15_v1
29	Severe_acute_respiratory_syndrome_coronavirus_2
30	Wigeon_coronavirus_HKU20
31	Rousettus_bat_coronavirus_HKU10

32	BtNv_AlphaCoV_SC2013
33	Canada_goose_coronavirus
34	BtMr_AlphaCoV_SAX2011
35	Rousettus_bat_coronavirus
36	Middle_East_respiratory_syndrome_related_coronavirus
37	BtRf_AlphaCoV_HuB2013
38	Lucheng_Rn_rat_coronavirus
39	Wencheng_Sm_shrew_coronavirus
40	Magpie_robin_coronavirus_HKU18
41	NL63_related_bat_coronavirus_v1
42	Bat_Hp_betacoronavirus_Zhejiang2013
43	Bat_coronavirus_BM48_31_BGR_2008
44	Bovine_coronavirus
45	Swine_enteric_coronavirus
46	Human_coronavirus_HKU1
47	Beluga_whale_coronavirus_SW1
48	Night_heron_coronavirus_HKU19
49	Rabbit_coronavirus_HKU14
50	Bat_coronavirus
51	Porcine_coronavirus_HKU15
52	Rhinolophus_bat_coronavirus_HKU2
53	Tylonycteris_bat_coronavirus_HKU4
54	Human_coronavirus_NL63
55	Ferret_coronavirus
56	Bat_coronavirus_v1
57	Shrew_coronavirus
58	Mink_coronavirus_strain_WD1127
59	White_eye_coronavirus_HKU16
60	Rodent_coronavirus
61	Sparrow_coronavirus_HKU17
62	Common_moorhen_coronavirus_HKU21
63	Betacoronavirus_HKU24
64	Turkey_coronavirus

65	Coronavirus_AcCoV_JC34
66	unidentified_human_coronavirus
67	Murine_hepatitis_virus
68	NL63_related_bat_coronavirus

Tabla 4. Listado de Virus Obtenidos. Fuente: Propia.

Con las secuencias de proteínas de los virus relacionados en la Tabla 4 se realizaron las acciones (automatizadas) indicadas en la Figura 17 Tabla 4y se obtuvieron las cantidades de archivos que se indican en la Tabla 3.

3.1 Observaciones.

1. Los resultados del proceso automatizado son los equivalentes al proceso de forma manual.
2. Se ha desarrollado la herramienta con el ideal de lograr reducir tiempos y errores en futuros proyectos de investigación ayudando a las investigaciones a enfocar sus recursos en otras labores.
3. Se realizo una prueba de la herramienta con un archivo CSV con otros especímenes y los resultados se pueden considerar igual de exitosos a los generados con los virus de este proyecto.
4. En el proceso de renombrar los archivos fasta, se hallaron especímenes con el mismo nombre, lo cual se resolvió agregando la sigla “v1” en el nombre de alguno de los archivos.

4. Alistamiento de Datos.

En el desarrollo de este apartado se muestra la generación de los BBH (descritos en la sección 0 del Marco Teórico) como metodología de alistamiento de los datos obtenidos en la primera sección del Capitulo 2: Preparación de los Datos., con el ideal de hallar la similitud entre las proteínas de los virus antes de realizar los procesamientos que permitan alcanzar los objetivos del proyecto y poder agruparlos en los clústeres.

4.1. Obtención de los Mejores Aciertos.

Los mejores aciertos son conocidos en inglés como "Best Hits" y algunos trabajos les asignan las siglas "BHT". Estos se pueden entender con la ayuda de la Figura 9 en la que se ejemplifica lo que es un alineamiento. Los alineamientos que tengan menor cantidad de cambios serán los que se enlisten como "BHT" o mejores aciertos. La Figura 10 se compone de diferentes aciertos, que logran ser etiquetados como "BHT" y uno de tipo "BBH" por ser un alineamiento que se tiene entre el Organismo 1 y el Organismo 2.

Para lograr la obtención de los BHT se han ejecutado los algoritmos programados para la metodología propuesta en (E. E. Ponce-de-Leon-Senti et al., 2020) y que han sido probados también en (Ponce de León Sentí et al., 2015) con resultados muy prometedores.

El primer algoritmo en ser ejecutado es el encargado de extraer los Mejores Aciertos (BHT) desde cada uno de los 4556 archivos BLAST (Tabla 3) generados en el Capitulo 2: Preparación de los Datos. referentes a los virus de la Tabla 4 que se derivan de la ejecución de las sentencias "blastp".

Esta labor se enfoca en recuperar las mejores coincidencias que se generaron de la ejecución de las sentencias por cada pareja de virus.

El algoritmo realiza la apertura de cada archivo y lo examina para extraer las proteínas del virus A y su posible homología con las proteínas del virus B. Estos valores los almacena en un nuevo archivo que sintetiza la información.

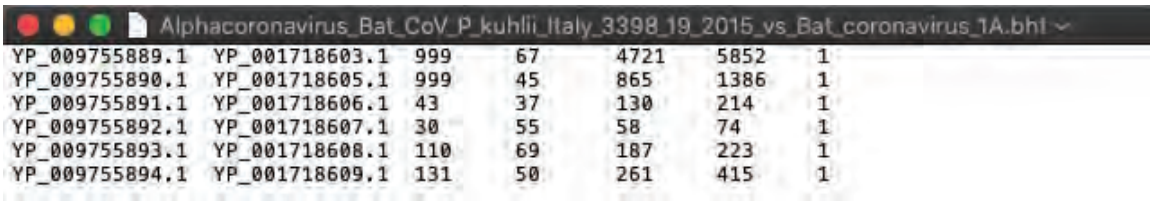


Figura 18. Captura de Archivo BHT. Fuente: Propia.

La estructura que tienen los archivos BHT es como se observa en la Figura 18, que muestra los mejores aciertos desde el virus "Alphacoronavirus_Bat_CoV_P_kuhlii_Italy_3398_19_2015" contrastados con el virus "Bat_coronavirus_1A". La primera columna se compone de las proteínas del primer virus, la segunda es la proteína con la cual a tenido una similitud, la tercera representa un valor de similitud, el cual para efectos prácticos entre mayor sea es más alta la similitud, esta también es conocida como el "e-value" o "expected value". La cuarta columna indica el porcentaje de hits positivos, la quinta la cantidad de estos hits y la sexta son el total de posibles hits que se podían alcanzar.

4.2. Obtención de los Mejores Aciertos Bidireccionales.

En la sección anterior se generaron un total de 4556 archivos con extensión (.BHT) resultado de la ejecución de los alineamientos originados por las sentencias "blastp" para cada uno de los proteomas comparado con todos los demás proteomas del estudio (en total 68).
Los mejores aciertos bidireccionales permiten generar una medida de similaridad entre las proteínas de los organismos como se muestra en (E. E. Ponce-de-Leon-Senti et al., 2020).
Para obtener los mejores aciertos bidireccionales (Bidirectional Best Hits) se realiza por medio de un algoritmo que consulta en los 4556 archivos de mejores aciertos (Best Hits) y compara la bidireccionalidad de los aciertos, es decir, la coincidencia desde el organismo 1 al organismo 2 y del organismo 2 al 1 como se ejemplifica en la Figura 10 con la flecha de color rojo. El resultado de este proceso se consolida en 68 archivos de salida que tienen la extensión (.BBH).

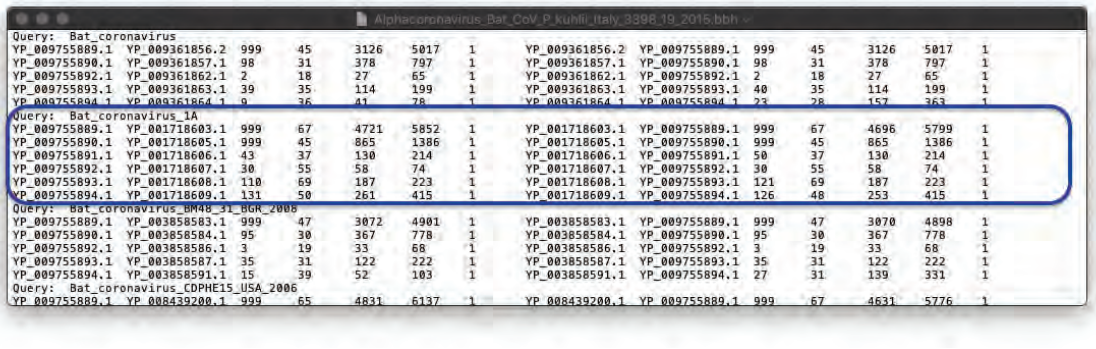
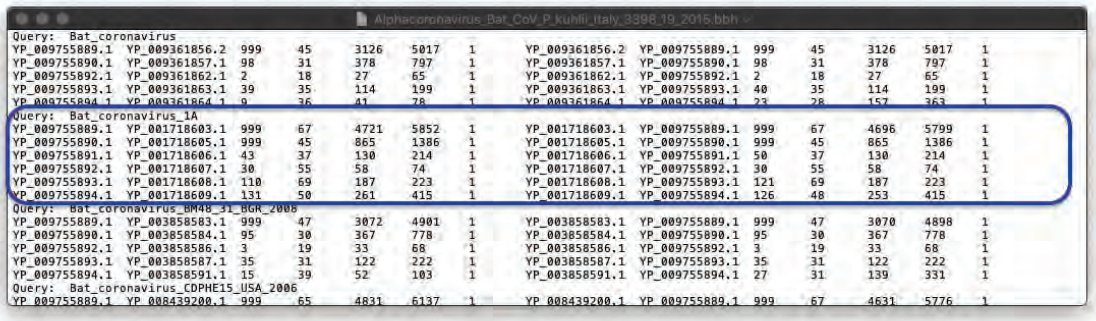
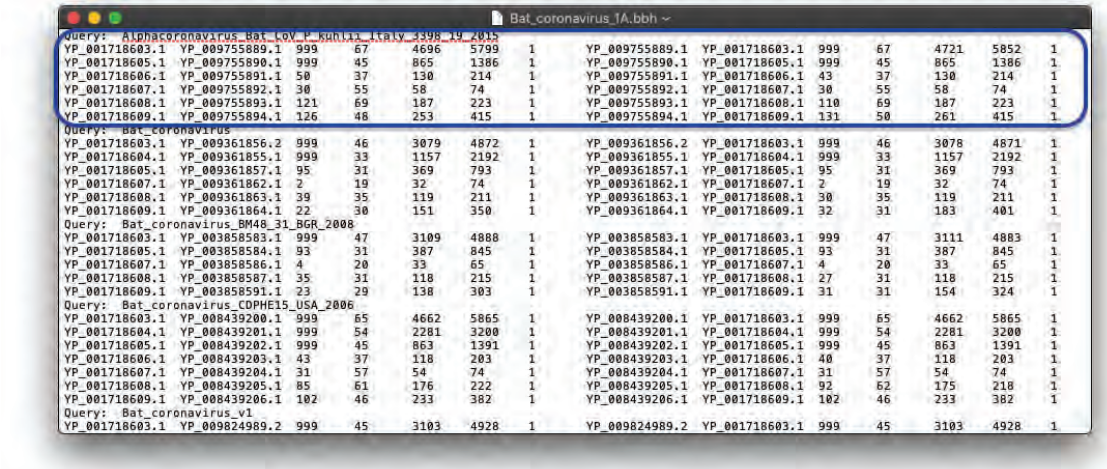


Figura 19. Captura de Pantalla. Primer ejemplo de BBH. Fuente: Propia.

Se puede apreciar en la



la estructura de un archivo (.BBH) que contiene los aciertos bidireccionales desde el organismo "Alphacoronavirus_Bat_CoV_P_kuhlii_Italy_3398_19_2015" con respecto de los demás 67 organismos del presente estudio. En la misma captura se han demarcado los aciertos que se comparte con el organismo "Bat_coronavirus_1A". Se aprecia que estos datos provienen de los archivos (.BHT). Sin embargo, aquí se tiene la óptica desde los dos organismos y sus respectivos valores de homología, al igual que los puntajes y los valores que indican el inicio y fin de cada acierto. Para evidenciar la bidireccionalidad de estos aciertos entre los organismos que se han tomado para este ejemplo se muestra en la



, la cual contiene los mejores aciertos bidireccionales que se han generado desde el organismo "Bat_coronavirus_1A" con respecto de los otros 67 organismos.

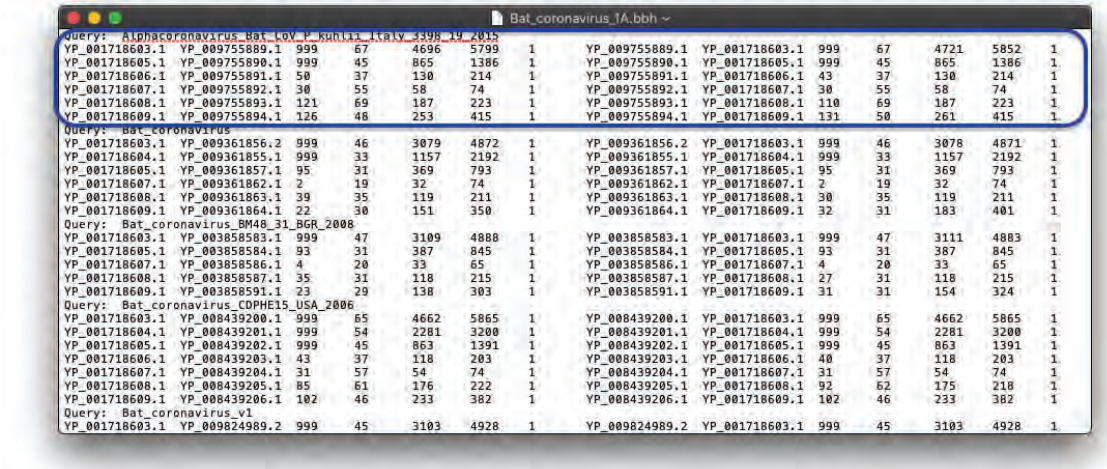
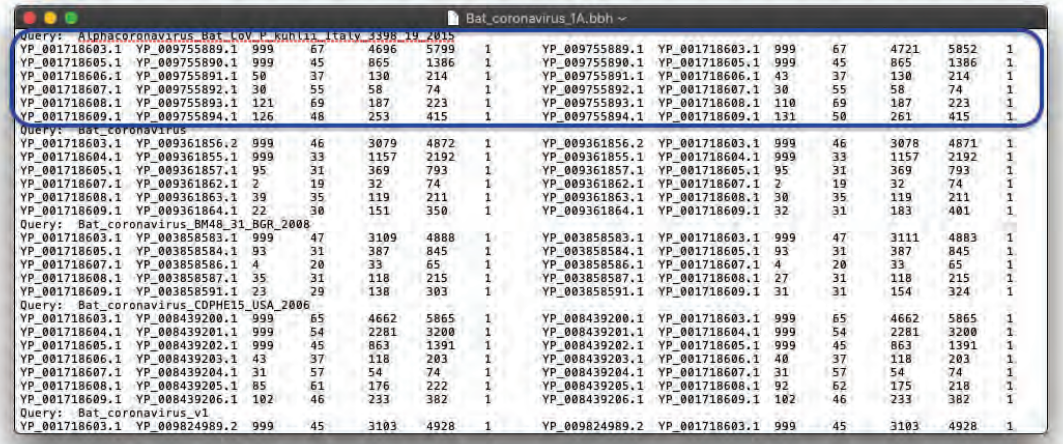
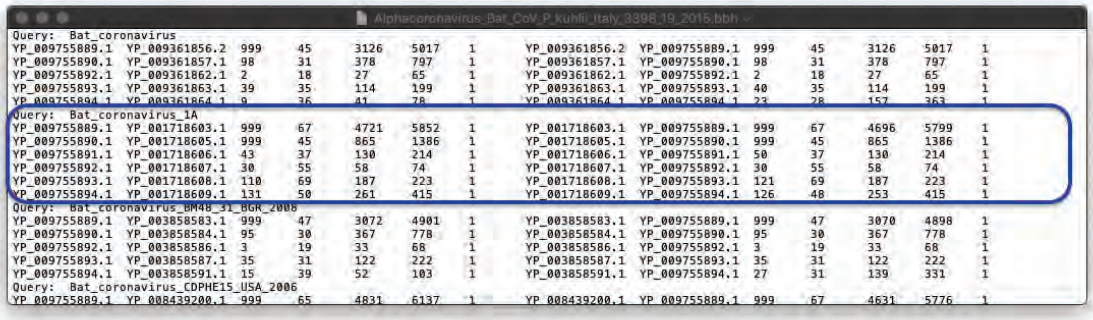


Figura 20. Captura de Pantalla. Segundo ejemplo de BBH. Fuente: Propia.

Se puede apreciar que los datos con respecto del organismo "Alphacoronavirus_Bat_CoV_P_kuhlii_Italy_3398_19_2015" demarcado en la



están consignados en forma de espejo en el archivo mostrado en la



.

Al consolidar la información de los aciertos entre las proteínas que conforman los virus de la Tabla 4, se puede decir que se tienen datos para poder iniciar a realizar diversos análisis como puede ser realizar un ordenamiento filogenético como los que se han presentado en (Ponce de León Sentí et al., 2015; E. E. Ponce-de-Leon-Senti et al., 2020; Reyes-Gallegos, 2019).

4.3. Matriz de Distancias.

Este proceso es el resultado de enlistar las distancias de la totalidad de las proteínas inmersas en el estudio. Se puede considerar que este es el ultimo paso del alistamiento de los datos para que estén dispuestos como entrada del algoritmo de clusterización. Teniendo en cuenta lo anterior no se le debe restar importancia y su correcta ejecución es clave para el avance del proyecto.

Para la construcción de la matriz de distancias se emplearon como datos de entrada los ya obtenidos 68 archivos de aciertos o hits bidireccionales (BBH) con los cuales se llevaron a cabo diferentes labores que permitan tener los datos para en siguientes pasos poder conocer los organismos que han sido agrupados.

En el proceso de esta matriz se toma cada proteína que ha tenido un BBH con otra para ser representadas a forma de un grafo pesado, del cual el peso de la arista será el valor del BBH. Se ha convenido que a mayor valor del BBH mayor homología entre las proteínas.

En la Figura 20 se han marcado los aciertos entre dos organismos, donde las primeras proteínas de la lista tienen un nivel alto de homología estas son las dos primeras filas que tienen 999 en la tercera columna. Ahora bien, en tercera fila se evidencia que el valor del acierto en un sentido es 50 y en el contrario se tiene un valor de 43, esto lleva a tomar en cada BBH el promedio en sus dos sentidos para su representación a manera del peso entre dos vértices de un grafo.

Figura 21. Representación de BBH en forma de aristas de un grafo. Fuente: Propia.

Para brindar un mejor contexto se ha creado la Figura 21 en la que se ven representadas en forma de aristas de un grafo los aciertos bidireccionales de la primera y tercera fila de la Figura 20. Al observar con mayor detenimiento la Figura 20 se puede observar que la proteína "YP_001718603.1" presenta aciertos bidireccionales con proteínas de otros organismos del estudio, como por ejemplo: "Bat_coronavirus", "Bat_coronavirus_BM48_31_BGR2008", "Bat_coronavirus_CDPHE15_USA_2006" y posiblemente otros más, con lo que se puede considerar que se tendrá un grafo con un número considerable de aristas y vértices que representen los BBH.

El resultado de la elaboración de la matriz de distancias se ha condensado en una matriz de 585 filas y 585 columnas, donde las etiquetas de las filas y columnas son los nombres de las proteínas y los valores de los aciertos o hits bidireccionales son los pesos entre ellas.

	NP_040829.1	NP_040831.1	NP_040832.1	NP_040833.1	NP_040834.1	NP_040835.1	NP_040836.1	NP_040837.1	NP_040838.1	NP_045298.1	NP_045299.2	NP_045300.1	NP_045301.1
NP_040829.1	10000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_040831.1	0.0	10000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	109.0	0.0
NP_040832.1	0.0	0.0	10000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_040833.1	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_040834.1	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_040835.1	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	0.0	0.0	0.0	31.0
NP_040836.1	0.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_040837.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	0.0	0.0
NP_040838.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0	0.0
NP_045298.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	0.0	0.0
NP_045299.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0	0.0
NP_045300.1	0.0	109.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10000.0	0.0
NP_045301.1	0.0	0.0	0.0	0.0	0.0	31.0	0.0	0.0	0.0	0.0	0.0	0.0	10000.0
NP_045302.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	27.0	0.0	0.0	0.0	0.0
NP_058422.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	999.0	0.0	0.0
NP_058423.1	999.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	999.0	0.0	0.0	0.0
NP_058424.1	0.0	108.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	93.0	0.0
NP_058425.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_058426.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_058427.2	0.0	0.0	0.0	0.0	0.0	22.0	0.0	0.0	0.0	0.0	0.0	0.0	48.0
NP_058428.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	26.0	0.0	0.0	0.0	0.0
NP_058429.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP_066134.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	999.0	0.0	0.0

Figura 22. Captura de Matriz de Distancias. Fuente: Propia.

Detallando la Figura 22 se observa la diagonal en la cual se encuentran los valores de la proteína con si misma. Para poder diferenciar este dato, se ha asignado un valor considerablemente mayor que los aciertos bidireccionales mas altos ente proteínas distintas, es decir que el valor de homología de una proteína consigo misma debe ser muy alto por ser ella misma, y se ha representado con "10000".

Esta forma matricial de presentar la información puede resultar interesante para consultas o revisiones visuales. Sin embargo, se requiere hacer un poco más de alistamiento para obtener otras representaciones de la misma información.

4.3.1 Lista de Aristas.

La forma de representación de la matriz de distancias mostrada en la Figura 22 se puede tomar como insumo para poder representar los aciertos o hits bidireccionales en forma de lista de aristas y de esta forma abordar la clusterización.

Resultado del proceso de apilar las proteínas que han tenido algún acierto bidireccional y su respectivo valor para tomarlo como peso en el grafo, se encontraron 11400 aristas para 585 vértices del grafo, el cual se puede describir como:

$$G = V, A$$

Donde *V* hace referencia a los vértices que componen al grafo que para el presente problema son las proteínas aportadas por cada virus.

A hace referencia a las aristas que se forman entre cada par de vértices. Para el grafo del problema se tendrá que :

$$G = 585, 11400$$

Ecuación 2. Cardinalidad del conjunto de vértices y conjunto de arístas.

585	11400	0
NP_040829.1	NP_058423.1	999.0
NP_040829.1	NP_073550.1	999.0
NP_040829.1	NP_150074.1	999.0
NP_040829.1	YP_001552235.1	999.0
NP_040829.1	YP_001718604.1	999.0
NP_040829.1	YP_001718611.1	999.0
NP_040829.1	YP_001876436.1	999.0
NP_040829.1	YP_001941165.1	999.0
NP_040829.1	YP_003029845.1	999.0
NP_040829.1	YP_007188578.1	999.0
NP_040829.1	YP_008439201.1	999.0
NP_040829.1	YP_008719931.1	999.0
NP_040829.1	YP_009019181.1	999.0
NP_040829.1	YP_009047203.1	999.0
NP_040829.1	YP_009194638.1	999.0
NP_040829.1	YP_009199240.1	999.0
NP_040829.1	YP_009256196.1	999.0
NP_040829.1	YP_009328934.1	999.0
NP_040829.1	YP_009361855.1	999.0
NP_040829.1	YP_009380519.1	999.0
NP_040829.1	YP_009513009.1	999.0
NP_040829.1	YP_009725295.1	999.0
NP_040829.1	YP_009755896.1	999.0
NP_040829.1	YP_009824966.1	999.0
NP_040829.1	YP_009824979.1	999.0

Figura 23. Captura de Lista de Aristas. Fuente: Propia.

En la Figura 23 se compila la información de la matriz de distancias de la Figura 22, excluyendo los pares de proteínas que no han tenido aciertos bidireccionales entre ellas. También se excluyen las coordenadas de la proteína con ella misma, es decir la diagonal de la matriz de distancias que tienen valores de 10000.

Para facilitar los procesos de computo se ha requerido crear una correspondencia para las proteínas asignándoles un número con el cual se identifiquen en el grafo, pues de esta manera será más ágil la búsqueda por números que al hacer la comparación de las cadenas de texto de las proteínas. A este archivo se le ha llamado "proteins_list.csv" (Figura 24).

	proteins_list
0	NP_040829.1
1	NP_040831.1
2	NP_040832.1
3	NP_040833.1
4	NP_040834.1
5	NP_040835.1
6	NP_040836.1
7	NP_040837.1
8	NP_040838.1
9	NP_045298.1
10	NP_045299.2
11	NP_045300.1
12	NP_045301.1
13	NP_045302.1
14	NP_058422.1
15	NP_058423.1
16	NP_058424.1
17	NP_058425.1
18	NP_058426.1
19	NP_058427.2
20	NP_058428.1
21	NP_058429.1

Figura 24. Captura de la lista de proteínas. Fuente: Propia.

Teniendo como referencia los datos de las listas mostrados en la Figura 23 y Figura 24 se puede tener el grafo completo en términos del número de vértice que corresponde a cada una de las proteínas, dando como resultado una lista de aristas que ahora se encuentra en términos del número asignado a cada proteína, pero es simplemente otra forma de representar la lista de la Figura 23. Una muestra de este resultado es la muestra que se observa en la Figura 25

585	11400	0
0	15	999.0
0	25	999.0
0	33	999.0
0	83	999.0
0	91	999.0
0	98	999.0
0	106	999.0
0	118	999.0
0	156	999.0
0	267	999.0
0	277	999.0
0	284	999.0
0	296	999.0
0	306	999.0
0	333	999.0
0	339	999.0
0	373	999.0
0	392	999.0
0	404	999.0
0	414	999.0
0	431	999.0
0	466	999.0

Figura 25. Captura de Lista de aristas en terminos de números. Fuente: Propia.

4.3.2 Verificación del Grafo.

Para avanzar en la comprensión del grafo derivado de los aciertos bidireccionales se propone verificar la característica de grafo conexo o conectividad lo que se entiende en que para cualquier pareja de nodos sean (a, b) debe existir como mínimo un camino para conectar el vértice *a* con el vértice *b* (Meza H. & Ortega F., 2004).

Lo anterior se realiza haciendo algunas modificaciones del algoritmo de Kruskal (Kruskal, 1956) que tiene por objetivo encontrar un árbol de expansión mínimo en un grafo que sea conexo, sin embargo, para fines prácticos se ha decidido realizar modificaciones para validar la conexión del grafo y buscar el árbol de expansión máxima buscando siempre los pesos o costos más altos del grafo.

En el problema se encontró que el grafo completo de la Ecuación 2 no es conexo y se compone por 9 componentes conexas que conforman un bosque de soluciones de se generaron de la ejecución del algoritmo de Kruskal. Una captura del resultado de esta ejecución se muestra en Figura 26.

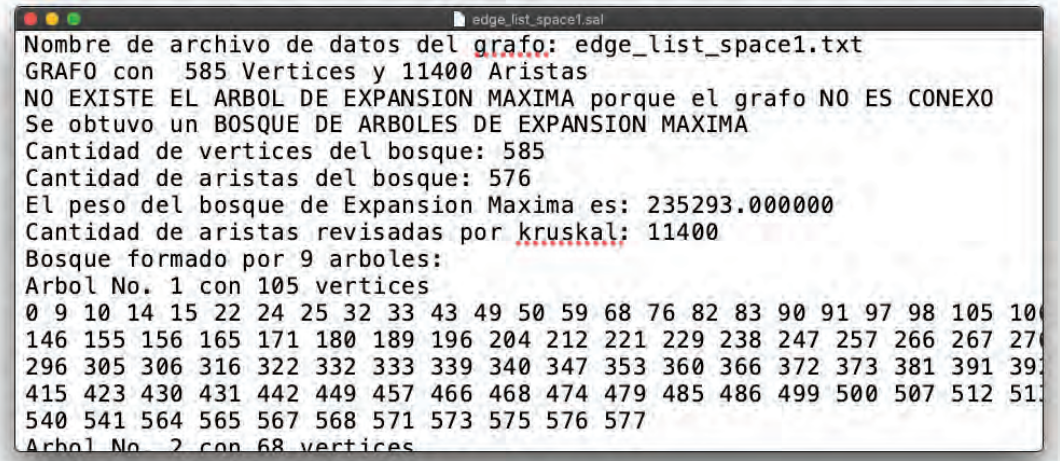


Figura 26. Captura resultado de Kruskal. Fuente: Propia.

El obtener estos componentes conexos permite tener grafos de menor tamaño y trabajarlos generando una menor carga computacional. En la siguiente tabla se muestra el resumen de los componentes conexos obtenidos.

Componente	Vértices	Aristas
1	105	2843
2	68	2278
3	268	1853
4	67	2211
5	67	2207
6	3	3
7	3	3
8	2	1
9	2	1

Tabla 5. Componentes Conexas. Fuente: Propia.

Con cada uno de los componentes conexos que se enlistan en la Tabla 5 se realizará por separado el procesamiento para la clusterización sin afectar la integridad del problema, pues cada uno de los componentes conexos pertenecen al grafo completo del problema.

4.3.3 Observaciones.

Para el capítulo que aquí concluye y que es un puente entre los datos antes de ser analizados con su tratamiento se hacen las siguientes anotaciones.

1. El uso de los términos Mejores o Mejor Aciertos Bidireccionales y Mejores Hits Bidireccionales se han usado indistintamente para hacer referencia al mismo concepto de BBH descrito en la sección que se encuentra la Figura 10.
2. Los archivos que se han generado en cada paso son archivos de texto plano que no contemplan una codificación o encriptación, pero se han usado algunas extensiones a discreción propia para identificar su contenido. En algunos para facilidad del desarrollo se ha optado por archivos con extensión CSV (Comma Separated Values).

Capitulo 3: Modelo Propuesto.

En el desarrollo de este capítulo primero se formaliza el planteamiento del problema de optimización específico para introducir las funciones objetivo y su respectiva explicación con las cuales se realiza la búsqueda de los agrupamientos de las proteínas. En segundo lugar, la selección de las metaheurísticas que se consideren para la clusterización de las proteínas de los organismos de la familia coronaviridae. Por último el capítulo cierra con la experimentación que muestra los resultados de las agrupaciones que se lograron clusterizar usando la metaheurística elegida con las funciones objetivo.

1. Planteamiento del Problema de Optimización Especifico.

La representación de las proteínas y las similaridades entre ellas por medio de nodos y aristas de grafos permite contemplar la propuesta de agrupar a las proteínas y sus relaciones con cliques, entendiendo estos cliques como un subconjunto $C \subseteq V$ de forma tal que existe una arista A para cualquier par de vértices de C (Ordoñez Guillén, 2014). Para lo anterior es necesario tener en cuenta la definición que se dio de un grafo en secciones anteriores.

$$G = (V, A)$$

Teniendo V, como la cantidad de vértices y A como la cantidad de aristas de un grafo G.
Un grafo G, puede tener varios cliques que cumplan las condiciones para ello, es decir, contar con uno o varios subgrafos que sean completos (Ponce et al., 2006). Las definiciones anteriores son necesarias para poder dar el entendimiento al problema del clique máximo (PCM) en el cual se requiere encontrar los cliques de mayor tamaño de un grafo dado (Bomze et al., 1999). En (Ordoñez Guillén, 2014) se cita que el problema del clique máximo es agrupado en los problemas de tipo NP-Completo y que esto fue demostrado en (Karp, 1972), indicando que los algoritmos que puedan resolver este problema de manera exacta tendrán un tiempo que se incrementa exponencialmente en función de los vértices que tenga el grafo al cual se está haciendo la búsqueda de sus cliques máximos.
Para condensar el planteamiento del problema específico se puede indicar que se buscará para cada una de las componentes conexas identificadas en la Tabla 5, el clique máximo a través de las funciones objetivo de la siguiente sección.

2. Selección de las Funciones Objetivo.

En el desarrollo de esta sección se enunciarán las funciones objetivo que se han utilizado para realizar la clusterización con el Algoritmo de Estimación de la Distribución y se mostrarán las definiciones de sus respectivos parámetros.

2.1. Parámetros de las funciones.

Se hace necesario describir los diferentes parámetros que se usarán en las funciones objetivo para poder dar un entendimiento al lector, buscando que la comprensión sea completa. Cada componente conexa requerirá de estos parámetros que serán evaluados en cada agrupación presentada evaluada por el algoritmo.

- Nodo: Es referente cada uno de los vértices del grafo obtenido. Se entenderá como un vértice del grafo con el cual se ha propuesto representar una proteína.
- MaxNodos: Es la cantidad máxima de nodos que se tienen para un clique obtenido. Se determina por la cantidad de organismos que tienen representación de al menos una proteína en la componente conexa. Se pueden tener casos de organismos que aporten más de una proteína, en este caso se cuenta como única participación.
- BBH (Bidirectional Best Hits): Como se ha enunciado en el apartado de Metodología de análisis de datos Ómicos y representado en la Figura 10, es el valor de similitud entre dos proteínas de manera bidireccional.
- ∑BBH: En las funciones objetivo se aplicará la sumatoria de todos los BBH que intervienen en el clique obtenido.
- MaxBBH: Es el valor máximo obtenido de un BBH. Se tomará uno por cada componente conexa. El valor máximo posible será de 999.0

2.2. Funciones Objetivo.

Los algoritmos de clusterización tienen el ideal de construir agrupaciones de elementos lo más similares entre elementos y a su vez que estos grupos sean altamente diferentes entre sí (Chen et al., 2018). Las funciones objetivo en los procesos de optimización apuntan a la maximización o minimización de un valor, en el momento de una optimización multiobjetivo, estas funciones pueden ser contrarias y el maximizar una puede reducir la otra o las otras dependiendo de la cantidad. En este caso las dos funciones objetivo se pretenden maximizar, sin embargo, el lograrlo puede resultar complicado por la naturaleza del problema y los parámetros que se utilizan en el proceso.

Teniendo en cuenta la Figura 2 para el presente problema se describirán las funciones objetivo con las que se ha de realizar la sección de experimentación.

2.2.1. Calidad de Aristas del subgrafo.

La función objetivo que se enuncia en este apartado será conocida como: Calidad de Aristas del subgrafo (Esqueda, 2020) y tiene como intención lograr identificar el máximo de similitud entre los nodos del clique que se ha encontrado, para esto se obtendrán valores normalizados entre 0 y 1 con los cuales se podrá percibir que tan semejantes son los nodos que están siendo parte del clique entre ellos mismos y razonar que tan similares son las proteínas que conforman el clique.

En la Ecuación 3 se puede visualizar la función respectiva en la cual el numerador se compone de la sumatoria de los BBH que integran el clique hallado. Luego el denominador contempla un producto entre el valor del BBH más alto en el clique con la cantidad de aristas del clique (lo que también se podría ejemplificar como el peso de las aristas por la cantidad de aristas). El cociente que se describe tendrá un valor entre 0 y 1 (cero y uno) donde valores cercanos al cero permitirán inferir que los valores de los BBH son bastante escasos y con ello que las similitudes entre ellos son pocas. Para el presente problema, podría entenderse como que las proteínas del clique en el que se está trabajando tienen poca semejanza. Por otra parte en el caso de valores cercanos al uno, se podrá inferir que las proteínas del clique poseen altos valores en sus BBH, siendo proteínas de bastante semejanza. Lo que se puede entender también como que los pesos de las aristas del grafo son de valores altos.

$$F_1 = \frac{\sum BBHs}{\left[\frac{(Nodos) * (Nodos - 1)}{2}\right] * MaxBBH}$$

Ecuación 3. Función Objetivo 1. Fuente: (Esqueda, 2020).

2.2.1.1. Ejemplo de Calidad de Aristas del subgrafo.

En este apartado se propondrá un ejemplo de la aplicación de la función presentada en la Ecuación 3 tomando por referencia los datos del componente conexo 6 de la Tabla 5. Lo anterior permitirá conocer el valor que se espera obtener de esta componente para la función. Los valores para los parámetros usando la componente conexa son:

Parámetro	Representación en problema de optimización (Figura 27)	Valor	Fuente de Obtención
Nodos	u_2	3	Componente Conexa 6
MaxBBH	u_3	53.0	Validación en Componente Conexa 6.
∑BBH	u_1	72.0	Sumatoria de BBHs de Componente Conexa 6.
MaxNodos	u_4	3	Cantidad de organismos que aportan al menos una proteína.

Tabla 6. Parámetros para función objetivo 1, basados en componente conexa 6.

Al tomar los valores de la Tabla 6 y aplicarlos en la función objetivo presentada en la Ecuación 3 se tiene:

$$\frac{72}{\left[\frac{(3) * (2)}{2}\right] * 53} = \frac{72}{159} = 0{,}452$$

Lo cual representa qué tanta similitud se puede presentar en los nodos del subgrafo, en este caso particular se evidencia que al tomar un clique con la totalidad de los nodos de la componente conexa (en este ejemplo 3 nodos) por las diferencias de valores de los BBH, se tendrá un valor que podría ser más alto, dado que el máximo será 1.

2.2.2. Cantidad de Nodos del subgrafo.

Para la segunda función objetivo se tendrá un cálculo también normalizado entre valores de 1 y 0 (uno y cero) con la intención de encontrar el clique de mayor tamaño posible en cada componente conexa. Es decir, al tener un valor cercano a cero se puede inferir que no todos los vértices están contenidos en el clique y que podría existir un clique de mayor tamaño. Al ser este valor igual a 1, indicará que todos los nodos se encuentran en el clique propuesto.

$$F_2 = \frac{Nodos}{MaxNodos}$$

Ecuación 4. Función Objetivo 2. Fuente: (Esqueda, 2020).

El numerador para la función presentada en la Ecuación 4, hace referencia a la cantidad de vértices que están contenidos en el clique mientras que el denominador es la cantidad máxima de vértices que se pueden tener para el clique dentro del respectivo componente conexo.

2.2.2.1. Ejemplo de Cantidad de Nodos.

En este ejemplo se retoman los parámetros de la Tabla 6 que se aplican en la función objetivo de la Ecuación 4. Ejemplo de ponerle valores a la función 2.
Para esta componente conexa se tiene que el máximo de nodos (o de proteínas que aportan al menos una proteína a la componente) es de 3. Entendiendo que es una componente conexa de 3 nodos con la posibilidad de generar un clique de máximo 3 nodos se tiene que:

$$\frac{3}{3} = 1$$

Esto muestra que se tiene un clique para esta componente conexa que puede contener todos los nodos, llevando a la función al máximo de su optimización.

Los valores de las secciones anteriores en que se muestran ejemplos de la aplicación de las funciones objetivo a usar son valores esperados del proceso de optimización que permiten decir que dicha componente puede tener una agrupación de tres proteínas que representan a tres organismos (es decir que agrupa al total de los organismos en un clique) y que el valor de similaridad basado en la normalización de los BBH que representan sus relaciones es de 0,452. Esto como ya se ha mencionado es debido a las diferencias de los valores de los BBH del clique.

En la siguiente figura se presenta la formalización del problema de optimización especifico, teniendo en cuenta las funciones descritas en este apartado (Ecuación 3 y Ecuación 4) adaptadas a la forma mostrada en la Figura 2, en la que se propone una representación de un problema multiobjetivo.

$$\max \quad f_1(\mathbf{X}) = \frac{u_1}{\left[\frac{u_2 * (u_2 - 1)}{2}\right] * u_3}$$

$$\max \quad f_2(\mathbf{X}) = \frac{u_2}{u_4}$$

$$\text{s.t.}$$

$$0 \leq u_1 \leq 999.0$$
$$1 \leq u_2 \leq 68$$
$$0 \leq u_3 \leq 999.0$$
$$1 \leq u_4 \leq 68$$

Figura 27. Problema de optimización. Fuente: Propia.

Las definiciones para los términos de la Figura 27 se indican en la Tabla 6 que contiene los valores para la componente conexa 6 en la segunda columna, de donde se define u_1 como $\sum$BBH lo cual es la sumatoria de los BBH (Hits Bidireccionales); se tiene u_2 para representar los Nodos que corresponden al subgrafo que se revisa. Con u_3 se tiene el valor máximo de los BBH MaxBBH y para u_4 se referencia el parámetro de MaxNodos. Para las funciones objetivo se ingresan los valores de los subgrafos a ser evaluados como cliques en un arreglo de 1 y 0 y se etiquetan con $\boldsymbol{X}$.

Consideraciones: Con las funciones objetivo que se han determinado para realizar la clusterización se debe tener en cuenta que se buscará por medio de la calidad de las aristas (Ecuación 3) agrupar las proteínas entre las que mayores valores presentan para los BBH, es decir, las proteínas que tienen mayor similitud, a la vez con la función de cantidad de nodos (Ecuación 4) se busca generar que cada agrupación sea lo más grande posible. También se debe resaltar que previamente se ha realizado una segmentación de las proteínas en las nueve (9) componentes conexas (Tabla 5), estas componentes conexas se entienden como las agrupaciones que son distintas entre sí mientras sus elementos tienen igualdad entre ellos.

3. Selección de la Metaheurística.

Se puede encontrar una primera aproximación para la identificación de las metaheurísticas a utilizar en la sección de Heurísticas y Metaheurísticas en la que se realizó un acercamiento a estas técnicas de manera general y clasificatoria según la literatura (Blum & Roli, 2003). Ahora bien, se plasmará una revisión de las técnicas metaheurísticas que por sus características generales se han contemplado para ser implementadas en este trabajo.

Para retomar el contexto de las heurísticas se generalizará el termino dentro del mundo de la Inteligencia Artificial como un procedimiento que trata de aportar soluciones a un problema buscando tener buen rendimiento tanto en la calidad de la solución que propone como para los recursos que requiere para su funcionamiento (Melián et al., 2003). En capítulos previos se han realizado descripciones generales de los conceptos principales, como la clasificación de los métodos heurísticos, a esto se le deben adicionar las propiedades que se tienen en cuenta al momento de evaluar un algoritmo heurístico y la solución que se dio a un problema con este.

- Eficiencia: Hace referencia a la capacidad computacional requerida, debería ser adecuada y alcanzable con los recursos existentes.
- Asertividad: La solución alcanzada por el algoritmo debe estar contenida en un rango cercano al optimo.
- Robustez: Se puede entender como la probabilidad de llegar a soluciones de baja calidad o alejadas del o de los óptimos propuestos.

En anteriores páginas también se ha tratado el concepto de metaheurística como la propuesta de adición a los algoritmos heurísticos de procesos de aleatorización, estrategias de aprendizaje entre otros con el ideal de lograr exploraciones de soluciones que puedan estar más allá de soluciones óptimas locales. En la reseña que se propone en (de Antonio Suárez, 2011) que se cita del capítulo escrito por Osman Ibrahim y Kelly James (Osman & Kelly, 1996) que: "*Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas de difícil optimización combinatoria en los que los heurísticos clásicos no son efectivos*". De igual forma es importante dar el respectivo crédito a la primera aparición del término de Metaheurística en la publicación de Fred Glover (Glover, 1986) donde proponía ver una búsqueda Tabú como una "Meta-heurística" superpuesta a otra heurística, que trata de buscar más allá de un óptimo local. Es visible que se ha ido engrosando

y complementando el entendimiento de este término a través de la literatura y las diferentes aplicaciones y desarrollos que se han venido derivando desde su aparición.

El impacto de las metaheurísticas es tal que para problemas clásicos se pueden encontrar grandes cantidades de propuestas de solución con estas técnicas, un ejemplo de ello es el problema de enrutamiento de vehículos. De este problema se pueden encontrar un nutrido número de publicaciones como se muestra en (Elshaer & Awad, 2020) del 2009 al 2017 se logran percibir alrededor de 300 documentos analizando dicho problema y las variaciones con que cuenta con un enfoque desde las metaheurísticas. De ese mismo documento resulta valioso rescatar el árbol de la Figura 28 que clasifica las metaheurísticas en dos grandes familias como lo son las basadas en una solución simple y las basadas en poblaciones. De esta última familia se pueden encontrar un par más de divisiones, que son las basadas en multitudes o enjambres como las colonias de hormigas que se inspiran en el fenómeno de las feromonas que usan las hormigas para comunicarse y guiar a su colonia (Blum & Roli, 2003). La otra división agrupa a las técnicas basadas en computación evolutiva.

En esta clasificación de computación evolutiva se encuentran los paradigmas que se ubican los Algoritmos de Estimación de la Distribución, que también se conocen como EDAs (Estimation of Distribution Algorithms). Este tipo de algoritmos se ha seleccionado para el presente proyecto y en las siguientes secciones se mostrarán sus características identificando sus beneficios y los retos que se tienen con la implementación de estas técnicas.

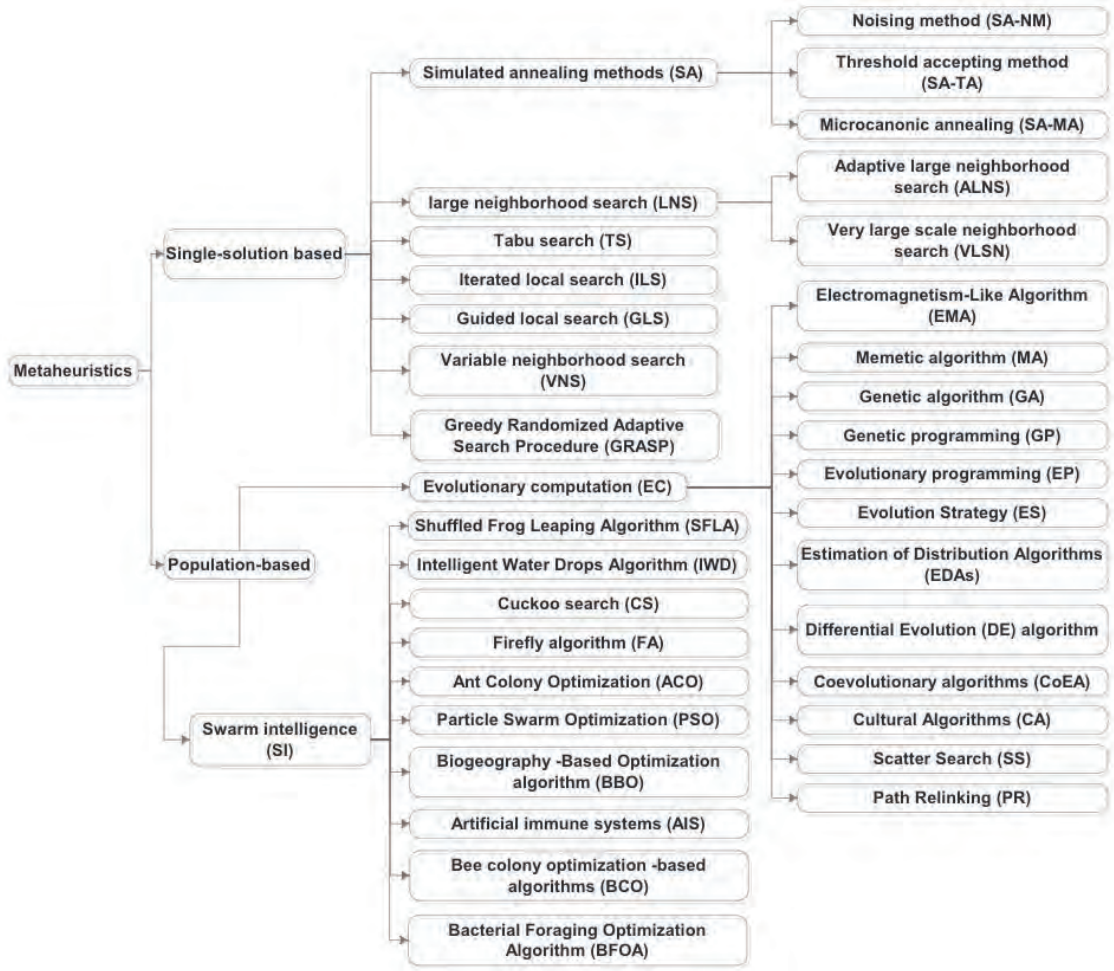


Figura 28. Árbol de clasificación de Metaheurísticas. Fuente: (Elshaer & Awad, 2020)

3.1. Algoritmos de Estimación de la Distribución.

Los algoritmos para la estimación de la distribución (EDA), de manera similar a como sucede en los algoritmos genéticos, trabajan basados en una población para representar un vector con la solución optimizada o en proceso de optimización. Los EDA cuentan con una fundamentación basada en la probabilidad y en poblaciones que evolucionan durante el progreso de su búsqueda (Blum & Roli, 2003). Para tener nuevas poblaciones de soluciones los EDA aplican una estimación de distribución conjunta de la población en un modelo que luego es muestreado (Hauschild & Pelikan, 2011). Algunas aplicaciones en las cuales los EDA han sido un elemento principal se pueden enlistar en diversas áreas del conocimiento, pasando por la bioinformática (Armañanzas et al., 2008) hasta el enrutamiento de sensores inalámbricos (Yuan et al., 2007).

Para entender el funcionamiento general de un EDA se propone en (Mendoza-Gonzalez et al., 2013) tomar el algoritmo a manera de bloques independientes y analizar cada uno como una función independiente con parámetros de entrada y salida, estos bloques también se pueden entender como los diferentes operadores que se requieren en la ejecución de un EDA de forma genérica. A continuación se describen dichos operadores.

- Inicialización: Se genera la población inicial.
- Evaluación: Método para validar los valores objetivo de la función con cada individuo de la población.
- Ordenamiento: Las soluciones de ordenan según la adaptabilidad que tengan.
- Selección: Se toma un subconjunto de individuos para construir el modelo.
- Aprendizaje: Se extrae información y se modela la distribución de las soluciones seleccionadas.
- Muestreo: Se construye la población nueva basada en el modelo.
- Finalización: Se entrega la población que tenga más cercanía con el óptimo de la función o funciones.

Las descripciones anteriores se basan en la propuesta presentada en (Mendoza-Gonzalez et al., 2013) y se representan en la Figura 29, en la que se agrupan los operadores que se ejecutarán repetitivamente hasta llegar a la condición de parada.

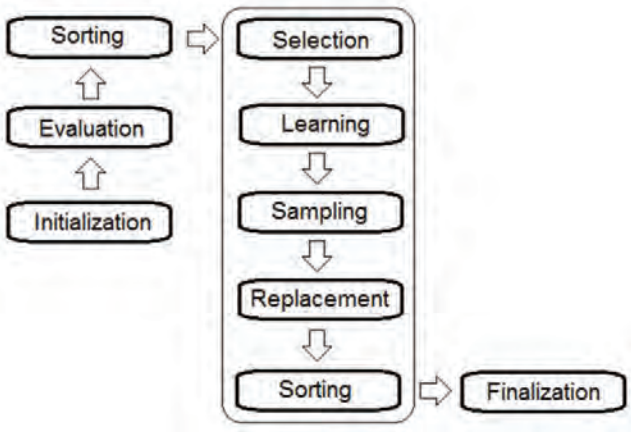


Figura 29. Bloques del proceso de la operación de EDA. Fuente: (Mendoza-Gonzalez et al., 2013).

Los EDA pueden ser clasificados en tres categorías basados en la relación de las variables del proceso como se comenta en (Mendoza et al., 2013) *univariados, bivariados y multivariados.*

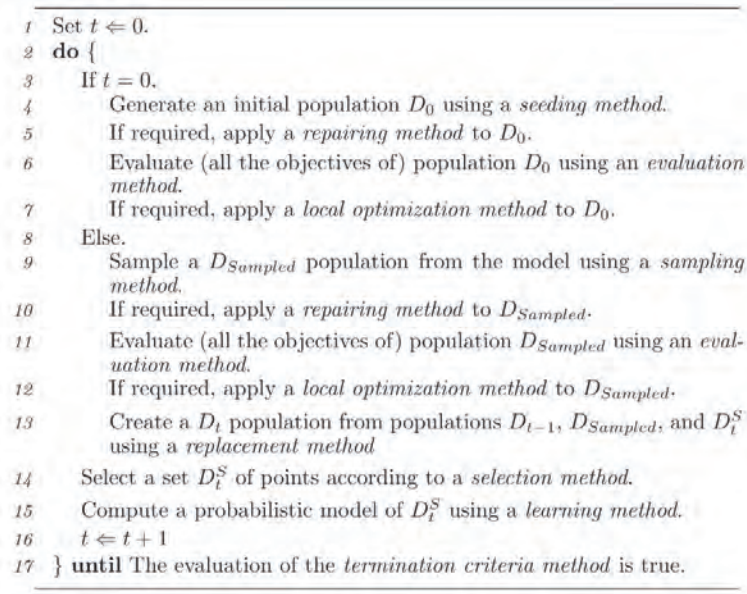
Para los EDA univariados se pueden encontrar algoritmos como el UMDA (Univariate Marginal Distribution Algorithm) y el PBIL (Population-Based Incremental Learning). Para los EDA bivariados se logra identificar en la literatura MIMIC (Mutual-Information-Maximizing Input Clustering) presentado en (de Bonet et al., 1997). Para el enfoque con que se desarrolla el presente trabajo, es necesario considerar diferentes variables y por eso se aplicará un EDA clasificado como multivariado, esto realizará una estimación conjunta de la probabilidad de la población teniendo en cuenta probabilidades condicionales, en las que cada una puede depender de más de una variable (Mendoza et al., 2013). Algunos algoritmos para la categoría de EDA multivariados son EcGA (Extended compact Genetic Algorithm) presentado en (Harik, 1999) y los BOA (Bayesian Optimization Algorithm) introducidos en (Pelikan et al., 1999).

3.2. MATEDA

Los diferentes algoritmos para la estimación de la distribución (EDA) que se han mencionado y otros que se pueden consultar en (Mendoza-Gonzalez et al., 2013) requieren para su propósito de realizar labores de optimización mono y multiobjetivo, bien sea en términos discretos o continuos, la implementación de diversos operadores y la construcción de estos consume tiempo, dedicación y otros recursos computacionales. Algunos investigadores han emprendido en el proceso de generar una herramienta que ofrezca diferentes métodos que son usados comúnmente para la ejecución de un EDA; esta herramienta se ha nombrado como MATEDA (Santana et al., 2010). MATEDA es un paquete de funciones computacionales que pretende solucionar problemas de optimización utilizando algún EDA, en estas funciones se incluyen métodos para procesar, extraer y visualizar características en diferentes modelos.

3.3. Implementación en MATEDA

Para aprovechar las bondades que ofrece MATEDA, sus creadores han dispuesto esta herramienta a manera de código libre y difundido un manual en el que ejemplifican su uso (Santana et al., 2009). Dada la gran cantidad de elementos contenidos en MATEDA, resulta necesario hacer un seguimiento del manual y poder aprovechar las funciones luego de su instalación. En la Figura 30 se muestra la forma en la que se describe un EDA en MATEDA a forma de pseudo-código que es concordante con la presentada en la Figura 29.



1 Set $t \Leftarrow 0$.
2 **do** {
3 If $t = 0$.
4 Generate an initial population D_0 using a *seeding method*.
5 If required, apply a *repairing method* to D_0.
6 Evaluate (all the objectives of) population D_0 using an *evaluation method*.
7 If required, apply a *local optimization method* to D_0.
8 Else.
9 Sample a $D_{Sampled}$ population from the model using a *sampling method*.
10 If required, apply a *repairing method* to $D_{Sampled}$.
11 Evaluate (all the objectives of) population $D_{Sampled}$ using an *evaluation method*.
12 If required, apply a *local optimization method* to $D_{Sampled}$.
13 Create a D_t population from populations D_{t-1}, $D_{Sampled}$, and D_t^S using a *replacement method*
14 Select a set D_t^S of points according to a *selection method*.
15 Compute a probabilistic model of D_t^S using a *learning method*.
16 $t \Leftarrow t + 1$
17 } **until** The evaluation of the *termination criteria method* is true.

Figura 30. Representación de un EDA en MATEDA. Fuente: (Santana et al., 2009).

Para poner en ejecución un EDA con requerimientos específicos usando MATEDA, es necesario definir las funciones objetivo y referenciarlas dentro del comando de MATEDA.
La documentación indica que la ejecución del EDA inicia con la función "RunEDA" la cual recibe por parámetros los siguientes datos.

- PopSize (Tamaño de la población): Es el tamaño de la población con la que se va a trabajar.
- n (Número de Variables): Es la cantidad de variables requeridas para ser optimizadas.
- F (Implementación funciones objetivo): Construcción de las funciones objetivo con las cuales se requiere hacer la optimización.
- card (Cardinalidad de las variables): Rango de valores en los cuales se encuentran las variables.
- cache (Valores que se requieren almacenar de la ejecución): Se definen que elementos se requieren almacenar en cada una de las iteraciones; Población, Modelo probabilístico, Población mejor ajustada.
- edaparams (Parámetros del EDA): Se indican los componentes que usa el EDA, recibe la población inicial, la forma del muestreo, el método de remplazamiento, el método de aprendizaje, el método de selección.

En la implementación de este proyecto se utiliza para los parámetros de la función “RunEDA” una población de 50 individuos, las funciones objetivo de la Ecuación 3 y de la Ecuación 4, se toma para cardinalidad valores 1 o 0 para indicar que una proteína pertenece o no al subgrafo. En los “edaparams” se usan:

- método de aprendizaje: Aprendizaje Gaussiano.
- método de muestreo: Muestreo Gaussiano.
- método de selección: Truncamiento.
- método de remplazamiento: Elitismo y Ordenamiento.
- método de parada: Máxima generación.

En la siguiente sección se muestran los resultados de la ejecución del EDA con los parámetros que se han enunciado.

4. Realización de Experimentos.

En esta sección se enlistan los experimentos realizados y los resultados generados. Las diferencias de parámetros entre cada experimento se muestran para poder tener una guia del desempeño y resultados. Es importante recalcar que el ideal es clusterizar proteínas buscando maximizar los resultados de las funciones objetivo, sin embargo, no se pretende encontrar los máximos de cada función y si proponer formas de ser encontrados estos máximos en la solución de otros problemas.

4.1. Parámetros de los Experimentos.

Como se ha mostrado en capítulos anteriores la herramienta MATEDA (Santana et al., 2009, 2010) es la que se ha usado para la ejecución de los experimentos. Teniendo en cuenta su versatilidad, flexibilidad y robustez.

La Tabla 7 muestra un resumen de los parámetros implementados en los experimentos que se realizaron, cada uno de estos experimentos se ejecutó con las 9 componentes conexas de la Tabla 5.

Experimento 1. • Condición de Paro: Máxima generación (5). • Método de Aprendizaje: Redes Bayesianas. • Método de Selección: Selección por Truncamiento. • Método de Remplazo: Elitismo. • Método de Semilla: Aleatorio.	Experimento 2. • Condición de Paro: Máxima generación (10). • Método de Aprendizaje: Redes Bayesianas. • Método de Selección: Selección por Truncamiento. • Método de Remplazo: Elitismo. • Método de Semilla: Aleatorio.
Experimento 3. • Condición de Paro: Máxima generación (5). • Método de Aprendizaje: Redes Bayesianas. • Método de Selección: Selección por Truncamiento. • Método de Remplazo: Elitismo. • Método de Semilla: Valores Fijos (Todos en 1).	Experimento 4. • Condición de Paro: Máxima generación (10). • Método de Aprendizaje: Redes Bayesianas. • Método de Selección: Selección por Truncamiento. • Método de Remplazo: Elitismo. • Método de Semilla: Valores Fijos (Todos en 1).

Tabla 7. Resumen de parámetros de Experimentos. Fuente: Propia.

4.2. Resultados de los Experimentos.

En las siguientes tablas se compilan los resultados de los experimentos de la Tabla 7. Cada uno de estos experimentos fue ejecutado con todas las componentes conexas que se encuentran en la Tabla 5

- Experimento 1.

Entrada			Salida				
Componente	Vértices	Aristas	F_1	F_2	Clique Obtenido	Clique Máx. Posible	Tiempo Ejecución (s)
1	105	2843	1,000	0,7015	47	67	2815.5308
2	68	2278	0,5717	0,5147	34	68	3637.25
3	268	1853	0,0550	0,1493	10	67	14614.2697
4	67	2211	0,0558	0,5672	38	67	2824.4111
5	67	2207	0,1179	0,5672	38	67	2149.813
6	3	3	1,000	0,6667	2	3	0.92916
7	3	3	1,000	0,6667	2	3	1.0431
8	2	1	1,000	1,000	2	2	0.54765
9	2	1	1,000	1,000	2	2	0.63651

Tabla 8. Resultados Experimento 1. Fuente: Propia.

En esta configuración de los parámetros, se resalta la obtención de un clique de tamaño 47 para la componente 1, también se evidencia que este clique logra su máximo para la función

de calidad de las aristas lo que se puede entender como que todos los BBH que intervienen en el subgrafo tienen el valor máximo posible, en cuanto a la función objetivo de cantidad de nodos, no se logra el máximo de los 67 posibles. Las componentes conexas 8 y 9, logran sus máximos para ambas funciones objetivo y esto se puede entender por el tamaño que tiene cada uno de los subgrafos, pues estos son 2 vértices conectados por medio de una única arista. Las componentes conexas 6 y 7 han logrado para la función objetivo de calidad de las aristas el valor máximo, pero no la cantidad máxima de nodos posibles.

- Experimento 2.

Entrada			Salida				
Componente	Vértices	Aristas	F_1	F_2	Clique Obtenido	Clique Máx. Posible	Tiempo Ejecución (s)
1	105	2843	1,000	0,7015	47	67	7991.3295
2	68	2278	0,4950	0,6324	43	68	7841.1983
3	268	1853	0,0544	0,2090	14	67	5903.1855
4	67	2211	0,0534	0,6866	46	67	8616.1252
5	67	2207	0,1175	0,5522	36	67	5679.9583
6	3	3	0,4528	1,000	2	3	2.1638
7	3	3	1,000	0,6667	2	3	3.3802
8	2	1	1,000	1,000	2	2	1.9708
9	2	1	1,000	1,000	2	2	2.0684

Tabla 9. Resultados Experimento 2. Fuente: Propia.

En este experimento se han adicionado 5 generaciones más para la condición de parada, se encuentran resultados similares a los presentados en la Tabla 8 al momento de contrastar la componente conexa 1, pues presenta los mismos valores para sus funciones objetivo, sin embargo, para las componentes 2, 3 y 4 se incrementaron los tamaños de los cliques y han cambiado los valores de la calidad de las aristas. De las componentes 7, 8 y 9 se mantienen los valores de las funciones objetivo, nuevamente respaldados por sus tamaños.

- Experimento 3.

Entrada			Salida				
Componente	Vértices	Aristas	F_1	F_2	Clique Obtenido	Clique Máx. Posible	Tiempo Ejecución (s)
1	105	2843	1,000	0,8060	54	67	3779.2012
2	68	2278	0,3474	1,000	68	68	5081.5688
3	268	1853	0,7525	0,1045	7	67	2880.989
4	67	2211	0,0471	1,000	67	67	5049.4411
5	67	2207	0,000	0,0149	0	67	1110.4538
6	3	3	0,4528	1,000	2	3	1.6907
7	3	3	0,8519	1,000	3	3	1.9916
8	2	1	1,000	1,000	2	2	1.0895
9	2	1	1,000	1,000	2	2	1.0169

Tabla 10. Resultados Experimento 3. Fuente: Propia.

Este experimento tiene una configuración de 5 generaciones en el parámetro de finalización de la ejecución del algoritmo y un cambio en la forma de generación de su población inicial, contemplando todos los valores del vector de la población en 1, es decir, se propone que todos los nodos en el inicio sean parte de la solución.

Resulta bastante interesante esta configuración de los experimentos, para la componente conexa 1, se logra un clique de tamaño 54 de un 67 posible y la calidad de las aristas para este clique está en su máximo posible. De igual forma en la componente conexa 2 se tiene un clique del tamaño máximo posible para la componente, 68 de 68, pero en la función de calidad de las

aristas el valor se reduce, esto muestra que los valores de las aristas se ven afectados, es decir que los valores de los BBH entre las proteínas que intervienen no tienen valores máximos posibles. También en la componente conexa 4 se logra un máximo para la función de cantidad de nodos, 67 de 67 sin embargo, al igual que en la componente 2 los valores para la calidad de las aristas del clique tienen un valor bastante bajo. Otro elemento que requiere dar atención es la componente 5, que en esta configuración no muestra un clique. La componente 7 al igual que la 8 y 9 logran un valor máximo para el tamaño del clique.

- Experimento 4.

Entrada			Salida				
Componente	Vértices	Aristas	F_1	F_2	Clique Obtenido	Clique Máx. Posible	Tiempo Ejecución (s)
1	105	2843	1,000	0,7761	52	67	11999.9844
2	68	2278	0,3474	1,000	68	68	26356.6113
3	268	1853	0,0512	0,1940	13	67	6737.8333
4	67	2211	0,0471	1,000	67	67	18804.03
5	67	2207	0,000	0,0149	0	67	5202.6671
6	3	3	0,4528	1,000	2	3	6.7982
7	3	3	0,8519	1,000	3	3	13.0505
8	2	1	1,000	1,000	2	2	4.1904
9	2	1	1,000	1,000	2	2	5.4277

Tabla 11. Resultados Experimento 4. Fuente: Propia.

El analizar el experimento 4 en la Tabla 11, resulta bastante similar a la Tabla 10 con el experimento 3, para la componente conexa 1 se ven reducidos los nodos del subgrafo. La componente conexa 3 se incrementa en algunas unidades pero este valor no resulta ser significativo. Las componentes 6, 7, 8 y 9 mantienen los mismos valores para sus funciones objetivo y se ve un incremento en los tiempos de ejecución, esto se esperaba, pues para esta configuración de experimentos se tienen más generaciones que ejecutar.

Discusión de Resultados.

Lograr un resultado con los valores máximos para cada una de las funciones objetivo se consideraría como una agrupación de proteínas, en la que cada una de ellas representa la participación de un organismo en el subgrupo y que a su vez, estas proteínas tienen un alto nivel de similitud. Es decir que en el subgrafo exista un nodo por cada organismo y que estos tengan valores muy altos en las aristas que los relacionan.

En la Tabla 5 de las componentes conexas resultantes, las componentes 6, 7, 8 y 9 son agrupaciones que tienen menos vértices en comparación con las componentes conexas 1, 2, 3, 4 y 5 que se son bastante cercanas a la cantidad total de organismos presentes en el problema (68 organismos). Las diferencias en el tamaño de las componentes generá diferencias en los tiempos de ejecución, situación que se puede contemplar en las tablas de los experimentos.

Para los experimentos 1 y 2 (con sus parámetros en la Tabla 7) con los resultados en la Tabla 8 y Tabla 9 se tiene una semilla por método aleatorio, lo que significa que el vector que representa las posibles soluciones en la inicialización del algoritmo se construye aleatoriamente. En estos dos experimentos que se diferencian por la condición de paro, se lograron cliques de hasta 47 vértices con valores de optimización de (1; 0,7015), lo que indica que de los 47 vértices que componen el subgrafo, sus aristas tienen los valores más altos, con lo que se representa una alta similaridad entre las proteínas que representa.

De los experimentos 3 y 4 con los resultados en la Tabla 10 y la Tabla 11 que tienen una semilla con todos los valores del vector de inicialización con un valor fijo en 1, y sus condiciones de paro son 5 y 10 generaciones respectivamente, se lograron resultados en con subgrafos del tamaño máximo posible pero con valores reducidos en los pesos de las aristas que conectan dichos vértices; es el caso de la componente 2 en los experimentos 3 y 4 en que se logra un máximo para la función 2 de tamaño del clique, con 68 vértices, pero se compromete la calidad del subgrafo, el valor de esa optimización fue de (0,3474; 1), esta situación se repite en los experimentos 3 y 4 para la misma componente. De estos 2 experimentos también llama la atención la componente 4, al lograrse un valor de optimización de (0,0471; 1) con un valor máximo en la cantidad de vértices posibles para la componente (67) sin embargo, el valor de la calidad resulta ser bajo, esto muestra que los valores de los BBH que participaron de este subgrafo no eran los más altos.

Las agrupaciones de las componentes 2 y 4 como ya se mencionó, resaltan por lograr un valor máximo en el tamaño, 68 y 67 respectivamente, con lo que se han contrastado las proteínas

que los componen con las proteínas que se obtuvieron en (Galvis-Motoa et al., 2021), en el que se indicó que se había hallado un clique de 68 proteínas por medio de los métodos propuestos en (Ponce de León Sentí et al., 2015, 2022) y luego fueron consultados en el CDD (*Conserved Domains Database (CDD) and Resources*, 2020), donde esta agrupación de proteínas resultaron ser de tipo "spike" o espina, (lo que da el nombre a la familia de virus). El resultado de esta comparación no mostro que fueran las mismas proteínas o del mismo tipo, lo que transforma los cliques mencionados en objetos de mayor interés, pues estas proteínas que los componen al ser similares propenden a generar características comunes para los virus que las contienen abriendo la puerta a encontrar nuevos entendimientos sobre el virus.

Ahora bien, otros cliques también generan bastante interés por su tamaño sin requerir ser máximo, como por ejemplo la componente 1, que a través de los experimentos proporciono valores de optimización desde 47 y hasta 54 proteínas de un máximo de 67 posibles y siempre un valor de optimización máximo para la función de calidad del clique, en el que las aristas del subgrafo, siempre proporcionaron valores superiores para alcanzar sus máximos; Estos cliques resultarán interesantes para ser revisados por expertos en el área biológica para tener nuevos conocimientos del virus.

Luego del proceso que se ha descrito en la extensión del documento, en la Figura 31 se propone la consolidación de las diferentes fases que se han llevado a cabo para lograr la clusterización.

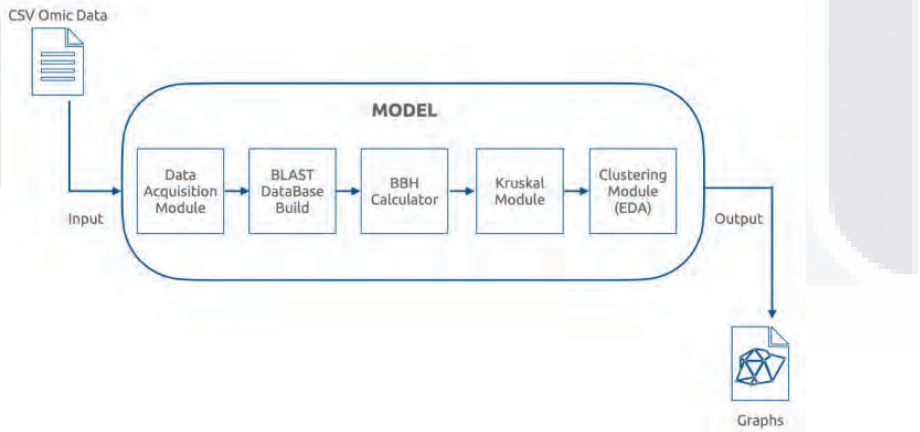


Figura 31. Modelo propuesto. Fuente: Propia.

En el modelo se contempla que la entrada de datos sea un fichero que contenga los datos de las rutas en las que se almacenan los datos ómicos. Con esta información se procede a la adquisición de datos, la construcción de la base de datos necesaria para la ejecución del BLAST.

La salida del BLAST, se usa para realizar los cálculos de los BBH y determinar las relaciones de las proteínas de los organismos a estudiar. El módulo de Kruskal se encarga de revisar si se tiene o no un grafo conexo, en caso de ser no conexo se generan las componentes conexas. Teniendo las componentes conexas, estas se llevan al EDA para poder encontrar los respectivos cliques que son la salida del modelo.

Externo al modelo presentado, se han desarrollado prototipos de Bots para realizar de manera automatizada la revisión de los clusters de proteínas generados por el modelo. También se ha desarrollado un prototipo a través de la dirección de la tesina: "Integración de los módulos de preprocesamiento en la metodología basada en mejores aciertos bidireccionales: Software UAA-PROT" del estudiante José Ramsés Moreno González de la carrera de Ingeniería en Computación Inteligente, que migra e integra gran parte de los módulos del modelo a un entorno web. Otro proyecto que se deriva del presente es el miniproyecto: MP-22-063: Módulo de Procesamiento de Información Ómica sobre un Servidor Web, con el que se pretende llevar los diferentes módulos integrados en el modelo a una versión basada en arquitectura web, con la que se permita el uso de estas herramientas a usuarios externos que requieran trabajar con datos ómicos de cualquier organismo.

Conclusiones.

Esta sección presenta las conclusiones del proyecto al igual que las aportaciones que se han generado, también se enuncian posibles caminos que puedan derivar.

Respecto a los objetivos específicos propuestos para el proyecto se puede concluir:

- La metaheurística seleccionada fue un Algoritmo de Estimación de la Distribución (EDA) que se implementó con el uso de la librería MATEDA.
- La inclusión de la librería MATEDA permite flexibilizar la modificación de los parámetros del EDA y brinda la posibilidad de usar diferentes métodos de aprendizaje, semilla, muestreo y hasta remplazo.
- Se definió el problema de optimización con dos funciones objetivo, basadas en la calidad de las aristas y cantidad de nodos.
- Se logro realizar la incorporación de las funciones objetivo en el modelo.
- Los procesos de optimización se pueden llevar a diferentes áreas del conocimiento, en este caso se han llevado al mundo de la biología molecular desde una perspectiva computacional.
- La elección de la librería MATEDA otorgo la facilidad de poder ejecutar experimentos con diferentes parámetros con requiriendo pocas modificaciones.
- En los experimentos que se realizaron se identificó que la configuración de arranque con una semilla compuesta por un vector de unos y un criterio de parada de 10 generaciones, permite hallar cliques con valores cercanos al optimo en las funciones objetivo.

Con el objetivo general se puede concluir teniendo en cuenta el modelo presentado en la Figura 31 en el que fue necesario contemplar etapas para el alistamiento de datos, como el módulo de adquisición de datos, el paso de automatización de las sentencias BLAST para la construcción de la base de datos local, la integración con la metodología para obtención de mejores aciertos bidireccionales y también una sección para la verificar del conjunto de datos como grafo conexo o no; todos estos procesos necesarios para tener los datos de forma concordante y poder realizar la clusterización teniendo en cuenta los cliques con los valores máximos en las funciones objetivo seleccionadas.

Respecto de los resultados de la clusterización obtenidos se rescata que los cliques que se han generado, agrupan especímenes de la familia coronaviridae por la similaridad de sus proteínas, esto basado en la previa búsqueda de los BBH. Una característica para resaltar del modelo es que se tiene en cuenta el proceso previo a la clusterización, es decir la obtención de los datos, su alistamiento y su entrada en la metaheurística, creando un proceso que mantiene la integridad de los datos y su tratamiento.

Dentro de los aportes que se desprenden del proyecto, se puede resaltar la automatización de la adquisición de datos desde los repositorios del NCBI, esto abre la puerta a la aplicación de la metodología de análisis de datos ómicos para otros organismos como hongos, bacterias, virus, líquenes entre otros, reduciendo los errores que se puedan causar por la realización de estas fases de obtención y alistamiento de forma manual, también permite que se descarguen mayor cantidad de datos en menor tiempo.

Las aportaciones que se han decantado del presente proyecto van más allá de los resultados, que se han obtenido. El potencial de este proyecto ha valido para la construcción de un prototipo de los módulos de adquisición y construcción de la base de datos BLAST; la aprobación de un miniproyecto para llevar la metodología de análisis de datos ómicos a la web y convertirla en una plataforma que pueda ser accedida en línea y permita nuevos estudios con diversos organismos.

Adicional a desarrollos e integraciones que puedan devenir de este proyecto, se extienden futuros trabajos que hagan una identificación de los clusters de proteínas obtenidos desde una perspectiva del mundo biológico, ya que podrían aportar información en la búsqueda de defender al ser humano de los diferentes integrantes de la familia coronaviridae.

Enunciar algunos de los retos que se presentaron en el proceso de desarrollo del proyecto es una aportación que no se debe desestimar; el primero fue la incursión en el contexto biológico de los virus, los cuales poseen una estructura proteómica reducida, esta característica computacionalmente es una ventaja, sin embargo, también se requiere de entender su forma de funcionamiento al no ser seres vivos y requerir un huésped para poder sobrevivir y replicarse. Lo segundo en el marco de los retos fue la selección de los datos, pues a pesar de existir más de 200 registros de organismos de esta familia, solo 68 se han curado, esta situación requirió tiempo hasta poner a punto el módulo de automatización de la adquisición. En tercer lugar definir un marco de trabajo para mantener ordenada y actualizada la información y los algoritmos que se aplicaron, en este caso se hizo uso de un servicio de control de versiones

(VCS) en la plataforma GitLab, que aporto el control de versiones de las diferentes piezas de software que se fueron generando, al mismo tiempo que la centralización de dichas piezas.

Glosario.

Aminoácido.	Moléculas base de las proteínas.
BBH.	Siglas para: Mejor Acierto Bidireccional (Bidirectional Best Hit).
BHT.	Siglas para: Mejor Acierto (Best Hit).
BLAST.	Siglas para Herramienta de Búsqueda de Alineamiento Local (Basic Local Allignment Search Tool).
Bot.	Abreviatura para Robot que hace una labor repetitiva de manera automatizada.
CDD.	Siglas para Base de Datos de Dominios Conservados (Conserved Domains Database).
Cluster.	Agrupación de datos.
CSV	Siglas para el formato de archivo de valores separados por comas (Comma Separated Value).
EDA.	Siglas para: Algoritmo de Estimación de Distribución (Estimation Distribution Algorithm).
Fasta.	Formato para documentos que contienen secuencias de información ómica.
FTP	Siglas para Protocolo de Transferencia de Archivos (File Transfer Protocol).
GitLab	Plataforma que ofrece servicio de almacenaje de repositorios de software.
MATEDA.	Librería para implementación de algoritmos EDA.
Metaheurística.	Procesos para búsqueda de soluciones aproximadas no exactas.
NCBI.	Siglas para Centro Nacional para la Información Biotecnológica (National Center for Biotechnology).

NIH	Siglas para Instituto Nacional de Salud (National Institutes of Health).
Proteína.	Sustancia química contenida en las células vivas compuesta por aminoácidos.
RefSeq.	Repositorio de información ómica.
VCS	Siglas para Sistema de Control de Versiones (Version Control System).

Bibliografía.

(87) COVID 19 CORONAVIRUS FISIOPATOLOGIA parte 1 - YouTube. (2020). https://www.youtube.com/watch?v=EL9lWBuhMOQ

Adiyaman, R., & McGuffin, L. J. (2019). Methods for the refinement of protein structure 3D models. *International Journal of Molecular Sciences*, *20*(9). https://doi.org/10.3390/ijms20092301

Allende, J. E. (1972). *Biosíntesis de proteínas y el código genético*. http://www.sidalc.net/cgi-bin/wxis.exe/?IsisScript=LIBROS.xis&method=post&formato=2&cantidad=1&expresion=mfn=002255

Altamiranda, J., Aguilar, J., & Hernández, L. (2008). Sistema de reconocimiento de patrones en bioinformática. *IFMBE Proceedings*, *18*, 573–577. https://doi.org/10.1007/978-3-540-74471-9_133

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2

Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J. L., Lozano, J. A., van de Peer, Y., Blanco, R., Robles, V., Bielza, C., & Larrañaga, P. (2008). *BioData Mining A review of estimation of distribution algorithms in bioinformatics*. https://doi.org/10.1186/1756-0381-1-6

Barreto Hernández, E. (2008). Bioinformática: una oportunidad y un desafío. *Revista Colombiana de Biotecnología*, *X*(1), 132–138.

Blank, J., & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Blank, J., & Deb, K. (2021). *PSAF: A Probabilistic Surrogate-Assisted Framework for Single-Objective Optimization*. https://doi.org/10.1145/3449639.3459297

BLAST: Basic Local Alignment Search Tool. (2016). https://blast.ncbi.nlm.nih.gov/Blast.cgi

Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, *35*(3), 268–308. https://doi.org/10.1145/937503.937505

Bomze, I. M., Budinich, M., Pardalos, P. M., & Pelillo, M. (1999). The Maximum Clique Problem. *Handbook of Combinatorial Optimization*, 1–74. https://doi.org/10.1007/978-1-4757-3023-4_1

Camacho, C., Madden, T., Tao, T., Agarwala, R., & Morgulis, A. (2019). BLAST Command Line Applications User Manual [Internet]. *Bethesda (MD): National Center for Biotechnology*

Information (US), *MD*, 1–28. https://www.ncbi.nlm.nih.gov/books/NBK279690/pdf/Bookshelf_NBK279690.pdf

Chen, Y., Tang, S., Bouguila, N., Wang, C., Du, J., & Li, H. L. (2018). A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data. *Pattern Recognition*, *83*, 375–387. https://doi.org/10.1016/j.patcog.2018.05.030

Conserved Domains Database (CDD) and Resources. (2020). https://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml

de Antonio Suárez, O. (2011). Una aproximación a la heuristica y metaheuristicas. *INGE@UAN - TENDENCIAS EN LA INGENIERÍA*, *1*(2), 44–51. https://revistas.uan.edu.co/index.php/ingeuan/article/view/217

de Bonet, J. S., Isbell, C. L., & Viola, P. (1997). *MIMIC: Finding Optima by Estimating Probability Densities*.

Deb, K. (2011). *Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction*. http://www.iitk.ac.in/kangal/deb.htm

Download BLAST Software and Databases Documentation. (2008). https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download

Duarte Muñoz, A. (2007). *Metaheurísticas*. Dykinson. http://ebookcentral.proquest.com/lib/univeraguascalientessp/detail.action?docID=3175838

Elshaer, R., & Awad, H. (2020). A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. *Computers and Industrial Engineering*, *140*. https://doi.org/10.1016/j.cie.2019.106242

Enjuanes Sánchez, L., López de Diego, M., Luis Nieto-Torres, J., & Manuel Jiménez-Guardeño José Ángel Regla-Nava, J. (2011). Emergencia de virus. Evolución y protección frente al coronavirus de la neumonía atípica SARS-CoV. In V. Larraga (Ed.), *Emergencia de virus. Evolución y protección frente al coronavirus de la neumonía atípica SARS-CoV* (1st ed., pp. 47–63).

Española, R. A. (2014). *heurístico, ca*. https://dle.rae.es/heurístico

Esqueda, J. (2020). *Diseño e implementación de una metaheurística MOEDA para el agrupamiento de proteínas modelado a través del problema de clique máximo [Tesina. Universidad Autónoma de Aguascalientes]*. Universidad Autonoma de Aguascalientes.

Feig, M. (2017). Computational protein structure refinement: almost there, yet still so far to go. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *7*(3). https://doi.org/10.1002/wcms.1307

Fernández González, M. (2019). *APLICACIÓN CON ARQUITECTURA ORIENTADA A SERVICIOS PARA OPTIMIZACIÓN MONO-OBJETIVO BASADA*.

Galvis-Motoa, S.-I., Ponce-de-Leon, E., Martin, E. M., & Cuellar-Garrido, D. (2021). Acquisition and preprocessing of proteomic data for Bidirectional Best Hits Methodology: a study case in the coronaviridae family. *RESEARCH IN COMPUTING SCIENCE*, *150*(9).

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability : a guide to the theory of NP-completeness*. 338.

Geer, L. Y., Marchler-Bauer, A., Geer, R. C., Han, L., He, J., He, S., Liu, C., Shi, W., & Bryant, S. H. (2010). The NCBI BioSystems database. *Nucleic Acids Research*, *38*, D492–D496. https://doi.org/10.1093/nar/gkp858

GenBank Overview. (2013). https://www.ncbi.nlm.nih.gov/genbank/

Genome List - Genome - NCBI. (n.d.-a). Retrieved July 5, 2020, from https://www.ncbi.nlm.nih.gov/genome/browse#!/viruses/

Genome List - Genome - NCBI. (n.d.-b). Retrieved June 9, 2021, from https://www.ncbi.nlm.nih.gov/genome/browse#!/viruses/

Gertz, E. M. (2005). *BLAST scoring parameters*. https://wiki.ices.utexas.edu/clsb/export/1990/branches/loopp_dyndb/data/T_local/blastdb/linux/blast-2.2.18/doc/scoring.pdf

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, *13*(5), 533–549. https://doi.org/10.1016/0305-0548(86)90048-1

Gómez-Moreno Calerra, C. (2000). *Estructura de proteínas*. Editorial Ariel.

Harik, G. (1999). *Linkage Learning via Probabilistic Modeling in the ECGA*.

Hauschild, M., & Pelikan, M. (2011). *A Survey of Estimation of Distribution Algorithms*.

Hernández, A. (2009). *Aminoácidos y proteínas*. El Cid Editor | apuntes. http://ebookcentral.proquest.com/lib/univeraguascalientessp/detail.action?docID=3180894

Karp, R. M. (1972). REDUCIBILITY AMONG COMBINATORIAL PROBLEMS t. In Raymond E. Miller & Thatcher. J (Eds.), *Complexity of Computer Computations* (pp. 85–103). Springer US.

Kruskal, J. B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, *7*(1), 48. https://doi.org/10.2307/2033241

Kuri, A. (2000). *Algoritmos Genéticos*.

Lange, K. (2013). *Optimization* (Vol. 95). Springer New York. https://doi.org/10.1007/978-1-4614-5838-8

Letko, M., Munster, V., & Munster, D. V. (2020). *Functional assessment of cell entry and receptor usage for lineage B β-coronaviruses, including 2019-nCoV*. https://doi.org/10.1101/2020.01.22.915660

Madden, T. (2008). *Appendices - BLAST® Command Line Applications User Manual*. Bethesda (MD): National Center for Biotechnology Information (US). https://www.ncbi.nlm.nih.gov/books/NBK279684/

Melián, B., Moreno Pérez, J. A., & Moreno Vega, J. M. (2003). Metaheurísticas: una visión global. *Revista Iberoamericana de Inteligencia Artificial Asociación Española Para La Inteligencia Artificial*, *7*(19), 7–28. http://www.aepia.org/revista

Mendoza, A., Ponce de León Sentí, E., & Diaz Diaz, E. (2013). Técnicas Para Contrarrestar La Pérdida De Variabilidad En Un Umda. *Revista Vínculos*, *10*(1), 186–195. https://doi.org/10.14483/2322939X.4709

Mendoza-Gonzalez, A., Ponce-De-Leon, E., & Diaz-Diaz, E. (2013). Classification Scheme of Multi-objective Estimation of Distribution Algorithms. In *2013 IEEE Congress on Evolutionary Computation*. https://doi.org/10.1109/CEC.2013.6557941

Meza H., O., & Ortega F., M. (2004). *GRAFOS Y ALGORITMOS* (EDITORIAL EQUINOCCIO, Ed.; Segunda). Universidad Simón Bolívar.

Michel Fernández González, I., Ramón, C., & Sardiñas, Q. (2018). *REVISIÓN CONCISA SOBRE HEURÍSTICAS PARA OPTIMIZACIÓN MONO-OBJETIVO*. 978–959.

Mirjalili, S. (1920). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, *27*. https://doi.org/10.1007/s00521-015-1920-1

Nieto-Torres, J. L., Dediego, M. L., Verdiá-Báguena, C., Jimenez-Guardeñ O, J. M., & Regla-Nava, J. A. (2014). Severe Acute Respiratory Syndrome Coronavirus Envelope Protein Ion Channel Activity Promotes Virus Fitness and Pathogenesis. *PLoS Pathog*, *10*(5), 1004077. https://doi.org/10.1371/journal.ppat.1004077

Nieto-Torres, J. L., Verdiá-Báguena, C., Jimenez-Guardeño, J. M., Regla-Nava, J. A., Castaño-Rodriguez, C., Fernandez-Delgado, R., Torres, J., Aguilella, V. M., & Enjuanes, L. (2015).

Severe acute respiratory syndrome coronavirus E protein transports calcium ions and activates the NLRP3 inflammasome. *Virology*, *485*, 330–339. https://doi.org/10.1016/J.VIROL.2015.08.010

Oppenheimer, A. (2018). *¡Sálvese quien pueda!: El futuro del trabajo en la era de la automatización*. Vintage Espanol.

Ordoñez Guillén, N. E. (2014). *Heurística para el problema MAX-CLIQUE basada en un modelo de cómputo con ADN utilizando GPU's*. CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR DE ENSENADA.

Osaba, E., Villar-Rodriguez, E., del Ser, J., Nebro, A. J., Molina, D., LaTorre, A., Suganthan, P. N., Coello Coello, C. A., & Herrera, F. (2021). A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. *Swarm and Evolutionary Computation*, *64*, 100888. https://doi.org/10.1016/J.SWEVO.2021.100888

Osguthorpe, D. J. (2000). Ab initio protein folding. *Current Opinion in Structural Biology*, *10*(2), 146–152. https://doi.org/10.1016/S0959-440X(00)00067-1

Osman Elkin, L. (2003). Rosalind Franklin and the Double Helix. *Citation: Physics Today*, *56*, 42. https://doi.org/10.1063/1.1570771

Osman, I. H., & Kelly, J. P. (1996). Meta-Heuristics: An Overview. *Meta-Heuristics*, 1–21. https://doi.org/10.1007/978-1-4613-1361-8_1

Our Mission - NCBI. (1988). https://www.ncbi.nlm.nih.gov/home/about/mission/

Overbeek, R., Fonstein, M., Pusch, G. D., & Maltsev, N. (1999). *The use of gene clusters to infer functional coupling* (Vol. 96). www.pnas.org.

Pascual-Iglesias, A., Canton, J., Ortega-Prieto, A. M., Jimenez-Guardeño, J. M., & Angel Regla-Nava, J. (2021). *An Overview of Vaccines against SARS-CoV-2 in the COVID-19 Pandemic Era*. https://doi.org/10.3390/pathogens10081030

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). *BOA: The Bayesian Optimization Algorithm*.

Pelta, D. A. (2002). *Algoritmos heurísticos en bioinformática*. 182. http://hdl.handle.net/10481/24513

Pérez Castillo, J. N., Rojas Quintero, C. A., & Vera Parra, N. E. (2014). *Biopython básico : manual práctico*. Espacios.

Pessoa Ferreira Lima, T. de. (2013). *AN AUTOMATIC METHOD FOR CONSTRUCTION OF MULTICLASSIFIER SYSTEMS BASED ON THE COMBINATION OF SELECTION AND FUSION*. https://repositorio.ufpe.br/handle/123456789/12457

Ponce de León Sentí, E. E., Cuéllar Garrido, L. D., Martínez Guerra, J. J., Torres-Soto, A., Torres-Soto, M. D., Diaz Diaz, E., Herrera Ambriz, R., Gutierrez Romo, S. E., & Veloz Cleto, F. (2015).

PROCESAMIENTO DE DATOS PROTEÓMICOS PARA LA VALIDACIÓN DE UNA METODOLOGÍA DE CONSTRUCCIÓN DE ÁRBOLES FILOGENÉTICOS.

Ponce de León Sentí, E. E., Diaz Diaz, E., Martínez Guerra, J. J., Torres-Soto, M. D., Torres-Soto, A., Mendoza González, A., Arellano Cardona, R., Esparza, A., & Quezada, F. (2013). *PROBLEMA DE COMPARACIÓN DE PROTEOMAS DE HONGOS USANDO UNA MEDIDA DE SIMILARIDAD SOBRE LOS MEJORES ACIERTOS BIDIRECCIONALES*.

Ponce de León Sentí, E. E., Reyes Gallegos, J. E., Cuéllar Garrido, L. D., Martin Álvarez Tostado, E. M., Díaz Díaz, E., Torres Soto, A., Torres Soto, M. D., & Martínez Guerra, J. J. (2022). Metodología eficiente para obtener cliques de proteínas mediante los mejores aciertos bidireccionales+. In *UN ACERCAMIENTO A LA INVESTIGACIÓN MULTIDISCIPLINAR EN LA UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES* (1st ed., Vol. 1, pp. 73–82). Universidad Autónoma de Aguascalientes. https://doi.org/10.33064/UAA/978-607-8782-60-4

Ponce, J. C., de León, E. E. P., Padilla, A., Padilla, F., & Zezzatti, A. O. O. (2006). Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo con un Optimizador Local K-opt. *Hífen, Uruguaiana*, *30*(58), 191–196.

Ponce-de-Leon-Senti, E., Diaz, E., Guardado-Muro, H., Cuellar-Garrido, D., Martinez-Guerra, J. J., Torres-Soto, A., Torres-Soto, D., & Hernandez-Aguirre, A. (2017). A Distance Measure for Building Phylogenetic Trees: A First Approach. In *Research in Computing Science* (Vol. 139, Issue 1). https://doi.org/10.13053/rcs-139-1-12

Ponce-de-Leon-Senti, E. E., Reyes-Gallegos, J. E., Cuellar-Garrido, L. D., Martin, E. M., Díaz Díaz, E., Torres-Soto, A., Torres-Soto, M. D., & Martinez-Guerra, J. J. (2020). METODOLOGÍA EFICIENTE PARA OBTENER CLIQUES DE PROTEÍNAS MEDIANTE LOS MEJORES ACIERTOS BIDIRECCIONALES. *Memorias Del 21 Seminario de Investigación de La Universidad Autónoma de Aguascalientes, in-Press.*, 13.

Rabi, F. A., al Zoubi, M. S., Al-Nasser, A. D., Kasasbeh, G. A., & Salameh, D. M. (2020). Sars-cov-2 and coronavirus disease 2019: What we know so far. In *Pathogens* (Vol. 9, Issue 3). MDPI AG. https://doi.org/10.3390/pathogens9030231

RefSeq: NCBI Reference Sequence Database. (2018). https://www.ncbi.nlm.nih.gov/refseq/

Reyes-Gallegos, J. E. (2019). *ALGORITMO EFICIENTE PARA LA AGRUPACIÓN DE PROTEÍNAS EN FAMILIAS BASADO EN MEJORES ACIERTOS BIDIRECCIONALES Y EL ÁRBOL FILOGENÉTICO [Tesina. Universidad Autónoma de Aguascalientes]* [Thesis Dissertation]. Universidad Autonoma de Aguascalientes.

Rincón Miranda, J. I. (2016). *Clusterización de hongos mediante metaheurísticas híbridas a partir de su información proteómica* [Universidad Autónoma de Aguascalientes].

http://bdigital.dgse.uaa.mx:8080/xmlui/bitstream/handle/11317/884/410351.pdf?sequence=1&isAllowed=y

Roldán Martínez, D. (2015). *Bioinformática : el ADN a un solo clic* [Book]. Ra-Ma.

Rothlauf, F. (2011). Design of Modern Heuristics. In *Natural Computing Series* (Vol. 25). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72962-4

Santana, R., Bielza, C., Larrañaga, P., Lozano, J. A., Echegoyen, C., Mendiburu, A., Armañanzas, R., & Shakya, S. (2010). Mateda-2.0: Estimation of Distribution Algorithms in MATLAB. *Journal of Statistical Software*, *35*(7), 1–30. http://www.jstatsoft.org/v35/i07

Santana, R., Echegoyen, C., Mendiburu, A., Bielza, C., Lozano, J. A., Larrañaga, P., Armañanzas, R., & Shakya, S. (2009). *MATEDA: A suite of EDA programs in Matlab* (Issue EHU-KZAA-IK-2/09). http://hdl.handle.net/10810/4622

TED Talk. (2019). David Baker: 5 challenges we could solve by designing new proteins. In *TED2019*. https://www.ted.com/talks/david_baker_5_challenges_we_could_solve_by_designing_new_proteins

Torres-Jiménez, J., & Pavón, J. (2014). Applications of metaheuristics in real-life problems. *Prog Artif Intell*, *2*, 175–176. https://doi.org/10.1007/s13748-014-0051-8

Wang, H., Xu, Z., Gao, L., & Hao, B. (2009). *A fungal phylogeny based on 82 complete genomes using the composition vector method*. https://doi.org/10.1186/1471-2148-9-195

Wang, J. P., & Tong, Q. (2009). Urban planning decision using multi-objective optimization algorithm. *2009 Second ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM 2009*, *4*, 392–394. https://doi.org/10.1109/CCCM.2009.5267600

Watson, J. D., & Crick, F. H. C. (1953). A Structure for Deoxyribose Nucleic Acid. *Nature*, *171*(1), 737–738.

Xu, S., & Li, Y. (2020). Beware of the second wave of COVID-19. *The Lancet*, *395*, 1321–1322. https://doi.org/10.1016/S0140-6736(20)30845-X

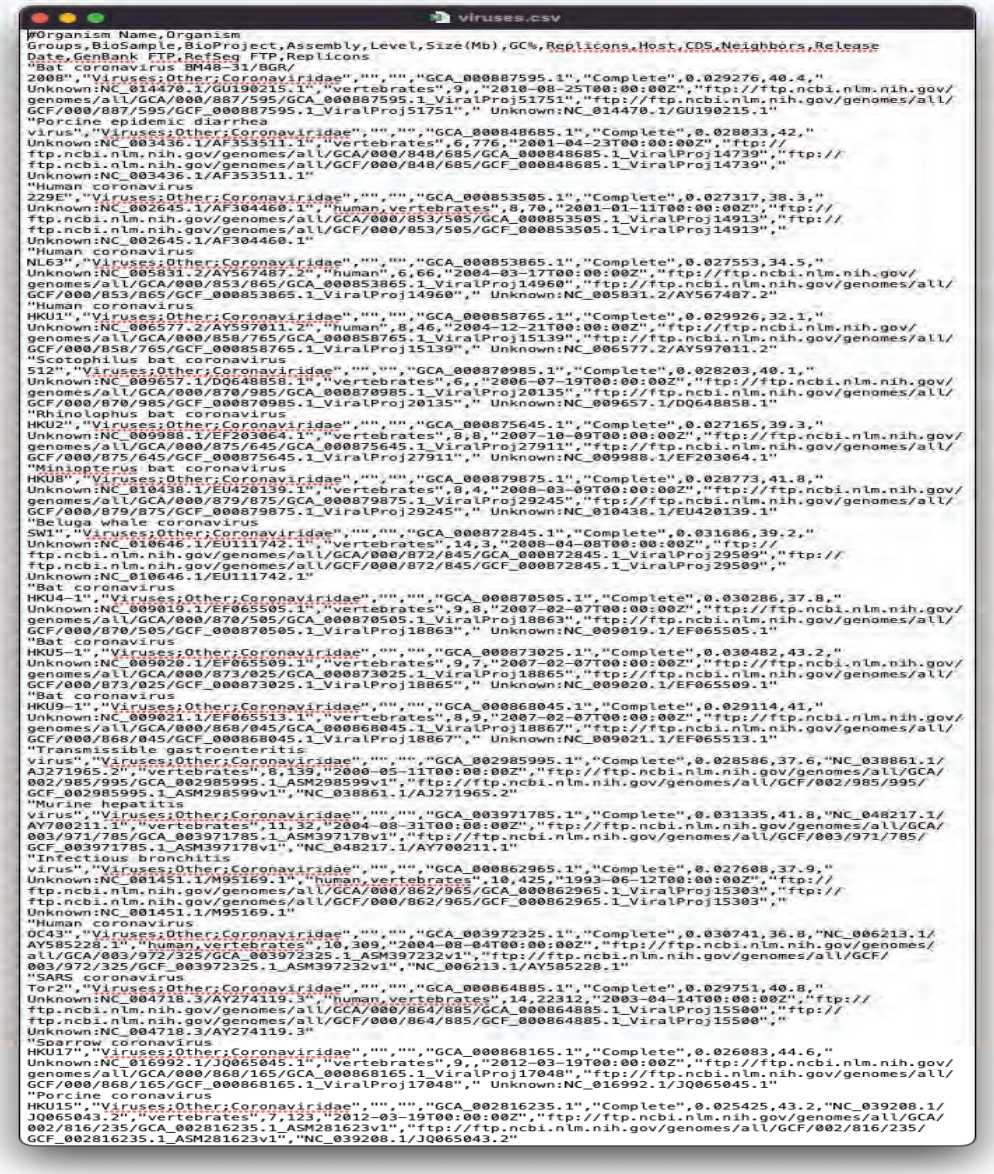
Yuan, B., Orlowska, M., & Sadiq, S. (2007). Finding the Optimal Path in 3D Spaces Using EDAs – The Wireless Sensor Networks Scenario. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4431 LNCS*(PART 1), 536–545. https://doi.org/10.1007/978-3-540-71618-1_59

Anexos.

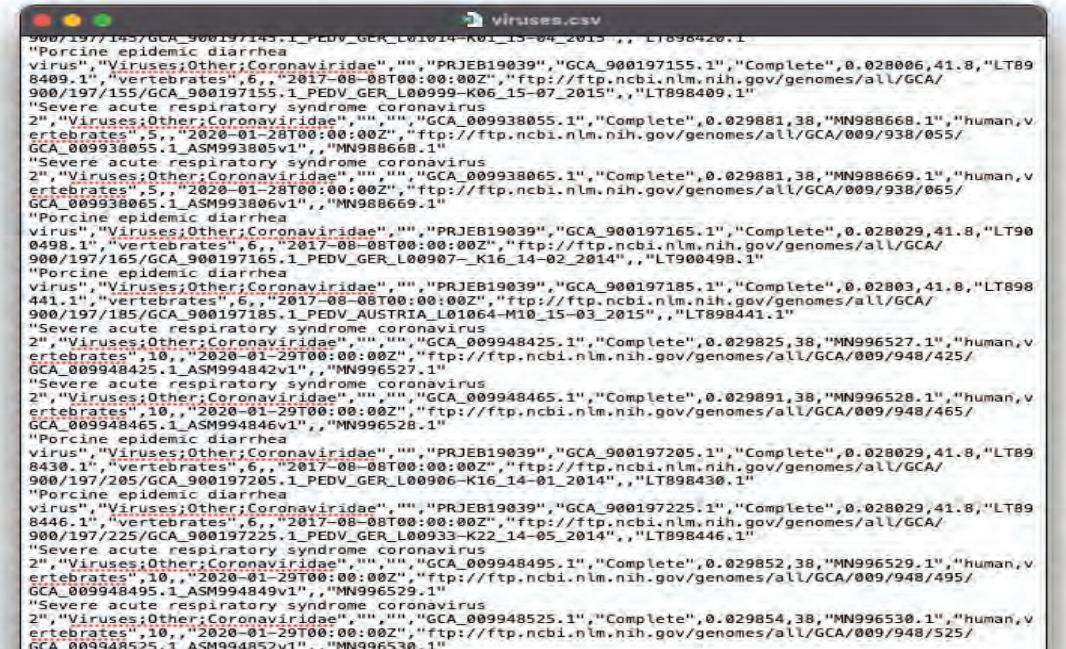
En esta sección se presentan anexos que complementan el entendimiento del proyecto.

Anexo A.

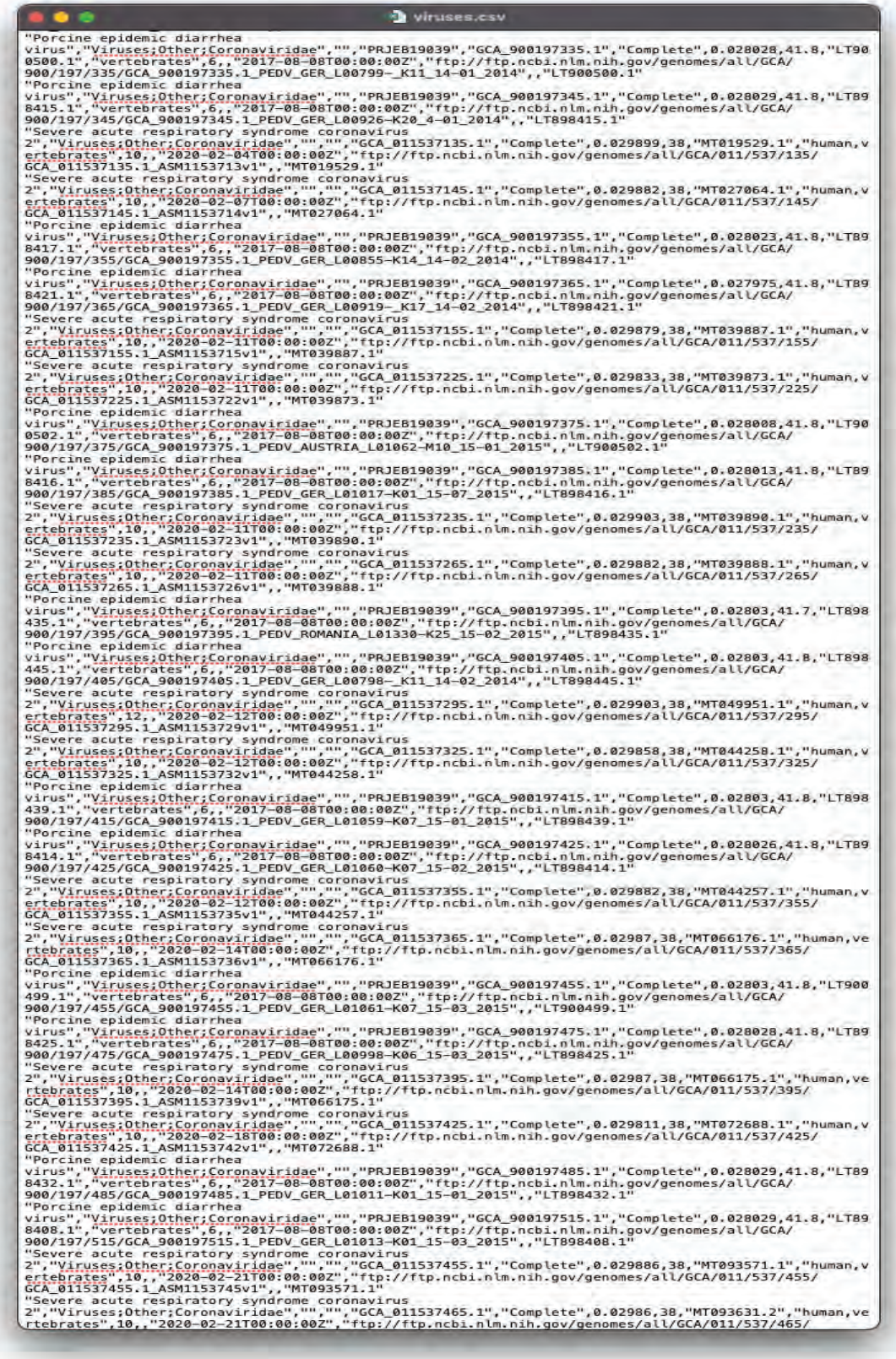
Fichero fuente CSV con el listado de direcciones de los organismos reportados en NCBI.

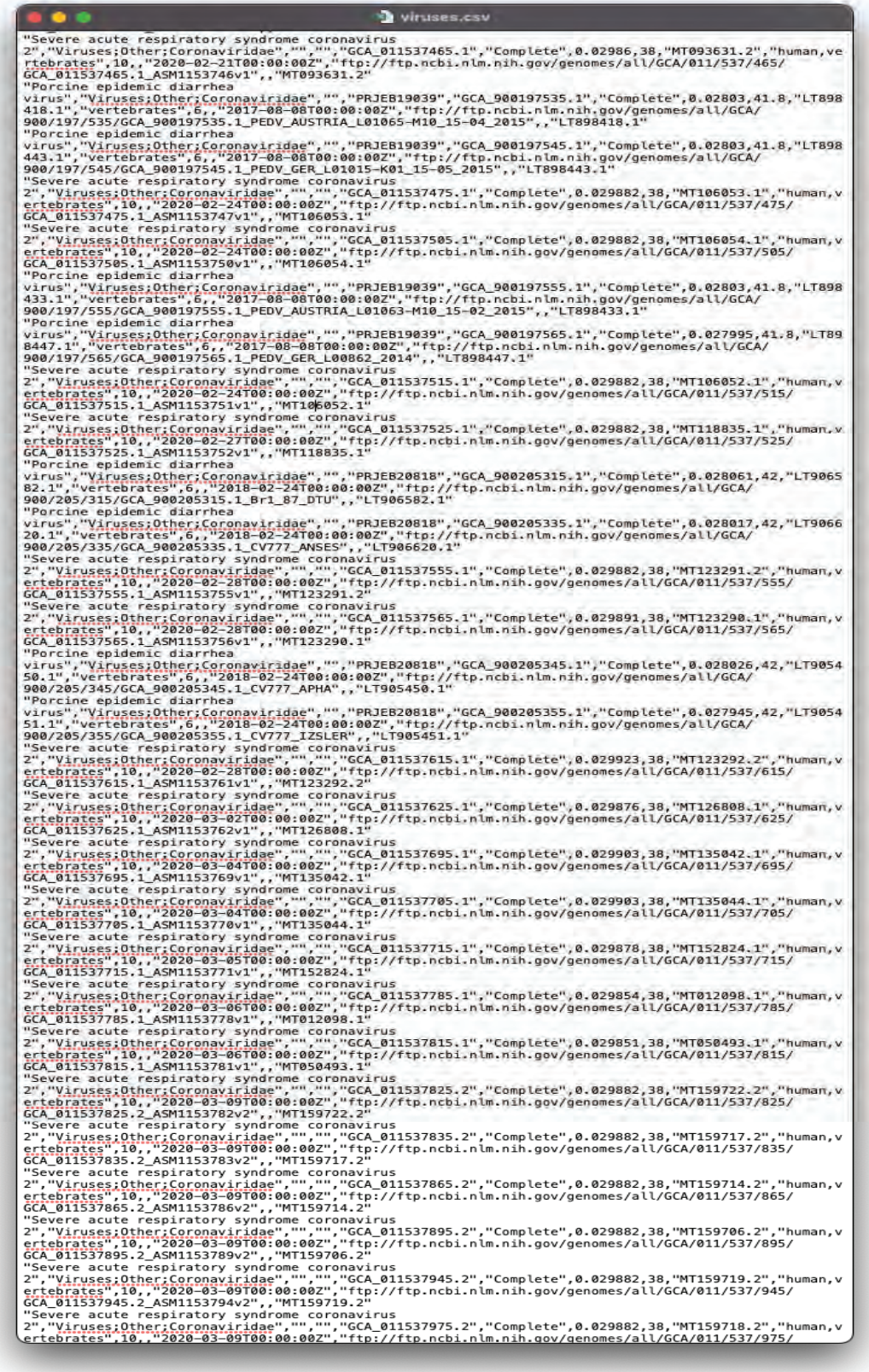

viruses.csv

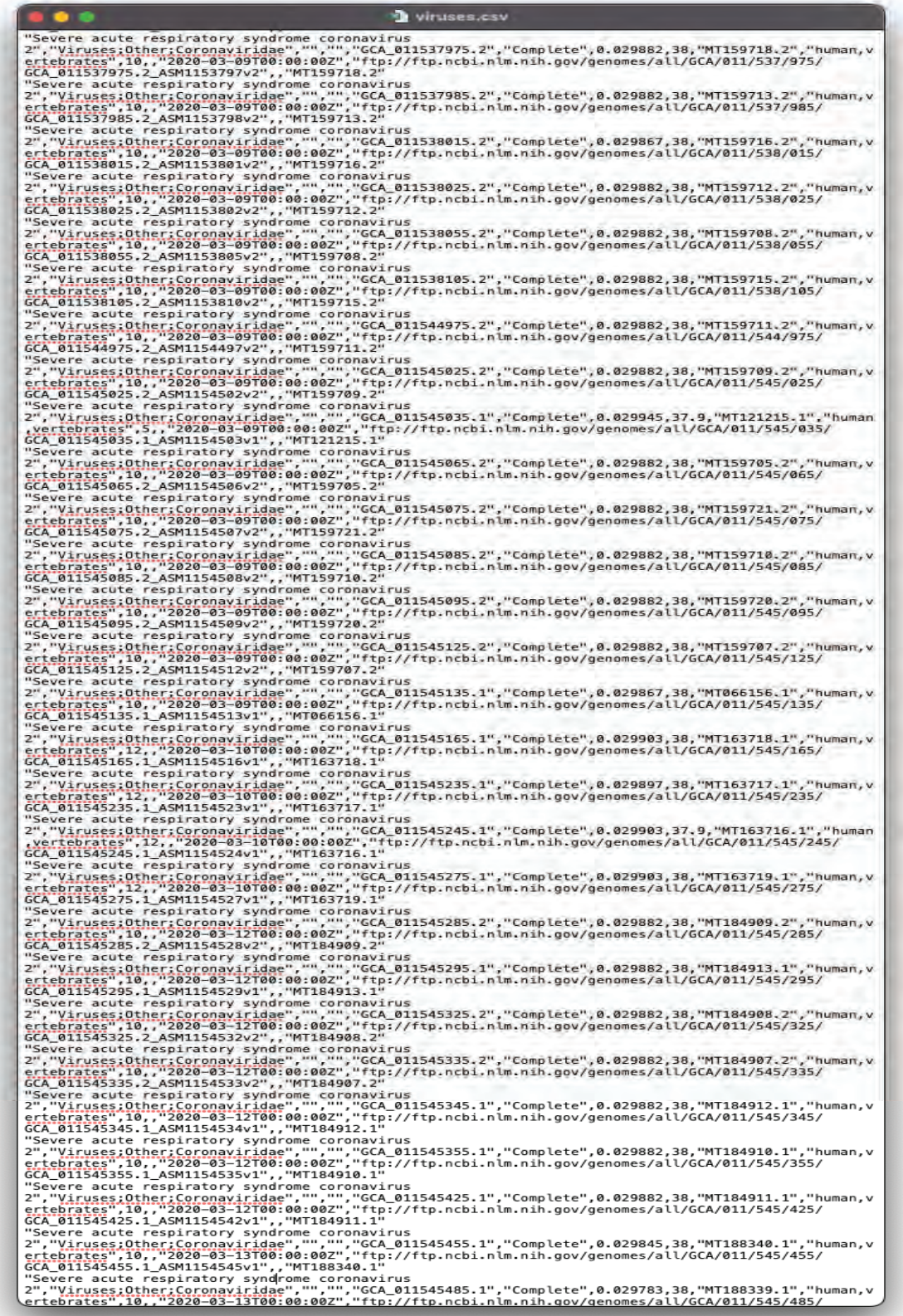
```
#Organism Name,Organism
Groups,BioSample,BioProject,Assembly,Level,Size(Mb),GC%,Replicons,Host,CDS,Neighbors,Release
Date,GenBank FTP,RefSeq FTP,Replicons
"Bat coronavirus BM48-31/BGR/
2008","Viruses;Other;Coronaviridae","","","GCA_000887595.1","Complete",0.029276,40.4,"
Unknown:NC_014470.1/GU190215.1","vertebrates",9,,"2010-08-25T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/887/595/GCA_000887595.1_ViralProj51751","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/887/595/GCF_000887595.1_ViralProj51751"," Unknown:NC_014470.1/GU190215.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","","GCA_000848685.1","Complete",0.028833,42,"
Unknown:NC_003436.1/AF353511.1","vertebrates",6,776,"2001-04-23T00:00:00Z","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/848/685/GCA_000848685.1_ViralProj14739","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/848/685/GCF_000848685.1_ViralProj14739","
Unknown:NC_003436.1/AF353511.1"
"Human coronavirus
229E","Viruses;Other;Coronaviridae","","","GCA_000853505.1","Complete",0.027317,38.3,"
Unknown:NC_002645.1/AF304460.1","human,vertebrates",8,70,"2001-01-11T00:00:00Z","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/853/505/GCA_000853505.1_ViralProj14913","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/853/505/GCF_000853505.1_ViralProj14913","
Unknown:NC_002645.1/AF304460.1"
"Human coronavirus
NL63","Viruses;Other;Coronaviridae","","","GCA_000853865.1","Complete",0.027553,34.5,"
Unknown:NC_005831.2/AY567487.2","human",6,66,"2004-03-17T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/853/865/GCA_000853865.1_ViralProj14960","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/853/865/GCF_000853865.1_ViralProj14960"," Unknown:NC_005831.2/AY567487.2"
"Human coronavirus
HKU1","Viruses;Other;Coronaviridae","","","GCA_000858765.1","Complete",0.029926,32.1,"
Unknown:NC_006577.2/AY597011.2","human",8,46,"2004-12-21T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/858/765/GCA_000858765.1_ViralProj15139","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/858/765/GCF_000858765.1_ViralProj15139"," Unknown:NC_006577.2/AY597011.2"
"Scotophilus bat coronavirus
512","Viruses;Other;Coronaviridae","","","GCA_000870985.1","Complete",0.028203,40.1,"
Unknown:NC_009657.1/DQ648858.1","vertebrates",6,,"2006-07-19T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/870/985/GCA_000870985.1_ViralProj20135","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/870/985/GCF_000870985.1_ViralProj20135"," Unknown:NC_009657.1/DQ648858.1"
"Rhinolophus bat coronavirus
HKU2","Viruses;Other;Coronaviridae","","","GCA_000875645.1","Complete",0.027165,39.3,"
Unknown:NC_009988.1/EF203064.1","vertebrates",8,8,"2007-10-09T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/875/645/GCA_000875645.1_ViralProj27911","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/875/645/GCF_000875645.1_ViralProj27911"," Unknown:NC_009988.1/EF203064.1"
"Miniopterus bat coronavirus
HKU8","Viruses;Other;Coronaviridae","","","GCA_000879875.1","Complete",0.028773,41.8,"
Unknown:NC_010438.1/EU420139.1","","vertebrates",8,4,"2008-03-09T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/879/875/GCA_000879875.1_ViralProj29245","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/879/875/GCF_000879875.1_ViralProj29245"," Unknown:NC_010438.1/EU420139.1"
"Beluga whale coronavirus
SW1","Viruses;Other;Coronaviridae","","","GCA_000872845.1","Complete",0.031686,39.2,"
Unknown:NC_010646.1/EU111742.1","vertebrates",14,3,"2008-04-08T00:00:00Z","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/872/845/GCA_000872845.1_ViralProj29509","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/872/845/GCF_000872845.1_ViralProj29509","
Unknown:NC_010646.1/EU111742.1"
"Bat coronavirus
HKU4-1","Viruses;Other;Coronaviridae","","","GCA_000870505.1","Complete",0.030286,37.8,"
Unknown:NC_009019.1/EF065505.1","vertebrates",9,8,"2007-02-07T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/870/505/GCA_000870505.1_ViralProj18863","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/870/505/GCF_000870505.1_ViralProj18863"," Unknown:NC_009019.1/EF065505.1"
"Bat coronavirus
HKU5-1","Viruses;Other;Coronaviridae","","","GCA_000873025.1","Complete",0.030482,43.2,"
Unknown:NC_009020.1/EF065509.1","vertebrates",9,7,"2007-02-07T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/873/025/GCA_000873025.1_ViralProj18865","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/873/025/GCF_000873025.1_ViralProj18865"," Unknown:NC_009020.1/EF065509.1"
"Bat coronavirus
HKU9-1","Viruses;Other;Coronaviridae","","","GCA_000868045.1","Complete",0.029114,41,"
Unknown:NC_009021.1/EF065513.1","vertebrates",8,9,"2007-02-07T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/868/045/GCA_000868045.1_ViralProj18867","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/868/045/GCF_000868045.1_ViralProj18867"," Unknown:NC_009021.1/EF065513.1"
"Transmissible gastroenteritis
virus","Viruses;Other;Coronaviridae","","","GCA_002985995.1","Complete",0.028586,37.6,"NC_038861.1/
AJ271965.2","vertebrates",8,139,"2000-05-11T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
002/985/995/GCA_002985995.1_ASM298599v1","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/002/985/995/
GCF_002985995.1_ASM298599v1","NC_038861.1/AJ271965.2"
"Murine hepatitis
virus","Viruses;Other;Coronaviridae","","","GCA_003971785.1","Complete",0.031335,41.8,"NC_048217.1/
AY700211.1","vertebrates",11,32,"2004-08-31T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
003/971/785/GCA_003971785.1_ASM397178v1","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/003/971/785/
GCF_003971785.1_ASM397178v1","NC_048217.1/AY700211.1"
"Infectious bronchitis
virus","Viruses;Other;Coronaviridae","","","GCA_000862965.1","Complete",0.027608,37.9,"
Unknown:NC_001451.1/M95169.1","human,vertebrates",10,425,"1993-06-12T00:00:00Z","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/862/965/GCA_000862965.1_ViralProj15303","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/862/965/GCF_000862965.1_ViralProj15303","
Unknown:NC_001451.1/M95169.1"
"Human coronavirus
OC43","Viruses;Other;Coronaviridae","","","GCA_003972325.1","Complete",0.030741,36.8,"NC_006213.1/
AY585228.1","human,vertebrates",10,309,"2004-08-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/
all/GCA/003/972/325/GCA_003972325.1_ASM397232v1","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/
003/972/325/GCF_003972325.1_ASM397232v1","NC_006213.1/AY585228.1"
"SARS coronavirus
Tor2","Viruses;Other;Coronaviridae","","","GCA_000864885.1","Complete",0.029751,40.8,"
Unknown:NC_004718.3/AY274119.3","human,vertebrates",14,22312,"2003-04-14T00:00:00Z","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/864/885/GCA_000864885.1_ViralProj15500","ftp://
ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/864/885/GCF_000864885.1_ViralProj15500","
Unknown:NC_004718.3/AY274119.3"
"Sparrow coronavirus
HKU17","Viruses;Other;Coronaviridae","","","GCA_000868165.1","Complete",0.026083,44.6,"
Unknown:NC_016992.1/JQ065045.1","vertebrates",9,,"2012-03-19T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/
genomes/all/GCA/000/868/165/GCA_000868165.1_ViralProj17048","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/
GCF/000/868/165/GCF_000868165.1_ViralProj17048"," Unknown:NC_016992.1/JQ065045.1"
"Porcine coronavirus
HKU15","Viruses;Other;Coronaviridae","","","GCA_002816235.1","Complete",0.025425,43.2,"NC_039208.1/
JQ065043.2","vertebrates",7,123,"2012-03-19T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
002/816/235/GCA_002816235.1_ASM281623v1","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/002/816/235/
GCF_002816235.1_ASM281623v1","NC_039208.1/JQ065043.2"
```

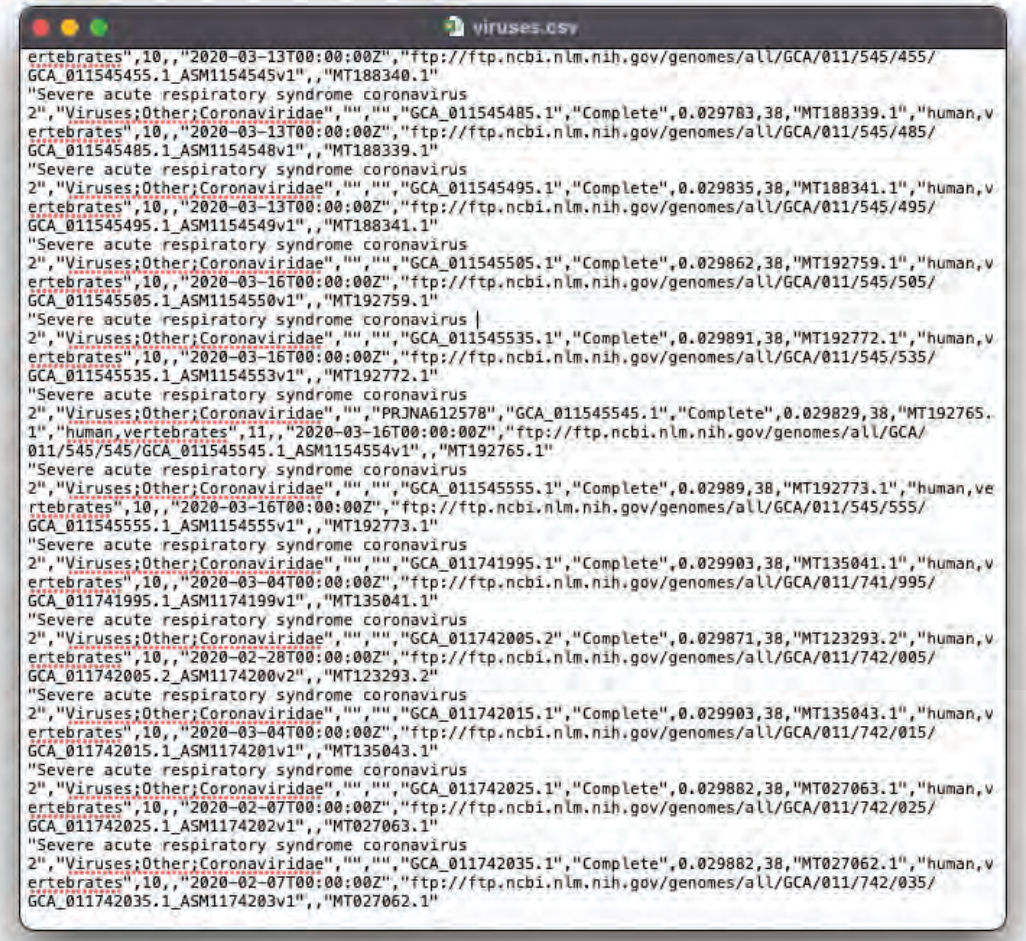

viruses.csv

```
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197155.1","Complete",0.028006,41.8,"LT89
8409.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/155/GCA_900197155.1_PEDV_GER_L00999-K06_15-07_2015",,"LT898409.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009938055.1","Complete",0.029881,38,"MN988668.1","human,v
ertebrates",5,,"2020-01-28T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/938/055/
GCA_009938055.1_ASM993805v1",,"MN988668.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009938065.1","Complete",0.029881,38,"MN988669.1","human,v
ertebrates",5,,"2020-01-28T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/938/065/
GCA_009938065.1_ASM993806v1",,"MN988669.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197165.1","Complete",0.028029,41.8,"LT90
0498.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/165/GCA_900197165.1_PEDV_GER_L00907-_K16_14-02_2014",,"LT900498.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197185.1","Complete",0.02803,41.8,"LT898
441.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/185/GCA_900197185.1_PEDV_AUSTRIA_L01064-M10_15-03_2015",,"LT898441.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009948425.1","Complete",0.029825,38,"MN996527.1","human,v
ertebrates",10,,"2020-01-29T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/948/425/
GCA_009948425.1_ASM994842v1",,"MN996527.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009948465.1","Complete",0.029891,38,"MN996528.1","human,v
ertebrates",10,,"2020-01-29T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/948/465/
GCA_009948465.1_ASM994846v1",,"MN996528.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197205.1","Complete",0.028029,41.8,"LT89
8430.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/205/GCA_900197205.1_PEDV_GER_L00906-K16_14-01_2014",,"LT898430.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197225.1","Complete",0.028029,41.8,"LT89
8446.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/225/GCA_900197225.1_PEDV_GER_L00933-K22_14-05_2014",,"LT898446.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009948495.1","Complete",0.029852,38,"MN996529.1","human,v
ertebrates",10,,"2020-01-29T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/948/495/
GCA_009948495.1_ASM994849v1",,"MN996529.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009948525.1","Complete",0.029854,38,"MN996530.1","human,v
ertebrates",10,,"2020-01-29T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/948/525/
GCA_009948525.1_ASM994852v1",,"MN996530.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197235.1","Complete",0.02803,41.8,"LT898
411.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/235/GCA_900197235.1_PEDV_GER_L01018-K01_15-08_2015",,"LT898411.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197245.1","Complete",0.02803,41.8,"LT898
413.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/245/GCA_900197245.1_PEDV_GER_L01020-K01_15-10_2015",,"LT898413.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_009948555.1","Complete",0.029857,38,"MN996531.1","human,v
ertebrates",10,,"2020-01-29T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/009/948/555/
GCA_009948555.1_ASM994855v1",,"MN996531.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","PRJNA613958","GCA_011536935.1","Complete",0.029893,38,"MT007544.
1","human,vertebrates",12,,"2020-01-31T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
011/536/935/GCA_011536935.1_ASM1153693v1",,"MT007544.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197275.1","Complete",0.027981,41.7,"LT89
8436.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/275/GCA_900197275.1_PEDV_ROMANIA_L01329-K25_15-01_2015",,"LT898436.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197285.1","Complete",0.027977,41.8,"LT89
8431.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/285/GCA_900197285.1_PEDV_GER_L00918-K17_14-01_2014",,"LT898431.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_011536975.1","Complete",0.029883,38,"MT019533.1","human,v
ertebrates",10,,"2020-02-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/536/975/
GCA_011536975.1_ASM1153697v1",,"MT019533.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_011537005.1","Complete",0.029899,38,"MT019531.1","human,v
ertebrates",10,,"2020-02-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/537/005/
GCA_011537005.1_ASM1153700v1",,"MT019531.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197295.1","Complete",0.02803,41.8,"LT898
444.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/295/GCA_900197295.1_PEDV_GER_L01420-K06_15-04_2015",,"LT898444.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197305.1","Complete",0.028029,41.8,"LT89
8410.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/305/GCA_900197305.1_PEDV_GER_L00908-K16_14-03_2014",,"LT898410.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_011537015.1","Complete",0.02989,38,"MT019532.1","human,ve
rtebrates",10,,"2020-02-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/537/015/
GCA_011537015.1_ASM1153701v1",,"MT019532.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_011537065.1","Complete",0.029889,38,"MT019530.1","human,v
ertebrates",10,,"2020-02-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/537/065/
GCA_011537065.1_ASM1153706v1",,"MT019530.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197315.1","Complete",0.02803,41.8,"LT898
423.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/315/GCA_900197315.1_PEDV_GER_L01012_K01_15-02_2015",,"LT898423.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197325.1","Complete",0.02803,41.8,"LT898
412.1","vertebrates",6,,"2017-08-08T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
900/197/325/GCA_900197325.1_PEDV_GER_L01019-K01_15-09_2015",,"LT898412.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_011537075.1","Complete",0.029882,38,"MT020881.1","human,v
ertebrates",10,,"2020-02-05T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/537/075/
GCA_011537075.1_ASM1153707v1",,"MT020881.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","GCA_011537085.1","Complete",0.029882,38,"MT020880.1","human,v
ertebrates",10,,"2020-02-05T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/537/085/
GCA_011537085.1_ASM1153708v1",,"MT020880.1"
"Porcine epidemic diarrhea
virus","Viruses;Other;Coronaviridae","","PRJEB19039","GCA_900197335.1","Complete",0.028028,41.8,"LT90
```









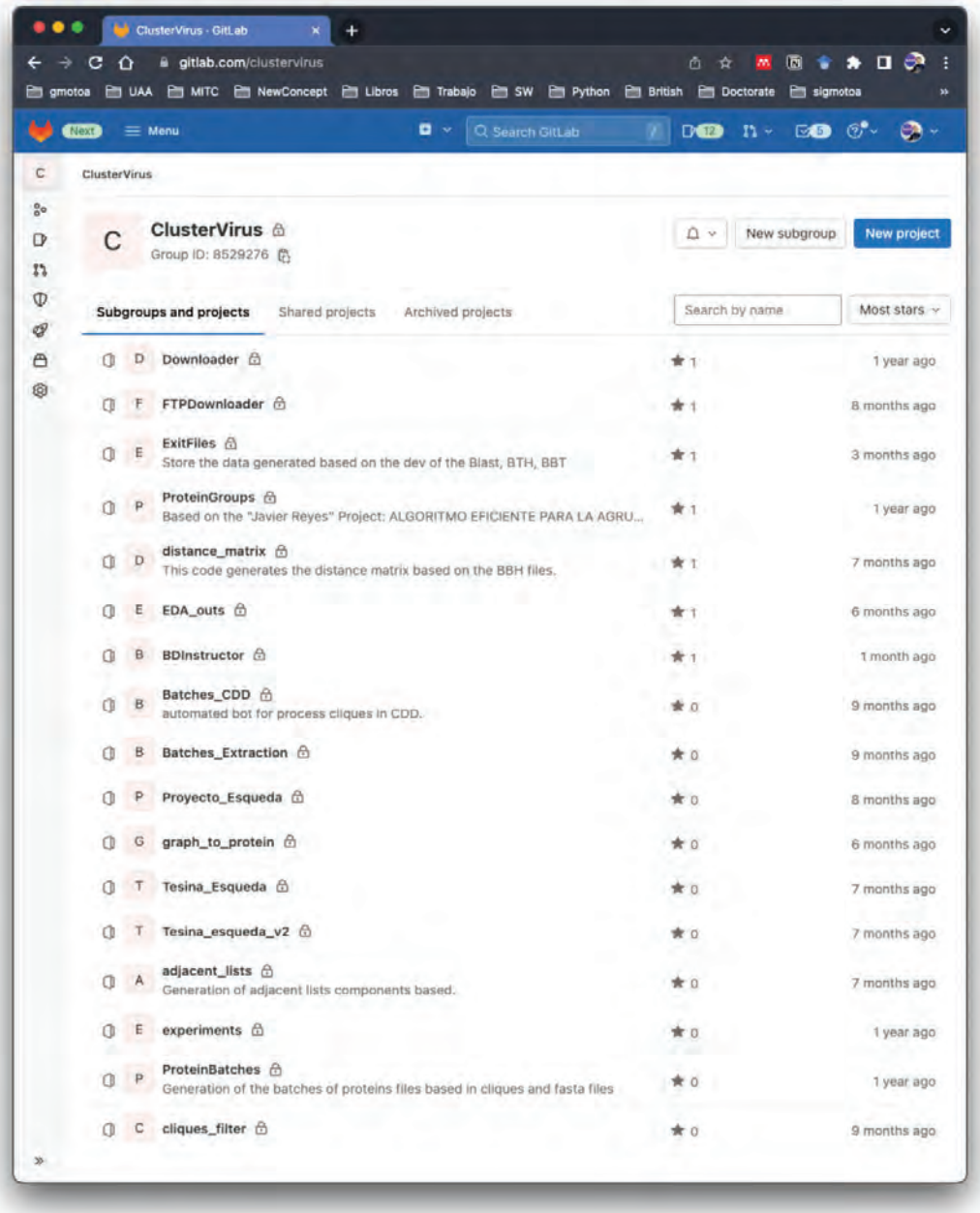
viruses.csv

```
ertebrates",10,,"2020-03-13T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/545/455/
GCA_011545455.1_ASM1154545v1",,"MT188340.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011545485.1","Complete",0.029783,38,"MT188339.1","human,v
ertebrates",10,,"2020-03-13T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/545/485/
GCA_011545485.1_ASM1154548v1",,"MT188339.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011545495.1","Complete",0.029835,38,"MT188341.1","human,v
ertebrates",10,,"2020-03-13T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/545/495/
GCA_011545495.1_ASM1154549v1",,"MT188341.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011545505.1","Complete",0.029862,38,"MT192759.1","human,v
ertebrates",10,,"2020-03-16T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/545/505/
GCA_011545505.1_ASM1154550v1",,"MT192759.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011545535.1","Complete",0.029891,38,"MT192772.1","human,v
ertebrates",10,,"2020-03-16T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/545/535/
GCA_011545535.1_ASM1154553v1",,"MT192772.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","PRJNA612578","GCA_011545545.1","Complete",0.029829,38,"MT192765.
1","human,vertebrates",11,,"2020-03-16T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/
011/545/545/GCA_011545545.1_ASM1154554v1",,"MT192765.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011545555.1","Complete",0.02989,38,"MT192773.1","human,ve
rtebrates",10,,"2020-03-16T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/545/555/
GCA_011545555.1_ASM1154555v1",,"MT192773.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011741995.1","Complete",0.029903,38,"MT135041.1","human,v
ertebrates",10,,"2020-03-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/741/995/
GCA_011741995.1_ASM1174199v1",,"MT135041.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011742005.2","Complete",0.029871,38,"MT123293.2","human,v
ertebrates",10,,"2020-02-28T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/742/005/
GCA_011742005.2_ASM1174200v2",,"MT123293.2"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011742015.1","Complete",0.029903,38,"MT135043.1","human,v
ertebrates",10,,"2020-03-04T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/742/015/
GCA_011742015.1_ASM1174201v1",,"MT135043.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011742025.1","Complete",0.029882,38,"MT027063.1","human,v
ertebrates",10,,"2020-02-07T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/742/025/
GCA_011742025.1_ASM1174202v1",,"MT027063.1"
"Severe acute respiratory syndrome coronavirus
2","Viruses;Other;Coronaviridae","","","GCA_011742035.1","Complete",0.029882,38,"MT027062.1","human,v
ertebrates",10,,"2020-02-07T00:00:00Z","ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/011/742/035/
GCA_011742035.1_ASM1174203v1",,"MT027062.1"
```

Anexo B.

Este anexo contiene el enlace para la solicitud de acceso al repositorio en GitLab en el que se encuentran las piezas de código empleadas para el proyecto.

https://gitlab.com/clustervirus



Anexo C.

Ponencia realizada en Colombia en el "Congreso Internacional de Ingeniería y Tecnologías de la Información en la Virtualidad. Mayo 2020 "

UNINPAHU
INSTITUCIÓN UNIVERSITARIA

FUNDACIÓN UNIVERSITARIA PARA EL DESARROLLO HUMANO
Facultad de Ingeniería y Tecnologías de la Información

Certi can que

SERGIO IVÁN GALVIS MOTOA

Asistió como conferencista al

CONGRESO INTERNACIONAL DE INGENIERÍA Y TECNOLOGÍAS DE LA INFORMACIÓN EN LA VIRTUALIDAD

Realizado los días 9 y 16 de mayo de 2020, en la Fundación Universitaria para el Desarrollo Humano UNINPAHU

CONFERENCIA:
Metaheurísticas como herramientas en la Bioinformática

En constancia se expide el presente certi cado.

María Fernanda Díaz Hernández
Decana Facultad de Ingeniería y Tecnologías de la Información

Juan Pablo Calixto Niño
Vicerrector Académico

Bogotá D.C, mayo de 2020

Anexo D.

Participación en modalidad ponencia en el "11er Congreso Internacional La Investigación en el Posgrado. Octubre 2021"

Anexo E.

Participación en el "Mexican International Conference on Artificial Intelligence, MICAI 2021".

The Mexican Society for Artificial Intelligence (SMIA)
and the Centro de Investigación en Computación del Instituto Politécnico Nacional

MICAI 2021

Award this certificate to

Sergio-Ivan Galvis-Motoa, Eunice Ponce-de-Leon, Eduardo M Martín and Daniel Cuellar-Garrido

for presentation of the paper entitled

Acquisition and preprocessing of proteomic data for Bidirectional Best Hits Methodology: a study case in the coronaviridae family

at the 20th Mexican International Conference on Artificial Intelligence, MICAI 2021.
Mexico City, Mexico, October 25 - 30, 2021.

Dr. Félix Castro Espinoza
SMIA President

Dr. Ildar Batyrshin
Program Chair

Dr. Alexander Gelbukh
Program Chair

Dr. Grigori Sidorov
Program Chair

Anexo F.

Carta de aceptación del articulo: "Acquisition and preprocessing of proteomic data for Bidirectional Best Hits Methodology: a study case in the coronaviridae family", en la revista "Research in Computing Science".

RESEARCH IN COMPUTING SCIENCE
ISSN 1870-4069

Centro de Investigación en Computación, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz, s/n, Col. La Escalera, CP 07320, DF, México
Tel.: +52-55-5729 6000, ext. 56518, 56653
http://www.rcs.cic.ipn.mx

Mexico City, September 1st, 2021

To whom it may concern:

Hereby I confirm that the paper

"Acquisition and preprocessing of proteomic data for Bidirectional Best Hits Methodology: a study case in the coronaviridae family"

by Sergio-Ivan Galvis-Motoa, Eunice Ponce-de-Leon, Eduardo M Martín and Daniel Cuellar-Garrido.

after thorough reviewing process is accepted for publication in our journal.

It is scheduled for the volume 150(9), 2021, which is now in the process of technical production.

With best regards,

……..
Dr. Grigori Sidorov
Editor-in-Chief

Anexo G.

Dirección de la tesina: "Integración de los módulos de pre procesamiento en la metodología basada en mejores aciertos bidireccionales: Software UAA-PROT"

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

Aguascalientes, Ags., a 28 de septiembre de 2021

EL DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN DE LA UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

OTORGA EL PRESENTE RECONOCIMIENTO A:

Sergio Iván Galvis Motoa

Por su apoyo como director del alumno **José Ramsés Moreno González** de la carrera de Ingeniero en Computación Inteligente en la tesina: **"Integración de los módulos de pre procesamiento en la metodología basada de mejores aciertos bidireccionales: Software UAA-PROT"**, realizada en el periodo enero-junio de 2021.

ATENTAMENTE
"Se Lumen Proferre"

DR. JULIO CESAR PONCE GALLEGOS
JEFE DEL DEPTO. DE CIENCIAS DE LA COMPUTACIÓN
CENTRO DE CIENCIAS BÁSICAS
UNIVERSIDAD AUTONOMA DE AGUASCALIENTES
Tel. 9 10 74 00 Ext. 51155

Anexo H.

Aceptación del miniproyecto: "Módulo de Procesamiento de Información Ómica sobre un Servidor Web".

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

SERGIO IVAN GALVIS MOTOA
INGENIERIA
CIENCIAS BÁSICAS
P R E S E N T E.

Por este conducto tengo el agrado de informarle que el Miniproyecto: **MP-22-063**, **Módulo de Procesamiento de Información Ómica sobre un Servidor Web,** ha sido:

APROBADO

Para participar en la convocatoria MINIPROYECTOS 2022 del Centro de Ciencias Básicas.

Así mismo, se le informa que el resultado final del miniproyecto se va a presentar, en formato digital o presencial en cartel, puede ser cualquier día entre el 28 al 30 de noviembre de 2022. Todos los profesores y alumnos participantes adquieren el compromiso de participar en este evento al momento de firmar la solicitud del proyecto.

Se le informa que debido a la **contingencia del SARS COV-2**, las fechas podrán ser sujetas a cambios, al igual que la forma de presentación.

Todos los participantes en los Miniproyectos que concluyan de manera satisfactoria recibirán una constancia digital que será enviada por la Secretaría de Investigación y Posgrado del Centro de Ciencias Básicas, Edificio 202 planta alta del 5 al 16 de diciembre del año en curso.

ATENTAMENTE
Aguascalientes, Ags. A 25 de marzo de 2022
"SE LUMEN PROFERRE"

M. EN C. JORGE MARTÍN ALFEREZ CHÁVEZ
DECANO DEL CENTRO DE CIENCIAS BÁSICAS.

DRA. HAYDÉE MARTINEZ RUVALCABA.
SECRETARIA DE INVESTIGACIÓN Y POSGRADO.
CENTRO DE CIENCIAS BÁSICAS.