Universitat Oberta de Catalunya

uoc.edu

Análisis filogenético de la óxido nitroso reductasa (*nosZ*): establecimiento del perfil de la proteína

Estudiante: David Lázaro Gimeno

Máster universitario en Bioinformática y bioestadística UOC-UB

Área 4

Consultor/a; Paloma María Pizarro Tobías

Nombre Profesor/a responsable de la asignatura: Antoni Pérez Navarro

Entrega: 06/2021

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Análisis filogenético de la óxido nitroso reductasa (*nosZ*): establecimiento del perfil de la proteína*
Nombre del autor:	*David Lázaro Gimeno*
Nombre del consultor/a:	*Paloma María Pizarro Tobías*
Nombre del PRA:	*Antoni Pérez Navarro*
Fecha de entrega (mm/aaaa):	05/2021
Titulación:	*Plan de estudios del estudiante*
Área del Trabajo Final:	*Trabajo final de máster*
Idioma del trabajo:	Castellano
Número de créditos:	15
Palabras clave	nosZ*, philogeny, protein modeling*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El objetivo del trabajo ha sido la realización de una revisión filogenética del gen *nosZ* en bacterias desnitrificantes existentes en la base de datos del NCBI. Para ello, en este estudio se han descargado y procesado mediante el uso de herramientas bioinformáticas disponibles, secuencias representantes de las especies debidamente anotadas una secuencia representante de este gen que pudiese ser asociada con la secuencia de interés.

Esta información ha sido la base para la realización de dos aproximaciones a un análisis filogenético. Por un lado se empleando herramientas de máxima verosimilitud a partir de secuencias de aminoácidos. Por otro lado, se ha realizado un análisis mediante análisis bayesiano de un subconjunto de secuencias de nucleótidos para las especies seleccionadas.

A partir de los resultados obtenidos se han elegido un representante de las agrupaciones establecidas en la bibliografía existente para realizar una predicción del perfil de la proteína.

Según los resultados obtenidos, ha sido posible obtener una topología para la filogenia con un amplio rango de representantes existentes en la base de datos pública a nivel taxonómico a nivel de especie frente a publicaciones previas.

Abstract (in English, 250 words or less):

The objective of the work has been to carry out a phylogenetic review of the *nosZ* gene in denitrifying bacteria existing in the NCBI database. For this, in this study, sequences representing the species duly annotated have been downloaded and processed using available bioinformatics tools, a representative sequence of this gene that could be associated with the sequence of interest.

This information has been the basis for carrying out two approaches to a phylogenetic analysis. On the one hand, maximum likelihood tools are used from amino acid sequences. On the other hand, an analysis has been carried out by means of Bayesian analysis of a subset of nucleotide sequences for the selected species.

From the results obtained, a representative of the groups established in the existing bibliography has been chosen to make a prediction of the protein profile.

According to the results obtained, it has been possible to obtain a topology for the phylogeny with a wide range of existing representatives in the public database at the taxonomic level at the species level compared to previous publications.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1 Contexto y justificación del trabajo.

1.1.1 Descripción general

El presente Trabajo Final de Máster (TFM) lleva por título “Análisis filogenético de la óxido nitroso reductasa (*nosZ*): establecimiento del perfil de la proteína”. La pregunta que se ha intentado responder era si: es posible obtener una filogenia actualizada para esta proteína en bacterias desnitrificantes, a partir de la información pública disponible en bases de datos públicas como es la del NCBI de una manera que pueda requerir menor supervisión por parte del investigador. Actualmente las bases de datos públicas disponen de numerosa información a nivel de secuencias de nucleótidos y proteínas para el gen de interés, si bien existe un elevado nivel de información duplicada, incompleta, o con errores arrastrados en las anotaciones que son fuente de dificultades a la hora de la elección de las secuencias y su uso posterior.

Por otra parte ha sido interesante evaluar a partir de la gran cantidad de información disponible qué nivel de predicción se puede obtener para el perfil de la proteína, dado que las bases de datos utilizadas para dichas predicciones requieren un gran esfuerzo de adecuación a la hora de incorporar nueva información, por lo que a pesar de toda la información existente toda no es aplicable para dichas predicciones.

Para darle valor añadido se ha evaluado la idoneidad en el uso de herramientas bioinformáticas que limiten o reduzcan la necesidad de la supervisión por parte del investigador para poder centrar los esfuerzos en identificar puntos clave en los que sí es necesaria su supervisión más detallada.

A continuación se enumeran los objetivos generales y específicos del TFM. Estos se consideran esenciales para el correcto desarrollo del trabajo, ya que sin ellos no cabríala posibilidad de culminarlo con éxito. Los objetivos generales engloban a su vez los objetivos específicos, más concretos, los cuales son necesarios para poder decir que el trabajo se habrá realizado con éxito.

1.2 Objetivos del Trabajo

Se estimaron tres objetivos generales, que cubren la mayor parte del trabajo, los cuales se consideraron esenciales para el correcto desarrollo del mismo, ya que de no haberse logrado alguno de los tres, el TFM no estaría completo. Los tres objetivos generales se engloban en ocho objetivos específicos, que desglosan de manera más precisa los objetivos generales a alcanzar. Si bien el tercer objetivo general inicialmente se estableció cronológicamente al final del proyecto, ha sido necesario tenerlo en cuenta y realizar adaptaciones durante todo el proyecto, ya que el uso y desarrollo de herramientas bioinformáticas estuvo en la esencia de todas las actividades realizadas.

1.2.1 . Objetivos generales:

• Realizar un análisis filogenético de las secuencias *nosZ* mediante máxima verosimilitud (ML) para secuencias de aminoácidos e inferencia bayesiana (Bayessian) en secuencias de nucleótidos.

• Establecer el perfil de la enzima óxido nitroso reductasa.

• Programación de scripts que automaticen partes de proceso.

1.2.2 Objetivos específicos:

• Descarga y filtrado de las secuencias correspondientes al gen *nosZ*.

• Identificación de secuencias y filtrado de información redundante, así como información que suponga contaminación o ruido.

• Alineamiento de las secuencias mediante algoritmos específicos para nucleótidos o aminoácidos.

• Evaluación de la existencia de señal filogenética.

• Realización de la filogenia mediante métodos de ML y Bayessian.

• Evaluación de las topologías obtenidas.

• Establecimiento del perfil de la proteína.

• Programación de script que automatice algunos de los pasos más significativos del proceso.

2. Enfoque y método a seguir:

Para poder contextualizar correctamente todo el enfoque seguido, se presenta inicialmente el workflow o flujo de trabajo, considerado para la correcta realización del estudio:

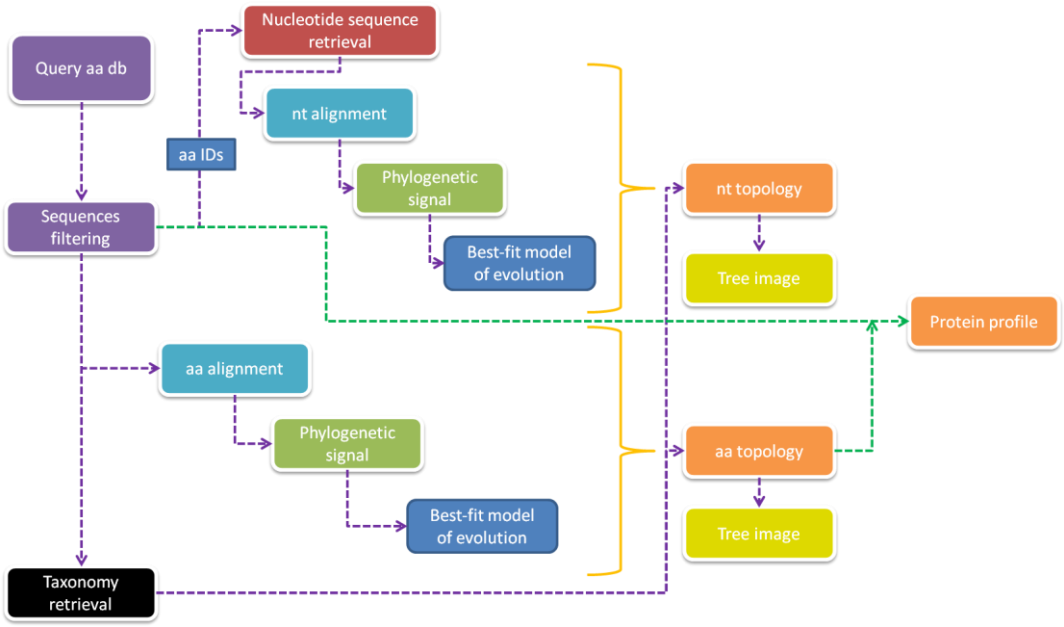


Figura 1: Workflow general

La propuesta de realización del TFM comenzó con la obtención, mediante uso de herramientas bioinfomáticas en terminal, de secuencias del gen de interés. En este punto existen diferentes bases de datos que se pueden interrogar. Por un lado la base de datos del *National Center for Biotechnology Information* (NCBI), dispone de la información aportada para nucleótidos y proteínas. Esta fuente de información dispone de herramientas para consulta por el terminal.

Una herramienta potente para la interrogación de la base de datos es *esearch* (1). La ventaja de esta base de datos son la amplia cantidad de información que puede ser obtenida tanto a nivel de secuencias en nucleótidos, como traducidas a proteínas para el objeto del estudio. La contrapartida pasa por tener un grado de información no correctamente filtrada que precisa de una adecuada aproximación para reducir la información que pueda ser redundante, no informativa o que pueda no responder a la información que se solicita. En este punto, se abrieron varias posibilidades de abordaje entre las cuales se incluyen: elección de herramientas de filtrado, búsqueda de posibilidades en la integración de la recuperación de información, tanto a nivel de nucleótidos como proteínas que pertenezcan a la misma accesión.

Alternativamente, otra propuesta inicialmente planteada interrogar la base de datos del *European Bioinformatics Institute* (EMBL-EBI), en concreto UNIPROT (2). Esta está curada a diferencia del NCBI. Esta aproximación presentaba la ventaja de tener una fuente de información ya previamente pre-procesada, si bien nunca quedaría completamente exenta de la necesidad de una supervisión por parte del investigador.

Existen herramientas específicas desarrolladas por la propia EMBL-EBI que permiten alejarse de las interrogaciones mediante web browser (3). La herramienta UniProtJAPI, desarrollada en java, permite la obtención de información mediante diferentes aproximaciones, y eso se traduce en la realización de un trabajo más preciso y de mayor calidad.

En ambas aproximaciones hubiesen requerido establecer límites que se aceptan en la información a utilizar. Para ello el uso de filtros específicos a aplicar en cada etapa del proceso debería ser el principal punto de partida en el proceso.

A partir de la información correctamente procesada, el siguiente paso fue el alineamiento de las secuencias. Existen diferentes algoritmos de alineamientos, *ClustalW* (4) (5) ampliamente utilizado en el alineamiento de nucleótidos, *MUSCLE* (6) empleado en el alineamiento de aminoácidos. Estas herramientas de alineamiento son empleadas habitualmente a nivel bioinformático e incorporadas en numerosas aplicaciones, si bien el uso mediante terminal fue el método elegido, evitando en la medida de lo posible el uso de herramientas que tengan que ocupar procesos en formato GUI en este paso del proceso.

Sería importante una comprobación del resultado de los alineamientos para poder descartar información incluida que pueda afectar en los siguientes pasos. Un punto crítico estaría el control de la longitud de los alineamientos, ya que secuencias incorporadas de longitud demasiado corta, pese a haber sido correctamente anotadas, supondrían la aparición de resultados que no se corresponderían con la realidad de las relaciones filogenéticas que se esperarían obtener. En este punto podía ser necesario la eliminación de dichas secuencias y la realización de un nuevo alineamiento de las mismas.

Una vez completado el paso de alineamientos, sería importante obtener dos tipos de información. Por un lado, la identificación de la existencia de señal filogenética en los alineamientos obtenidos. Esto puede ser accesible mediante *TreePuzzle* (7) (8), e intentará ser explicado con mayor detenimiento durante la realización del trabajo. Por otro, una vez identificada la señal filogenética, será necesario conocer el modelo evolutivo que mejor se ajusta a los datos. Para secuencias de nucleótidos, se utilizará *ModelTest* o su versión Java, *jModelTest* (9) (10) (11). En el caso de proteínas, se usará *prottest* (12) (13). Paralelamente, se planteó evaluar *ModelTest-NG* (14), que es la nueva versión que incluye ambos programas.

Las secuencias alineadas pueden presentar numerosos gaps siendo necesaria la eliminación de estas regiones, ya que puede incrementar la necesidad de computación. Existe la posibilidad de utilizar la herramienta *Gbloks* (15) (16) y reevaluación mediante la repetición de los pasos anteriores para comprobar que no exista pérdida de la calidad de la señal filogenética obtenida.

A partir de este punto, se esperaría estar en condiciones de proceder a inferir la filogenia de las secuencias obtenidas. Para ello, la propuesta pasa por el uso del software *RAxML* (Randomized Axelerated Maximum Likelihood) en el caso de secuencias de proteínas (17). Se emplearía la versión VI-HPC (high performance computing).

Para la realización del análisis de secuencias de nucleótidos, se utilizará el software *MrBayes 3.2* (18), una versión mejorada frente a la versión original (19), ya que permite la recuperación del proceso y la ampliación del número de generaciones si no se ha obtenido una convergencia adecuada de los resultados.

Los resultados obtenidos en archivos Newick format Tree file (20), podrán ser visualizados empleando bien herramientas en *Python*, bien herramientas en formato GUI como *FigTree* (21).

Para la construcción del perfil de la proteína, se elegirían representantes de la topología final obtenida para los clados I y II. Las secuencias de proteínas de estos dos clados se utilizarán para obtener una predicción de estructura por reconocimiento de plegado o *threading*. Para ello se procederá al análisis mediante la plataforma Phyre2 (22). Los resultados obtenidos serán también analizados mediante *gene investigator* y en las bases de datos de SCOP y UNIPROT.

2.1 Planificación del Trabajo

Para cumplir con los objetivos del proyecto, se identificaron una serie de tareas a cumplir, teniendo en cuenta el plan docente a cumplir y los hitos establecidos para los diferentes entregables a preparar.

1. Definición de los contenidos del trabajo.

Esta tarea coincide en el tiempo con la PEC0 del TFM y su hito correspondiente.

2. Plan de trabajo.

Esta tarea coincide con la temporalización de la PEC 1. Consistió en la elaboración del Plan de Trabajo donde se desglosan todos los puntos de la presente memoria: contexto y justificación, del trabajo; descripción de objetivos generales y específicos; explicación del enfoque metodológico a aplicar; plasmación de un calendario de actividades; identificación de los resultados que se esperan obtener; una estructuración del proyecto donde se concreta el contenido de cada una de las partes del proyecto tal como las describe el Plan docente; y finalmente un soporte bibliográfico que respalde el contenido científico-técnico del proyecto a realizar.

3 Desarrollo del trabajo fase 1.

Esta primera fase de trabajo fue la más extensa debido por un lado a que trató los aspectos iniciales de la obtención de información de las bases de datos públicas, como a su correcta manipulación. Además soportó la mayor carga temporal al incluir la computación desde el ordenador del estudiante de los programas filogenómicos que se plantean usar en el presente documento.

Si bien el Plan docente establece dos fases de trabajo, se ampliaron intencionadamente solapando con el desarrollo en fase 2 del proyecto como medida de prevención en la extensión en el tiempo de los procesos de computación.

3.1 Descarga de secuencias (nuc, aa).

Durante la realización de esta sub-fase se interrogó mediante el uso de herramientas como *esearch* o la herramienta *UniProtJAPI* las bases de datos del NCBI o EMBL-EBI para recuperar secuencias de aminoácidos y nucleótidos que respondiesen al gen de estudio, en este caso *nosZ*.

3.2 Filtrado de resultados.

La experiencia previa en la búsqueda mediante herramientas en el navegador web y en el terminal indica que con frecuencia las bases de datos arrojan información que no se corresponde específicamente con el gen de interés, y en ocasiones aportan información más allá de la secuencia específica que se busca. Sobre todo a nivel de nucleótidos, donde la incorporación de genomas completos bacterianos de gran tamaño hace que la búsqueda específica de secuencias tenga que ejecutarse de manera más precisa. Es por ello que se estableció un tiempo a priori mayor del necesario, para poder establecer los mecanismos óptimos que se puedan llegar a ejecutar para el correcto filtrado de los resultados.

Un punto complejo partiendo de las secuencias de aminoácidos fue la obtención de las secuencias de nucleótidos asociadas. El uso de la herramienta *t-blastn* del NCBI por ejemplo se conoce que aporta en ocasiones secuencias de genomas completos que es compleja su manipulación. Por ello se buscaron fórmulas de poder obtener secuencias específicas de nucleótidos que fuesen los que den la secuencia de aminoácidos de la especie seleccionada.

3.3 Generación de modelos.

A partir de las secuencias debidamente filtradas y previamente a la realización de los análisis filogenéticos fue necesario la realización de tareas de alineamiento múltiple de secuencias, la búsqueda de un modelo evolutivo y el tratamiento de las secuencias alineadas para minimizar las necesidades computacionales.

En este punto el alineamiento de secuencias de aminoácidos se realizó mediante *MUSCLE* (6), y en el caso de nucleótidos mediante *ClustalW* (4) (5). Atendiendo a los archivos de salida generados, si se precisase una re-manipulación de los datos para que puedan ser interpretados por los siguientes programas se buscarían los comandos adecuados.

Con las secuencias alineadas son necesarios por un lado la identificación de señal filogenética y por otro lado el establecimiento del modelo evolutivo que mejor se ajusta a las secuencias. Para la identificación de señal filogenética el programa *TreePuzzle*, permite mediante el análisis de las secuencias alineadas la identificación de sitios informativos. Al final de su ejecución ofrece un informe y una representación gráfica del análisis realizado donde muestra si los datos tienen señal filogenética.

Las secuencias alineadas pueden producir gran cantidad de espacios no alineados o gaps, identificados como (-). En la medida de lo posible podía necesario llegar a un consenso entre el número de gaps entre las secuencias y el nivel de señal filogenética. Para ello se estableció como necesario el uso del programa *Gblocks*. Este programa

permitiría utilizando la configuración de parámetros sobre los alineamientos, evaluar qué conjuntos dentro del total de las secuencias alineadas podrán eliminarse manteniendo bloques de secuencias alineadas y reduciendo en número de gaps. Esta herramienta permitirá evitar posiciones que estén pobremente alineadas y regiones divergentes tanto para secuencias de nucleótidos como de aminoácidos. Si bien existe una herramienta web que tiene unos parámetros seleccionables mediante cuadros de selección, la herramienta en un ordenador local permite un ajuste más fino de los parámetros, por lo que se optará por esta segunda opción a la hora del procesado.

Dado que el filtrado podría ser excesivo, podría ser necesaria una reevaluación de la señal filogenética mediante el software *TreePuzzle*.

3.4 Análisis de secuencias ML.

Mediante el programa *RAxML* se ejecutaría el análisis filogenético utilizando el modelo evolutivo para proteínas identificado en el paso 3.3, de las secuencias de aminoácidos seleccionadas. Si bien existe una versión *RAxML-NG* de reciente publicación, que permite la generación de un archivo checkpoint para poder relanzar los procesos, debido a que las especificaciones indican que está pensado para su utilización en clúster, no podrán utilizarse en el presente TFM.

El programa realiza (Randomized Axelerated Maximum Likelyhood) un cálculo de Máxima verosimilitud basado en inferencia para grandes árboles filogenéticos. Originalmente deriva de *fastDNAml* el cual a su vez derivó del programa de Felsestein *dnaml* incluido en el paquete *PHYLIP*. De hecho permite trabajar con archivos en este formato como input para la realización de los análisis.

Debido a que se necesita una elevada cantidad de memoria atendiendo a la cantidad de datos a analizar. Se utilizó la herramienta de cálculo de la siguiente dirección web: https://cme.h-its.org/exelixis/web/software/raxml/. Con esta evaluación se valoró si era posible la realización mediante los recursos propios disponibles.

Una vez confirmado se iniciaría el análisis y se supervisará periódicamente el estado de avance del análisis para aplicar las medidas correctoras que fuesen necesarias en caso de suceder algún contratiempo.

El archivo con formato *newik* resultante, fue utilizado para establecer la topología de la filogenia. Podría visualizarse mediante el programa *Figtree* y se buscaría alternativamente poder plasmarlo mediante un script en *Python*.

3.5 Análisis de secuencias Bayes.

Mediante el uso del programa MrBayes 3.2 se ejecutó un análisis filogenómico de las secuencias de nucleótidos atendiendo al modelo que obtenido en el paso 3.3 para las secuencias de nucleótidos alineadas.

MrBayes 3.2, a diferencia de la versión de *RAxML,* presenta la ventaja que adaptaron el programa para incluir un archivo del tipo *checkpoint*, en el caso que fuese necesario relanzar el número de generaciones a analizar si no se produjese confluencia entre los

resultados de los runs y las cadenas que se generen en cada iteración para obtener una topología robusta.

Este programa utiliza modelos de Marcov Montecarlo (MMC). Debido a las características del ordenador que tiene el estudiante se intentará ejecutar un análisis en 4 runs con 4 cadenas y se analizó la convergencia de las cadenas mediante el programa *Tracer* para visualizar si se produce convergencia de las cadenas y cuántas generaciones serán necesarias.

Una vez finalizado el análisis se procederá a solicitar que se generen los archivos de síntesis de parámetros y de topología. Este paso será exclusivamente realizado por parte de investigador, por lo que será necesaria su intervención en este momento.

El archivo con formato *newik* resultante, sería utilizado para establecer la topología de la filogenia.

4. Desarrollo del trabajo fase 2.

La segunda fase de desarrollo del Trabajo Final de Máster se centró en la obtención de perfiles de la proteína mediante el uso de herramientas en la web, Phyre y un análisis de los perfiles obtenidos.

4.1 Establecimiento perfil de la proteína.

Debido a que esta parte del trabajo no podía ser realizado a nivel de computación local, se optó por el uso de la plataforma Phyre2. Phyre (acrónimo de Protein Homology/analogY Recognition Engine). Es un servicio web para la predicción de estructuras de proteínas tridimensionales de proteínas. Esta plataforma emplea principios y técnicas de modelado por homología combinados con modelos de Markov ocultos (Hidden Marcov Models o HMM). Estos perfiles permiten aproximar la composición de una secuencia de aminoácidos basada en mutaciones observadas en unas secuencias relacionadas, por lo que podremos considerar que es un rastro digital de la evolución de una proteína en particular.

Al tenerse conocimiento de la diferenciación entre clados I y II en las filogenias, se trató de establecer el perfil de la proteína para una o varias especies del clado I y clado II para intentar identificar diferencias si existiesen, en su estructura tridimensional.

Además se trató de dar una explicación sobre el modelo teórico y su ajuste contra los modelos que presenten identidad en la base de datos de Phyre.

4.2 Programación script.

Una vez ensayados y conocidos los comandos, programas y necesidades que ha precisado el proyecto para de manera parcialmente automatizada, proceder al análisis de un conjunto de datos a partir de los parámetros establecidos, comenzó la fase de programación de uno o varios scritps que permitan al automatización en la medida de lo posible de todos, o al menos los aspectos más importantes del trabajo. Ello requerirá un cambio en el planteamiento de los procedimientos paso a paso realizados

para que el programa o los programas puedan utilizar un archivo de información o un conjunto de comandos sencillos que puedan realizar el máximo número de pasos necesarios sin tener necesidad de una supervisión por parte del investigador de estar introduciéndolos secuencialmente. Dentro de estos programas, se intentó potenciar el orden de los datos y que el investigador pueda realizar una supervisión durante el proceso o después del proceso, en el caso de necesitar confirmar la información de los archivos intermedios generados en los diferentes procesos.

4.3 Testing script.

Una vez programado, y con el fin de probar la posibilidad de ejecutar el máximo número de pasos posibles, se planteó un testado del programa con un conjunto reducido de datos, de manera que se intentase comprobar que efectivamente realiza una manipulación de acuerdo a las especificaciones programadas, para reducir el tiempo de intervención del investigador, identificando los fallos que puedan producirse para poder subsanarlos. Los resultados y gráficos que se obtengan no serían objetivo del resultado final de la filogenia, sino de la capacidad del programa de evitar supervisión y tiempo necesitado por el investigador para lanzar diferentes programas. En esta parte del proyecto se consultó con la tutora aquellos puntos que presenten una dificultad no superable por parte del alumno para la mejor realización del mismo.

5. Cierre de la memoria.

En esta fase de proyecto se completará la redacción de la memoria, incluyendo los últimos resultados obtenidos, una discusión de los resultados alcanzados y el nivel de compleción del trabajo así como las conclusiones finales a las que se ha llegado en la realización del proyecto. Esta fase se coordinará con la tutora la mejor presentación de todo el trabajo realizado de cara a obtener el mejor entendimiento del mismo.

6. Elaboración de la presentación.

Se trató de la fase final del proyecto previamente a la defensa pública del proyecto. Consistirá en la plasmación gráfica y sintética de los resultados más relevantes obtenidos durante la realización del Trabajo Final de Máster propuesto. Con la intención de presentar de manera sencilla y comprensible todo lo alcanzado durante la ejecución, haciendo especial énfasis en los puntos más relevantes que se hayan desarrollado durante el tiempo de la investigación.

Redacción de la memoria. La memoria final ha sido un documento vivo que ha estado en actualización periódica durante toda la realización del TFM. Si bien se presentaron diferentes entregables coincidiendo con cada uno de los hitos que se presentan en el punto 4.3, la memoria será un documento que tendrá que redactarse con todos los resultados obtenidos que aporten información relevante al mismo. Además dentro de la estrategia de ejecución la revisión de bibliografía, la incorporación de nuevas citas, los cambios en el plan de ejecución son cuestiones que se podrían ir reflejando en la medida que sucediesen, de manera que se facilitase una mejora final del documento a presentar.

A continuación se incluye el cronograma inicialmente planteado.

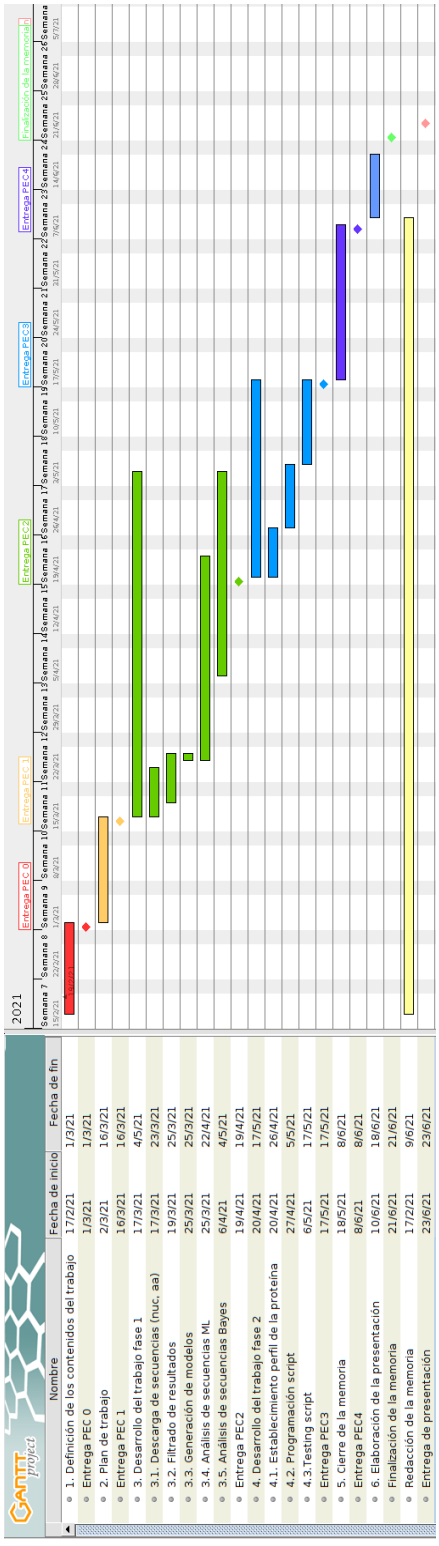


Figura 2: Diagrama de Gantt con planificación en la ejecución del proyecto

3. Estado del arte

El nitrógeno (N) es un átomo imprescindible para la vida que está incluido dentro de los ciclos biogeoquímicos. En la atmosfera se acumula el gas nitrógeno (N_2) y óxidos de nitrógeno (NOx). Los óxidos de nitrógeno son empleados por algunas bacterias para su metabolismo. Su acumulación excesiva en la atmósfera tiene efectos perjudiciales. El nitrógeno va a ser fijado e incorporado a través de una serie de reacciones del nitrógeno por parte de bacterias hongos y cianobacterias pasando a nitrato (NO_3), estando biodisponible para otros seres vivos. Los desechos producidos por estos seres vivos (donde incluimos principalmente plantas, animales, bacterias), son expulsados al medio en forma de ión nitrito (NO_2) o de amoníaco (NH_3).

En este punto nitrito y amoníaco pueden volver a formar nitratos mediante un proceso de nitrificación, o puede retornar a la atmósfera en forma de nitrógeno, siendo este último el punto de partida del presente Trabajo Final de Máster.

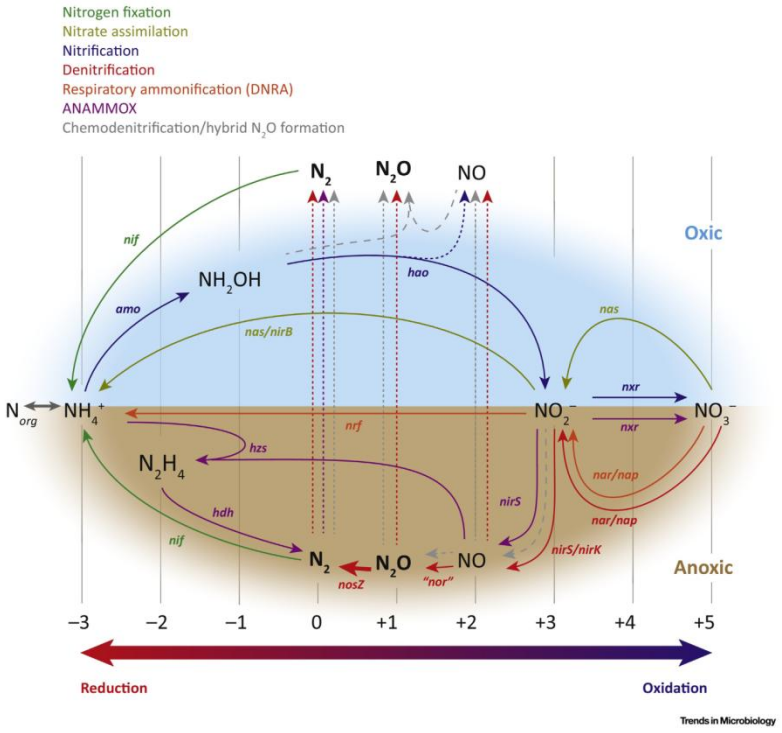


Figura 3: Ciclo biogeoquímico del nitrógeno con diferenciación de condiciones de oxidación/reducción así como rutas de nitrificación desnitrificación con indicación de los genes *nor* y *nosZ* participantes en los pasos de desnitrificación (7)

La desnitrificación es la reducción respiratoria del nitrato o nitrito a N_2 o N_2O y generalmente acontece bajo condiciones anaeróbicas. La reducción del óxido nitroso es el paso final en las reacciones de desnitrificación y es catalizada por el enzima óxido nitroso reductasa. El óxido nitroso es un gas implicado en la destrucción de la capa de ozono así como en el calentamiento global (23), (24). El gen codificante de

esta reacción (*nosZ*), ha sido empleado históricamente para la identificación de clústeres de especies (25), (26). La existencia de especies que parecen tener una vía alternativa como *W. succinogenes* (27) parece que abren otras vías interesantes para el modelado. Un aspecto importante es que para que se pueda producir una desnitrificación completa del N_2O a N_2, bien para su nueva fijación y reintroducción en el ciclo o bien su paso a la atmósfera, es esencial la participación como catalizador, de la óxido nitroso reductasa.

Actualmente, la proteína NosZ es la única enzima capaz de activar el óxido nitroso inerte. Su mecanismo de reacción incluye dos sitios de unión a metal, pero no está completamente dilucidado (28).

Dentro de las últimas revisiones sobre este gen, los mayores linajes en la composición se organizan en dos clados para *nosZ* (I y II). La mayor parte de las secuencias para el gen *nosZ* en el clado I, aglutinan a Alfaproteobacterias y Betaproteobacterias. Las secuencias del clado II, predominantemente incluyen a Bacterioidetes, Firmicutes, Gammaproteobacerias y Epsilonproteobacerias (29).

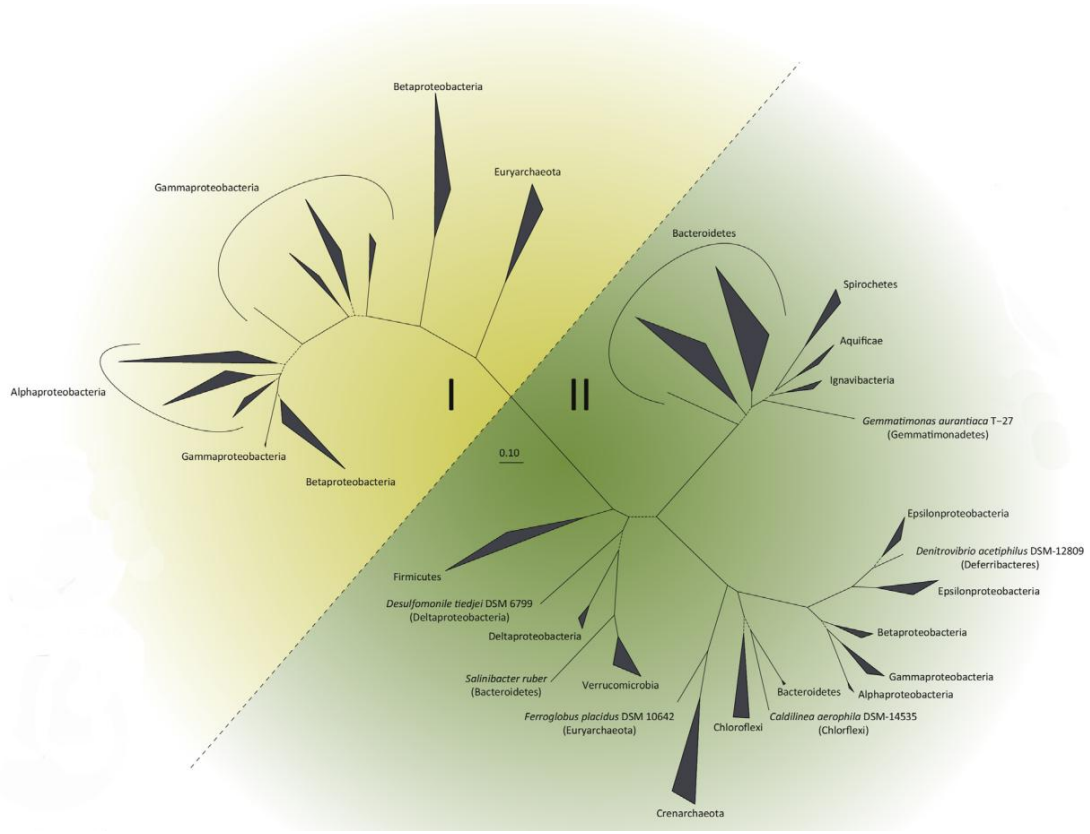


Figura 4: Filogenia basada en ML de bacterias con gen *nosZ*. Adaptado de la publicación de (29)

Si bien la familia de genes *nos* ha sido estudiada durante numerosos años, queda gran cantidad de conocimiento pendiente de dilucidar, por lo que actualmente se siguen presentando artículos y revisiones que tratan esta familia de genes, mejorando la compresión que se tiene sobre la misma.

La generación de información mediante técnicas de secuenciación masiva ha sufrido un incremento exponencial en los últimos años con un descenso en el coste asociado en la generación de dicha información.

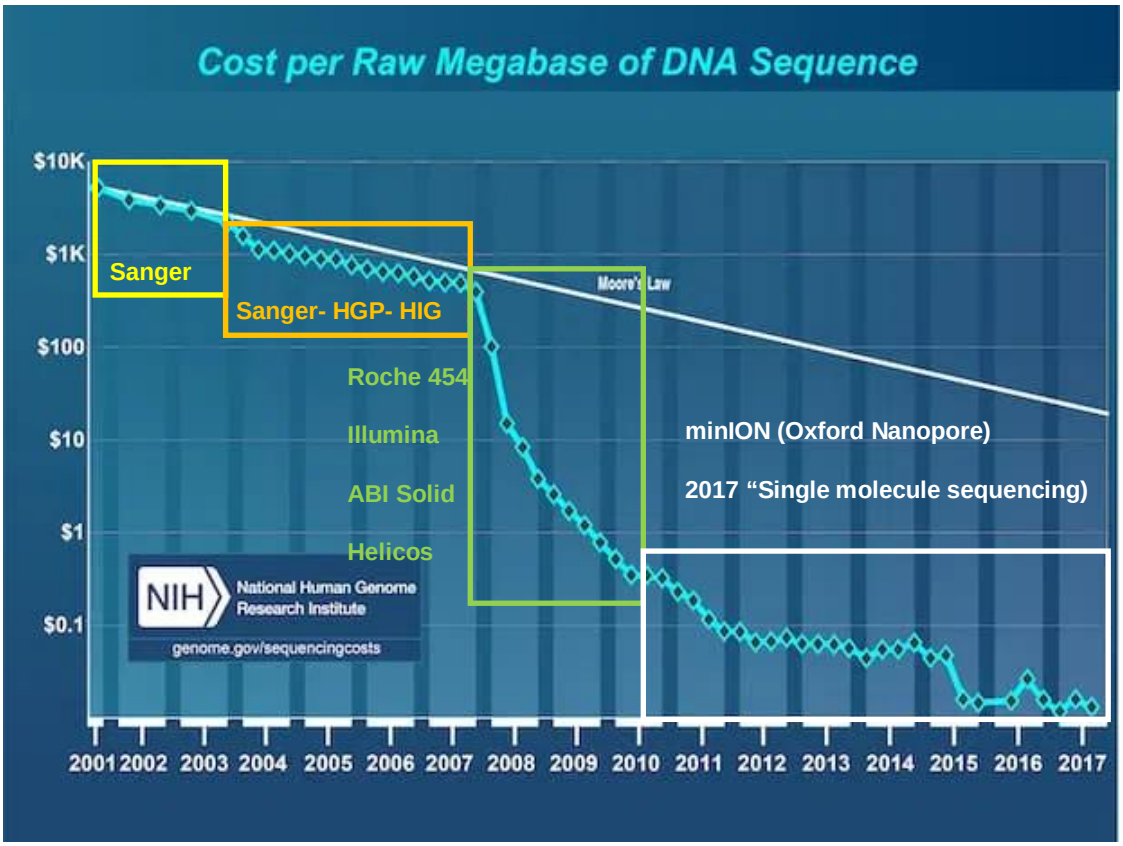


Figura 5: Evolución en coste para la generación de secuencias desde principios de siglo. Fuente NCBI (adaptado).

Todos estos resultados son incorporados en bases de datos, muchas de ellas públicas, con la contrapartida de tener un grado de información no correctamente incorporada. En el caso de bacterias, la secuenciación masiva de muestras obtenidas de numerosas fuentes, desde especies conocidas y bien identificadas a muestras obtenidas de la naturaleza y sobre las cuales se hace un screening y ensamblaje de información general sobre los resultados de la secuenciación masiva. Como resultado, las bases de datos están atestadas de gran cantidad de información que en ocasiones se convierte en difícilmente manejable a nivel del investigador.

Surge por ello la necesidad de poder investigar las bases de datos mediante el uso de herramientas informáticas para poder manipular los datos y obtener unos resultados que puedan ser incorporados a los siguientes procesos que interesen en el proceso de la investigación.

Cuando los recursos de computación son grandes, los investigadores se pueden plantear descargar toda la información de las bases de datos públicas y trabajarlas en servidores (clúster) que les permitan agilizar el análisis de la información contenida y establecimiento de los conjuntos de datos que son aptos para su investigación.

En el caso de no disponer de dichos recursos, se debe transferir estas búsquedas a los servidores donde están contenida esta información menos procesada.

4. Metodología

4.1 Descarga de secuencias (nuc, aa).

El primer paso en el procedimiento ha sido la obtención de un conjunto de secuencias de proteínas para el gen de interés. El motivo ha sido por un lado, debido al enfoque del proyecto, el que se va a realizar un modelado de la proteína, trabajar en primera instancia contra secuencias protéicas. Un segundo motivo es que las bases de datos no están completamente revisadas y existen posibilidades de pequeños fallos de anotación, lo que cabe la posibilidad de encontrarse ruido en las secuencias, entendiéndose como secuencias nombradas como algo semejante a la proteína de interés pero no presentar correspondencia. Finalmente, y debido a que el código genético es degenerado, iniciar la búsqueda a partir de secuencias de nucleótidos dificultaría inicialmente una aproximación visualmente más controlable de las secuencias de proteínas alineadas.

Con estas premisas, y mediante el uso de herramientas en terminal se ha realizado una búsqueda en el lado del servidor del NCBI mediante la herramientas herramientas `esearch` y `efetch` que forman parte del conjuto E-utilities. Las normas inican que se debería proceder al uso de la herramienta incluyendo, por lo que se tratará en el apartado de establecimiento de scripts.

El comando `esearch` provee lista de UIDs que presenten identidad (match) con un texto que se incluya en el comando. Como resultado devuelve los resultados de la búsqueda en el servidor. El comando necesita que se le indique la base de datos sobre la cuál se requiere realizar la búsqueda. Dado que la selección ha sido iniciar la búsqueda de secuencias de proteínas la base de datos seleccionada fue “protein”.

El comando EFetch (`efetch` en línea de comandos). Produce una salida en una variedad de formatos, resultando de interés para el trabajo el formato fasta.

Estos dos comandos se combinaron mediante tuberías (pipe) para que la salida del primer comando fuese procesado por el segundo comando. Finalmente en este punto se derivó la salida final generada a un archivo nosz_prot.fasta. A continuación se incluye el comando empleado:

```
$ esearch -db protein -query "nosZ" | efetch -format fasta > nosz_prot.fasta
```

Para tener una primera aproximación del número de secuencias que el servidor devuelve, en el terminal, mediante el comando `grep` y la búsqueda del símbolo “>” que es el elemento que en los archivos fasta indica el inicio del nombre de la secuencia que vendrá a continuación. Como resultado se indicó que el servidor envió 44.607 secuencias que de alguna manera presentaron identidad con los términos de búsqueda. La salida de esta búsqueda se derivó a un archivo temporal, a efectos de poder realizar una primera visión del contenido mediante un editor de texto plano.

La revisión de el resultado puso de manifiesto la existencia de numerosa información correspondiente a especies no identificadas, especies repetidas e incluso genes que no se correspondían con el gen de interés tal y como se muestra en la siguiente figura:

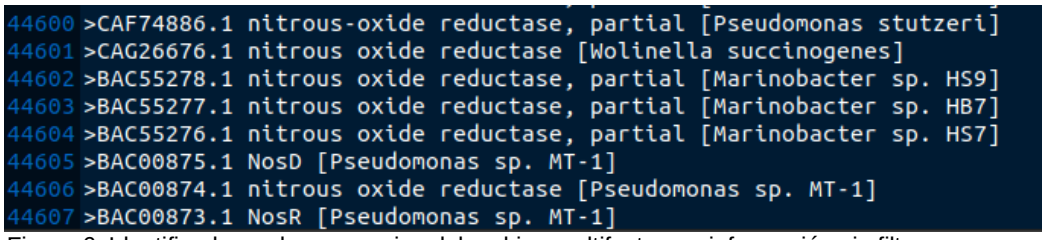


Figura 6: Identificadores de secuencias del archivo multifasta con información sin filtrar

Para la correcta ejecución y vista la problemática acontecida, se optó por la estrategia de proceder al filtrado de las secuencias de proteínas y una vez completada trabajar la recuperación de secuencias de nucleótidos. No obstante se realizará una descarga de secuencias de nucleótidos, tal y como se describirá más adelante si bien no se procedió a una descarga de secuencias en crudo (raw).

A partir de esta información se optó por procesar la información para aminoácidos y a partir de los resultados finales obtenidos repetir obtener la información en nucleótidos partiendo de la selección final realizada, de manera que hubiese correspondencia entre secuencias de aminoácidos y secuencia de nucleótidsos.

4.2 Filtrado de resultados

La decisión sobre el un modo correcto de filtrarlos datos planteó diferentes típos de uso de palabras clave, aquellas a incluir, y aquellas a excluir. Tras la realización de varios ensayos se optó por un filtrado en dos pasos. En el primero y dado que el nombre del gen mayoritariamente no apareció, pero sí la descripción de nitrous oxide reductase, salvo que el uso de nitrous no favoreció la búsqueda, se realizó una búsqueda de todas aquella líneas que hiciesen match contra las keywords oxide reductase mediante el comando grep y derivando la salida del archivo nosz_prot.fasta a un archivo lista1.txt. Como resultado se redujeron el número de identificadores de línea a 26.194.

Tras un filtraje de secuencias que tuiesen un match positivo, el siguiente paso de filtrado consistió en realizar una búsqueda que dejase únicamente aquellas líneas que no tuviesen una serie de keywords (match negativo), nuevamente mediante el uso de la herramienta grep.

Se valoraron diferentes conjuntos de keywords para discriminar negativamente y eliminar aquellas líneas que tuviesen dichas palabras. Finalmente los keywords seleccionados fueron:

- sp.
- Uncultured
- Unclassified
- Culture
- \[bacterium

- proteobacteria
- endosymbiont
- \proteobacterium
- givision
- gamma
- gaceae
- bacterium]
- bacterium
- partial
- nosD
- nosY
- norL
- nosR

Como resultado del filtrado negativo mediante el uso del comando grep, se redujeron el número total de salidas filtradas a 4.586 regitros.

Con una información que se valoró como adecuada para poder continuar con el procedimiento, el siguiente paso fue la eliminación de especies duplicadas. Dado que no se tenía conocimiento de una relación sobre las secuencias duplicadas y su idoneidad se siguió la regla de eliminación del resto de secuencias que tuviesen el mismo nombre a partir del primer registro devuelto por el servidor del NCBI que proveyó las secuencias. Si bien la metodología no es 100% fiable, pero ofrece suficientes garantías para una serie de pasos de filtrado, se dividió la información contenida en cada uno de los registros, entre todo aquello precedente a la indicación de la especie, indicada entre corchetes (símbolos [y]) y la información contenida entre corchetes. De esta manera se obtuvieron línea a línea dos listados llamados lista3_1.txt y lista3_2.txt respectivamente.

Con el fin que línea a línea cada uno de los registros se quedasen únicamente con un registro único por especies se concatenaron los comandos paste y sort. Paste une horizontalmente el contenido de archivos y sort ordena los registros. Al tener separados por un lado la información de las especies y por otro lado los nombres de las especies, al comando sort se le adicionó la opción -u (unique) –k2,3 (que toman los elementos 2 y 3 del segundo archivo en esta concatenación horizontal). Los elementos 2 y tres del segundo archivo correspondieron a priori al género y especie. Como resultado se obtuvieron 1109 registros en un archivo de salida nombrado como lista3.txt.

En este punto fue necesario una revisión manual de los nombres finalmente obtenidos y la corrección de erratas, ya que la existencia de nombres entre corchetes que comenzaron por Candidatus, no permitieron obtener el nombre del binomio género especie, sino que devolvieron salidas del tipo Candidatus acompañado por el nombre del género.

Se eliminaron manualmente secuencias correspondientes a pir||S24384 (que estaba duplicado para la especie *Pseudomonas stutzeri*), GBC71360.1 (correspondiente a un archeon sin identificar), KFM19712.1 (identificado como grupo marino no identificado)

y WP_060528325.1 que fue identificado en la búsqueda de nucleótidos como *Sedimenticola taunini* y que estuvo duplicado en los listados que se presentarán maś adelante.

Tras la revisión manual de 1109 secuencias, finalmente se generó un archivo llamado lista4.txt con 1103 registros.

A modo informativo en este punto y utilizando los comados gawk y sort se elaboró un listado de género únicos, obteniéndose un listado con un total de **451 géneros únicos** y un total de **1103 especies únicas** de proteínas para el gen de interés. En la siguiente figura se resumen el conjunto de comandos empleados así como el número de líneas de información esenciales para cada una de ellas.

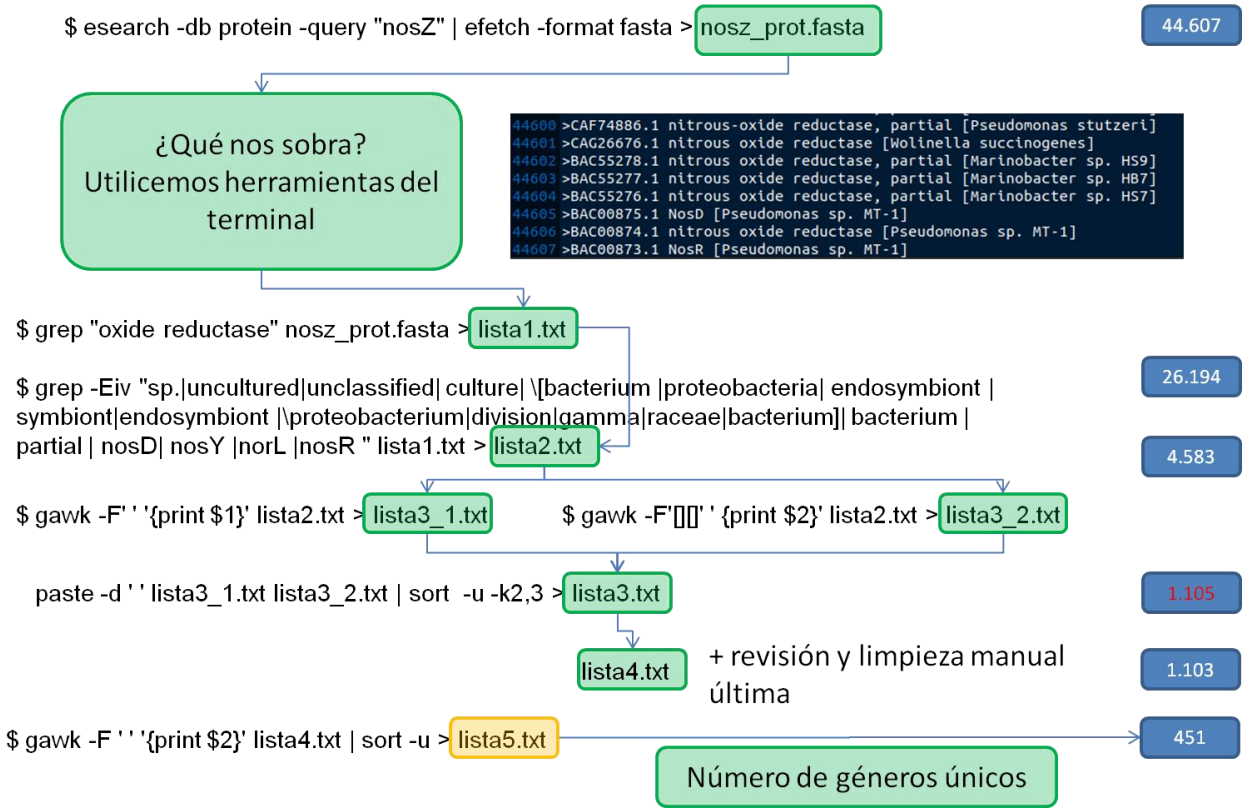


Figura 7: Resumen de la estrategia de filtrado

Esta información ya procesada, fue necesaria para poder recuperar a partir de una lista de secuencias, el conjunto de secuencias en formato fasta. En este caso el siguiente paso en el procedimiento fue generar una lista de IDs de proteínas que pudiesen recuperar selectivamente a partir del archivo nosz_prot.fasta un archivo multisecuencia con 1.104 secuencias con su identificador y la secuencia fasta correspondiente.

Existen numerosos scripts que permiten un parseo de la información de un archivo multifasta a partir de una lista. En mi caso opté por hacer uso de una herramienta ya generada por Joe Healey (https://github.com/jrjhealey/bioinfo-tools/blob/master/fastafetcher.py), que utiliza un pequeño script para recuperar la información según las necesidades.

El script necesita de un archivo multifasta y de un listado de secuencias, además de indicar un archivo de salida. Al ejecutarse el programa buscó el identificador en el archivo multifasta en bucle de manera que por cada match exporta al archivo de salida la secuencia completa. De esta manera se pasó de un archivo multifasta de 44.607 secuencias mediante el uso del script y con la incorporación de un listado de 1.104 entradas a un archivo multifasta de 1.103 secuencias fasta. El output fue nombrado como proteinas_limpio.fasta.

Figura 8: Explicación del flujo de información del script fastafetcher.py

Una vez obtenido un archivo multifasta con las secuencias de proteínas, se utilizó el listado de secuencias de nucleótidos a partir de los IDs de las proteínas recuperadas y debidamente filtradas y procesadas.

Para la recuperación de las secuencias de nucleótidos, el primer paso consistió en obtener la información necesaria sobre las secuencias de nucleótidos de interés a partir de los IDs de secuencias de proteínas. Para ello se empleó la herramienta `efectch` contra la base de datos ipg. El recurso ipg (de Identical Protein Groups) es un recurso que, a priori, contiene una entrada para cada proteína traducida qe se encuentre en varios recuros del NCBI, incluyendo las regiones anotadas en Genbank y RefSeq, así como SwissProt y PDB. Como output, aporta información sobre Id, Origen de la información (RefSeq, INSDC,…), Nucleotide Accesion, Start, Stop, Strand, Protein, Protein Name, Organism, Strain y Assembly. Sin embargo la separación de campos fue algo controvertida.

De cara a poder lanzar el comando, se utilizó la estrategia de separar del documento lista4.txt, la columna correspondiente al ID eliminando la información. El objetivo fue no tener el símbolo ”>” ni el binomio género especie. En última instancia sólo fue necesario ir seleccionando los grupos y concatenando los nombres en una única línea separando los IDs mediante comas.

De manera esquemática a continuación se mostrarán el código empleado, ya que el output obtenido se volcó en primera instancia a un documento info_nucleotidos.txt. Este documento fue necesario manipularlo debido a que los campos correspondientes al binomio género especie al descargar los datos no se organizaban mediante un separador de campo adecuado.

```
$ efetch -db ipg -id 'lista_IDs_separados_por_coma' -format ipg | grep 'RefSeq'> info_nucleotidos.txt
```

Los campos se manipularon manualmente para eliminar la información que no correspondiese al binomio género especie. Además el número de identificadores de secuencias obtenidas fue mayor al número de IDs, por lo que se precisó a posteriori un tratamiento de los datos para la eliminación de secuencias con especies repetidas. Dado que esta tarea se realizó con las especies ordenadas alfabéticamente en un documento de hoja de cálculo, la tarea no supuso mayor esfuerzo que el controlar los nombres duplicados en una columna concreta, eliminando las filas que duplicasen el contenido de la celda con los datos relevantes. Se generó un archivo secuencias_nt_a_descargar.txt que fue utilizado posteriormente.

Con el documento debidamente preparado, se procedió a una comparación de listas entre el listado de proteínas con sus ID correspondientes y las especies frente al listado recargado de ipg. Las no correspondencias que no aparecieron tras la descarga de secuencias de nucleótidos, se identificaron en el listado para poder recuperar la información ya en última instancia de manera manual. El procedimiento básico fue teniendo una lista con nombres e IDs de aminoácidos, y una lista con nombres e IDs para nucleótidos, se compararon línea a línea en un documento de hoja de cálculo con una función booleana de igualdad entre los nombres de las especies. En la medida que se recorrió línea a línea, al aparecer la booleano FALSO, se buscó el punto de discrepancia para insertar espacios en blanco en la columna de nucleótidos e IDs correspondientes para poder identificar la secuencias que no se recuperaron.

Con todos estos datos debidamente procesados, el siguiente paso fue la descarga de las secuencias en formato fasta. Un aspecto que se tuvo en cuenta fue el hecho que la información de los archivos resultantes podía estar en el strand positivo (+) o en el strand negativo (-). Para que este hecho no fuese un problema se preparó un pequeño script en Python (si bien está en fase de intentar mejorar) llamado nt_download.py: (incluido en el anexo 2).

En este paso se consiguió que a medida que se solicitó la descarga de las secuencias al NCBI, se fuesen separando en dos archivos, de manera que el archivo con las secuencias en strand negativo se pudiesen cambiar al strand positivo mediante otro script nombrado como reverse_complement.py (incluido en el anexo 2).

Las secuencias que no pudieron ser descargadas y procesadas automáticamente fueron descargadas manualmente directamente del NCBI. El documento Suplemento1.xlsx incluye en la pestaña Sequences IDs. En el listado de secuencias, identificando en color azul aquellas que fue necesario recuperar la información manualmente en color rojo aquellas que se detectaron como secuencias que debieron ser descartadas por identificación posterior en el proceso de recuperación manual.

Los archivos forward.fasta, forward2.fasta y manual.fasta se concatenaron mediante un script llamado merge2files.py (incluido en anexo 2).

Una figura para poder resumir los pasos realizados hasta el momento se presenta seguidamente:

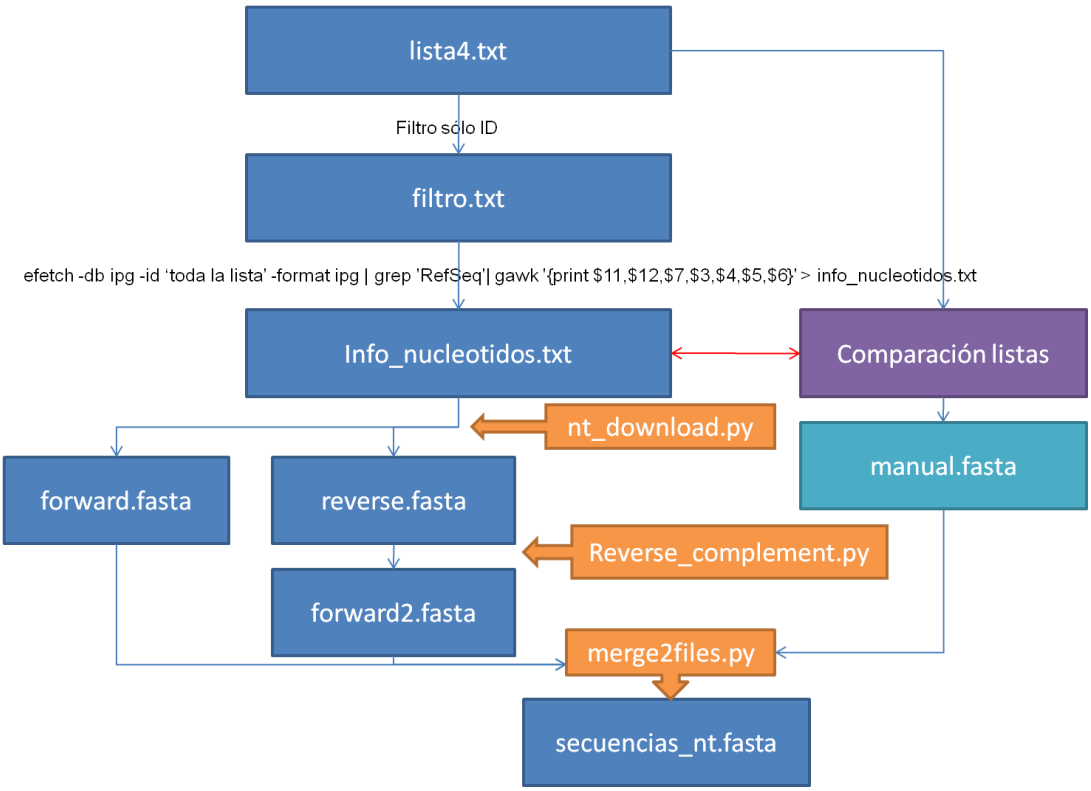


Figura 9: Esquema del proceso de descarga de secuencias de nucleótidos a partir de listado final de aminoácidos

4.3 Generación de modelos.

Gblocs

Se empleó el programa Gblocs de Castresana (15) (16), en el equipo local con las condiciones por defecto para identificar si cabría la posibilidad de reducir información a partir de posiciones de nucleótidos o aminoácidos en los alineamientos que pudiesen no tener un número representativo y que incrementase las necesidades de computación.

Señal filogenética con TreePuzzle

Para buscar si las secuencias alineadas tenían señal filogenética se empleó el programa TreePuzzle. Dado que las secuencias debían estar en formato phylip para su análisis, se utilizó el script fasta2phylip.py incluido en el anexo 2.

Los archivos formato phylip quedaron sin toda la longitud de la cadena del identificador, quedando presente 10 caracteres de la parte inicial del ID, que utilizamos en la tabla de descripción de las especies con sus ID para nucleótidos y proteínas.

Modelos evolutivos

Prottest

Para el establecimiento en la elección del modelo evolutivo en secuencias alineadas de aminoácidos que mejor se pueda adaptar a los datos descargados, se empleó el programa Prottest. Se trata de un programa implementado en java por el grupo de David Posada de la Universidad de La Coruña. Como input necesitó un archivo de proteínas alineado que disponíamos de pasos previos.

Los modelos evolutivos evaluados fueron JTT, LG, DCMut, MtREV, MtMam, MtArt, Dayhoff, WAG, RtREV, CprEV, Bosum62, VT, VIVb, HIVw y FLU. Los modelos se testaron bajo el criterio BIC y al LnL.

Modeltest

Modeltest, o su versión java, jmodeltest, es un programa permite obtener los valores de verosimilitud (Likelyhood scores) para secuencias alineadas de nucleótidos. Como input acepta archivos en formato phylip y nexus. Si bien se realizaron los cálculos sobre un archivo con formato phylip. Se generó un archivo tipo nexus para su posterior utilización con el programa MrBayes, el cual precisará un archivo tipo nexus. Para realizar la conversión, se procedió, por cuestión de tiempo a utilizar directamente el programa jemboss que permite un cambio rápido de formatos, si bien se continuará trabajando más adelante en la búsqueda de alternativas que se puedan automatizar en la media de lo posible.

En la ejecución de modeltest se solicitó la realización de un análisis de verosimilitud. Dado que el tamaño es significativamente mayor que el de las mismas secuencias en aminoácidos, se optó por incluir un setting con menos modelos. Para computar las verosimilitudues se pudieron dedicar 2 de los cuatro núcleos del ordenador, por lo que en los Likelihood settings se incluyeron 3 esquemas de sustitución, por lo que se calcularon 24 modelos y se mantuvieron el número de categorías en 4 con tasa de variación incluyendo gamma y sitios invariantes (+G, +I) tal, y como se muestra en la siguiente captura de pantalla.

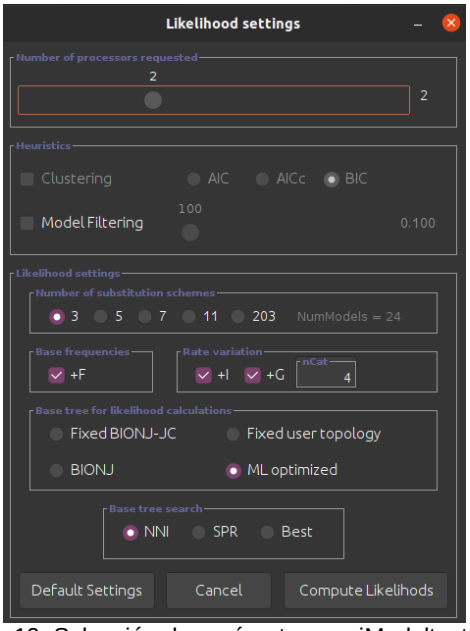


Figura 10: Selección de parámetros en jModeltest

4.4 Análisis de secuencias ML.

El alinamiento de secuencias de aminoácidos se ha inicializado su análisis mediante el programa RAxML. Se solicitó que ejecutase un análisis de bootstrap rápido con 100 réplicas y posteriormente procedió a la realización de la búsqueda de la mejor topología. El programa en terminal se ha llamado mediante el siguiente comando:

```
raxmlHPC-PTHREADS-AVX -T 2 -f a -x 1 -p 1 -# 100 -m PROTGAMMAWAG -s aligned_muscle.fasta.phylip -n TEST
```

4.5 Análisis de secuencias Bayes.

Tras generar en el apartado 4.3 un archivo tipo nexus para la ejecución con el programa MrBayes, se procedió a la adecuación de los datos de ejecución al final del archivo. Para ello se incorporó en la cabecera la información del modelo evolutivo, se inició con un número de 1.000.000 de generaciones y sin la existencia de un archivo checkpoint para que empezase a generarlo, además se le incorporó la opción que

generase un resumen de parámetros y de árboles al finalizar la ejecución por si hubiese convergencia de las cadenas y los valores fuesen adecuados. El resultado inicial de la ejecución se presenta en el siguiente resumen:

```
END;
begin mrbayes;
     set autoclose=no nowarn=yes;
     charset nosZ =  1 -  2314;
     partition currentPartition = 1: nosZ;
     set partition = currentPartition;
     lset applyto=(1);
          lset nst=6 rates=gamma;
     unlink    statefreq=(all)    revmat=(all)    shape=(all)
pinvar=(all);
     prset applyto=(all) ratepr=variable;
     mcmcp   ngen=   6000000   relburnin=yes   printfreq=1000
checkpoint=yes   checkfreq=1000   append=no   samplefreq=1000
nruns=2 nchains=4;
        mcmc;
        sump;
        sumt ;
end;
```

4.6. Establecimiento perfil de la proteína

Una vez obtenida la topología en los análisis mediante máxima verosimilitud y un análisis de un segmento reducido de las secuencias en formato de nucleótidos mediante análisis bayesiano, un apartado continuación para el estudio de la proteína nosZ fue el establecimiento de un perfil de la proteína.

Para ello se eligieron secuencias de aminoácidos de las especie *Wolinella succinogenes* y *Algoriphagus lacus* del clado II por un lado, así como la secuencia de la especie *Pelagimonas varians* del clado I como base para una búsqueda del perfil. A continuación las secuencias de aminoácidos para cada especies utilizadas en los análisis mediante el programa Phyre2.

>Pelagimonas varians

MSEETTQKMSVTRRGLLGATATGAAVAATGVGSALLGAGEAKAATGQPWHLAPGDL
DEYYGFWSSGQSGELRILGMPSMRELMRIPVFNRCSATGWGQTNESLKILTEGLLPE
TKEFLAANGKVTYDNGDLHHPHMSFTDGTYDGRFLFMNDKANTRVARVRCDVMKCD
KIIEIPNAHDIHGLRPQKFPRTGYVFANGEHEAPLVNDGTILDEPEQYVNIFTAIDGDEM
EVAWQVIVSGNLDNTDCDYQGKYAFSTSYNSEMGMTLAEMTEAELDHVVVFNIKAIE
EAVANGDYQELKGVKVVDGRKGAPGKLTRYIPIPNSPHGVNAAPDKRHICINGKLSPT
VSVIDVEKLDALFDADADPRSAVVAEPQLGLGPLHTAFDNRGNAYTTLFLDSQVVKW
NIQDAIASYGGSDVDPIKDKVDVHYQPGHNSTSMGETAEADGKWLISMNKFSKDRFL
NVGPLKPENEQLIDISGDKMKVVHDGPTFAEPHDSIIVHRSKVTPVDVWDRNDPMWE

DARVQAAADGIDLEDGAEEIIRDKDDPTKVRVYMTSVAPVFSWEKFEVNQGDEVTVY
VTNLDDVDDVTHGFCMANFGVAMEVGPQATASVTFVAERPGVHWFYCQWFCHALH
MEMRGRMFVKPREA

>Wolinella succinogenes

MQRLLKQSLVVTASLLALGTASLASSDLQTIMKERKLTEKDVLAAAKTYQPSGRKDEF
VVFSSGGQSGQILVYGVPSMRIYKYIGVFTPEPWQGYGFDDDSKKVLRQGDIRGREI
NWGDTHHPNFTEKNGEYVGDYLFINDKANPRIAVVNLHDFETTQIVVNPIMKSEHGGS
FVTPNTEYVIEASQYAAPLDHQYHPIEEYEAVFRGAVTLWKFDYAKGKIDEKASFSLEF
PPYMQDLSDAGKGESFGWAFTNSFNSEMYTGGIEKGLPPFEAGMSRNDTDYMHVY
NWQMLEKLAQDPKNYKIYHGHRVISIEAAVKAGALFLIPEPKSPHGVDVSPDGRYIVV
GGKLDTHASVYDFRKIKQLIDKKEFIGADPYGIPILDMKKTLHGQVELGLGPLHHTYDA
QDGIIYTSLYVDSQIVKWDYKNLKVLDRVNVHYNIGHLDSMEGKSAKPKGKYALALDK
LSIDRFNPVGPLHPQNHQLIDIGGPKMELIYDLPIPLGEPHDVISIAADKLKPQVTYPMG
TNSRTGKQHEAMTLAGQERVERKGNEVKIYGTLIRSHINPEHVTVNKGDKVTFYLTNL
ERAQDETHGFAVSGYNVHASVEPGKTVAVTFTADEEGVFPYYCTEFCSALHLEMMG
YLYVKDPKKKYESVKELKLQKMSKEQLESEYKKVIATNKATDDVIQSVVKFLKDKNYA
KYPKVKSLVEDALDQYGKIGEVKAKADESYKKGDVNGAILWEYQVWQYMVKTADVG
LRAKNNLAKELATPMKPAAQKGEEAYLKGGCNGCHVIGQVSSGPDLTGVLSRHENAE
KWVFDFIKNPASKYEEDYVKTMINYFNLRMPNQHMNDQEIKDIIEYLKWIDENAGLF

>Algoriphagus lacus

MKKNVWLLAIGLIAVALAPSCKPKGAQSAITGDAASKVYVAPGTHDEFYNIVSGGFNG
QMSVVGLPSGRVFKILPVFSVHPENGWGFSEETKPMLNTSHGFVPWDDLHHIAISMT
NAEHDARWAFGNANNTPRIARVDLTSFKTVEILEIPNSAGNHSSPFITQNTEYVVAGTR
FSVPVGENSDVSIDSYKENFKGYISFIGVDPKTGDMDLSFQIEAPPFNFDLARAGKNKS
HGWFFFSTYNTEMAHTLLEVNASQKDKDFIMAVNWKKAEEYVKAGKGKKVKAKYAH
NKMNEETHTATSTIMEEVLVLDPTELKDIVYFIPCPKSPHGTDTDPSGEYIVGSGKLAA
VIPVFSFTKMLAAIENKDFEGDVNGMPVLKYESVLHGEVEKPGLGPLHTEFDGKGFAY
TSMFVSSEIVKWNIETLEVVDRVPTYYSVGHLMIPGGPTGKPHGKYMVAYNKITKDRY
LPTGPELTQSAQLYDISGDKMQLLLDFPTIGEPHYAEAFPAALIKDKQVKFYPLEKNNH
PYALKGVADARVERKGNEIHVYMTSIRSNFRPDNIEGVNVGDVVYFHVTNLEQDWDV
PHGFSIKGANTAEILIMPGETSTYKWTPDRPGVFPFYCTDFCSALHQEMQGYLRVSPK
GVNTPLKYSTGEAQ

La secuencia en formato fasta se aportó como información base para el análisis por parte de la plataforma Phyre2 (Protein Homology/analogY Recognition Engine V 2.0).

Inicialmente el programa tomó una secuencia de aminoácidos y generó una estructura tridimensional de la proteína buscando realizar una predicción de la estructura terciaria.

La secuencia que se envía es buscada entre las secuencias homólogas mediante PSI-Blast (Position Specific Iterated Blast). En su forma básica de funcionamiento lo que hace es realizar BLAST con una secuencia y, a partir de los resultados, construye un perfil. La siguiente búsqueda la realiza con ese perfil, lo que permitirá encontrar, idealmente, nuevos homólogos, homólogos remotos. Con esos nuevos homólogos

genera un nuevo perfil, el cual, idealmente, contendrá mayor cantidad de información y podrá realizar otra búsqueda en un proceso iterativo.

Seguidamente se reúnen estas secuencias y las conserven en un HMM este modelo capturado de este modelo se utiliza algo similar a una huella dactilar de la evolución de la proteína. Paralelamente de los modelos conocidos se extrae sus secuencias, se realiza PSI-blast y se realiza un Hidden Marckov Model (HMM) de los modelos conocidos, generando una gran base de datos.

Con nuestro modelo oculto de Markov se escanea contra la matriz de matrices ocultas de Markov de la base de datos conocidos, produciendo un alineamiento que permite generar un modelo que es ajustado cuando la identidad de la secuencia es incluso menor al 15%. Intentando buscar esos plegamientos que tengan un mayor grado de reconocimiento.

El programa también incluye una simulación de plegado *ab initio* para modelar regiones sin homología detectable con estructuras conocidas.

El pipeline que sigue la plataforma incluye entre otros los siguientes pasos:

Detección de homólogos de secuencia mediante PSI-Blast.

Predicción de estructura secundaria mediante PSI-pred y Diso-pred.

Construcción de un modelo oculto de Markov de la secuencia a partir de los homólogos detectados anteriormente.

Escaneo del modelo HMM contra una biblioteca de proteínas resueltas experimentalmente.

Construcción de modelos 3D basados en las alineaciones de los modelos HMM contra los modelos HMM con proteínas conocidas.

Modelado de inserciones y delecciones utilizando una biblioteca del bucles, un procedimiento de ajustes en términos empíricos y energéticos.

Modelado de las cadenas laterales de aminoácidos utilizando una biblioteca de rotámeros.

4.7Programación script.

El objetivo en el desarrollo del presente punto se focalizó como una continuación en el desarrollo de herramientas que unificasen las anteriormente utilizadas, tanto a partir de scripts propios como en la adecuación de scripts de terceros. Si bien el término script puede ser abordado desde varias aproximaciones, la intencionalidad en este punto ha sido el intento de aglutinar de manera ordenada el conjunto de programas que se han utilizado de manera diseminada a lo largo de los puntos anteriores. Desafortunadamente no se ha podido desarrollar un código efectivo que aglutinase de manera ordenada cada uno de los pasos como consecuencia de la complejidad en el rango de lenguajes utilizados.

No obstante durante la realización de los apartados anteriores, sí se ha generado código para la realización de numerosos pasos en el proceso, habiéndose realizado parcialmente el objetivo en este punto. Por ello la culminación de este apartado se consideraría como parcialmente logrado. La adecuación de los pequeños programas que se han mostrado con anterioridad y que están completamente desarrollados en el anexo, permiten ser utilizados como punto de partida para continuar construyendo y mejorando el código.

5. Resultados

Filtrado de resultados

Para conocer las características de las secuencias de proteínas y nucleótidos y poder realizar un filtrado sobre longitud de secuencias, en el caso de tener secuencias que se alejasen sobre el promedio del tamaño que se esperaría de las secuencias se utilizó el programa `seqstats` incluido en `biosquid`. Los comandos y resultados se presentan a continuación.

```
seqstat -a proteinas_limpio.fasta > statistcs_aa.txt
```

Para secuencias de nucleótidos:

```
eqstat -a secuencias_nt.fasta > statistics_nt.txt
```

En las secuencias de nucleótidos no se encontraron secuencias para las especies *Brucella vulpis, Klebsiella pneumonia, Meiothermus hypogaeus, Methylobacterium soli, Moritella viscosa, Parvibaculum lavamentivorans, Pseudomonas knackmussii, Pseudomonas syringae* y *Pseudomonas viridiflava*.

A continuación se muestra una tabla resumen de las características de las secuencias:

Format: FASTA Type (of 1st seq): Protein	Format: FASTA Type (of 1st seq): DNA
Number of sequences: 1103 Total # residues: 721930 Smallest: 107 Largest: 875 Average length: 653.0	Number of sequences: 1093 Total # residues: 2160486 Smallest: 375 Largest: 2628 Average length: 1974.9

Tabla 1 : Resumen de características de secuencias de nucleótidos y aminoácidos

Para el alineamiento de proteínas se utilizó el comando de terminal:

```
$ muscle -in limpio.fasta -out aligned_muscle.fasta -maxiters 16 -diags -sv -distance1 kbit20_3
```

Las secuencias de nucleótidos se alinearon mediante clustalW. De cara a automatizar el proceso se utilizó un script (aligner.py, incluido en el anexo 2), para el proceso. De momento en construcción y ser irá cambiando en la fase de desarrollo del script.

El resultado del alineamiento de proteínas. Pese a indicarle por defecto un número máximo de 16 iteraciones, con 6 iteraciones completó el proceso.

```
MUSCLE v3.8.1551 by Robert C. Edgar

http://www.drive5.com/muscle

This software is donated to the public domain.

Please cite: Edgar, R.C. Nucleic Acids Res 32(5), 1792-97.
```

```
limpio 1103 seqs, lengths min 107, max 875, avg 653
00:00:12    23 MB(-1%)  Iter   1  100.00%  K-bit distance matrix
00:00:29  456 MB(-24%)  Iter   1  100.00%  Align node
00:00:29  459 MB(-25%)  Iter   1  100.00%  Root alignment
00:00:56  461 MB(-25%)  Iter   2  100.00%  Refine tree
00:00:57  461 MB(-25%)  Iter   2  100.00%  Root alignment
00:00:57  461 MB(-25%)  Iter   2  100.00%  Root alignment
00:16:45  461 MB(-25%)  Iter   3  100.00%  Refine biparts
00:33:59  461 MB(-25%)  Iter   4  100.00%  Refine biparts
00:51:36  461 MB(-25%)  Iter   5  100.00%  Refine biparts
01:00:30  461 MB(-25%)  Iter   6  100.00%  Refine biparts
```

Gblocs

En el caso de secuencias alineadas de aminoácidos, fue necesario un paso previo de transformación del archivo phylip a formato fasta con nombres reducidos para que el programa no abortase. El resultado de la ejecución de Gblocks generó un archivo informativo en formato html, el cual no dio indicio de eliminación de posiciones en aminoácidos reduciendo el tamaño de bloques tal y como se muestra en la siguiente captura de pantalla:

```
Parameters used
Minimum Number Of Sequences For A Conserved Position: 553
Minimum Number Of Sequences For A Flanking Position: 938
Maximum Number Of Contiguous Nonconserved Positions: 8
Minimum Length Of A Block: 10
Allowed Gap Positions: None
Use Similarity Matrices: Yes

Flank positions of the 0 selected block(s)
Flanks:

New number of positions in aligned_muscle.fasta-gb:  0  (0% of the original 1443 positions)
```

Figura 11: Resultado Gblocs secuencias en aminoácidos

En el caso de secuencias alineadas de nucleótidos, el programa nuevamente no identificó con las condiciones estándar, la necesidad de realizar ningún bloque de agrupamiento, tal y como se muestra en la siguiente captura de pantalla:

```
Parameters used
Minimum Number Of Sequences For A Conserved Position: 547
Minimum Number Of Sequences For A Flanking Position: 929
Maximum Number Of Contiguous Nonconserved Positions: 8
Minimum Length Of A Block: 10
Allowed Gap Positions: None
Use Similarity Matrices: Yes


Flank positions of the 0 selected block(s)
Flanks:

New number of positions in aligned_secuencias_nt_sl2.fasta-gb:  0  (0% of the original 3290 positions)
```

Figura 12: Resultado Gblocks secuencias nucleótidos

Señal filogenética con TreePuzzle

Los resultados de búsqueda de señal filogenética en aminoácidos no presentaron ningún problema. En nucleótidos, en cambio, sí que hubo problema, por lo que se identificaron las secuencias problemáticas para poner un nombre temporal y poder controlarlas en las siguientes fases del proceso.

En el lanzamiento de TreePuzzle en el terminal se modificaron las siguientes opciones:

```
GENERAL OPTIONS (no quartet puzzling available - too many sequences)
 b                 Type of analysis?  Tree reconstruction
 k              Tree search procedure?  Evaluate user defined trees
 z     Compute clocklike branch lengths?  No
 e               Parameter estimates?  Approximate (faster)
 x          Parameter estimation uses?  1st input tree
SUBSTITUTION PROCESS
 d         Type of sequence input data?  Auto: Amino acids
 m               Model of substitution?  Auto: WAG (Whelan-Goldman 2000)
 f             Amino acid frequencies?  Estimate from data set
RATE HETEROGENEITY
 w         Model of rate heterogeneity?  Uniform rate
Quit [q], confirm [y], or change [menu] settings: b
```

De manera que el tipo de análisis fue el siguiente

```
b                 Type of analysis?  Likelihood mapping
```

Ya con estos settings establecidos se ejecutó el programa

A continuación se presentan los resultados para proteínas y nucleótidos

Resultados en terminal para proteínas

```
Writing parameters to file outfile
Writing pairwise distances to file outdist
Performing likelihood mapping analysis


All results written to disk:
      Puzzle report file:         outfile
      Likelihood distances:       outdist
      Likelihood mapping diagram: outlm.eps


The parameter estimation took 0.00 seconds (= 0.00 minutes = 0.00 hours)
The ML step took              0.00 seconds (= 0.00 minutes = 0.00 hours)
The puzzling step took        0.00 seconds (= 0.00 minutes = 0.00 hours)
The computation took 737.00 seconds (= 12.28 minutes = 0.20 hours)
    including input 784.00 seconds (= 13.07 minutes = 0.22 hours)
```

La ocupación de las áreas en el primer gráfico de distribución de puntos se reparte hacia las esquinas desde el centro. Por otro lado la ocupación de las áreas es uniforme en un 33% y a nivel de cuartetos para tres regiones y para cuartetos en siete regiones se reparten nuevamente en los cuartetos de los extremos (con un porcentaje alrededor del 32-33% lo que indica una buena señal).

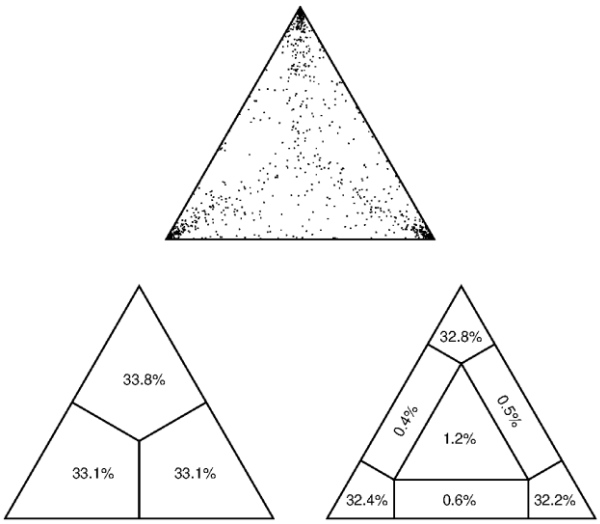


Figura 13: Resultados gráficos TreePuzzle para secuencias de aminoácidos

```
Performing likelihood mapping analysis


All results written to disk:
      Puzzle report file:         outfile
      Likelihood distances:       outdist
      Likelihood mapping diagram: outlm.eps


The parameter estimation took 77280.00 seconds (= 1288.00 minutes = 21.47 hours)
The ML step took              0.00 seconds (= 0.00 minutes = 0.00 hours)
The puzzling step took        0.00 seconds (= 0.00 minutes = 0.00 hours)
The computation took 78378.00 seconds (= 1306.30 minutes = 21.77 hours)
    including input 78407.00 seconds (= 1306.78 minutes = 21.78 hours)
```

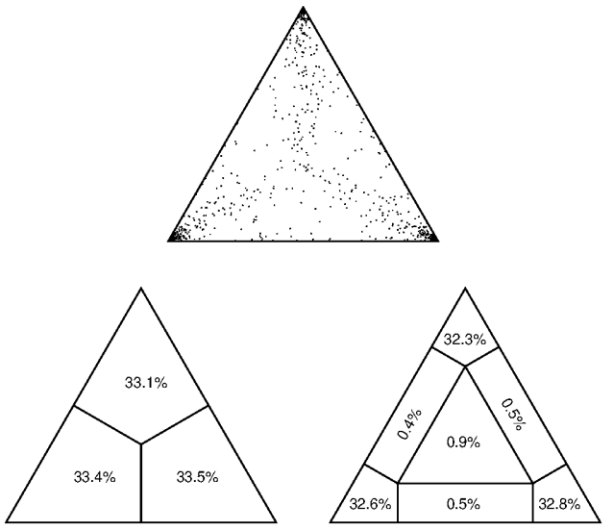


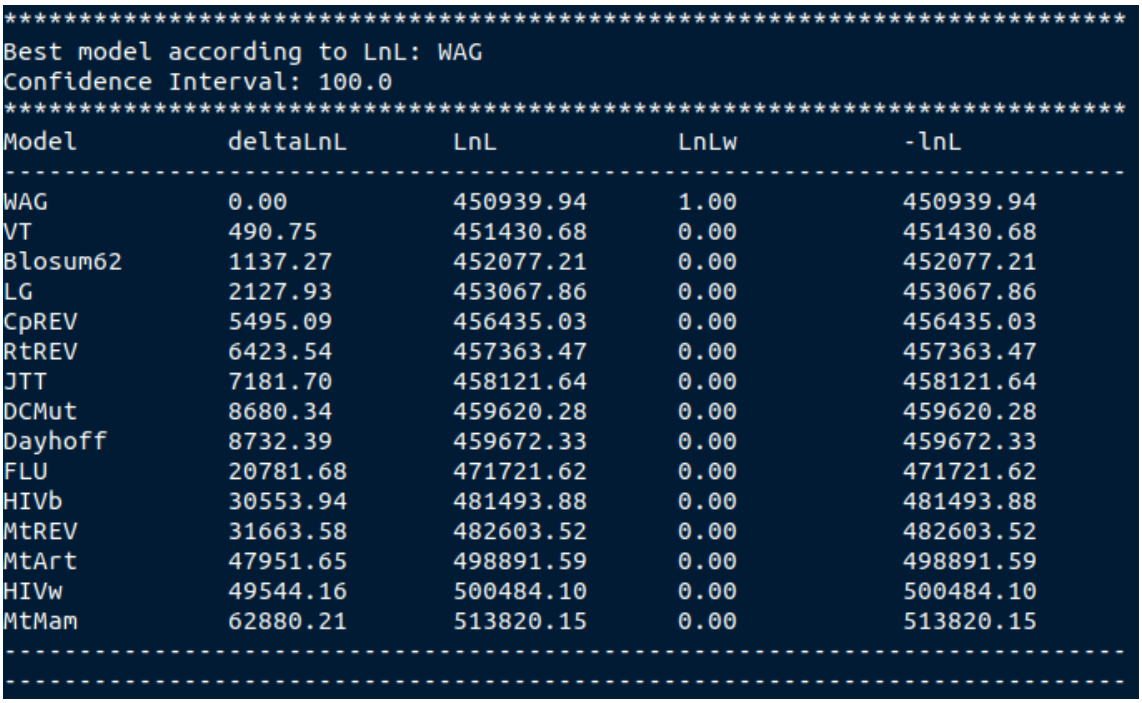
Figura 14: Resultados TreePuzzle para secuencias de nucleótidos

Modelos evolutivos

Para la búsqueda de modelos evolutivos en secuencias de aminoácidos el uso del programa Prottest, utilizando los criterios de selección BIC y LnL se presentan a continuación sendas figuras con ambos criterios.

```
**************************************************************************
Best model according to BIC: WAG
Confidence Interval: 100.0
**************************************************************************
Model          deltaBIC       BIC            BICw           -lnL
--------------------------------------------------------------------------
WAG            0.00           917920.11      1.00           450939.94
VT             981.49         918901.60      0.00           451430.68
Blosum62       2274.53        920194.64      0.00           452077.21
LG             4255.85        922175.96      0.00           453067.86
CpREV          10990.18       928910.28      0.00           456435.03
RtREV          12847.07       930767.18      0.00           457363.47
JTT            14363.40       932283.51      0.00           458121.64
DCMut          17360.69       935280.79      0.00           459620.28
Dayhoff        17464.78       935384.89      0.00           459672.33
FLU            41563.35       959483.46      0.00           471721.62
HIVb           61107.89       979027.99      0.00           481493.88
MtREV          63327.16       981247.27      0.00           482603.52
MtArt          95903.31       1013823.41     0.00           498891.59
HIVw           99088.31       1017008.42     0.00           500484.10
MtMam          125760.42      1043680.53     0.00           513820.15
--------------------------------------------------------------------------
--------------------------------------------------------------------------
```

Figura 15: Resultados del modelo evolutivo aminoácidos con criterio BIC



```
************************************************************************
Best model according to LnL: WAG
Confidence Interval: 100.0
************************************************************************
Model          deltaLnL       LnL            LnLw           -lnL
------------------------------------------------------------------------
WAG            0.00           450939.94      1.00           450939.94
VT             490.75         451430.68      0.00           451430.68
Blosum62       1137.27        452077.21      0.00           452077.21
LG             2127.93        453067.86      0.00           453067.86
CpREV          5495.09        456435.03      0.00           456435.03
RtREV          6423.54        457363.47      0.00           457363.47
JTT            7181.70        458121.64      0.00           458121.64
DCMut          8680.34        459620.28      0.00           459620.28
Dayhoff        8732.39        459672.33      0.00           459672.33
FLU            20781.68       471721.62      0.00           471721.62
HIVb           30553.94       481493.88      0.00           481493.88
MtREV          31663.58       482603.52      0.00           482603.52
MtArt          47951.65       498891.59      0.00           498891.59
HIVw           49544.16       500484.10      0.00           500484.10
MtMam          62880.21       513820.15      0.00           513820.15
------------------------------------------------------------------------
------------------------------------------------------------------------
```

Figura 16: Resultados modelo evolutivo aminoácidos con criterio LnL

Ambos criterios dieron como el mejor modelo evolutivo el modelo Whelan And Goldman (30).

Para la selección de modelos evolutivos en secuencias de nucleótidos, pasadas 50 horas de análisis el avance fue del 0%. Se abortó el proceso y se estableció una lista con el 25% (275 secuencias de las inicialmente descargadas) alrededor de la media de longitud de las secuencias (653 nucleótidos), de manera que se asumió que podía existir un mínimo sesgo en la representación, por aquellas especies que pudieran ser más abundantes en el listado principal. El documento anexo1.xlxs incluye en la pestañan Reduced_NT un listado del resumen de las especies seleccionadas A continuación se presenta el análisis de las secuencias que se utilizaron:

Format:	FASTA
Type (of 1st seq):	DNA
Number of sequences:	275
Total # residues:	538024
Smallest:	1900
Largest:	1974
Average length:	1956.5

Tabla 2 : Características de listado reducido de secuencias de nucleótidos seleccionados

Se volvió a realizar el alineamiento con ClustalW, eliminando las líneas con “*” para poder transformar el archivo en formato phylip y se procedió a revisar la existencia de señal filogenética con TreePuzzle. Nuevamente se siguió teniendo señal filogenética:

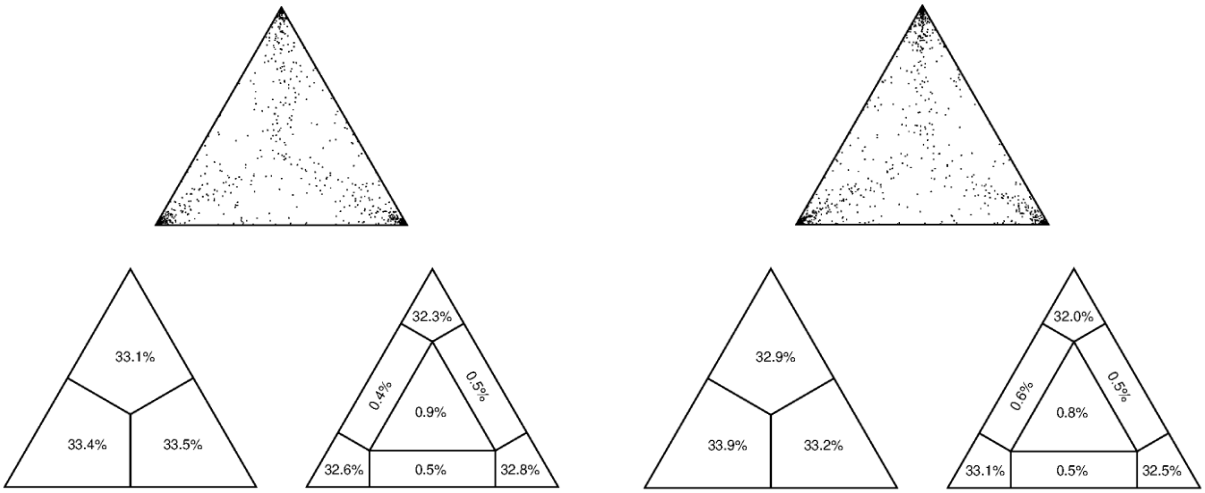


Figura 17: Comparación resultados TreePuzzle para nucleótidos. A la izquierda los resultados con todo el dataset, a la derecha con el dataset reducido a 275 especies.

El resultado del análisis de las secuencias de nucleótidos alineadas tras el análisis con jModeltest para 275 especies una vez completado, se analizó mediante los criterios de AIC (Akaike's information criterion) y el criterio BIC (Bayessian information criterion). En este paso el objetivo fue obtener el modelo evolutivo que mejor se adaptase a los datos de cara a su análisis posterior mediante el programa MrBayes. La siguiente tabla resume los resultados de los análisis mediante el criterio de AIC:

ID	Name	Partition	-lnL	p	AIC	deltaAIC	weight	cumWeight	uDelta
1	JC	000000	304468,0055	548	610032011	73278.8251	0.0	1.0	-
2	JC+I	000000	298703,0308	549	598504.0616	61750.8757	0.0	1.0	-
3	JC+G	000000	274089,6679	549	549277.3359	12524.15	0.0	1.0	-
4	JC+I+G	000000	273732,5565	550	548565.1129	11811927	0.0	1.0	-
5	F81	000000	304311,1522	551	609724.3044	72971.1185	0.0	1.0	-
6	F81+I	000000	298450,3335	552	598004667	61251.4811	0.0	1.0	-
7	F81+G	000000	273570,9576	552	548245.9152	11492.7293	0.0	1.0	-
8	F81+I+G	000000	273208,2023	553	547522.4046	10769.2187	0.0	1.0	-
9	K80	10010	301690,0179	549	604478.0357	67724.8498	0.0	1.0	-
10	K80+I	10010	295857,9063	550	592815.8126	56062.6267	0.0	1.0	-
11	K80+G	10010	270882389	550	542864778	6111.5921	0.0	1.0	-
12	K80+I+G	10010	270491,7006	551	542085.4012	5332.2153	0.0	1.0	-
13	HKY	10010	301450,1238	552	604004.2475	67251.0616	0.0	1.0	-
14	HKY+I	10010	295585,7105	553	592277421	55524.2351	0.0	1.0	-
15	HKY+G	10010	270338,0298	553	541782.0595	5028.8736	0.0	1.0	-
16	HKY+I+G	10010	269938,08	554	540984.1599	4230974	0.0	1.0	-
17	SYM	12345	299673,4465	553	600452893	63699.7071	0.0	1.0	-
18	SYM+I	12345	293908,0643	554	588924.1286	52170.9427	0.0	1.0	-
19	SYM+G	12345	269248,9648	554	539605.9297	2852.7438	0.0	1.0	-
20	SYM+I+G	12345	268883,7486	555	538877.4972	2124.3113	0.0	1.0	-
21	GTR	12345	299189,2204	556	599490.4407	62737.2548	0.0	1.0	-
22	GTR+I	12345	293371,2194	557	587856.4388	51103.2529	0.0	1.0	-

23	GTR+G	12345	268144,8753	557	537403.7506	650.5647	0.0	1.0	-
24	GTR+I+G	12345	267818593	558	536753.1859	0.0	1.0	1.0	-

Tabla 3: Modelos evolutivos evaluados mediante el criterio AIC

El resultado para este criterio resolvió el modelo GTR+I+G (Generalised Time Reversible con sitios Invariantes y variaciones en distribución gamma entre sitios (31)) como el que mejor resolvería la relación entre las secuencias a analizar. Además de este resultado, también se realizaron los cálculos de los resultados mediante el criterio BIC que se resumen en la siguiente tabla:

ID	Name	Partition	-lnL	p	AIC	deltaBIC	weight	cumWeight	uDelta
1	JC	000000	304482.9481	548	613211.1059	73252.9259	0.0	1.0	-
2	JC+I	000000	298680.1751	549	601613.3066	61655.1266	0.0	1.0	-
3	JC+G	000000	274088.2588	549	552429.474	12471.294	0.0	1.0	-
4	JC+I+G	000000	273732.3059	550	551725.315	11767.135	0.0	1.0	-
5	F81	000000	304276.6473	551	612821.7444	72863.5644	0.0	1.0	-
6	F81+I	000000	298424.8702	552	601125.937	61167.757	0.0	1.0	-
7	F81+G	000000	273594.1426	552	551464.4817	11506.3017	0.0	1.0	-
8	F81+I+G	000000	273205.5528	553	550695.0488	10736.8689	0.0	1.0	-
9	K80	010010	301701.7976	549	607656.5516	67698.3716	0.0	1.0	-
10	K80+I	010010	295884.1941	550	596029.0913	56070.9113	0.0	1.0	-
11	K80+G	010010	270881.7833	550	546024.2697	6066.0897	0.0	1.0	-
12	K80+I+G	010010	270489.3032	551	545247.0563	5288.8763	0.0	1.0	-
13	HKY	010010	301453.554	552	607183.3046	67225.1246	0.0	1.0	-
14	HKY+I	010010	295573.3318	553	595430.607	55472.427	0.0	1.0	-
15	HKY+G	010010	270333.9305	553	544951.8042	4993.6243	0.0	1.0	-
16	HKY+I+G	010010	269947.6649	554	544187.0198	4228.8398	0.0	1.0	-
17	SYM	012345	299668.7035	553	603621.3504	63663.1704	0.0	1.0	-
18	SYM+I	012345	293879.5602	554	592050.8105	52092.6305	0.0	1.0	-
19	SYM+G	012345	269248.9241	554	542789.5383	2831.3583	0.0	1.0	-
20	SYM+I+G	012345	268887.5199	555	542074.4766	2116.2967	0.0	1.0	-
21	GTR	012345	299191.0327	556	602689.2489	62731.069	0.0	1.0	-
22	GTR+I	012345	293368.9059	557	591052.7421	51094.5621	0.0	1.0	-
23	GTR+G	012345	268134.0079	557	540582.9461	624.7661	0.0	1.0	-
24	GTR+I+G	012345	267817.7515	558	539958.18	0.0	1.0	1.0	-

Tabla 4: Modelos evolutivos evaluados mediante el criterio BIC

Nuevamente, el análisis de resultados concluyó que el modelo GTR+I+G fue el más adecuado para el análisis posterior. Esta unicidad en la selección de criterios es importante tenerla en cuenta, dado el hecho que una discrepancia entre ambos criterios necesitaría de una evaluación posterior. Si bien la diferencia entre uno y otro criterio, de manera sucinta, difieren en el coste en la penalización. En el criterio de AIC dado un tamaño de muestra n puede tomar modelos más grandes, más complejos, que el criterio BIC. En este caso ambos resultados convergen, por lo que nos refuerzan la idea que el modelo GTR con gamma y sitios invariantes sería el modelo adecuado.

Análisis de secuencias ML.

Tras 22 días de ejecución del análisis se completó el proceso y se obtuvo un archivo tipo newik con la topología. Debido a que el archivo newik contenía únicamente las identificaciones sin tener una correspondencia con los nombres de las especies, de manera que se evaluaron alternativas para poder obtener una topología que pudiese ser interpretable.

La solución finalmente encontrada fue el uso del script taxnameconverter.pl del Center for Integrative Bioniformatics Vienna desarrollado por H.A. Schmidt y accesible en la siguiente página web (http://www.cibiv.at/software/taxnameconvert/). El script necesita un archivo de tipo csv, con el listado de correspondencias a reemplazar y el archivo de la topología resultante del análisis filogenético realizado. Para ello se hizo uso del anexo I que contenía la información necesaria y se utilizaron las columnas del identificador y el binomio de la especie correspondiente.

Dado que el archivo newick únicamente contenía los 10 primeros caracteres de la cadena de texto, se utilizó el siguiente comando para tener una lista correctamente manipulable:

```
$ awk -F'\t' '{ print substr($1,1,10)"\t"$2}' lista.csv > lista_aa.csv
```

Una vez comprobada que la lista se generó correctamente se revisó que no hubiese retornos de carro al final del documento ya que el script es sensible a ese tipo de errores en la manipulación de las listas. Finalmente el comando utilizado para el reemplazo de los IDs por los nombres de las especies fue el siguiente:

```
$ perl taxnameconvert.pl -f 1 -t 2 -Q 0 lista_aa.csv RAxML_bestTree.TEST > RAxML_bestTree.nwk
```

A la hora de visualizar el archivo se propuso inicialmente el uso de la herramienta Figtree, sin embargo el programa no soportaba la realización de una apertura en el campo par la visualización en un campo mayor de las especies listadas. Esto controbuyó a la búsqueda de un software alternativo para la visualización. El programa elegido fue Dendroscope, versión 3.7.5 desarrollado por Daniel H. Huson (32). Esta herramienta permitió poder maximizar en un rango mayor la topología facilitando la exportación de un archivo en formato .png donde se pudiese observar con mayor detalle las especies renombradas. Sin embargo y debido al tamaño de la topología no se pudo visualizar en la imagen el total de las especies, si bien sí se pudo obtener una información acorde al esfuerzo realizado en la topología.

Finalmente la imagen resultante se procesó con el programa Gimp para mejorar la representación pudiendo incluir capas que facilitasen la información observable en la topología, como la discriminación entre los clados I y II descritos ampliamente en la bibliografía existente.

A continuación se presenta una versión reducida de la misma, si bien se aporta como documento anexo una imagen que puede ser visualizada a tamaño real.

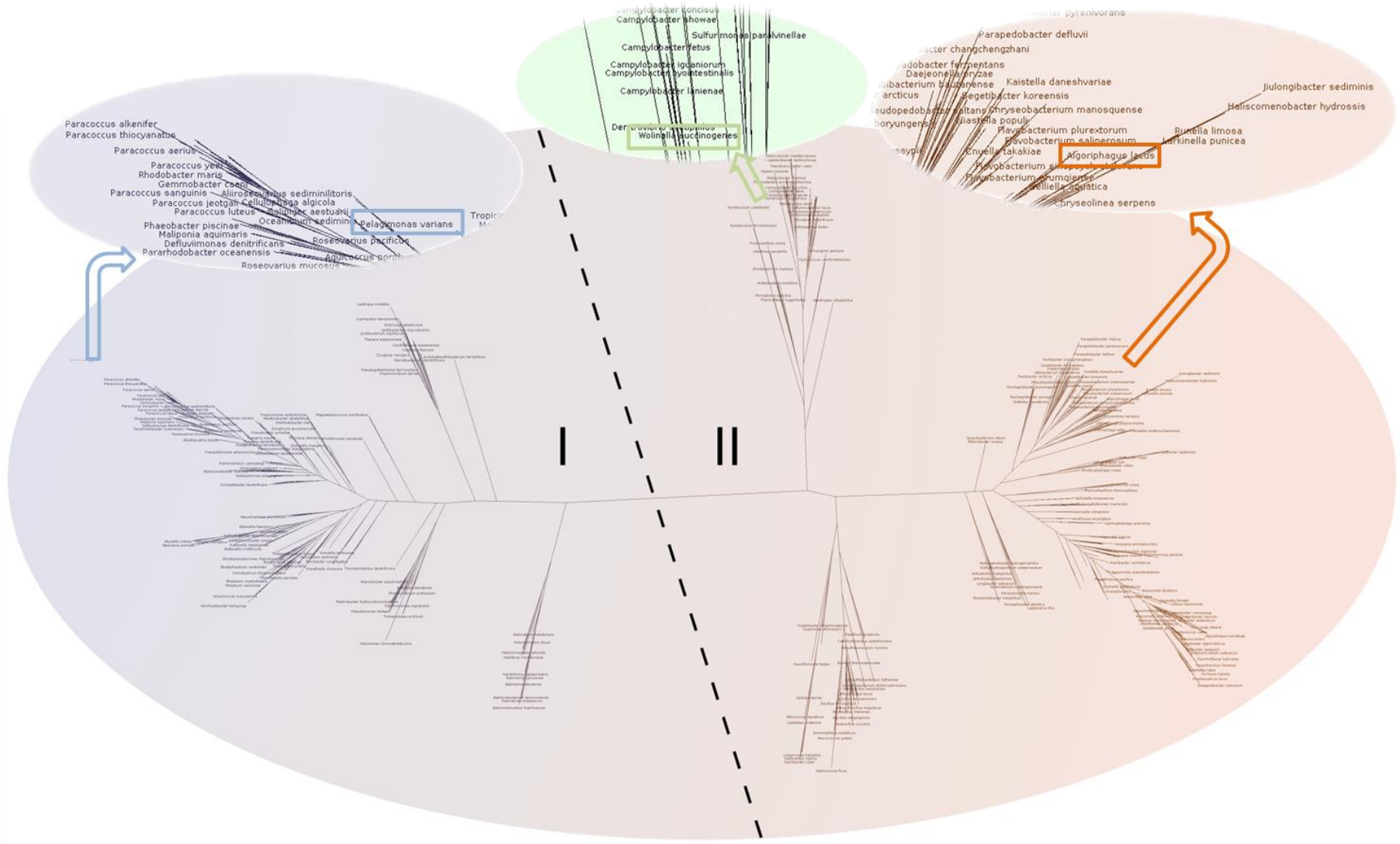


Figura 18: Árbol filogenético obtenido mediante maximum likelyhood

Análisis de secuencias Bayes.

Con 13 millones de generaciones y habiendo comentado con anterioridad la falta de convergencia en algunas de las métricas se decidió no continuar con el análisis ya que no se observó tendencia a converger. Si bien durante el desarrollo del ejercicio el valor del LnL dio un salto y convergieron las dos cadenas, no fue hasta pasados más de 8 millones de generaciones que se produjo un salto en la convergencia, si bien el resto de parámetros, LnPr, TL y m{1} (el valor log prior probability log(p(θ)), el total tree length y tasa específica de la partición o ratemult), continuaron con una falta de convergencia y soporte. Se evaluó la idoneidad de realizar un descarte mayor de información, con niveles de hasta el 75% de la información generada por el análisis, pero surgieron nuevamente falta de convergencia para otros parámetros, por lo que se vio como inviable la continuidad del ejercicio con tan solo un 25% de las secuencias originalmente identificadas, filtradas, y seleccionadas, pese a tener señal filogenética. A continuación se incluyen capturas de pantalla de los resultados de Tracer para los parámetros no convergentes:

El LnL dio un salto entre los dos runs a partir de 8.1 millones de generaciones, y para no incluir esta información la información de todas estas generaciones no podría ser utilizada ya que conlleva resultados discrepantes en los siguientes puntos.

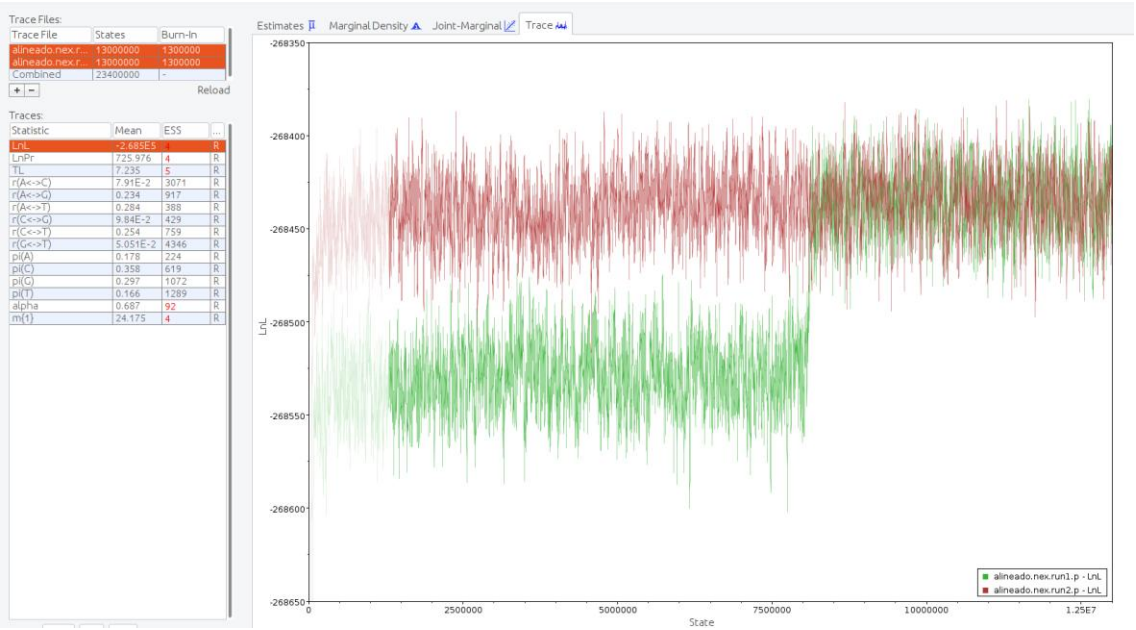


Figura 19: Representación en Tracer, de los valores de LnL para los dos runs, descartando el 10% de los resultados iniciales. Se observa cómo a partir de 8*10^6 se produjo convergencia de los dos runs.

Los valores de LnPr se mantuvieron estables en uno de los runs, mientras que en el otro incrementaron la distancia frente a la cadena estable sin observarse una tendencia a la convergencia

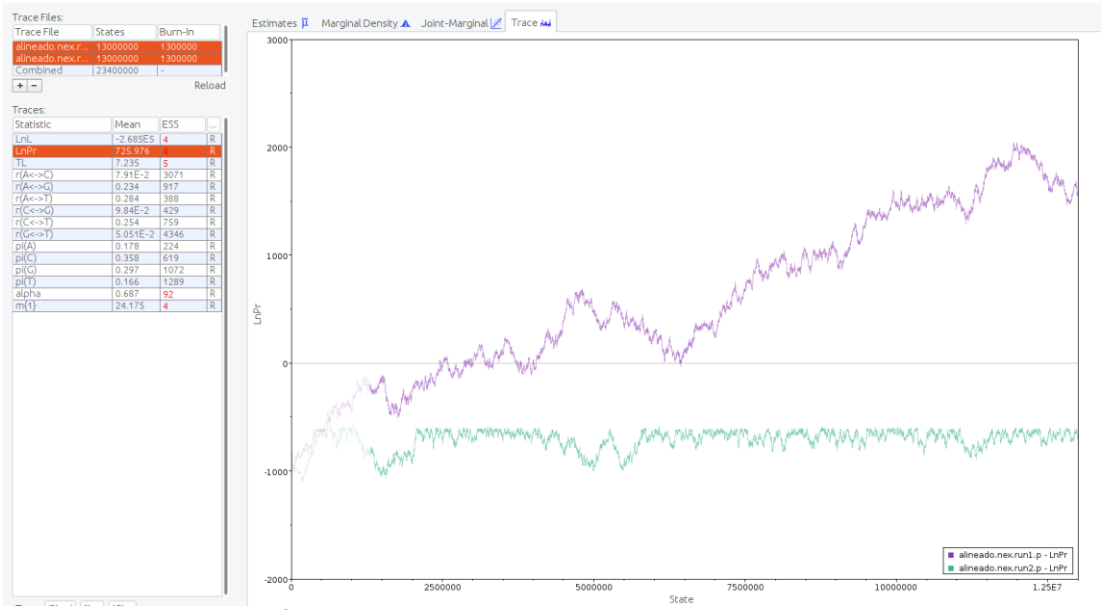


Figura 20: Representación en Tracer, de los valores de LnPr para los dos runs, descartando el 10% de los resultados iniciales. Se observa la no convergencia de los runs, con uno de los runs estables y el otro con tendencia a separarse.

Los valores para TL se estabilizaron en una distancia que se estabilizó a partir del 10 millones de generaciones con oscilaciones pero sin cambiar de los umbrales en cada run.

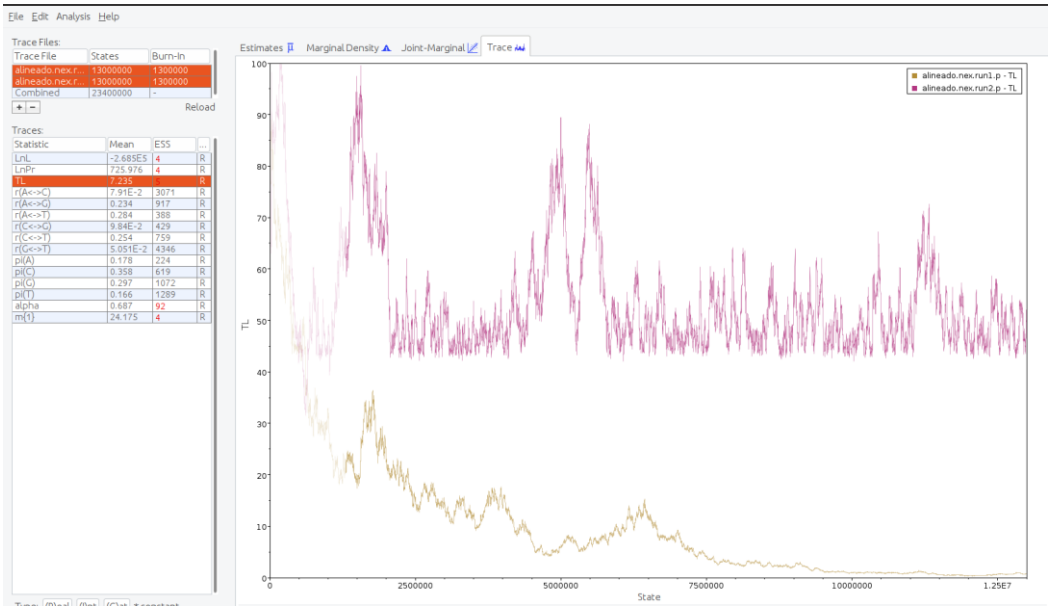


Figura 21: Representación en Tracer, de los valores TL para los dos runs, descartando el 10% de datos iniciales Se aprecia la estabilización de los dos runs sin obtenerse convergencia.

Los valores para m{1} estuvieron muy próximos a 0 en uno de los runs y el otro tuvo tendencias ascendentes y descendentes, pero sin una tendencia clara a la convergencia.

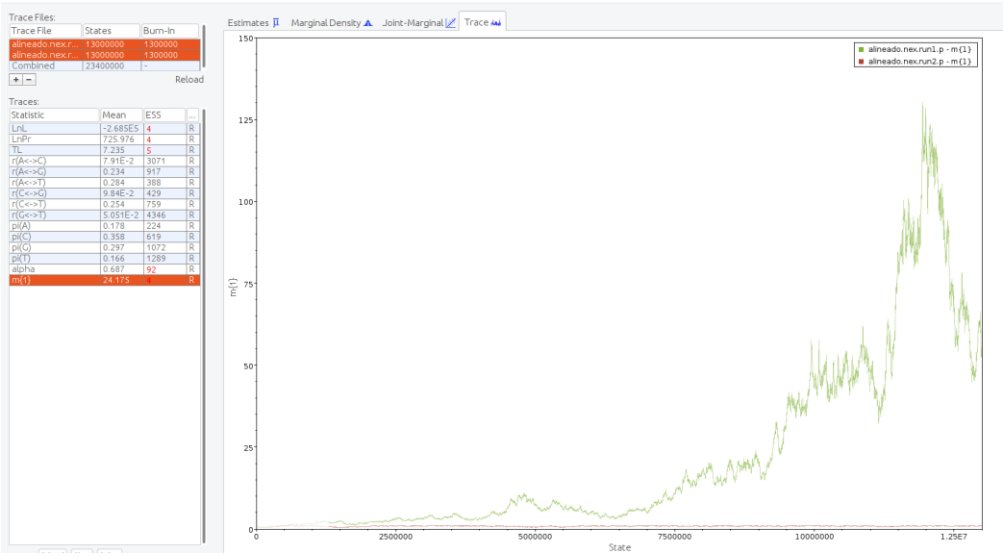


Figura 22: Representación con Tracer, de los valores para m{1}, descartando el 10% de los datos iniciales. Se aprecia la estabilización de uno de los runs mientras sin producirse convergencia entre ambos.

Estos resultados y el tiempo de computación empleado supusieron la toma de decisión de no continuar más allá y proceder a la representación de la topología final.

Para ello se realizó de manera similar al apartado de ML, obteniendo un listado de identificadores y de especies, en este caso a partir del listado reducido incluido en el anexo I, y generando una archivo .csv. Se aplicó el mismo comando para reducir el número de caracteres de la columna del identificador a 10 y con el scritp taxnameconvert.pl se procedió al reemplazo de las etiquetas del archivo newik obtenido al archivo de topología con extensión .nex.con.tre generado por el programa MrBayes.

La visualización nuevamente se realizó con el programa Dendroscope, el cual permitió la realización de una ampliación del área de trabajo para observar los nombres de las especies. A continuación se incluye una imagen del resultado obtenido.

La imagen mostró una elevada distancia entre taxones del clado I y del clado II y un fuerte agrupamiento dentro de los clados. Entre ambos grupos quedaron algunas especies que si bien se debería evaluar si se produjo un fenómeno de atracción de ramas largas (traducido del inglés del fenómeno conocido como long branch attraction). La atracción de ramas largas ocurre porque cuando acontecen cambios de estado relativamente numerosos a lo largo de los linajes, los cambios aleatorios pueden comenzar a pesar más que los no aleatorios, filogenéticamente informativos. La ubicación filogenética de un taxón con una rama larga puede ser incierta y puede influir indebidamente en la ubicación de otros taxones.

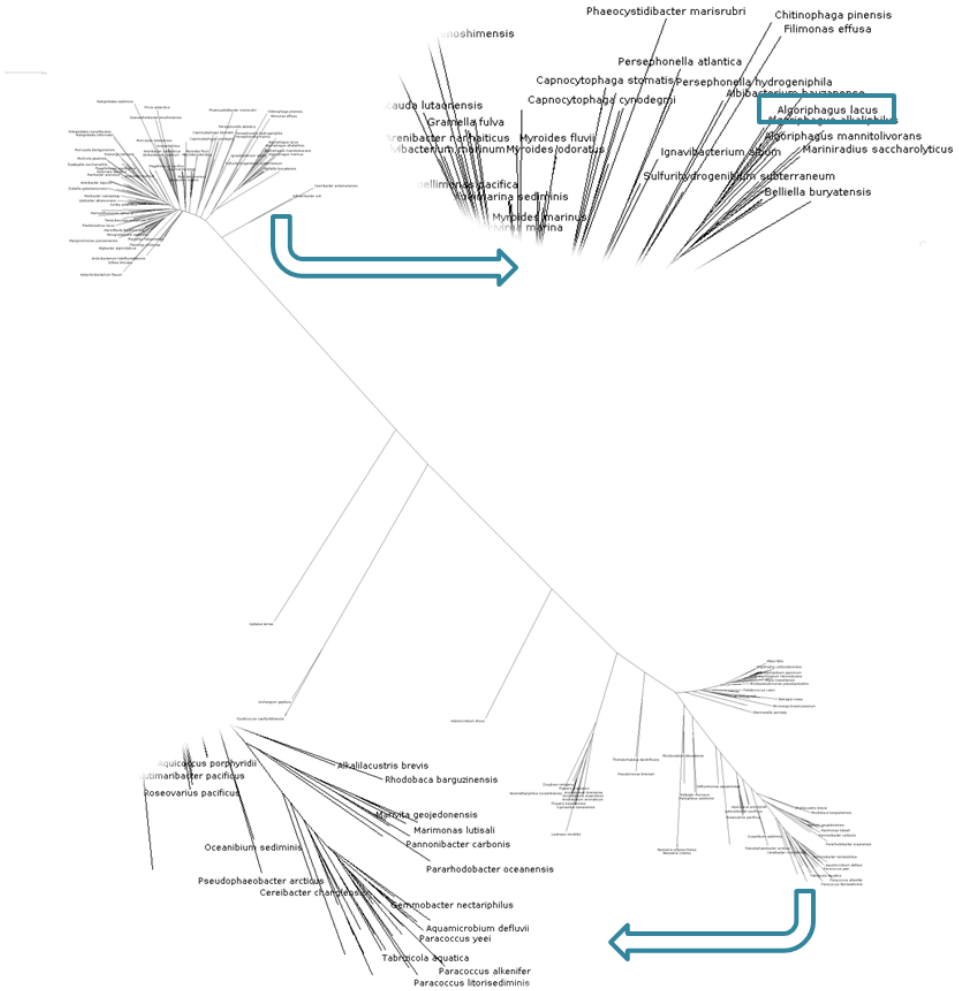


Figura 23: Árbol filogenético de secuencias nucleotídicas por inferencia bayesiana

Taxonomía de las secuencias.

Este es un apartado que no se trató en la memoria de contenido, pero resultó necesario incorporar. A partir de la lista de taxones que tenemos en el documento anexo, se copió la lista de especies y se pegó en la siguiente dirección web.

https://www.ncbi.nlm.nih.gov/Taxonomy/TaxIdentifier/tax_identifier.cgi

Como resultado devolvió un documento que contenía los identificadores para cada uno de los niveles taxonómicos, comenzando por especie. Aquellos que no devolvieron una taxonomía se anotaron para buscar información que pudiese incluirse de manera manual. En total resultaron 4 casos y no de todos ellos se pudo obtener información concluyente sobre su taxonomía, si bien fueron identificados como bacterias.

Separando por los pipes(|) el contenido del archivo, se pudo obtenerla columna correspondiente a los niveles taxonómicos y se sacó la primera columna correspondiente a las especies. A partir de ahí se hizo uso del script taxonomy.py (incluido en el anexo 2).

Ejecutado el script, indicado como output que derive a un archivo .csv, se pudo incluir la información y alinear línea a línea con el documento de hoja de cálculo y tener toda la taxonomía, descubrí que por error se introdujeron 3 especies de moluscos, a un nivel tan adelantado, parar todo el proceso no tiene sentido, si bien se justificará (taxones pintados en rojo en la pestaña). Si bien es un resultado que no se esperaba atendiendo a las constricciones empleadas en pasos anteriores, se decidió optar por seguir a sabiendas que entre todas secuencias de especies diferentes 3 resultaron contaminantes. Estas secuencias no aparecieron en la selección de las secuencias de nucleótido. Un resumen de toda la información está incorporada en el documento Anexo1.xlsx, pestaña taxonomy.

Establecimiento perfil de la proteína.

Una vez realizado el análisis en Phyre2 utilizando la secuencia de aminoácidos para *Wolinella succinogenes*, que atendiendo a la bibliografía presentan alternativas en el proceso de desnitrificación (27), el programa devolvió varios templates:

C2iwkB_: Se trata de una forma unida al inhibidor de óxido nitroso reductasa de 2 cicloclastos de Achromobacter con una resolución de 1,7 angstrom.

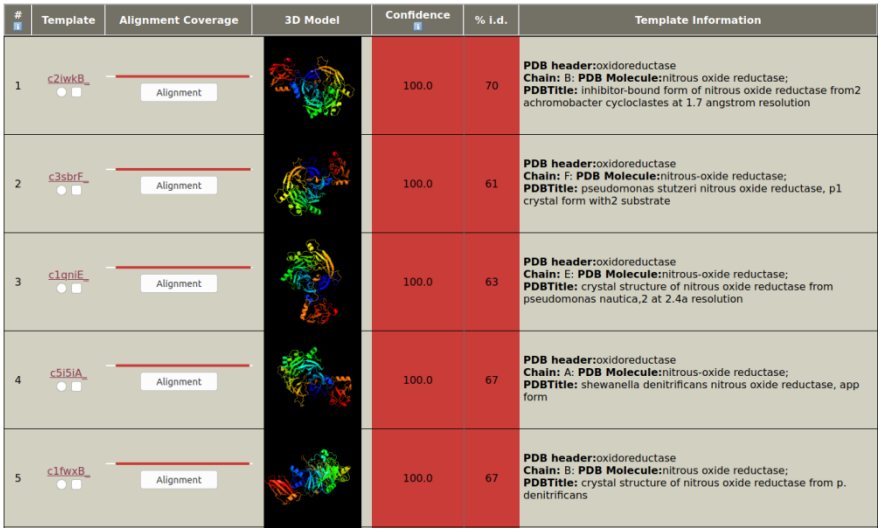
C3ssbrF_: Se trata de una óxido nitroso reductasa, forma cristalina p1 con sustrato 2 para *Pseudomonas stutzeri*.

C1qniE_: Se trata de una estructura cristalina de óxido nitroso reductasa de *Pseudomonas nautica*, 2 a una resolución de 2.4Å.

C5i5iA_: Identificada como una óxido reductasa de *Shewanella denitrificans*.

C1fwxB_: Identificada como estructura cristalina de óxido nitroso reductasa de *P. desnitrificans*.

A continuación se presenta una captura de pantalla de los resultados obtenidos:



#	Template	Alignment Coverage	3D Model	Confidence	% i.d.	Template Information
1	c2iwkB_	Alignment		100.0	70	**PDB header:**oxidoreductase **Chain:** B: **PDB Molecule:**nitrous oxide reductase; **PDBTitle:** inhibitor-bound form of nitrous oxide reductase from2 achromobacter cycloclastes at 1.7 angstrom resolution
2	c3sbrF_	Alignment		100.0	61	**PDB header:**oxidoreductase **Chain:** F: **PDB Molecule:**nitrous-oxide reductase; **PDBTitle:** pseudomonas stutzeri nitrous oxide reductase, p1 crystal form with2 substrate
3	c1qniE_	Alignment		100.0	63	**PDB header:**oxidoreductase **Chain:** E: **PDB Molecule:**nitrous-oxide reductase; **PDBTitle:** crystal structure of nitrous oxide reductase from pseudomonas nautica,2 at 2.4a resolution
4	c5i5iA_	Alignment		100.0	67	**PDB header:**oxidoreductase **Chain:** A: **PDB Molecule:**nitrous-oxide reductase; **PDBTitle:** shewanella denitrificans nitrous oxide reductase, app form
5	c1fwxB_	Alignment		100.0	67	**PDB header:**oxidoreductase **Chain:** B: **PDB Molecule:**nitrous oxide reductase; **PDBTitle:** crystal structure of nitrous oxide reductase from p. denitrificans

Figura 24: Resumen de los cinco primeros templates obtenidos para *Wolinella succinogenes*

En todos los casos la identidad se presentó con una oxidoreductasa con una confianza en los cinco hits de un 100%, indicando una elevada homología entre la secuencia aportada frente a los resultados propuestos.

A nivel de identidad, los valores también fueron elevados con un porcentaje que osciló entre un 70 y un 67%, siendo unos valores considerablemente aceptables.

El modelo obtenido tuvo una confianza del 100% y una cobertura sobre el modelo del 92%, sobre el total de 585 residuos.

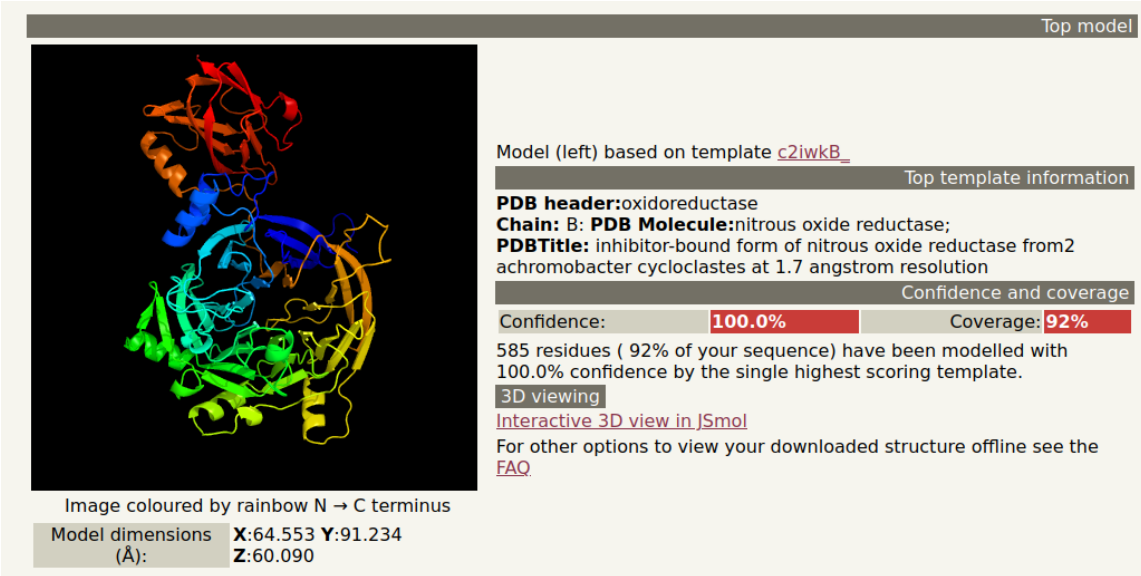


Figura 25: Resumen del mejor modelo de predicción obtenido para *Wolinella succinogenes*

Para complementar la información se buscó en la página web de SCOP la familia a la que pertenece y el resultado fue que el fold 20011013 7-blade beta-propeller como ancestro, que incluía las supefamilia y familia de la Nitrous oxido reductasa, dominio N-terminal (superfamilia 30016874 y familia 4002739).

Tras la primera aproximación con una especie que en la literatura había presentado particularidades, se procedió a la selección de dos especies, una por clado para buscar la existencia en la detección de diferencias en uno y otro clado. Se seleccionarlon las especies *Pelagimonas varians* y *Algoriphagus lacus* correspondientes al clado I y clado II respectivamente.

En el caso de *Pelagimona varians* y una vez realizado el análisis, el programa devolvió varios templates:

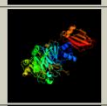
#	Template	Alignment Coverage	3D Model	Confidence	% i.d.	Template Information
1	c2iwkB_	Alignment		100.0	71	**PDB header:**oxidoreductase **Chain:** B: **PDB Molecule:**nitrous oxide reductase; **PDBTitle:** inhibitor-bound form of nitrous oxide reductase from2 achromobacter cycloclastes at 1.7 angstrom resolution Run Investigator
2	c5i5iA_	Alignment		100.0	65	**PDB header:**oxidoreductase **Chain:** A: **PDB Molecule:**nitrous-oxide reductase; **PDBTitle:** shewanella denitrificans nitrous oxide reductase, app form Run Investigator
3	c3sbrF_	Alignment		100.0	59	**PDB header:**oxidoreductase **Chain:** F: **PDB Molecule:**nitrous-oxide reductase; **PDBTitle:** pseudomonas stutzeri nitrous oxide reductase, p1 crystal form with2 substrate Run Investigator
4	c1qniE_	Alignment		100.0	61	**PDB header:**oxidoreductase **Chain:** E: **PDB Molecule:**nitrous-oxide reductase; **PDBTitle:** crystal structure of nitrous oxide reductase from pseudomonas nautica,2 at 2.4a resolution Run Investigator
5	c1fwxB_	Alignment		100.0	73	**PDB header:**oxidoreductase **Chain:** B: **PDB Molecule:**nitrous oxide reductase; **PDBTitle:** crystal structure of nitrous oxide reductase from p. denitrificans Run Investigator

Figura 26: Resumen de los cinco primeros templates obtenidos para *Pelagimonas varians*

C2iwkB_: Se trata de una forma unida al inhibidor de óxido nitroso reductasa de 2 cicloclastos de Achromobacter con una resolución de 1,7 angstrom.

C5i5iA_: Identificada como una óxido reductasa de *Shewanella denitrificans*.

C3ssbrF_: Se trata de una óxido nitroso reductasa, forma cristalina p1 con sustrato 2 para *Pseudomonas stutzeri*.

C1qniE_: Se trata de una estructura cristalina de óxido nitroso reductasa de *Pseudomonas nautica*, 2 a una resolución de 2.4Å.

C1fwxB_: Identificada como estructura cristalina de óxido nitroso reductasa de *P. desnitrificans*.

En todos los casos la identidad se presentó con una oxidoreductasa con una confianza en los cinco hits de un 100%, indicando una elevada homología entre la secuencia aportada frente a los resultados propuestos.

A nivel de identidad, los valores también fueron elevados con un porcentaje que osciló entre un 71 y un 59%, siendo unos valores considerablemente aceptables.

El modelo obtenido tuvo una confianza del 100% y una cobertura sobre el modelo del 92%, sobre el total de 589 residuos.

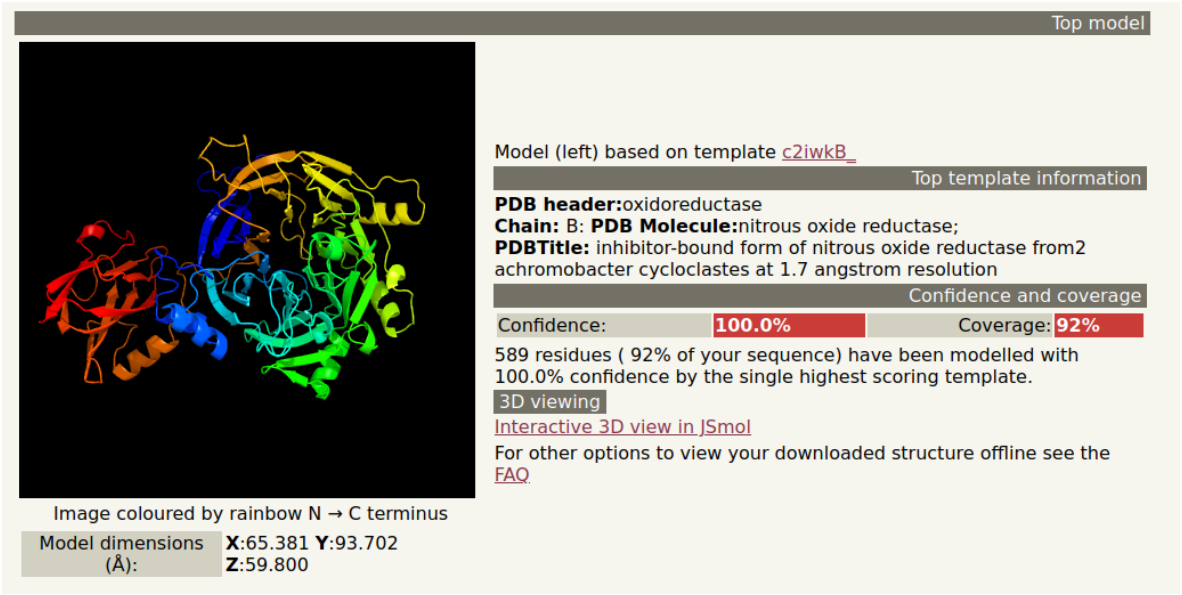


Figura 27: Resumen del mejor modelo para *Pelagimonas varians*

En el caso de *Algoriphagus lacus* y una vez realizado el análisis, el programa devolvió varios templates:

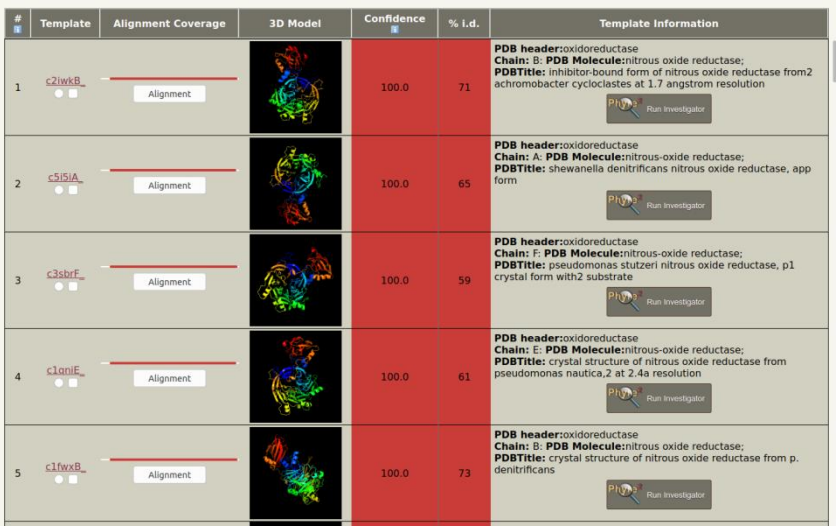


Figura 28: Resumen de los cinco primeros templates obtenidos para *Algoriphagus lacus*

C2iwkB_: Se trata de una forma unida al inhibidor de óxido nitroso reductasa de 2 cicloclastos de Achromobacter con una resolución de 1,7 angstrom.

C5i5iA_: Identificada como una óxido reductasa de *Shewanella denitrificans*.

C3ssbrF_: Se trata de una óxido nitroso reductasa, forma cristalina p1 con sustrato 2 para *Pseudomonas stutzeri*.

C1qniE_: Se trata de una estructura cristalina de óxido nitroso reductasa de *Pseudomonas nautica*, 2 a una resolución de 2.4Å.

C1fwxB_: Identificada como estructura cristalina de óxido nitroso reductasa de *P. desnitrificans*.

En todos los casos la identidad se presentó con una oxidoreductasa con una confianza en los cinco hits de un 100%, indicando una elevada homología entre la secuencia aportada frente a los resultados propuestos.

A nivel de identidad, los valores también fueron elevados con un porcentaje que osciló entre un 73 y un 59%, siendo unos valores considerablemente aceptables.

El modelo obtenido tuvo una confianza del 100% y una cobertura sobre el modelo del 92%, sobre el total de 589 residuos.

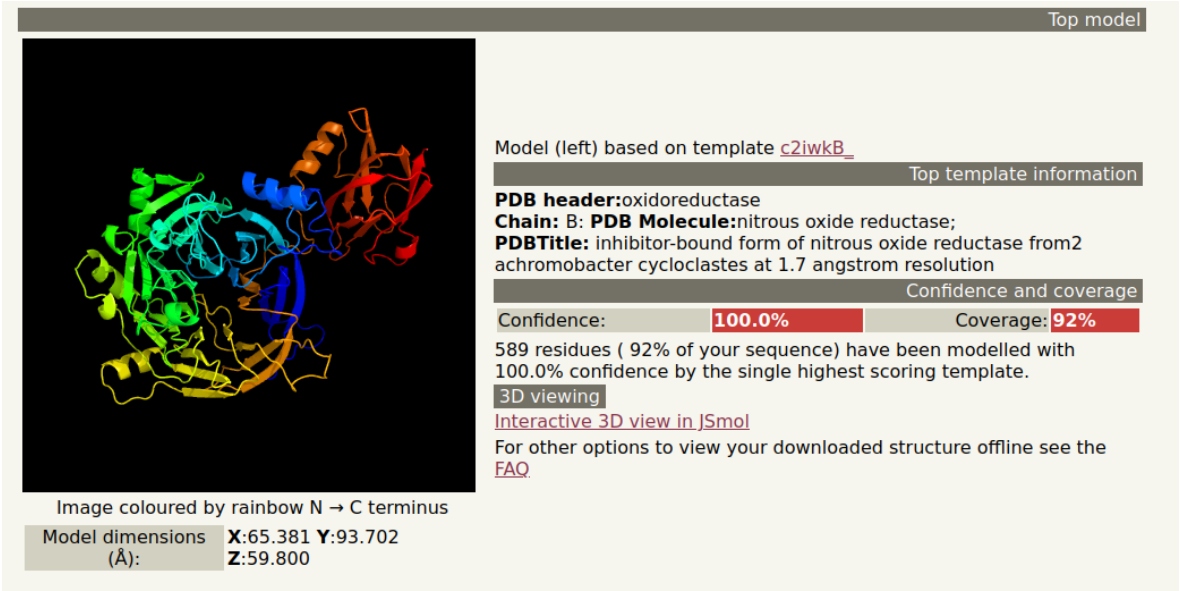
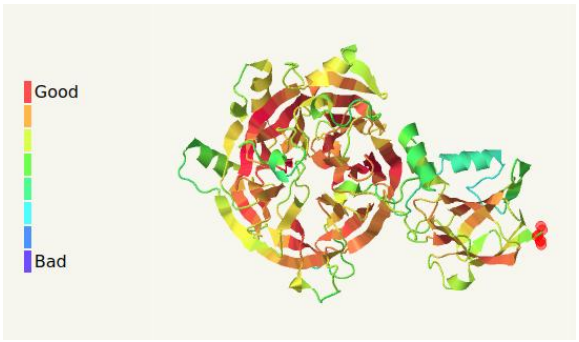


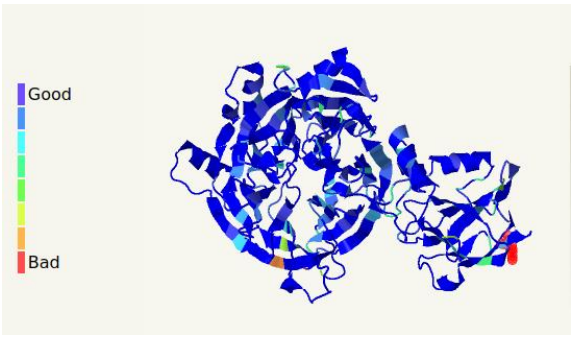
Figura 29: Resumen del mejor modelo para *Algoriphagus lacus*

Se procedió a la revisión mediante la opción Run investigator del primer resultado para ambas especies tanto para los análisis de calidad como de función.

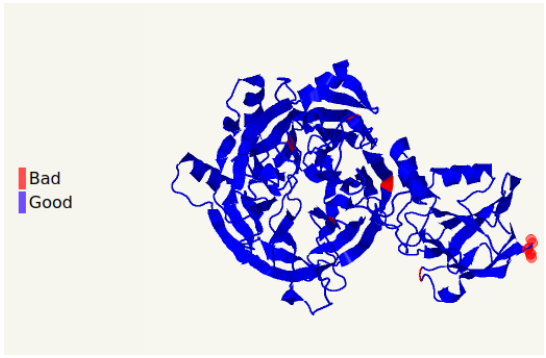
En el caso de *Pelagimona varians* se obtuvieron los siguiente resultados gráficos para los análisis de calidad:



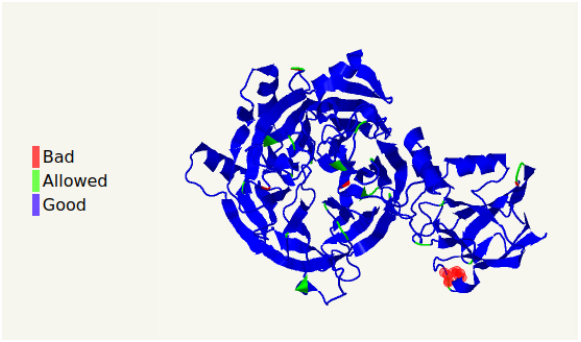
a) ProQ2 quality assesment



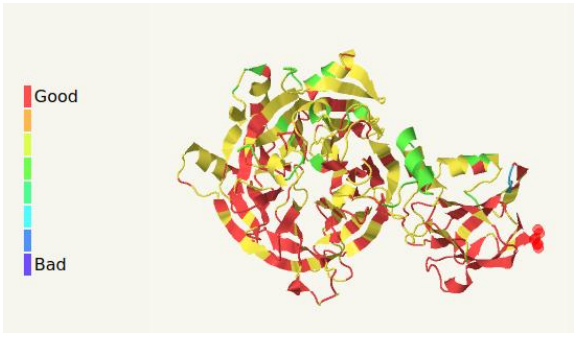
b) Clashes



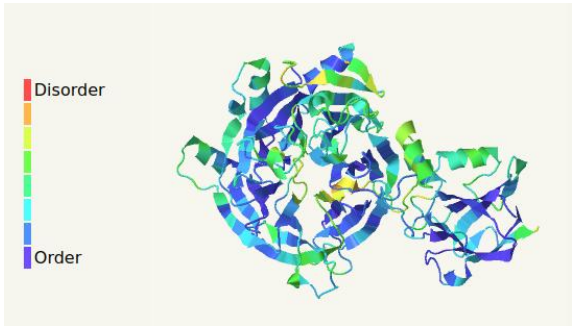
c) Rotamers



d) Ramachandran analysis



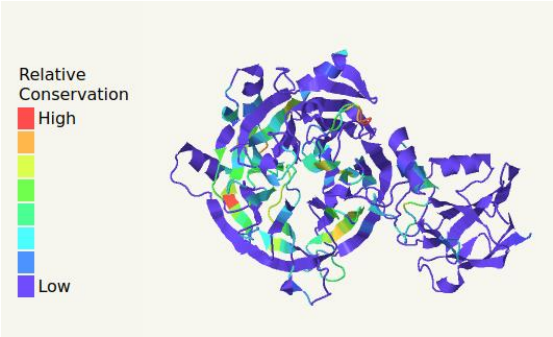
e) Alignment confidence



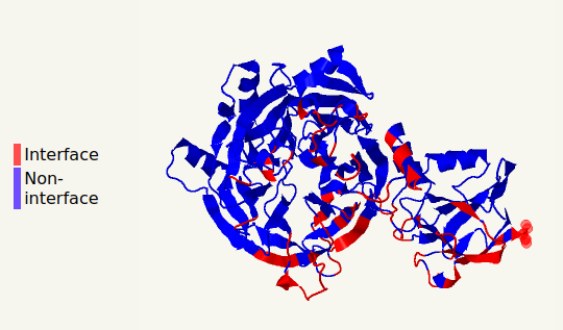
f) Disorder

Figura 30: Resultados análisis de calidad para *Pelagimona varians*

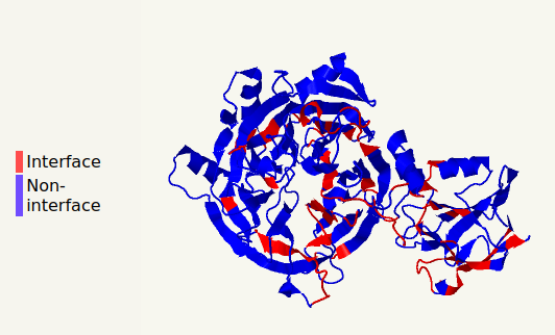
En el caso de los análisis de función se obtuvieron los siguientes resultados gráficos:



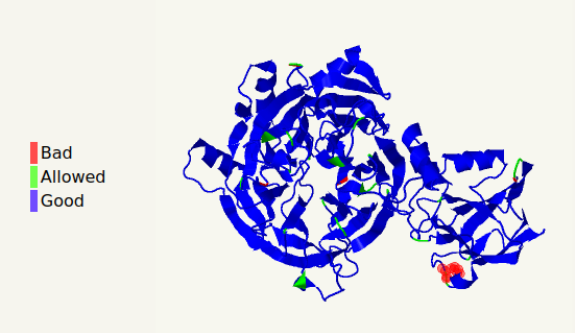
a) Conservation



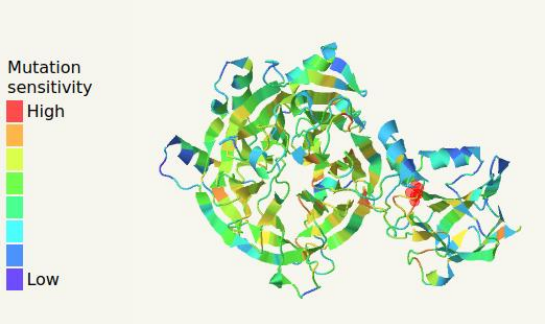
b) ProtinDB Interface



c) PI-Site Interface



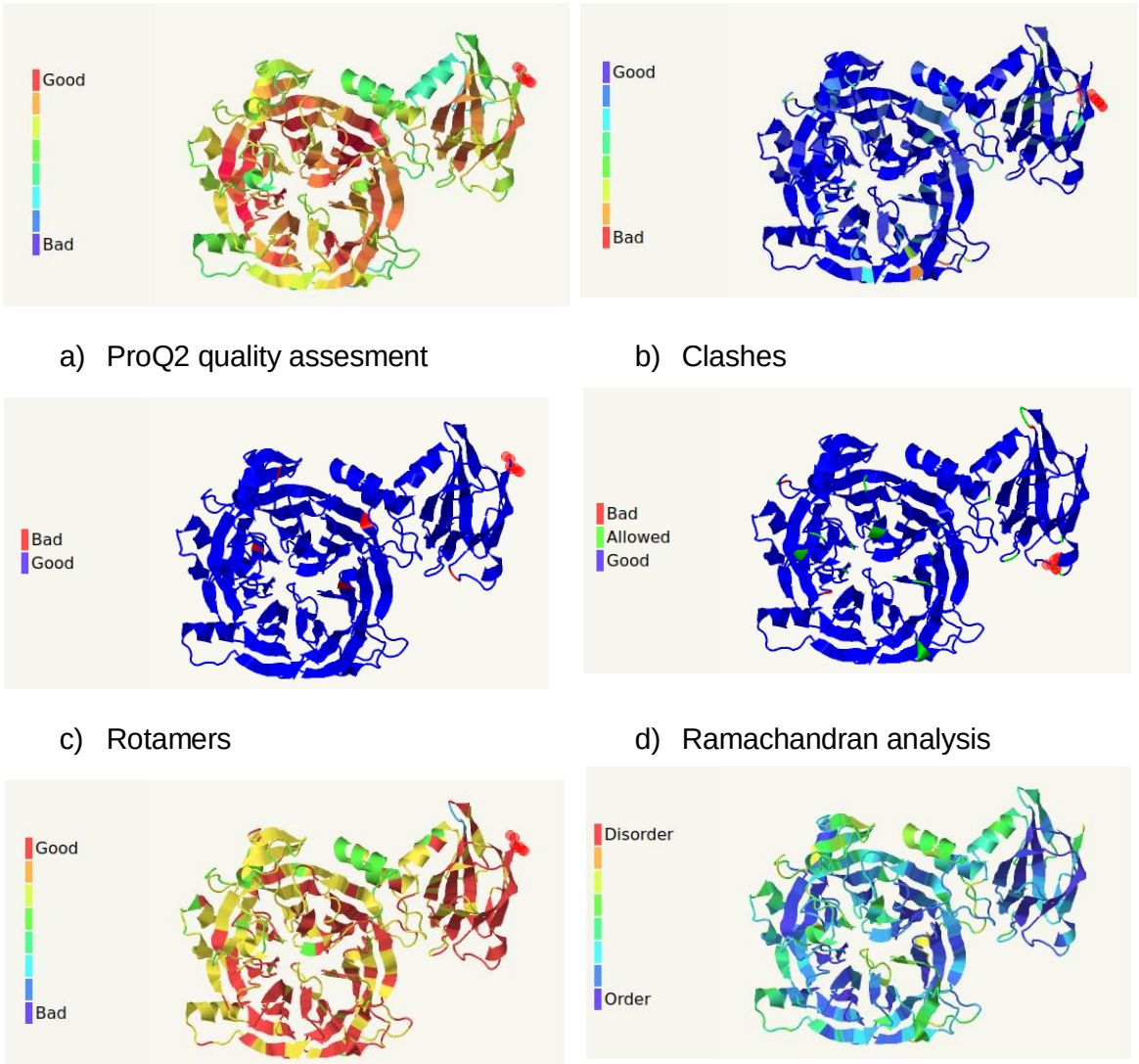
d) Pocket detection



e) Mutational sensitivity

Figura 31: Resultados de análisis de función para *Pelagimona varians*

En el caso de *Algoriphagus lacus* se obtuvieron los siguientes resultados gráficos para los análisis de calidad:



a) ProQ2 quality assesment

b) Clashes

c) Rotamers

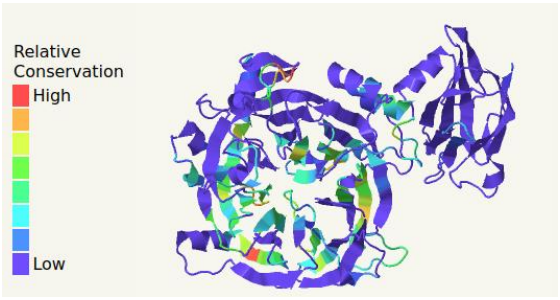
d) Ramachandran analysis

e) Alignment confidence

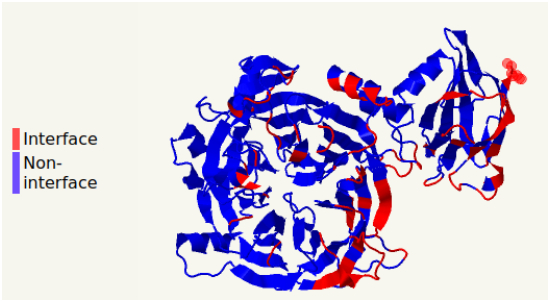
f) Disorder

Figura 32: Resultados de análisis de calidad para *Algoriphagus lacus*

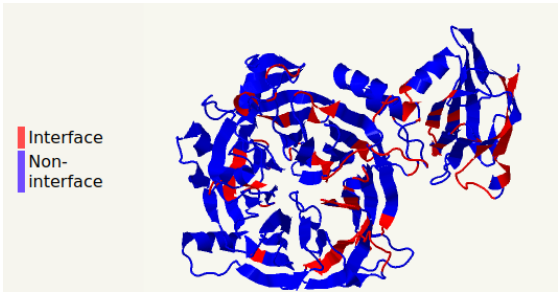
En el caso de los análisis de función se obtuvieron los siguientes resultados gráficos:



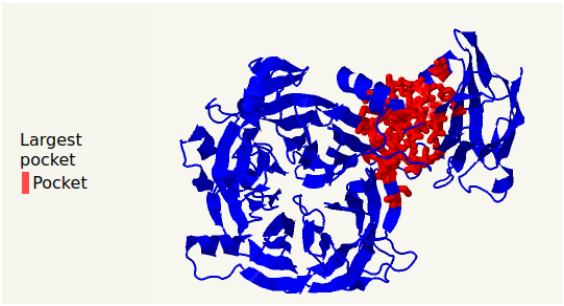
a) Conservation



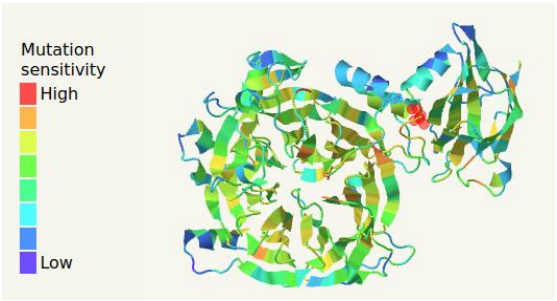
b) ProtinDB Interface



c) PI-Site Interface



d) Pocket detection



e) Mutational sensitivity

Figura 33: Resultados de análisis de función para *Algoriphagus lacus*

En general y en ambos casos observamos que la estructura es de tipo globular. Contiene dos dominios, un dominio de hélice β de siete palas N-terminal. Atendiendo a la literatura la proteína se produce en forma apo y está desprovista de cobre (33).

6. Discusión

Los resultados obtenidos a partir de la interrogación de una base de datos pública, unida al proceso de filtrado y adecuación de la información han permitido realizar una aproximación al tratamiento de un gran volumen de datos. Esto ha permitido al investigador poder hacer interrogaciones a las bases de datos de manera acertada, pudiendo recuperar mayor cantidad de información que mediante la búsqueda combinada de argumentos para el filtrado de especies mediante el uso de navegadores web.

Mediante el uso de las herramientas empleadas en la investigación en comparación a la búsqueda mediante navegadores web también ha permitido una ventaja importante, dado que la información existente en las bases de datos no está suficientemente adecuada, lo que provoca la aparición de numerosa información contaminante que resulta compleja de manipular mediante las opciones de un navegador.

Si bien el proceso de filtrado de la información se considera que tiene puntos de mejora, la obtención mediante un proceso con pocos pasos de identificadores únicos que captan una parte significativa de los datos disponibles permite una menor necesidad de inversión en la revisión de grandes listados con duplicidades así como anotaciones incorporadas que son una fuente de ruido en las interrogaciones a la información contenida en los servidores.

Una limitación detectada ha residido en el acceso a la información incorporada entre nucleótidos y aminoácidos. El intento inicial por realizar el estudio partiendo en la descarga de información en forma de nucleótidos, y la recuperación posterior en aminoácidos no resultó exitosa. Por lo que la alternativa de iniciar la investigación sobre secuencias de aminoácidos y posteriormente la recuperación de las secuencias de nucleótidos asociada, se considera que ha sido una ventaja en el desarrollo del procedimiento en el tratamiento de los datos.

La posibilidad de controlar el flujo de información y repetir los procesos intermedios pudiendo comparar listas manejables se considera como un punto fuerte del proceso, el cual permite valorar la información generada en cada paso y la valoración en el cambio de la parametrización.

La topología obtenida a partir de secuencias de aminoácidos ha sido semejante a la de (29), con la diferencia del nivel de agrupación presentado. El presente estudio buscó la originalidad de recuperar el máximo número de especies existentes en una base de datos pública frente a (29), quienes utilizaron los datos de Sanford et al. (34).

Como contrapartida, la topología obtenida a nivel de secuencias de nucleótidos se encontró con un resultado, pese a tener menor número de secuencias, la existencia de un fenómeno de atracción de ramas largas. Este fenómeno requeriría evaluar diferentes paradigmas analíticos que escapan al alcance del presente trabajo, entre ellos la repetición de los análisis con un conocimiento más profundo de los taxa, de

manera que se pudiesen seleccionar aquellos que no tuviesen una controversia en su composición filogenética.

Las predicciones en el perfil de la proteína empleando un representante del clado I y del clado II como han sido Pelagimona varians y Algoriphagus lacus, en general han mostrado una aproximación aceptable a nivel de identificación general de la estructura, con resultados que se ajustaron a los modelos existentes en Phyre2.

Teniendo en cuenta que el esfuerzo en establecer estas bases de datos tan ajustadas, limitan la cantidad de perfiles incluidos en la misma, se considera que pese a no aportar resultados significativamente discrepantes entre cada uno de los clados, sí que se acepta que el resultado es coherente en ambos casos.

Como consecuencia del hecho de disponer de una base de datos más reducida, los valores del análisis de función mostraron que los perfiles posiblemente requiriesen de aproximaciones a su estructura mediante técnicas analíticas y diferente aproximación bioinformática, como la obtención de la estructura cristalina de la proteína en las especies no conocidas y su análisis mediante resonancia magnético nuclear y el análisis de los resultados mediante algoritmos de predicción específicos.

El futuro de los perfiles de proteínas requiere un esfuerzo intenso, dada la diferencia en órdenes de magnitud entre estructuras de proteínas resueltas frente al número de proteínas existentes en las bases de datos. Este hecho aporta la necesidad de continuar realizando a futuro un mayor esfuerzo y dedicación en un trabajo de investigación básica para incrementar el nivel de conocimiento necesario como punto de partida para su empleo en investigación tanto básica como aplicada.

7. Conclusiones

Los resultados de este proyecto son un refuerzo de estudios previos que hacen un análisis sobre la distribución de este gen a nivel filogenético, con la aportación de disponer de una información actualizada y reproducible mediante el uso de herramientas en terminal. Hay que tener en cuenta que estos resultados no son definitivos y se considera que tiene margen de mejora, dado que el nivel de información existente y que debiera ser revisado es elevado. No obstante y con todas las limitaciones, las topologías obtenidas son suficientemente consistentes.

La práctica totalidad de los objetivos planteados en la planificación del proyecto se han alcanzado, salvo el desarrollo de un script con mayor alcance y que pudiese aglutinar parte de las tareas realizadas en bash y mediante el uso de scripts propios y generados en lenguajes Perl y Python.

Se puede recomendar el uso de la información generada y del código, para integrar en un pipeline más completo que pueda materializar de manera más automatizada los procesos ejecutados secuencialmente por separado. Ya que si bien se ha conseguido un distanciamiento bastante efectivo frente al uso de herramientas en web browser, un conocimiento sobre el gen de estudio y sobre aspectos identificables por la experiencia de los investigadores puede suponer una diferencia importante en la supervisión de los procesos. Todo esto podría reducir la tarea del investigador frente a una interpretación mejorada del lenguaje utilizado en las anotaciones existentes en las bases de datos públicas.

8. Glosario

- **Alineamiento**: representación y comparación de dos o más cadenas de ADN, ARN, estructuras primarias de proteínas donde se evidencian zonas de similitud en un formato de matriz.
- **Bayesian inference**: método de inferencia estadística basada en el teorema de Bayes para actualizar de manera dinámica la probabilidad de una hipótesis en un proceso iterativo.
- **Clado**: conjunto de ramificaciones obtenidas a partir de un corte en un árbol filogenético.
- **Filogenia**: Relación de parentesco entre especies o taxones a partir de su ADN, ARN u otros datos biológicos de interés.
- **Gap**: espacio o intervalo que rompe con la continuidad.
- **GUI**: acrónimo del inglés para definir la interfaz gráfica del usuario (Graphical User Interface) y separarlo de la interacción mediante líneas de comandos.
- **HPC**: acrónimo de High Performing Computing. Se refiere a la práctica de agregar potencia de computación de manera que permita ofrecer un rendimiento más alto para resolver problemas en análisis de datos en diferentes ámbitos.
- **Maximum likelyhood (ML)**: método de estimación de parámetros de una distribución de probabilidad que busca maximizar la verosimilitud asumiendo que el modelo de los datos observados es el más probable. En este caso pretende encontrar la topología del árbol que confiera la mayor probabilidad a las características de las especies.
- **Markov chain Monte Carlo**: método estadístico que comprenden una serie de algoritmos para el muestreo de una distribución de probabilidad.
- **Modelo evolutivo**: algoritmo computacional que permite reconstruir un árbol filogenético que represente una hipótesis evolutiva a partir de la información aportada.
- **Run**: Cada uno de los árboles filogenéticos independientes de partida, que se ejecutarán independientemente en mrBayes, y sobre los que se evaluarán sus resultados para identificar la convergencia en sus parámetros en la topología final.
- **Script**: lista de comandos que ejecuta un determinado programa o motor de secuencias de comandos.
- **Topología**: Diagraman que representa las relaciones evolutivas entre organismos obtenidas en los análisis filogenéticos. Se trata de hipótesis que intentan reflejar la evolución de especies a partir de ancestros comunes.
- **Señal filogenética**: evaluación de la potencial existencia de dirección y cambios evolutivos en un conjunto de secuencias alineadas.

9. Bibliografía

1. J, Kans. Entrez direct: E-utilities on the UNIX command line. En: Entrez Programming Utilities. [Online].; 2020.

2. U C. UniProt: a worldwide hub of protein knowledge. Nucleic acids research. 2019; 47(D1): D506-15.

3. Mcwilliam H, Valentin F, Goujon M, Li W, Narayanasamy M, Martin J, et al. Web services at the european bioinformatics institute-2009. Nucleic acids research. 2009; 37(suppl_2): W6-W10.

4. Thompson J, Higgins G, Gibson T. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids research. 1994; 22(22): 4673-80.

5. K-B L. ClustalW-MPI: ClustalW analysis using distributed and parallel computing. Bioinformatics. 2003; 19(12): 1585-6.

6. RC E. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research. 2004; 32(5): 1792-7.

7. Strimmer K VHA. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. Molecular biology and evolution. 1996; 13(7): 964-9.

8. Strimmer K VHA. Likelihood-mapping: a simple method to visualize phylogenetic content of a sequence alignment. In Proceedings of the National Academy of Sciences; 1997. p. 6815-9.

9. Posada D, Crandall KA. Modeltest: testing the model of DNA substitution. Bioinformatics. 1998; 14(9): 817-8.

10. D., Posada. jModelTest: phylogenetic model averaging. Molecular biology and evolution. 2008; 25(7): 1253-6.

11. Darriba D TGDRPD. jModelTest 2: more models, new heuristics and parallel computing. Nature methods. 2012; 9(8): 772-2.

12. Abascal F ZRPD.. ProtTest: selection of best-fit models of protein evolution. Bioinformatics. 2005; 21(9): 2104-5.

13. Darriba D TGDRPD. ProtTest 3: fast selection of best-fit models of protein evolution. Bioinformatics. 2011; 27(8): 1164-5.

14. Darriba D, Posada D, Kozlov AM, Stamatakis A, Morel B, Flouri T. ModelTest-NG:

a new and scalable tool for the selection of DNA and protein evolutionary models. Molecular Biology and Evolution. 2020; 37(1): 291-4.

15. J, Castresana. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Molecular biology and evolution. 2000; 17(4): 540-52.

16. Talavera G, Castresana J. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. Systematic biology. 2007; 56(4): 564-77.

17. A, Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics. 2006; 22(21): 2688-90.

18. Ronquist F, Teslenko M, Van Der Mark P, Ayres DL, Darling A, Höhna S, et al. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. Systematic biology. 2012; 61(3): 539-42.

19. Huelsenbeck JP RF. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics. 2001; 17(8): 754-5.

20. Cardona G, Rosselló F, Valiente G. Extended Newick: it is time for a standard representation of phylogenetic networks. BMC bioinformatics. 2008; 9(1): 1-8.

21. A, Rambaut. FigTree. Tree figure drawing tool. [Online].; 2009. Available from: http://tree.bio.ed.ac.uk/software/figtree/.

22. Kelley LA MSYCWMSM. The Phyre2 web portal for protein modeling, prediction and analysis. Nature protocols. 015; 10(6): 845-58.

23. Hahn J CP. The role of fixed nitrogen in atmospheric photochemistry. Philosophical Transactions of the Royal Society of London B, Biological Sciences. 1982;: 521-41.

24. Wang W YYLAMTHJ. Greenhouse effects due to man-made perturbations. Science. 1976; 194(4266): 685-90.

25. Scala DJ, Kerkhof LJ. Nitrous oxide reductase (nosZ) gene-specific PCR primers for detection. FEMS Microbiology Letters. 1998; 162(1): 61-8.

26. Scala DJ KL. Diversity of nitrous oxide reductase (nosZ) genes in continental shelf. Applied and environmental microbiology. 1999; 65(4): 1681-7.

27. Teraguchi S HT. Purification and some characteristics of a cytochrome c-containing nitrous oxide reductase from *Wolinella succinogenes*. Journal of Biological Chemistry. 1989; 264(4): 1972-9.

28. Zhang L WAPBMCEO. Functional assembly of nitrous oxide reductase provides insights into copper site maturation. In Proceedings of the National Academy of

Sciences; 2019. p. 12822-7.

29. Hallin S PLLFSRJC. Genomics and ecology of novel N2O-reducing microorganisms. Trends in Microbiology. 2018; 26(1): 43-55.

30. Whelan S, Goldman N. A General Empirical Model of Protein Evolution Derived from MultipleProtein Families Using a Maximum-Likelihood Approach. Molecular Biology and Evolution. 2001; 18(5): 691-99.

31. Tavaré S, Miura R. Some probabilistic and statistical problems in the analysis of DNA sequences. American Mathematical Society. 1986; 17: 57-86.

32. Huson, Daniel H; Richter, Daniel C; Rausch, Christian; Dezulian, Tobias; Franz, Markus; Rupp, Regula. Dendroscope: An interactive viewer for large phylogenetic trees. BMC Bioinformatics. 2007; 8(460).

33. Dassa, Elie. CHAPTER 1 - PHYLOGENETIC AND FUNCTIONAL CLASSIFICATION OF ABC (ATP-BINDING CASSETTE) SYSTEMS**ABSCISSE. In Higgins I, Holland B, P.C. Cole S, Kuchler K, F. Higgins C, editors. ABC Proteins. London: Academic Press; 2003. p. 3-35.

34. Sanford, Robert A.; Wagner, Darlene D; Wu, Qinzhong; Chee-Sanford, Joanne C.; Thomas, Sara H.; Cruz-García, Claribel; Rodríguez, Gina; Massol-Deyá, Arturo; Krishnani, Kishore K.; Ritalahti, Kristi M.; Nissen, Silke; Konstantinidis, Konsstantios T.; Löffler, Frank E.. Unexpected nondenitrifier nitrous oxide reductase gene diversity and abundance in soils. PNAS. 2012; 109(48): 19709-19714.

10. Anexos

Anexo 1.

Documento en formato hoja de cálculo con las tablas generadas con los identificadores de las especies utilizadas a nivel de aminoácidos, nucleótidos y la taxonomía asociada a las especies utilizadas.

Anexo 2.

Código de scritps utilizados en la realización de la primera parte del TFM

nt_download.py

```
import glob
import os
import shutil
import pathlib
import wget
from contextlib import redirect_stdout
from alive_progress import alive_bar


# Abrir archivo con listado bueno de aquello que quiero descarga
a_file = open("secuencias_nt_a_descargar.txt ",'r')
lines = a_file.readlines()
for line in lines:
        l_split = line.split(" ") # cambiar por tabulador según generes tabla
        # Separar secuencias por strands
        if '+' in l_split:
                print(os.system("wget \'https://www.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide&id="+l_split[0]+"&seq_start="+l_split[1]+"&seq_stop="+l_split[2]+"&rettype=fasta&retmode=text\'  -O /dev/stdout | tee -a nucleotidos_forward.fasta"))
        else:
```

```
                print(os.system"wget
\'https://www.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide&id="+l_split[0]+"
&seq_start="+l_split[1]+"&seq_stop="+l_split[2]+"&rettype=fasta&retmode=text\'       -O
/dev/stdout | tee -a nucleotidos_reverse.fasta"))

a_file.close()
```

reverse_complement.py

```
import sys


inFile = open(sys.argv[1],'r')


nuc = {'A':'T','T':'A','G':'C','C':'G','K':'M','M':'K','R':'Y','Y':'R','S':'W','W':'W','B':'V','V':'B','H':'G','D':'C','X':'N','N':'N'}


def revComp(seq):
    rev = ''
    for i in range(len(seq) - 1,-1,-1):
        rev += nuc[seq[i]]

    return rev

header = ''
seq = ''
for line in inFile:
    if line[0] == ">":
        if header != '':
            print (header)
            print (revComp(seq.upper()))

        header = line.strip()
        seq = ''
    else:
        seq += line.strip()
```

```
print (header)
print (revComp(seq.upper()))
infile.close()
```

merge2files.py

```
#!/usr/bin/env python
import shutil
with open('secuencias_nt.fasta','wb') as wfd:
    for f in ['forward.fasta','forward2.fasta','manual.fasta']:
        with open(f,'rb') as fd:
            shutil.copyfileobj(fd, wfd)
```

aligner.py

```
#!/usr/bin/env python
#Run clustalw and parse the output.
# Call standard libraries
import sys
import subprocess

# Call Biopython libraries
from Bio.Align.Applications import ClustalwCommandline
from Bio import AlignIO
from Bio.Align import AlignInfo

f_to_parse= input("Let's start! \nPlease type the file to parse: ")
f_output = "aligned_"+ f_to_parse

# Create the command line to run clustalw
```

```
cline = ClustalwCommandline(infile=f_to_parse, outfile=f_output)


# Perform the alignment

return_code = subprocess.call(str(cline), shell=(sys.platform != "ubuntu"))

assert return_code == 0, "Calling ClustalW failed"


# Parse the output

alignment = AlignIO.read(f_output, "clustal")

print(alignment)
```

fasta2phylip.py

```
#!/usr/bin/env python

import os

import argparse

from Bio import AlignIO

import sys


name = os.path.basename(sys.argv[0]) #get scriptname from actual script filename that was called

parser=argparse.ArgumentParser(description="Converts alignments in FastaFormat to (strict & interleaved; relaxed & interleaved if '-r' is set) phylip format. Will raise error if alignments contain dots (\".\"), so replace those with dashes (\"-\") beforehand (e.g. using sed)")

parser.add_argument('-i','--input', action = "store", dest = "input", required = True, help = "(aligned) input fasta")

parser.add_argument('-o','--output', action = "store", dest = "output", help = "Output filename (default = <Input-file>.phylip")

parser.add_argument('-r','--relaxed', action = "store_true", dest = "relaxed", default = False, help = "output in \"relaxed\" phylip format. (default = False --> Output is strict phylip")

args=parser.parse_args()
```

```
if not args.output:
        args.output = args.input + ".phylip"
def main():
        infile = open(args.input, "r")
        outfile = open(args.output, "w")
        alignments = AlignIO.parse(infile, "fasta")
        if args.relaxed:
                AlignIO.write(alignments, outfile, "phylip-relaxed")
        else:
                AlignIO.write(alignments, outfile, "phylip")
        infile.close()
        outfile.close()
        sys.stderr.write("\nfinished\n")

main()
```

taxonomy.py

```
import cs
from ete3 import NCBITaxa
# si no lo tienes instalado pip install ete3

ncbi = NCBITaxa()

def get_desired_ranks(taxid, desired_ranks):
    lineage = ncbi.get_lineage(taxid)
    names = ncbi.get_taxid_translator(lineage)
    lineage2ranks = ncbi.get_rank(names)
    ranks2lineage = dict((rank,taxid) for (taxid, rank) in
lineage2ranks.items())
    return{'{}_id'.format(rank): ranks2lineage.get(rank,
'<not present>') for rank in desired_ranks}

def main(taxids, desired_ranks, path):
    with open(path, 'w') as csvfile:
```

```
        fieldnames = ['{}_id'.format(rank) for rank in
desired_ranks]
    writer = csv.DictWriter(csvfile, delimiter='\t',
fieldnames=fieldnames)
    writer.writeheader()
    for taxid in taxids:
        writer.writerow(get_desired_ranks(taxid,
desired_ranks))

if __name__ == '__main__':
    taxids = [<listado de número de especie>]
    desired_ranks = ['kingdom', 'phylum', 'class', 'order',
'family', 'genus', 'species']
    results = list()
    for taxid in taxids:
        results.append(list())
        results[-1].append(str(taxid))
        ranks = get_desired_ranks(taxid, desired_ranks)
        for key, rank in ranks.items():
            if rank != '<not present>':
                results[-
1].append(list(ncbi.get_taxid_translator([rank]).values())[0]
)
            else:
                results[-1].append(rank)

    #generate the header
    header = ['Original_query_taxid']
    header.extend(desired_ranks)
    print('\t'.join(header))

    #print the results
    for result in results:
        print('\t'.join(result))
```