Universidad Europea

Máster en Bioinformática

Estudio a nivel masivo del proteoma compartido por las especies del género *Bifidobacterium*

Autor: Aida Vaquero Rey

Tutor: Alfonso Benítez Páez

Curso 2022-2023

AGRADECIMIENTOS

En primer lugar, quiero agradecer a Alfonso Benítez Páez por haberme dado la oportunidad de realizar un trabajo tan apasionante en su grupo de investigación, y haberme guiado en todo el proceso.

También quiero hacer una mención especial a Rocío González Soltero, ya que ha estado siempre dispuesta a ayudarme en todo, respondiéndome correos incluso en festivos. Gracias por tu implicación y dedicación Rocío.

Desde una parte más personal, quiero agradecer a Jordi, que me ha aguantado a lo largo de todo el proceso y ha sido el mejor compañero y amigo del mundo. A mi madre y a mi hermana por apoyarme, entenderme y ser unos pilares fundamentales para mí.

Pero sobre todo estos agradecimientos van pensados y dedicados a dos personas, sin las cuales este Trabajo de Fin de Máster no hubiese sido posible. Gracias a mi padre por estar ahí, día tras día, por ayudarme a enfrentar cada problema que se puso en el camino, y remover cielo, mar y tierra por mí. Y gracias a Sergio, mi principal apoyo emocional, mi motivación de sacar este trabajo adelante. Gracias por creer en mí cuando ni yo era capaz de hacerlo, por ser increíblemente comprensivo, por aguantarme, por tus palabras de ánimo, y por supuesto, gracias por haberme ayudado en absolutamente todo. Todo lo que pueda decir se queda corto.

Por último, quiero agradecerme a mí misma, porque nunca me había enfrentado a algo tan desconocido, lo que ha implicado un trabajo muy duro. Me siento orgullosa por no haber tirado la toalla, cuando parecía que nada estaba a mi favor. Por haber podido sacar este trabajo adelante, pese al haber sido un año en mi vida lleno de cambios, y pese las limitaciones de tiempo.

ÍNDICE

RESUMEN

El género *Bifidobacterium* es uno de los más predominantes en el tracto gastrointestinal humano (TGI), relacionado con el metabolismo de los carbohidratos derivados del hospedador y de la dieta. Se sabe que las especies de *Bifidobacterium* más predominantes en TGI son las más adaptadas a este nicho, sin embargo, se desconocen los mecanismos moleculares.

Objetivos: El objetivo general del presente trabajo fue realizar un análisis masivo de genomas completos de *Bifidobacterium* con hospedador humano, identificando los genes codificantes para estudiar las funciones proteicas compartidas en el total de genomas.

Material y métodos: Se partió de un total de 822 genomas completos recogidos de la base de datos BV-BRC del género *Bifidobacterium.* Se llevaron a cabo distintos filtros para obtener únicamente genomas no contaminados y pertenecientes al género *Bifidobacterium* con hospedador humano. Se llevaron a cabo predicciones de genes codificantes, agrupamientos múltiples y alineamientos múltiples de secuencias con el fin de realizar una anotación funcional a través de agrupamientos de genes ortólogos (COG). Para todo ello, se utilizó el software R y distintas herramientas instaladas en una máquina virtual con sistema operativo Linux, distribución Ubuntu.

Resultados: *B.longum* fue la especie más representada en el conjunto de estudio. Por otra parte, el análisis de la calidad de los genomas reveló 15 genomas contaminados, que se eliminaron para los posteriores análisis. De los agrupamientos múltiples, el realizado con un 85% de identidad de secuencia fue el que mejor representó el proteoma central. Con este umbral, la categoría J (traducción, estructura y biogénesis ribosómica) fue la más representada. Por último, se identificaron 7 familias distintas relacionadas con el metabolismo de los carbohidratos (CAZy) utilizando un umbral de agrupamiento del 50% de identidad, representando el proteoma extendido.

Conclusiones: Los genomas analizados presentaron una alta calidad que sostiene los resultados obtenidos. Por otro lado, el establecimiento del proteoma central y la detección de las familias CAZy permitirán futuros estudios de divergencia funcional relacionada con el metabolismo de carbohidratos.

Palabras clave: *Bifidobacterium*, proteoma central, proteoma extendido, familias CAZy

ABSTRACT

The genus *Bifidobacterium* is one of the most predominant in the human gastrointestinal tract (GIT), related to the metabolism of carbohydrates derived from the host and diet. It is known that the most predominant *Bifidobacterium* species in the GIT are the most adapted to this niche; however, the molecular mechanisms are unknown.

Objectives: The overall objective of the present work was to perform a massive analysis of complete genomes of *Bifidobacterium* with human host, identifying the coding genes to study the shared protein functions in the total genomes.

Material and methods: A total of 822 complete genomes collected from the BV-BRC database of the genus *Bifidobacterium* were used. Different filters were performed to obtain only uncontaminated genomes belonging to the genus Bifidobacterium with human host. Coding gene predictions, multiple clustering and multiple sequence alignments were carried out in order to perform functional annotation through clustering of orthologous genes (COG). For all this, R software and different tools installed in a virtual machine with Linux operating system, Ubuntu distribution, were used.

Results: *B.longum* was the most represented species in the study set. On the other hand, genome quality analysis revealed 15 contaminated genomes, which were eliminated for further analysis. Of the multiple groupings, the one performed with 85% sequence identity best represented the core proteome. At this threshold, category J (translation, structure and ribosomal biogenesis) was the most represented. Finally, 7 distinct families related to carbohydrate metabolism (CAZy) were identified using a clustering threshold of 50% identity, representing the expanded proteome.

Conclusions: The genomes analyzed presented a high quality that supports the results obtained. On the other hand, the establishment of the core proteome and the detection of CAZy families will allow future studies of functional divergence related to carbohydrate metabolism.

Key words: *Bifidobacterium*, proteome core, extended proteome, CAZy families

1. INTRODUCCIÓN

1.1. Características del género *Bifidobacterium*

Las especies del género *Bifidobacterium* representan uno de los grupos microbianos dominantes en el intestino de diversos animales, siendo especialmente prevalentes durante el periodo de lactancia de los seres humanos y otros mamíferos. Se trata de bacterias Gram-positivas, anaeróbicas obligatorias, no productoras de gas, pertenecientes al filo Actinomyceota y la familia Bifidobacteriaceae (Sgorbati et al., 1995). Dentro del género *Bifidobacterium* las especies de *B. breve* y *B. longum* destacan por ser las más prevalentes en la microbiota intestinal en niños y adultos (Lee & O'Sullivan, 2010). Algunos autores describen que *B. bifidum* y *B.breve* son especies características de los niños, mientras que *B.adolescentis* y *B.catenulatum* de los adultos (Satti et al., 2018).

Estas bacterias, transferidas verticalmente a recién nacidos a través de la leche materna y por contacto directo con el canal vaginal, se han asociado con el bienestar humano. Están involucradas en la modulación de la respuesta inmune, habiéndose asociado con su fortalecimiento (Yoshioka et al., 1983), la atenuación de reacciones inflamatorias y alérgicas (Sheil et al., 2006; Satti et al., 2018), el alivio de la dermatitis atópica (Satti et al., 2018), e incluso, su ausencia en la microbiota intestinal infantil se ha relacionado con enfermedades autoinmunes y asma (Cukrowska et al., 2020). Por otra parte, también juegan un papel relevante en la inhibición del crecimiento de microorganismos perjudiciales (Devika & Raman, 2019; Benítez-Páez et al., 2020), permiten la síntesis de nutrientes (Scholz-Ahrens et al., 2001) y el acortamiento de la diarrea (Satti et al., 2018). Recientemente, también se ha descubierto la implicación de las bifidobacterias en el establecimiento y mantenimiento de las conexiones intestino-cerebro (Luck et al., 2020).

Como es sabido, la variabilidad del genoma de los organismos procariotas del mismo grupo taxonómico se obtiene de pérdidas y ganancias de genes (Kashtan et al., 2014; Kettler et al., 2007). El estudio de la evolución y la dinámica del genoma de los microorganismos con hospedador humano se puede realizar a través de la

metagenómica mediante el uso de herramientas bioinformáticas especializadas. Los análisis genómicos comparados y funcionales de las especies del género *Bifidobacterium* permiten extraer conclusiones acerca de las divergencias genéticas y adaptaciones que les permiten sobrevivir y ser competitivas en diversos nichos como la mucosa oral o la mucosa intestinal, además de explicar las diferencias intra-especie de las capacidades metabólicas, evasión del sistema inmunitario adaptativo del hospedador y adhesión al intestino del hospedador (O'Callaghan & van Sinderen, 2016).

El estudio de la filogenia basada en el genoma codificante central, es decir, el proteoma compartido por las todas las especies del género *Bifidobacterium*, permite encontrar huellas genómicas derivadas de presión de selección específica del hospedador y, por tanto, explicar las adaptaciones que les confieren capacidad competitiva con otras bacterias intestinales. Del género *Bifidobacterium* se puede destacar la capacidad de utilizar del hospedador humano los oligosacáridos que se encuentran en la leche materna (Satti et al., 2018), lo que le confiere una ventaja en dicho nicho, al poseer características sacarolíticas. Esto también confiere accesibilidad metabólica a los carbohidratos complejos de la dieta y/o derivados del hospedador, lo que explicaría que fuese el clado dominante en la microbiota intestinal de bebés (Lugli et al., 2018; Satti et al., 2018). Por otra parte, los apéndices proteicos de las bifidobacterias o pili bifidobacterianos presentan especial relevancia en la adhesión a carbohidratos y proteínas de la matriz extracelular, lo que conduce a un éxito ecológico de las bifidobacterias en el intestino de los mamíferos (Milani et al., 2017). Todo ello supone una potente fuerza evolutiva que ha dado forma al genoma bifidobacteriano, conteniendo más de un 8% del total de sus genes involucrados en el metabolismo de los carbohidratos (Lugli et al., 2018; Satti et al., 2018).

Las bifidobacterias se han utilizado ampliamente como probióticos comerciales, pero los mecanismos moleculares responsables de su simbiosis no se conocen con exactitud. Se han llevado a cabo estudios basados en la búsqueda a nivel de genoma de rasgos probióticos entre todas las cepas de bifidobacterias, determinando qué especies podrían conferir beneficios para la salud a individuos con carencia de nutrientes. El

conocimiento de las firmas metabólicas específicas también se podría aplicar a diseño de intervenciones dietéticas personalizadas (Fushinobu et al., 2021; Deb, 2022).

También se han realizado estudios que han demostrado la transferencia horizontal de genes y eventos de ganancia-pérdida de genes a lo largo de la evolución de las especies del género *Bifidobacterium*, dando lugar a la divergencia de las funciones metabólicas (Deb, 2022). No obstante, aún es necesario profundizar en el conocimiento acerca de la divergencia funcional y el potencial probiótico de todas las bifidobacterias, así como estudiar la dinámica del contenido genómico y las funciones metabólicas asociadas a este género.

Trabajos previos del laboratorio que alberga el presente trabajo de investigación han realizado un análisis comparativo de 822 genomas completos de bacterias pertenecientes al género *Bifidobacterium*, detectando inconsistencias en las bases de datos en cuanto a la anotación de las especies y estableciendo relaciones taxonómicas estables dentro del género, atendiendo a los valores obtenidos mediante la herramienta FastANI (Bahilo, 2022). El presente Trabajo de Fin de Máster (TFM) partirá de este conjunto de genomas para llevar a cabo la identificación del proteoma central del género *Bifidobacterium* y su anotación funcional a través de agrupamientos de grupos ortólogos (COG). Este trabajo sentará las bases para futuros estudios de divergencia funcional y evolución molecular de genes relacionados con el metabolismo de los carbohidratos de las especies más predominantes en el TGI del género *Bifidobacterium*.

Pese a que se han llevado a cabo estudios de las estructuras tridimensionales de las proteínas bifidobacterianas implicadas en la absorción, degradación y metabolismo de los hidratos de carbono, se requiere una mayor investigación a nivel estructural y bioquímico de las vías catalíticas, así como de las enzimas y proteínas transportadoras implicadas. A su vez, descifrar los mecanismos estructurales de los componentes proteínicos es fundamental para revelar la coevolución molecular entre las bifidobacterias y los seres humanos, debido al éxito en las relaciones simbióticas con distintos hospedadores, entre ellos, los seres humanos (Fushinobu & Abou Hachem, 2021).

1.2. Herramientas de análisis

Para poder abordar los estudios que ayuden a la comprensión de la evolución molecular, dinámica génica y divergencia funcional, es necesario el uso de herramientas computacionales que permitan extraer dicha información de forma precisa y eficiente en tiempo.

En primer lugar, es necesario partir de genomas de alta calidad, lo que derivará en resultados robustos, evitando interpretaciones biológicas erróneas, como por ejemplo las debidas a inferencias falsas sobre la ubicación filogenética. La calidad del genoma se evalúa atendiendo a la fragmentación y distribución de los contigs ensamblados, a la fracción del genoma de origen capturado y a la contaminación producida de fragmentos genómicos procedentes de otras fuentes (Orakov et al., 2021; Deb, 2022).

Esto es de gran importancia ya que los genomas depositados en las bases de datos pueden contener contaminación producida tanto *in vitro* como *in silico* (Orakov et al., 2021; Deb, 2022). La contaminación originada *in vitro* se refiere a la derivada del procesamiento de las muestras en el laboratorio, como la contaminación de los medios de cultivo. Por otra parte, la contaminación *in silico*, es aquella producida durante el análisis computacional de las muestras, siendo este tipo de contaminación la más frecuente (Deb, 2022). Ambas pueden generar dos tipos de contaminación: redundante y no redundante. La redundante se refiere a la repetición de un segmento genómico en un genoma, mientras que la no redundante se puede producir por fragmentos extraños de múltiples fuentes procedentes de linajes no relacionados que originan genomas quiméricos debido al reemplazamiento de un segmento genómico del genoma original o debido la expansión del mismo por la incorporación de dichos fragmentos (Orakov et al., 2021; Cornet & Baurain, 2022; Deb, 2022).

Además, el volumen de datos generados derivados de estos estudios requiere de herramientas bioinformáticas capaces de realizar análisis de calidad automatizados, de forma rápida y precisa. Un ejemplo es *GUNC*, que permite detectar el quimerismo y la contaminación de genomas procarióticos, cuantificándolo incluso a bajos niveles. Esta herramienta permite obtener una puntuación de separación de clados (CSS), permitiendo indicar la profundidad filogenética aproximada en la que divergieron los

distintos genomas de origen, y una puntuación de representación de referencia (RSS), que permite estimar cómo de cerca está representado un genoma de consulta con respecto a un conjunto de referencia (Orakov et al, 2021). Otra herramienta utilizada para la evaluación de la calidad del genoma es *CheckM* (Parks et al., 2015).

A su vez, en el análisis de genomas bacterianos, se puede destacar la herramienta *PRODIGAL*, que permite predecir la secuencia genes codificantes microbianos (Hyatt et a, 2010). Dicha información es el punto de partida para llevar a cabo estudios de caracterización funcional de proteínas, búsqueda de regiones altamente conservadas entre familias génicas, etc. Este tipo de herramientas permiten conocer la secuencia de aminoácidos de manera rápida y automatizada (Hyatt et a, 2010).

En el presente trabajo también interesa conocer las familias proteicas del género de *Bifidobacterium* que son comunes a todas las especies y cepas, es decir, identificar el proteoma central. Es importante resaltar que el proteoma central es una parte de lo que se denomina pan-proteoma. El pan-proteoma se refiere al conjunto de genes codificantes no redundantes pertenecientes a organismos taxonómicamente relacionados, y está compuesto por: proteoma central, proteoma accesorio y proteoma único. El proteoma único hace referencia a los genes codificantes específicos de cada genoma, mientras que el proteoma accesorio se define como aquel cuyos genes codificantes están presentes en 2 o más especies o cepas (Costa et al., 2020). Además, en el presente TFM se emplea el término de proteoma extendido para explicar los genes codificantes compartidos por todos los genomas con mayor grado de variabilidad. Una herramienta ampliamente utilizada para poder establecer familias proteicas a lo largo del pan-proteoma es *CD-HIT*, que permite realizar agrupamientos múltiples de secuencias, ya sea de aminoácidos o de nucleótidos, en “clusters” que compartan un porcentaje de identidad especificado por el usuario. Con esta información se pueden llevar a cabo distintos análisis, entre ellos, de anotación funcional, estudio de la estructura o función de las familias proteicas, etc. Esta herramienta supone un ahorro computacional con respecto a las utilizadas anteriormente como *BLASTP* (Yooseph et al., 2007; Fushinobu & Abou Hachem, 2021; Deb, 2022).

Por otro lado, las herramientas computacionales utilizadas para el análisis de alineamiento múltiple de secuencias (MSA) permiten obtener información acerca de la conservación de los aminoácidos en familias de proteínas a nivel de secuencia, y el papel funcional de los residuos en la(s) proteína(s) de interés, indicando los residuos más importantes en una familia proteica determinada. Algunos ejemplos de métodos para llevar a cabo este tipo de análisis son *MUSCLE*, *PROBCONS, T-COFFEE* o *MAFFT*, diferenciándose en cuanto a precisión y costo computacional. Por otro lado, el uso de visores como *JALVIEW* o *SEAVIEW* permite localizar regiones altamente conservadas (Benítez-Páez et al., 2012).

A través de estos MSA o de secuencias únicas, también es común realizar análisis de modelos ocultos de Markov para detectar secuencias homólogas en bases de datos. Algunas de las herramientas utilizadas para esta función son *HMMER3* y *PSI/PHI-BLAST* (Benítez-Páez et al., 2012; Prakash et al., 2017).

Con todo ello, se podrían realizar análisis de anotación y divergencia funcional. *eggNOG-mapper* es una herramienta disponible como recurso web que puede anotar funcionalmente secuencias novedosas utilizando grupos ortólogos y filogenias de la base de datos eggNOG. En cuanto al estudio de la divergencia se puede destacar DIVERGE (Gu & Vander Velden, 2002). Dicho algoritmo proporciona información acerca del desplazamiento de la tasa sitio-específica de los aminoácidos (divergencia funcional tipo I) y del desplazamiento radical de los aminoácidos (divergencia funcional tipo II) (Gu, 2006).

2. HIPÓTESIS Y OBJETIVOS

Hipótesis:

Tras el análisis del proteoma central y extendido compartido en todos los genomas estudiados del género *Bifidobacterium*, se espera encontrar familias proteicas pertenecientes a proteínas ortólogas con funciones principales propias del género, así como familias proteicas específicas de especie que sugieran evidencia de divergencia funcional relacionada con el metabolismo de los carbohidratos.

Objetivos:

El **objetivo general** del presente trabajo fue realizar un análisis masivo de genomas completos de *Bifidobacterium* con hospedador humano, identificando los genes codificantes para estudiar las funciones proteicas compartidas en el total de genomas.

Los **objetivos específicos** fueron:

- Detección y análisis de la contaminación en más de 800 genomas de especies del género *Bifidobacterium*, depositados en la base de datos BV-BRC y curados manualmente en cuanto a su asignación taxonómica vía estándar ANI.
- Predicción de los genes codificantes y agrupamiento múltiple de secuencias para la identificación del genoma central y extendido de todos los genomas estudiados.
- Alineamiento múltiple de secuencias con el fin de anotar funcionalmente las familias proteicas del proteoma central y extendido a nivel de género.
- Identificación de familias proteicas que puedan sugerir evidencia de divergencia funcional relacionada con el metabolismo de los carbohidratos.

3. METODOLOGÍA

3.1. Plan de datos

Se partió de un total de 822 genomas de bacterias pertenecientes al género *Bifidobacterium* procedentes del repositorio de dominio público Bacterial and Viral Bioinformatics Resource Center (BV-BRC) [https://www.bv-brc.org/] y dos controles

externos (*E.coli* K-12 MG1655 *y Lactobacillus gasseri* ATCC 33323). En un trabajo previo se depuraron manualmente en cuanto a su asignación taxonómica mediante la herramienta FastANI, y se indicaron los genomas pertenecientes al género *Bifidobacterium* con hospedador humano. También se indicaron las especies no descritas o mal anotadas basándose en un valor de identidad media superior al 95%, evaluando relaciones intra-especies e inter-especie.

Por tanto, se partió de archivos de genomas ensamblados en formato FASTA con extensión ".fna" (FASTA nucleotide file), que hace referencia a la secuencia de ácidos nucleicos.

Los datos con los que se han trabajado se ajustan a los principios FAIR ("Findable", "Accessible", "Interoperable", "Reusable", por sus siglas en inglés) ya al tratarse de datos procedentes de un repositorio público, son datos encontrables en BV-BRC con los siguientes filtros de búsqueda: Genome_quality "Good" y Host_group "Human". Además, se puede acceder de forma pública al código y los metadatos usados en el trabajo previo para obtener los genomas de los que se parte en el presente trabajo (https://github.com/Tonibg2/TFMbioinformatica).

Con la finalidad de que los resultados generados en el presente Trabajo Fin de Máster sean reproducibles y atiendan a los principios FAIR, se puede consultar tanto el código referente al flujo de trabajo como las bases de datos utilizadas en el siguiente GitHub (https://github.com/aidavr/TFM-Aida-Vaquero-Rey). En el **Anexo 1** se especifica con detalle los archivos subidos a dicho GitHub.

3.2. Flujo de trabajo

En primer lugar, con el software R (v4.2.2) se llevó a cabo un análisis descriptivo de la base de datos para conocer la frecuencia absoluta por especie y por subespecie. Además, se analizó la distribución y los valores perdidos de las variables de interés que hacían referencia a la calidad de los genomas (medido con *CheckM*).

Posteriormente, se procedió a utilizar el paquete de Python llamado *GUNC* (Genome UNClutterer, versión 1.0.5) para corroborar calidad de los genomas (Orakov et al., 2021).

Esta herramienta permite detectar el grado de contaminación, y cuantificar con precisión el quimerismo genómico. El código fuente de *GUNC* puede consultarse en el siguiente enlace https://github.com/grp-bork/gunc (Orakov et al., 2021).

Para el uso de *GUNC* fue necesario previamente la descarga de la base de datos de *proGenomes 2.1* de 13G (Mende et al., 2020). La instalación se realizó a través del administrador de paquetes conda (https://docs.conda.io/en/latest/), utilizando el canal Bioconda, ampliamente utilizado como distribuidor de software en bioinformática (https://anaconda.org/bioconda/gunc) (Grüning et al., 2018). Previamente fue necesario instalar *PRODIGAL* v2.6.3 para la consulta de genes codificantes (Hyatt et al, 2010) y *Diamond* v2.0.4 (Buchfink et al., 2015; Zukancic et al., 2020) para mapear las secuencias de las proteínas predichas contra genomas representativos derivados de la bases de datos *proGenomes 2.1*. De este modo, se pudo obtener las puntuaciones asociadas a la contaminación de los genomas y su representación en la referencia (Orakov et al., 2021).

Posteriormente, con el software R se interpretaron los metadatos asociados a los genomas analizados y se eliminaron los genomas distintos a *Bifidobacterium* (*E. faecalis* y *C. avidum*), los *Bifidobacterium* no relacionados como huéspedes del humano (*B.scardovii, B.gallicum* y *B.thermophilum*) y los genomas clasificados como contaminados en *GUNC*.

A continuación, se realizó una predicción de genes codificantes a través de la herramienta de código abierto *PRODIGAL* (v2.6.3), PROkaryotic DYnamic programming Gene-finding ALgorithm. Consultar código fuente en el siguiente enlace: https://github.com/hyattpd/Prodigal (Hyatt et al., 2010). Para ello, se utilizó de input los archivos FASTA con extensión ".fna". Como output se obtuvieron archivos con extensión ".gff" (formato que obtiene por defecto), así como ficheros de aminoácidos y DNA para todos los genes predichos, con extensiones ".faa" y ".fa", respectivamente.

Para el agrupamiento de genes de acuerdo a las similitudes e identidades de secuencia de aminoácidos de las especies de *Bifidobacterium* se utilizó la herramienta *CD-HIT* v4.8.1 (https://github.com/weizhongli/cdhit). *CD-HIT* clasifica las secuencias en orden de longitud decreciente, y establece la secuencia de mayor longitud como

representativa del primer grupo o "cluster". A través de "palabras cortas" (*k-mers*), *CD-HIT* compara si la secuencia consulta cumple con el umbral de identidad establecido, de manera que, si se cumple se agrupa dentro del cluster y, si no lo cumple, se convierte en la secuencia representativa del siguiente cluster (Huang et al., 2010). De esta manera, se compararon 12 grados de agrupamiento con distintos umbrales de identidad de secuencia (95%, 90%, 85%, 80%, 75%, 70%, 65%, 60%, 55%, 50%, 45% y 40%) con el fin de identificar el genoma codificante central, es decir, las proteínas del género de *Bifidobacterium* que son comunes a todas las especies y cepas. El input introducido en *CD-HIT* fueron los archivos con extensión ".faa" obtenidos en *PRODIGAL*, mientras que el output obtenido fueron dos tipos de archivos: un archivo FASTA con todas secuencias representativas y un archivo de texto con extensión ".clstr" que recogió todos los clusters generados para el umbral de identidad de secuencia seleccionado.

Una vez obtenidos los clusters para cada umbral de identidad, se depuraron los archivos ".clstr" en R para su representación gráfica y análisis. Posteriormente, se llevó a cabo el método de Elbow y el método de la Tasa de cambio para determinar el umbral de identidad idóneo para la recuperación de la mayoría de las secuencias relevantes del proteoma central. A su vez, se calculó la segunda derivada de la tasa de cambio para identificar el mínimo local.

Dichas métricas se llevaron a cabo asumiendo un margen de error del 5% para establecer el total de genes faltantes en cada uno de los ensamblajes para los 787 genomas iniciales. Por tanto, se recuperaron los clusters que contuvieron un número de secuencias igual o superior al 95% de 787, es decir, el número de los genomas totales.

Tras ello, se realizaron Alineamientos Múltiples de Secuencias (MSA) de las secuencias de proteínas utilizando *MUSCLE* v3.8.1551 (http://drive5.com/muscle/) (Edgar, 2004). Con los ficheros FASTA generados del alineamiento se construyeron los perfiles de modelos ocultos de Markov (HMM) mediante algoritmo *hmmbuild* implementado en *HMMER3* v3.3.2, lo que permitió obtener un sistema de puntuación por posición. Dichos perfiles se utilizaron como input para construir las secuencias proteicas consenso para cada cluster con el algoritmo *hmmemit* de *HMMER3* v3.3.2. (http://hmmer.org/) (Prakash et al., 2017).

Posteriormente, se utilizó el servidor web *eggNOG-mapper* v2.1.12 (http://eggnog-mapper.embl.de/) para la anotación funcional de las secuencias de proteínas consenso. Los datos obtenidos se depuraron y se representaron con el software R v4.2.2.

Todo el flujo de trabajo anteriormente explicado aparece esquematizado en la sección de Anexos (**Anexo 2**). Además, el código, tanto de Linux como de R, se puede consultar en el GitHub creado para el presente TFM (https://github.com/aidavr/TFM-Aida-Vaquero-Rey). En el "**README.md**" se encuentra el código en Linux y bash, mientras que el código en R se puede consultar en el archivo llamado "**script_R**". También se ha adjuntado el código de R, con los outputs que genera, en el **Anexo 3**.

3.3. Materiales

Se utilizó una máquina virtual con software VMware Workstation 17 Player en sistema operativo Linux (distribución Ubuntu 22.04.3 LTS) para la descarga y la ejecución de las herramientas *GUNC*, *PRODIGAL*, *CD-HIT* y *HMMER*. Debido al alto costo computacional de los análisis realizados, la máquina virtual requirió de 54.8 GB de memoria, 4 procesadores, y un espacio en el disco duro de 240 GB de 894 totales. Todo ello, se llevó a cabo en un ordenador físico (host) Intel® Core™ i3-9100F CPU (3,6 GHz), con una RAM instalada de 64 GB, y un sistema operativo Windows 10 Pro (versión 22H2).

Por otra parte, con el software R (v4.2.2) se llevó a cabo la depuración, manejo y exploración de la base de datos, así como de los outputs obtenidos en *GUNC*, *CD-HIT* y *eggNOG-mapper*. También se llevaron a cabo con R los análisis estadísticos y las representaciones gráficas.

Se utilizaron librerías básicas para el manejo, transformación, reorganización y visualización de datos como "string", "dplyr", "readr", "tidyr" y "tidyverse".

Para las representaciones gráficas se utilizaron "ggplot2", "rstatix" y "ggpubR", y para la generación de tablas se emplearon "gt" y "gtExtras".

La funcionalidad de cada una de las librerías se explica con mayor detalle en el código de R (consultar **Anexo 3**).

3.4. Análisis estadístico

Se evaluó el supuesto de normalidad mediante el test de Shapiro-Wilk. Cuando el p-valor fue menor o igual a 0.05 se aceptó la hipótesis alternativa de que los datos no seguían una distribución normal.

Al no cumplir el supuesto de normalidad, se llevó a cabo la prueba de U de Mann-Whitney, para estudiar diferencias estadísticamente significativas entre variables cualitativas no relacionadas con dos categorías en función de una variable cuantitativa continua.

También se utilizó, para distribuciones no normales, la prueba no paramétrica de Kruskal-Wallis, con el fin de determinar si existían diferencias estadísticamente significativas entre tres o más categorías en función de una variable continua.

Se determinó que las diferencias eran estadísticamente significativas cuando el valor de "p" fue igual o menor a 0.05.

Por otra parte, para estudiar la correlación de dos variables cuantitativas con distribución no normal se llevó a cabo la correlación de Spearman.

4. RESULTADOS

4.1. Análisis descriptivo de la base de datos

En primer lugar, se realizó un análisis descriptivo de la base de datos con 822 observaciones (genomas bacterianos) y 73 variables (disponible en: https://github.com/aidavr/TFM-Aida-Vaquero-Rey).

Los genomas bacterianos utilizados inicialmente para el presente trabajo mostraron una alta integridad y baja contaminación media, según los datos de *CheckM* recogidos en dicha base de datos (**Tabla 1**). No obstante, en ambos casos el porcentaje de valores perdidos fue alto (**Tabla 1**).

Tabla 1. Análisis descriptivo de la integridad y contaminación de *CheckM*, procedentes del repositorio BV-BRC.

Variable	% Valores perdidos	Media	Mediana	SD
Integridad	48.8%	99.9	100	0.5
Contaminación	74.5%	1.4	0.9	1.7

Por otro lado, se determinó la frecuencia absoluta por especie. Esto reveló la existencia de 17 especies diferentes, siendo *Bifidobacterium longum* la más representativa (**Figura 1**). No obstante, 18 genomas de *Bifidobacterium* no fueron registrados a nivel de especie. Este análisis descriptivo también permitió detectar 14 genomas de *Enterococcus faecalis* y 1 genoma de *Cutibacterium avidum*, así como 3 especies de *Bifidobacterium* (2 de *B.scardovii,* 2 de *B.gallicum* y 1 de *B.thermophilum*) no relacionadas con el humano (**Figura 1**), según se caracterizó en un trabajo previo (Bahilo, 2022).

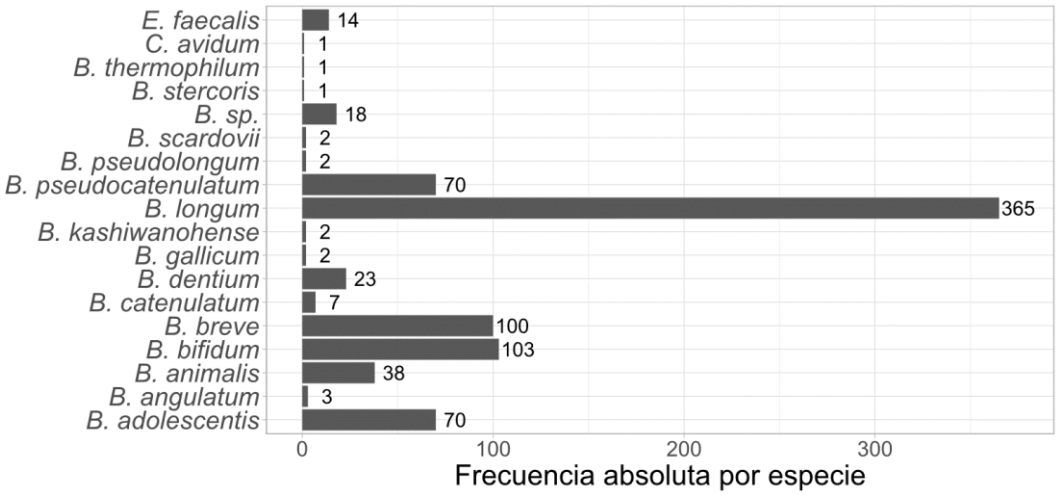


Figura 1. Frecuencia absoluta para cada una de las especies recogidas en la base de datos inicial.

Únicamente genomas pertenecientes a *B. logum*, *B. catenulatum* y *B. animalis* fueron descritos a nivel de subespecie, indicadas en la **Figura 2**. No obstante, estudiando la frecuencia relativa de cada una de estas especies, un 49.04%, un 85.71% y un 57.89% no se recogieron a nivel de subespecie, respectivamente. Las subespecies más representativas de cada una de las especies fueron *B. logum longum* (45.48%) y *B.*

animalis lactis (36.84%). La única subespecie registrada para *B. catenulatum* fue *B. catenulatum kashiwanohense* (14.29%).

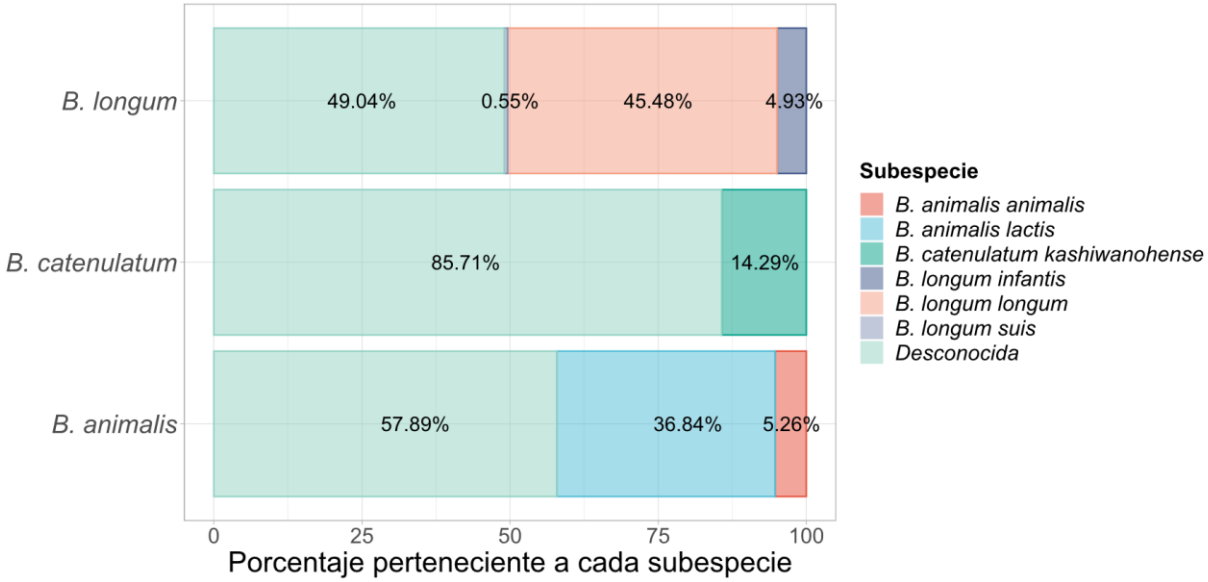


Figura 2. Porcentaje de cada una de las subespecies descritas para *B. logum*, *B. catenulatum* y *B. animales.* También se indica el porcentaje de dichas especies no descritas a nivel de subespecie (Desconocida).

4.2. Resultados *GUNC*

Teniendo en cuenta lo obtenido en el análisis descriptivo, se realizaron varios filtrados en la base de datos antes de proceder al análisis de los resultados de *GUNC*. Por un lado, se eliminaron los genomas distintos al género *Bifidobacterium* (*E. faecalis* y *C. avidum*) y los *Bifidobacterium* con hospedador no humano (*B.scardovii, B.gallicum* y *B.thermophilum*). Por otro lado, se reasignó *B. kashiwanohense* como *B. catenulatum* y *B. stercoris* como *B. adolescentis,* según lo obtenido en un trabajo previo (Bahilo, 2022). De este modo, se obtuvieron 10 especies distintas de *Bifidobacterium* (sin incluir *B. spp*), y un total de 802 genomas (ver código en **Anexo 3**).

4.2.1. *Determinar en qué especies es más común encontrar contaminaciones*

De los 822 genomas introducidos como input en *GUNC*, únicamente 15 genomas se clasificaron como contaminados (**Tabla 2**).

De estos 15 genomas contaminados, 8 correspondieron a *B. longum* (53%), seguido de 3 correspondientes a *B. spp* (20%), y únicamente 1 genoma correspondiente a *B. breve*, *B. bifidum*, *B. animalis* y *B.adolescentis*, representando un 7% en cada caso (**Figura 3**).

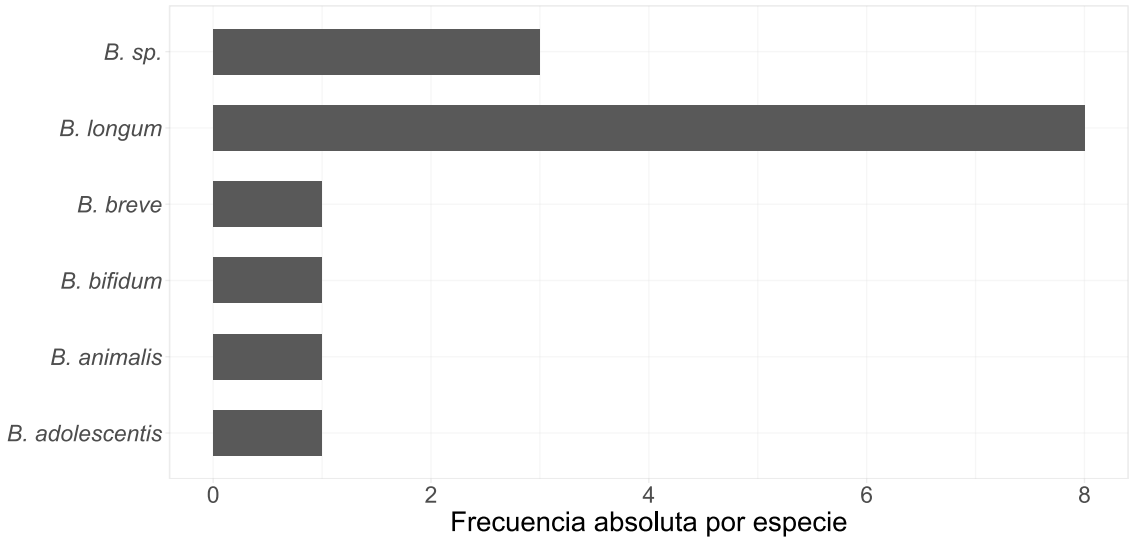


Figura 3. Frecuencia absoluta para cada una de las especies de *Bifidobacterium* contaminadas, según la herramienta de *GUNC*.

4.2.2. *Evaluación de la contaminación de los genomas*

Para conocer la calidad de los genomas se estudiaron las variables CSS (puntuación de separación de clados) y T_{eff} (número de clados efectivos distintos).

En los genomas contaminados la media y la mediana de CSS superó en todos los casos el umbral de 0.45, mientras que en los genomas no contaminados fue inferior, confirmando lo que se recoge en la literatura (**Tabla 2)** (Orakov et al., 2021). Al no cumplirse una distribución normal de los datos (**Tabla 2**), se llevó a cabo la prueba U de Mann-Whitney, que reveló diferencias estadísticamente significativas en el CSS (p-valor < 0.05) entre el grupo de los genomas contaminados y los no contaminados (**Figura 4a**).

El T_{eff}, que estima tanto la contaminación redundante como la no redundante, presentó valores medios de 0.050 (0.040 - 0.110) en los genomas contaminados, frente a 0.002 (0.000 - 0.080) en los genomas no contaminados. (**Tabla 2**). Las diferencias entre grupos se evaluaron a través de la prueba U de Mann-Whitney, dado que los datos no cumplieron el supuesto de normalidad (**Tabla 2**). Se obtuvieron diferencias estadísticamente significativas entre el grupo de genomas contaminados y no contaminados para el T_{eff} (p-valor < 0.05) (**Figura 4b**).

Por otro lado, para conocer cómo de representados estuvieron los genomas problema en la referencia se estudió el RSS (puntuación de representación en la referencia). La media, la mediana y los valores mínimos y máximos del RSS fueron similares en ambos grupos, presentando una puntuación cercana a 1, es decir, el máximo de representación en la referencia (**Tabla 2**). Se comprobó estadísticamente mediante la prueba no paramétrica U de Mann-Whitney, ya que los datos no siguieron una distribución normal (**Tabla 2**). El p-valor mayor a 0.05 indicó que las diferencias no fueron estadísticamente significativas entre ambos grupos (**Figura 4 c**).

Tabla 2. Análisis descriptivo del CSS, T_{eff} y RSS para los genomas contaminados (False) y no contaminados (True).

								Shapiro-Wilk	
	Filtro *GUNC*	**N**	**Perdidos**	**Media**	**Mediana**	**DE**	**Min-Max**	**W**	**p**
CSS	False	15	0	0.848	0.960	0.186	0.450 -1.000	0.793	0.003
	True	787	0	0.006	0.000	0.037	0.000 - 0.440	0.150	< .001
T_{eff}	False	15	0	0.054	0.050	0.018	0.040 - 0.110	0.671	< .001
	True	787	0	0.002	0.000	0.011	0.000 - 0.080	0.157	< .001
RSS	False	15	0	0.890	0.900	0.033	0.820 - 0.940	0.945	0.452
	True	787	0	0.899	0.910	0.034	0.750 - 0.970	0.930	< .001

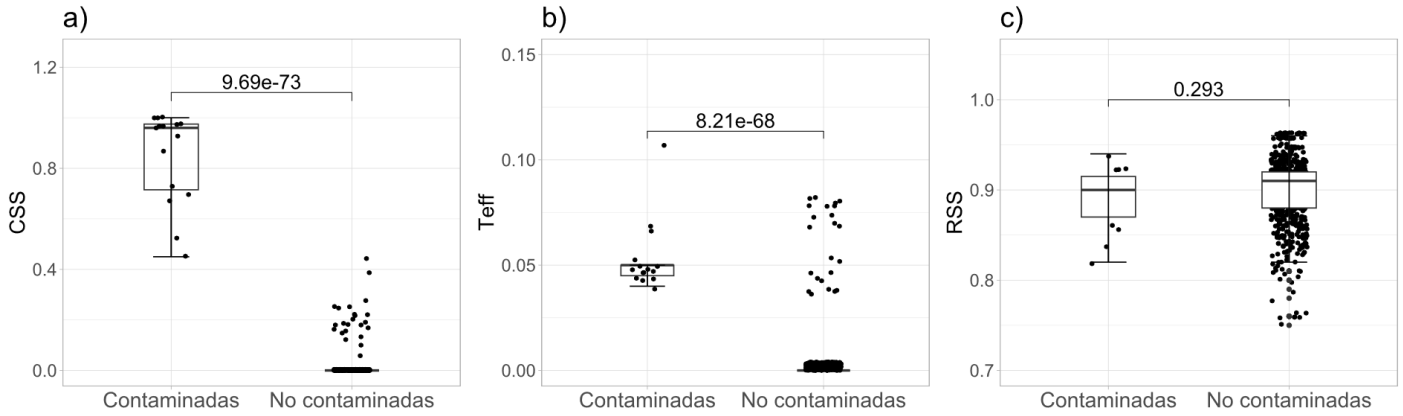


Figura 4. Representación de la distribución de los datos correspondientes al CSS, el T_{eff} y la RSS en función de si los genomas están contaminados o no (según el filtro establecido en *GUNC*). Se realizó una prueba estadística de U de Mann-Whitney: a) p-valor = 9.69^{-73}, b) p-valor = 8.21^{-68}, c) p-valor = 0.293.

Para determinar si existía una correlación entre el CSS y el RSS, se comprobó previamente la normalidad de los datos mediante la prueba de Shapiro-Wilk. El p-valor inferior a 0.05 indicó que los datos no cumplieron los supuestos para una prueba paramétrica (**Tabla 2**). Por ello, se llevó a cabo un análisis de correlación de Spearman. De la matriz de correlaciones se obtuvo un coeficiente de Spearman cercano a 0 (Rho de Spearman = -0.166), indicando una correlación débil, y un p-valor inferior a 0.05, que demostró una correlación significativa entre las variables. A mayor CSS (mayor probabilidad de que el genoma esté contaminado), menor RSS (menor representación del genoma consultado en el genoma de referencia).

Del mismo modo, se estudió si existía una correlación entre el T_{eff} y el CSS. Se realizó un análisis de correlación de Spearman, dado que la distribución de los datos no era normal (p- valor < 0.05 en la prueba de Shapiro-Wilk, **Tabla 2**). Se demostró una fuerte correlación entre ambas variables (Rho de Spearman = 0.999) estadísticamente significativa (p-valor < 0.001). Esto se traduce en que a medida que aumenta el CSS también lo hace el $T_{eff.}$

4.2.3. *Determinar si existe sesgo de contaminación en función de la tecnología de secuenciación*

Previamente, se depuró la base de datos con R v4.2.2 para reducir el número de categorías para la variable "Plataforma de secuenciación". Inicialmente se partió de 42 plataformas de secuenciación diferentes y se consiguió reducir a 8 categorías distintas. Para ello, se siguieron los siguientes criterios: 1) se agruparon todas las que se referían a la misma plataforma pero recibían distinto nombre en la base de datos; 2) aquellos genomas que habían sido obtenidos a través de varias plataformas de secuenciación se agruparon dentro de la categoría "Varios"; 3) todas las plataformas referentes a Illumina (MiSeq, HiSeq, NextSeq) se agruparon en una única categoría; 4) las dos plataformas de Roche 454 (454 Life Sciences GS FLX Titanium Chemistry y 454 Life Sciences GS FLX 454 System) se agruparon en una única categoría. Por último, se eliminaron todos aquellos genomas que no contenían ningún registro de plataforma de secuenciación ("NA").

Se observó que la mayoría de los genomas se habían obtenido mediante secuenciación con la plataforma de Illumina (**Tabla 3**).

Tabla 3. Tabla de contingencia de las distintas plataformas de secuencias en función de la contaminación de los genomas.

	Contaminación según *GUNC*		
Plataforma de secuenciación	**Contaminadas**	**No contaminadas**	**Total**
454	0	19	19
Illumina	12	615	627
Ion Torrent	0	31	31
Oxford Nanopore GridION	0	1	1
PacBio	0	70	70
Sanger	0	2	2
Varios	2	18	20
Total	14	756	770

Para comprobar si existía un sesgo de contaminación en función de la plataforma de secuenciación empleada (CSS ~ Plataforma de secuenciación), se llevó a cabo una prueba no paramétrica Kruskal-Wallis, dado que los datos no siguieron una distribución normal (prueba de Shapiro-Wilk con p-valor inferior a 0.05). Los resultados revelaron diferencias estadísticamente no significativas del valor del CSS entre las distintas plataformas de secuenciación empleadas (p-valor = 0.2636).

Por otro lado, se evaluó la posibilidad de un sesgo de contaminación en función de la plataforma de secuenciación empleada, teniendo en cuenta únicamente los genomas contaminados. La prueba de Shapiro-Wilk confirmó la no normalidad de los datos y la prueba de U de Mann-Whitney determinó que no existían diferencias estadísticamente significativas (p-valor = 0.854) (**Figura 5**).

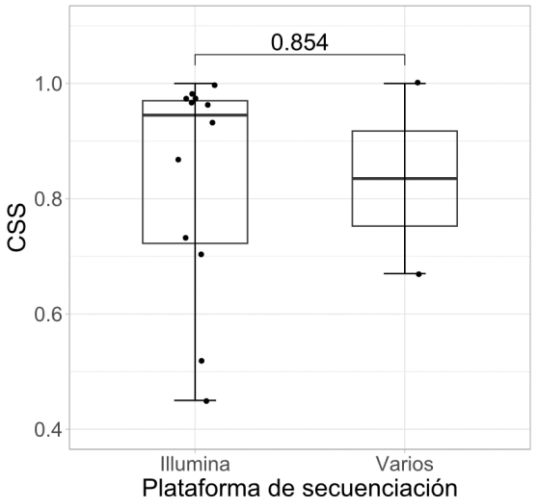


Figura 5. Representación de la distribución de los datos de CSS pertenecientes a los genomas contaminados, en función de la plataforma de secuenciación. Se realizó una prueba estadística de U de Mann-Whitney (p-valor = 0.854).

4.3. Resultados *PRODIGAL*

Para llevar a cabo la predicción de los genes codificantes mediante *PRODIGAL*, se eliminaron previamente los controles externos (*E.coli* K-12 MG1655 y *Lactobacillus gasseri* ATCC 33323), así como los archivos FASTA (.fna) pertenecientes a *E. faecalis*, *C. avidum*, *B.scardovii, B.gallicum* y *B.thermophilum*. Por último, se eliminaron los 15

genomas clasificados como contaminados por *GUNC*, obteniendo un total de 787 genomas para utilizar como input de *PRODIGAL* (**Anexo 3**).

De output, se obtuvieron 787 archivos con extensión ".faa", ".gff" y ".fa". Los archivos con extensión ".faa" se utilizaron como input de *CD-HIT*, cuyos resultados se muestran a continuación.

De los 787 genomas analizados se obtuvieron una media de 1946 genes codificantes (SD= ± 184,49 y CI 95%= 1933-1958).

4.4. Resultados del agrupamiento de múltiple de secuencias

4.4.1. *Identificación del pan-proteoma y el proteoma central del género Bifidobacterium*

Se representó el pan-proteoma de los 787 genomas analizados con el fin de identificar el proteoma central para cada uno de los umbrales de identidad de secuencia de aminoácidos, definidos en *CD-HIT* (desde el umbral 50% hasta el 90%) (**Figura 6**). Se observó que el número de secuencias contenidas en cada cluster difirió en función del porcentaje de identidad de secuencia. A su vez, se obtuvo un mayor número de clusters a medida que se aumentó el umbral de identidad de secuencia (**Figura 6, Tabla 4**).

Por otra parte, se observó mayor densidad de clusters con un número bajo de secuencias, correspondiendo al proteoma único y al proteoma accesorio, lo que indicó una gran variabilidad del pan-proteoma de *Bifidobacterium*. Sin embargo, la proporción de clusters obtenidos con un número de secuencias representativas del proteoma central (en torno a 750) fue mucho menor (**Figura 6**).

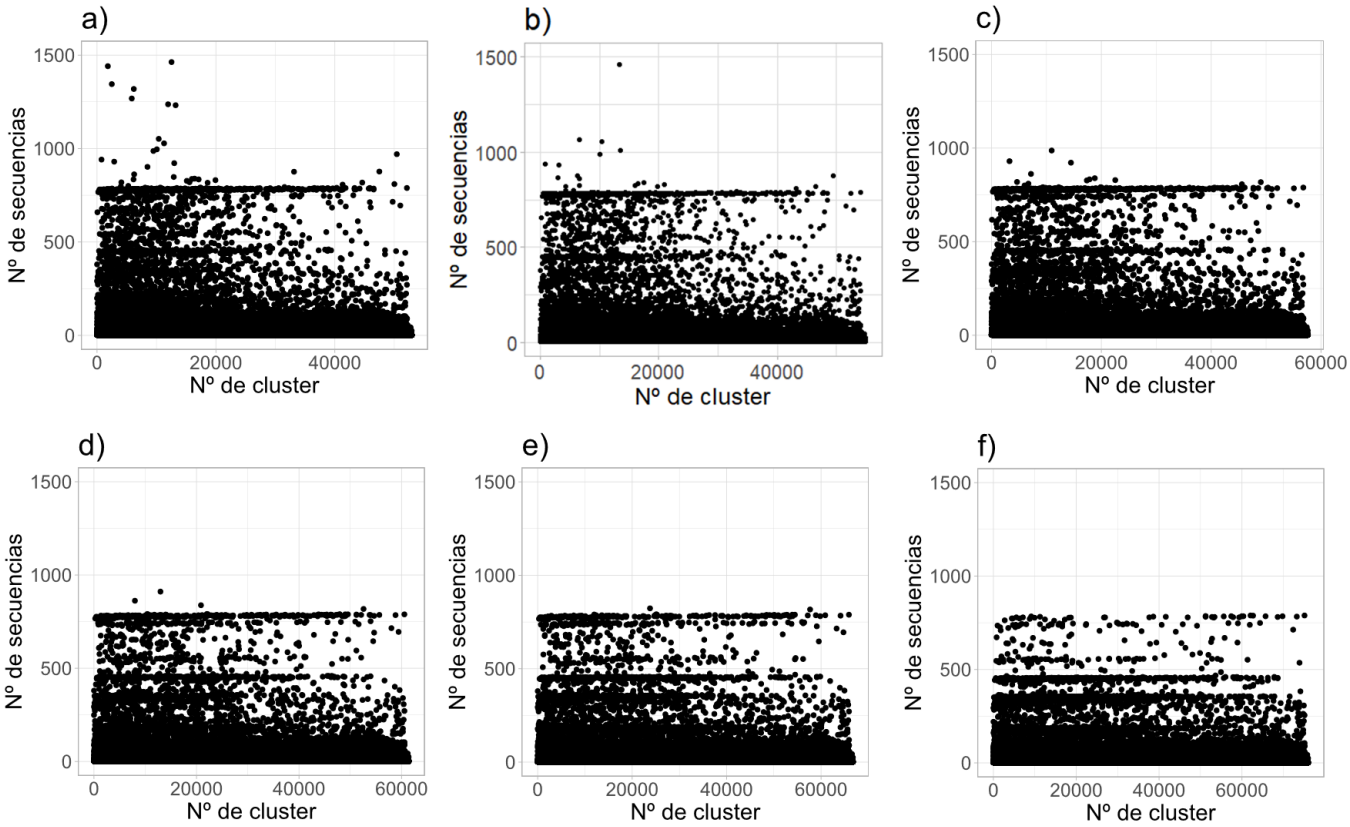


Figura 6. Representación del número de secuencias (eje y) en función del número de clusters generados por *CD-HIT* (eje x) para distintos umbrales de identidad de secuencia: a) 40%, b) 50%, c) 60%, d) 70%, e) 80%, f) 90%.

Tabla 4. Número de clusters proteicos obtenidos para cada umbral de identidad de secuencia.

Identidad de secuencia	Nº de cluster proteicos
40%	52939
45%	53888
50%	54913
55%	56173
60%	57607
65%	59358
70%	61388
75%	63786
80%	66741
85%	70482
90%	75975
95%	88086

4.4.2. *Método de Elbow y Tasa de cambio*

El método de Elbow y la Tasa de Cambio revelaron que el umbral de identidad de 85% fue el que proporcionó la mejor información sobre el proteoma central, es decir, donde se capturaron la mayoría de las secuencias relevantes para el proteoma central (**Figura 7**). Pese a que la representación en el método de Elbow podría sugerir que a partir las identidades 85, 80 y 75 se produce un descenso en la curva menos pronunciado (**Figura 7a**), se corroboró con la Tasa de Cambio que el punto donde la métrica se estabiliza, alcanzando un mínimo local y, por tanto, donde muestra un cambio más lento, fue el umbral de identidad de 85 (**Figura 7b**).

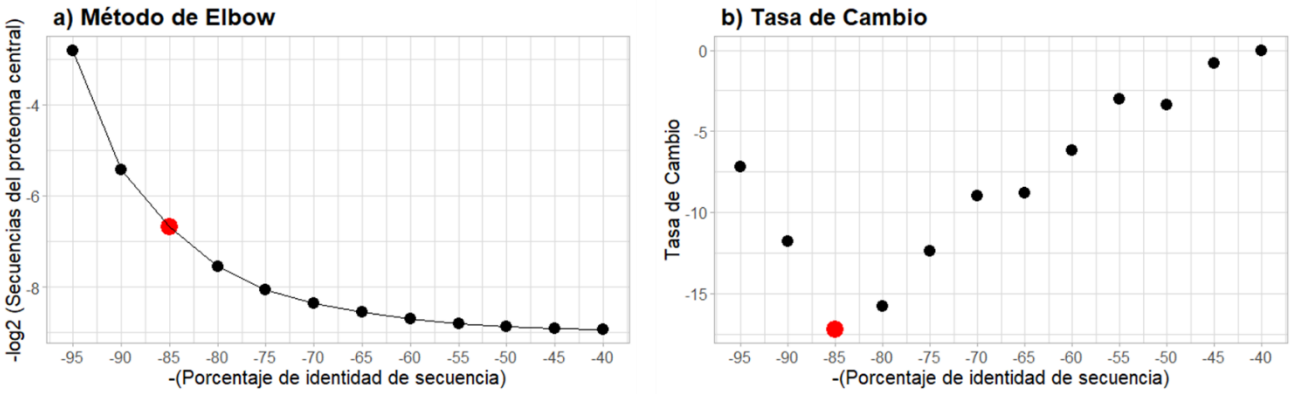


Figura 7. Representación de las secuencias que conforman el proteoma en función de los distintos umbrales de identidad utilizados en *CD-HIT*, mediante dos métricas: **a)** Método de Elbow, **b)** Método de la Tasa de Cambio (cálculo de diferencia entre los valores de la métrica en pasos sucesivos).

4.5. Anotación funcional con *eggNOG-mapper*

Por último, se llevó a cabo la anotación funcional del proteoma central con *eggNOG-mapper* para los umbrales de identidad de 50 y 85. Pese a que el umbral de identidad de 50 no fue soportado por el método de Elbow ni la Tasa de Cambio, se decidió incluir en el análisis con el objetivo de identificar familias proteicas CAZy (relacionadas con el metabolismo de los carbohidratos), ya que en este umbral se recuperaron mayor número de familias proteicas.

Para el umbral de identidad de 50 se obtuvieron 463 familias proteicas en el proteoma central (23.8% sobre la media del tamaño de genes/genoma), mientras que para el de 85 se obtuvo 101 (5.2% sobre la media del tamaño de genes/genoma).

Teniendo en cuenta los dos umbrales, la mayoría familias proteicas COG (clusters de grupos ortólogos) estaban relacionadas con funciones de traducción, estructura y biogénesis ribosómica en primer lugar (18.58% y 34.62% para los umbrales de 50 y 85, respectivamente), y de transporte y metabolismo de aminoácidos en segundo lugar (14.41% y 12.5% para los umbrales de 50 y 85, respectivamente) (**Figura 8**).

Las categorías COG con menor porcentaje de familias proteicas en el umbral de 50 fueron la “V” (mecanismos de defensa) con un 1.04% y la “U” (tráfico intracelular, secreción y transporte vesicular) con un 1.46%, relacionadas con el metabolismo, y la “D” (control del ciclo celular, división celular, partición de cromosomas) con un 1.25%, relacionada con procesos celulares y señalización (**Figura 8a**).

En el umbral de 85, al ser mucho más restrictivo, se perdieron las categorías “V” y “D”. Por tanto, las categorías con menor proporción en este umbral fueron la “U” con un 0.96%, la “P” (trasporte y metabolismo de iones inorgánicos) con 1.92%, relacionadas con el metabolismo, y la “M” (biogénesis de la pared celular/membrana/envolvente) con un 1.92%, relacionadas con los procesos celulares y señalización (**Figura 8b**).

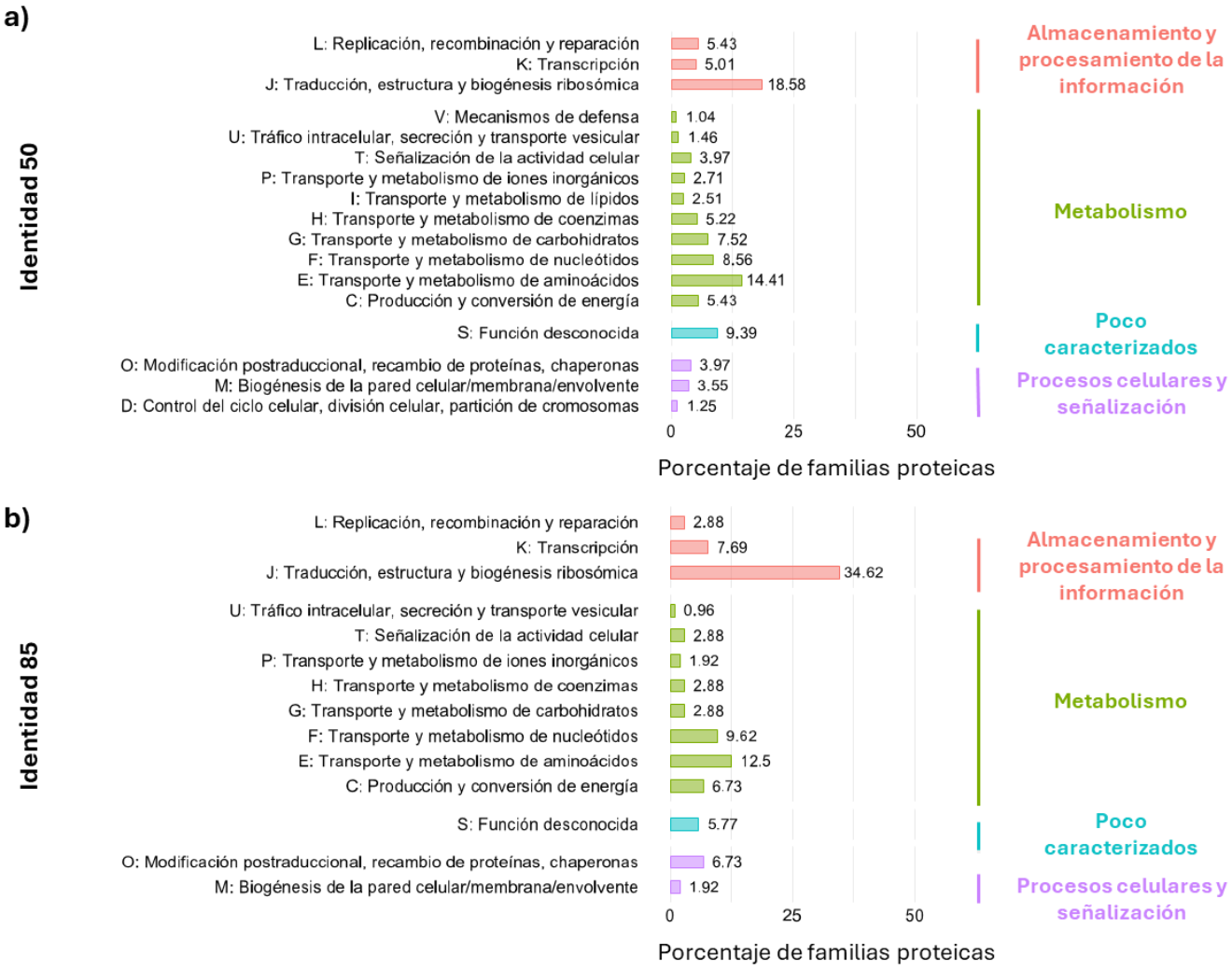


Figura 8. Porcentaje de familias proteicas COG obtenidas de la anotación funcional con *eggNOG-mapper* para los dos umbrales de identidad de secuencia estudiados: a) Identidad 50 (correspondiente al proteoma extendido); b) Identidad 85 (correspondiente al proteoma central).

De manera global se calcularon los porcentajes correspondientes a cada una de las 4 categorías principales que engloban a las categorías COG: "almacenamiento y procesamiento de la información", "metabolismo", "poco caracterizados" y "procesos celulares y señalización". Se observó que la categoría de metabolismo fue la más representada en el umbral de 50 (52.83%), mientras que en el umbral de 85 la categoría con mayor porcentaje fue almacenamiento y procesamiento de la información (45.19%).

No obstante, la segunda categoría más representada en el umbral de 85 fue la categoría de metabolismo (40.37%) (**Tabla 5**).

Tabla 5. Porcentaje obtenido para cada una de las categorías principales según la clasificación COG.

Umbral de identidad de secuencia	Almacenamiento y procesamiento de la información	Metabolismo	Poco caracterizados	Procesos celulares y señalización
50	29.02%	52.83%	9.39%	8.77%
85	45.19%	40.37%	5.77%	8.65%

Por último, también se identificaron las familias CAZy, obtenidas a través de la anotación funcional con *eggNOG-mapper*. En la **Tabla 6** se puede observar que para el umbral de 50 se obtuvieron 7/10 familias CAZy distintas. Sin embargo, utilizando una identidad del 85% en la secuencia de aminoácidos para realizar los clusters (correspondientes a las familias proteicas) no se identificó ninguna familia CAZy.

Las familias CBM48 y GH13 aparecieron en dos clusters, mientras que el resto de las familias CAZy fue específica de cluster. Por otro lado, se obtuvo un mayor número de la familia CAZy GH13 correspondiente a las glicosidas hidrolasas, en comparación con el resto (**Tabla 6**).

Tabla 6. Número de proteínas CAZy obtenidas por *eggNOG-mapper* para los umbrales de identidad de secuencia 50 y 85**.** GT: familia de la glucosiltransferasa. GH: Familia de las glicosidas hidrolasas. Los números a continuación de las letras que designan la familia proteica se refieren a la subfamilia.

Clasificación CAZy	Nº de secuencias 50%	Nº de secuencias 85%
GT35	1	0
CBM48	2	0
GH77	1	0
GH13	3	0
GH38	1	0
GT4	1	0
GT28	1	0

5. DISCUSIÓN

En este trabajo se identificó el proteoma central y extendido del género *Bifidobacterium* y se estudiaron las funciones de las familias proteicas que lo componen, abordando dicha tarea con la perspectiva de sentar las bases para un posterior estudio de la divergencia funcional y evolución molecular de las dos especies de *Bifidobacterium* más predominantes en la microbiota intestinal humana: *B. longum* y *B.breve*. Ello les conferiría una ventaja evolutiva frente a otras especies del género menos abundantes y/o les brindaría una eficiencia diferencial entre ellas para la obtención de energía a partir de la utilización de diferentes carbohidratos.

La evaluación de la contaminación con *GUNC* reveló que la mayoría de los genomas contaminados pertenecieron a la especie *B. longum* (**Figura 3**). Esto fue congruente con lo obtenido en el análisis descriptivo, ya que esta especie supuso un 44.4% de todos los genomas iniciales (365/822) (**Figura 1**).

Por otra parte, únicamente se encontraron 15/822 genomas clasificados como contaminados (valores de CSS < 0.45) (**Figura 4a**). Esto sugiere que las secuencias eran de gran calidad, lo que concuerda con los análisis descriptivos realizados de la contaminación e integridad obtenidos por la herramienta *CheckM* (metadata del repositorio BV-BRC), pese a un alto porcentaje de datos perdidos para dichas variables (**Tabla 1**).

CheckM es una herramienta ampliamente utilizada para evaluar la calidad de los genomas procarióticos (Parks et al., 2015; Orakov et al., 2021; Cornet & Baurain, 2022) que al igual que *BUSCO*, *EukCC*, *ConFinR* y *Forty-Two* utiliza conjuntos de genes marcadores para estimar la contaminación e integridad de los genomas (Cornet & Baurain, 2022). No obstante, algunos de estos métodos conllevan sesgos en las estimaciones de la calidad del genoma, sobre todo a la hora de estimar la contaminación no redundante (Cornet & Baurain, 2022). En el caso de *CheckM* se debe tener en cuenta una posible sobreestimación de la integridad y subestimación de la contaminación, además de presentar limitaciones a la hora de estimar la calidad de genomas reducidos de linajes nuevos (Parks et al., 2015; Orakov et al., 2021). Por ello, en el presente Trabajo

de Fin de Máster se decidió utilizar *GUNC*, que se basa en la homogeneidad del linaje de contigs individuales para establecer la contaminación del genoma (Orakov et al., 2021). Además, *GUNC* tiene en cuenta la confusión de contaminación por transferencia horizontal de genes y actualmente es la única herramienta que ofrece una puntuación de la representación en la referencia (Cornet & Baurain, 2022; Orakov et al., 2021). Sin embargo, es una herramienta novedosa, de la cual actualmente no existe bibliografía aplicando su funcionalidad, a excepción del trabajo publicado por Tadrent et al. (2022), que incorporaron *GUNC* en un flujo de trabajo automatizado basado en la construcción de genomas procarióticos a partir de metagenomas; y la revisión realizada por Cornet & Baurain (2022).

La mayoría de los valores de CSS en el grupo de genomas no contaminados se agruparon en torno a 0. No obstante, se pudo observar un grupo de valores atípicos que mostraron valores de CSS más altos, sin llegar a superar el umbral de 0.45 (**Figura 4a**). Es posible que estos genomas tomasen valores más altos de CSS porque algunos contigs correspondiesen a genes que no se encuentran bien representados en los genomas de referencia con los que se comparan. Esto se verificó encontrando el mismo número de valores atípicos con puntaciones de representación en la referencia menores (**Figura 4c**). Además, esto podría ser apoyado con la correlación de Spearman realizada entre el CSS y el RSS (a mayor CSS menor RSS), aunque el coeficiente de correlación fue bajo. Por otra parte, al analizar el T_{eff}, también se encontró el mismo número de valores atípicos en el grupo de genomas no contaminados con valores similares a los genomas contaminados (**Figura 4b**), lo que indica que dichos valores atípicos tendrían un número efectivo de clados distintos similar al de los genomas contaminados. Esto concuerda con lo obtenido en la correlación de Spearman realizada entre el CSS y el T_{eff} (a mayor CSS, mayor T_{eff}). Por tanto, todo ello sugiere que: **1)** Dichos valores atípicos podrían corresponder a falsos negativos, asignados incorrectamente al grupo de genomas no contaminados, ya que como se recoge en la bibliografía, *GUNC* no es capaz de detectar con precisión el quimerismo de los clados de linajes que no aparecen bien representados en la referencia (Orakov et al., 2021); **2)** Dichos valores atípicos contienen contigs pertenecientes a linajes novedosos en relación a la referencia, que podrían haberse obtenido a través de transferencia horizontal entre linajes evolutivamente lejanos y,

por ello, reciben valores más bajos de RSS y altos de T_{eff} (Choi & Kim, 2007; Orakov et al., 2021).

No obstante, en general, se obtuvo una alta representación en el genoma de referencia (RSS), independientemente de si los genomas se clasificaron como contaminados como no contaminados (valor mínimo = 0.75), lo que supone un indicador de la especificidad (probabilidad de cometer falsos positivos o error tipo I en estadística). Sin embargo, los autores no hacen referencia a la sensibilidad (probabilidad de cometer falsos negativos o error tipo II en estadística) (Orakov et al., 2021). Es por ello que, como línea futura de esta investigación se propone una inspección manual de los genomas que presentaron valores atípicos clasificados como "no contaminados" a través de diagramas de Sankey. Otra línea futura podría ir dirigida a la comparación de estos resultados con los obtenidos utilizando *Kraken2*, que junto con *GUNC*, es una de las herramientas con las que se han obtenido resultados de alta precisión en la estimación de la contaminación en genomas procarióticos. A su vez, al igual que *GUNC*, evalúa la calidad del genoma teniendo en cuenta el genoma completo (Cornet & Baurain, 2022).

Por otra parte, es sabido que una de las causas de contaminación de los genomas causadas por problemas experimentales puede deberse a la secuenciación (Cornet & Baurain, 2022). Sin embargo, no se obtuvieron diferencias estadísticamente significativas en el CSS entre plataformas de secuenciación empleadas para la obtención de cada uno de los genomas. Es posible que no se observasen diferencias significativas debido al bajo tamaño muestral que suponen los genomas contaminados, es decir, los que presentaron valores de CSS superiores a 0.45 (**Tabla 3, Figura 5**).

La obtención de altos niveles de RSS de media, junto con la depuración realizada de genomas contaminados, así como la eliminación de genomas no pertenecientes al género *Bifidobacterium* y *Bifidobacterium* no relacionados con hospedador humano, sostuvo la consistencia de los análisis realizados en el presente Trabajo de Fin de Máster.

Por otro lado, de los 12 umbrales de identidad estudiados (40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 y 95), los resultados revelaron que el umbral de 85 fue el que mejor explicaba la configuración más probable del proteoma central (**Figura 7**). No obstante, únicamente

se obtuvieron 101 familias (5.2% sobre la media de genes de un genoma) correspondientes al proteoma central para este umbral de identidad, a diferencia de las 463 familias obtenidas para el umbral de 50, correspondiente al proteoma extendido. Esto podría explicarse porque el umbral de 50 es mucho menos restrictivo que el umbral de 85, es decir, las secuencias de aminoácidos únicamente comparten un 50% y, por ende, es posible que se recuperen familias proteicas derivadas de genes parálogos, en vez de únicamente de ortólogos. Los genes ortólogos corresponden a aquellos genes que divergieron por especiación evolutiva, presentando una función biológica equivalente, mientras que los parálogos divergieron por eventos de duplicación seguidos de especiación y están menos conservados que los ortólogos (Costa et al., 2020).

Por otro lado, hay que tener en cuenta el número de genomas analizados. En el presente TFM se llevó a cabo un análisis de 787 genomas pertenecientes al género *Bifidobacterium*. Por tanto, era esperable que el tamaño del proteoma central fuese reducido. Otros estudios han demostrado que a medida que aumenta el número de genomas analizados se reduce el tamaño del proteoma central (Heo et al., 2020; Sun et al., 2015). También hay que tener en cuenta el número de especies analizadas, dado que la probabilidad de compartir genes se reduce a medida que se incorporan especies (Mosquera-Rendón et al, 2016). En este TFM se analizaron 10 especies de *Bifidobacterium* distintas, es decir, justo el número en el que la bibliografía recoge un drástico descenso de genes compartidos (Satti et al., 2018).

El trabajo publicado por Mosquera-Rendón et al. (2016) también apoyó la elección del umbral de 85% de identidad de secuencia, ya que obtuvieron un proteoma central que representaba un 14% sobre el total de genomas analizados en *Pseudomonas aeruginosa*. Teniendo en cuenta los resultados obtenidos sobre la predicción de genes codificantes del pan-genoma de *Bifidobacterium*, el genoma codificante central correspondió con aproximadamente un 5.2%. Es comprensible que el porcentaje obtenido sea menor que en el trabajo de Mosquera-Rendón et al. (2016), porque dichos autores obtuvieron el proteoma central en una única especie, mientras que en el presente TFM se obtuvo a nivel de género. En otro estudio obtuvieron 724 genes

pertenecientes al proteoma central de *Bifidobacterium* de los 6980 genes totales que conformaron el pan-genoma, por lo que el genoma central supuso aproximadamente un 10% para 19 genomas de 9 especies distintas de este género (Lukjancenko et al., 2012).

Por otra parte, se considera que dos genes están conservados y pertenecen a la misma familia génica cuando comparten un 50% de identidad en su secuencia de aminoácidos (Lukjancenko et al., 2012). Por ello, también se evaluó el umbral de identidad de 50, referenciándolo como proteoma extendido, es decir, aquel que es compartido por todos los genomas, pero en el que se asume mayor grado de variabilidad génica codificante al ser menos restrictivo en la agrupación de ortólogos. Esto explica que el genoma codificante extendido correspondiese con un mayor porcentaje del genoma total (23.8%) que el obtenido en el genoma codificante central (5.2%). Además, en el este umbral es donde se identificaron familias proteicas CAZy, es decir, familias de enzimas activas de carbohidratos. Este hallazgo es de gran importancia puesto que se sabe que en el pan-proteoma de *Bifodobacterium* existe un alto porcentaje de enzimas dedicadas al metabolismo de los carbohidratos, jugando un papel fundamental en la degradación de los glicanos derivados de la dieta, así como en los derivados del hospedador (Abdelhamid & El-Dougdoug, 2021). Por tanto, una futura línea de investigación del presente trabajo irá dirigida al estudio de la variación en los aminoácidos que puedan conducir a una divergencia funcional, poniendo el foco en las especies *B. longum* y *B breve,* con el fin de comprender sus adaptaciones específicas en la utilización de los carbohidratos en el tracto gastrointestinal humano. Estudios llevados a cabo en 95 genomas de *B. bifidum* encontraron familias proteicas CAZy que incluyeron glicosihidrolasas, glicotransferasas, módulos de unión a carbohidratos y enterasas de carbohidratos (Abdelhamid & El-Dougdoug, 2021). En el presente TFM *B. bifidum* fue la segunda especie más representada, después de *B. longum*, con una frecuencia absoluta similar a *B. breve* (**Figura 1**).

Las categorías COG más representadas en el proteoma para los dos umbrales de identidad estudiados fueron la “J” (Traducción, estructura y biosíntesis ribosómica) y la “E” (transporte y metabolismo de aminoácidos) (**Figura 8**), coincidiendo con lo recogido

por los autores Lukjancenko et al. (2012). Además, los resultados mostrados en el presente TFM también demuestran que el porcentaje correspondiente a la categoría "G" (función de metabolismo y transporte de carbohidratos) se obtuvo conforme a lo esperado en el proteoma extendido de *Bifidobacterium* (6.8%) (Lukjancenko et al., 2012), suponiendo un 7.52% (**Figura 8a**). Sin embargo, para el proteoma central se obtuvo un 2.88% en esta categoría COG (**Figura 8b**).

Otros trabajos analizaron las categorías COG en el pan-proteoma de *Bifidobacterium*. En el trabajo publicado por Satti et al., (2018) determinaron que la categoría "E" fue la tercera más representada en el pan-proteoma de *Bifidobacterium* (7%), mientras que el primer puesto correspondió a funciones desconocidas (29%), seguidas de las categorías "R" (Predicción de función general únicamente) y la "G" (Metabolismo y transporte de carbohidratos), representando ambas un 8%. Resultados similares se reportaron en el trabajo de Lukjancenko et al. (2012), donde las familias COG más sobre-representadas correspondieron con funciones poco caracterizadas (R y X), metabolismo (14.4% G y 11.2% E) y relacionadas con almacenamiento y procesamiento de la información (13.2% L y 10.6% K). Las categorías COG menos representadas correspondieron con funciones relacionadas con señalización y procesos celulares (Lukjancenko et al., 2012), lo que apoya los resultados del presente TFM (**Figura 8**).

Por último, en el proteoma extendido el mayor porcentaje de familias proteicas se clasificaron en la categoría principal COG de metabolismo (**Figura 8a**), al igual que obtuvieron otros autores (Lukjancenko et al., 2012; Satti et al., 2018).

6. CONCLUSIONES

1. La especie más representada en los 822 genomas de estudio fue *B. longum*, y a nivel de subespecie, *B.longum longum*.
2. El 98.2% de los genomas no presentó contaminación, con valores altos de representación media en la referencia, lo que sostiene la robustez de los resultados mostrados en el presente TFM.
3. De los 12 umbrales de identidad analizados para los agrupamientos múltiples de secuencias, el de 85% fue el que mejor explicó la composición del proteoma central de *Bifidobacterium*.
4. La anotación funcional a través del agrupamiento de genes ortólogos reveló 101 familias proteicas para el umbral de identidad de 85%, representando un 5.2% en el proteoma central, y 463 familias proteicas para el umbral de 50%, representando un 23.8% en el proteoma extendido.
5. La obtención de un 5.2% en el proteoma central puede explicarse por el número de genomas analizados y su estudio a nivel de género.
6. La categoría J (traducción, estructura y biogénesis ribosómica) y E (transporte y metabolismo de los aminoácidos), relacionadas con el metabolismo, fueron las más representadas en el proteoma central y extendido de *Bifidobacterium*.
7. Se identificaron 10 familias proteicas CAZy pertenecientes a 7 categorías distintas en el proteoma extendido de *Bifidobacterium*.
8. La categoría CAZy más abundante fue la GH13, que corresponde con las familias de la glicosidas hidrolasas.
9. Los hallazgos presentados en el actual TFM permiten sentar las bases para futuros estudios de divergencia funcional y evolución molecular en el metabolismo de los carbohidratos que ayuden a comprender la predominancia de *B.longum* y *B.breve* en el tracto gastrointestinal humano.

BIBLIOGRAFÍA

Bahilo, A (2023). Alineamientos de genoma completo y su uso en la identificación taxonómica de aislados bacterianos y especiación de cepas del género Bifidobacterium (trabajo de fin de máster). Universidad de Valencia, España.

Abdelhamid, A. G., & El-Dougdoug, N. K. (2021). Comparative genomics of the gut commensal Bifidobacterium bifidum reveals adaptation to carbohydrate utilization. *Biochemical and Biophysical Research Communications*, *547*, 155-161. https://doi.org/10.1016/j.bbrc.2021.02.046

Benítez-Páez, A., Cárdenas-Brito, S., & Gutiérrez, A. J. (2012). A practical guide for the computational selection of residues to be experimentally characterized in protein families. *Briefings in Bioinformatics*, *13*(3), 329-336. https://doi.org/10.1093/bib/bbr052

Benítez-Páez, A., Olivares, M., Szajewska, H., Pieścik-Lech, M., Polanco, I., Castillejo, G., Nuñez, M., Ribes-Koninckx, C., Korponay-Szabó, I. R., Koletzko, S., Meijer, C. R., Mearin, M. L., & Sanz, Y. (2020). Breast-Milk Microbiota Linked to Celiac Disease Development in Children: A Pilot Study From the PreventCD Cohort. *Frontiers in Microbiology*, *11*, 1335. https://doi.org/10.3389/fmicb.2020.01335

Buchfink, B., Xie, C., & Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, *12*(1), 59-60. https://doi.org/10.1038/nmeth.3176

Choi, I.-G., & Kim, S.-H. (2007). Global extent of horizontal gene transfer. *Proceedings of the National Academy of Sciences*, *104*(11), 4489-4494. https://doi.org/10.1073/pnas.0611557104

Cornet, L., & Baurain, D. (2022). Contamination detection in genomic data: More is not enough. *Genome Biology*, *23*(1), 60. https://doi.org/10.1186/s13059-022-02619-9

Costa, S. S., Guimarães, L. C., Silva, A., Soares, S. C., & Baraúna, R. A. (2020). First Steps in the Analysis of Prokaryotic Pan-Genomes. *Bioinformatics and Biology Insights*, *14*, 1177932220938064. https://doi.org/10.1177/1177932220938064

Cukrowska, B., Bierła, J. B., Zakrzewska, M., Klukowski, M., & Maciorkowska, E. (2020). The Relationship between the Infant Gut Microbiota and Allergy. The Role of Bifidobacterium breve and Prebiotic Oligosaccharides in the Activation of Anti-Allergic Mechanisms in Early Life. *Nutrients*, *12*(4), 946. https://doi.org/10.3390/nu12040946

Deb, S. (2022). Pan-genome evolution and its association with divergence of metabolic functions in Bifidobacterium genus. *World Journal of Microbiology & Biotechnology*, *38*(12), 231. https://doi.org/10.1007/s11274-022-03430-1

Devika, N. T., & Raman, K. (2019). Deciphering the metabolic capabilities of Bifidobacteria using genome-scale metabolic models. *Scientific Reports*, *9*(1), 18222. https://doi.org/10.1038/s41598-019-54696-9

Edgar, R. C. (2004). MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, *32*(5), 1792-1797. https://doi.org/10.1093/nar/gkh340

Fushinobu, S., & Abou Hachem, M. (2021). Structure and evolution of the bifidobacterial carbohydrate metabolism proteins and enzymes. *Biochemical Society Transactions*, *49*(2), 563-578. https://doi.org/10.1042/BST20200163

Grüning, B., Dale, R., Sjödin, A., Chapman, B. A., Rowe, J., Tomkins-Tinch, C. H., Valieris, R., & Köster, J. (2018). Bioconda: Sustainable and comprehensive software distribution for the life sciences. *Nature Methods*, *15*(7), Article 7. https://doi.org/10.1038/s41592-018-0046-7

Gu, X. (2006). A simple statistical method for estimating type-II (cluster-specific) functional divergence of protein sequences. *Molecular Biology and Evolution*, *23*(10), 1937-1945. https://doi.org/10.1093/molbev/msl056

Gu, X., & Vander Velden, K. (2002). DIVERGE: Phylogeny-based analysis for functional-structural divergence of a protein family. *Bioinformatics (Oxford, England)*, *18*(3), 500-501. https://doi.org/10.1093/bioinformatics/18.3.500

Heo, S., Lee, J.-S., Lee, J.-H., & Jeong, D.-W. (2020). *Comparative Genomic Analysis of Food-Originated Coagulase-Negative Staphylococcus: Analysis of Conserved Core Genes and Diversity of the Pan-Genome*. *30*(3), 341-351. https://doi.org/10.4014/jmb.1910.10049

Huang, Y., Niu, B., Gao, Y., Fu, L., & Li, W. (2010). CD-HIT Suite: A web server for clustering and comparing biological sequences. *Bioinformatics (Oxford, England)*, *26*(5), 680-682. https://doi.org/10.1093/bioinformatics/btq003

Hyatt, D., Chen, G.-L., Locascio, P. F., Land, M. L., Larimer, F. W., & Hauser, L. J. (2010). Prodigal: Prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, *11*, 119. https://doi.org/10.1186/1471-2105-11-119

Kashtan, N., Roggensack, S. E., Rodrigue, S., Thompson, J. W., Biller, S. J., Coe, A., Ding, H., Marttinen, P., Malmstrom, R. R., Stocker, R., Follows, M. J., Stepanauskas, R., & Chisholm, S. W. (2014). Single-cell genomics reveals hundreds of coexisting subpopulations in wild Prochlorococcus. *Science (New York, N.Y.)*, *344*(6182), 416-420. https://doi.org/10.1126/science.1248575

Kettler, G. C., Martiny, A. C., Huang, K., Zucker, J., Coleman, M. L., Rodrigue, S., Chen, F., Lapidus, A., Ferriera, S., Johnson, J., Steglich, C., Church, G. M., Richardson, P., & Chisholm, S. W. (2007). Patterns and implications of gene gain and loss in the evolution of Prochlorococcus. *PLoS Genetics*, *3*(12), e231. https://doi.org/10.1371/journal.pgen.0030231

Lee, J.-H., & O'Sullivan, D. J. (2010). Genomic insights into bifidobacteria. *Microbiology and Molecular Biology Reviews: MMBR*, *74*(3), 378-416. https://doi.org/10.1128/MMBR.00004-10

Luck, B., Engevik, M. A., Ganesh, B. P., Lackey, E. P., Lin, T., Balderas, M., Major, A., Runge, J., Luna, R. A., Sillitoe, R. V., & Versalovic, J. (2020). Bifidobacteria shape host neural circuits during postnatal development by promoting synapse formation and microglial function. *Scientific Reports*, *10*(1), 7737. https://doi.org/10.1038/s41598-020-64173-3

Lugli, G. A., Milani, C., Duranti, S., Mancabelli, L., Mangifesta, M., Turroni, F., Viappiani, A., van Sinderen, D., & Ventura, M. (2018). Tracking the Taxonomy of the Genus Bifidobacterium Based on a Phylogenomic Approach. *Applied and Environmental Microbiology*, *84*(4), e02249-17. https://doi.org/10.1128/AEM.02249-17

Lukjancenko, O., Ussery, D. W., & Wassenaar, T. M. (2012). Comparative Genomics of Bifidobacterium, Lactobacillus and Related Probiotic Genera. *Microbial Ecology*, *63*(3), 651-673. https://doi.org/10.1007/s00248-011-9948-y

Mende, D. R., Letunic, I., Maistrenko, O. M., Schmidt, T. S. B., Milanese, A., Paoli, L., Hernández-Plaza, A., Orakov, A. N., Forslund, S. K., Sunagawa, S., Zeller, G., Huerta-Cepas, J., Coelho, L. P., & Bork, P. (2020). proGenomes2: An improved database for accurate and consistent habitat, taxonomic and functional annotations of prokaryotic genomes. *Nucleic Acids Research*, *48*(D1), D621-D625. https://doi.org/10.1093/nar/gkz1002

Milani, C., Duranti, S., Bottacini, F., Casey, E., Turroni, F., Mahony, J., Belzer, C., Delgado Palacio, S., Arboleya Montes, S., Mancabelli, L., Lugli, G. A., Rodriguez, J. M., Bode, L., de Vos, W., Gueimonde, M., Margolles, A., van Sinderen, D., & Ventura, M. (2017). The First Microbial Colonizers of the Human Gut: Composition, Activities, and Health Implications of the Infant Gut Microbiota. *Microbiology and Molecular Biology Reviews: MMBR*, *81*(4), e00036-17. https://doi.org/10.1128/MMBR.00036-17

Mosquera-Rendón, J., Rada-Bravo, A. M., Cárdenas-Brito, S., Corredor, M., Restrepo-Pineda, E., & Benítez-Páez, A. (2016). Pangenome-wide and molecular evolution analyses of the Pseudomonas aeruginosa species. *BMC Genomics*, *17*, 45. https://doi.org/10.1186/s12864-016-2364-4

O'Callaghan, A., & van Sinderen, D. (2016). Bifidobacteria and Their Role as Members of the Human Gut Microbiota. *Frontiers in Microbiology*, *7*, 925. https://doi.org/10.3389/fmicb.2016.00925

Orakov, A., Fullam, A., Coelho, L. P., Khedkar, S., Szklarczyk, D., Mende, D. R., Schmidt, T. S. B., & Bork, P. (2021). GUNC: Detection of chimerism and contamination in

prokaryotic genomes. *Genome Biology*, *22*(1), 178. https://doi.org/10.1186/s13059-021-02393-0

Parks, D. H., Imelfort, M., Skennerton, C. T., Hugenholtz, P., & Tyson, G. W. (2015). CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Research*, *25*(7), 1043-1055. https://doi.org/10.1101/gr.186072.114

Prakash, A., Jeffryes, M., Bateman, A., & Finn, R. D. (2017). The HMMER Web Server for Protein Sequence Similarity Search. *Current Protocols in Bioinformatics*, *60*, 3.15.1-3.15.23. https://doi.org/10.1002/cpbi.40

Satti, M., Tanizawa, Y., Endo, A., & Arita, M. (2018). Comparative analysis of probiotic bacteria based on a new definition of core genome. *Journal of Bioinformatics and Computational Biology*, *16*(03), 1840012. https://doi.org/10.1142/S0219720018400127

Scholz-Ahrens, K. E., Schaafsma, G., van den Heuvel, E. G., & Schrezenmeir, J. (2001). Effects of prebiotics on mineral metabolism. *The American Journal of Clinical Nutrition*, *73*(2 Suppl), 459S-464S. https://doi.org/10.1093/ajcn/73.2.459s

Sgorbati, B., Biavati, B., & Palenzona, D. (1995). The genus Bifidobacterium. En B. J. B. Wood & W. H. Holzapfel (Eds.), *The Genera of Lactic Acid Bacteria* (pp. 279-306). Springer US. https://doi.org/10.1007/978-1-4615-5817-0_8

Sheil, B., MacSharry, J., O'Callaghan, L., O'Riordan, A., Waters, A., Morgan, J., Collins, J. K., O'Mahony, L., & Shanahan, F. (2006). Role of interleukin (IL-10) in probiotic-mediated immune modulation: An assessment in wild-type and IL-10 knock-out mice. *Clinical and Experimental Immunology*, *144*(2), 273-280. https://doi.org/10.1111/j.1365-2249.2006.03051.x

Sun, Z., Zhang, W., Guo, C., Yang, X., Liu, W., Wu, Y., Song, Y., Kwok, L. Y., Cui, Y., Menghe, B., Yang, R., Hu, L., & Zhang, H. (2015). Comparative genomic analysis of 45 type strains of the genus Bifidobacterium: A snapshot of its genetic diversity and evolution. *PloS One*, *10*(2), e0117912. https://doi.org/10.1371/journal.pone.0117912

Tadrent, N., Dedeine, F., & Hervé, V. (2022). SnakeMAGs: A simple, efficient, flexible and scalable workflow to reconstruct prokaryotic genomes from metagenomes. *F1000Research*, *11*, 1522. https://doi.org/10.12688/f1000research.128091.2

Yooseph, S., Sutton, G., Rusch, D. B., Halpern, A. L., Williamson, S. J., Remington, K., Eisen, J. A., Heidelberg, K. B., Manning, G., Li, W., Jaroszewski, L., Cieplak, P., Miller, C. S., Li, H., Mashiyama, S. T., Joachimiak, M. P., van Belle, C., Chandonia, J.-M., Soergel, D. A., … Venter, J. C. (2007). The Sorcerer II Global Ocean Sampling Expedition: Expanding the Universe of Protein Families. *PLoS Biology*, *5*(3), e16. https://doi.org/10.1371/journal.pbio.0050016

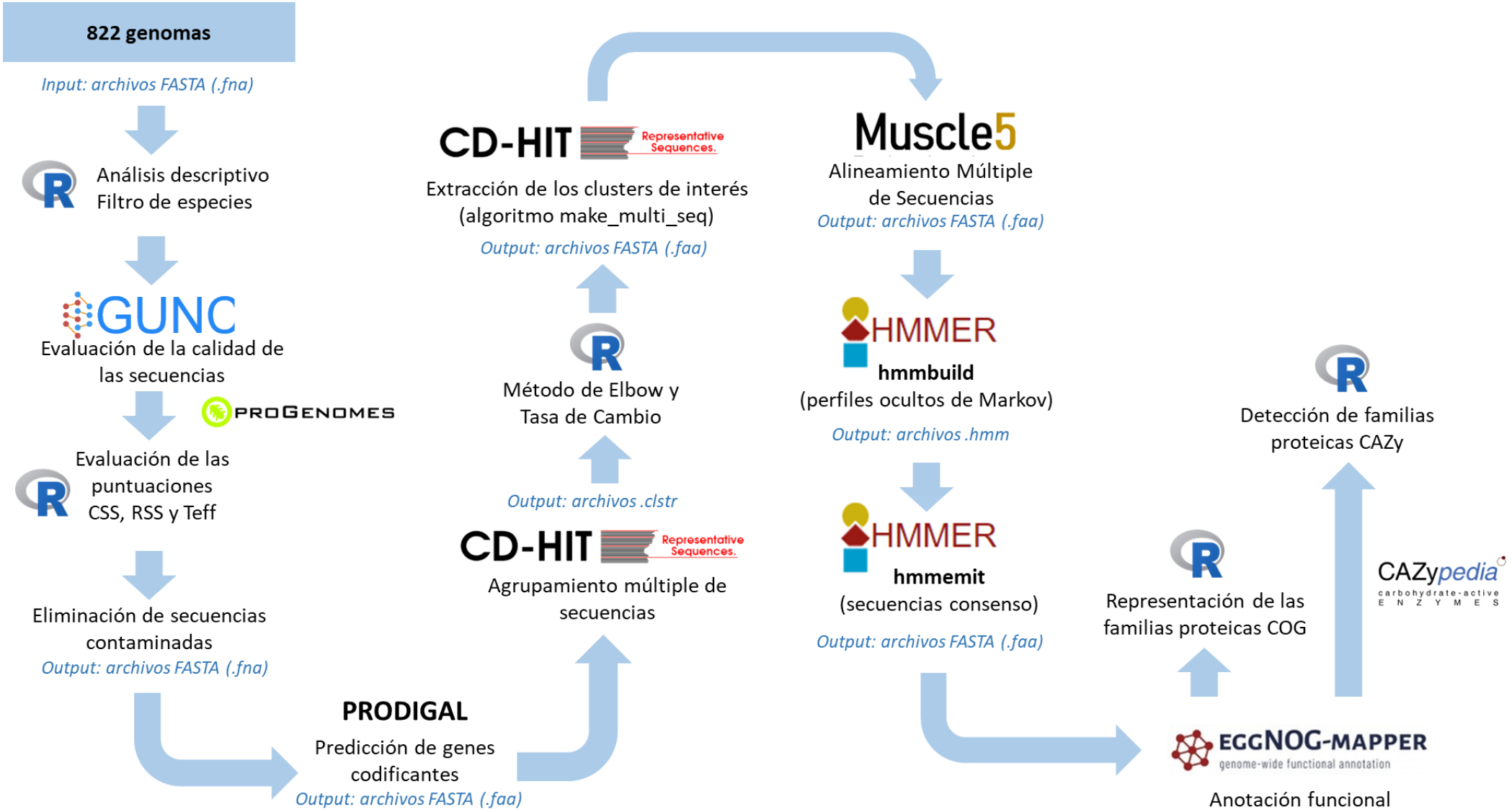
Yoshioka, H., Iseki, K., & Fujita, K. (1983). Development and differences of intestinal flora in the neonatal period in breast-fed and bottle-fed infants. *Pediatrics*, *72*(3), 317-321.

Zukancic, A., Khan, M. A., Gurmen, S. J., Gliniecki, Q. M., Moritz-Kinkade, D. L., Maddox, C. W., & Alam, M. T. (2020). Staphylococcal Protein A (spa) Locus Is a Hot Spot for Recombination and Horizontal Gene Transfer in Staphylococcus pseudintermedius. *MSphere*, *5*(5), e00666-20. https://doi.org/10.1128/mSphere.00666-20

ANEXOS

Anexo 1. Listado de archivos que se encuentran en el github creado para el presente trabajo de fin de máster (https://github.com/aidavr/tfm-aida-vaquero-rey)

README.md	Se explica el flujo de trabajo y el código utilizado en Linux, especificando inputs necesarios y outputs generados.
Script_R.qmd	Archivo R Quatro donde se especifica todo el código llevado a cabo en R, explicando el flujo de trabajo y sus respectivos pasos. Para ejecutar el código se deberán cambiar los directorios correspondientes. Este código con los outputs que genera también puede consultarse a continuación en el **Anexo 3**.
Umbrales_CD-HIT	Carpeta donde se encuentran los data frames generados tras el bucle realizado para poder leer y manipular los datos contenidos en el archivo ".clsrt" (generados en *CD-HIT*). Este bucle es computacionalmente muy costoso y por ello se han subido dichos archivos. En la carpeta se encuentran 12 archivos ".txt" correspondientes a cada uno de los umbrales de identidad que se realizó en *CD-HIT*. La manera de importarlos en R se explica en el archivo "Script_R.qmd".
BVBRC_genome.txt	Base de datos obtenida a partir del repositorio de dominio público Bacterial and Viral Bioinformatics Resource Center (BV-BRC) donde aparece reflejado el metadata de las 822 especies de *Bifidobacterium* con las que se trabaja inicialmente. Se utiliza en el análisis de los datos especificado en el código de R.
BBDD_output_GUNC	Carpeta que contiene la base de datos obtenida como output de *GUNC*. Se utiliza en el análisis de los datos especificado en el código de R. Se trata de un archivo con extensión ".tsv"
BBDD_output_eggNOG	Carpeta que contiene las bases de datos para el umbral de identidad 50 y 85, obtenidas como output del servidor *eggNOG-mapper*. Se utiliza en el análisis de los datos especificado en el código de R.



Anexo 2. Flujo de trabajo

Anexo 3. Código del flujo de trabajo en R

Para acceder al código relativo al flujo de trabajo en Linux consultar el siguiente enlace:

https://github.com/aidavr/TFM-Aida-Vaquero-Rey

Análisis descriptivo de la base de datos

Se inicia el presente trabajo con una base de datos con 822 observaciones (genomas bacterianos) y 73 variables

```
#Se cargan las librerías necesarias

library (stringr)# manipulación rápida de textos
library(dplyr) # manipulación y transformación de datos en data frames
library (tidyr) # transformación y reorganización de datos
library (tidyverse) # manipulación, análisis de datos y visualización de datos
library(readr) #lectura de datos tabuales en distintos formatos

#Se establece directorio de trabajo

setwd("C:/Users/aidav/Desktop/Secuencias_genomas_TFM")

#Se lee el txt que contiene toda la información relativa a los genomas

base_datos <- read_tsv("BVBRC_genome.txt")
```

Se observa cómo es la base de datos (descomentar para correr el script)

```
#head (base_datos) #para observar las 6 primeras filas
#str (base_datos) #para observar la estructura de la base de datos
```

```
# Se depura la base de datos

db_depurada <- base_datos %>%
  separate(`Genome Name`, into = c("V1", "V2","V3","V4","V5"),
           sep = " ", extra = "drop") %>%
  mutate(species = paste(V1, V2, sep = " ")) %>%
  mutate(tmp = paste(V3, V4, sep = " ")) %>%
  mutate(tmp = ifelse(grepl("subsp",tmp),tmp, "")) %>%
  mutate(subspecies = paste(species, tmp, " ")) %>%
  select(-c(V1, V2,V3,V4,V5,tmp))
```

```
#Seleccionar las variables de interés

db_depurada <- db_depurada %>%
  select("Genome ID", "species",
         "subspecies",
         "Checkm Completeness",
```

```
         "Checkm Contamination",
         "Sequencing Platform")
```

Se exploran las variables cualitativas de interés

```
db_depurada %>% distinct(species)# total=18
```

```
# A tibble: 18 × 1
   species
   <chr>
 1 Bifidobacterium breve
 2 Bifidobacterium kashiwanohense
 3 Bifidobacterium scardovii
 4 Bifidobacterium animalis
 5 Bifidobacterium bifidum
 6 Bifidobacterium thermophilum
 7 Bifidobacterium adolescentis
 8 Bifidobacterium longum
 9 Bifidobacterium pseudocatenulatum
10 Enterococcus faecalis
11 Bifidobacterium pseudolongum
12 Bifidobacterium angulatum
13 Bifidobacterium catenulatum
14 Bifidobacterium dentium
15 Bifidobacterium sp.
16 Cutibacterium avidum
17 Bifidobacterium gallicum
18 Bifidobacterium stercoris
```

```
db_depurada %>% distinct(`Sequencing Platform`)
```

```
# A tibble: 43 × 1
   `Sequencing Platform`
   <chr>
 1 <NA>
 2 Ion Torrent
 3 Illumina MiSeq
 4 Illumina
 5 PacBio RSII
 6 Illumina HiSeq
 7 Sanger dideoxy sequencing; 454; Illumina
 8 Illumina Hiseq 2000
 9 454; ABI3730
10 Illumina HiSeq 2500
# ℹ 33 more rows
```

```
# Se observan las subespecies distintas que aparecen en la BBDD
# (no todos los genomas están caracterizadas a nivel de subespecie)

db_depurada %>% filter(grepl("subsp", subspecies)) %>%
  distinct(subspecies) # total=6
```

```
# A tibble: 6 × 1
  subspecies
  <chr>
1 "Bifidobacterium animalis subsp. lactis  "
2 "Bifidobacterium longum subsp. longum  "
3 "Bifidobacterium animalis subsp. animalis  "
4 "Bifidobacterium longum subsp. infantis  "
5 "Bifidobacterium longum subsp. suis  "
6 "Bifidobacterium catenulatum subsp. kashiwanohense  "
```

Visualización de los datos (tablas y figuras)

```
#Librerías para realizar tablas y figuras

library(gt) # para la creación de tablas

library(ggplot2) # para realizar gráficos

library(gtExtras) # extensión de "gt" para formatear y personalizar tablas
```

Tabla 1. Análisis descriptivo de la integridad y contaminación de CheckM, procedentes del repositorio BV-BRC.

```
tabla_1 <- db_depurada %>% select ("Checkm Completeness","Checkm Contamination")

gt_plt_summary(tabla_1)
```

tabla_1

822 rows x 2 cols

COLUMN	PLOT OVERVIEW	MISSING	MEAN	MEDIAN	SD
Checkm Completeness	92.3 … 100.0	48.8%	99.9	100.0	0.5
Checkm Contamination	0.1 … 10.0	74.5%	1.4	0.9	1.7

Figura 1. Frecuencia absoluta para cada una de las especies recogidas en la base de datos inicial.

- *Creación de la tabla asociada a la Figura 1*

```
tabla_sup_1 <- as.data.frame(table(db_depurada$species))

gt::gt(tabla_sup_1) %>%
  tab_header(title= md("**Tabla Suplementaria 1**"),
             subtitle = "Frecuencia absoluta por especie") %>%
  cols_label(Var1 = "Especie", Freq = "F. absoluta")
```

Tabla Suplementaria 1

Frecuencia absoluta por especie

Especie	F. absoluta
Bifidobacterium adolescentis	70
Bifidobacterium angulatum	3
Bifidobacterium animalis	38
Bifidobacterium bifidum	103
Bifidobacterium breve	100
Bifidobacterium catenulatum	7
Bifidobacterium dentium	23
Bifidobacterium gallicum	2
Bifidobacterium kashiwanohense	2
Bifidobacterium longum	365
Bifidobacterium pseudocatenulatum	70
Bifidobacterium pseudolongum	2
Bifidobacterium scardovii	2
Bifidobacterium sp.	18
Bifidobacterium stercoris	1
Bifidobacterium thermophilum	1
Cutibacterium avidum	1
Enterococcus faecalis	14

- *Modificación de datos para representarlos en la Figura 1*

```
tabla_sup_1 <- tabla_sup_1 %>%
  mutate(Var1 = str_replace_all(Var1, "Bifidobacterium", "B.")) %>%
  mutate(Var1 = str_replace_all(Var1, "Cutibacterium", "C.")) %>%
  mutate(Var1 = str_replace_all(Var1, "Enterococcus", "E."))
```

- *Creación de la Figura 1:*

```
tabla_sup_1 %>% ggplot(aes(x=Var1, y=Freq)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  geom_text(aes(label = Freq), nudge_y = 12, size = 4) +
  theme_light() +
  xlab("") +
  ylab("Frecuencia absoluta por especie")+
  theme(axis.title.x = element_text( size = 15),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 15, face = "italic"))
```

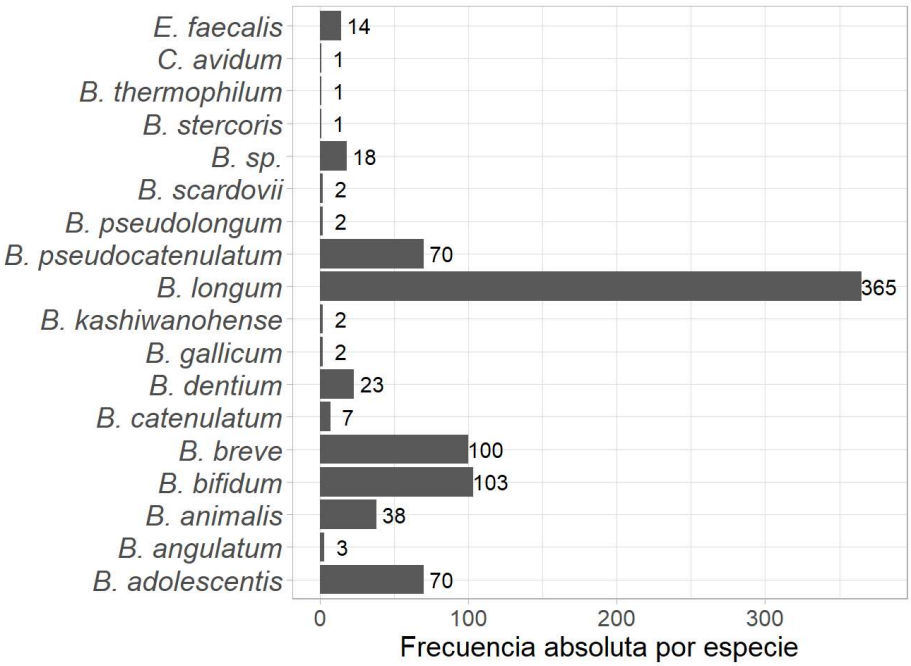


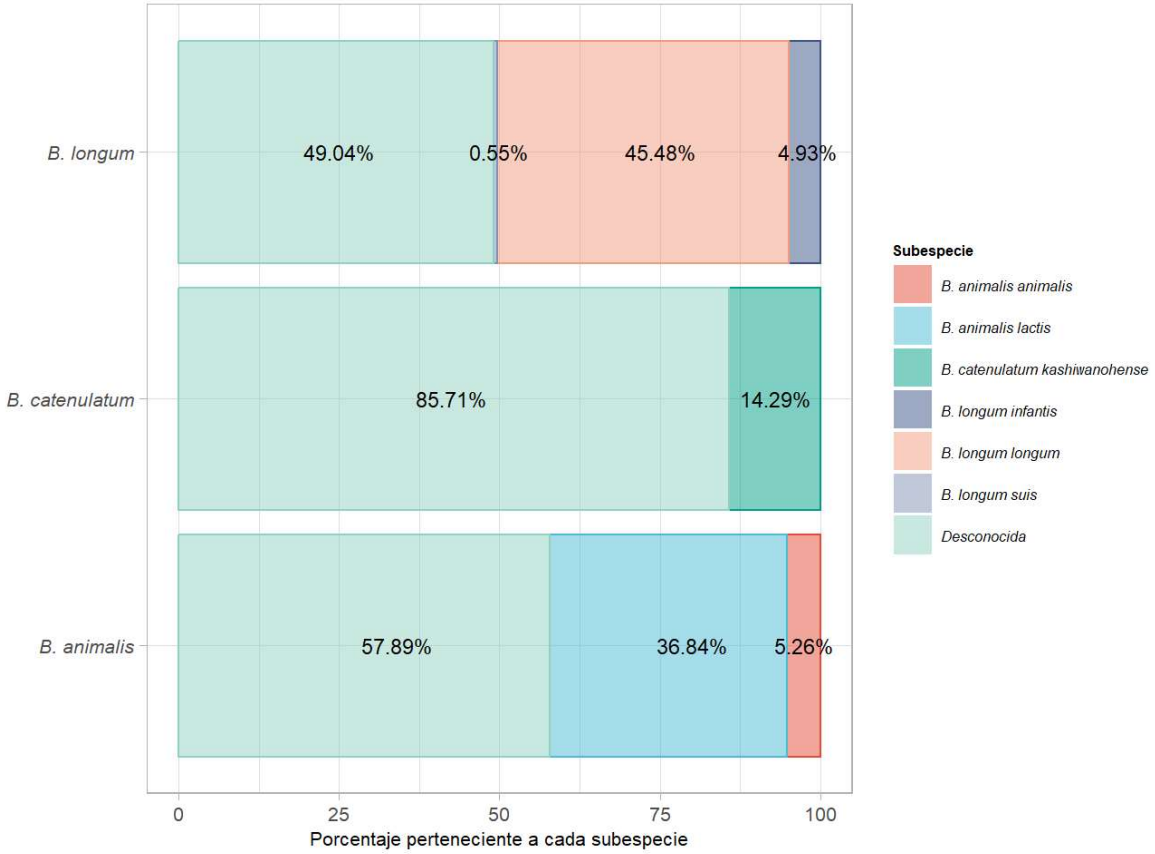
Figura 2. Porcentaje de cada una de las subespecies descritas para B. logum, B. catenulatum y B. animales. También se indica el porcentaje de dichas especies no descritas a nivel de subespecie (Desconocida).

- *Preparación de los datos para la realización de la Figura 2:*

```
# Primero se obtienen los porcentajes/subespecies

db_depurada_perc <- db_depurada %>%
  dplyr::filter(species == "Bifidobacterium animalis" |
                  species == "Bifidobacterium longum" |
                  species == "Bifidobacterium catenulatum") %>%
  dplyr::mutate(subspecies = ifelse(
    grepl("subsp", subspecies), subspecies, "Desconocida")) %>%
  dplyr::group_by(species) %>%
  dplyr::mutate(total = n()) %>%
  ungroup() %>%
  dplyr::group_by(subspecies, species) %>%
  dplyr::mutate(total_subspecies = n()) %>%
  ungroup() %>%
  dplyr::mutate(percent= (total_subspecies/total)*100)
```

- *Elaboración de la Figura 2*

```r
library(ggsci) # paletas de colores adicionales para gráficos en ggplot2

y_pos <- c(97.5, 80, 34, 93, 42.5, 98, 75, 50, 25)

db_depurada_perc %>%
  mutate(species = str_replace_all(species, "Bifidobacterium", "B.")) %>%
  mutate(subspecies = str_replace_all(subspecies, "Bifidobacterium", "B.")) %>%
  mutate(subspecies = str_replace_all(subspecies, "subsp.", "")) %>%
  mutate(subspecies = str_replace_all(subspecies, "  ", " ")) %>%
  select(species, subspecies, total, total_subspecies, percent) %>%
  group_by(species, subspecies) %>%
  mutate(percent = round(percent, 2)) %>%
  distinct(subspecies, .keep_all = TRUE) %>%
  arrange(species, subspecies,  percent) %>%
  ggplot(aes(x= species, y= percent, color = as.factor(subspecies),
             fill= as.factor(subspecies))) +
  geom_bar(stat= "identity", alpha= .5) +
  coord_flip() +
  geom_text(aes(x= species, y= y_pos,
                label = paste(percent, "%", sep="")),
            color = "black", size = 3) +
  scale_color_npg() +
  scale_fill_npg()+
  theme_light()+
  guides(fill=guide_legend(title = "Subespecie"), color="none")+
  ylab("Porcentaje perteneciente a cada subespecie")+
  xlab("")+
  theme(axis.title.x = element_text( size = 8),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8, face = "italic"),
        legend.text = element_text (size = 6, face = "italic"),
        legend.title = element_text (size = 6, face = "bold"))
```



GUNC

Obtención de resultados

Este paso se llevó a cabo en linux, con la herramienta GUNC. El código se aporta en el github creado para este Trabajo de Fin de Máster (README.md)

Tratamiento previo de los outputs obtenidos

1. Se importa la base de datos generada como output de GUNC

```
    setwd("D:/RESULTADOS_PRUEBAS_TFM/resultados_fna_GUNC")
    results <- read.csv("GUNC.progenomes_2.1.maxCSS_level.tsv", sep = "\t")
```

2. Se eliminan los controles

```
results <- results %>%
  filter (genome != c("Escherichia_coli_str_K12_MG1655","Lacto_324831.13"))
```

3. Las bbdd comparten el orden de las observaciones pero no hay ninguna variable compartida en ambas. Por ello, se genera un identificador (“Id”) compartido.

```
db_depurada <- db_depurada %>%
  rownames_to_column("Id")
```

```
results <- results %>%
  rownames_to_column("Id")
```

4. Se compila la bbdd descriptiva con la bbdd que se genera como output de GUNC

```
db_conjunta <- db_depurada %>%
  inner_join(results, by= "Id")
```

5. Se eliminan las especies distintas a *Bifidobacterium* y los *Bifidobacterium* con hospedador no humano antes de proceder con el análisis del output de GUNC

```
#Se obtienen 802 bacterias

db_conjunta <- db_conjunta %>%
      filter (grepl("Bifidobacterium", species)) %>%
      filter (!grepl("scardovii", species)) %>%
      filter (!grepl("gallicum", species)) %>%
      filter (!grepl("thermophilum", species)) %>%
      mutate(species = str_replace_all(
        species,
        "Bifidobacterium kashiwanohense",
        "Bifidobacterium catenulatum")) %>%
      mutate(species = str_replace_all(
        species,"Bifidobacterium stercoris",
        "Bifidobacterium adolescentis"))
```

```
# 10 especies diferentes + Bifidobacterium sp.

db_conjunta %>% distinct(species)
```

```
# A tibble: 11 × 1
   species
   <chr>
 1 Bifidobacterium breve
 2 Bifidobacterium catenulatum
 3 Bifidobacterium animalis
 4 Bifidobacterium bifidum
 5 Bifidobacterium adolescentis
 6 Bifidobacterium longum
 7 Bifidobacterium pseudocatenulatum
 8 Bifidobacterium pseudolongum
 9 Bifidobacterium angulatum
10 Bifidobacterium dentium
11 Bifidobacterium sp.
```

Análisis de los resultados de GUNC

- **Cálculo de los genes codificantes obtenidos de todos los genomas de *Bifidobacterium***

```
# Media, mediana, rango intercuartílico, min y max
summary(db_conjunta$n_genes_called)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1381    1827    1948    1946    2057    2627
```

```
# Desviación estándar
sd(db_conjunta$n_genes_called)
```

```
[1] 184.4924
```

```
# Cálculo del intervalo de confianza (CI 95%)

t <- t.test(db_conjunta$n_genes_called)

CI <- t$conf.int
print(CI)
```

```
[1] 1932.873 1958.449
attr(,"conf.level")
[1] 0.95
```

- **Determinar en qué especies es más común encontrar contaminaciones**

Se escogen sólo aquellas secuencias contaminadas:

```
filter_results_false <- db_conjunta %>%
  filter (pass.GUNC == "False")
```

Frecuencias absolutas y relativas por especies pertenecientes a los genomas contaminados:

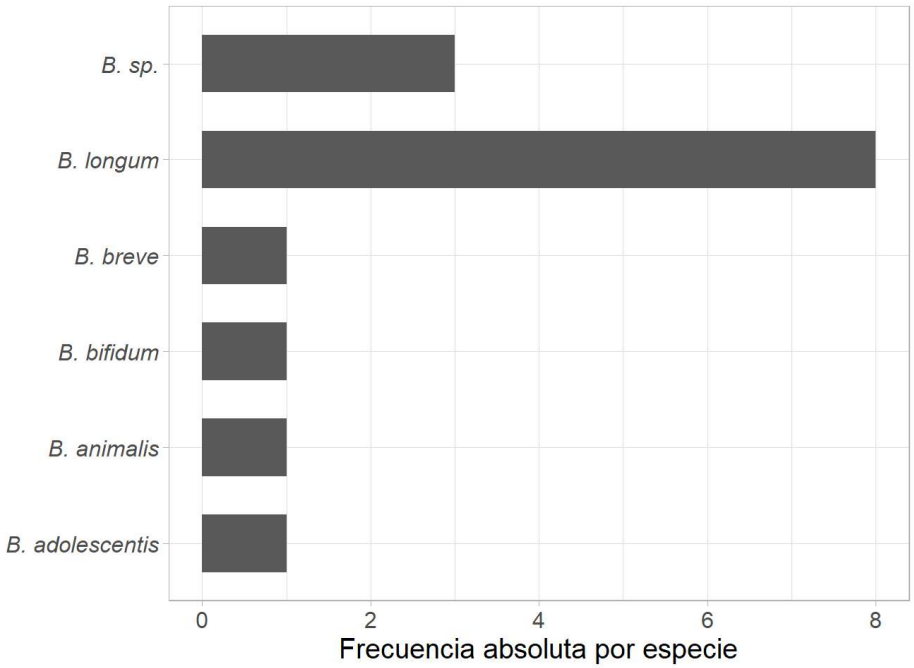
```
t_abs <- table(filter_results_false$species)

t_abs <- as.data.frame(t_abs)

t_rel <- t_abs %>%
  mutate (Freq_rel = (prop.table(t_abs$Freq)*100)) %>%
  mutate(Freq_rel = round(Freq_rel, digits = 2)) %>%
  mutate(Freq_rel = paste(Freq_rel, "%", sep=""))

gt::gt(t_rel) %>%
  tab_header(title= md(""),subtitle = md (
"**Frecuencia absoluta de las especies con genoma contaminado**")) %>%
  cols_label(Var1 = md("**Especie**"),
             Freq = md("**F. absoluta**"),
             Freq_rel =md("**F. relativa**"))
```

Frecuencia absoluta de las especies con genoma contaminado		
Especie	**F. absoluta**	**F. relativa**
Bifidobacterium adolescentis	1	6.67%
Bifidobacterium animalis	1	6.67%

Frecuencia absoluta de las especies con genoma contaminado		
Especie	**F. absoluta**	**F. relativa**
Bifidobacterium bifidum	1	6.67%
Bifidobacterium breve	1	6.67%
Bifidobacterium longum	8	53.33%
Bifidobacterium sp.	3	20%

Figura 3. Frecuencia absoluta para cada una de las especies de *Bifidobacterium* correspondientes a los genomas contaminados.

```
t_abs %>%
  mutate(Var1 = str_replace_all(Var1, "Bifidobacterium", "B.")) %>%
  ggplot(aes(x=Var1, y=Freq)) +
  geom_bar(stat = "identity", width = .6) +
  coord_flip() +
  theme_light() +
  xlab("") +
  ylab("Frecuencia absoluta por especie") +
  theme(axis.text.y = element_text(size= 12, face= "italic"),
        axis.text.x = element_text(size = 12),
        axis.title.x = element_text(size = 15))
```



- **Determinar si existe relación del CSS con el RSS y con T_{eff}**

Tabla 2. Análisis descriptivo del CSS (Clade Separation Score), T_{eff} (Effective number of Surplus Clades) y RSS (Reference Representation Score) para los genomas contaminados (False) y no contaminados (True).

```
## CSS

#Se analiza la normalidad

model1 <- lm(db_conjunta$clade_separation_score ~ db_conjunta$pass.GUNC)
Shapiro1 <- shapiro.test(model1$residuals)
Shapiro1[["p.value"]]
```

```
[1] 4.003971e-49
```

```
# Estadísticos descriptivos

descriptivos1 <- db_conjunta %>%
  group_by(pass.GUNC) %>%
  summarize(
        media = mean(clade_separation_score, na.rm = TRUE),
        mediana = median(clade_separation_score, na.rm = TRUE),
        minimo = min(clade_separation_score, na.rm = TRUE),
        maximo = max(clade_separation_score, na.rm = TRUE),
        n_total = n(),
        valores_perdidos = sum(is.na(clade_separation_score)),
        desviacion_estandar = sd(clade_separation_score, na.rm = TRUE))

print (descriptivos1)
```

```
# A tibble: 2 × 8
  pass.GUNC   media mediana minimo maximo n_total valores_perdidos
  <chr>       <dbl>   <dbl>  <dbl>  <dbl>   <int>            <int>
1 False     0.848      0.96   0.45   1         15                0
2 True      0.00596    0      0      0.44     787                0
# ℹ 1 more variable: desviacion_estandar <dbl>
```

```
## RSS

# Se analiza la normalidad

model2 <- lm(db_conjunta$reference_representation_score ~ db_conjunta$pass.GUNC)
Shapiro2 <- shapiro.test(model2$residuals)
Shapiro2[["p.value"]]
```

```
[1] 4.119734e-19
```

```
# Estadísticos descriptivos

descriptivos2 <- db_conjunta %>%
  group_by(pass.GUNC) %>%
  summarize(
        media = mean(reference_representation_score, na.rm = TRUE),
        mediana = median(reference_representation_score, na.rm = TRUE),
        minimo = min(reference_representation_score, na.rm = TRUE),
        maximo = max(reference_representation_score, na.rm = TRUE),
        n_total = n(),
```

```
        valores_perdidos = sum(is.na(reference_representation_score)),
        desviacion_estandar = sd(reference_representation_score, na.rm = TRUE))

print (descriptivos2)
```

```
# A tibble: 2 × 8
  pass.GUNC media mediana minimo maximo n_total valores_perdidos
  <chr>     <dbl>   <dbl>  <dbl>  <dbl>   <int>            <int>
1 False     0.89     0.9    0.82   0.94      15                0
2 True      0.899    0.91   0.75   0.96     787                0
# ℹ 1 more variable: desviacion_estandar <dbl>
```

```
## T eff

# Se analiza la normalidad

model3 <- lm(db_conjunta$n_effective_surplus_clades ~ db_conjunta$pass.GUNC)
Shapiro3 <- shapiro.test(model3$residuals)
Shapiro3[["p.value"]]
```

```
[1] 3.925633e-50
```

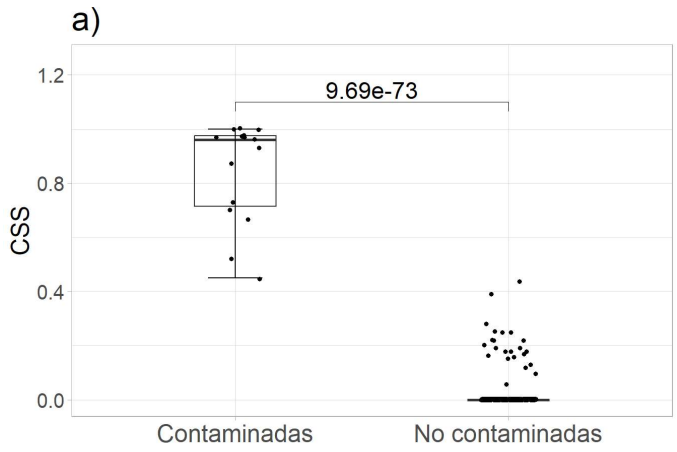
```
# Estadísticos descriptivos

descriptivos3 <- db_conjunta %>%
group_by(pass.GUNC) %>%
  summarize(
        media = mean(n_effective_surplus_clades, na.rm = TRUE),
        mediana = median(n_effective_surplus_clades, na.rm = TRUE),
        minimo = min(n_effective_surplus_clades, na.rm = TRUE),
        maximo = max(n_effective_surplus_clades, na.rm = TRUE),
        n_total = n(),
        valores_perdidos = sum(is.na(n_effective_surplus_clades)),
        desviacion_estandar = sd(n_effective_surplus_clades, na.rm = TRUE))

print (descriptivos3)
```

```
# A tibble: 2 × 8
  pass.GUNC   media mediana minimo maximo n_total valores_perdidos
  <chr>       <dbl>   <dbl>  <dbl>  <dbl>   <int>            <int>
1 False     0.054      0.05   0.04   0.11      15                0
2 True      0.00177    0      0      0.08     787                0
# ℹ 1 more variable: desviacion_estandar <dbl>
```

Figura 4. Representación de la distribución de los datos correspondientes al CSS, el Teff y la RSS en función de si los genomas están contaminados o no (según el filtro establecido en GUNC). Se realizó una prueba estadística de U de Mann-Whitney: a) p-valor = 9.69*10e-73, b) p-valor = 8.21*10e-68, c) p-valor = 0.293.

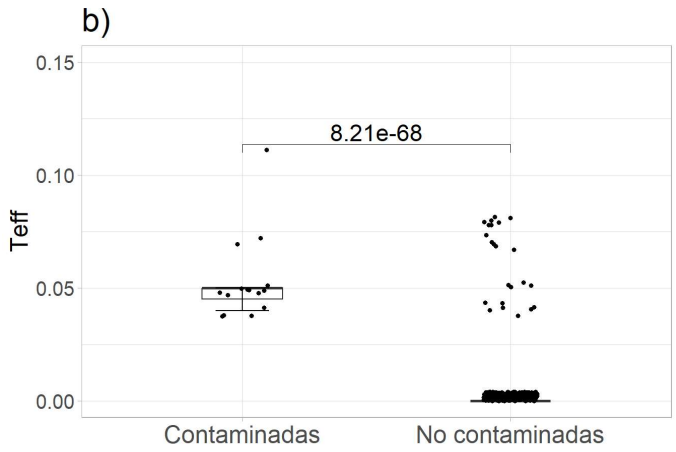
```
library(rstatix) # para realizar análisis estadísticos
library(ggpubr) # para la crear y personalizar gráficos de alta calidad
```

```
# Figura 4a)

stat1 <- rstatix::wilcox_test(
  data= db_conjunta, clade_separation_score ~ pass.GUNC)
stat1 <- stat1 %>% rstatix::add_xy_position(x= "pass.GUNC", dodge = .8 )
stat1[1,8] <- 1.1

db_conjunta %>%
  ggplot(aes(x= pass.GUNC, y= clade_separation_score)) +
  geom_boxplot(alpha= 0, width= .3) +
  geom_point(size= 1.3,
             position = position_jitter(width = .1)) +
  theme_light() +
  stat_boxplot(geom= "errorbar", width = 0.2) +
  stat_pvalue_manual(stat1, label = "p", step.increase = 1, size = 6) +
  ylab("CSS")+
  xlab("")+
  scale_x_discrete(labels= c("Contaminadas", "No contaminadas"))+
  ggtitle("a)")+
  theme(axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 18),
        title = element_text(size= 20))+
  ylim (0, 1.25)
```

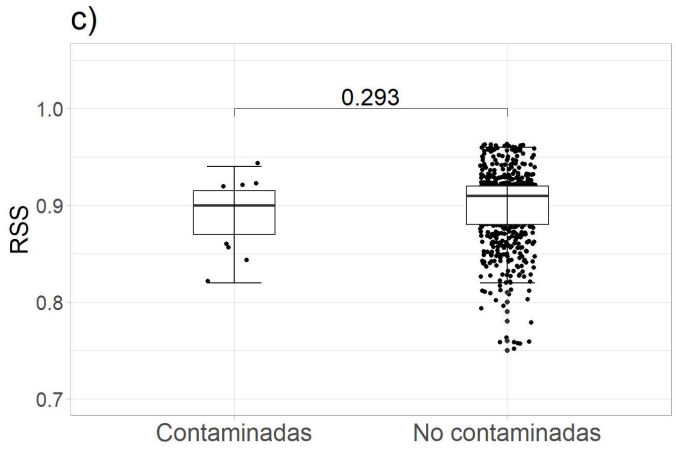



```
# Figura 4b)

stat2 <- rstatix::wilcox_test(
  data= db_conjunta, n_effective_surplus_clades ~ pass.GUNC)

stat2 <- stat2 %>% rstatix::add_xy_position(x= "pass.GUNC", dodge = .8)

db_conjunta %>%
  ggplot(aes(x= pass.GUNC, y= n_effective_surplus_clades)) +
  geom_boxplot(alpha= 0, width= .3) +
  geom_point(size= 1.3,position = position_jitter(width = .1)) +
  theme_light() +
  stat_boxplot(geom= "errorbar", width = 0.2) +
  stat_pvalue_manual(stat2, label = "p", step.increase = 1, size = 6) +
```

```
  ylab("Teff")+
  xlab("")+
  scale_x_discrete(labels= c("Contaminadas", "No contaminadas"))+
  ggtitle("b)")+
  theme(axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 18),
        title = element_text(size= 20))+
  ylim (0, 0.15)
```




```
#Figura 4c)

stat3 <- rstatix::wilcox_test(
  data= db_conjunta, reference_representation_score ~ pass.GUNC)
stat3 <- stat3 %>% rstatix::add_xy_position(x= "pass.GUNC", dodge = .8 )
stat3[1,8] <- 1

db_conjunta %>%
  ggplot(aes(x= pass.GUNC, y= reference_representation_score)) +
  geom_point(size= 1.3,position = position_jitter(width = .1)) +
  geom_boxplot(alpha= 1, width= .3) +
  theme_light() +
  stat_boxplot(geom= "errorbar", width = 0.2) +
  stat_pvalue_manual(stat3, label = "p", step.increase = 1, size = 6) +
  ylab("RSS")+
  xlab("")+
  scale_x_discrete(labels= c("Contaminadas", "No contaminadas"))+
  ggtitle("c)")+
  theme(axis.text.x = element_text(size = 18),
        axis.text.y = element_text(size = 15),
        axis.title.y = element_text(size = 18),
        title = element_text(size= 20))+
  ylim (0.7, 1.05)
```



- **Determinar si existe correlación entre el CSS y el RSS**

```
corr <- cor.test(
  x=db_conjunta$clade_separation_score,
  y=db_conjunta$reference_representation_score, method = 'spearman')

corr
```

```
    Spearman's rank correlation rho

data:  db_conjunta$clade_separation_score and db_conjunta$reference_representation_score
S = 99556732, p-value = 6.954e-06
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.1579757
```

- **Determinar si existe correlación entre el CSS y el T_{eff}**

```
corr <- cor.test(
  x=db_conjunta$clade_separation_score,
  y=db_conjunta$n_effective_surplus_clades, method = 'spearman')

corr
```

```
    Spearman's rank correlation rho

data:  db_conjunta$clade_separation_score and db_conjunta$n_effective_surplus_clades
S = 85033, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.999011
```

- **Determinar si existe sesgo de contaminación en función de la tecnología de secuenciación**

Para visualizar los distintos tipos de secuenciadores y depurarlo posteriormente

```
# Descomentar para correr script

#table(db_conjunta$`Sequencing Platform`)

db_conjunta %>% distinct(`Sequencing Platform`)
```

```
# A tibble: 42 × 1
   `Sequencing Platform`
   <chr>
 1 <NA>
 2 Ion Torrent
 3 Illumina
 4 PacBio RSII
 5 Illumina HiSeq
 6 Illumina Hiseq 2000
 7 454; ABI3730
 8 Illumina MiSeq
 9 Roche 454 titanium
10 Roche 454 titanium; Illumina Hiseq/2000
# ℹ 32 more rows
```

Depurar para reducir el número de categorías para la variable "Plataforma de secuenciación"

```
db_conjunta <- db_conjunta %>%
  mutate(`Sequencing Platform` = ifelse
         (grepl
           (",|;|/",`Sequencing Platform`),"Varios", `Sequencing Platform`))%>%
  mutate(`Sequencing Platform` = ifelse
         (grepl
           ("Pac",`Sequencing Platform`),"PacBio", `Sequencing Platform`)) %>%
  mutate(`Sequencing Platform` = ifelse
         (grepl
           ("Ion",`Sequencing Platform`),"Ion Torrent",`Sequencing Platform`)) %>%
  mutate(`Sequencing Platform` = ifelse
         (grepl("454", `Sequencing Platform` ), "454", `Sequencing Platform`)) %>%
  mutate(`Sequencing Platform` = ifelse
         (grepl
           ("NextSeq 500",`Sequencing Platform`),
           "Illumina",`Sequencing Platform`))%>%
  mutate(`Sequencing Platform` = ifelse
         (grepl
           ("Hi",`Sequencing Platform`),
           "Illumina HiSeq",`Sequencing Platform`)) %>%
  mutate(`Sequencing Platform` = ifelse
         (grepl
           ("Mi",`Sequencing Platform`),
           "Illumina MiSeq",`Sequencing Platform`))%>%
  mutate(`Sequencing Platform`= ifelse
         (grepl
           ("Illumina",`Sequencing Platform` ),"Illumina",`Sequencing Platform`))

table(db_conjunta$`Sequencing Platform`)
```

```
                  454                Illumina             Ion Torrent
                   18                     611                      29
Oxford Nanopore GridION                PacBio                  Sanger
                    1                      70                       2
                Varios
                   19
```

Eliminar los "NA" de la variable Plataforma de Secuenciación

```
sum(is.na(db_conjunta$`Sequencing Platform`))
```

```
[1] 52
```

```
db_conjunta_sin_na <- db_conjunta %>%
  drop_na(`Sequencing Platform`)
```

Tabla 3. Tabla de contingencia de las distintas plataformas de secuencias en función de la contaminación de los genomas.

```
table(db_conjunta_sin_na$`Sequencing Platform`, db_conjunta_sin_na$pass.GUNC)
```

```
                       False True
  454                      0   18
  Illumina                12  599
  Ion Torrent              0   29
  Oxford Nanopore GridION  0    1
  PacBio                   0   70
  Sanger                   0    2
  Varios                   2   17
```

CSS en función de la plataforma de secuenciación (todos los genomas: contaminados y no contaminados)

```
db_conjunta_sin_na <- db_conjunta_sin_na %>%
  rename(Platform = `Sequencing Platform`)

# Se analiza la normalidad

model <- lm(
  db_conjunta_sin_na$clade_separation_score ~ db_conjunta_sin_na$Platform)

Shapiro <- shapiro.test(model$residuals)
Shapiro[["p.value"]]
```

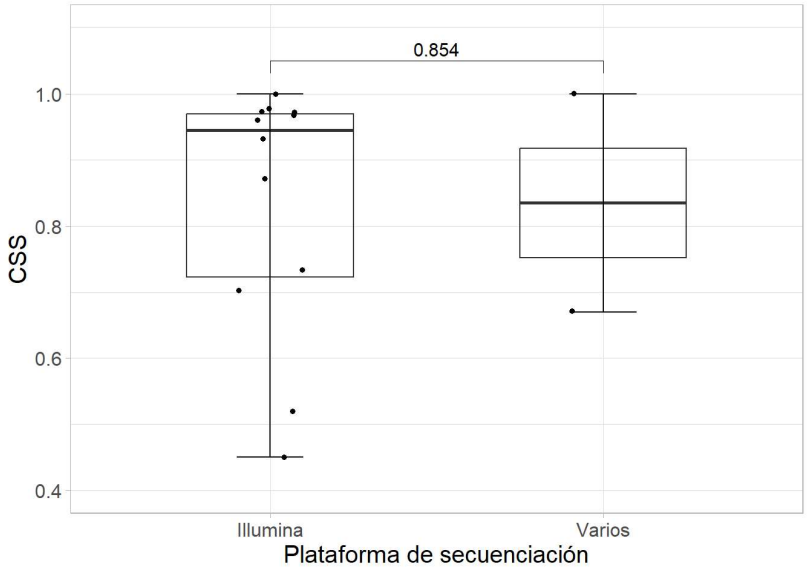
```
[1] 1.932959e-48
```

```
# Prueba no paramétrica de Kruskal-Wallis (7 categorías)

kruskal.test(
  formula= db_conjunta_sin_na$clade_separation_score ~ db_conjunta_sin_na$Platform)
```

```
    Kruskal-Wallis rank sum test

data:  db_conjunta_sin_na$clade_separation_score by db_conjunta_sin_na$Platform
Kruskal-Wallis chi-squared = 7.6655, df = 6, p-value = 0.2636
```

Figura 5. Representación de la distribución de los datos de CSS pertenecientes a los genomas contaminados, en función de la plataforma de secuenciación. Se realizó una prueba estadística de U de Mann-Whitney (p-valor = 0.854).

```
df <- db_conjunta_sin_na %>%
  filter(pass.GUNC == "False") %>%
  mutate(Platform = as.factor(Platform))


# Prueba no paramétrica U de Mann-Witney (2 categorías: Illumina y Varios)

stat4 <- rstatix::wilcox_test(data = df, clade_separation_score ~ Platform)
stat4 <- stat4 %>% rstatix::add_xy_position(x= "Platform", dodge = .8 )
stat4[1,8] <- 1.05

db_conjunta_sin_na %>%
  filter(pass.GUNC == "False") %>%
  ggplot(aes(x= Platform, y= clade_separation_score)) +
  geom_boxplot(alpha= 0, width= .5) +
  geom_point(size= 1.3,position = position_jitter(width = .1)) +
  theme_light() +
  stat_boxplot(geom= "errorbar", width = 0.2) +
  stat_pvalue_manual(stat4, label = "p", step.increase = 1, size= 4) +
  ylab("CSS")+
  xlab("Plataforma de secuenciación")+
  theme(axis.text.x = element_text(size = 12),
        axis.title.x = element_text(size = 15),
        axis.text.y = element_text( size = 12),
        axis.title.y = element_text(size = 15))+
  ylim(0.4, 1.1)
```



PRODIGAL

Preparación del input

En primer lugar, se deben eliminar los controles y las muestras contaminadas (según los resultados obtenidos en GUNC) de los archivos FASTA originales con extensión "fna".

Para ello, se establece el directorio donde se encuentran estos archivos:

```
setwd("D:/RESULTADOS_PRUEBAS_TFM/genome_fna_passGUNC")
```

Se crea una variable con el mismo nombre que los ficheros originales (añadiendo la extensión .fna)

```
muestras <- results %>%
  mutate (muestras = paste(genome, ".fna", sep = ""))
```

Se filtran controles y muestras contaminadas:

```
controles <- muestras %>%
  select (muestras) %>%
  filter (muestras == c("Escherichia_coli_str_K12_MG1655.fna","Lacto_324831.13.fna"))

muestras_contaminadas <- muestras %>%
  filter (pass.GUNC == "False") %>%
  select (muestras)
```

Se eliminan los archivos originales (descomentar bucles para ejecutarlos):

```
vector_muestras <- as.vector(muestras_contaminadas)
vector_controles <- as.vector (controles)

#for(i in vector_muestras){file.remove(i)}

#for(i in vector_controles){file.remove(i)}
```

Se lleva a cabo el mismo proceso con especies que no pertenecen a *Bifidobacterium* y especies de *Bifidobacterium* no hospedadoras del humano (descomentar bucle para ejecutarlo)

```
db_conjunta_fna <- db_conjunta %>%
  mutate (genome.fna = paste(genome, ".fna", sep = ""))

filtrado_especies <- db_conjunta_fna %>%
  select(species, genome.fna)

filtrado_especies %>% distinct(species)
```

```
# A tibble: 11 × 1
   species
   <chr>
 1 Bifidobacterium breve
 2 Bifidobacterium catenulatum
```

```
 3 Bifidobacterium animalis
 4 Bifidobacterium bifidum
 5 Bifidobacterium adolescentis
 6 Bifidobacterium longum
 7 Bifidobacterium pseudocatenulatum
 8 Bifidobacterium pseudolongum
 9 Bifidobacterium angulatum
10 Bifidobacterium dentium
11 Bifidobacterium sp.
```

```
filtrado_especies <- filtrado_especies %>%
  filter (species =="Enterococcus faecalis"|
            species =="Cutibacterium avidum"|
            species =="Bifidobacterium scardovii"|
            species =="Bifidobacterium gallicum"|
            species =="Bifidobacterium thermophilum")

filtrado_especies <- as.vector (filtrado_especies$genome.fna)

# Descomentar bucle para ejecutarlo

#for(i in filtrado_especies){file.remove(i)}
```

Obtención de resultados

Este paso se llevó a cabo en linux, con la herramienta PRODIGAL. El código se aporta en el README del github creado para este Trabajo de Fin de Máster.

Los outputs generados en PRODIGAL (archivos con extensión .faa) son el input de CD-HIT.

CD-HIT

Obtención de resultados

Este paso se llevó a cabo en linux, con la herramienta CD-HIT. El código se aporta en el github creado para este Trabajo de Fin de Máster (README.md)

Tratamiento de datos

Se realiza un bucle para poder trabajar con el output generado por CD-HIT (archivo con extensión ".clstr" donde se encuentran definidos los clusters). Hay un archivo clstr por cada porcentaje de identidad de secuencia que se probó en CD-HIT (40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90 y 95).

Debido a que el bucle es computacionalmente muy costoso y tarda mucho tiempo, se han almacenado todos los data frames generados para cada una de las identidades en el github creado para este Trabajo de Fin de Máster (Carpeta "Umbrales_CD-HIT"). Se indica cómo se deben importar a continuación.

Para ejecutar el bucle es necesario descomentarlo.

```
#Se establece el directorio de trabajo donde se encuentran los archivos

setwd("D:/RESULTADOS_DEFINITIVOS_TFM/CD-HIT/output_CD-HIT")
```

```
# Se realiza el bucle

identidades <-  c("db_40.clstr", "db_45.clstr",
                  "db_50.clstr", "db_55.clstr",
                  "db_60.clstr", "db_65.clstr",
                  "db_70.clstr", "db_75.clstr",
                  "db_80.clstr", "db_85.clstr",
                  "db_90.clstr", "db_95.clstr")

#resultados <- list()
#for (i in identidades) {
#  df <- read.csv(i, sep="\t", row.names= NULL, header= FALSE, stringsAsFactors = FALSE)
#  n = nrow(df)
#  name <- tools::file_path_sans_ext(basename(i)) # Eliminar extensión .clstr
#  x = 0
#  print(name)
#  numbers_only <- function(x) !grepl("\\D", x)
#  for (row in (1:nrow(df))) {
#    if (numbers_only(df[row,1]) == TRUE) {
#      df[row,1] <- x}
#    else {NULL}
#    x <- df[row,1]
#    print(row)
#  }
#  df <- df %>%
#    mutate(V1= str_replace_all(V1, ">", "")) %>%
#    drop_na() %>%
#    group_by(V1) %>%
#    mutate(n_secuencias = n()) %>%
#    distinct(V1, .keep_all =TRUE) %>%
#    ungroup() %>%
#    mutate (n_grupos=row_number())
#  resultados[[name]] <- df
#}

#n_clstr40 <- resultados$db_40
#n_clstr45 <- resultados$db_45
#n_clstr50 <- resultados$db_50
#n_clstr55 <- resultados$db_55
#n_clstr60 <- resultados$db_60
#n_clstr65 <- resultados$db_65
#n_clstr70 <- resultados$db_70
#n_clstr75 <- resultados$db_75
#n_clstr80 <- resultados$db_80
#n_clstr85 <- resultados$db_85
#n_clstr90 <- resultados$db_90
#n_clstr95 <- resultados$db_95

#Se guardan las tablas generadas

#write.table(n_clstr40, "n_clstr40.txt")
#write.table(n_clstr45, "n_clstr45.txt")
#write.table(n_clstr50, "n_clstr50.txt")
```

```
#write.table(n_clstr55, "n_clstr55.txt")
#write.table(n_clstr60, "n_clstr60.txt")
#write.table(n_clstr65, "n_clstr65.txt")
#write.table(n_clstr70, "n_clstr70.txt")
#write.table(n_clstr75, "n_clstr75.txt")
#write.table(n_clstr80, "n_clstr80.txt")
#write.table(n_clstr85, "n_clstr85.txt")
#write.table(n_clstr90, "n_clstr90.txt")
#write.table(n_clstr95, "n_clstr95.txt")
```

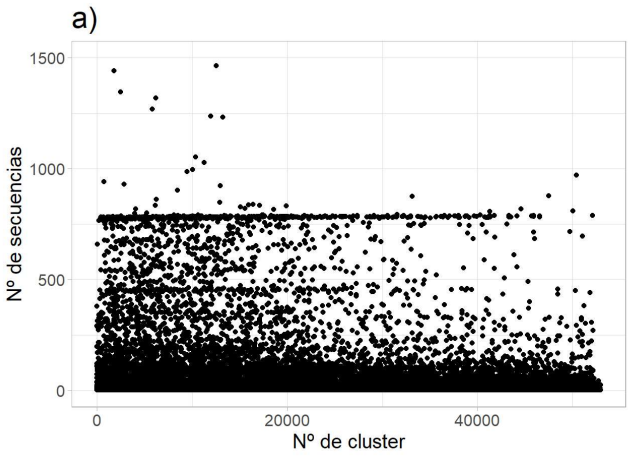
Para importar todos los data frames generados de la ejecución del bucle anterior:

```
setwd("D:/RESULTADOS_DEFINITIVOS_TFM/CD-HIT")

n_clstr40 <- read.csv("n_clstr40.txt", sep="")%>% select(-(V2))
n_clstr45 <- read.csv("n_clstr45.txt", sep="")%>% select(-(V2))
n_clstr50 <- read.csv("n_clstr50.txt", sep="")%>% select(-(V2))
n_clstr55 <- read.csv("n_clstr55.txt", sep="")%>% select(-(V2))
n_clstr60 <- read.csv("n_clstr60.txt", sep="")%>% select(-(V2))
n_clstr65 <- read.csv("n_clstr65.txt", sep="")%>% select(-(V2))
n_clstr70 <- read.csv("n_clstr70.txt", sep="")%>% select(-(V2))
n_clstr75 <- read.csv("n_clstr75.txt", sep="")%>% select(-(V2))
n_clstr80 <- read.csv("n_clstr80.txt", sep="")%>% select(-(V2))
n_clstr85 <- read.csv("n_clstr85.txt", sep="")%>% select(-(V2))
n_clstr90 <- read.csv("n_clstr90.txt", sep="")%>% select(-(V2))
n_clstr95 <- read.csv("n_clstr95.txt", sep="")%>% select(-(V2))
```

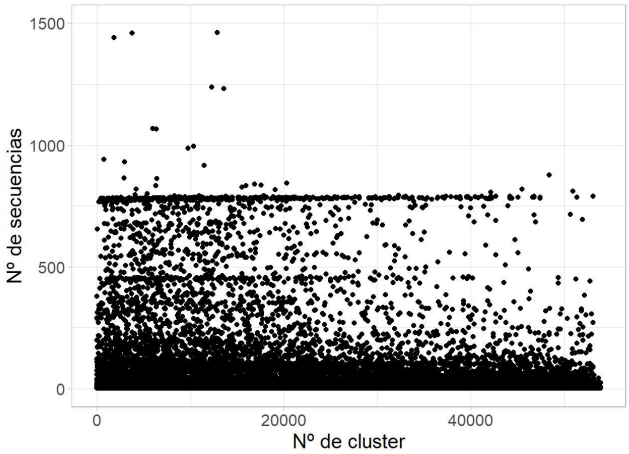
Visualización de resultados

Figura 6. Representación del número de secuencias (eje y) en función del número de cluster generados por CD-HIT (eje x) para distintos umbrales de identidad de secuencia.

```
n_clstr40%>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  ggtitle("a)")+
  theme(title = element_text(size = 18))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

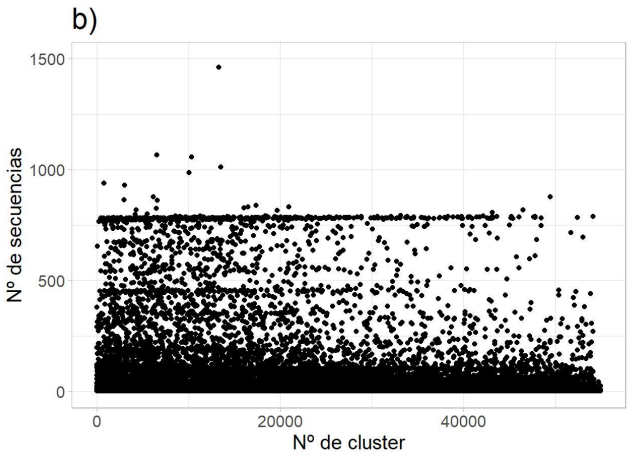



```
n_clstr45 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

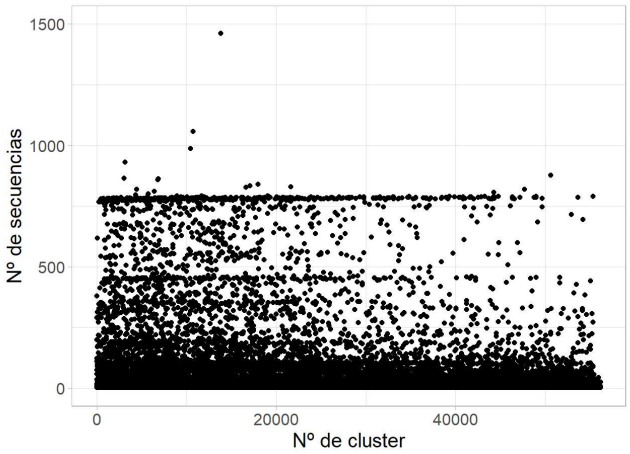



```
n_clstr50 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  ggtitle("b)")+
```

```
  theme(title = element_text(size = 18))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

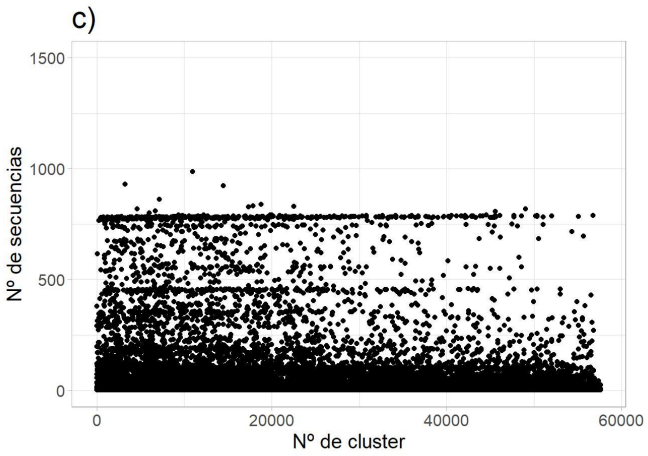



```
n_clstr55 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

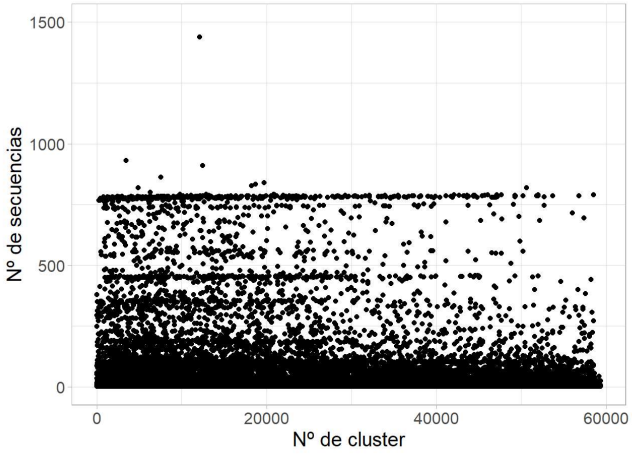



```
n_clstr60 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
```

```
  theme(axis.text = element_text(size = 12))+
  ggtitle("c)")+
  theme(title = element_text(size = 18))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

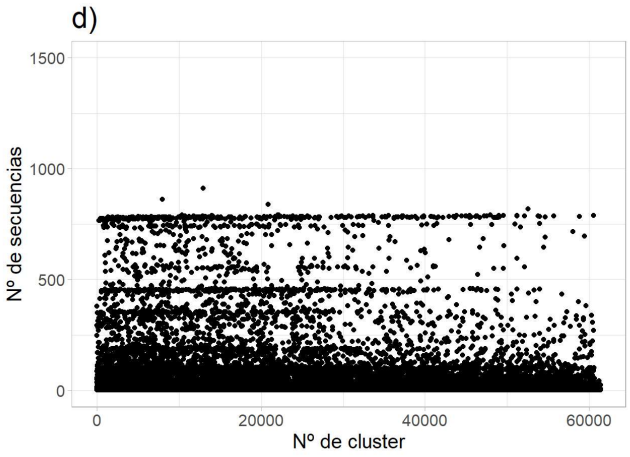



```
n_clstr65 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

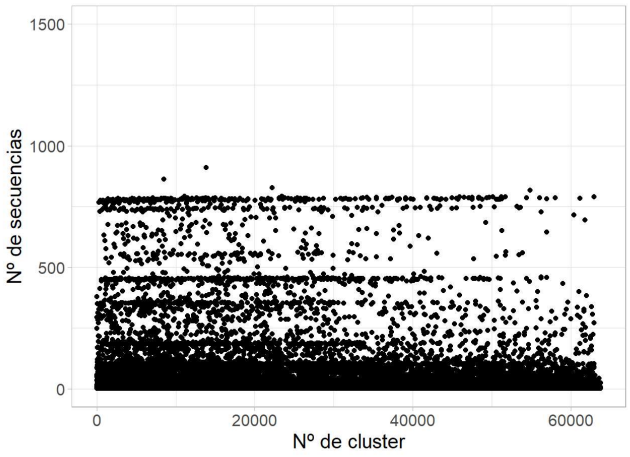



```
n_clstr70 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
```

```
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  ggtitle("d)")+
  theme(title = element_text(size = 18))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

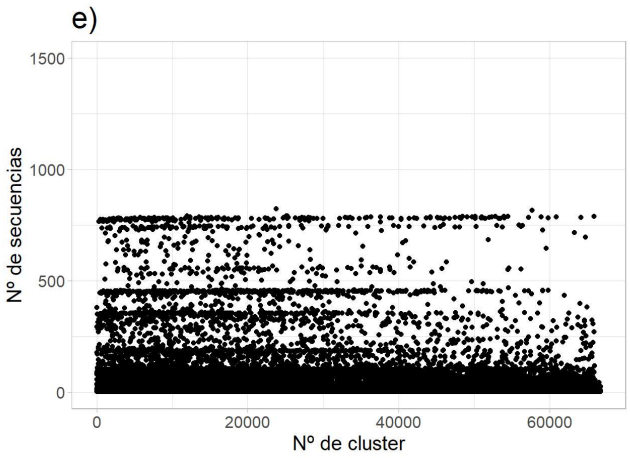



```
n_clstr75 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

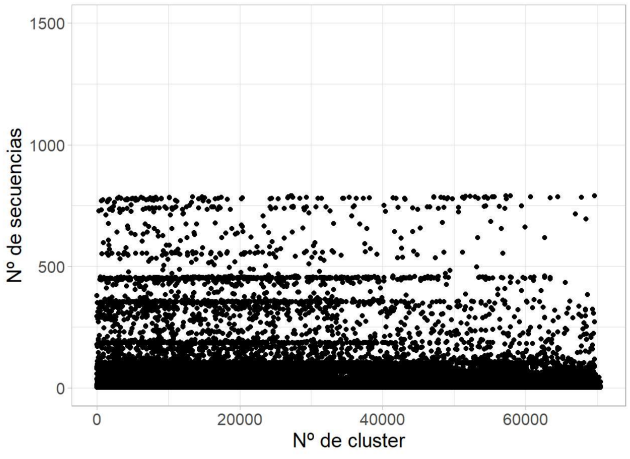



```
n_clstr80 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
```

```
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  ggtitle("e)")+
  theme(title = element_text(size = 18))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

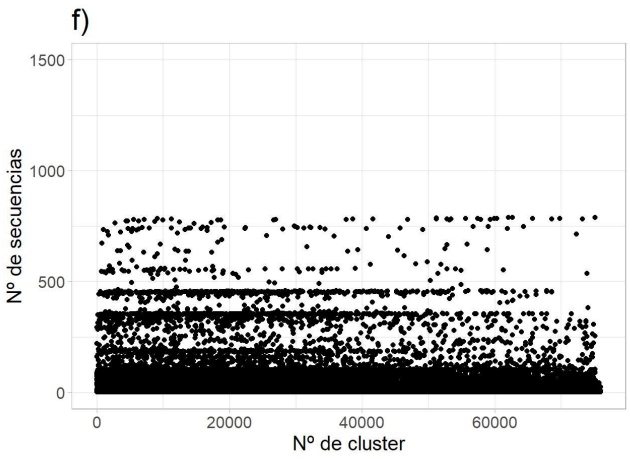



```
n_clstr85 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```

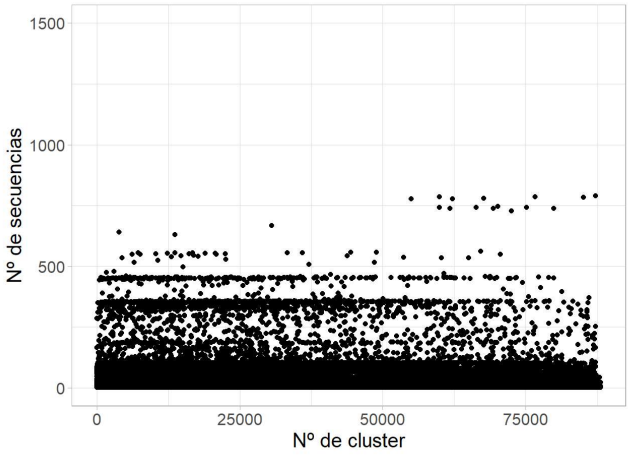



```
n_clstr90 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
```

```
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  ggtitle("f)")+
  theme(title = element_text(size = 18))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```




```
n_clstr95 %>%
  ggplot(aes(x=n_grupos, y= n_secuencias))+
  xlab("Nº de cluster")+
  ylab("Nº de secuencias")+
  geom_point()+
  theme_light()+
  ylim(0,1500)+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.margin = margin(0.5, 1, 0.5, 0.5, unit = "cm"))
```



Selección de los clusters que conforman el genoma codificante central

- El número 787 hace referencia al número total de secuencias con las que se ha trabajado.
- Se multiplica por 0.95 para obtener los clusters que contengan un porcentaje de secuencias igual o superior al 95% de secuencias totales. No se escoge el 100% para no descartar la ausencia de genes en ciertos genomas que no estén completos, dando así un porcentaje de error del 5%.

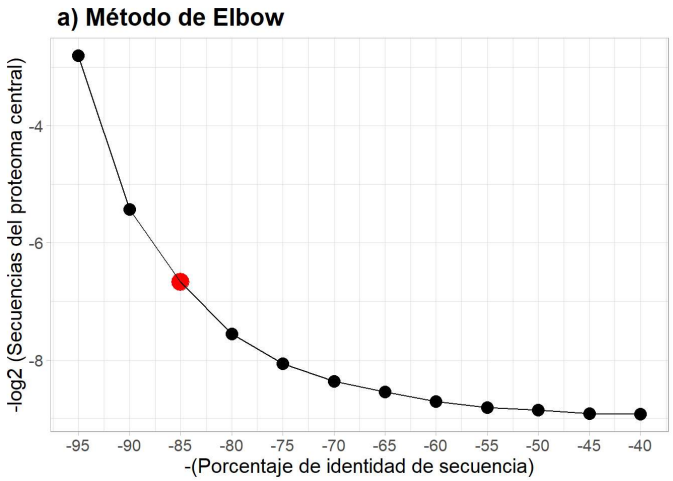
```
sub_clstr40<-subset(n_clstr40, n_secuencias >= 787*0.95)
sub_clstr45<-subset(n_clstr45, n_secuencias >= 787*0.95)
sub_clstr50<-subset(n_clstr50, n_secuencias >= 787*0.95)
sub_clstr55<-subset(n_clstr55, n_secuencias >= 787*0.95)
sub_clstr60<-subset(n_clstr60, n_secuencias >= 787*0.95)
sub_clstr65<-subset(n_clstr65, n_secuencias >= 787*0.95)
sub_clstr70<-subset(n_clstr70, n_secuencias >= 787*0.95)
sub_clstr75<-subset(n_clstr75, n_secuencias >= 787*0.95)
sub_clstr80<-subset(n_clstr80, n_secuencias >= 787*0.95)
sub_clstr85<-subset(n_clstr85, n_secuencias >= 787*0.95)
sub_clstr90<-subset(n_clstr90, n_secuencias >= 787*0.95)
sub_clstr95<-subset(n_clstr95, n_secuencias >= 787*0.95)
```

Método de Elbow

Para seleccionar el umbral de identidad óptimo para el análisis del genoma codificante central

```
df<-as.data.frame(matrix(, ncol=3, nrow=12,
                         dimnames=list(c(), c("Identity","Core","Clusters"))))

df[,1]<-c(95,90,85,80,75,70,65,60,55,50,45,40)

df[,2]<-c(nrow(sub_clstr95),nrow(sub_clstr90),
          nrow(sub_clstr85),nrow(sub_clstr80),
          nrow(sub_clstr75), nrow(sub_clstr70),
          nrow(sub_clstr65), nrow(sub_clstr60),
          nrow(sub_clstr55),nrow(sub_clstr50),
          nrow(sub_clstr45), nrow(sub_clstr40))

df[,3]<-c(nrow(n_clstr95),nrow(n_clstr90),
          nrow(n_clstr85),nrow(n_clstr80),
          nrow(n_clstr75),nrow(n_clstr70),
          nrow(n_clstr65),nrow(n_clstr60),
          nrow(n_clstr55),nrow(n_clstr50),
          nrow(n_clstr45), nrow(n_clstr40))

#Figura 7.a.

df %>%
  ggplot(aes(x= (0-df$Identity), y= -log2(df$Core)))+
  xlab("-(Porcentaje de identidad de secuencia)")+
  ylab("-log2 (Secuencias del proteoma central)")+
  geom_point() +
  scale_x_continuous(breaks = 0-df$Identity)+
  geom_point(aes(color = ifelse(
    (0 - df$Identity) == -85, "PuntoRojo", "Otros"),
                 size = ifelse(
```

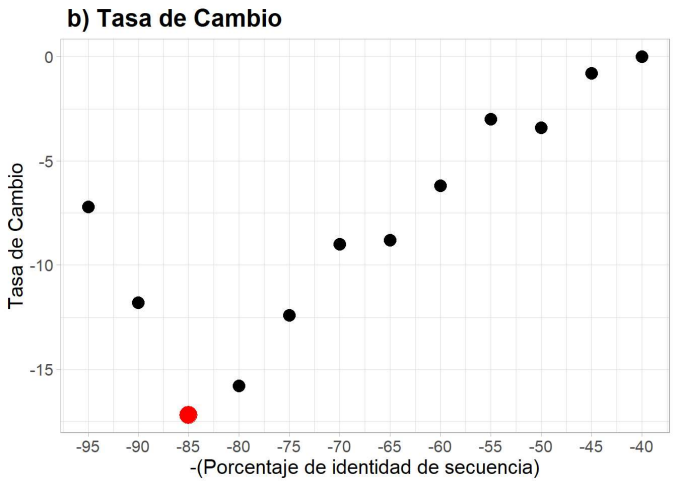
```
                  (0 - df$Identity) == -85, "PuntoGrande", "Otros")))+
  scale_color_manual(values = c("PuntoRojo" = "red", "Otros" = "black")) +
  scale_size_manual(values = c("PuntoGrande" = 6, "Otros" = 4)) +
  guides(color = FALSE, size = FALSE)+
  theme_light()+
  geom_line()+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
  theme(axis.text = element_text(size = 12))+
  ggtitle(" a) Método de Elbow")+
  theme (plot.title = element_text(size=18, face="bold"))
```



Método de Tasa de Cambio

```
tasa_cambio <- diff(df$Core) / diff(df$Identity)

tasa_cambio <- c(tasa_cambio, 0)

df <- df %>%
  mutate(tasa_cambio= tasa_cambio) %>%
  mutate (identidad_invertida = (Identity*(-1)))

#Figura 7.b.

df %>%
  ggplot(aes(x = identidad_invertida, y = tasa_cambio)) +
  geom_point() +
  xlab("-(Porcentaje de identidad de secuencia)") +
  ylab("Tasa de Cambio") +
  scale_x_continuous(breaks = df$identidad_invertida)+
  geom_point(aes(color = ifelse(identidad_invertida == -85, "PuntoRojo", "Otros"),
                 size = ifelse(
                   identidad_invertida == -85, "PuntoGrande", "Otros")))+
  scale_color_manual(values = c("PuntoRojo" = "red", "Otros" = "black")) +
  scale_size_manual(values = c("PuntoGrande" = 6, "Otros" = 4)) +
  guides(color = FALSE, size = FALSE)+
  theme_light()+
  theme(axis.title.x= element_text(size=15))+
  theme(axis.title.y= element_text(size=15))+
```

```
  theme(axis.text = element_text(size = 12))+
  ggtitle(" b) Tasa de Cambio")+
  theme (plot.title = element_text(size=18, face="bold"))
```



Se comprueba de forma matemática:

```
# Calcular la segunda derivada (cambio en la tasa de cambio)
segunda_derivada <- diff(diff(tasa_cambio)) / diff(df$Identity[-1])

# Imprimir la segunda derivada
print(segunda_derivada)
```

```
 [1]  0.16 -1.36 -0.40  0.00  0.64 -0.48 -0.12  0.72 -0.60  0.36
```

```
# Encontrar el índice del mínimo local en la segunda derivada
indice_minimo_local <- which.min(segunda_derivada)
print(indice_minimo_local)
```

```
[1] 2
```

```
# El punto de inflexión corresponde a la variable Identity
punto_inflexion <- c(df$Identity[indice_minimo_local + 1])

# Imprimir el punto de inflexión
print(punto_inflexion)
```

```
[1] 85
```

Extracción de los clusters con el número de secuencias de interés

Este paso se llevó a cabo en linux, con el algoritmo make_multi_seq de CD-HIT. El código se aporta en el github creado para este Trabajo de Fin de Máster (README.md).

MUSCLE y HMMER3

Posteriormente se realizaron alineamientos múltiples de secuencias, así como generación de secuencias consenso con las herramienttas MUSCLE y HMMER3. Para más información consultar el README del

github creado para este Trabajo de Fin de Máster.

EggNOG

Obtención de resultados

Este paso se llevó a cabo en linux, con el servidor-web eggNOG-mapper (consultar cómo debe introducirse el input en el README del github creado para este Trabajo de Fin de Máster).

Proteoma expandido

Análisis de los resultados para el umbral de identidad 50%. Se lee el output de anotación de proteínas generado por EggNOG-mapper y se seleccionan las variables de interés.

```
setwd ("D:/RESULTADOS_DEFINITIVOS_TFM/eggNOG/Identidad_50")

eggNOG_50 <- read.csv("out.emapper.annotations",
                 sep = "\t",
                 comment.char = "#",
                 header = FALSE,
                 na.strings = "-")

colnames(eggNOG_50) <- c("query", "seed_ortholog", "evalue",
                         "score", "eggNOG_OGs",
                         "max_annot_lvl", "COG_category", "Description",
                         "Preferred_name","GOs", "EC", "KEGG_ko",
                         "KEGG_Pathway", "KEGG_Module", "KE",
                         "GG_Reaction", "KEGG_rclass", "BRITE", "CAZy",
                         "BiGG_Reaction", "PFAMs")


select_eggNOG_50 <-eggNOG_50 %>%
  select("query", "COG_category", "KEGG_ko", "CAZy")


#Se explora la variable COG_caterogy
table(select_eggNOG_50$COG_category)
```

```
 C  CE   D   E  EF EGP  EH  ET   F  FG  FP   G  GK   H   I  IM   J   K KLT  KT
25   1   6  61   1   2   2   2  38   1   1  31   2  23  11   1  89  18   2   2
 L   M   O   P   S   T   U   V
24  16  19  10  45  13   7   5
```

```
COG_50 <- select_eggNOG_50 %>%
  separate_rows(COG_category, sep="", convert= F) %>%
  filter(COG_category != "") %>%
  group_by(COG_category) %>%
  mutate(n = n()) %>%
  distinct(n) %>%
```

```
  ungroup() %>%
  mutate (porcentaje = round(n/sum(n)*100, 2))
```

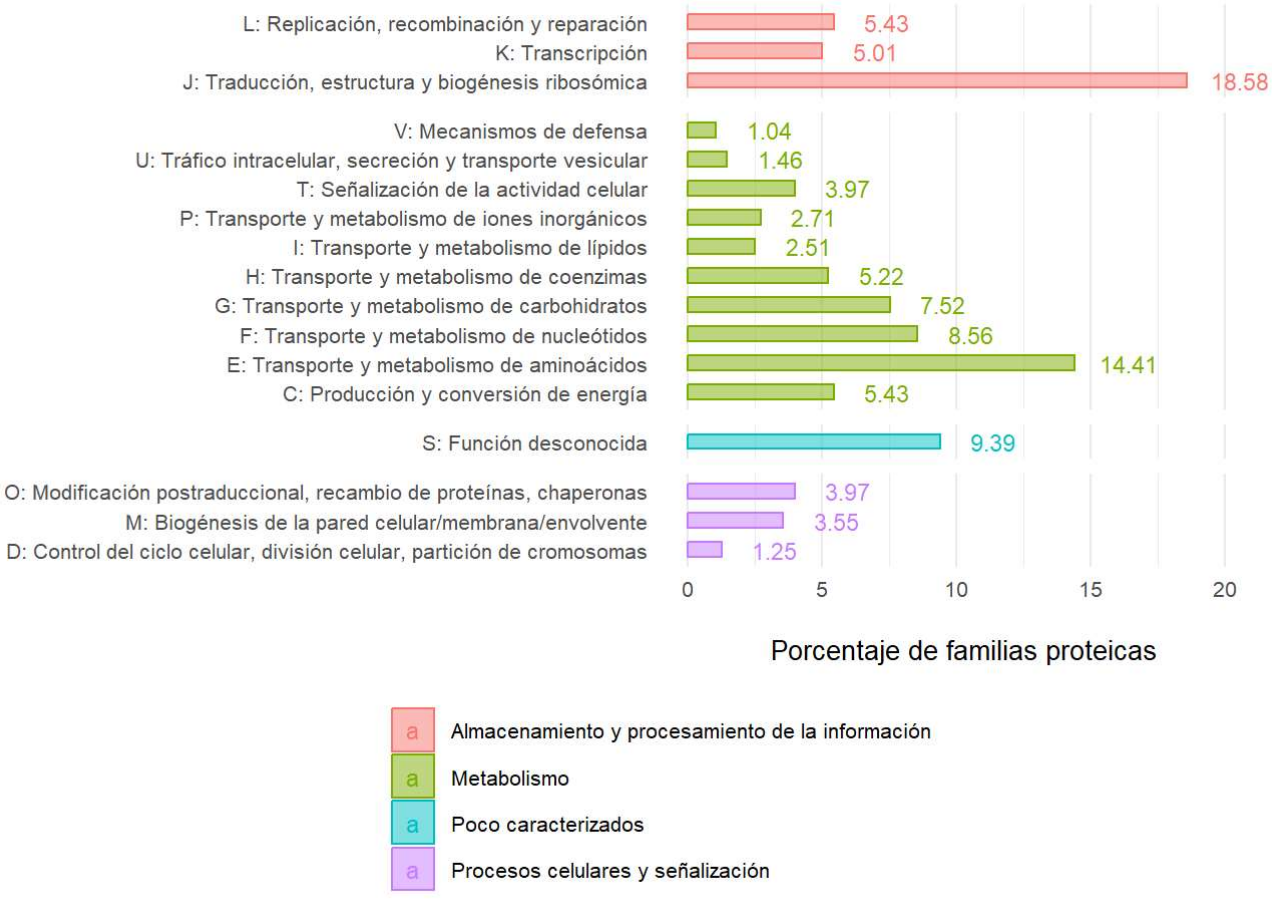
Se crea un data frame con la anotación por categoría

```
annotation <- data.frame(COG_category = factor(
  c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N",
    "O", "P", "Q", "R", "S", "T", "U", "V", "W", "Y", "Z"),
  levels=c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M",
           "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "Y", "Z")),
  labels = c(
    "A: Procesamiento y modificación del ARN",
    "B: Estructura y dinámica de la cromatina",
    "C: Producción y conversión de energía",
    "D: Control del ciclo celular, división celular, partición de cromosomas",
    "E: Transporte y metabolismo de aminoácidos",
    "F: Transporte y metabolismo de nucleótidos",
    "G: Transporte y metabolismo de carbohidratos",
    "H: Transporte y metabolismo de coenzimas",
    "I: Transporte y metabolismo de lípidos",
    "J: Traducción, estructura y biogénesis ribosómica",
    "K: Transcripción", "L: Replicación, recombinación y reparación",
    "M: Biogénesis de la pared celular/membrana/envolvente",
    "N: Motilidad celular",
    "O: Modificación postraduccional, recambio de proteínas, chaperonas",
    "P: Transporte y metabolismo de iones inorgánicos",
    "Q: Biosíntesis, transporte y catabolismo de metabolitos secundarios",
    "R: Predicción de función general únicamente",
    "S: Función desconocida",
    "T: Señalización de la actividad celular",
    "U: Tráfico intracelular, secreción y transporte vesicular",
    "V: Mecanismos de defensa",
    "W: Estructuras extracelulares",
    "Y: Estructura nuclear",
    "Z: Citoesqueleto"))
```

Se unifica con el data frame con el que se está trabajando y se representa gráficamente:

```
#Figura 8 a)

COG_50 %>% inner_join(annotation) %>%
  mutate(
    Leyenda = case_when(
      COG_category == "C" ~ "Metabolismo",
      COG_category == "D" ~ "Procesos celulares y señalización",
      COG_category == "E" ~ "Metabolismo",
      COG_category == "F" ~ "Metabolismo",
      COG_category == "G" ~ "Metabolismo",
      COG_category == "H" ~ "Metabolismo",
      COG_category == "I" ~ "Metabolismo",
      COG_category == "J" ~ "Almacenamiento y procesamiento de la información",
      COG_category == "K" ~ "Almacenamiento y procesamiento de la información",
      COG_category == "L" ~ "Almacenamiento y procesamiento de la información",
      COG_category == "M" ~ "Procesos celulares y señalización",
```

```
    COG_category == "O" ~ "Procesos celulares y señalización",
    COG_category == "P" ~ "Metabolismo",
    COG_category == "Q" ~ "Metabolismo",
    COG_category == "S" ~ "Poco caracterizados",
    COG_category == "T" ~ "Metabolismo",
    COG_category == "U" ~ "Metabolismo",
    COG_category == "V" ~ "Metabolismo")) %>%
ggplot(aes(x=labels, y= porcentaje, color= Leyenda, fill = Leyenda)) +
geom_bar(stat = "identity", alpha= .5, width = 0.5) +
coord_flip() +
geom_text(aes(label = porcentaje), nudge_y = 2, size= 3) +
facet_grid(rows = vars(Leyenda),
           scales = "free_y", switch = "y", space = "free_y") +
theme_minimal(base_family = "Roboto Condensed") +
theme(plot.margin = margin(0.5, 0.5, 3, 0.5, unit = "cm"),
      strip.text.y = element_text(angle = 270, face = "bold", size = 6),
      strip.placement = "outside",
      axis.title.x = element_text(
        margin = margin(t = 0.5, b = 0.5, unit = "cm", ), size= 10),
      axis.title.y = element_blank(),
      axis.text = element_text(size = 8),
      panel.grid.major.y = element_blank(),
      strip.text.y.left = element_blank(),
      legend.position = c(0,-0.4),
      legend.text = element_text(size = 8),
      legend.title = element_blank())+
ylab("Porcentaje de familias proteicas")
```



Se repite el mismo proceso para representar las anotaciones funcionales del umbral de 85% .

Proteoma central

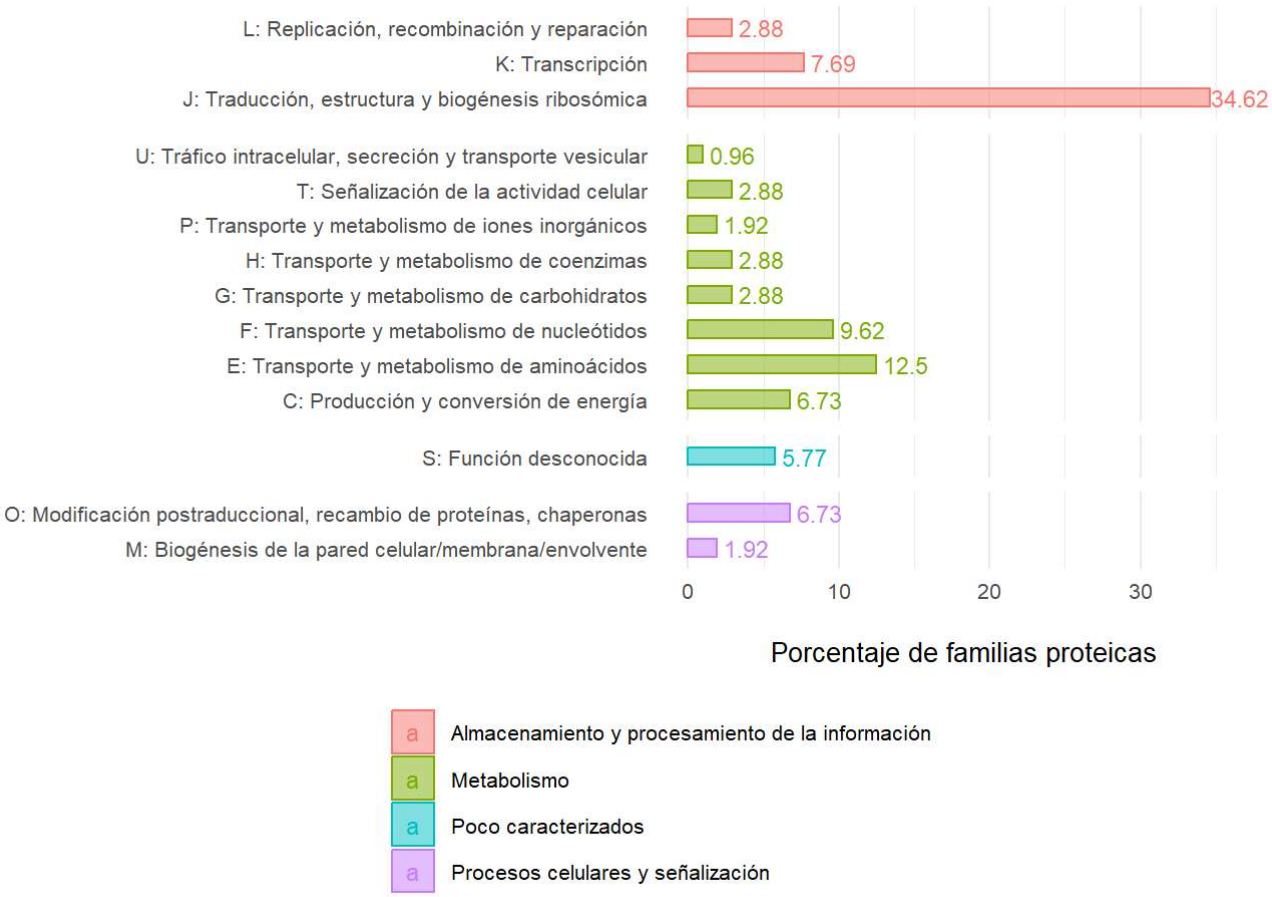
Análisis de los resultados para el umbral de 85%

```
setwd ("D:/RESULTADOS_DEFINITIVOS_TFM/eggNOG/Identidad_85")

eggNOG_85 <- read.csv("out.emapper.annotations",
                  sep = "\t",
                  comment.char = "#",
                  header = FALSE,
                  na.strings = "-")

colnames(eggNOG_85) <- c("query", "seed_ortholog",
                         "evalue", "score", "eggNOG_OGs",
                         "max_annot_lvl", "COG_category", "Description",
                         "Preferred_name",
                         "GOs", "EC", "KEGG_ko", "KEGG_Pathway",
                         "KEGG_Module", "KE",
                         "GG_Reaction", "KEGG_rclass", "BRITE", "CAZy",
                         "BiGG_Reaction", "PFAMs")


select_eggNOG_85 <-eggNOG_85 %>%
  select("query", "COG_category", "KEGG_ko", "CAZy")


#Se explora la variable COG_caterogy
table(select_eggNOG_85$COG_category)
```

```
 C  CE   E  EF   F   G   H   J   K KLT  KT   L   M   O   P   S   T   U
 6   1  11   1   9   3   3  36   6   1   1   2   2   7   2   6   1   1
```

```
COG_85 <- select_eggNOG_85 %>%
  separate_rows(COG_category, sep="", convert= F) %>%
  filter(COG_category != "") %>%
  group_by(COG_category) %>%
  mutate(n = n()) %>%
  distinct(n) %>%
  ungroup() %>%
  mutate (porcentaje = round(n/sum(n)*100, 2))


#Figura 8 b)

COG_85 %>% inner_join(annotation) %>%
  mutate(Leyenda = case_when(
    COG_category == "C" ~ "Metabolismo",
    COG_category == "D" ~ "Procesos celulares y señalización",
    COG_category == "E" ~ "Metabolismo",
    COG_category == "F" ~ "Metabolismo",
```

```
    COG_category == "G" ~ "Metabolismo",
    COG_category == "H" ~ "Metabolismo",
    COG_category == "I" ~ "Metabolismo",
    COG_category == "J" ~ "Almacenamiento y procesamiento de la información",
    COG_category == "K" ~ "Almacenamiento y procesamiento de la información",
    COG_category == "L" ~ "Almacenamiento y procesamiento de la información",
    COG_category == "M" ~ "Procesos celulares y señalización",
    COG_category == "O" ~ "Procesos celulares y señalización",
    COG_category == "P" ~ "Metabolismo",
    COG_category == "Q" ~ "Metabolismo",
    COG_category == "S" ~ "Poco caracterizados",
    COG_category == "T" ~ "Metabolismo",
    COG_category == "U" ~ "Metabolismo",
    COG_category == "V" ~ "Metabolismo")) %>%
  ggplot(aes(x=labels, y= porcentaje, color= Leyenda, fill = Leyenda)) +
  geom_bar(stat = "identity", alpha= .5, width = 0.5) +
  coord_flip() +
  geom_text(aes(label = porcentaje), nudge_y = 2, size =3) +
  facet_grid(rows = vars(Leyenda),
             scales = "free_y", switch = "y", space = "free_y") +
  theme_minimal(base_family = "Roboto Condensed") +
  theme(plot.margin = margin(0.5, 0.5, 3, 0.5, unit = "cm"),
        strip.text.y = element_text(angle = 270, face = "bold", size= 6),
        strip.placement = "outside",
        axis.title.x = element_text(
          margin = margin(t = 0.5, b = 0.5, unit = "cm"), size = 10),
        axis.title.y = element_blank(),
        axis.text = element_text(size = 8),
        panel.grid.major.y = element_blank(),
        strip.text.y.left = element_blank(),
        legend.position = c(0,-0.4),
        legend.text = element_text(size = 8),
        legend.title = element_blank())+
  ylab("Porcentaje de familias proteicas")
```


L: Replicación, recombinación y reparación
K: Transcripción
J: Traducción, estructura y biogénesis ribosómica
U: Tráfico intracelular, secreción y transporte vesicular
T: Señalización de la actividad celular
P: Transporte y metabolismo de iones inorgánicos
H: Transporte y metabolismo de coenzimas
G: Transporte y metabolismo de carbohidratos
F: Transporte y metabolismo de nucleótidos
E: Transporte y metabolismo de aminoácidos
C: Producción y conversión de energía
S: Función desconocida
O: Modificación postraduccional, recambio de proteínas, chaperonas
M: Biogénesis de la pared celular/membrana/envolvente
2.88
7.69
34.62
0.96
2.88
1.92
2.88
2.88
9.62
12.5
6.73
5.77
6.73
1.92
0
10
20
30
Porcentaje de familias proteicas
a
Almacenamiento y procesamiento de la información
a
Metabolismo
a
Poco caracterizados
a
Procesos celulares y señalización