UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Estadística e Investigación Operativa Aplicadas y Calidad

Análisis de datos de microbioma para la predicción y caracterización de enfermedades

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Análisis de Datos, Mejora de Procesos y Toma de Decisiones

AUTOR/A: Nieto Romero, Maria Camila

Tutor/a: Tarazona Campos, Sonia

CURSO ACADÉMICO: 2022/2023

RESUMEN

El microbioma humano está constituido por el conjunto de microorganismos (bacterias, arqueas, virus, hongos y protistas) que están en nuestro cuerpo, y se ha demostrado en múltiples investigaciones el rol fundamental que tiene en la salud, así como su relación con muy diversas enfermedades o con la respuesta de los pacientes a distintos tratamientos. En concreto, se ha probado que el análisis de la microbiota es de gran ayuda para la predicción y caracterización de enfermedades tales como diabetes, hepáticas, o intestinales, entre otras. Por ello, esta es una línea de investigación prometedora, ya que trata de mejorar el diagnóstico y tratamiento de diferentes enfermedades, lo cual supondría un avance importante para la medicina.

Sin embargo, el análisis bioinformático y estadístico de datos de microbioma medidos con técnicas de secuenciación de alto rendimiento no está exento de retos. Por una parte, está la dimensión de los datos, ya que se suele trabajar con cientos de variables pero con no más de 100 pacientes en la mayoría de los casos. Además, son datos con una alta variabilidad y ruido debido a la técnica, y con un elevado número de valores nulos. Esto implica que se debe elegir cuidadosamente el pre-proceso o transformación de los datos, previo al análisis estadístico. Por otra parte, se requieren modelos de clasificación que minimicen el error de predicción pero que funcionen bien bajo supuestos de multicolinealidad y que permitan una adecuada interpretación de los resultados, de tal forma que los biólogos, médicos, etc. puedan identificar qué microorganismos son más relevantes para el diagnóstico o tratamiento de la enfermedad estudiada. Todas estas cuestiones son todavía objeto de discusión entre la comunidad científica y aún están por definir unos procedimientos estandarizados para el análisis de este tipo de datos.

Así pues, el objetivo de este estudio es contribuir al establecimiento de un protocolo de análisis, tanto en la parte del pre-proceso de los datos como en la metodología estadística empleada para su análisis. Para ello, se utilizará una colección de 6 bases de datos de microbioma de pacientes sanos y enfermos públicamente accesibles para el estudio de 5 patologías diferentes: cirrosis hepática, enfermedad inflamatoria intestinal (Inflammatory Bowel Desease, IBD), obesidad, diabetes tipo 2 y cáncer colorrectal.

Los métodos de aprendizaje automático comúnmente empleados en este tipo de análisis son la regresión logística, el Random Forest o las máquinas de soporte vectorial (SVM). Sin embargo, las metodologías basadas en la proyección sobre estructuras latentes como es la Regresión en Mínimos Cuadrados Parciales (PLS), han sido poco o nada utilizadas en la literatura para abordar este problema. En este trabajo, aplicaremos en concreto el PLS Discriminante (PLS-DA) como método de clasificación y compararemos los resultados con los obtenidos mediante los métodos mencionados para identificar las ventajas o desventajas de cada uno, así como la influencia del pre-procesado de los datos en dichos resultados.

ABSTRACT

The human microbiome is made up of the set of microorganisms (bacteria, archaea, viruses, fungi and protists) that are in our body, and the fundamental role it plays in health has been demonstrated in multiple investigations, as well as its relationship with very diverse diseases or with the response of patients to different treatments. Specifically, it has been proven that the analysis of the microbiota is of great help for the prediction and characterization of diseases such as diabetes, hepatic, or intestinal, among others. For this reason, this is a promising line of research, since it tries to improve the diagnosis and treatment of different diseases, which would mean an important advance for medicine.

However, bioinformatics and statistical analysis of microbiome data measured with high-throughput sequencing techniques is not without its challenges. On the one hand, there is the dimension of the data, since we usually work with hundreds of variables but with no more than 100 patients in most cases. In addition, they are data with high variability and noise due to the technique, and with a high number of null values. This implies that the pre-processing or transformation of the data must be chosen carefully, prior to the statistical analysis. On the other hand, classification models are required that minimize the prediction error but that work well under assumptions of multicollinearity and that allow an adequate interpretation of the results, in such a way that biologists, doctors, etc. can identify which microorganisms are most relevant for the diagnosis or treatment of the disease studied. All these issues are still the subject of discussion among the scientific community and standardized procedures for the analysis of this type of data have yet to be defined.//[0.2cm]

Thus, the objective of this study is to contribute to the establishment of an analysis protocol, both in the pre-processing of the data and in the statistical methodology used for its analysis. For this, a collection of 6 publicly accessible microbiome databases of healthy and sick patients will be used for the study of 5 different pathologies: liver cirrhosis, inflammatory bowel disease (Inflammatory Bowel Desease, IBD), obesity, type 2 diabetes and cancer. colorectal.//[0.2cm]

The machine learning methods commonly used in this type of analysis are logistic regression, Random Forest or support vector machines (SVM). However, methodologies based on the projection on latent structures such as Regression in Partial Least Squares (PLS), have been little or not used in the literature to address this problem. In this paper, we will specifically apply the PLS Discriminant (PLS-DA) as a classification method and we will compare the results with those obtained by the aforementioned methods to identify the advantages or disadvantages of each one, as well as the influence of the pre-processing of the data in these results.

AGRADECIMIENTOS

Todo el trabajo realizado fue posible gracias al apoyo incondicional de mis padres y hermano, quienes estuvieron a mi lado en los momentos difíciles y siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Sonia, mi tutora, cuya paciencia y dedicación en incontables ocasiones agradezco profundamente, sin sus palabras y correcciones no hubiese podido terminar, todo esto lo llevaré para aplicar en mi futuro profesional.

Gracias, también, a mis amigos y mis compañeros de máster, que me dieron su apoyo. Y a todos los docentes que han sido en algún momento parte de mi camino universitario y a quien agradezco por todos los conocimientos necesarios que tengo actualmente.

Nada de esto hubiera sido posible sin ustedes.

María Camila Nieto Romero
Valencia, 2022

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

El concepto de salud se torna cada vez más complejo a raíz de que se encuentran nuevos criterios y nuevas influencias de factores tanto internos como externos a esta. Se acostumbra a pensar que el ser humano es un entidad que se autorregula, siendo capaz de nutrirse y reproducirse. Durante el estudio de la medicina se ha considerado al hombre como enemigo de los microorganismos que lo componen, sobre todo en la era del SIDA, antrax, ébola y demás, donde se promovía el terror a los virus y bacterias [15] y la mejor arma como humanos era el desarrollo de nuevos y mejores antibióticos. Sin embargo, en la ultima década se ha establecido que la interacción entre el ser humano y los microorganismos es mucho más compleja [32], ya que el ser humano es un superorganismo que cuenta con millones de microorganismos que tienen funciones vitales para él [11]. De hecho no hay modo en el cual los humanos podamos vivir si no es en simbiosis con las bacterias beneficiosas que es lo que se llama eubiótica [15].

A este conjunto de microorganismos que colonizan nuestro organismo se le conoce como microbiota. La microbiota se define como la comunidad ecológica de microorganismos comensales, simbióticos y patógenos [8]. El microbioma se subdivide en cuatro grupos : virus, bacterias, hongos y parásitos. Los virus son las partículas infecciosas de menor tamaño y necesitan de las células huésped para su replicación [34]. Las bacterias, son microorganismos procariotas , es decir, microorganismos sencillos.Su clasificación se hace a partir de tamaño, forma y disposición espacial. Algunas bacterias mantienen una relación parasitaria temporal, otras lo hacen de manera permanente [34]. Los hongos por el contrario, son microorganismos eucariotas pertenecientes al reino fungi, como la levadura o el moho. Por último están los parásitos, que son los más complejos. Su ciclo de vida con el ser humano puede ser una relación permanente o atravesar etapas de desarollo[34].

Muchos de estos tipos de microbiota no tienen efectos directos sobre la salud del ser humano, o bien no se han descubierto o mantienen una simple relación de convivencia [8]. Es importante resaltar que la microbiota de cada individuo es única, es como una huella dactilar microbiana y como se explicará más adelante su composición dependerá de muchos factores.

Según los cálculos, aproximadamente la mitad de nuestras células son microbios, es decir, tenemos la misma cantidad de bacterias que de células humanas. Por lo tanto, el ser humano, está formado por una comunidad dinámica e interactiva de células humanas y microbianas [29]. Se ha comprobado que muchos de estos mircroorganismos ayudan a la absorción de nutrientes, aportan componentes esenciales o mejoran el funcionamiento del sistema inmunitario. Por esta razón, cualquier alteración en la comunidad microbiana puede ser la causa de ciertas enfermedades. Numerosas patologías están asociadas a este tipo de desequilibrios, como la enfermedad Crohn, obesidad, asma, hipertensión o trastornos de la piel [8].

Gracias a la ciencia, hoy en día conocemos la biodiversidad de microbios, que su composición es diferente en cada persona y que hay factores que influyen en el cambio de dicha diversidad. También se ha observado que las bacterias de cada parte del cuerpo son muy diferentes. La mayor diversidad microbiana la encontramos en el tracto intestinal y en la boca, la piel tiene diversidad media y donde

menos tipos de bacterias encontramos es en la vagina [29]. Sin embargo, existe en diferentes partes del cuerpo Figura 1.1. Otro de los descubrimientos que se han realizado, es que la microbiota cambia con la edad. Desde el mismo momento del nacimiento, se comienza a tener microbios. La composición de esta microbiota dependerá de muchos factores: la forma de nacer, dieta, ambiente, etc.

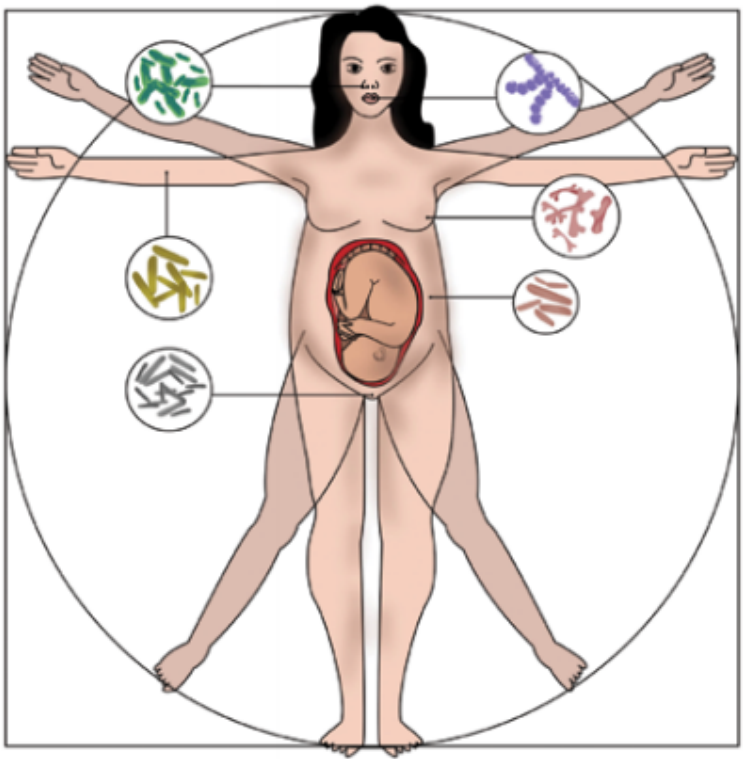

Figura 1.1: Ambientes donde habitan los microbiomas

Como se ha comentado anteriormente, se ha comprobado el papel crítico que juega la microbiota en la biología y salud de una persona. Donde se tiene mayor evidencia es en la parte nutricional [9] y en la defensa contra los patógenos. También evita la colonización de microorganismos patógenos, mantiene las barreras o refuerza uniones entre células epiteliales entre otras contribuciones. También juega un papel importante en otros aspectos que definen al ser humano como el sistema inmune, las funciones cerebrales o la secuencia de nuestro genoma. Por lo tanto la microbiota influye en nuestra forma de ser, tanto en la personalidad como en el estado emocional , ya que esto depende del cerebro, y la microbiota podría tener un rol crucial en las funciones nerviosas relacionados con el comportamiento, así como en el desarrollo neuronal y por lo tanto en enfermedades neurodegenerativas [29].

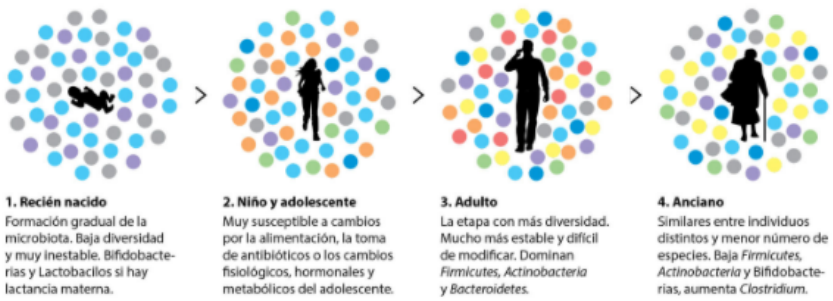

Figura 1.2: Evolución de la microbiota intestinal con la edad

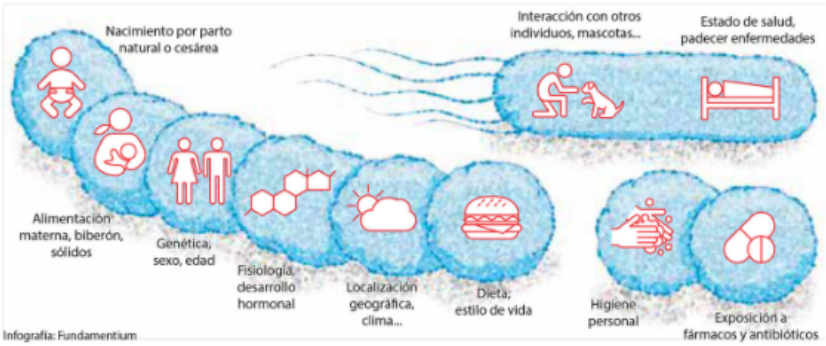

Figura 1.3: Factores que influyen en la composición de la microbiota

Las alteraciones de la microbiota no solo inducen patologías, también tienen efectos a largo plazo en la fisiología que deriva en depresión, autismo, o enfermedades neurodegenerativas como Alzheimer o Parkinson. De lo que aún no se tiene conocimiento total es de si las alteraciones de la microbiota son la causa o el efecto de la enfermedad, pero hay datos que indican que ambos factores se relacionan. Por ejemplo, se ha demostrado que la presencia de la bacteria Fusobacterium nucleatum es frecuente en tejidos de cáncer colorrectal. No solo se está investigando el efecto de la microbiota en la inhibición o desarrollo de una enfermedad sino también en la efectividad de algunos tratamientos y del aumento o disminuición de sus efectos secundarios [29].

Por lo tanto, como hemos visto la microbiota puede influir en múltiples aspectos de nuestra salud, por lo que se han desarrollado maneras de intervenir la microbiota para mantener la salud como son el uso de probióticos, prebióticos o transplantes de microbiota. Sin embargo, en este campo aún falta mucha investigación, ya que manipular o restaurar la microbiota es muy complicado, por ser un complejo consorcio de millones de microbios y células, que cambian entre individuos y a lo largo del tiempo. Por ello, cada vez mas los estudios se enfocan en conocer la composición de la microbiota, para desarrollar medidas preventivas, diagnósticas y terapéuticas basadas en esta [29]. Entre los avances se encuentra la transferencia de microbiota fecal, con una popularidad creciente, ya que se usa para tratar la diarrea recurrente causada por Clostridium difficile [9].

Todo esto abre la posibilidad de explorar muchos campos, como desarrollo de probióticos para identificar estirpes beneficiosas y establecer productos microbianos responsables de efectos antiinflamatorios o antioxidantes. Algo similar se hace a la hora de determinar la relevancia de los probióticos en la alimentación para plantear dietas nutricionales [4].

Por lo tanto, respecto al papel de la microbiota en la salud en un futuro va a ser sin duda parte de la medicina personalizada. Se está investigando ya su relación con la propensión a la obesidad, alergias, asma u otras enfermedades; todo esto considerando las circunstancias individuales de cada persona a lo largo de su vida.

En 1908 el profesor Elie Metchnikoff recibió el premio Nobel de medicina por sus investigaciones sobre inmunidad prestó especial atención en sus estudios a la flora intestinal y asoció la longevidad de algunas personas al consumo regular de lactofermentos que ayudan a mantener el microbioma saludable. A pesar de sus aportes, esta área de investigación cayó en el olvido y solo recientemente se ha redescubierto la investigación en el microbioma humano [15].

Desde hace muchos años, se han realizado esfuerzos para estudiar y describir la microbiota utilizando análisis bacterianos tales como cultivos in vitro, o secuenciación 16s pero con estos métodos muchos de los microorganismos no se pueden estudiar debido a que muchos tipos celulares requieren de circunstancias específicas para su correcto estudio, por esto se han venido implementado técnicas de cultivo independiente, que permiten aumentar la cantidad de datos ya que muchas cepas 'incultivables' que no habían podido ser analizadas hasta ahora, ya pueden ser estudiadas, conociendo

sus características, funciones y relaciones con otros seres vivos [8].

Los métodos de secuenciación empezaron en la década de 1990, siendo la subunidad 16s del ARN ribosomal la diana universal para la identificación bacteriana a partir del ADN, empleando métodos que usan la fragmentación del ADN y secuenciación con el método de Sanger. Este enfoque se usa hasta los años 2004-2005. Con el avance en la tecnología en todos los ámbitos se han podido desarrollar nuevos métodos que secuencian directamente el ADN fragmentado, sin la necesidad de clonar como se hacía anteriormente. Traen muchas ventajas: menor coste, reducción de tiempos y aceptable calidad de los datos. Sin embargo, uno de los problemas es el pequeño tamaño de las secuencias, que varían entre 150 y 500 pares, además de posibles errores de lectura. Actualmente las tecnologías más habituales son Solexa, de Illumina e IonTorrent de ThermoFisher [2].

El uso de técnicas moleculares de secuenciación masiva, conocidas como next generation sequencing, para determinar la composición de la microbiota ha permitido identificar y asignar taxonómicamente la mayoría de microorganismos sin cultivarlos como se hacía anteriormente. Algunas de estas técnicas microbiológicas se detallarán a continuación [9].

La primera técnica es shotgun(escopeta), en la que se analiza de forma aleatoria todo el ADN en la muestra, por lo tanto, se estudia la diversidad microbiana a gran escala. La muestra está compuesta por el genoma de los diferentes microorganismos, por lo que las lecturas de secuenciación se ensamblarán y alinearán a los genomas de referencia de la base de datos. Una de las desventajas de esta técnica es la existencia de lecturas que no corresponden a los genomas de referencia [20], por lo que la identificación de bacterias altamente divergentes es difícil mediante este enfoque [30].

La segunda técnica es la dirigida por el 16sRNA. La mayoría de estudios usan la secuenciación usando el gen 16rRNA para bacterias, y el gen 18s rRNA para especies eucariotas y la región nuclear de espaciador transcrito ribosomal (ITS) para el estudio de hongos [30]. Este gen es conocido como ARN ribosomal 16s, y consta de unas regiones ampliamente conservadas en el reino bacteriano conocidas como V1-V9, que son específicas de bacterias y permiten diferenciar entre géneros y especies. Tras secuenciar los fragmentos se comparan con la información obtenida en las bases de datos. Existen varias bases de datos de secuencias de referencia y taxonomía, que permiten determinar qué bacterias estaban presentes o no en las muestras [20].

El proceso de extracción de la información la podemos observar de manera general a continuación:

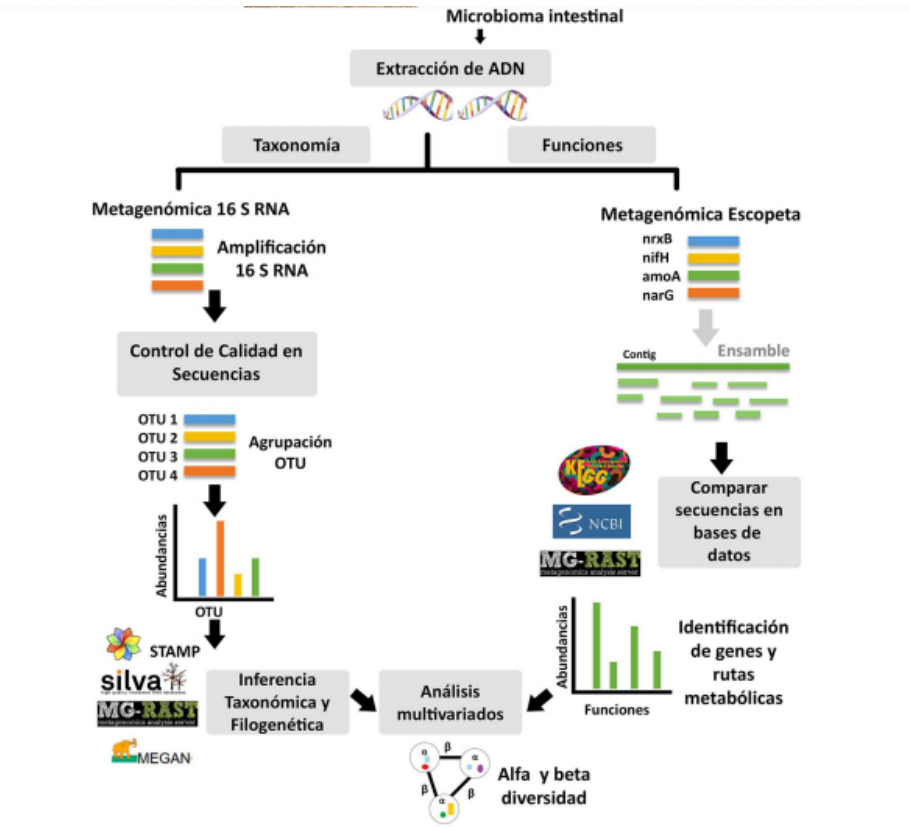

Figura 1.4: Proceso de análisis metagenómico

En el análisis de datos de las diferentes técnicas, se hace uso de la última clasificación taxonómica vigente propuesta por Richard Worse en 1990, quien en base a los estudios genéticos sobre el ARN establece una clasificación filogenética en tres dominios: Bacteria, Archaea y Eukarya [49], como se observa en la Figura 1.5.

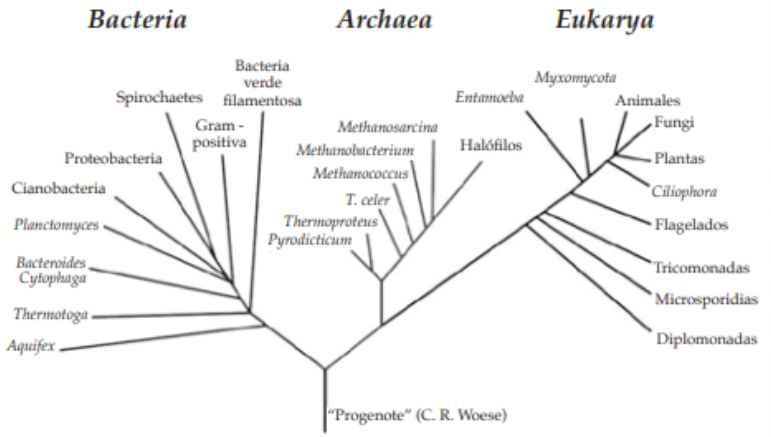

Figura 1.5: Clasificación Taxonómica según C.R. Worse

En cuanto a la recolección de las muestras para cualquiera de las técnicas, cada zona del organismo alberga diferente microbiota, por lo tanto la muestra dependerá de la zona a estudiar. La gran mayoría de los estudios se centran en la microbiota intestinal, ya que es la más numerosa y la que más implicación tiene en la salud. Se suelen usar las heces ya que son rápidas de obtener y no es un método invasivo. Una de las desventajas de este método de obtención es que las heces no representan la totalidad de la microbiota, y las bacterias de los tramos intestinales más superiores están degradadas

impidiendo su correcta detección. También se suelen usar biopsias, pero el problema radica en que solo se recoge la microbiota de un punto concreto [9].

Para la recolección de muestras, hay que tener en cuenta algunas circunstancias en las que se pueden introducir sesgos importantes que afectarán a los resultados, como la elección de muestra, conservación y transporte, y el método con el cual se extraen los ácidos nucleicos [2].

En cuanto al procesamiento de la muestra, dentro de la misma pueden existir variaciones en la composición de su microbiota. Por ello se realiza la homogeneización antes de extraer y luego se realiza la extracción y purificación de los ácidos nucleicos de calidad, es decir, sin arrastrar sustancias que puedan inhibir reacciones de PCR. Finalmente, se aplican las técnicas de secuenciación [9].

El presente trabajo final del master está enmarcado dentro de los objetivos del programa de cooperación Europea en Ciencia y Tecnología llamado "Statistical and machine learning techniques in human microbiome studies"(COST).[48]. El objetivo principal del proyecto es crear una red de colaboración para investigar el uso de técnicas estadísticas de machine learning para el estudio de enfermedades a partir de datos del microbioma obtenidos mediante técnicas de secuenciación [48].

Dentro del proyecto ML4Microbiome, existen varios grupos de trabajo que tienen como objetivos la revisión de métodos estadísticos desarrollados hasta el momento para el análisis de datos de microbiota, la recopilación de datos públicos, y la comparación de estrategias de procesamiento y análisis de dichos datos para proponer un protocolo de análisis estandarizado.

En este trabajo, se ha contribuido a la consecución de estos objetivos. En concreto, hemos obtenido y organizado los datos utilizados en el artículo 'Machine learning meta-analysis of large metagenomic datasets: tools and biological insights' y hemos comparado sobre ellos distintas estrategias de preprocesado y varias técnicas estadísticas como: PLS-DA, Máquinas de soporte vectorial y Random Forest. Las 6 bases de datos de microbioma que hemos utilizado, estudian cinco enfermedades diferentes que se describen a continuación:

- **Enfermedad Inflamatoria Intestinal:** La característica de esta enfermedad es como su nombre indica, una inflamación intestinal mantenida que puede ser provocada por la activación de receptores del sistema inmune, principalmente bacterias. La microbiota intestinal presenta una reducción en la biodiversidad, especialmente en Firmicutes [2].

  Otro tipo de enfermedad, dentro de este grupo intestinal es la enfermedad de Crohn, en la que se evidencia una reducción en el grupo Clostridium IV, que se relaciona con la producción de ácidos grasos.
- **Cáncer Colorectal:** La microbiota intestinal podría estar implicada en el cáncer colorectal a través de la producción de metabolitos tóxicos o por provocar una respuesta inmune exagerada ante el estímulo bacteriano. Estudios que se han realizado en animales muestran que tras la adquisición de una microbiota intestinal específica, estos son capaces de reproducir el proceso tumoral[2].

  "Las especies bacterianas que se han relacionado con el cáncer de colon pertenecen a los géneros Bacteroides, Fusobacterium, Clostridium y/o Lactobacillus, pero es importante destacar que no existe suficiente consistencia científica. Sin embargo, sí se ha detectado una asociación estadísticamente significativa entre una bacteriemia/ endocarditis por Streptococcus gallolyticus subsp. gallolyticus y la existencia de un cáncer de colon, generalmente sin diagnosticar"[2].
- **Obesidad:** La obesidad es una enfermedad compleja que es considerada por la OMS como epidemia global. Su origen aún no está claro, los factores a los que se le suele atribuir son la predisposición genética, los malos hábitos alimentarios, un estilo de vida sedentario, hipotiroidismo, administración de algunos fármacos, etc [17].
  Recientemente se ha encontrado que dentro de las múltiples funciones de la microbiota in-

testinal, una de las de mayor importancia es la de contribuir a la digestión de alimentos para obtener energía. La implicación de la microbiota en esta enfermedad se ha demostrado en animales, pacientes y gemelos donde se ha observado una disminución de la densidad del filo Bacteroidetes. Otro hecho importante es el descubrimiento de que, determinadas especies se asocian con fenotipos delgados, y confieren un papel protector frente a la obesidad [2].

- **Diabetes Tipo II:** Esta enfermedad es un desorden complejo y se sospecha que la microbiota intestinal puede influenciar ya que se ha visto que estos pacientes presentan diferencias en la composición de la microbiota frente a los pacientes sanos. Sin embargo, aún se sigue investigando esta relación [2].
- **Cirrosis:** Se ha comprobado que las alteraciones del microbioma intestinal como una disbiosis con endotoxemia puede incidir en trastornos de las funciones hepáticas a causa de la disminución de las bifidobacterias o el incremento enterobacterias.
  En base a estos estudios hemos visto relaciones de enferemedades con cambio en el microbiota, como se observar en la Figura 1.6.

Enfermedad	Cambios en la microbiota
Diabetes tipo 2	↓ *Firmicutes*, ↓ *Clostridia*, ↑ *Bacteroidetes-Prevotella versus* ↓ *Clostridia coccoides-Eubacterium rectale*, ↑ *β-proteobacteria*, ↑ *ratio Firmicutes/Bacteroidetes.*
Alergias	↓ *Lactobacillus spp*, ↑*Bifidobacterium adolescentis*, ↓*Clostridium difficile*, ↓*Helicobacter pylori.*
Enfermedad de Crohn	↑ *Bacteroides ovatus*, ↑ *Bacteroides vulgatus* ↓*Bacteroides uniformis*
Autismo	↑*Bacteroidetes*, ↑ *Proteobacteria*, ↓ *Actinobacteria*, ↓*Firmicutes.*
Cáncer gástrico	↑ *Helicobacter pylori.*
Obesidad	↓*Bacteroidetes*, ↑ *Lactobacillus*, ↑ *ratio Firmicutes/Bacteroidetes*, ↑*Methanobrevibacter smithii.*

Figura 1.6: Ejemplos de disbiosis en algunas enfermedades (incluida la obesidad). Jose C. Clemente, et al. "The Impact of the Gut Microbiota on Human Health: An Integrative View". Cell (2012) 148; 1258-1270.

1. Preparación de los datos: Los datos metagenómicos poseen características únicas como las diferencias que existen en la profundidad de secuenciación , escasez de información (debido a la alta presencia de ceros), y la gran varianza de la distribución (sobre-dispersión). Estos atributos hacen que sea inapropiado aplicar métodos estadísticos directamente sobre estos datos, y que sea necesario procesarlos previamente para eliminar o reducir estos sesgos introducidos por la propia tecnología de la secuenciación [12].

2. Técnicas estadísticas:

Las técnicas estadísticas que más se usan en el análisis del microbioma son los relacionados con el análisis de abundancia diferencial. Lo que se quiere es identificar taxones microbianos que son abundantes o insuficientes para algunas personas con ciertas condiciones de referencia. Para esto se han aplicado una variedad de herramientas basadas en modelos de regresión lineal logística, binomial

negativa, modelos normales que tienen en cuenta la presencia de muchos ceros, se han usado también métodos bayesianos, o modelos multivariantes [51].

Para examinar datos microbiomas se tienen métodos de las estadísticas multivariadas clásicas como el análisis de componentes principales (PCA), Análisis de correlación canónica (CCA), Análisis de factores múltiples (MFA) o una variación del Análisis de componentes principales con variables instrumentales (PCA-IV). En base a estos se han creado nuevas metodologías como el análisis de CoInertia (CoIA) el cual facilita el análisis de la variación de la abundancia de especies en función de ciertas condiciones ambientales, o el MFA el cual es visto como una versión refinada del PCA para este campo [47] Figura 1.7.

En comparación a estos enfoques clásicos, se han usado otras herramientas como regresión en mínimos cuadrados parciales (PLS), que es adecuado para encontrar predictores en presencia de matrices de respuesta de alta dimensión, o el análisis de correspondencia canónico (CCpnA) para analizar un conjunto de datos de recuento. A diferencia de los métodos estadísticos tradicionales, el enfoques de estos modelos es más complejo por lo tanto requiere un cálculo más extenso que las técnicas tradicionales [47].

Property	Algorithms
Analytical solution	Concat. PCA, CCA, CoIA, MFA, PTA, Statico/Costatis
Require covariance estimate	Concat. PCA, CCA, CoIA, MFA, PTA, Statico/Costatis
Sparsity	SPLS, Graph-Fused Lasso, Graph-Fused Lasso
Tuning parameters	*Sparsity*: Graph-Fused Lasso, PMD, SPLS *Number of Factors*: PCA-IV, Red. Rank Regression, Mixed-Membership CCA Prior *Parameters*: Mixed- Membership CCA, Bayesian Multitask Regression
Probabilistic	Mixed-Membership CCA, Bayesian Multitask Regression
Not Normal or Nonlinear	CCpNA, Mixed-Membership CCA, Bayesian Multitask Regression
>2 Tables	Concat. PCA, CCA, MFA, PMD
Cross-Table Symmetry	Concat. PCA, CCA, CoIA, Statico/Costatis, MFA, PMD

Figura 1.7: Comparación de algunos métodos usados en el estudio de microbioma

Debido a el efecto de multicolinealidad de los datos, se suele utilizar la regresión ridge o lasso para los coeficientes.

	Taxa				
Sample	**1**	**2**	**...**	***K***	***Total***
1	X_{11}	X_{12}	...	X_{1K}	$N_{1\cdot}$
2	X_{21}	X_{22}	...	X_{2K}	$N_{2\cdot}$
⋮	⋮	⋮	⋱	⋮	⋮
P	X_{P1}	X_{P2}	...	X_{PK}	$N_{P\cdot}$
Total	$N_{\cdot 1}$	$N_{\cdot 2}$	...	$N_{\cdot K}$	$N_{\cdot\cdot}$

doi:10.1371/journal.pone.0052078.t001

Figura 1.8: Formato de datos para p sujetos con K niveles taxonómicos.

Los datos de recuento como los presentados en la Figura 1.8 se suelen analizar utilizando una distribución multinomial. Ya que a medida que aumenta el número de puntos de muestra (es decir, el

número de lecturas) dentro de cada muestra, las frecuencias de taxones en todas las muestras convergen al mismo valor. Cuando los datos exhiben una sobredispersión, este resultado de convergencia no ocurre (es decir, las frecuencias de taxones en todas las muestras no convergen a los mismos valores) y el modelo multinomial es incorrecto ya que la prueba de hipótesis basada en el modelo multinomial en presencia de sobredispersión puede resultar en un mayor error de tipo I , por esta razón para los casos de sobredispersión es mejor hacer uso del modelo Dirichlet [23].

En cuanto a las pruebas de hipótesis, se podria considerar comparar muestras de microbiomas con un vector de frecuencias de taxones planteado por el investigador o obtenido en estudios anteriores. Esto podría hacerse para probar si las nuevas muestras provienen de la misma población o de una diferente de muestras anteriores, como comparar una población con los controles sanos. Esta prueba es análoga a una prueba t de una muestra en estadística clásica ($H_1 : \pi = \pi_0$). Otro uso de estas técnicas son comparar la composición del microbioma entre muestras de sujetos con diferentes condiciones (sanos vs enfermos) ($H_1 = \pi_0 \neq \pi_1$). Cuando hay más de dos condiciones se usa el análisis de la varianza [23].

Un objetivo complementario es la predicción de enfermedad. Esto motiva al uso diferentes métodos para desarrollo y evaluación de modelos. Por ejemplo, se ha aplicado regresión penalizada multivariante, o máquinas de soporte vectorial junto a la validación cruzada para evaluar la precisión de la predicción [51].

Por todo esto una de las principales motivaciones para realizar este trabajo de investigación es la necesidad de aplicar herramientas estadísticas diferentes a las ya usadas comúnmente con el fin de no solo poder predecir a través de los datos de la microbiota el estado de enfermedad de una persona, sino entender qué bacterias son las que más influyen en la presencia o no de una enfermedad, ya que son muy escasos los estudios dedicados a investigar y desarrollar metodologías estadísticas que generen resultados de clasificación y importancia de bacterias, probando diferentes aplicaciones metodológicas para poder saber la mejor manera de proceder dependiendo de la naturaleza de los datos y con esto ayudar a futuros análisis.

La estructura a grandes rasgos de este trabajo contiene los objetivos planteados que se quieren lograr, la metodología, en la cual encontramos la descripción de los datos usados, las técnicas de pre-procesamiento que se implementaron al igual que las técnicas estadísticas tanto de modelación como de validación y comparación de modelos. Luego, se tienen los resultados en los que hablamos de los descubrimientos más importantes, la comparación de los modelos, los resultados del artículo base (Pasolli) y los resultados obtenidos. En base a estos se muestran las conclusiones, en las cuales también se presentará una guía de maneras de proceder adecuadas para el correcto tratamiento de los datos de microbioma y mejora de precisión de los modelos para predecir una enfermadad a partir de ellos.

CAPÍTULO 2

OBJETIVOS

2.1. OBJETIVO GENERAL

Se ha demostrado en diferentes estudios el papel fundamental del microbioma humano en el inicio y desarrollo de enfermedades tales como la enfermedad inflamatoria intestinal, la diabetes, la cirrosis o incluso enfermedades neurodegenerativas. Actualmente, la cuantificación de las distintas especies que conforman nuestra microbiota se obtiene mediante técnicas de alto rendimiento (secuenciación), que permiten estudiar en un mismo experimento todas las especies presentes en la muestra biológica pero que, al mismo tiempo, generan datos ruidosos y de alta dimensionalidad que requieren de procedimientos específicos de procesado y de técnicas multivariantes para su análisis.

El presente trabajo está enmarcado en el proyecto "Statistical and machine learning techniques in human microbiome studies", que es un proyecto europeo COST en el que colaboran más de 30 países. Así pues, en este estudio abordamos uno de los objetivos principales de dicho proyecto: proponer procedimientos de análisis estandarizados para el estudio de enfermedad con datos de microbioma humano. En concreto, se propondrán y compararán sobre un conjunto variado de bases de datos de microbioma distintas estrategias de análisis que incluyen tanto la forma de procesar los datos de microbioma como el método estadístico de clasificación aplicado para discriminar entre enfermos y controles.

2.2. OBJETIVOS ESPECÍFICOS

- Obtener, organizar, procesar y explorar distintas bases de datos públicas de enfermedades para el análisis de las mismas.
- Proponer y aplicar diferentes tratamientos a todas las bases de datos, tanto para normalización como filtrado de variables.
- Seleccionar y aplicar diferentes técnicas estadísticas multivariantes para la discriminación entre sanos y enfermos a partir de su microbiota.
- Evaluar y comparar el desempeño de las distintas técnicas estadísticas aplicadas en combinación con el procesado de los datos, utilizando para ello distintas medidas de la tasa de error.
- Analizar las diferencias de desempeño de las estrategias comparadas para entender su relación con las características de las bases de datos.

CAPÍTULO 3

METODOLOGÍA

3.1. DATOS ANALIZADOS

En este trabajo se han analizado datos de microbioma medidos en muestras de heces mediante técnicas de secuenciación masiva *(shotgun metagenomics)* de 6 estudios . Las 6 bases de datos con las que hemos trabajado se encuentran disponibles públicamente en http://segatalab.cibio.unitn.it/tools/metaml y corresponden al estudio de 5 enfermedades.

Cada matriz de datos contiene la abundancia relativa de las bacterias, y una colección de covariables en las cuales se incluyen variables demográficos, entre otras. El número de pacientes y OTUS(Unidad taxonómica operativa) de cada una de las enfermedades se recoge en la tabla 3.1.

3.1.1. Cirrosis

La cirrosis es una enfermedad hepática que puede darse por diversas causas como abuso de alcohol, obesidad o infecciones de virus [39].

La base de datos de Cirrosis contiene 232 pacientes de China, tanto con cirrosis como pacientes sanos,y se construye usando la metodología MetaHIT (Metagenómica de Tracto Intestinal Humano). Las covariables de las que se tiene información para los pacientes son 14, sin embargo, para el análisis de diferenciación entre enfermos y sanos nos quedaremos únicamente con 9 covariables.

3.1.2. Cáncer colorectal

Es un tipo de cáncer que se origina en el recto o el colon, con el crecimiento en el revestimiento interno de pólipos conocidos también como adenomas. Este tipo de cáncer es uno de los cánceres más comunes, con más de 1.2 millones de nuevos casos al año [21].La base de datos contiene 121 pacientes de Francia, dentro de ellos hay 48 casos de pacientes con Cáncer colorectal, a los cuales se les realizó la colonoscopia para diagnosticar ya sea adenoma , Cáncer Colorectal o la ausencia de enfermedad. Los adenomas de menos de 10 cm se consideran controles para el estudio.

3.1.3. Enfermedad Inflamatoria Intestinal

La enfermedad inflamatoria intestinal (EII) engloba dos patologías, la colitis ulcerosa y la enfermedad de Crohn. Las dos patologías alteran la capacidad del organismo para la digestión de alimentos y absorción de nutrientes. Se diferencian en la zona de afectación: la colitis ulcerosa se ubica en la pared del intestino grueso, mientras que la enfermedad de Crohn puede estar en cualquier parte del aparato digestivo.

Dentro de la base de datos hay pacientes de España y Dinamarca. Todos los pacientes de Dinamarca son controles, mientras que los pacientes de España son pacientes sanos o con enfermedades inflamatorias intestinales crónicas como Crohn o colitis ulcerosa [37].

3.1.4. Obesidad

Es una enfermedad que consiste en tener una cantidad excesiva de grasa corporal. Aumenta el riesgo de otras enfermedades tales como cardiopatías, diabetes y algunos tipos de cánceres. Es el resultado de factores hereditarios , dietas o el entorno, entre otros.

Se obtiene la información de 253 voluntarios de los Comités Éticos de la Región Capital de Dinamarca. Todos los individuos fueron examinados después de un ayuno nocturno con muestras de sangre y mediciones antropométricas. Se extrajo el ADN y se secuenció creando los perfiles metagenómicos [27].

3.1.5. Diabetes Tipo 2

La diabetes tipo II es una enfermedad muy común. Se diagnostica cuando el cuerpo deja de producir o usar la insulina,que es una hormona que ayuda a la glucosa a entrar a las células para darles energía. La ausencia de insulina hace que el nivel de glucosa en la sangre aumente y provoca problemas serios en el corazón, ojos, riñones, encías, entro otros. Los factores de riesgo de la diabetes son la obesidad, antecedentes familiares y falta de ejercicio físico.

Se realizó un estudio en mujeres de dos cohortes una en China y otra en Europa, donde se evaluaron los perfiles metagenómicos de personas sanas y con diabetes. Se tienen dos bases de datos relacionadas con la misma enferemedad. La base de datos T2D es la realizada en mujeres en China, mientras que la base WT2D es la realizada con muestras de diferentes países de Europa [22],[38].
En cuanto a las covariables cada estudio recogió diferente número de variables asociadas a la enfermedad y demográficas.

3.2. PRE-PROCESAMIENTO DE LOS DATOS

Para cada uno de los estudios detallados anteriormente, se dispone de tres tipos distintos de información: La descripción taxonómica, la abundancia de cada OTU en cada paciente y las variables auxiliares (covariables).

- Descripción taxonómica: Debido a que las bacterias vienen con su categorización de la bacteria (Familia, orden, especie, tipo) unas con mayor detalle de información que otras, se crea esta tabla para relacionar el número OTU con su descripción taxonómica, en concreto, se estudiarán aquellas bacterias (u OTU's) que tengan información taxonómica a nivel de especie.
- OTU's: En esta matriz se recoge para cada OTU su abundancia en cada uno de los pacientes. Recordemos que la abundancia de una bacteria viene estimada por el número de lecturas de secuenciación asignadas a dicha bacteria en los pasos previos de alineación de lecturas a los genomas de referencia.
- Información de las muestras: Contiene la información para cada paciente respecto a ciertas variables auxiliares que pueden ser demográficas o propias del estado de la enfermedad.

Las tres tablas de cada estudio se relacionaron usando el paquete de R phyloseq [31] que permitió filtrar la información de las bacterias a nivel de especie. Por tanto, en los pasos siguientes, se trabajará con la abundancia de las especies la matriz de datos X tiene la siguiente forma:

X	M_1	M_2	M_3	M_4	...	M_j	...	...	M_p	
I_1	45	47	321	0	...	...	...	...	1236	S_1
I_2	4	0	0	0	...	...	...	...	437	S_2
...	58	0	346	18	...	...	...	...	4	...
I_i	0	0	0	287	...	x_{ij}	...	...	32	S_i
...	0	222	1	0	...	...	...	...	55	...
...	3	7	0	67	...	...	...	...	9	...
I_n	79	498	784	97	...	...	...	...	0	S_n

Figura 3.1: Muestra datos de abundancia

Donde:

- I_i: El i-ésimo individuo.
- M_j : La j-ésima bacteria (u OTU).
- x_{ij} : La abundacia para el individuo i de la j-ésima bacteria.
- n: Total de individuos.
- p: Total de bacterias.
- La profundidad de secuenciación de la muestra del paciente *i* es el total de lecturas de secuenciación asociadas a todas las bacterias para ese individuo y viene dada por

$$S_i = \sum_{j=1}^{p} x_{ij}$$

La profundidad de secuenciación es importante debido a que determina que los datos sean composicionales y que las abundancias en los diferentes pacientes puedan ser comparables.

3.2.1. Filtrado previo de variables

El propósito de este filtrado previo de datos es eliminar las variables (especies) con baja presencia en las muestras porque no son informativas para el posterior análisis estadístico, y además introducen ruido.

Después de considerar la especie de cada bacteria y eliminar tanto las OTU's sin información de especie como las bacterias que tienen valor 0 en todos los pacientes, se registra el número de variables que se conservan (p) y se realiza un estudio de prevalencia:

Se usaron dos métodos que se quieren comparar:
Filtrado mínimo de datos (S1) : EL procedimiento elimina las especies que tienen abundancia cero en todas las muestras analizadas.

Especies de Baja abundancia (S2) : Se decide trabajar con aquellas especies que están presentes en más de un 20 % de las muestras de casos o controles ya que muchas de las especies con menos de esta prevalencia se pueden deber a errores de secuenciación o contaminaciones de bajo nivel.

Para tomar la decisión del nivel óptimo de 20 % se realizó el siguiente análisis:

Tomando las bacterias que tienen alguna información en alguna de las muestras, registramos el número (%) de individuos con abundancia mayor a 0 ,y los representamos en un gráfico acumulado, donde en el eje X se encuentra el número de individuos que tomaríamos y en el eje Y el número de bacterias con las cuales podríamos contar. Y podemos observar y tomar la decisión del número de individuos mínimo que debe tener la bacteria para que entre al análisis.

Estos filtros se recomiendan encarecidamente antes de cualquier análisis estadístico [12] y se aplican haciendo uso de las funciones de los paquetes microbiome [24], hildiv [3] y MixOmics [16].

Al aplicar los filtros S1 y S1 a todas las bases de datos con la ayuda de los siguientes paquetes de R , estas son las dimensiones resultantes:

Enfermedad	No. Variables S1	No. variables S2	No. casos	No. controles
Cirrosis (CIRR)	530	190	118	114
Cancer Colorectal (CRC)	493	184	48	73
Enfermedades intestinales inflamatorias (IBD)	425	175	25	85
Obesidad (OBES)	446	155	164	89
Diabetes Tipo 2 (T2D)	556	166	170	174
Diabetes Tipo 1 (WT2D)	367	165	53	43

Tabla 3.1: Resultados de bases de datos luego de aplicación filtos

3.2.2. Transformación de los datos

La normalización es una transformación de los datos que tiene el potencial de eliminar los sesgos y variaciones introducidos en las muestras por el proceso de secuenciación.

Sea la matriz de los datos de cada estudio $X \in \mathbf{N}^{nxp}$ donde se tiene n vectores filas que representan las muestras o sujetos y p- columnas que representan las categoría taxonómica de la bacteria, en este caso la especie. Así x_{ii} representa la abundancia de j en el sujeto i donde $i = 1, \cdots, n$ y $j = 1, \cdots, p$.

Se han considerado los siguientes tipos de normalización:

TSS (Total Sum Scaling): (N1) Es una normalización usada comúnmente que elimina el sesgo técnico relacionado con la profundidad de secuenciación diferente en las diferentes muestras. Para cada muestra i, se divide cada una de las abundancias individuales de cada especie por la profundidad de secuenciación S_i correspondiente. Esto nos lleva a un vector i transformado tal que la suma final de este vector es 1 ($\forall j : 1, \cdots, p$) [5]. Posteriormente se aplica una transformación logarítmica.

$$TSS(x_{ij}) = log(\frac{x_{ij}}{s_i} 10^6 + 1)$$

[5]

CLR (Centered Log Ratio): (N2) Propuesta por Aitchison, es una métrica de proporcionalidad usada en datos composicionales,es decir, que son parte de un todo (Ej: porcentajes). Dado que los datos de microbioma se obtienen a través de técnicas de secuenciación, el número de lecturas para cada muestra viene limitado por el número máximo de secuencias que proporciona el secuenciador, por lo que los valores de abundancia absoluta no son informativos, ya que dependen de la profundidad de secuenciación [42].

En esta transformación se basa en el logaritmo de la relación entre los conteos individuales de cada especie y la media geométrica de todas las especies.

El CLR esta propuesto así:

$$CLR(X_{ij}) = log\left[\frac{x_{ij} + 1}{G_i}\right]$$

Donde se divide respecto a la media geométrica de toda la composición $G_i = \sqrt{\prod_{j=1}^{p}(x_{ij} + 1)}$. Existen otras transformaciones para datos composicionales, por ejemplo, usando como denominador una especie fija en vez de la media geométrica de todas ellas.

Los dos métodos son robustos cuando no hay problemas de escasez de datos ya que no es posible hacer los cálculos cuando hay presencia de conteos de cero [12], así que para evitar resultados indefinidos se adiciona 1 unidad a todas las lecturas [5].

Tras aplicar los dos tipos de filtrado y de normalización se obtienen 4 matrices de datos diferentes dependiendo de la combinación de filtrado y transformación realizada, como se observa en la figura 3.2.

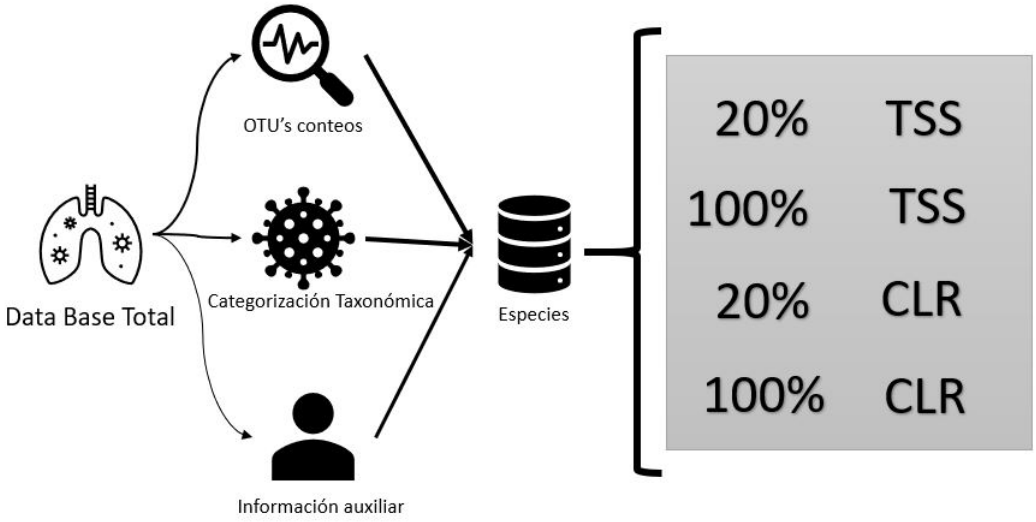

Figura 3.2: Flujo de preparación de los datos

3.2.3. Exploración mediante PCA:

Empezamos realizando un modelo PCA (usando el paquete FactoMineR [25]) en cada una de las enfermedades , con el fin de evaluar si en realidad existen variables (bacterias) que me ayuden a diferenciar entre personas enfermas y controles al igual que evaluar la existencia de outliers , verificar si esto supone problemas para el modelamiento de los datos y evaluar si la normalización es efectiva es examinar si las agrupaciones de datos normalizados es mejor que antes de normalizar, reflejando características biológicas y no características técnicas. Dada la alta dimensionalidad de este tipo de datos, se realizó mediante un PCA.

Lo primero a verificar la presencia de datos anómalos , para esto evaluamos calculan los puntajes T2 (valores extremos) y SCR (valores atípicos), se compararan teniendo en cuenta un limite de confianza de un 99 %. Ya que aun queda el 1 % de falsos positivos, descartamos ese número de los que exceden al limite y eliminamos el resto de anómalos.

T2 Hotelling:

$$T_i^2 = \frac{(Scores_i)^2}{\sum_i^k (\lambda_i)}$$

donde los scores son las coordenadas para las primeras dos componentes y λ_i el valor propio.

Luego calculamos el límite de confianza al 99 % del estadístico F99, con el que se compara los scores y se toma la decisión de si es o no un anómalo:

$$F_{99} : \frac{K(n^2-1)}{(n*(n-k))} * F_{0,99,K,n-K}$$

donde K es el número de componentes, en este caso 2, n el número de individuos y $F_{0,99,K,n-K}$

el valor crítico de la distribución F-Schnedecor. Si el valor $T_i^2 > F99$ consideramos este un valor atípico.

Debido a que el 1 % se consideraran falsos positivos, calculamos para cada enfermedad cuantos de estos anómalos serán realmente falsos positivos:

$$N_{Anmalos_Final} = individuos(T_i^2 > F99) - (0{,}01 * n)$$

Tomando así los

$$N_{Anmalos_Final}$$

individuos más anómalos.

SCR:

Para el cálculo de la suma de cuadrados residuales para cada individuo calculamos:

$$Error_i = X - Scores_i Cargas^T$$

Luego calculamos $SCR = \sum(Error_i)^2$. También se calcula el valor contra la que se hará la prueba: $\chi_L im = g * \chi_{0,99,h}$ donde $g = \frac{var(SCR)}{2*S\bar{C}R}$ y $h = \frac{2*(S\bar{C}R)^2}{var(SCR)}$.

Al igual que con la T2 se tiene en cuenta el número de falsos positivos.

3.3. MODELOS DE CLASIFICACIÓN

3.3.1. PLS-DA

Los métodos multivariantes de regresión como la regresión en componentes principales (PCR) y regresión en mínimos cuadrados parciales (PLS) tienen una gran popularidad en muchos campos, especialmente en aquellas situaciones donde hay muchas variables correlacionadas y reducido tamaño muestral [18].

El método de PCR realiza un análisis de componentes principales en las variables predictoras y luego realiza un modelo de regresión lineal múltiple utilizando como regresores las componentes principales obtenidas. Por otro lado, PLS construye unos componentes en la matriz de predictores X y en la matriz de respuesta Y que son combinaciones lineal en X e Y respectivamente, y que maximizan la covarianza entre X e Y:

$$argmax_{u_h^t u_h = 1; v_h^t v_h = 1} cov(u_h' X, v_h' Y)$$

Donde u_h y v_h son los vectores singulares de la descomposición en valores singulares (SVD) de $X^t Y$ para cada dimensión h. Estos vectores, u_h y v_h son llamados los loadings (cargas) [26].

El modelo del PLS está formulado como sigue:

$$Y = XB + E$$

Donde B es la matriz de los coeficientes de regresión, E es la matriz de residuales (nxp), X es la matria nxJ de datos.

El problema en muchos casos es la singularidad de la matriz $X^T X$. Puede ser por problemas de multicolinealidad o porque el número de predictores es más grande que el número de obervaciones.

Por lo tanto, PLS resuelve este problema descomponiendo la matriz X en una matriz ortogonal de scores T y una matriz de cargas Λ , y el vector Y se descompone en una matriz de scores T y una matriz de cargas Q. Siendo E y F las matrices de residuos. Hay dos ecuaciones fundamentales en el modelo PLS[18]:

$$X = T\Lambda^T + E$$

$$Y = TQ^T + F$$

[18] Si se define la matriz de pesos W, podríamos escribir la matriz de scores como

$$T = XW(\Lambda^T W)^{-1}$$

y reemplazándolo en el modelo PLS se obtiene

$$Y = XW(\Lambda^T W)^{-1}Q^T + F$$

[18]

donde la matriz de coeficientes de regresión B están dados por

$$\hat{B} = W(\Lambda^T W)^{-1}Q^T$$

[18]

El análisis de mínimos cuadrados discriminante (PLS-DA) es una variante del PLS mencionado anteriormente. Es usada cuando la variable Y es categórica y se transforma en una matriz con tantas variables dummies como categorías, donde van a tener un valor de 1 ó 0.

3.3.2. Random Forest

El modelo Random Forest es el método más famoso de bagging, entrenar diferentes modelos y ensamblar los resultados de lo mismo, en este caso estarían formados por un conjunto de árboles de decisión individuales (modelos predictivos formados por reglas binarias), cada uno entrenado con una muestra ligeramente distinta de los datos de entrenamiento generada mediante bootstrapping. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles entrenados anteriormente los cuales forman el modelo [44]. En cada uno de los árboles se eligen diferentes variables predictoras para la construcción del mismo.

A continuación mostraremos, los pasos para realizar un Random Forest [44](Figura 3.3):

1. Generar B pseudo-muestras de entrenamiento a partir de los datos originales por bootsrapping.
2. Se entrena cada uno de los árboles con su respectiva muestra sin procedimientos de poda con un subconjunto de las variables originales.
3. Para cada nueva observación, se hace la predicción en los B árboles, se utiliza la moda o media (dependiendo la naturaleza de la variable Y) de la variable cualitativa de los B árboles. Se le da más importancia a los árboles que tienen resultados más precisos, es decir, tienen menos error a la hora de predecir el resultado de los datos entrenados.
   [44]

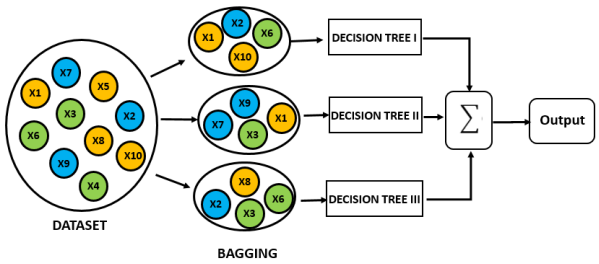

Figura 3.3: Procedimiento RandomForest.

Tomado en: https://bookdown.org/content/2031/ensambladores-random-forest-parte-i.html

La baja correlación entre los modelos (diferentes árboles entrenados), es lo que nos ayuda al éxito de esta metodología, ensamblar estos modelos permite mayor precisión que realizarlos individualmente, esto debido a que cada uno de los árboles cubre la debilidad de los otros, es decir, algunos árboles tendrán debilidades en el ajuste de algún tipo de datos o la falta de uso de una variable, pero otro puede estar prediciendo correctamente estos datos por lo que al agruparlos esas debilidades se cubren con las fortalezas teniendo así un modelo más preciso [54].

En cuanto a las variables, aleatoriamente se van escogiendo el subconjunto para cada árbol. Estas variables son las usadas para crear los nodos en cada uno de los árboles.

Una de las desventajas del random forest es la pérdida de la interpretabilidad del modelo. Sin embargo, se han desarrollado estrategias para cuantificar la importancia de los predictores: importancia por permutación e impureza de nodos [44].

Importancia por permutación:

1. Crear el conjunto de árboles que van a formar el modelo.
2. Calcular una métrica de error en cada árbol.
3. Para cada predictor (j) permutar los valores de esa variable y el resto dejarlo constante. Recalcular la métrica del error.
4. Calcular el incremento del error de la métrica para la variable j:

$$\% = \frac{error_j - error_{j-1}}{error_{j-1}}$$

[44]

Incremento de la pureza de nodos

Cuantifica el incremento total en la pureza de los nodos debido a divisiones en las que participa el predictor (promedio de todos los árboles).Se calcula en cada división de los árboles, el descenso conseguido en la medida empleada (índice Gini, error cuadrático medio, entropía, ...). Cuanto mayor sea este valor medio, mayor la contribución del predictor al modelo [44].

3.3.3. Máquinas de soporte vectorial

Las máquinas de soporte vectorial son clasificadores que se fundamentan en el Maximal Margin Classifier, que a su vez, se basa en el concepto de hiperplano. Fue una metodología presentada en COLT-92 por Boser, Guyon, Vapnik. En un espacio p-dimensional, un hiperplano se define como un subespacio plano y afín de dimensiones $p - 1$. La definición generalizada de un hiperplano es [43]:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p = 0$$

Todos los puntos que cumplen la ecuación pertenecen al hiperplano. Cuando x no cumple la ecuación, el punto x cae a un lado o al otro del hiperplano. Así pues, se puede entender que un hiperplano divide un

espacio p-dimensional en dos mitades [43].

Si pasamos esto a modelos estadísticos, se dispone de n observaciones, cada una con p predictores y una variable respuesta de dos niveles. Se emplean hiperplanos para construir el clasificador que permita predecir a qué grupo pertenece cada observación dependiendo los valores de sus predictores [43].

Suponiendo una clasificador perfectamente lineal, entonces el hiperplano de separación que cumple:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p < 0 - > y = 1$$
$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p > 0 - > y = -1$$

[43]

Ahora el problema a resolver es que existen infinitos hiperplanos, lo que hace necesario un método que permita seleccionar uno de ellos como clasificador óptimo. Para esto se selecciona el maximal margin hyperplane, que es aquel que se encuentra más alejado de todas las observaciones de entrenamiento. Esto se consigue por medio de métodos de optimización. En la mayoría de los casos es imposible construir un hiperplano perfecto , ya que usualmente los datos vienen con ruido lo que hace más difícil el problema, por lo que es preferible que clasifique bien la mayoría de los datos, para esto se construye un Support Vector Classifier [43].

Para la construcción del clasificador, se usa optimización convexa, en la cual un parámetro importante es el valor C, el cual controla el grado de severidad que se toleran. Cuanto más C se aproxime a cero, menos se penalizan los errores, es decir, C controla el balance entre sesgo y varianza. Este valor se estima mediante validación cruzada [43].

$$C = \frac{1}{\alpha}$$

Las máquinas de soporte vectorial son básicamente una generalización en un espacio de dimensión mayor de la idea del Support Vector Classifier. Para decidir qué dimensión es la correcta se usan los Kernel [43]. El papel que cumplen los kernel es generar una nueva dimensión que ayude a la separación de las clases. Entre muchos de los tipos de kernel tenemos: Kernel Lineal, Kernel Polinómico, Kernel Gaussiano, etc. Así pues, el hiperplano para separar las clases sería la siguiente función:

$$f(x, a, b) = \sum_{k \in SV}^{n} y_k a_k K(x_k, x) + b$$

[46] donde:

- $K(x_k, x)$: la función kernel a usar.
- a_k: constantes de ecuación lineal.
- y_k: valores observados.
- b: constante si b=0 es un hiperplano lineal.

3.4. VALIDACIÓN DE LOS MODELOS

3.4.1. Validación Cruzada

Los métodos de validación son estrategias que buscan estimar la capacidad de predicción de los modelos cuando se aplican a datos nuevos, calculando métricas que midan la bondad del modelo. Sabemos que la estimación del error tiende a ser variable dependiendo de los datos que se usan para entrenamiento y validación, debido a las posibles desviaciones que pueda haber por el reparto aleatorio de las observaciones [45]. Lo que se busca es un modelo estable que no cambie los resultados de precisión al cambiar los datos en los que se usa, más precisamente, que evite el sobreajuste.

Procedimiento validación cruzada k-fold reptida: El método de validación cruzada de k-fold está basado en la validación cruzada hold-out (Figura 3.4) y consiste básicamente en dividir los datos en k subconjuntos disjuntos. Entrenamos el modelo con (k-1) de los subconjuntos y se evalúa la precisión del modelo con el subconjunto que no se usó de entrenamiento. Este proceso se repite k veces cambiando los subconjuntos de

entrenamiento y el de validación en cada iteración.El error medio que se obtiene con los k análisis , permite evaluar la validez del modelo [36].

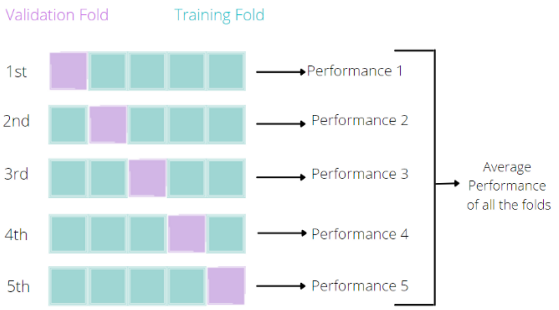

Figura 3.4: Ejemplo de Validación Cruzada con k=5.

Las ventajas de este método es una estimación más precisa del error , ya que equilibra el sesgo y la varianza, gracias a las diferentes muestras de entrenamiento que son distintas, lo que nos lleva a menor varianza al promediar las estimaciones de error [45].

Para encontrar los parámetros óptimos y el punto de corte óptimo se usa una validación cruzada de 10 grupos con 10 repeticiones, en cada una de las enfermedades partimos la base de datos original en 10 grupos de manera aleatoria y del mismo tamaño, en cada ejecución del modelo se usa un subgrupo como test para validar el modelo y el resto (k-1 grupos) para entrenar el modelo. Esto se hace para cada uno de los folds probando en cada una de estas ejecuciones un número diferente de parametros dependiendo el modelo que se este evaluando .Guardamos para cada grupo, el grupo test , las probabilidades para el modelo, y el número de componentes usado para cada modelo.

Para evaluar el corte óptimo para cada uno de los grupos y los componentes evaluamos puntos de corte de 0 a 1 con corte en cada 0.05 (0,0.05,0.1,0.15,etc..). Evaluamos los resultados por medio del Índice de Youden y escogemos aquel punto de corte y conjunto de parámetros que maximice esta medida.

Los parámetros a optimizar en cada modelo fueron:

- PLSDA: Número de componentes (1 a 10).
- SVM: Cost y gamma.
- RF: se dejan fijos los parametros de Pasolli y solo se optimiza el punto de corte óptimo.

Para la construcción del modelo final usando los parámetros óptimos se crean unos nuevos folds y 10 repeticiones con los que sacamos las medidas de ajuste (F1, Precisión, AUC) en cada repetición.

3.4.2. Medidas de error de clasificación

En un modelo de clasificación binaria (por ejemplo, casos y controles) se puede construir la matriz de confusión para comparar los valores predichos con los valores observados en la muestra.

		Observado	
		Casos	Controles
Predicho	Casos	VP	FP
	Controles	FN	VN

Tabla 3.2: Matriz de confusión

Los verdaderos positivos (VP) son aquellos valores que son observados como casos y el modelo predijo correctamente como caso, en el caso de VN son los controles predichos correctamente. En caso de los FP son aquellos predichos como casos pero son controles y los FN son aquellos que son Casos pero son predichos como controles.

En base a la matriz de confusión construida se han calculado las siguientes medidas:

1. Overall Accuracy: Porcentaje de observaciones predichas correctamente.

$$OA = \frac{VP + VN}{n}$$

   Donde n es el total de observaciones predichas (Datos validación).
2. La precisión: Número de Casos correctos dividido por le número de muestras predichas como casos.

$$Precision = \frac{VP}{VP + FP}$$

3. Recall: Número de casos correctos dividido por el número de casos que hay en la muestra.

$$Recall = \frac{VP}{VP + FN}$$

4. Score F1: Es la media armónica de la precisión y el recall.

$$F1 = \frac{2(Precision * recall)}{(Precision + recall)}$$

5. Índice de Youden: Su valor puede ser de -1 a 1, entre más cerca este el valor a 1 los resultados son muchos mejores. El valor máximo del índice es usado como criterio para seleccionar el corte óptimo. Se calcula como:

$$IY = Sensibilidad + Especificidad - 1$$

También hemos utilizado la **Curva ROC** (Figura 3.5), que es un gráfico obtenido a partir de la sensibilidad ($\frac{VP}{VP+FN}$) y especificidad ($\frac{VN}{FP+VN}$). Cambiando el punto de corte de la probabilidad predicha (valor desde el cual se decide que probabilidad va a representar a cada categoría), se calcula su respectiva sensibilidad y especificidad. La curva ROC lo que representa es la sensibilidad y especificidad en cada uno de los posibles puntos de corte, los ejes X y Y adoptan valores entre 0 y 1. Se representa una línea desde el punto de origen al punto $(1, 1)$, llamada la linea de no discriminación [10].

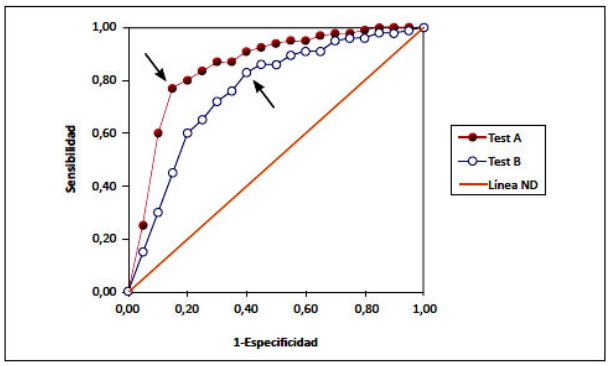

Figura 3.5: Curva ROC representación

Viendo la figura, podríamos interpretar la curva ROC como la ilustración de la proporción de verdaderos positivos (Eje Y) versus la proporción de falsos positivos (eje X) para cada punto de corte. Como se mencionó antes, se tiene una línea de referencia, que describe lo que sería la curva ROC de un modelo que es incapaz de discriminar controles y enfermos, debido a que para todo el recorrido de puntos de corte siempre se tiene la misma proporción de verdaderos positivos y falsos positivos. Lo que uno busca es que al variar el punto de corte se vaya trazando una curva ROC lo más lejana a esta línea de referencia [10].

Para determinar la capacidad discriminativa, es decir, la habilidad para distinguir personas sanas y enfermas, en base a la curva ROC, se estima el parámetro AUC, que es el área bajo la curva ROC del modelo. A partir de esto podemos ver qué tan bueno es el modelo para discriminar a lo largo de todos los puntos de corte posibles. A medida que el AUC se acerque al valor de 1 mayor es la capacidad discriminativa [10].

3.5. COMPARACIÓN DE MODELOS

3.5.1. ANOVA de medidas repetidas

Una vez obtenidos los valores del F1-score para cada uno de los modelos de clasificación y bases de datos con distintos pre-procesados, en cada repetición de la validación cruzada k-fold, se aplicó un modelo ANOVA para analizar si había diferencias significativas en la tasa de error de predicción media (medida con F1-score) para los distintos modelos (PLS-DA, RF y SVM), bases de datos o pre-procesado de las mismas (las cuatro combinaciones de filtrados S1 y S2 con normalizaciones N1 y N2). Dado que todos los modelos y pre-procesados se aplicaron sobre las mismas bases de datos, estas se consideraron un factor de bloqueo y se aplicó un modelo ANOVA de medidas repetidas con la función aov del paquete básico de R. Se incluyeron en dicho modelo tanto los efectos simples como las interacciones dobles.
Como post-hoc test para comparar dos a dos los niveles de los efectos estadísticamente significativos, se aplicó el método de las medias marginales estimadas (EMM) proporcionado por la librería emmeans [40].

3.5.2. Modelos lineales mixtos

Finalmente, otro de los objetivos de este trabajo era entender si las características propias de una base de datos de microbioma pueden influir o no en el desempeño de los modelos predictivos, en nuestro caso, modelos de clasificación para predecir enfermedad. Con este fin, se proponen y calculan los siguientes parámetros relacionados con características de datos de microbioma:

- Classes imbalance: Desequilibrio entre el tamaño muestral del grupo control y el grupo de enfermos. Calculado como el número de sujetos en el grupo minoritario dividido entre el número de sujetos en el grupo mayoritario. Por tanto, valores cercanos a 1 indican un buen equilibrio, mientras que valores mucho menores a 1 indican que los dos grupos tienen tamaños muy diferentes.
- Sample size: Tamaño muestral total, es decir, número de sujetos en la base de datos.
- Sparsity: Porcentaje de valores cero en la base de datos.

De nuevo, todos estos parámetros se miden sobre las mismas repeticiones (runs) de la validación cruzada de cada base de datos (BBDD), por lo que se decidió aplicar un modelo lineal mixto, en el que el efecto de la run+BBDD se consideró aleatorio, y el resto de efectos (modelo de clasificación, pre-procesado, imbalance, sample size y sparsity) se consideraron fijos. De nuevo, la variable respuesta fue el F1-score y se utilizó la librería lme4 para la estimación de los modelos.

Dado que solo se disponía de 6 bases de datos para este estudio y por tanto de poca variación en los valores de los parámetros propuestos, se decidió categorizar estas variables numéricas para eliminar ruido y mejorar la interpretación de las interacciones entre los efectos significativos. La categorización se realizó de forma que en cada nivel se dispusiera de 3 bases de datos, es decir, tomando la mediana. En concreto:

- Classes imbalance: Bajo (L) si era mayor que 0.75 y Alto (H) en caso contrario.
- Sample size: Grande (H) si más de 200 sujetos y Pequeño (L) en caso contrario.
- Sparsity: Alta (H) si tiene más del 78 % de ceros y Baja (L) en caso contrario.

CAPÍTULO 4

RESULTADOS

4.1. PRE-PROCESAMIENTO DE LOS DATOS

Al realizar un estudio descriptivo de las bases de datos encontramos que muchas bacterias tenían baja o nula abundancia en la mayoría de los casos, por lo cual se eliminarán.

En primer lugar, se trabajó a nivel de especie y se eliminan todas las bacterias con valor 0 en todos los pacientes. Se registró el número de variables restantes (p) y se realizó un estudio de prevalencia, que consistió en lo siguiente:

- Para cada especie y en cada condición (control/enfermedad) contar el % de individuos con abundancia mayor a 0 (Prevalencia casos , prevalencia controles)
- Sacar la prevalencia máxima y mínima de cada una de las especies.
- Representar un gráfico acumulado de la prevalencia mínima y el número de especies que superan ese número.

Como ejemplo presentemos los resultados de una de las enfermedades (Figura 4.1), las otras podemos verlas en el anexo y pre-procesamiento de los datos.

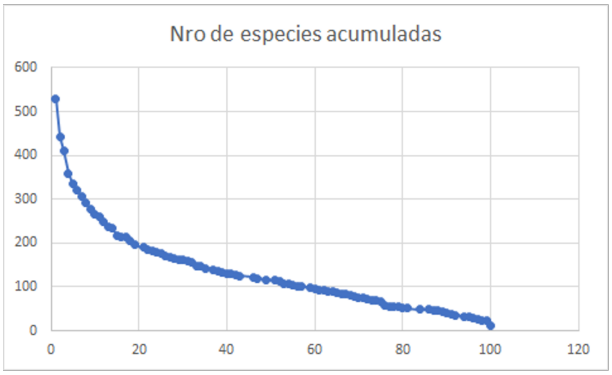

Figura 4.1: Número de especies acumuladas por prevalencia mínima para la Base de datos T2D

A partir de esto se fijaron diferentes niveles de prevalencia y el número de bacterias que conservaríamos en cada una de las bases de datos. La siguiente tabla muestra cuántas bacterias se estudiarían en cada una de las bases de datos para distintos niveles de prevalencia.

% Prevalencia	Cirrosis	Obesidad	Colorectal	IBD	T2D	WT2D
20	137	142	161	140	140	139
40	93	100	96	99	95	86
60	59	68	57	60	52	62

Tabla 4.1: Número de bacterias usando distintos niveles de prevalencia

Decidimos fijar una prevalencia mínima 20 % para no descartar demasiadas bacterias en el estudio (filtro S2), que se comparó con la opción de no aplicar ningún filtro de prevalencia (filtro S1).

Cada una de las bases de datos resultantes de aplicar los filtros S1 y S2 fueron normalizadas mediante los métodos TSS (N1) y CLR(N2), tal y como se describe en la sección de metodología.

4.2. ANÁLISIS DESCRIPTIVO

A continuación se utilizó PCA para explorar cada una de las matrices obtenidas al combinar los filtrados S1 y S2 con las normalizaciones N1 y N2.Consideramos los datos escalados y sin transformaciones adicionales a las normalizaciones propuestas.

Encontramos un valor anómalo en la base de datos de WT2D (Muestra 86) en el pre procesado S2-N1. Mostraremos a continuación las 4 gráficas de la T2 de hotelling , para observar el comportamiento de la observación anómala.

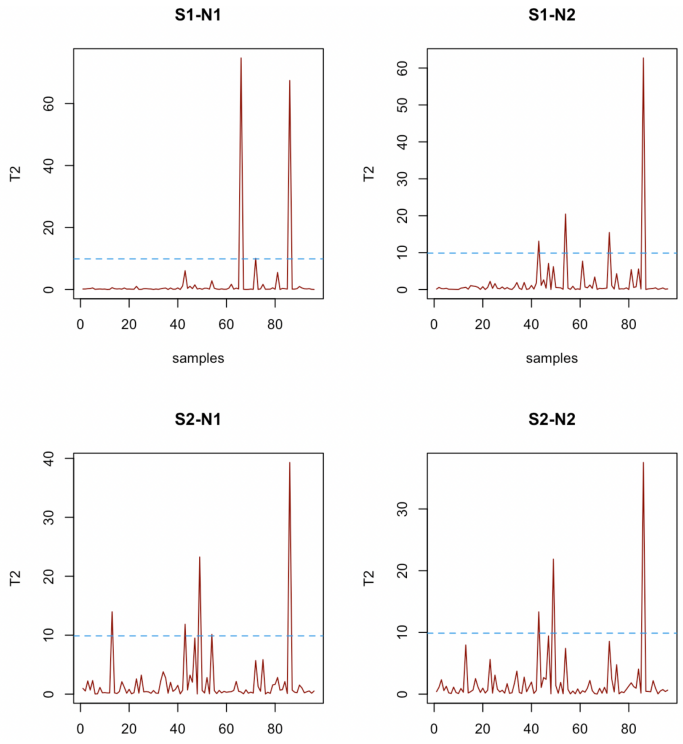

Figura 4.2: Gráficas T2 de Hotelling- Base de datos WT2D

Como podemos ver en las gráficas la muestra 86 es un anómalo muy severo que convendría eliminar. Se decide eliminar este individuo. Esto conlleva a que 2 bacterias tengan abundancias nulas en todos los individuos por lo que también las eliminamos, y rehacemos la normalización de los datos.

Luego exploramos las gráficas de scores de cada pre-procesado, para evaluar las diferencias entre controles y casos.

La figura a continuación muestra los resultados para cirrosis, con una separación aceptable entre casos y controles (aunque no perfecta) en la primera componente principal.

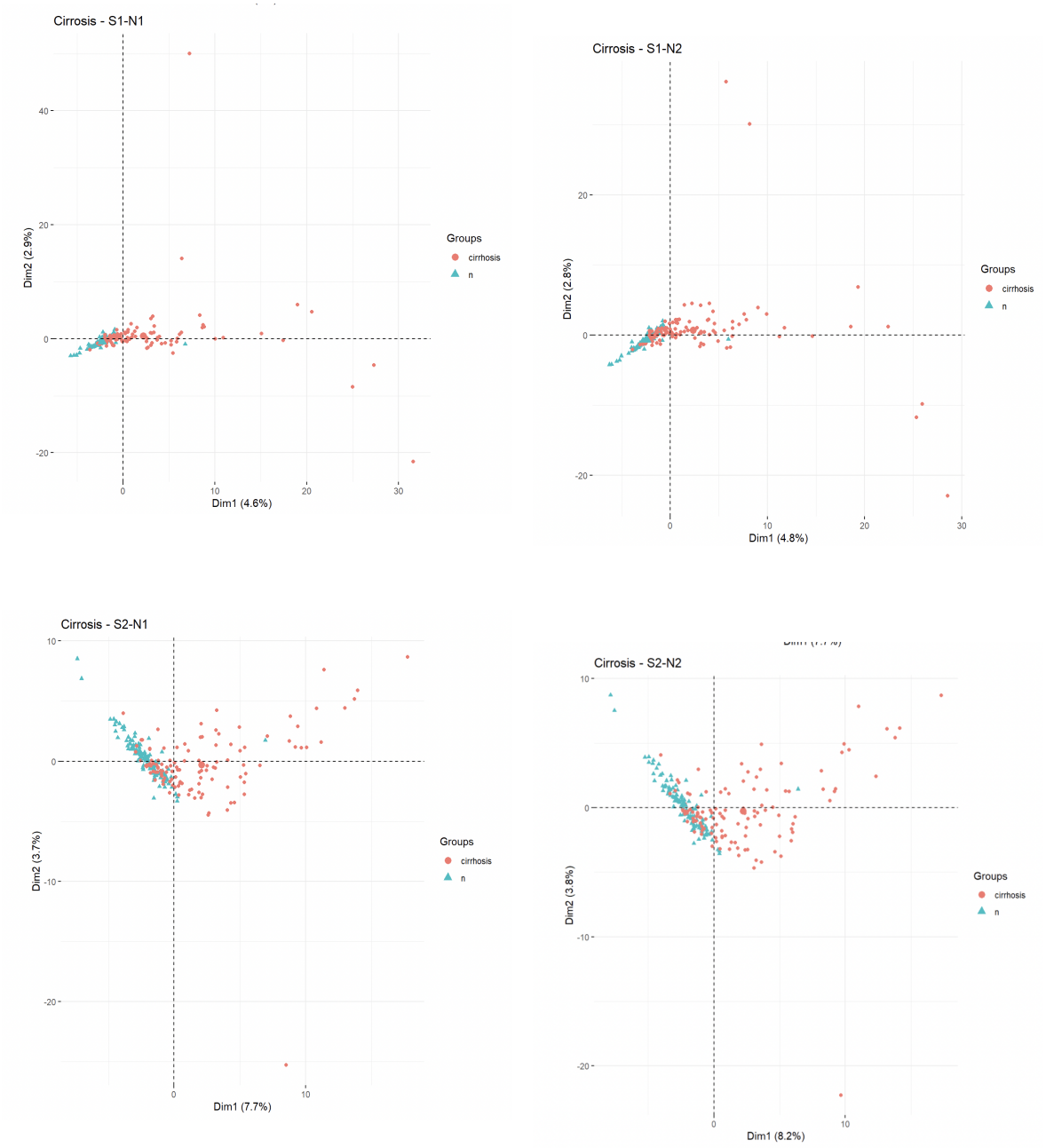

Figura 4.4: PCA en Cirrosis: Gráficas de scores para las dos primeras componentes principales

Para todas las enfermedades se hizo el mismo ejercicio y lo van a poder encontrar en el Anexo II.

4.3. MODELOS DE CLASIFICACIÓN

Para los tres métodos PLS-DA, SVM y Random Forest, los parámetros que se optimizaron en cada una de las técnicas fueron:

- PLS-DA: Número de componentes, de 1 a 10.
- SVM: $cost = (2^{-5}, 2^{-3}, 2, 2^3, 2^5, 2^7, 2^9, 2^{11}, 2^{13}, 2^{15})$ y $gamma = (2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2, 2^3)$.
- Random forest: ninguno ya que se usaron los mismos parámetros usados en Pasolli ntree= 500 y $mtry = \sqrt{p}$.

Además, en los tres modelos se optimizó el valor de corte de la probabilidad de clasificación.

Los resultados se pueden observar en la figura 4.5.

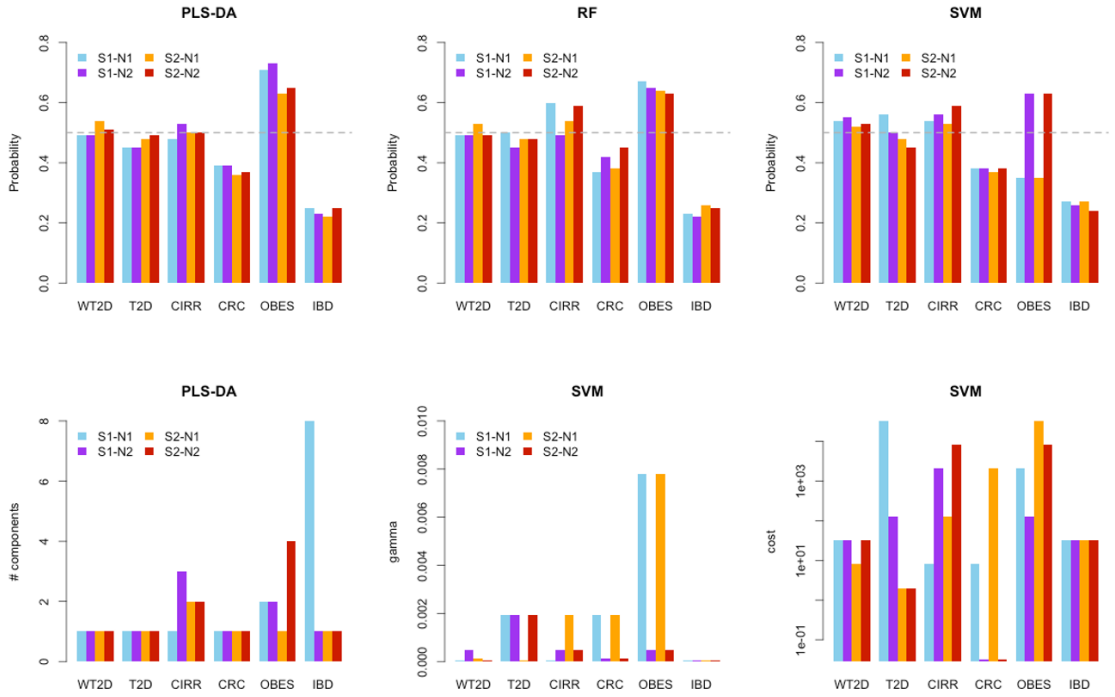

Figura 4.5: valores de los parámetros optimizados para cada modelo de clasificación, BBDD y tipo de pre-procesado.

Podemos observar que en cuanto al punto de corte óptimo de la probabilidad en las tres técnicas y en las enfermedades T2D, WTD y Cirrosis están alrededor del 0.5, pero si observamos las otras 3 enfermedades en Cáncer Colorectal (alrededor 0.4) y IBD (alrededor 0.2) tiene valores más bajos en las tres técnicas , mientras que Obesidad tiene unos valores muy grandes en las diferentes técnicas. Si nos centramos ahora en la variabilidad de los puntos de corte dentro de los diferentes pre-tratamientos (S1-N1, S1-N2, S2-N1, S2-N2) la obesidad en SVM y PLSDA es donde más variaciones vemos.

El número de componentes para PLS-DA en casi todas las enfermedades es igual a 1 a excepción de algunas donde al parecer por la propia variabilidad de los datos necesitan mayor número de componentes para explicar lo mismo que las variables originales.

Y finalmente el parámetro cost y gamma para la técnica SVM es muy variable dependiendo la enfermedad y el pre-tratamiento que se haya usado.

4.4. COMPARACIÓN DE MODELOS

Para comparar los modelos vamos a realizar un ANOVA de medidas repetidas para ver si existen diferencias significativas en la métrica F1 de los modelos dependiendo la técnica que se usa, de la BBDD o del pre-procesado. La tabla ANOVA se muestra a continuación:

Resultados ANOVA's			
Variables	**GL**	**F-valor**	**p-valor**
Base de datos	5	207.7	<2e-16
Residuales	24		
Técnica	1	145.09	1.16e-11
Técnica*Base de datos	5	19.07	1.16e-7
Residuales	24		
Pre-procesado	3	59.46	<2e-16
Pre-procesado*Base	15	15.69	<2e-16
Residuales	72		
Pre-procesado*Técnica	32	7.664	0.000133
Residuales	87		

Tabla 4.2: Resultados Anova's

Al tener un pvalor menor al 5 % se puede decir que hay diferencias significativas en el F1- score debidos tanto a la base de datos estudiada como la técnica estadística que se aplique dentro de esta. También concluimos que el efecto del pre-procesamiento que se realice es significativo y que esto va relacionado tanto con la enfermedad como con la técnica.

Se muestra a continuación en la figura 4.6 los intervalos de confianza para la media marginal estimada de F1-score de las diferentes técnicas:

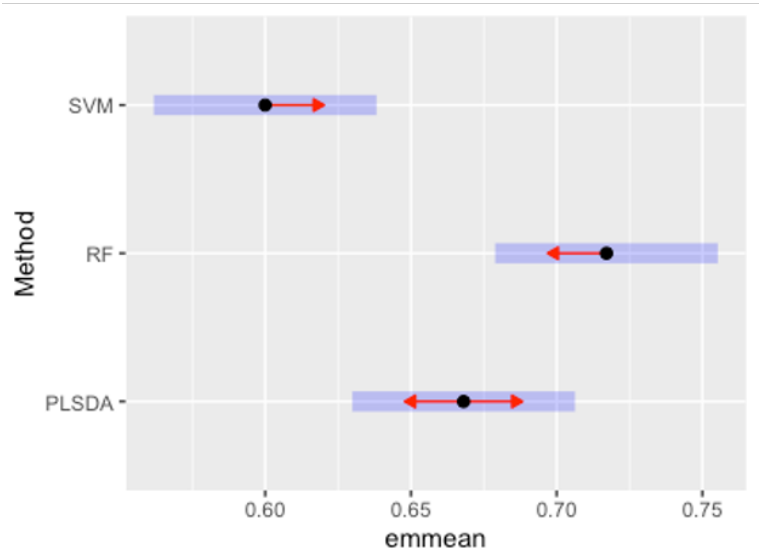

Figura 4.6: Medias marginales estimadas del F1-score para los modelos de clasificación comparados.

A partir de la Figura 4.5, podemos ver que, cuando las flechas rojas no se solapan, existen diferencias significativas entre las técnicas. Así pues, SVM es significativamente peor que las otras dos técnicas, mientras que RF es el mejor método.

Ahora queremos observar si esto cambia dependiendo la enfermedad que se está estudiando:

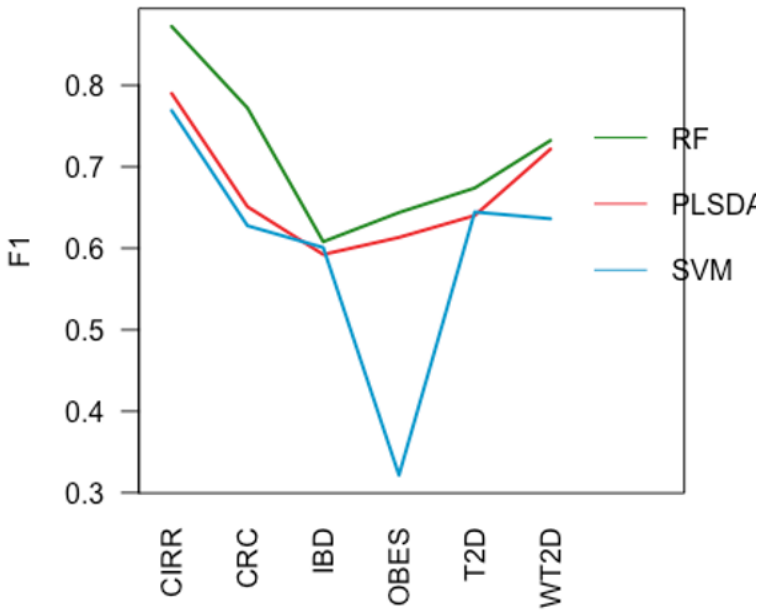

Figura 4.7: Gráfico de interacción para F1 por enfermedad dependiendo del modelo de clasificación

A la vista, de la figura 4.7, el modelo Random Forest es aquel que tiene mejores resultados en todas las enfermedades seguido de cerca por el PLSDA, el desempeño del modelo SVM es significativamente peor en la enfermedad Obesidad y WT2D. Ahora, al mirar por enfermedad, aquella donde no se han tenido buenos resultados en ninguna de las tres técnicas es IBD.

A continuación, se analizan en más detalle los dos mejores métodos, RF y PLSDA (Figura 4.8).

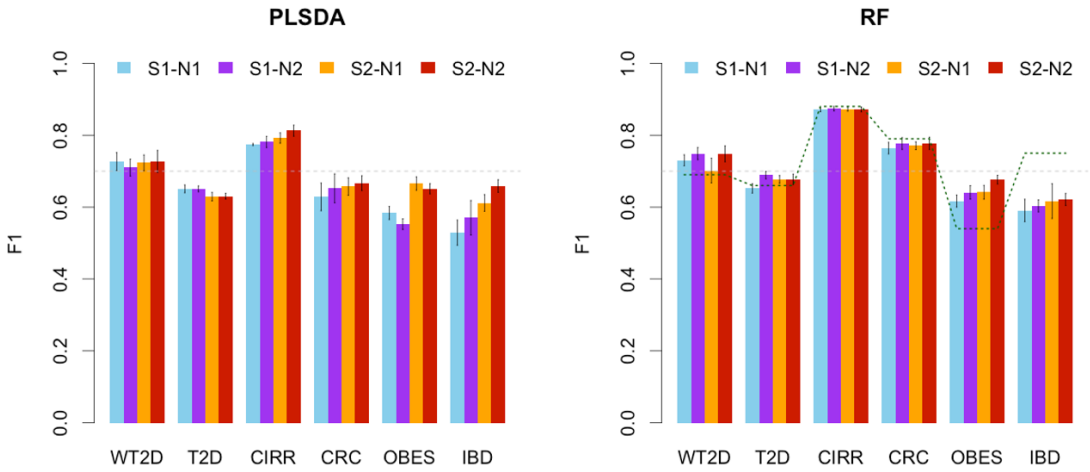

Figura 4.8: Gráfica F1 por enfermedad dependiendo del pre procesamiento

La altura de las barras representa el F1 medio en las distintas repeticiones de la CV, y las barras verticales son la desviación típica. La linea horizontal gris es el valor de la referencia 0.7 para F1, mientras que la línea punteada en RF es el valor obtenido por Pasolli [35].

Para el modelo PLSDA observamos para la enfermedad obesidad y quizas también para IBD una mejora

importante si se usa el filtrado donde se eliminan aquellas especies con baja abundancia (S2). Sin embargo, en el resto de enfermedades no se observa mucha diferencia en el F1 según el pre-procesado aplicado.

En cuanto al RF, la línea punteada explica el desempeño de Pasolli. Por lo tanto, primero podemos observar que con todas los pre tratamientos se consiguen mejores medidas que en Pasolli en la base de datos de obesidad y en general también en WT2D. Pero por el contrario para la base IBD empeora la métrica F1 con respecto a Pasolli. Ahora si nos basamos únicamente en los resultados de nuestros modelos no vemos un cambio realmente importante dentro de las enfermedades dependiendo del pre tratamiento que se hace debido a que las métricas son muy parecidas.

Vimos en el ANOVA que el pre procesado es importante y que su desempeño va a depender de la base de datos y de la técnica.

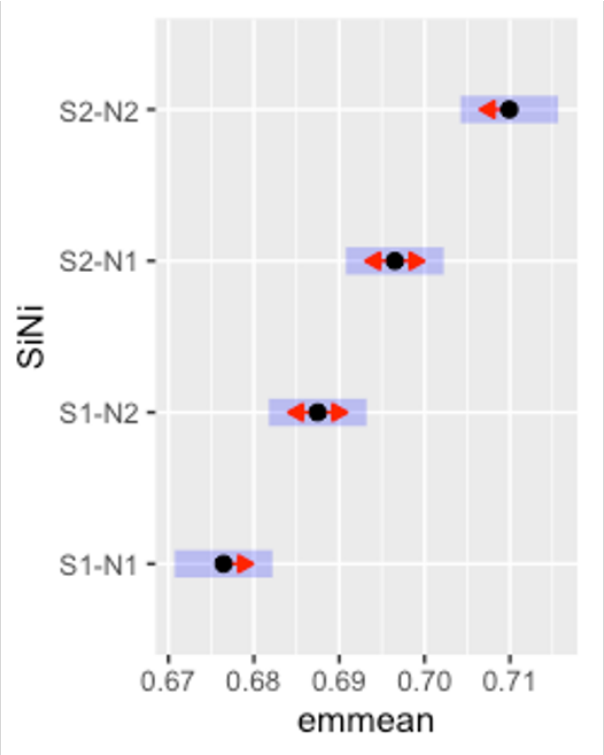

Figura 4.9: Medias marginales estimadas del F1-score para los pre-tratamientos

La mejor combinación de pre procesamiento es aquella donde se realiza el filtrado de las especies con abundancias bajas y la normalización CLR (Figura 4.9).

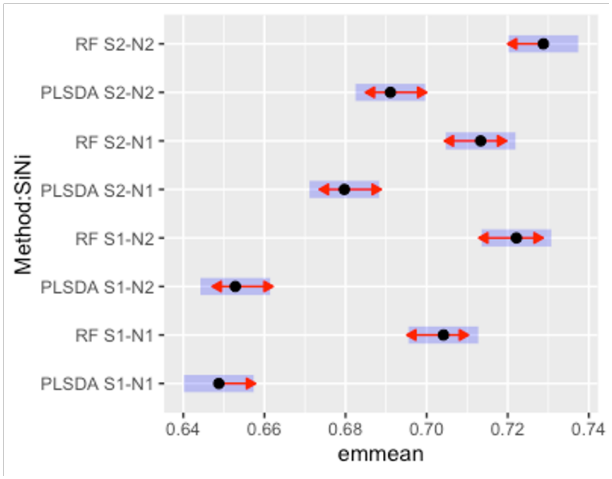

Figura 4.10: Medias marginales estimadas del F1-score para los modelos de clasificación comparados y los pre tratamientos.

A la vista de la figura 4.10, si nos centramos en los resultados únicamente para el RF las mejores métricas se consiguen con los pre procesamientos S2-N2 y S1-N2 lo que nos indica que la normalización CLR es la que mejores resultados obtiene y el filtrado tiene menor impacto. Por el contrario, si nos centramos en los resultados del PLSDA las mejores métricas se consiguen con los pre procesamientos S2-N2 y S2-N1 donde el filtrado de especies de menor abundancia es aquello que impacta más mientras que la normalización pareciese no afectar demasiado.

4.5. INFLUENCIA DE LAS CARACTERÍSTICAS DE LOS DATOS EN EL DESEMPEÑO DEL MODELO

Las bases de datos de cada enfermedad viene con características muy propias, el equilibrio en el tamaño muestral de los controles y los casos es muy diferente en cada una de las bases, el tamaño de la muestra y la presencia de ceros (*sparsity*) que hacen mucho más difícil el ajuste de los modelos. Por tal razón, se median estas tres métricas en las diferentes bases de datos para determinar después la influencia de estas en la correcta clasificación del modelo (Figura 4.11).

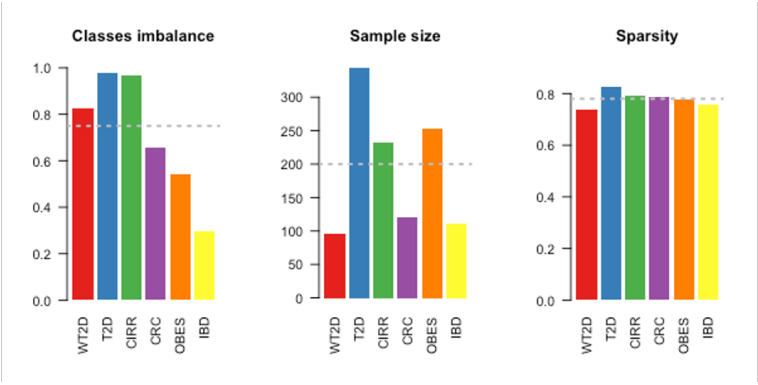

Figura 4.11: Métricas (Clases imbalance, sample size y sparsity) para cada base de datos

Se observa un equilibrio casi perfecto en las bases T2D y Cirrosis pero en cuanto a las bases CRC, Obesidad e IBD se nota un desequilibrio mayor. En cuanto al tamaño de la muestra las bases WT2D, CRC e IBD son las que menor número de observaciones tienen. En cuanto a la presencia de ceros vemos que en todas las bases encontrando una presencia mayor al 60 %.

Mediante un Modelo lineal mixto vamos a observar si estas características junto con interacciones con las técnicas y pre procesamientos impactan de alguna forma en los resultados de los modelos (F1).

Efecto	Grados de libertad	Valor F	P valor
Técnica	(1, 191)	249.6	$< 2{,}2e-16$
Imbalance	(1, 26)	10.93	0.0027
Pre procesamiento	(3, 191)	20.86	9.769e-12
Tamaño de la muestra	(1, 26)	0.77	0.385
Sparsity	(1, 26)	7.03	0.013
Técnica x imbalance	(1, 191)	25.40	1.073e-6
Técnica x preprocesamiento	(3 , 191)	7.07	0.0001553
Técnica x tamaño muestra	(1 , 191)	4.46	0.0358
Técnica x sparsity	(1 , 191)	126.10	$< 2{,}2e-16$
Imbalanceo x pre procesamiento	(3, 191)	12.44	1.811e-07
preprocesamiento x tamaño muestra	(3 , 191)	4.55	0.0041
preprocesamiento x sparsity	(3, 191)	5.36	0.0014

Tabla 4.3: Resultados modelo lineal

Vemos que hay muchas características con p-valores menores al 5 % por lo que consideramos que tienen un efecto en el F1 del modelo. Si vemos esto gráficamente para poder observar qué valores o en qué dirección es la influencia de cada una se obtiene la Figura 4.12:

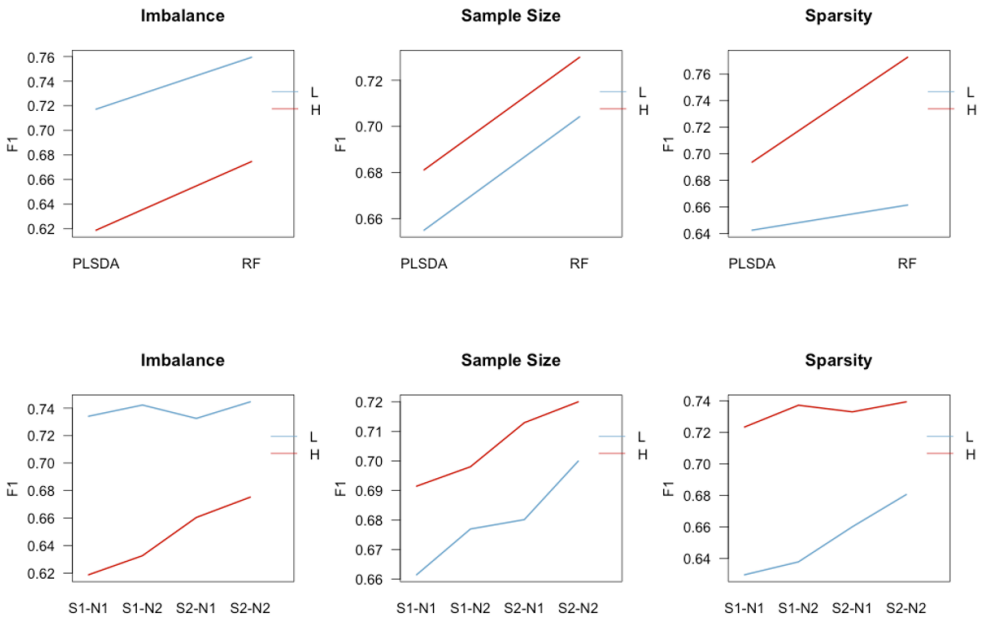

Figura 4.12: Gráficas de interacción entre las características de las BBDD y el modelo de clasificación (fila superior) o el pre-procesado (fila inferior).

Las características de los datos que dan mejores resultados son bases de datos con clases equilibradas, mayores tamaños de muestra y mayor grado de sparsity, este sobre todo en el modelo Random Forest. Si vemos las figura 4.12 se concluye que el efecto de normalizar en los datos equilibrados desaparece ya que no se ve algún cambio . En cuanto a los datos más desequilibrados sí es importante el preprocesamiento, al igual pasa con la *sparsity*. Cuando está alta, el efecto del preprocesamiento pareciera nulo pero si no se tiene una alta *sparsity* se podría decir que el pre procesamiento sí impacta en los resultados.

CAPÍTULO 5

CONCLUSIONES

5.1. CONCLUSIONES

En este Trabajo Final de Máster, se ha evaluado el impacto del pre-proceso de datos de microbioma y del modelo de clasificación utilizado para predecir enfermedad en el error de clasificación del modelo. Para llevar a cabo dicho estudio, se han utilizado 6 bases de datos públicas de microbioma ([35]) con pacientes de distintas enfermedades y sujetos sanos. Tras realizar el debido pre-procesado y limpieza de los datos, los análisis exploratorios y el correcto modelado de cada una de las variantes de las bases de datos generadas por las distintas estrategias de pre-procesado, podemos concluir:

- Se han procesado los datos originales para reducir los datos de cuantificación de la microbiota a nivel de especies de bacterias, mediante librerías de R diseñadas para este tipo específico de datos.
- Posteriormente, a cada base de datos, se le ha aplicado dos tipos de filtrado más o menos restrictivos para eliminar bacterias de baja abundancia. Sobre cada una de las matrices filtradas en cada base de datos (S1 y S2), se ha realizado una transformación de los datos para hacer comparables las muestras y tratar la naturaleza composicional de los datos: TSS (N1) y CLR (S2), por lo que de cada una de las 6 bases de datos originales, hemos obtenido 4 matrices de cuantificación diferentes sobre las que ajustar los modelos de clasificación. Se realizaron distintas exploraciones de los datos mediante PCA y se detectó un paciente anómalo en la base de datos de WT2D que fue eliminado para no sesgar los resultados de los modelos de clasificación.
- Se comparó el método de clasificación PLS-DA (poco utilizado en la literatura para el análisis de datos de microbioma) con modelos de aprendizaje automático más extendidos en este contexto como Random Forest y SVM. Para ello, se optimizaron los hiperparámetros de los modelos, incluyendo el valor de corte para la probabilidad de clasificación, mediante validación cruzada. En concreto, se aplicó k-fold repetido con 10 folds y varias repeticiones y se calculó el valor del F1-score como medida del error de clasificación.
- Se analizaron los resultados del error de clasificación de forma descriptiva y mediante un modelo ANOVA de medidas repetidas. Se observaron mejores resultados de discriminación entre sanos y enfermos con el modelo Random Forest, seguido de cerca por el PLSDA y, por último, a mayor distancia, por SVM. El efecto del pre-procesado en el error de clasificación (F1 score) es diferente según el modelo de clasificación aplicado pero, en general, se obtienen mejores resultados filtrando las bacterias de baja abundancia (S2) y normalizando los datos mediante CLR (N2).
- Por último, se evaluó el impacto de las características propias de las BBDD en los resultados del F1-score. Se tuvieron en cuenta tres parámetros: el desequilibrio entre los grupos caso y control en cuanto al número de sujetos en cada uno, el tamaño muestral total y el porcentaje de valores nulos. Se analizaron los resultados mediante modelos lineales mixtos y se observó que dichas características de la BBDD tienen un efecto significativo en el desempeño de los modelos y estrategias de pre-procesado, obteniéndose mejores resultados cuando la variable respuesta está equilibrada en casos y controles (a pesar de haber optimizado el corte de la probabilidad de clasificación), para tamaños muestrales más altos y cuando el porcentaje de ceros en la matriz de datos de microbioma es más elevado.

5.2. LÍNEAS FUTURAS DE INVESTIGACIÓN

- Integrar en los análisis variables como edad, sexo, raza, que puedan ayudar a tener un mejor desempeño de los modelos.

- Probar técnicas de selección de variables y generar la interpretación de las mismas.

BIBLIOGRAFÍA

[1] Acousticbiotech. (2020). «Impacto de la profundidad de la secuenciación en la caracterización del microbioma y la resistencia», dirección: https://spa.acousticbiotech.com/impact-sequencing-depth-characterization-microbiome-132195.

[2] D. P. S. Alarcón Cavero Tesera D
Aurie Giusseppe, M. R. del Campo y F. M. Manuel, «Procedimientos en microbiologıa clınica», *Recomendaciones de la Sociedad Española de Enfermedades Infecciosas y Microbiologıa Clınica*, 2016.

[3] A. Alberdi, «hilldiv: an R package for the integral analysis of diversity based on Hill numbers», *bioRxiv*, 2019. dirección: https://www.biorxiv.org/content/10.1101/545665v1.

[4] C. N. Arrieta, «Importancia de la microbiota para la salud», *El farmacéutico: profesión y cultura*, n.º 593, págs. 18-25, 2020.

[5] M. Badri y col., «Normalization methods for microbial abundance data strongly affect correlation estimates», *bioRxiv*, pág. 406 264, 2018.

[6] E. Bandera-Fernández y L. Pérez-Pelea, «Los modelos lineales generalizados mixtos. Su aplicación en el mejoramiento de plantas», *Cultivos tropicales*, vol. 39, n.º 1, págs. 127-133, 2018.

[7] D. Bates y col., «Fitting Linear Mixed-Effects Models Using lme4», *Journal of Statistical Software*, vol. 67, n.º 1, págs. 1-48, 2015. DOI: 10.18637/jss.v067.i01.

[8] L. Brenes-Guillén, «Proyecto Microbioma Humano», *Revista de Biologıa Tropical*, Blog-Blog, 2019.

[9] R. del Campo-Moreno y col., «Microbiota en la salud humana: técnicas de caracterización y transferencia», *Enfermedades Infecciosas y Microbiologıa Clınica*, vol. 36, n.º 4, págs. 241-245, 2018.

[10] J. Cerda y L. Cifuentes, «Uso de curvas ROC en investigación clınica: Aspectos teórico-prácticos», *Revista chilena de infectologıa*, vol. 29, n.º 2, págs. 138-141, 2012.

[11] R. M. G. Cervantes y G. B. R. Sánchez, «La microbiota»,

[12] J. Chong y col., «Using MicrobiomeAnalyst for comprehensive statistical, functional, and meta-analysis of microbiome data», *Nature Protocols*, vol. 15, n.º 3, págs. 799-821, 2020.

[13] J. Dagnino y col., «Análisis de varianza», *Revista chilena de anestesia*, vol. 43, n.º 4, págs. 306-310, 2014.

[14] ——, «Comparaciones múltiples», *Revista Chilena de Anastesia*, vol. 43, n.º 1, págs. 311-312, 2014.

[15] R. Ecoosfera. (2016). «El fascinante microbioma humano (qué es y por qué es importante que lo sepas).», dirección: https://ecoosfera.com/2014/12/el-fascinante-microbioma-humano-que-es-y-porque-es-importante-que-lo-sepas/.

[16] R. F y col., «mixOmics: An R package for 'omics feature selection and multiple data integration», *PLoS computational biology*, vol. 13, n.º 11, e1005752, 2017. dirección: http://www.mixOmics.org.

[17] L. Fontán Garcıa-Rodrigo, «Generalidades del microbioma humano y su relación con la obesidad», 2016.

[18] M. Fordellone, A. Bellincontro y F. Mencarelli, «Partial least squares discriminant analysis: A dimensionality reduction method to classify hyperspectral data», *arXiv preprint arXiv:1806.09347*, 2018.
[19] I. S. Garcıa, «TRABAJO FIN DE GRADO MICROBIOMA Y DESARROLLO FARMACÉUTICO»,
[20] F. Garrigues, *Genética Médica Blog*, 2017.
[21] A. Jemal y col., «Global cancer statistics», *CA: a cancer journal for clinicians*, vol. 61, n.º 2, págs. 69-90, 2011.
[22] F. H. Karlsson y col., «Gut metagenome in European women with normal, impaired and diabetic glucose control», *Nature*, vol. 498, n.º 7452, págs. 99-103, 2013.
[23] P. S. La Rosa y col., «Hypothesis testing and power calculations for taxonomic-based human microbiome data», *PloS one*, vol. 7, n.º 12, e52078, 2012.
[24] L. Lahti y S. Shetty, *microbiome R package*, 2012-2019.
[25] S. Lê, J. Josse y F. Husson, «FactoMineR: A Package for Multivariate Analysis», *Journal of Statistical Software*, vol. 25, n.º 1, págs. 1-18, 2008. DOI: 10.18637/jss.v025.i01.
[26] K.-A. Lê Cao, S. Boitard y P. Besse, «Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems», *BMC bioinformatics*, vol. 12, n.º 1, pág. 253, 2011.
[27] E. Le Chatelier y col., «Richness of human gut microbiome correlates with metabolic markers», *Nature*, vol. 500, n.º 7464, págs. 541-546, 2013.
[28] M. L. M. Lindemann y col., «Procedimientos en Microbiología Clínica.», *Sociedad Española de Enfermedades Infecciosas y Microbiología Clínica (SEIMC).*, 2016.
[29] I. López-Goñi, «Microbioma humano: universo en nuestro interior», *Revista de la*,
[30] M. J. Mantilla y R. G. T. Sáez, «Enfoque metagenómico para la caracterización del microbioma de aves corral. Revisión», *Revista Colombiana de Biotecnologıa*, vol. 21, n.º 2, págs. 77-97, 2019.
[31] P. J. McMurdie y S. Holmes, «phyloseq: An R package for reproducible interactive analysis and graphics of microbiome census data», *PLoS ONE*, vol. 8, n.º 4, e61217, 2013. dirección: http://dx.plos.org/10.1371/journal.pone.0061217.
[32] M. C. Moreno del Castillo, J. Valladares-Garcıa y J. Halabe-Cherem, «Microbioma humano», *Revista de la Facultad de Medicina (México)*, vol. 61, n.º 6, págs. 7-19, 2018.
[33] A. S. Moya, «Microbioma y secuenciación masiva», *Revista española de quimioterapia*, vol. 30, n.º 5, págs. 305-311, 2017.
[34] P. R. Murray, K. S. Rosenthal y M. A. Pfaller, *Microbiologıa médica*. Elsevier Health Sciences, 2017.
[35] E. Pasolli y col., «Machine learning meta-analysis of large metagenomic datasets: tools and biological insights», *PLoS computational biology*, vol. 12, n.º 7, e1004977, 2016.
[36] L. Pérez Planells y col., «Análisis de métodos de validación cruzada para la obtención robusta de parámetros biofısicos», *Revista Española de Teledetección, 2015, vol. 44, p. 55-65*, 2015.
[37] J. Qin y col., «A human gut microbial gene catalogue established by metagenomic sequencing», *nature*, vol. 464, n.º 7285, págs. 59-65, 2010.
[38] J. Qin y col., «A metagenome-wide association study of gut microbiota in type 2 diabetes», *Nature*, vol. 490, n.º 7418, págs. 55-60, 2012.
[39] N. Qin y col., «Alterations of the human gut microbiome in liver cirrhosis», *Nature*, vol. 513, n.º 7516, págs. 59-64, 2014.
[40] D. G. Quesada, J. S. Obando y A. V. Montero, «Impacto del desbalance en los tamaños de muestra por tratamiento sobre el desempeño de la prueba de comparaciones múltiples de Tukey», *SERENGUETI*, pág. 38, 2019.
[41] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022. dirección: https://www.R-project.org/.
[42] J. Rivera Pinto y col., «Statistical methods for the analysis of microbiome compositional data in HIV studies», Tesis doct., Universitat de Vic-Universitat Central de Catalunya, 2018.

[43] J. A. Rodrigo. (2017). «Máquinas de Vector Soporte (Support Vector Machines, SVMs)», dirección: https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines.

[44] ——, (2020). «Árboles de decisión, random forest, gradient boosting y C5.0», dirección: https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting#Random_Forest.

[45] ——, (2020). «Validación de modelos predictivos: Cross-validation, OneLeaveOut, Bootstraping», dirección: https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap#K-Fold_Cross-Validation.

[46] N. Sánchez Anzola, «Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario.», *ODEON-Observatorio de Economıa y Operaciones Numéricas*, n.º 9, 2015.

[47] K. Sankaran y S. P. Holmes, «Multitable methods for microbiome data integration», *Frontiers in genetics*, vol. 10, pág. 627, 2019.

[48] E. cooperation in science y technology, «Memorandum of Understanding for the implementation of the COST Action "Statistical and machine learning techniques in human microbiome studies" (ML4Microbiome) CA18131», *COST*, vol. 26, 2018.

[49] J. Uberos, «Microbiota perinatal: Revisión de su importancia en la salud del recién nacido», *Arch Argent Pediatr*, vol. 118, n.º 3, e265-70, 2020.

[50] P. Vermeesch, A. Ressentini y E. Garzanti, «An R package for statistical provenance analysis», *Sedimentary Geology*, vol. 336, págs. 14-25, 2016. dirección: https://doi.org/10.1016/j.sedgeo.2016.01.009.

[51] L. Waldron, «Data and statistical methods to analyze the human microbiome», *Msystems*, vol. 3, n.º 2, 2018.

[52] T. Wang, H. Zhao y col., «Structured subcomposition selection in regression and its application to microbiome data analysis», *Annals of Applied Statistics*, vol. 11, n.º 2, págs. 771-791, 2017.

[53] R. Yamada y col., «Interpretation of omics data analyses», *Journal of human genetics*, págs. 1-10, 2020.

[54] T. Yiu. (2019). «Understanding Random Forest How the Algorithm Works and Why it Is So Effective», dirección: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[55] G. Zeller y col., «Potential of fecal microbiota for early-stage detection of colorectal cancer», *Molecular systems biology*, vol. 10, n.º 11, pág. 766, 2014.

ANEXOS

ANEXO A

ANEXOS

A.0.1. ANEXO I. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de desarrollo sostenible	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.			X	
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

El presente TFM se relaciona de manera alta con el objetivo de desarrollo sostenible de salud y bienestar, esto debido a que se busca por medio de técnicas estadísticas y procesamiento de datos del microbioma humano entender el impacto del estado de la microbioma con el desarrollo o predisposición de tener enfermedades y así dar herramientas para la lucha contra las enfermedades y causas de muerte.

Además por medio de este trabajo, queremos sentar la base para mejorar el análisis del microbioma por medio de técnicas de machine learning para dar la base a futuras investigaciones que puedan estudiar el como la desigualdad social en los países y entre países, género, estratos socio-económicos, entre otros, también impacta la microbioma y por lo tanto el efecto de estas características genera mayores tasas de enfermos en algunas trastornos e incluso tasas de mortalidad.

A.0.2. ANEXO II. PRE PROCESAMIENTO CÓDIGO

Lectura y Construcción de datos

Se presenta el código, paso a paso para la construcción (Filtros y normalización) de la base de datos Cirrosis, esto mismo se hizo con todas las bases de datos.//

```
library(readxl)
library(phyloseq)
library(microbiome)
library(ggplot2)
library(hilldiv)
library(mixOmics)
library(FactoMineR)
library(factoextra)
library(provenance)
library(forcats)
library(ggpubr)
library(gridExtra)
library(knitr)
##############################################################################################
#####################   CIRROSIS ##############################################################
Taxa_cirrosis <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                                  sheet = "Taxa_cirrosis")

OTU_cirrosis <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                           sheet = "OTU_cirrosis")
Sample_cirrosis <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                          sheet = "Sample_cirrosis")

otu_names=OTU_cirrosis$...1
OTU_cirrosis=OTU_cirrosis[,-1]
row.names(OTU_cirrosis) <- otu_names

OTU_cirrosis <- as.matrix(OTU_cirrosis)

tax_names=Taxa_cirrosis $...1
Taxa_cirrosis =Taxa_cirrosis [,-1]
row.names(Taxa_cirrosis )<-tax_names

Taxa_cirrosis  <- as.matrix(Taxa_cirrosis)

OTU_CIRROSIS = otu_table(OTU_cirrosis, taxa_are_rows = TRUE)
TAX_CIRROSIS = tax_table(Taxa_cirrosis)

SAMPLES_NAMES=Sample_cirrosis$...1
Sample_cirrosis=Sample_cirrosis[,-1]
row.names(Sample_cirrosis) <- SAMPLES_NAMES
Sample_cirrosis = sample_data(Sample_cirrosis)
sample_names(Sample_cirrosis)<-colnames(OTU_CIRROSIS)

cirrosis <- phyloseq(OTU_CIRROSIS, TAX_CIRROSIS, Sample_cirrosis)
cirrosis_species <- tax_glom(cirrosis, taxrank="Species")


Casos=subset_samples(cirrosis_species, disease=="cirrhosis")
Controles=subset_samples(cirrosis_species, disease=="n")
pr_casos <- as.data.frame(prevalence(Casos, detection=0, sort=TRUE, count=FALSE))
pr_controles=prevalence(Controles, detection=0, sort=TRUE, count=FALSE)
```

```
pr_controles=as.data.frame(pr_controles)
Prevalencia=as.data.frame(cbind(pr_casos, pr_controles[, "pr_controles"]
[match(rownames(pr_casos), rownames(pr_controles))]))
names(Prevalencia)[names(Prevalencia) == "pr_casos"] <- "Prevalencia_casos"
names(Prevalencia)[names(Prevalencia) == "pr_controles[, \"pr_controles\"]
[match(rownames(pr_casos), rownames(pr_controles))]"] <- "Prevalencia_controles"
Prevalencia$Prevalencia_max <-round(apply(Prevalencia, 1, max),2)*100
Prevalencia$Prevalencia_min <-round(apply(Prevalencia, 1, min),2)*100


keepTaxa = rownames(Prevalencia)[(Prevalencia$Prevalencia_max >= 20)]
Muestra_20_cirrosis=prune_taxa(keepTaxa, cirrosis_species)


#NORMALIZACIÓN
S1_N1_CIRROSIS=tss(cirrosis_species@otu_table)
S2_N1_CIRROSIS=tss(Muestra_20_cirrosis@otu_table)
S1_N2_CIRROSIS=CLR(cirrosis_species@otu_table+1)
S2_N2_CIRROSIS=CLR(Muestra_20_cirrosis@otu_table+1)



S1_NORMAL_cIRR=list(S1_TSS_CIRR=S1_N1_CIRROSIS,S1_CLR_CIRR=S1_N2_CIRROSIS)
S2_NORMAL_cIRR=list(S2_TSS_CIRR=S2_N1_CIRROSIS,S2_CLR_CIRR=S2_N2_CIRROSIS)
Filtro_CIRR=list(S1_Pasolli_CIRR=cirrosis_species@otu_table,S2_20_CIRR=Muestra_20_cirrosis@otu_
Base_total_cirrosis=list(Base=cirrosis@otu_table)
Cirrosis_lista=list(Original=Base_total_cirrosis,Filtros_cirr=Filtro_CIRR,
S1_NORMAL=S1_NORMAL_cIRR,S2_NORMAL=S2_NORMAL_cIRR)

TaxaCirrosis=list(cirrosis_species@tax_table,Muestra_20_cirrosis@tax_table)
‘‘‘

‘‘‘{r resto,echo = FALSE,warning=FALSE,message=FALSE}
#############################################################################################
################################### Cancer Colorectal ########################################
Taxa_Cancer_Colorectal <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                        sheet = "Taxa_Cancer Colorectal")
OTU_Cancer_Colorectal <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                                    sheet = "OTU_Cancer Colorectal")
Sample_Cancer_Colorectal <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                                   sheet = "Sample_Cancer Colorectal")

otu_names=OTU_Cancer_Colorectal$...1
OTU_Cancer_Colorectal=OTU_Cancer_Colorectal[,-1]
row.names(OTU_Cancer_Colorectal) <- otu_names

OTU_Cancer_Colorectal <- as.matrix(OTU_Cancer_Colorectal)

tax_names=Taxa_Cancer_Colorectal$...1
Taxa_Cancer_Colorectal =Taxa_Cancer_Colorectal[,-1]
row.names(Taxa_Cancer_Colorectal)<-tax_names

Taxa_Cancer_Colorectal<- as.matrix(Taxa_Cancer_Colorectal)

OTU_Cancer_Colorectal = otu_table(OTU_Cancer_Colorectal, taxa_are_rows = TRUE)
Taxa_Cancer_Colorectal = tax_table(Taxa_Cancer_Colorectal)

SAMPLES_NAMES=Sample_Cancer_Colorectal$...1
```

```
Sample_Cancer_Colorectal=Sample_Cancer_Colorectal[,-1]
row.names(Sample_Cancer_Colorectal) <- SAMPLES_NAMES
Sample_Cancer_Colorectal = sample_data(Sample_Cancer_Colorectal)
sample_names(Sample_Cancer_Colorectal)<-colnames(OTU_Cancer_Colorectal)

Cancer_Colorectal<- phyloseq(OTU_Cancer_Colorectal, Taxa_Cancer_Colorectal, Sample_Cancer_Colorectal)
Cancer_Colorectal_species <- tax_glom(Cancer_Colorectal, taxrank="Species")

Casos=subset_samples(Cancer_Colorectal_species, group=="crc")
Controles=subset_samples(Cancer_Colorectal_species, group=="control")
pr_casos <- as.data.frame(prevalence(Casos, detection=0, sort=TRUE, count=FALSE))
pr_controles=prevalence(Controles, detection=0, sort=TRUE, count=FALSE)
pr_controles=as.data.frame(pr_controles)
Prevalencia=as.data.frame(cbind(pr_casos, pr_controles[, "pr_controles"]
[match(rownames(pr_casos), rownames(pr_controles))]))
names(Prevalencia)[names(Prevalencia) == "pr_casos"] <- "Prevalencia_casos"
names(Prevalencia)[names(Prevalencia) == "pr_controles[, \"pr_controles\"]
[match(rownames(pr_casos), rownames(pr_controles))]"] <- "Prevalencia_controles"
Prevalencia$Prevalencia_max <-round(apply(Prevalencia, 1, max),2)*100
Prevalencia$Prevalencia_min <-round(apply(Prevalencia, 1, min),2)*100


keepTaxa = rownames(Prevalencia)[(Prevalencia$Prevalencia_max >= 20)]
Muestra_20_CancerColorectal=prune_taxa(keepTaxa, Cancer_Colorectal_species)

#NORMALIZACIÓN
S1_N1_CancerColorectal=tss(Cancer_Colorectal_species@otu_table)
S2_N1_CancerColorectal=tss(Muestra_20_CancerColorectal@otu_table)
S1_N2_CancerColorectal=CLR(Cancer_Colorectal_species@otu_table+1)
S2_N2_CancerColorectal=CLR(Muestra_20_CancerColorectal@otu_table+1)

S1_NORMAL_CANCERCOL=list(S1_TSS_CANCERCOL=S1_N1_CancerColorectal,
S1_CLR_CANCERCOL=S1_N2_CancerColorectal)
S2_NORMAL_CANCERCOL=list(S2_TSS_CANCERCOL=S2_N1_CancerColorectal,
S2_CLR_CANCERCOL=S2_N2_CancerColorectal)
Filtro_CANCERCOL=list(S1_Pasolli_CANCERCOL=Cancer_Colorectal_species@otu_table,
S2_20_CANCERCOL=Muestra_20_CancerColorectal@otu_table)
Base_total_CANCERCOL=list(Base=Cancer_Colorectal@otu_table)
CANCERCOL_lista=list(Original=Base_total_CANCERCOL,Filtros_CANCERCOL=Filtro_CANCERCOL,
S1_NORMAL=S1_NORMAL_CANCERCOL,S2_NORMAL=S2_NORMAL_CANCERCOL)

##############################################################################################
############################### IBD ##########################################################
Taxa_IBD <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                                    sheet = "Taxa_IBD")
OTU_IBD <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                      sheet = "OTU_IBD")
Sample_IBD <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                      sheet = "Sample_IBD")

otu_names=OTU_IBD$...1
OTU_IBD =OTU_IBD [,-1]
row.names(OTU_IBD ) <- otu_names

OTU_IBD  <- as.matrix(OTU_IBD)

tax_names=Taxa_IBD$...1
Taxa_IBD =Taxa_IBD[,-1]
```

```
row.names(Taxa_IBD)<-tax_names

Taxa_IBD<- as.matrix(Taxa_IBD)

OTU_IBD= otu_table(OTU_IBD, taxa_are_rows = TRUE)
Taxa_IBD= tax_table(Taxa_IBD)

SAMPLES_NAMES=Sample_IBD $...1
Sample_IBD =Sample_IBD [,-1]
row.names(Sample_IBD ) <- SAMPLES_NAMES
Sample_IBD = sample_data(Sample_IBD )
sample_names(Sample_IBD )<-colnames(OTU_IBD)

IBD<- phyloseq(OTU_IBD, Taxa_IBD, Sample_IBD)
IBD_species <- tax_glom(IBD, taxrank="Species")

Casos=subset_samples(IBD_species, ibd=="y")
Controles=subset_samples(IBD_species, ibd=="n")
pr_casos <- as.data.frame(prevalence(Casos, detection=0, sort=TRUE, count=FALSE))
pr_controles=prevalence(Controles, detection=0, sort=TRUE, count=FALSE)
pr_controles=as.data.frame(pr_controles)
Prevalencia=as.data.frame(cbind(pr_casos, pr_controles[, "pr_controles"]
[match(rownames(pr_casos), rownames(pr_controles))]))
names(Prevalencia)[names(Prevalencia) == "pr_casos"] <- "Prevalencia_casos"
names(Prevalencia)[names(Prevalencia) == "pr_controles[, \"pr_controles\"]
[match(rownames(pr_casos), rownames(pr_controles))]"] <- "Prevalencia_controles"
Prevalencia$Prevalencia_max <-round(apply(Prevalencia, 1, max),2)*100
Prevalencia$Prevalencia_min <-round(apply(Prevalencia, 1, min),2)*100


keepTaxa = rownames(Prevalencia)[(Prevalencia$Prevalencia_max >= 20)]
Muestra_20_IBD=prune_taxa(keepTaxa, IBD_species)


#NORMALIZACIÓN
S1_N1_IBD=tss(IBD_species@otu_table)
S2_N1_IBD=tss(Muestra_20_IBD@otu_table)
S1_N2_IBD=CLR(IBD_species@otu_table+1)
S2_N2_IBD=CLR(Muestra_20_IBD@otu_table+1)


S1_NORMAL_IBD=list(S1_TSS_IBD=S1_N1_IBD,S1_CLR_IBD=S1_N2_IBD)
S2_NORMAL_IBD=list(S2_TSS_IBD=S2_N1_IBD,S2_CLR_IBD=S2_N2_IBD)
Filtro_IBD=list(S1_Pasolli_IBD=IBD_species@otu_table,
S2_20_IBD=Muestra_20_IBD@otu_table)
Base_total_IBD=list(Base=IBD@otu_table)
IBD_lista=list(Original=Base_total_IBD,Filtros_IBD=Filtro_IBD,
S1_NORMAL=S1_NORMAL_IBD,S2_NORMAL=S2_NORMAL_IBD)

##############################################################################################
############################## Obesidad ######################################################
Taxa_Obesidad <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                        sheet = "Taxa_Obesidad")
OTU_Obesidad <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                           sheet = "OTU_Obesidad")
Sample_Obesidad <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                          sheet = "Sample_Obesidad")
```

```
otu_names=OTU_Obesidad $...1
OTU_Obesidad =OTU_Obesidad [,-1]
row.names(OTU_Obesidad ) <- otu_names

OTU_Obesidad  <- as.matrix(OTU_Obesidad )

tax_names=Taxa_Obesidad$...1
Taxa_Obesidad=Taxa_Obesidad[,-1]
row.names(Taxa_Obesidad)<-tax_names

Taxa_Obesidad<- as.matrix(Taxa_Obesidad)

OTU_Obesidad= otu_table(OTU_Obesidad, taxa_are_rows = TRUE)
Taxa_Obesidad= tax_table(Taxa_Obesidad)

SAMPLES_NAMES=Sample_Obesidad $...1
Sample_Obesidad =Sample_Obesidad[,-1]
row.names(Sample_Obesidad) <- SAMPLES_NAMES
Sample_Obesidad= sample_data(Sample_Obesidad)

sample_names(Sample_Obesidad)<-colnames(OTU_Obesidad)

Obesidad<- phyloseq(OTU_Obesidad, Taxa_Obesidad, Sample_Obesidad)
Obesidad_species <- tax_glom(Obesidad, taxrank="Species")


Casos=subset_samples(Obesidad_species, disease=="obesity")
Controles=subset_samples(Obesidad_species, disease=="leaness")
pr_casos <- as.data.frame(prevalence(Casos, detection=0, sort=TRUE, count=FALSE))
pr_controles=prevalence(Controles, detection=0, sort=TRUE, count=FALSE)
pr_controles=as.data.frame(pr_controles)
Prevalencia=as.data.frame(cbind(pr_casos, pr_controles[, "pr_controles"]
[match(rownames(pr_casos), rownames(pr_controles))]))
names(Prevalencia)[names(Prevalencia) == "pr_casos"] <- "Prevalencia_casos"
names(Prevalencia)[names(Prevalencia) == "pr_controles[, \"pr_controles\"]
[match(rownames(pr_casos), rownames(pr_controles))]"] <- "Prevalencia_controles"
Prevalencia$Prevalencia_max <-round(apply(Prevalencia, 1, max),2)*100
Prevalencia$Prevalencia_min <-round(apply(Prevalencia, 1, min),2)*100

keepTaxa = rownames(Prevalencia)[(Prevalencia$Prevalencia_max >= 20)]
Muestra_20_Obesidad=prune_taxa(keepTaxa, Obesidad_species)

#NORMALIZACIÓN
S1_N1_Obesdidad=tss(Obesidad_species@otu_table)
S2_N1_Obesdidad=tss(Muestra_20_Obesidad@otu_table)
S1_N2_Obesdidad=CLR(Obesidad_species@otu_table+1)
S2_N2_Obesdidad=CLR(Muestra_20_Obesidad@otu_table+1)

S1_NORMAL_Obesdidad=list(S1_TSS_Obesdidad=S1_N1_Obesdidad,
S1_CLR_Obesdidad=S1_N2_Obesdidad)
S2_NORMAL_Obesdidad=list(S2_TSS_Obesdidad=S2_N1_Obesdidad,
S2_CLR_Obesdidad=S2_N2_Obesdidad)
Filtro_Obesdidad=list(S1_Pasolli_Obesdidad=Obesidad_species@otu_table,
S2_20_Obesdidad=Muestra_20_Obesidad@otu_table)
Base_total_Obesdidad=list(Base=Obesidad@otu_table)
Obesidad_lista=list(Original=Base_total_Obesdidad,
Filtros_Obesdidad=Filtro_Obesdidad,S1_NORMAL=S1_NORMAL_Obesdidad,S2_NORMAL=S2_NORMAL_Obesdidad)
```

```
#############################################################################################
################################### T2D #####################################################
Taxa_T2D <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                       sheet = "Taxa_T2D")
OTU_T2D <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                      sheet = "OTU_T2D")
Sample_T2D <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                     sheet = "Sample_T2D")

otu_names=OTU_T2D$...1
OTU_T2D=OTU_T2D[,-1]
row.names(OTU_T2D) <- otu_names

OTU_T2D<- as.matrix(OTU_T2D)

tax_names=Taxa_T2D$...1
Taxa_T2D=Taxa_T2D[,-1]
row.names(Taxa_T2D)<-tax_names

Taxa_T2D<- as.matrix(Taxa_T2D)

OTU_T2D= otu_table(OTU_T2D, taxa_are_rows = TRUE)
Taxa_T2D= tax_table(Taxa_T2D)

SAMPLES_NAMES=Sample_T2D$...1
Sample_T2D=Sample_T2D[,-1]
row.names(Sample_T2D) <- SAMPLES_NAMES
Sample_T2D= sample_data(Sample_T2D)

sample_names(Sample_T2D)<-colnames(OTU_T2D)

T2D<- phyloseq(OTU_T2D,Taxa_T2D,Sample_T2D)
T2D_species <- tax_glom(T2D, taxrank="Species")

Casos=subset_samples(T2D_species, disease=="t2d")
Controles=subset_samples(T2D_species, disease=="n")
pr_casos <- as.data.frame(prevalence(Casos, detection=0, sort=TRUE, count=FALSE))
pr_controles=prevalence(Controles, detection=0, sort=TRUE, count=FALSE)
pr_controles=as.data.frame(pr_controles)
Prevalencia=as.data.frame(cbind(pr_casos, pr_controles[, "pr_controles"]
[match(rownames(pr_casos), rownames(pr_controles))]))
names(Prevalencia)[names(Prevalencia) == "pr_casos"] <- "Prevalencia_casos"
names(Prevalencia)[names(Prevalencia) == "pr_controles[, \"pr_controles\"]
[match(rownames(pr_casos), rownames(pr_controles))]"] <- "Prevalencia_controles"
Prevalencia$Prevalencia_max <-round(apply(Prevalencia, 1, max),2)*100
Prevalencia$Prevalencia_min <-round(apply(Prevalencia, 1, min),2)*100

keepTaxa = rownames(Prevalencia)[(Prevalencia$Prevalencia_max >= 20)]
Muestra_20_T2D=prune_taxa(keepTaxa, T2D_species)


#NORMALIZACIÓN
S1_N1_T2D=tss(T2D_species@otu_table)
S2_N1_T2D=tss(Muestra_20_T2D@otu_table)
S1_N2_T2D=CLR(T2D_species@otu_table+1)
S2_N2_T2D=CLR(Muestra_20_T2D@otu_table+1)
```

```
S1_NORMAL_T2D=list(S1_TSS_T2D=S1_N1_T2D,S1_CLR_T2D=S1_N2_T2D)
S2_NORMAL_T2D=list(S2_TSS_T2D=S2_N1_T2D,S2_CLR_T2D=S2_N2_T2D)
Filtro_T2D=list(S1_Pasolli_T2D=T2D_species@otu_table,
S2_20_T2D=Muestra_20_T2D@otu_table)
Base_total_T2D=list(Base=T2D@otu_table)
T2D_lista=list(Original=Base_total_T2D,Filtros_T2D=Filtro_T2D,
S1_NORMAL=S1_NORMAL_T2D,S2_NORMAL=S2_NORMAL_T2D)



##############################################################################################
#################################### WT2D ####################################################
Taxa_WT2D <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                        sheet = "Taxa_WT2D")
OTU_WT2D <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                         sheet = "OTU_WT2D")
Sample_WT2D <- read_excel("EJEMPLO PHYLOSEQ.xlsx",
                        sheet = "Sample_WT2D")
otu_names=OTU_WT2D$...1
OTU_WT2D=OTU_WT2D[,-1]
row.names(OTU_WT2D) <- otu_names

OTU_WT2D<- as.matrix(OTU_WT2D)

tax_names=Taxa_WT2D$...1
Taxa_WT2D=Taxa_WT2D[,-1]
row.names(Taxa_WT2D)<-tax_names

Taxa_WT2D<- as.matrix(Taxa_WT2D)

OTU_WT2D= otu_table(OTU_WT2D, taxa_are_rows = TRUE)
Taxa_WT2D= tax_table(Taxa_WT2D)

SAMPLES_NAMES=Sample_WT2D$...1
Sample_WT2D=Sample_WT2D[,-1]
row.names(Sample_WT2D) <- SAMPLES_NAMES
Sample_WT2D= sample_data(Sample_WT2D)

sample_names(Sample_WT2D)<-colnames(OTU_WT2D)

WT2D<- phyloseq(OTU_WT2D,Taxa_WT2D,Sample_WT2D)
WT2D_species <- tax_glom(WT2D, taxrank="Species")

Casos=subset_samples(WT2D_species, disease=="t2d")
Controles=subset_samples(WT2D_species, disease=="n")
pr_casos <- as.data.frame(prevalence(Casos, detection=0, sort=TRUE, count=FALSE))
pr_controles=prevalence(Controles, detection=0, sort=TRUE, count=FALSE)
pr_controles=as.data.frame(pr_controles)
Prevalencia=as.data.frame(cbind(pr_casos, pr_controles[, "pr_controles"]
[match(rownames(pr_casos), rownames(pr_controles))]))
names(Prevalencia)[names(Prevalencia) == "pr_casos"] <- "Prevalencia_casos"
names(Prevalencia)[names(Prevalencia) == "pr_controles[, \"pr_controles\"]
[match(rownames(pr_casos), rownames(pr_controles))]"] <- "Prevalencia_controles"
Prevalencia$Prevalencia_max <-round(apply(Prevalencia, 1, max),2)*100
Prevalencia$Prevalencia_min <-round(apply(Prevalencia, 1, min),2)*100

keepTaxa = rownames(Prevalencia)[(Prevalencia$Prevalencia_max >= 20)]
```

```
Muestra_20_WT2D=prune_taxa(keepTaxa, WT2D_species)

#NORMALIZACIÓN
S1_N1_WT2D=tss(WT2D_species@otu_table)
S2_N1_WT2D=tss(Muestra_20_WT2D@otu_table)
S1_N2_WT2D=CLR(WT2D_species@otu_table+1)
S2_N2_WT2D=CLR(Muestra_20_WT2D@otu_table+1)



S1_NORMAL_WT2D=list(S1_TSS_WT2D=S1_N1_WT2D,S1_CLR_WT2D=S1_N2_WT2D)
S2_NORMAL_WT2D=list(S2_TSS_WT2D=S2_N1_WT2D,S2_CLR_WT2D=S2_N2_WT2D)
Filtro_WT2D=list(S1_Pasolli_WT2D=WT2D_species@otu_table,
S2_20_WT2D=Muestra_20_WT2D@otu_table)
Base_total_WT2D=list(Base=WT2D@otu_table)
WT2D_lista=list(Original=Base_total_WT2D,Filtros_WT2D=Filtro_WT2D,
S1_NORMAL=S1_NORMAL_WT2D,S2_NORMAL=S2_NORMAL_WT2D)
```

Se cargan las dos listas una donde vienen la abundancia de los OTU's y otra donde viene la información de la muestra (Covariables), para cada base de datos:

```
    DatosFinal=list(WT2D=WT2D_lista,T2D=T2D_lista,Cirrosis=Cirrosis_lista,
    CancerColorectal=CANCERCOL_lista,obesidad=Obesidad_lista,ibd=IBD_lista)
Datos_sample=list(Cirrosis=cirrosis_species@sam_data,WT2D=WT2D_species@sam_data,
T2D=T2D_species@sam_data,Cancer_col=Cancer_Colorectal_species@sam_data,
Obesidad=Obesidad_species@sam_data,IBD=IBD_species@sam_data)
Datos_Taxa=list(WT2D=Taxa_WT2D,T2D=Taxa_T2D,Cirrosis=TAX_CIRROSIS,
CancerColorectal=Taxa_Cancer_Colorectal,Obsesidad=Taxa_Obesidad,IBD=Taxa_IBD)
save(DatosFinal,Datos_sample,Datos_Taxa,file = "DatosSOv2.0.RData")
```

A.0.3. ANEXO III. RESULTADOS PCA PARA TODAS LAS BASES DE DATOS Y DATOS ANOMALOS.

Para la normalización TSS (N1), cuando se aplica PCA sobre datos transformados logarítmicamente (logaritmo natural), se suma 1 a todos los datos para evitar infinitos en los ceros.

```
    res.pca.final=vector("list", length = length(DatosFinal))
names(res.pca.final) = names(DatosFinal)
 for (i in seq(1:6)) {
   res.pca.final[[i]] = vector("list", length = 4)
   names(res.pca.final[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
   l = 1
  for (j in c(3,4)) {
    for (k in c(1,2)) {
      if (k == 1) {
        res.pca.final[[i]][[l]] = vector("list", length = 4)
        names(res.pca.final[[i]][[l]]) = c("log-NE", "log-ESC", "NE", "ESC")
        # log(N1+1)
        res.pca.final[[i]][[l]][[1]] = PCA(scale(log(t(DatosFinal[[i]][[j]][[k]]+1)),
                                           center = TRUE, scale = FALSE),
                                    scale.unit = FALSE, graph = FALSE, ncp = 2)
        # log(N1+1) escalado
        res.pca.final[[i]][[l]][[2]] = PCA(scale(log(t(DatosFinal[[i]][[j]][[k]]+1)),
                                           center=TRUE,scale=TRUE),
                                    scale.unit = FALSE, graph = FALSE, ncp = 2)
        # N1 sin escalar
        res.pca.final[[i]][[l]][[3]] = PCA(scale((t(DatosFinal[[i]][[j]][[k]]))),
                                           center = TRUE, scale = FALSE),
                                    scale.unit = FALSE, graph = FALSE, ncp = 2)
```

```
        # N1 escalado
        res.pca.final[[i]][[l]][[4]] = PCA(scale((t(DatosFinal[[i]][[j]][[k]])),
                                     center = TRUE, scale = TRUE),
                                  scale.unit = FALSE, graph = FALSE, ncp = 2)
        } else { # k= 2 (CLR)
        res.pca.final[[i]][[l]] = vector("list", length = 2)
        names(res.pca.final[[i]][[l]]) = c("NE", "ESC")
        # N2 sin escalar
        res.pca.final[[i]][[l]][[1]] = PCA(scale(t(DatosFinal[[i]][[j]][[k]]),
                                     center = TRUE, scale = FALSE),
                                  scale.unit = FALSE, graph = FALSE, ncp = 2)
        # N2 escalado
        res.pca.final[[i]][[l]][[2]] = PCA(scale(t(DatosFinal[[i]][[j]][[k]]),
                                     center = TRUE, scale = TRUE),
                                  scale.unit = FALSE, graph = FALSE, ncp = 2)
        }
      l = l+1
    }
  }
 }
```

Loading plots:

```
    for (i in 1:length(res.pca.final)) {
  cat(names(res.pca.final)[i], sep = "\n")
  for (j in 1:length(res.pca.final[[i]])) {
    cat(names(res.pca.final[[i]])[j], sep = "\n")
    par(mfrow = c(1,2))
    for (k in 1:length(res.pca.final[[i]][[j]])) {
      plot(res.pca.final[[i]][[j]][[k]]$var$coord, col = "red3", asp = 1,
           main = names(res.pca.final[[i]][[j]])[k])
    }
  }
}
```

A la vista de los loading plots, se decide escalar siempre las variables y no transformar logarítmicamente los datos normalizados mediante TSS.

Datos atípicos o anómalos:

En cuanto a los datos anómalos, se calculan los valores T2 (valores extremos) y SCR (valores atípicos) y su límite de confianza al 99 %. Dado que podemos tener un 1 % de falsos positivos, se descarta ese número de los que exceden el límite y se eliminan el resto de anómalos o atípicos.

```
    Datos_anomalos <- function(acp) {
  K = 2
  # T2
  misScores = acp$ind$coord[,1:K]
  eig.val=get_eigenvalue(acp)
  miT2 = colSums(t(misScores**2) / eig.val[1:K,1])
  I = nrow(acp$ind$coord)
  F99 = K*(I**2 - 1)/(I*(I - K)) * qf(0.99, K, I-K)
  anomalasT2 = which(miT2 > F99)
  if (round(0.05*I, 0) > length(anomalasT2)) {
  anomalasT2 = NULL
  } else {
    num = length(anomalasT2)-round(0.05*I, 0)
    anomalasT2 = names(sort(miT2, decreasing = TRUE)[1:num])
  }
```

```
  # SCR
  misLoadings = sweep(acp$var$coord, 2, sqrt(acp$eig[1:K,1]), FUN="/")
  X = as.matrix(acp$call$X)
  myE = X - misScores %*% t(misLoadings)
  mySCR = rowSums(myE^2)
  g = var(mySCR)/(2*mean(mySCR))
  h = (2*mean(mySCR)^2)/var(mySCR)
  chi2lim = g*qchisq(0.99, df = h)
  anomalasSCR = which(mySCR > chi2lim)
  if (round(0.05*I, 0) > length(anomalasSCR) ) {
    anomalasSCR = NULL
    } else {
      num = length(anomalasSCR)-round(0.05*I, 0)
      anomalasSCR = names(sort(mySCR, decreasing = TRUE)[1:num])
    }
  anomalas = list(anomalasT2, anomalasSCR)
  names(anomalas) = c("T2", "SCR")
  return(anomalas)
}
```

Función para detectar anomalos:

```
    for (i in 1:length(res.pca.final)) {
  for (j in 1:length(res.pca.final[[i]])) {
    res.pca.final[[i]][[j]] = res.pca.final[[i]][[j]]$ESC
  }
}
losanomalos = lapply(res.pca.final,
                     function (x) lapply(x, Datos_anomalos))
```

Eliminación Anomalos

```
    par(mfrow = c(2,2))
for (i in 1:4) {
  acp = res.pca.final$WT2D[[i]]
  K = 2
  # T2
  misScores = acp$ind$coord[,1:K]
  eig.val=get_eigenvalue(acp)
  miT2 = colSums(t(misScores**2) / eig.val[1:K,1])
  I = nrow(acp$ind$coord)
  F99 = K*(I**2 - 1)/(I*(I - K)) * qf(0.99, K, I-K)
  plot(1:length(miT2), miT2, type = "l", col = "red4",
       xlab = "samples", ylab = "T2", main = names(res.pca.final$WT2D)[i])
  abline(h = F99, lty = 2, col = 4)
  points(which(names(miT2) == "SAMPLE 86"), miT2["SAMPLE 86"], col = 4, pch = 15)
}
```

A.0.4. ANEXO IV. Optimización para los modelos (Hiperparámetros y Cutoff.)

Optimización PLSDA:

```
    Datos_sample = Datos_sample[c(2:3,1,4:6)]
# Cambiamos a "d" y "n" en todas las BBDD y as? es m?s f?cil automatizar
Datos_sample[[1]]$disease<-factor(Datos_sample[[1]]$disease,labels=c("n","d"))
Datos_sample[[2]]$disease<-factor(Datos_sample[[2]]$disease,labels=c("n","d"))
Datos_sample[[3]]$disease<-factor(Datos_sample[[3]]$disease,labels=c("d","n"))
```

```
Datos_sample[[4]]$disease<-factor(Datos_sample[[4]]$disease,labels=c("d","n","n"))
Datos_sample[[5]]$disease<-factor(Datos_sample[[5]]$disease,labels=c("n","d"))
Datos_sample[[6]]$disease<-factor(Datos_sample[[6]]$disease,labels=c("d","d","n"))

# Reordenamos niveles para que 0=n y 1=d
for (i in 1:6) {
  Datos_sample[[i]]$disease = relevel(Datos_sample[[i]]$disease, "n")
}



# C?lculo probabilidades predichas PLS-DA ----------------------------------

# Para 10 componentes, k-fold con k=10 y r=5

nComp = 10
nK = 10
nRep = 5

Folds = vector("list", length = 6)  # selección de k-folds
probPred=vector(mode = "list", length = 6)
names(Folds) = names(probPred) = names(DatosFinal)

compo = paste0("ncp", 1:nComp)
compo = rep(compo, each = nRep)
repe = paste0("r", 1:nRep)
repe = rep(repe, nComp)
comprep = apply(cbind(compo, repe), 1, paste, collapse = "_")

set.seed(123)


for (i in 1:6) {  # enfermedades
  proced = Datos_sample[[i]]$disease
  Folds[[i]] = createMultiFolds(proced,k=nK,times=nRep)
  probPred[[i]] = vector("list", length = 4)
  names(probPred[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  sn = 1
  for (j in 3:4) { # S1, S2
    #if (i==1 & j == 4) { proced = proced[-c(86)]}
    for (k in 1:2) {  # N1, N2

      Xda = t(DatosFinal[[i]][[j]][[k]])  # pacientes en filas
      if (k == 1) { Xda = log(Xda*10^6 + 1) }

      datos=data.frame(Xda,proced)
      probPred[[i]][[sn]] = matrix(NA, nrow = nrow(datos), ncol = nComp*nRep) # col = comp*rep
      colnames(probPred[[i]][[sn]]) = comprep
      rownames(probPred[[i]][[sn]]) = rownames(datos)
      for (comp in 1:nComp) { # componentes PLS-DA
        for (rep in 1:nRep) {  # repeticiones del k-fold
          for(fold in 1:nK) {  # k folds
            fff = fold + nK*(rep-1)
            micol = paste0("ncp", comp, "_r", rep)
            misfil = rownames(datos)[-Folds[[i]][[fff]]]
            train<-datos[Folds[[i]][[fff]],]
            test<-datos[-Folds[[i]][[fff]],]
            train=na.omit(train)
```

```
            modelo <- opls(train[,-ncol(train)],train$proced, crossvalI=1, permI = 0,
                           predI = comp, scaleC="standard", fig.pdfC="none", info.txtC="none")
            X = test[,rownames(modelo@loadingMN)]
            X = scale(X, center = modelo@xMeanVn, scale = modelo@xSdVn)
            Ypred = as.matrix(X) %*% modelo@coefficientMN
            Ypred = as.numeric(Ypred*modelo@ySdVn + modelo@yMeanVn)
            probPred[[i]][[sn]][misfil,micol] = Ypred
          }
        }
      }
      sn = sn+1
    }
  }
}


save(Folds, probPred, file="ResultsPLSDA.Rdata")




# Cálculo el ?ndice de Youden para cortes y componentes -------------------------

YoudenCalc = function(corte, probab, Yobs) {
  yPred = factor(ifelse(probab < corte, "n", "d"))
  CM = caret::confusionMatrix(yPred, Yobs, positive = "d")
  J = sum(CM$byClass[c("Specificity", "Sensitivity")]) - 1
  return(J)
}


cortes = seq(0.1,1,0.01)
myYouden = vector("list", length = 6)
names(myYouden) = names(DatosFinal)

for (i in 1:6) {
  myYouden[[i]] = vector("list", length = 4)
  names(myYouden[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  for (j in 1:4) {
    AllYouden = NULL
    for (k in 1:length(comprep)) {

      miJ = sapply(cortes,
                   function (ccc) YoudenCalc(corte = ccc,
                                             probab = (probPred[[i]][[j]][,k]),
                                             Yobs=Datos_sample[[i]]$disease ))
      AllYouden = rbind(AllYouden, miJ)
    }
    rownames(AllYouden) = comprep
    colnames(AllYouden) = cortes
    tmp = aggregate(AllYouden, by = list("comp" = compo), mean)
    rownames(tmp) = tmp$comp
    tmp = t(tmp[,-1])
    myYouden[[i]][[j]] = tmp
  }
}
```

```
save(Folds, probPred, myYouden, file="ResultsPLSDA.Rdata")



# Optimización de corte y n?mero componentes ---------------------------------

load("ResultsPLSDA.Rdata", verbose = TRUE)

optims = function (matriz) {
  elem = which.max(matriz)
  k <- arrayInd(elem, dim(matriz))
  ncp = colnames(matriz)[k[,2]]
  ncp = substr(ncp, 4, nchar(ncp))
  ooo = c(rownames(matriz)[k[,1]], ncp)
  ooo = as.numeric(ooo)
  names(ooo) = c("corte", "ncp")
  return(ooo)
}

optimsPLSDA = lapply(myYouden, sapply, optims)

save(Folds, probPred, myYouden, optimsPLSDA, file="ResultsPLSDA.Rdata")
```

Optimización SVM:

```
    library(phyloseq) # para cargar los datos
library(caret)  # k-fold CV
require(e1071)
load("~/Documents/TFM/TFM Final/DatosSORData")
load("~/Documents/TFM/TFM Final/ResultsPLSDA.Rdata") #Carga Folds
Datos_sample = Datos_sample[c(2:3,1,4:6)]

# Cambiamos a "d" y "n" en todas las BBDD y as? es m?s f?cil automatizar
Datos_sample[[1]]$disease<-factor(Datos_sample[[1]]$disease,labels=c("n","d"))
Datos_sample[[2]]$disease<-factor(Datos_sample[[2]]$disease,labels=c("n","d"))
Datos_sample[[3]]$disease<-factor(Datos_sample[[3]]$disease,labels=c("d","n"))
Datos_sample[[4]]$disease<-factor(Datos_sample[[4]]$disease,labels=c("d","n","n"))
Datos_sample[[5]]$disease<-factor(Datos_sample[[5]]$disease,labels=c("n","d"))
Datos_sample[[6]]$disease<-factor(Datos_sample[[6]]$disease,labels=c("d","d","n"))

#Valores a evaluar
costV = c(2^{-5},2^{-3},2,2^{3},2^{5},2^{7},2^{9},2^{11},2^{13},2^{15})
gammaV =c(2^{-15},2^{-13},2^{-11},2^{-9},2^{-7},2^{-5},2^{-3},2^{-1},2,2^{3})

## Reordenamos niveles para que 0=n y 1=d
for (i in 1:6) {
  Datos_sample[[i]]$disease = relevel(Datos_sample[[i]]$disease, "n")
}


nK = 10
nRep = 5

probPred=vector(mode = "list", length = 6)
```

```
names(Folds) = names(probPred) = names(DatosFinal)

cost = paste0("cost", 1:length(costV))
cost= rep(cost, each = 10)
cost=rep(cost,each=5)
gamma= paste0("gamma", 1:length(gammaV))
gamma = rep(gamma, 1)
gamma = rep(gamma, 10)
gamma = rep(gamma, 5)
repe = paste0("r", 1:nRep)
repe = rep(repe, each=10)
repe = rep(repe, 10)
comprep = apply(cbind(repe,cost,gamma), 1, paste, collapse = "_")

comp=apply(cbind(cost,gamma), 1, paste, collapse = "_")

set.seed(123)
for (i in 1:6) {  # enfermedades
  proced = Datos_sample[[i]]$disease
  probPred[[i]] = vector("list", length = 4)
  names(probPred[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  sn = 1
  for (j in 3:4) { # S1, S2
    #if (i==1 & j == 4) { proced = proced[-c(86)]}
    for (k in 1:2) {  # N1, N2

      Xda = t(DatosFinal[[i]][[j]][[k]])  # pacientes en filas
      if (k == 1) { Xda = log(Xda*10^6 + 1) }

      datos=data.frame(Xda,proced)
      probPred[[i]][[sn]] = matrix(NA, nrow = nrow(datos), ncol = length(comprep)) # col = comp
      colnames(probPred[[i]][[sn]]) = comprep
      rownames(probPred[[i]][[sn]]) = rownames(datos)
      for (c in 1:length(costV)) { # cost
        for (g in 1:length(gammaV)){
        for (rep in 1:nRep) {  # repeticiones del k-fold
          for(fold in 1:nK) {  # k folds

            fff = fold + nK*(rep-1)
            micol = paste0("r", rep,"_","cost",c,"_","gamma",g)
            misfil = rownames(datos)[-Folds[[i]][[fff]]]
            train<-datos[Folds[[i]][[fff]],]
            test<-datos[-Folds[[i]][[fff]],]
            train=na.omit(train)
            modelo <- svm(factor(train$proced) ~ .,data=train[,-ncol(train)],
    kernel="radial",cost=costV[c],gamma=gammaV[g],scale=F,probability=TRUE)
            Ypred = predict(modelo,test,probability=TRUE)
            probPred[[i]][[sn]][misfil,micol] <-
            (as.data.frame(attr(Ypred, "probabilities")))[,2]
            print(paste0(i,j,k,micol))
          }
        }
       }
      }
      sn = sn+1

    }
  }
```

```
}
save(probPred, file="ResultsSVM.Rdata")


# C?lculo el ?ndice de Youden para cortes y componentes -------------------------
load("~/Documents/TFM/TFM Final/ResultsSVM.Rdata")
YoudenCalc = function(corte, probab, Yobs) {
  yPred = factor(ifelse(probab < corte, "n", "d"))
  CM = caret::confusionMatrix(yPred, Yobs, positive = "d")
  J = sum(CM$byClass[c("Specificity", "Sensitivity")]) - 1
  return(J)
}


cortes = seq(0.1,1,0.01)
myYouden = vector("list", length = 6)
names(myYouden) = names(DatosFinal)

for (i in 1:6) {
  myYouden[[i]] = vector("list", length = 4)
  names(myYouden[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  for (j in 1:4) {
    AllYouden = NULL
    for (k in 1:length(comprep)) {

      miJ = sapply(cortes,
                   function (ccc) YoudenCalc(corte = ccc,
                                             probab = (probPred[[i]][[j]][,k]),
                                             Yobs=Datos_sample[[i]]$disease

                   ))
      AllYouden = rbind(AllYouden, miJ)
    }
    rownames(AllYouden) = comprep
    colnames(AllYouden) = cortes
    tmp = aggregate(AllYouden, by = list("comp" = comp), mean)
    rownames(tmp) = tmp$comp
    tmp = t(tmp[,-1])
    myYouden[[i]][[j]] = tmp
  }
}


save(probPred, myYouden, file="ResultsSVM.Rdata")
# Optimización de corte y n?mero componentes ----------------------------------

load("ResultsSVM.Rdata", verbose = TRUE)

optims = function (matriz) {
  elem = which.max(matriz)
  k <- arrayInd(elem, dim(matriz))
  ncp = colnames(matriz)[k[,2]]
  gamma= substr(ncp,nchar(ncp) , nchar(ncp))
  cost=substr(ncp, start = 5, stop = 5)
  ooo = c(rownames(matriz)[k[,1]], gammaV[as.numeric(gamma)],costV[as.numeric(cost)])
  ooo = as.numeric(ooo)
  names(ooo) = c("corte", "gamma","cost")
  return(ooo)
```

```
}
optimsSVM = lapply(myYouden, sapply, optims)

save(probPred, myYouden, optimsSVM, file="ResultsSVM.Rdata")
```

Optimización RF:

```
    load("~/Documents/TFM/TFM Final/DatosSO.RData")
load("~/Documents/TFM/TFM Final/ResultsPLSDA.Rdata") #Carga Folds

Datos_sample = Datos_sample[c(2:3,1,4:6)]
# Cambiamos a "d" y "n" en todas las BBDD y as? es m?s f?cil automatizar
Datos_sample[[1]]$disease<-factor(Datos_sample[[1]]$disease,labels=c("n","d"))
Datos_sample[[2]]$disease<-factor(Datos_sample[[2]]$disease,labels=c("n","d"))
Datos_sample[[3]]$disease<-factor(Datos_sample[[3]]$disease,labels=c("d","n"))
Datos_sample[[4]]$disease<-factor(Datos_sample[[4]]$disease,labels=c("d","n","n"))
Datos_sample[[5]]$disease<-factor(Datos_sample[[5]]$disease,labels=c("n","d"))
Datos_sample[[6]]$disease<-factor(Datos_sample[[6]]$disease,labels=c("d","d","n"))

# Reordenamos niveles para que 0=n y 1=d
for (i in 1:6) {
  Datos_sample[[i]]$disease = relevel(Datos_sample[[i]]$disease, "n")
}

####Modelo
nK = 10
nRep = 5
probPred=vector(mode = "list", length = 6)
names(Folds) = names(probPred) = names(DatosFinal)
repe = paste0("r", 1:nRep)
comprep = repe



set.seed(123)
for (i in 1:6) {  # enfermedades
                  proced = Datos_sample[[i]]$disease
                  probPred[[i]] = vector("list", length = 4)
                  names(probPred[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
                  sn = 1
                  for (j in 3:4) { # S1, S2
                    #if (i==1 & j == 4) { proced = proced[-c(86)]}
                    for (k in 1:2) {  # N1, N2

                      Xda = t(DatosFinal[[i]][[j]][[k]])  # pacientes en filas
                      if (k == 1) { Xda = log(Xda*10^6 + 1) }

                      datos=data.frame(Xda,proced)
                      probPred[[i]][[sn]] = matrix(NA, nrow = nrow(datos),
                      ncol = length(comprep))
                      # col = comp*rep
                      colnames(probPred[[i]][[sn]]) = comprep
                      rownames(probPred[[i]][[sn]]) = rownames(datos)

                          for (rep in 1:nRep) {  # repeticiones del k-fold
                            for(fold in 1:nK) {  # k folds
```

```
                            fff = fold + nK*(rep-1)
                            micol = paste0("r", rep)
                            misfil = rownames(datos)[-Folds[[i]][[fff]]]
                            train<-datos[Folds[[i]][[fff]],]
                            test<-datos[-Folds[[i]][[fff]],]
                            train=na.omit(train)
                            modelo <- randomForest(factor(train$proced) ~ .,
                            data=train[,-ncol(train)],ntree=500,mtry=sqrt(ncol(train)-1))
                            Ypred = predict(modelo,test,type="prob")
                            probPred[[i]][[sn]][misfil,micol] <-(as.data.frame(Ypred))[,2]
                            print(paste0(i,j,k,micol))
                          }
                        }

                      sn = sn+1

                    }
                  }
                }
save(probPred, file="ResultsRF.Rdata")
# C?lculo el ?ndice de Youden para cortes y componentes -------------------------
load("~/Documents/TFM/TFM Final/ResultsRF.Rdata")
YoudenCalc = function(corte, probab, Yobs) {
  yPred = factor(ifelse(probab < corte, "n", "d"))
  CM = caret::confusionMatrix(yPred, Yobs, positive = "d")
  J = sum(CM$byClass[c("Specificity", "Sensitivity")]) - 1
  return(J)
}


cortes = seq(0.1,1,0.01)
myYouden = vector("list", length = 6)
names(myYouden) = names(DatosFinal)

for (i in 1:6) {

  myYouden[[i]] = vector("list", length = 4)
  names(myYouden[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  for (j in 1:4) {

    AllYouden = NULL
    for (k in 1:length(comprep)) {

      miJ = sapply(cortes,
                   function (ccc) YoudenCalc(corte = ccc,
                                             probab = (probPred[[i]][[j]][,k]),
                                             Yobs= Datos_sample[[i]]$disease
                   ))
      AllYouden = rbind(AllYouden, miJ)
    }
    rownames(AllYouden) = comprep
    colnames(AllYouden) = cortes
    tmp<-colMeans(AllYouden)
    tmp=as.data.frame(tmp)
    tmp$corte = cortes
    #tmp = t(tmp[,-1])
    myYouden[[i]][[j]] = tmp
```

```
  }
}



save(probPred, myYouden, file="ResultsSF.Rdata")

# Optimizaci?n de corte y n?mero componentes ----------------------------------

load("~/Documents/TFM/TFM Final/ResultsRF.Rdata")

optims = function (matriz) {
  elem = which.max(matriz$tmp)
  k <- arrayInd(elem, dim(matriz))
  ooo = c(rownames(matriz)[k[,1]])
  ooo = as.numeric(ooo)
  names(ooo) = c("corte")
  return(ooo)

}

optimsRF = lapply(myYouden, sapply, optims)

save(probPred, myYouden, optimsRF, file="ResultsRF.Rdata")

#Performance: AUC, F1-Score
```

A.0.5. ANEXO V. MODELOS FINALES.

```
    ####  MODELO
library(phyloseq) # para cargar los datos
library(ropls) # PLS-DA
library(caret)  # k-fold CV
library(randomForest)
library(pROC)
load("~/Documents/TFM/TFM Final/DatosSO.RData")
load("~/Documents/TFM/TFM Final/ResultsRF.Rdata")
load("~/Documents/TFM/TFM Final/ResultsSVM.Rdata")
load("~/Documents/TFM/TFM Final/ResultsPLSDA.Rdata")

Datos_sample = Datos_sample[c(2:3,1,4:6)]
# Cambiamos a "d" y "n" en todas las BBDD y as? es m?s f?cil automatizar
Datos_sample[[1]]$disease<-factor(Datos_sample[[1]]$disease,labels=c("n","d"))
Datos_sample[[2]]$disease<-factor(Datos_sample[[2]]$disease,labels=c("n","d"))
Datos_sample[[3]]$disease<-factor(Datos_sample[[3]]$disease,labels=c("d","n"))
Datos_sample[[4]]$disease<-factor(Datos_sample[[4]]$disease,labels=c("d","n","n"))
Datos_sample[[5]]$disease<-factor(Datos_sample[[5]]$disease,labels=c("n","d"))
Datos_sample[[6]]$disease<-factor(Datos_sample[[6]]$disease,labels=c("d","d","n"))

# Reordenamos niveles para que 0=n y 1=d
for (i in 1:6) {
  Datos_sample[[i]]$disease = relevel(Datos_sample[[i]]$disease, "n")
}
```

```
# C?lculo probabilidades predichas PLS-DA ----------------------------------

# Para 10 componentes, k-fold con k=10 y r=5

nComp = c(2,4,6,8)
nK = 10
nRep = 5

Folds = vector("list", length = 6)  # selección de k-folds
probPredPLSDA=vector(mode = "list", length = 6)
probPredSVM=vector(mode = "list", length = 6)
probPredRF=vector(mode = "list", length = 6)
names(Folds) = names(probPredPLSDA) =names(probPredSVM)=names(probPredRF)= names(DatosFinal)

comp=c()
set.seed(123)
for (i in 1:6) {  # enfermedades
  proced = Datos_sample[[i]]$disease
  Folds[[i]] = createMultiFolds(proced,k=nK,times=nRep)
  probPredPLSDA[[i]] = vector("list", length = 4)
  names(probPredPLSDA[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  probPredSVM[[i]] = vector("list", length = 4)
  names(probPredSVM[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  probPredRF[[i]] = vector("list", length = 4)
  names(probPredRF[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
  sn = 1
  for (j in 3:4) { # S1, S2
    for (k in 1:2) {  # N1, N2
      Xda = t(DatosFinal[[i]][[j]][[k]])  # pacientes en filas
      if (k == 1) { Xda = log(Xda*10^6 + 1) }
      datos=data.frame(Xda,proced)
      probPredPLSDA[[i]][[sn]] = matrix(NA, nrow = nrow(datos), ncol = 5) # col = comp*rep
      colnames(probPredPLSDA[[i]][[sn]]) =c(1:5)
      rownames(probPredPLSDA[[i]][[sn]]) = rownames(datos)


      probPredSVM[[i]][[sn]] = matrix(NA, nrow = nrow(datos), ncol = 5) # col = comp*rep
      colnames(probPredSVM[[i]][[sn]]) = c(1:5)
      rownames(probPredSVM[[i]][[sn]]) = rownames(datos)

      probPredRF[[i]][[sn]] = matrix(NA, nrow = nrow(datos), ncol = 5) # col = comp*rep
      colnames(probPredRF[[i]][[sn]]) = c(1:5)
      rownames(probPredRF[[i]][[sn]]) = rownames(datos)

      if (j==3 & k==1){ comp[1]= c(1)}
      if(j==3 & k==2) { comp[1]= c(2)}
      if(j==4 & k==1) { comp[1]= c(3)}
      if(j==4 & k==2) { comp[1]= c(4)}



        for (rep in 1:nRep) {  # repeticiones del k-fold
          for(fold in 1:nK) {  # k folds
            fff = fold + nK*(rep-1)
            micol = rep
            misfil = rownames(datos)[-Folds[[i]][[fff]]]
            train<-datos[Folds[[i]][[fff]],]
            test<-datos[-Folds[[i]][[fff]],]
```

```
            train=na.omit(train)

            modelo <- opls(train[,-ncol(train)],train$proced, crossvalI=1, permI = 0,
                           predI = optimsPLSDA[[i]][2,comp], scaleC="standard", fig.pdfC="none"
            X = test[,rownames(modelo@loadingMN)]
            X = scale(X, center = modelo@xMeanVn, scale = modelo@xSdVn)
            Ypred = as.matrix(X) %*% modelo@coefficientMN
            Ypred = as.numeric(Ypred*modelo@ySdVn + modelo@yMeanVn)
            probPredPLSDA[[i]][[sn]][misfil,micol] = Ypred


            modelo <- randomForest(factor(train$proced) ~ .,data=train[,-ncol(train)],ntree=500
            Ypred = predict(modelo,test,type="prob")
            probPredRF[[i]][[sn]][misfil,micol] <-(as.data.frame(Ypred))[,2]

            modelo <- svm(factor(train$proced) ~ .,data=train[,-ncol(train)],kernel="radial",co
            Ypred = predict(modelo,test,probability=TRUE)
            probPredSVM[[i]][[sn]][misfil,micol]<-(as.data.frame(attr(Ypred, "probabilities")))

            print(paste0(i,j,k,micol))

          }
        }

      sn = sn+1
    }
  }
}
save(probPredSVM,probPredRF,probPredPLSDA,file="ModelosFinal.Rdata")

#############################################################################################################
load("~/Documents/TFM/TFM Final/ModelosFinal.Rdata")

YoudenCalc = function(corte, probab, Yobs) {
  yPred = factor(ifelse(probab < corte, "n", "d"))
  CM = caret::confusionMatrix(as.factor(yPred), as.factor(Yobs), positive = "d")
  Resultados=c(CM$byClass[c("Precision")],CM$byClass[c("Recall")],CM$byClass[c("F1")],CM$overal
  return(Resultados)
}

myResultsPLSDA = vector("list", length = 6)
names(myResultsPLSDA) = names(DatosFinal)

myResultsSVM = vector("list", length = 6)
names(myResultsSVM) = names(DatosFinal)

myResultsRF= vector("list", length = 6)
names(myResultsRF) = names(DatosFinal)
for (i in 1:6) {
  myResultsPLSDA [[i]] = vector("list", length = 4)
  names(myResultsPLSDA [[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")

  myResultsSVM[[i]] = vector("list", length = 4)
  names(myResultsSVM[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")

  myResultsRF[[i]] = vector("list", length = 4)
  names(myResultsRF[[i]]) = c("S1-N1", "S1-N2", "S2-N1", "S2-N2")
```

```
  for (j in 1:4) {
    AllResultsPLSDA = NULL
    AllResultsSVM = NULL
    AllResultsRF = NULL
    for (k in 1:5) {

      miJPLSDA = YoudenCalc(corte = optimsPLSDA[[i]][1,j],
                                          probab = (probPredPLSDA[[i]][[j]][,k]),
                                          Yobs=Datos_sample[[i]]$disease )
      AllResultsPLSDA = rbind(AllResultsPLSDA, miJPLSDA)


      miJSVM = YoudenCalc(corte = optimsSVM[[i]][1,j],
                          probab = (probPredSVM[[i]][[j]][,k]),
                          Yobs=Datos_sample[[i]]$disease )
      AllResultsSVM = rbind(AllResultsSVM, miJSVM)

      miJRF= YoudenCalc(corte = optimsRF[[i]][j],
                            probab = (probPredRF[[i]][[j]][,k]),
                            Yobs=Datos_sample[[i]]$disease )
      AllResultsRF = rbind(AllResultsRF, miJRF)
    }
    rownames(AllResultsPLSDA) = 1:5
    myResultsPLSDA[[i]][[j]] = AllResultsPLSDA

    rownames(AllResultsSVM) = 1:5
    myResultsSVM[[i]][[j]] = AllResultsSVM

    rownames(AllResultsRF) = 1:5
    myResultsRF[[i]][[j]] = AllResultsRF

  }
}

save(myResultsRF,myResultsSVM,myResultsPLSDA,file="ResultadosFinal.Rdata")


MediasPLSDA = vector("list", length = 6)
names(MediasPLSDA) = names(DatosFinal)

MediasSVM= vector("list", length = 6)
names(MediasSVM) = names(DatosFinal)

MediasRF= vector("list", length = 6)
names(MediasRF) = names(DatosFinal)

for (i in 1:6) {
  MediasPLSDA[[i]]<-sapply(myResultsPLSDA[[i]],colMeans,na.rm = TRUE)
  MediasRF[[i]]<-sapply(myResultsRF[[i]],colMeans,na.rm = TRUE)
  MediasSVM[[i]]<-sapply(myResultsSVM[[i]],colMeans,na.rm = TRUE)
}

save(MediasPLSDA,MediasRF,MediasSVM,file="MediasFinal.Rdata")
```