Trabajo fin de grado

Desarrollo de un modelo estadístico para la clasificación de anticuerpos

Autora:
Elisa Grao Andrés

Directores:
Sergio Pérez Gaviro
David Luna Cerralbo

Septiembre 2023

Índice

1. Introducción

El desarrollo de fármacos basados en anticuerpos ha alcanzado una notable relevancia en los últimos años. Un paso clave en este proceso de desarrollo es la humanización de anticuerpos. Se trata este, no obstante, de un proceso largo y costoso. Es por ello que en este trabajo se utiliza un método ya desarrollado con anterioridad en [1] y [2], el cual caracteriza la distribución estadística de las secuencias de anticuerpos humanos. Se analizará si cambiando la base de datos de secuencias empleada en este modelo se logran extraer resultados más precisos.

La función de los anticuerpos en la defensa del organismo es de vital importancia. Cuando una sustancia extraña, denominada antígeno, penetra en nuestro organismo, nuestro sistema inmunitario activa una respuesta que generalmente implica la producción de anticuerpos. Los anticuerpos son proteínas que serán capaces de identificar y neutralizar el antígeno, propiciando su destrucción o bloqueo.

La molécula de anticuerpo está formada por cuatro cadenas polipeptídicas: dos son de mayor tamaño y se denominan cadenas pesadas (VH); las otras dos, de menor tamaño, se conocen como cadenas ligeras (VL). Estas cadenas están unidas entre sí por un tipo de enlace conocido como puente disulfuro y conforman una estructura con forma de "Y". Cada molécula de anticuerpo presenta dos regiones, la región constante y la región variable, siendo esta última la encargada de reconocer el agente extraño y, por tanto, es específica para cada antígeno.

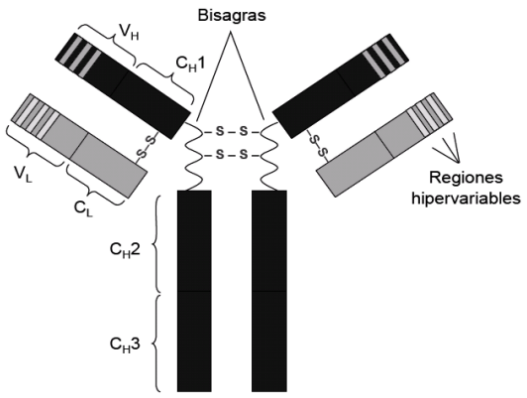


Figura 1: Estructura básica de un anticuerpo [3]. Las cadenas pesadas se muestran en negro y las cadenas ligeras, en gris

La región variable está formada a su vez por siete zonas diferentes de aminoácidos, cuatro de las cuales son las denominadas 'región marco' (*Framework Region, FR*), cuya función principal pasa por ser un soporte estructural para las tres zonas restantes, conocidas como regiones hipervariables (*Complementary Determining Regions, CDR*). Estas regiones contactan directamente con el antígeno y lo complementan, es decir, se produce un acoplamiento específico entre anticuerpo y antígeno para inhibir la toxicidad de este último. Son, por tanto, zonas que poseen una especificidad concreta para un antígeno dado.

En el proceso de obtención de anticuerpos para uso terapéutico en humanos, en un primer paso se infectan a animales, generalmente ratones, cuyo sistema inmunitario generará los anticuerpos

específicos para combatir la enfermedad suministrada. Si extraemos estos anticuerpos y los inyectamos directamente en el organismo de un humano, su cuerpo lo identificará como un agente extraño que hay que eliminar. Es en este momento donde entra en juego la humanización de anticuerpos. Se trata así de modificar los anticuerpos murinos de tal forma que sean tolerados por el organismo humano, pero que a su vez sean capaces de conservar su especificidad para combatir el antígeno. Concretamente, este proceso consiste en extraer las zonas *CDR* murinas e insertarlas en una región marco *FR* humana, suficientemente semejante al dominio murino. En la mayoría de ocasiones esto no es suficiente y son necesarias mutaciones en los aminoácidos de la región marco *FR* hasta hallar el anticuerpo que cumpla con los requisitos deseados. Como no se conocen de forma exacta las mutaciones requeridas, se trata de un procedimiento experimental basado en prueba y error y, por tanto, muy costoso.

El método desarrollado en [1] utiliza una base de datos conformada por secuencias de anticuerpos humanos para aprender de ella, denominada base de datos 'learning', y emplea otra base conformada por secuencias de anticuerpos humanos y murinos para realizar las predicciones, denominada base de datos 'test'. Estas predicciones son buenas, pero se pretende comprobar si es posible mejorarlas, empleando para ello nuevas bases de datos.

2. Métodos

El método desarrollado en [1] hace uso, en primer lugar, de una base de datos de anticuerpos humanos que toma como aprendizaje. Es esta base de datos la que se va a cambiar a lo largo del trabajo. Además, se emplea una segunda base de datos para realizar las predicciones, denominada base 'test'. El método modeliza las regiones variables de los anticuerpos proponiendo una distribución gaussiana, basándose en las correlaciones fenotípicas entre aminoácidos en distintas posiciones de una misma cadena y de distintas cadenas. Las diferentes bases de datos que se van a emplear a lo largo del trabajo como bases de aprendizaje son la base de datos 'learning' empleada en [1], la base de datos creada artificialmente, obtenida a partir de 'IGOR' (*Inference and Generation of Repertoires*) [5], así como la base de datos obtenida del repositorio de secuencias recopiladas 'OAS' (*Observed Antibody Space*) [6]. En este trabajo se utilizan únicamente las cadenas pesadas (VH) de los anticuerpos para obtener los diferentes resultados, a diferencia de las cadenas VH-VL empleadas en [1], ya que los archivos de secuencias obtenidos a partir de 'IGoR' y del repositorio 'OAS' están conformados solo por esta porción del anticuerpo. En el siguiente apartado se explican detalladamente estas bases. Posteriormente, se expondrán las ideas principales del modelo desarrollado en [1], así como el concepto de *scores* presentado en [1].

2.1. Bases de datos

En este trabajo se han utilizado diferentes bases de datos que contienen secuencias de anticuerpos. Se procede en este apartado a enumerar y explicar con detalle cada una de ellas:

• Base de datos 'learning': esta base de datos fue empleada en [1] y contiene 1309 secuencias de anticuerpos humanos experimentales. A lo largo del trabajo nos referimos a este archivo como base 'learning'. Toda secuencia cuenta con una longitud de 149 aminoácidos. Esto se consigue empleando

la herramienta ANARCI [4] con el sistema de alineamiento AHo.

Con este sistema, se consiguen alinear aquellos dominios que en cuanto a función y estructura son homólogos entre secuencias. Además, poder contar con secuencias del mismo tamaño facilita el trabajo con ellas. Para lograr el alineamiento deseado, se insertan espacios, (*gaps*, en inglés) que se representan con un guión '-'. En la Figura 2 se puede ver un ejemplo de cómo quedarían alineados los extractos de dos cadenas siguiendo el sistema de alineamiento AHo.

E	V	Q	L	L	E	W	-	G	A	G	L	L	K	P	S	E	T	L	S	L	T	C	A	V	Y	G	-	G	S	F	S	G
-	-	-	L	E	E	S	-	G	G	D	L	V	Q	P	G	R	S	L	R	L	S	C	S	T	S	G	-	F	S	F	G	D

Figura 2: Ejemplo del método de alineamiento AHo. Se muestran los extractos de dos secuencias de anticuerpos de la base 'learning'

No se puede trabajar directamente con estas bases de datos conformadas por las letras de los aminoácidos correspondientes, ya que son valores de tipo *Char*. Siguiendo la estrategia usada en [2], cada aminoácido se transforma en una serie de 20 números *Floats* que toman los valores 0 o 1. De forma que los gaps se transforman en secuencias de veinte 0's, y cada una de las 20 letras correspondientes a los 20 aminoácidos que construyen nuestras bases de datos se transforman en secuencias de diecinueve 0's y la posición restante ocupada por el valor 1. Es esta posición que ocupa el 1 dentro de las 20 posibles la que indicará el aminoácido en cuestión que se está codificando. Por tanto, se tiene que cada cadena de aminoácidos es un vector real de dimensión 20L, ya que las secuencias compuestas por L residuos son transformadas a secuencias binarias, de forma que la nueva longitud adquirida las transforma en secuencias de tamaño 20L. Se puede ver en la Figura 3, suponiendo que se cuenta únicamente con 3 tipos de aminoácidos y una cadena de 5 aminoácidos, la idea de esta codificación de secuencias.

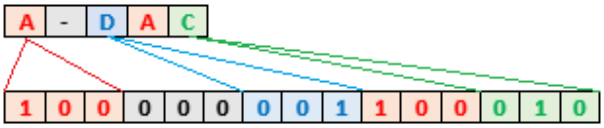


Figura 3: Ejemplo de codificación de secuencias. En este ejemplo se cuenta con 3 tipos de aminoácidos, por lo que cada letra se codifica empleando un bloque de 3 números. Además, el gap que aparece se transforma en tres 0's

• Base de datos 'test': base de secuencias humanas y murinas empleadas en [1], donde se usó de nuevo el sistema ANARCI [4] para alinear las cadenas de acuerdo al método AHo. En nuestro caso se emplea únicamente la cadena pesada (VH). Se tiene una base de datos de 1388 secuencias humanas y una base de datos de 1379 secuencias murinas con 149 aminoácidos de longitud. A lo largo del trabajo nos referimos a estos archivos como base 'test'.

• Base de datos 'IGoR': archivo de secuencias creado artificialmente mediante 'IGoR' (*Inference and Generation of Repertoires*) [5]. 'IGoR' es una herramienta de software que procesa cadenas de anticuerpos y asimila estadísticas, con un papel importante en el sistema inmunitario, para generar secuencias de anticuerpos. Esta base de datos cuenta con las cadenas VH de longitud 149

aminoácidos. A lo largo del trabajo nos referimos a este archivo como base 'IGoR'.

• Base de datos 'OAS': base obtenida del repositorio *Observed Antibody Space* (OAS) [6] donde están registradas innumerables secuencias de anticuerpos experimentales. El archivo con el que se trabaja lo forman cadenas VH de longitud de 149 aminoácidos. A lo largo del trabajo nos referimos a este archivo como base 'OAS'.

2.2. Introducción al Modelo gaussiano multivariante

Se introduce ahora el Modelo gaussiano multivariante explicado en [1]. Este modelo asume que las secuencias alineadas de residuos que conforman la base de datos siguen una distribución normal o gaussiana, dada por la siguiente notación:

$$p(x^m|\mu,\Sigma) = \mathcal{N}(\mu,\Sigma) \tag{1}$$

con {m=1,...,M} siendo M el número total de secuencias.

Se recuerda que x^m es un vector real de dimensión $20L$ como hemos justificado en la sección 2.1. A su vez, μ es el vector de medias y Σ es la matriz de covarianzas.

La media y la covarianza empíricas asociados a una base de datos vienen dadas por las siguientes expresiones:

$$\bar{x}_i = \frac{1}{M_e}\sum_{m=1}^{M} x_i^m w_m \qquad (i = 1, ..., L) \tag{2}$$

$$\bar{C}_{ij} = \frac{1}{M_e}\sum_{m=1}^{M} (x_i^m - \bar{x}_i)(x_j^m - \bar{x}_j) w_m \tag{3}$$

donde M es el número de secuencias totales que tenemos en nuestra base de datos, $M_e = \sum_{m=1}^{M} w_m$ y el parámetro w_m es el peso asociado a cada cadena.

Será útil a lo largo del trabajo el uso de la norma de Frobenius de la matriz de la covarianza pesada. Su expresión viene dada por:

$$\|\bar{C}\| = \sqrt{\sum_{i=1}^{20L}\sum_{j=1}^{20L} (\bar{C}_{ij})^2} \tag{4}$$

A continuación, se define el peso w_m de la siguiente manera:

$$\omega_m = \frac{1}{1 + \sum_{l\neq m} \theta(\alpha_{lm} - \Omega L)} \tag{5}$$

donde θ es la función escalón.

En el argumento de la función escalón de la ecuación (5) aparece el factor de similitud entre secuencias α_{lm}. El valor α_{lm} se establece teniendo en cuenta los aminoácidos idénticos entre cadenas, excluyendo los gaps, de la siguiente forma:

$$\alpha_{lm} = \sum_{i=1}^{L} \delta_{A_i^l A_i^m}(1 - \delta_{A_i^l, '-'}) \tag{6}$$

donde A_i^m es el aminoácido de la posición i de la secuencia m y δ_{ij} es la conocida como delta de Kronecker.

Así, se tiene una función escalón positiva cuando la similitud entre secuencias supere o sea igual al umbral establecido como $L\Omega$, donde Ω es un parámetro que junto a los pesos w_m es introducido para subsanar una posible falta de independencia entre secuencias. De manera que se pesan menos aquellas secuencias que sean similares entre sí.

Para inferir los valores más idóneos μ y Σ de una base de datos dada, se asume que estos parámetros siguen una distribución previa de tipo *Wishart Normal Inversa* (*NIW*):

$$p^{pr}(\mu, \Sigma) = NIW(\eta, \kappa, \Lambda, \nu) = N(\mu|\eta, \frac{\Sigma}{\kappa})IW(\Sigma|\Lambda, \mu) \tag{7}$$

Usando el teorema de Bayes, se puede calcular la distribución a posteriori, empleando de nuevo la distribución *NIW*:

$$p^{post}(\mu, \Sigma|X) \propto p(X|\mu, \Sigma)p^{pr}(\mu, \Sigma) = NIW(\eta', \kappa', \Lambda', \nu') \tag{8}$$

donde X es la base de datos con la que estamos trabajando. Tras realizar un desarrollo expuesto en [1], se deduce:

$$\langle\mu\rangle_{post} = \lambda\eta + (1-\lambda)\bar{x} \tag{9}$$

$$\langle\Sigma\rangle_{post} = \lambda U - (1-\lambda)\bar{C} + \lambda(1-\lambda)(\bar{x}-\eta)(\bar{x}-\eta)^T \tag{10}$$

Se eligen los parámetros η y U como los correspondientes a la media y la covarianza de una muestra uniformemente distribuida. El valor que tome el parámetro λ indicará cuánto peso se otorga a la probabilidad previa, siendo esta potenciada si λ es próximo a 1 e insignificante cuando λ es próximo a 0, dando en este caso el mayor peso a las variables empíricas de la media y la covarianza.

Finalmente en [1] se hace uso del concepto denominado 'puntuación de humanidad' (*Humanness score*, en inglés), cuya función es la distinción de secuencias humanas de murinas. Matemáticamente, esta puntuación se obtiene de la siguiente forma: dada una base de datos X, la distribución predictiva posterior para una nueva secuencia y viene dada por:

$$p(y|X) = t_N\left(\frac{M}{1-\lambda} + 2, \langle\mu\rangle_{post}, \left(1 + \frac{1-\lambda}{M}\right)\langle\Sigma\rangle_{post}\right) \tag{11}$$

donde se ha introducido la densidad de probabilidad de la distribución t multivariante:

$$t_p(\rho, \mu, S) = \frac{\Gamma(\frac{\rho+p}{2})}{\Gamma(\frac{\rho}{2})(\rho\pi)^{p/2}}|S|^{-\frac{1}{2}}\left(1 + \frac{1}{\rho}(y-\mu)^T S^{-1}(y-\mu))\right)^{-\frac{\rho+p}{2}} \tag{12}$$

La puntuación de humanidad de una secuencia y viene dada por:

$$\log p(y|X) \tag{13}$$

3. Resultados

En este apartado se presentan los resultados obtenidos tras evaluar la capacidad predictiva del modelo empleando diferentes bases de datos. La capacidad predictiva es la efectividad que tiene el modelo clasificando anticuerpos entre las categorías de humanos y murinos. Se utilizan la base 'learning' que ya fue empleada en [1], la base 'IGoR' [5] y la base 'OAS' [6] como bases de aprendizaje.

3.1. Clasificación usando la base 'learning'

En primer lugar, se presenta la capacidad predictiva trabajando con la base de datos 'learning'. Estos resultados ya fueron calculados en [1] y [7] para la cadena VH-VL completa. En este trabajo se obtienen los resultados trabajando únicamente con la cadena VH, los cuales nos servirán como base comparativa cuando se trabaje con los nuevos elencos de secuencias. Además, se evalúa cómo cambian estas predicciones si se modifican los parámetros Ω y λ, cuya definición puede consultarse en la sección 2.2.

Para evaluar la capacidad predictiva, lo primero será hallar los parámetros de Ω y λ óptimos para la base 'learning'. Como ya se comentó en la sección 2.2, el parámetro Ω se emplea para asignar pesos a las diferentes secuencias que conforman la base de datos para evitar posibles sesgos estadísticos, dando un peso inferior a aquellas secuencias que sean muy similares entre sí. El valor de Ω óptimo para una base de datos dada puede estimarse como aquel que maximiza la norma de la covarianza pesada ($\|\bar{C}\|$), asociada a esta misma base y cuya expresión se especifica en la ecuación (4) de la sección 2.2. Mediante un código en Python obtenido de [8], en la Figura 4 se muestra $\|\bar{C}\|$ en función de Ω.

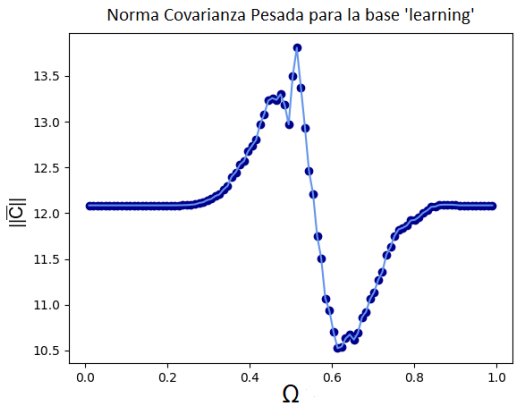
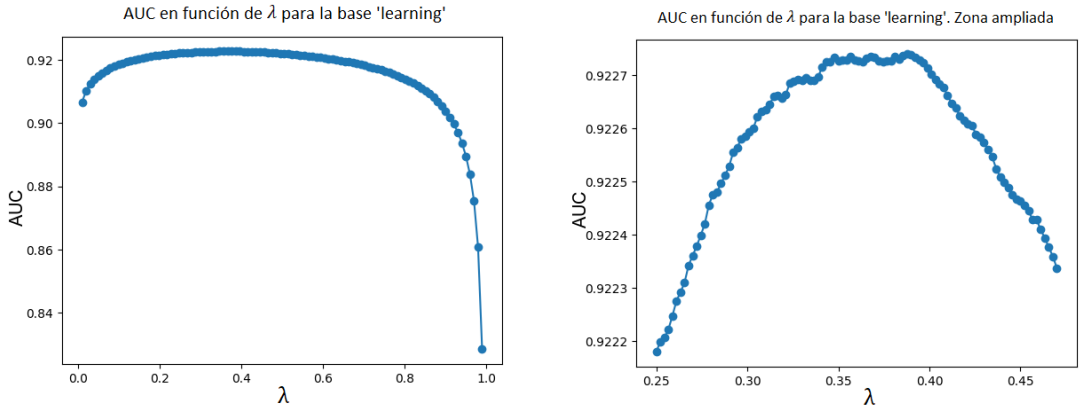


Figura 4: Norma de la covarianza pesada en función de Ω para la base 'learning'. El valor de Ω óptimo será aquel que maximice la norma de la covarianza pesada

Se observa cómo el valor basal coincide tanto para valores de Ω pequeños como valores de Ω elevados cercanos a 1. El valor de la norma de la covarianza pesada alcanza su máximo para Ω=0.513, siendo este, por tanto, su valor óptimo.

Se evalúa ahora el parámetro λ. El valor de λ óptimo es aquel que maximiza el área bajo la curva ROC (AUC) (véase Anexo 5.1). Para obtener estas curvas, se utiliza uno de los códigos en

Python obtenido de [8]. Se emplea en el modelo la base 'learning' y se recorren los distintos valores de λ, calculando para cada uno de ellos, los *scores*, véase apartado 2.2, de la base 'test'. El resultado se presenta en la Figura 5 donde el valor óptimo viene dado por λ=0.390.



(a) AUC en función de λ

(b) AUC en función de λ. Zona ampliada

Figura 5: AUC en función del parámetro λ para la base 'learning'. El valor de λ se estima como aquel que es capaz de maximizar el valor de AUC

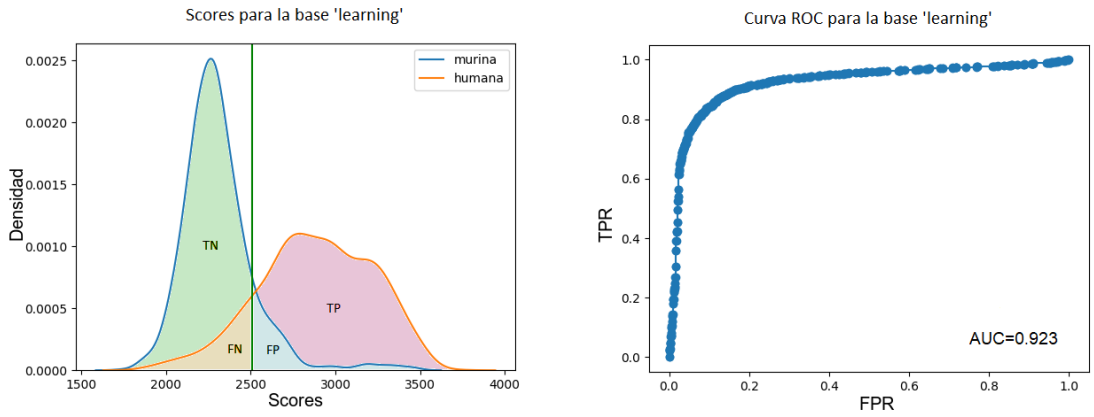
A continuación, se trata de optimizar los parámetros Ω y λ utilizando otro enfoque. Se realiza un barrido de los valores de dichos parámetros, calculando el correspondiente valor de AUC asociado a cada pareja de valores Ω y λ. Se obtiene que AUC disminuye para valores de λ extremos, especialmente para valores de λ cercanos a 1, lo que pone de manifiesto que dar todo el peso a la probabilidad previa y ningún peso a las variables empíricas empeora AUC. Con valores de $\lambda \approx 0.01$ ocurre lo opuesto, el peso en su mayoría es otorgado a las variables empíricas, y el valor de AUC también se reduce. Se obtiene que AUC es ligeramente más alta para valores de Ω entre 0.30 y 0.55 y a su vez para valores de λ entre 0.35 y 0.65. En este rango se encontraban los valores de Ω y λ considerados como óptimos usando el criterio que maximiza la norma de la covarianza pesada. Más allá de este comportamiento genérico, no se ha podido estimar unos nuevos valores fiables de Ω y λ haciendo uso del AUC. Por lo tanto, se procede a evaluar la capacidad predictiva del modelo usando los valores Ω=0.513 y λ=0.390 obtenidos anteriormente.

Para evaluar la capacidad predictiva se hace uso de la definición de *scores* (véase apartado 2.2). Asimismo, se obtiene la curva ROC correspondiente y se calcula el valor del umbral óptimo. Este es el punto de la curva ROC con el valor máximo del índice de Youden (YI)(véase Anexo 5.1). Se obtiene como valor umbral U=2506.83. De manera que todos los anticuerpos con *scores* superiores al establecido por el umbral serán considerados como anticuerpos humanos, mientras que aquellos con *scores* inferiores se consideran anticuerpos murinos. Se obtiene que el área bajo la curva ROC es AUC=0.923. Esto significa que existe un 92.3 % de probabilidad de que dados dos anticuerpos, uno humano y otro murino, el modelo los clasifique correctamente.

En la Tabla 1 se reflejan los valores concretos de la capacidad predictiva para la base 'learning' y en la Figura 6 se muestran las gráficas de *scores* y la curva ROC obtenida.

M_{humana}	M_{murina}	TP	FP	TN	FN	AUC	YI
1388	1379	1204	168	1211	184	0.923	0.746

Tabla 1: Valores ROC para la clasificación usando la base 'learning'. M_{humana} corresponde al número de secuencias humanas de la base 'test', M_{murina} simboliza el número de secuencias murinas de la base 'test', TP expresa los verdaderos positivos, FP los falsos positivos, TN los verdaderos negativos, FN los falsos negativos, AUC es el área bajo la curva ROC y finalmente YI simboliza el índice de Youden



(a) *Scores* (b) Curva ROC

Figura 6: Capacidad predictiva para la base de datos 'learning'. En el panel izquierdo se muestra la distribución de *scores* para los anticuerpos humanos y murinos. La línea azul representa los anticuerpos murinos, la línea naranja representa los anticuerpos humanos y la línea verde vertical establece el valor umbral. En el panel derecho, se representa la curva ROC correspondiente

Finalmente, en la Tabla 2 se muestra la comparación entre los valores de los parámetros Ω, λ y los valores ROC que se han obtenido empleando únicamente la cadena VH y los resultados que se obtuvieron en el trabajo [1], utilizando la cadena VH-VL completa.

Cadena	Ω	λ	AUC	YI
VH	0.513	0.390	0.923	0.746
VH-VL	0.489	0.1	0.969	0.891

Tabla 2: Comparación de resultados de la cadena VH y la cadena VH-VL. Se confrontan los parámetros Ω, λ y los valores ROC que se han obtenido usando la cadena VH con los resultados que se obtuvieron en [1] para la cadena completa VH-VL

Se observa que mientras el parámetro Ω no difiere de manera excesiva entre ambos casos, el parámetro λ obtenido para la cadena VH-VL en [1] toma un valor que le da más peso a las variables empíricas de la media y la covarianza que el obtenido para la cadena VH. Además, los valores ROC son mejores para la cadena VH-VL.

3.2. Clasificación usando la base 'IGoR'

En este apartado se exponen los resultados obtenidos trabajando con una nueva base de datos obtenida a partir de 'IGoR' [5] que contiene 40000 secuencias. Se trabaja con este subconjunto de secuencias del total de las 99000 generadas mediante 'IGoR', ya que el ordenador utilizado en el análisis no poseía la capacidad de memoria suficiente como para trabajar con un número de secuencias tan elevado. Se pretende comprobar si con esta nueva base la capacidad predictiva del modelo mejora.

En primer lugar, se analiza la norma de la covarianza pesada en función de Ω con el fin de determinar el valor de Ω óptimo. Su comportamiento inesperado nos llevará a hacer un estudio exhaustivo de la base 'IGoR'. Se va a averiguar que las secuencias que conforman esta base contienen un elevado número de gaps y se analizará el efecto que esto tiene tanto en el comportamiento de la norma de la covarianza como en la capacidad predictiva en función del número máximo de gaps permitido por secuencia. Cuando se haga posteriormente referencia al número máximo de gaps permitido por secuencia, se empleará la notación N_g.

En la Figura 7 se muestra la gráfica de la norma de la covarianza pesada en función de Ω.

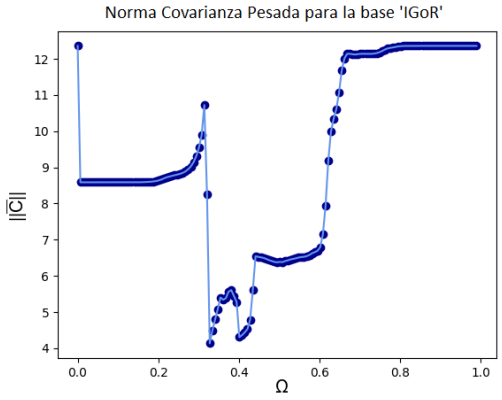
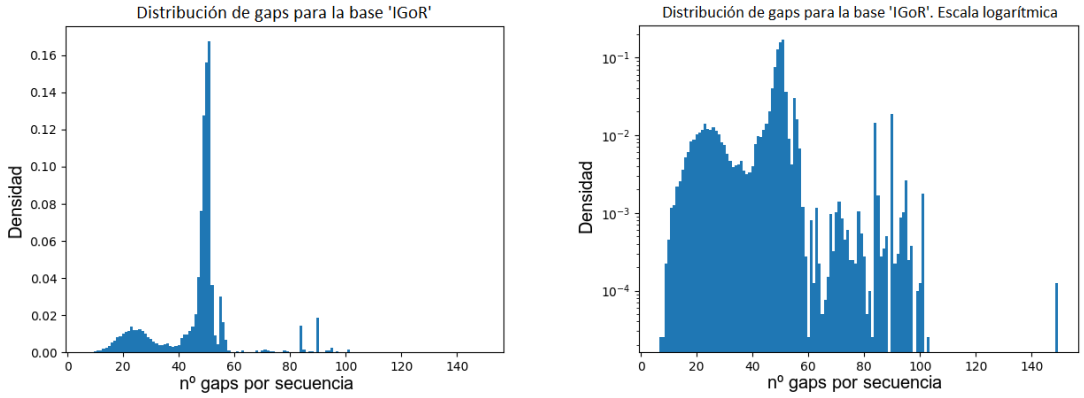


Figura 7: Norma de la covarianza pesada en función de Ω para la base 'IGoR'

Como se observó en [1] y [7] y se ha obtenido en la sección 3.1, la forma de la Figura 7 difiere totalmente de la obtenida en la Figura 4, donde se empleó la base 'learning'. El valor de la norma de la covarianza pesada debe coincidir para valores extremos del parámetro Ω. En este caso, aunque se cumple, se observa un descenso abrupto de dicho valor para valores de Ω pequeños. El comportamiento obtenido en la Figura 7 no es un hecho puntual. Se han escogido diferentes subconjuntos de la base 'IGoR' de 99000 secuencias y el resultado obtenido es similar al de la Figura 7. Ante el comportamiento inesperado, se procede a examinar la base 'IGoR' y se descubre que existen varias secuencias que poseen un número de gaps elevado. Por tanto, se analiza cuántos gaps tienen por secuencia el elenco de la base 'IGoR' y el resultado se representa en la Figura 8.



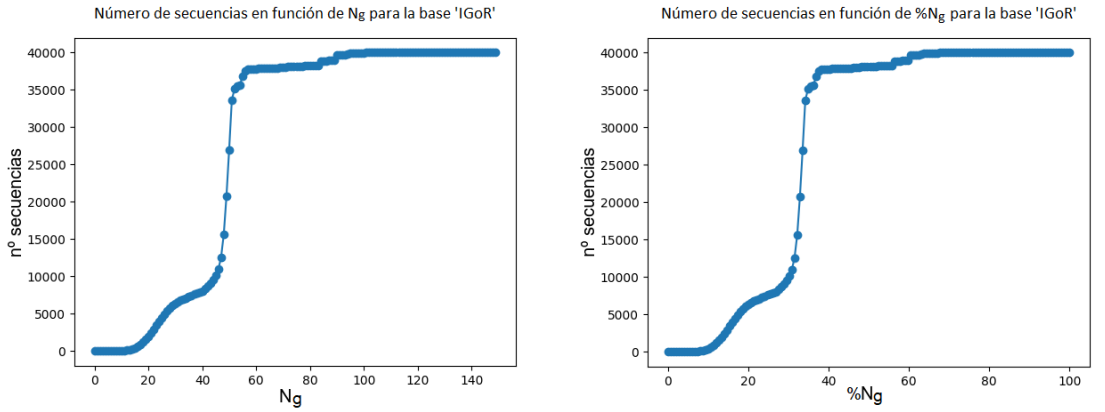
(a) Distribución de gaps

(b) Distribución de gaps. Escala logarítmica

Figura 8: Histograma del número de gaps por secuencia para la base 'IGoR'. En la gráfica de la izquierda se ve la distribución del número de gaps por secuencia para la base 'IGoR'. En la gráfica de la derecha observamos los mismos datos pero con el eje vertical en escala logarítmica

En la Figura 8 (a) se observa que la mayoría de las secuencias tienen en torno a 50 gaps. Sin embargo, en la Figura 8 (b), se revela que existen secuencias con un número muy elevado de gaps. Incluso varias secuencias, en concreto 5, están conformadas completamente por todo gaps.

En la Figura 9 se representa el número de secuencias que tendría nuestra base de datos en función de N_g, así como del porcentaje asociado a este parámetro.



(a) Número de secuencias en función de N_g

(b) Número de secuencias en función de $\%N_g$

Figura 9: Número de secuencias en la base 'IGoR' en función de N_g y $\%N_g$

Se quiere averiguar si el hecho de contar con secuencias con un número elevado de gaps, incluso unas pocas formadas al completo por gaps, puede ser la razón de la distorsión de la Figura 7. Por tanto, para profundizar en este efecto y poder así quedarnos con una base de secuencias 'fiables' con las que trabajar, se realiza un barrido del comportamiento de la gráfica de la norma de la covarianza pesada en función de N_g.

Se inicia el barrido eliminando solo aquellas 5 secuencias formadas por todo gaps del elenco total de 40000. Con esta nueva base desaparece el efecto de la caída en la norma de la covarianza pesada para valores pequeños de Ω. El efecto de eliminar solo 5 secuencias formadas por todo gaps cambia sustancialmente el comportamiento, que puede visualizarse en la Figura 10.

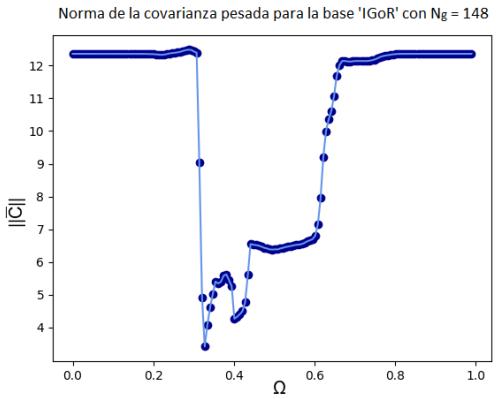
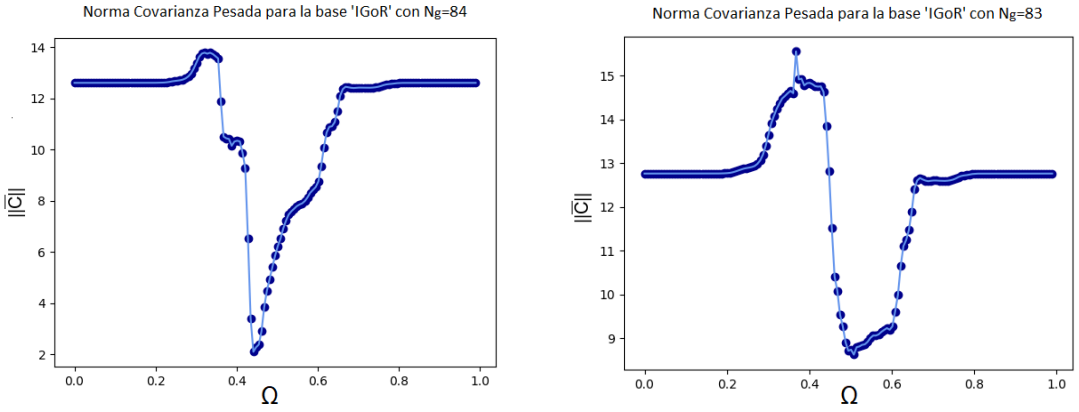


Figura 10: Norma de la covarianza pesada en función de Ω para la base 'IGoR' con $N_g = 148$. Se han eliminado solo 5 secuencias formadas por todo gaps del conjunto total compuesto por 40000 secuencias

Se sigue realizando el barrido, siendo más restrictivos con el valor de N_g. En la Figura 9 se encuentra un ligero descenso en el número de secuencias, cuando se pasa de un valor N_g=84 a un valor N_g=83, en la zona de $\approx 56\,\%$. La forma de la gráfica de la norma de la covarianza pesada en función de Ω se tranforma como se muestra en la Figura 11.



(a) Norma de la covarianza pesada para N_g=84 (b) Norma de la covarianza pesada para N_g=83

Figura 11: Cambio en la gráfica de la norma de la covarianza pesada para la base 'IGoR'. La gráfica cambia de forma tras pasar de una base con un valor N_g=84 (izqa.) a otra con un valor N_g=83 (dcha.). Se han eliminado 572 secuencias con 84 gaps para obtener la gráfica de la derecha

La Figura 11 (a), correspondiente a N_g=84, contiene 38831 secuencias. En ella se aprecia la

aparición de un pequeño máximo en torno a Ω=0.35 y un mínimo en torno a Ω=0.4. La Figura 11 (b) contiene un total de 38259 secuencias y se asemeja en cuanto a forma a la Figura 4 obtenida con la base 'learning'. Aparece un máximo del valor de la norma de la covarianza pesada en torno a Ω=0.4, seguida de un mínimo en torno a Ω=0.5. Se observa, de nuevo, cómo el leve cambio de pasar de una base con N_g=84 a otra base con N_g=83, lo que supone prescindir de 572 secuencias de un total de ≈ 38000, produce un cambio importante en la forma de la gráfica de la norma de la covarianza pesada.

Se sigue limitando el valor N_g, para ver cómo evoluciona el comportamiento de $\|\bar{C}\|$. El siguiente cambio relevante se encuentra en la zona con $N_g = 40$ (≈ 27 %) donde se reduce la base 'IGoR' a solamente 8037 secuencias. Por motivos de espacio, el comportamiento en la zona entre N_g=83 y N_g=40 no cuenta con la importancia suficiente para mostrarse aquí, pero puede consultarse en el Anexo 5.2. Se presenta en la Figura 12 la gráfica de la norma de la covarianza pesada para N_g=40, la cual se hace máxima en Ω=0.489 y cuyo mínimo aparece en Ω=0.664.

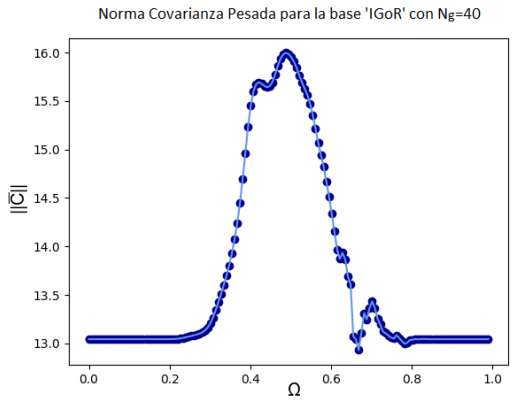
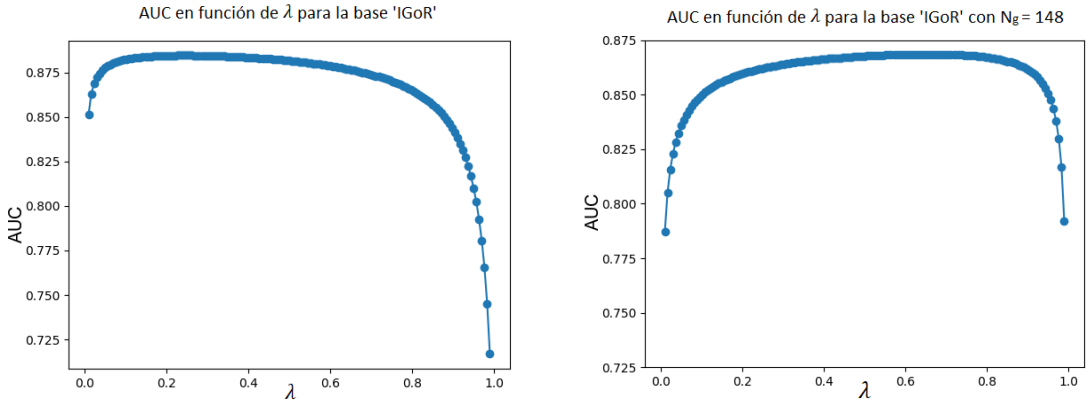


Figura 12: Norma de la covarianza pesada en función de Ω para la base 'IGoR' con $N_g = 40$

Se ha descubierto que si se limita el valor de N_g, los valores del parámetro Ω para los cuales se hace máxima y mínima la norma de la covarianza pesada se desplazan hacia la derecha. Esta evolución se visualiza en la Figura 30 del Anexo 5.2. Siendo más restrictivos con el valor de N_g, con valores por debajo de N_g=40, las gráficas obtenidas no difieren en exceso de la Figura 12, haciéndose la norma de la covarianza pesada de nuevo máxima para valores en torno a Ω=0.5.

Tras haber realizado un análisis exhaustivo de lo que ocurre con el parámetro Ω, se analiza, a continuación, lo que sucede con el parámetro λ. Se pretende ver cómo evoluciona el valor de λ óptimo en función del valor de N_g. En la Figura 13 (a) se muestra la gráfica del AUC en función de λ, donde se ha utilizado la base 'IGoR' inicial compuesta por 40000 secuencias y Ω = 0.3, valor en torno al cual se hace máxima la norma de la covarianza pesada, cuando se emplea la base 'IGoR' con un N_g elevado y poco restrictivo. En este caso, se obtiene que el valor óptimo de λ que maximiza el valor de AUC es λ=0.238. Por tanto, se otorga en este caso un mayor peso a las variables empíricas de la media y la covarianza. Asimismo, se muestra en la Figura 13 (b) la gráfica de AUC en función de λ como resultado de eliminar las 5 secuencias compuestas por todo gaps del elenco total de la base 'IGoR'. De nuevo tomando Ω = 0.3, el valor óptimo se sitúa en λ= 0.647.

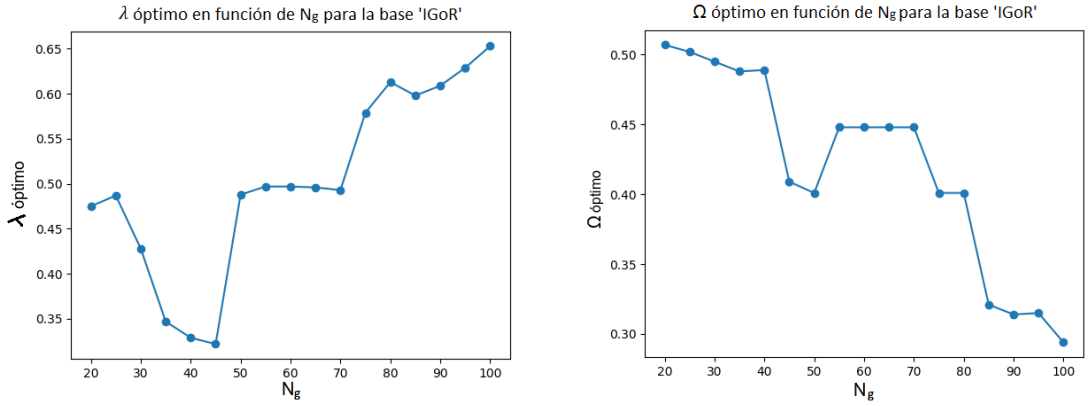


(a) AUC en función de λ para base 'IGoR' (b) AUC en función de λ para base 'IGoR' con N_g=148

Figura 13: AUC en función de λ para la base 'IGoR'. En el panel de la izquierda se representa AUC en función de λ para la base 'IGoR' inicial y en el panel de la derecha se muestra esta misma representación para la base 'IGoR' restringida a N_g=148

El comportamiento de AUC tras eliminar las 5 secuencias cambia sustancialmente. Eliminar únicamente 5 secuencias, supone que ahora el peso recaiga mayormente en la probabilidad previa, y en consecuencia un cambio total en el valor de λ óptimo.

Nos preguntamos así cómo va a cambiar el valor de λ óptimo si se continúa restringiendo el valor de N_g. En la Figura 14 (a) se muestran los diferentes valores de λ óptimo obtenidos en función del valor N_g. En cada uno de los casos se ha empleado el correspondiente valor de Ω que maximiza la norma de la covarianza. Se muestra en la Figura 14 (b), además, la evolución de este parámetro Ω en función de N_g.



(a) $\lambda_{óptimo}$ en función de N_g (b) $\Omega_{óptimo}$ en función de N_g

Figura 14: Evolución de los parámetros λ y Ω en función de N_g para la base 'IGoR'. En el panel de la izquierda se muestra λ óptimo en función de N_g y en el panel de la derecha se muestra Ω óptimo en función de N_g

Asimismo, se adjuntan en la Tabla 3 el número de secuencias M y los correspondientes valores de Ω y λ óptimos representados en la Figura 14, según el valor de N_g empleado. También se incluyen los valores de AUC e YI en función del parámetro N_g que se calculan posteriormente en el apartado 3.2.1.

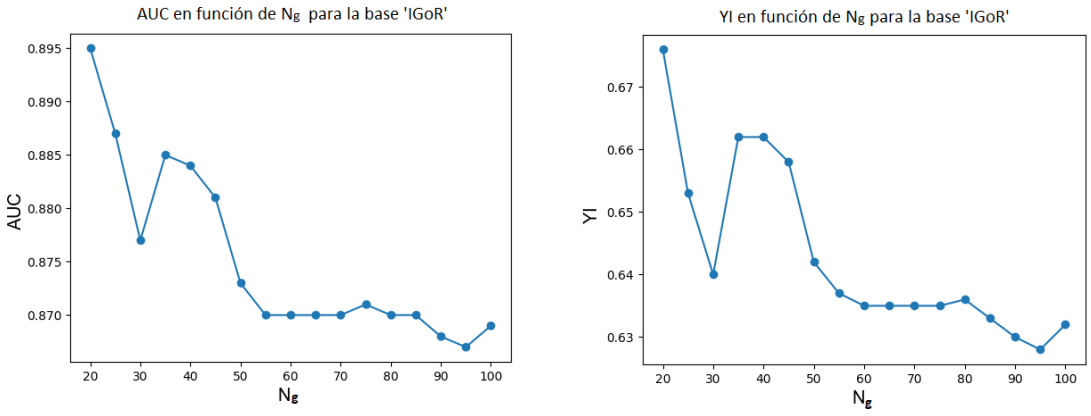
N_g	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
M	1998	4408	6391	7294	8037	10137	26962	36847	37819	37914	38016	38158	38252	38898	39689	39890	39924
$\Omega_{óptimo}$	0.507	0.502	0.495	0.488	0.489	0.409	0.401	0.448	0.448	0.448	0.448	0.401	0.401	0.321	0.314	0.315	0.294
$\lambda_{óptimo}$	0.475	0.487	0.428	0.347	0.329	0.322	0.488	0.497	0.497	0.496	0.493	0.579	0.613	0.598	0.609	0.629	0.653
AUC	0.895	0.887	0.877	0.885	0.884	0.881	0.873	0.870	0.870	0.870	0.870	0.871	0.870	0.870	0.868	0.867	0.869
YI	0.676	0.653	0.640	0.662	0.662	0.658	0.642	0.637	0.635	0.635	0.635	0.635	0.636	0.633	0.630	0.628	0.632

Tabla 3: Valores de $\Omega_{óptimo}$, $\lambda_{óptimo}$, AUC e índice de Youden (YI) en función de N_g, así como el número de secuencias (M) correspondiente para cada caso

Observando la Figura 14, así como la Tabla 3, se pueden destacar varios puntos. En primer lugar, se percibe, que al contrario de lo que sucedía al eliminar las 5 secuencias con todo gaps, restrigir N_g entre los valores de 100 y 45 supone una tendencia genérica a que el valor de λ óptimo se reduzca. De esta manera, alcanza su valor mínimo en torno a N_g=45, donde toman mayor peso las variables empíricas. Seguir reduciendo el valor de N_g más allá de N_g=40 supone, en cambio, que el valor de λ óptimo vuelva a aumentar. Tras evaluar el comportamiento de Ω y λ para la base 'IGoR' se procede en el siguiente apartado a analizar la capacidad predictiva.

3.2.1. Capacidad predictiva para la base 'IGoR'

Con lo visto anteriormente, se plantea cómo va a variar la capacidad de predicción del modelo para la base 'IGoR' en función de N_g. Se toman una serie de 17 puntos que van desde N_g=20 a N_g=100 de 5 en 5. Se calculan los valores de AUC para cada una de las bases 'IGoR' restringidas al valor de N_g correspondiente y se toma el valor de los parámetros Ω y λ obtenidos para cada base, véase la Tabla 3 para consultar dichos parámetros.



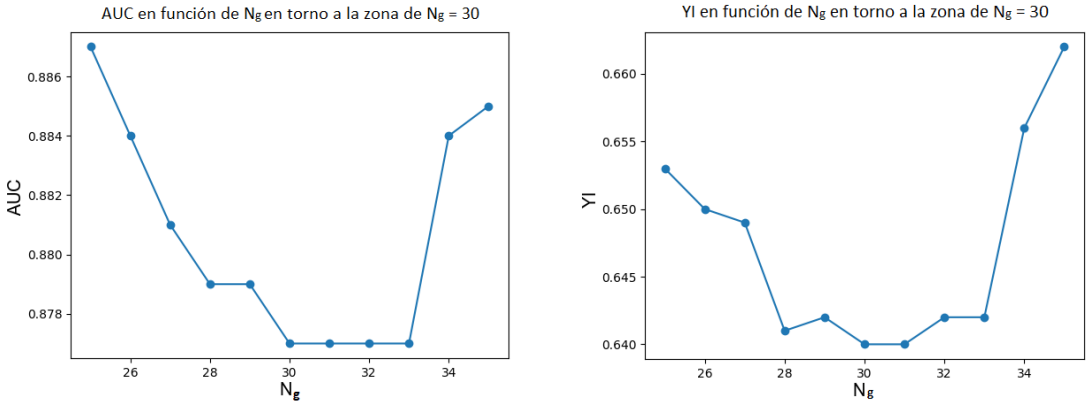
(a) AUC en función N_g (b) YI en función de N_g

Figura 15: Estudio de la capacidad predictiva para la base 'IGoR', en función del parámetro N_g

Los valores de AUC e YI correspondientes a la Figura 15 se adjuntan en la Tabla 3. El valor de AUC y del índice de Youden (YI) desciende a medida que N_g aumenta. El punto que corresponde a N_g=20 tiene tanto el área AUC como el índice de Youden (YI) más elevados. Cabe destacar que aparece un mínimo local en N_g=30. La capacidad predictiva empeora si se consideran secuencias con un número de gaps elevado, a pesar de que estas secuencias supongan un porcentaje pequeño del total (véase Figura 9). Se puede deducir que existe un problema en la generación de secuencias mediante 'IGoR'. Este programa genera secuencias incompletas que ANARCI completa con demasiados gaps, con lo cual, la base 'IGoR' contiene muchas secuencias con muy poca información que se deben eliminar.

Desafortunadamente, los resultados para la base 'IGoR' no logran mejorar los resultados obtenidos para la base 'learning', siendo de hecho peores resultados. Como se muestra en la Figura 15 y en la Tabla 3, para la base 'IGoR' los valores de AUC se encuentran en un rango entre 0.87 y 0.9 y los valores del índice de Youden(YI) se sitúan en un rango entre 0.63 y 0.68, mientras para la base 'learning' se habían obtenido AUC=0.923 y YI= 0.746 (véase Tabla 1).

Cabe destacar el mínimo local inesperado que se aprecia en torno a $N_g = 30$. Aunque se trata de una fluctuación pequeña, este hecho nos desconcierta. Las gráficas de la Figura 15 parecen mostrar un aumento en la capacidad predictiva a medida que se reduce el número de gaps permitidos por secuencia (N_g), pero justo en esta zona, reducir el número de gaps permitidos empeora la capacidad predictiva momentáneamente. Se analiza con mayor profundidad lo ocurrido en esta zona, mostrando en la Figura 16 qué ocurre en los puntos en torno a N_g=30, en concreto entre N_g=25 y N_g=35.



(a) AUC en función N_g

(b) YI en función de N_g

Figura 16: Estudio de la capacidad predictiva para la base 'IGoR', en función del parámetro N_g en la zona en torno a N_g=30

Se aprecia con más detalle, en la Figura 16, el descenso que se producía en la capacidad predictiva en torno a los N_g=30 para la Figura 15. Esto pone de manifiesto que existe un comportamiento aparentemente anómalo en esta zona, aunque se desconoce a qué se debe.

Tras el análisis realizado y mostrado en la Figura 15 y la Tabla 3, donde se observa que reduciendo N_g a valores por debajo de 45 la capacidad predictiva mejora, cabe realizar un análisis final donde

tomando el conjunto total de 99000 secuencias generadas mediante 'IGoR', se realice un filtrado para varios valores de N_g con los que se ha visto que la capacidad predictiva mejora. Se escogen los valores de N_g=20, N_g=25, N_g=35, N_g=40 y N_g=45. Así tendremos un mayor número de secuencias en nuestra base de datos y analizaremos si de esta forma se logran obtener mejores resultados para la capacidad predictiva. En la Tabla 4 se adjuntan los valores de AUC e YI obtenidos, así como el conjunto de secuencias (M) para cada valor de N_g correspondiente.

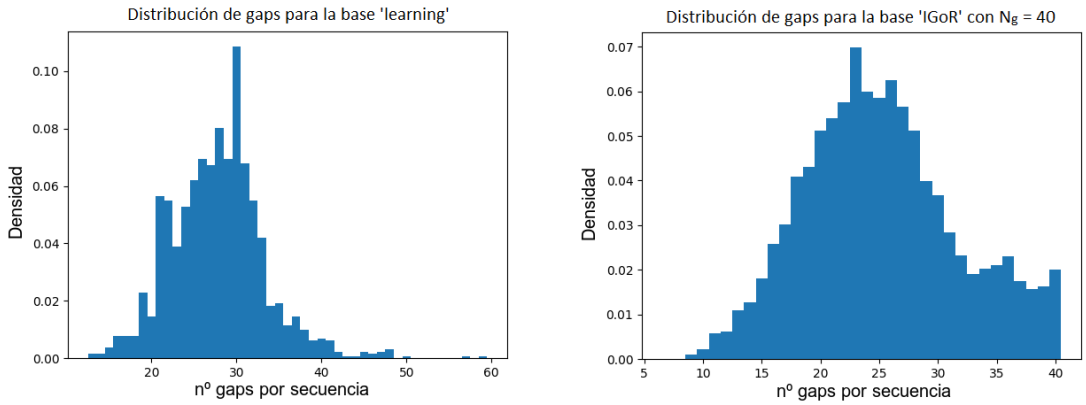
N_g	M	AUC	YI
20	4866	0.895	0.671
25	10870	0.886	0.652
35	18049	0.886	0.662
40	19831	0.885	0.662
45	24904	0.881	0.656

Tabla 4: Valores del AUC e índice de Youden (YI) en función de N_g para la base 'IGoR' con 99000 secuencias

Si se comparan la Tabla 3 y la Tabla 4 para los valores de N_g presentados en esta última no se observa que la capacidad predictiva mejore si aumenta el número de secuencias que conforman la base de datos y por tanto el resultado extraído es que con la base 'IGoR' no se consiguen igualar ni mejorar los resultados obtenidos para la base 'learning' (véase Tabla 1).

3.2.2. Profundizando en la base 'IGoR'

Con el objetivo de profundizar en el motivo del comportamiento anómalo debido al elevado número de gaps por secuencia en la base 'IGoR', se realiza un análisis tanto de la distribución de gaps para la base 'IGoR' con $N_g = 40$ como de las distancias de hamming. Se compara en la Figura 17 la distribución del número de gaps para la base 'learning' y la base 'IGoR' con $N_g = 40$.

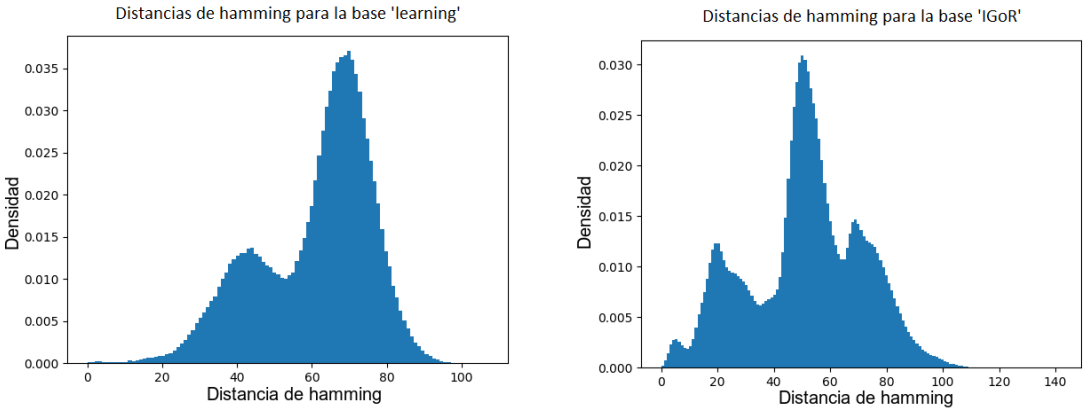


(a) Distribución de gaps. Base 'learning' (b) Distribución de gaps. Base 'IGoR'

Figura 17: Histograma del número de gaps por secuencia. En el panel de la izquierda se muestra la distribución de gaps por secuencia para la base 'learning' y en el panel de la derecha la distribución de gaps por secuencia para la base 'IGoR' con N_g=40

En la Figura 17 se observa cómo, en el caso de la base 'learning', la mayoría de las secuencias cuentan con un número de gaps en torno a 30, mientras que para la base 'IGoR' con N_g=40, la mayoría de secuencias cuentan con un número de gaps en torno a 25. Se recuerda que utilizando todo el conjunto 'IGoR' sin restringir a un valor de N_g concreto, se obtenía la Figura 8, donde la mayoría de secuencias tenían en torno a 50 gaps.

Con el fin de seguir explorando el efecto del número de gaps por secuencia, se analizan las distancias de hamming para la base 'IGoR' inicial (con 40000 secuencias), así como para la base 'learning'. Las distancias de hamming se calculan entre cada una de las secuencias con el resto de secuencias que conforman la base de datos.



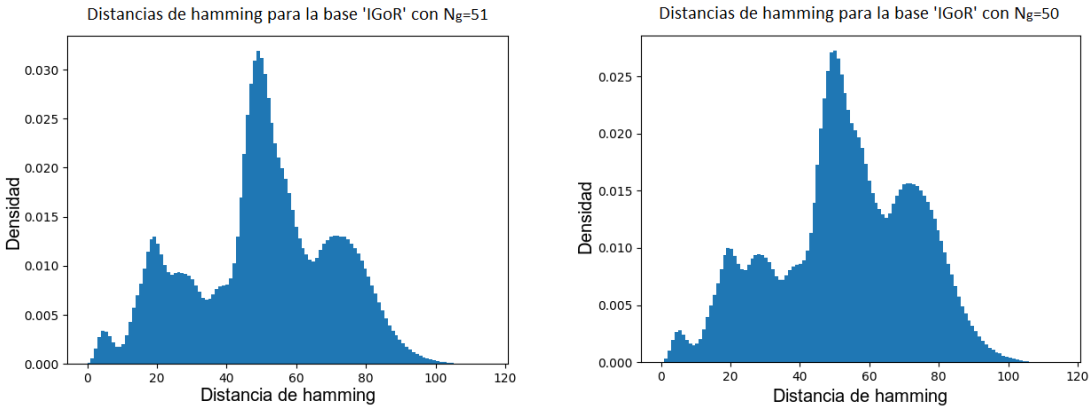
(a) Distancia de hamming. Base 'learning' (b) Distancia de hamming. Base 'IGoR'

Figura 18: Distancias de hamming para la base 'learning' y la base 'IGoR'

En la Figura 18 se observa un comportamiento sustancialmente diferente entre las distancias

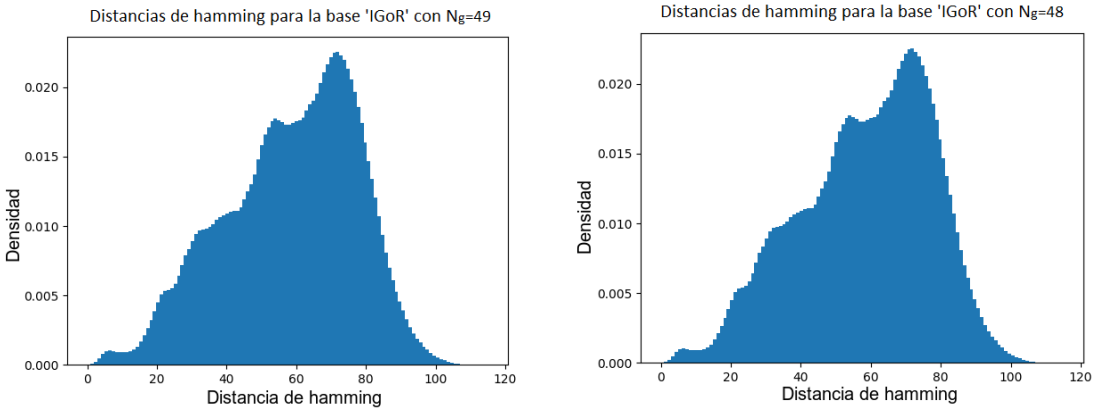
de hamming para ambas bases. Mientras para la base 'learning' aparecen 2 picos, para la 'IGoR' aparecen 4 picos. Esta diferencia de comportamiento, probablemente, se debe al hecho de que existen secuencias con un número de gaps muy alto en la base 'IGoR', de manera que los valores de las distancias de hamming pueden salir distorsionados respecto a los obtenidos en la base 'learning'.

Cabe preguntarnos cómo varía la gráfica de las distancias de hamming si se emplea el archivo de secuencias 'IGoR' eliminando aquellas 5 secuencias formadas totalmente por gaps. El resultado es similar al visualizado en la Figura 18 (b). De hecho, la forma se mantiene similar a esta figura hasta limitar la gráfica a valores en torno a N_g=50, donde la forma de la gráfica comienza a evolucionar como se muestra en las Figuras 19, 20 y 21.



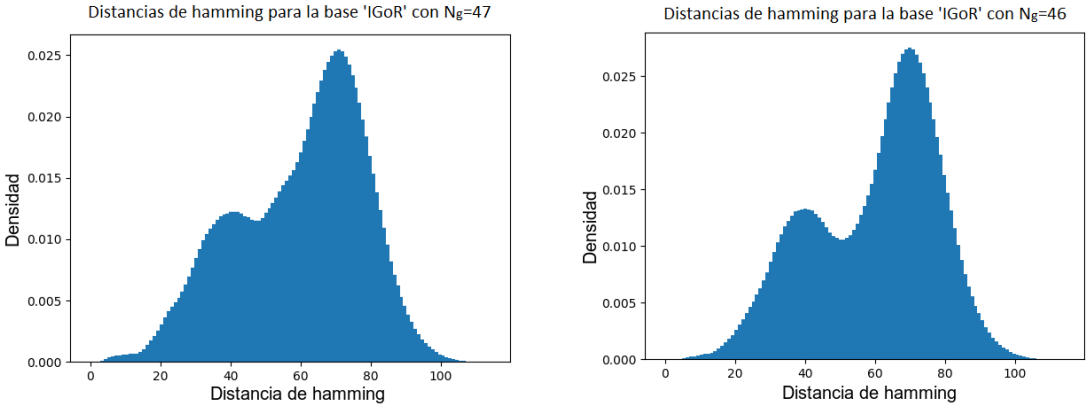
(a) Base 'IGOR' con N_g=51 (b) Base 'IGOR' con N_g=50

Figura 19: Distancias de hamming para la base 'IGoR' restringida a los valores de N_g=51 y N_g=50



(a) Base 'IGOR' con N_g=49 (b) Base 'IGOR' con N_g=48

Figura 20: Distancias de hamming para la base 'IGoR' restringida a los valores de N_g=49 y N_g=48

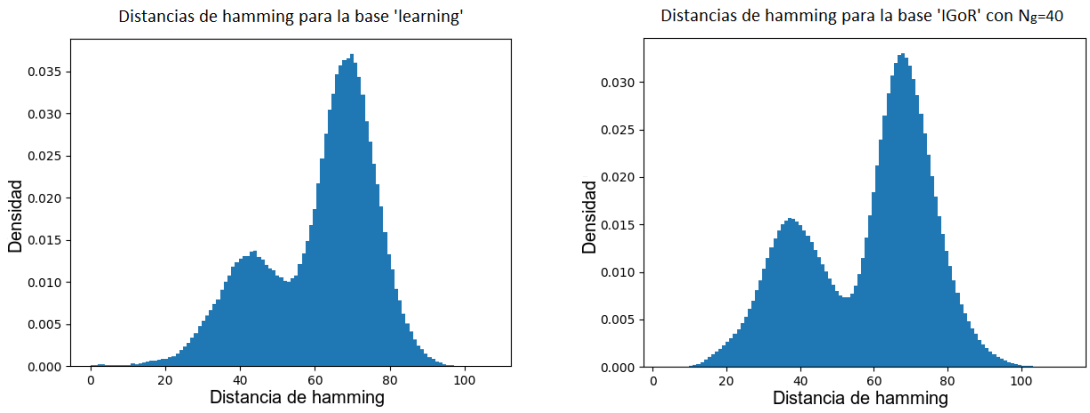


(a) Base 'IGOR' con N_g=47 (b) Base 'IGOR' con N_g=46

Figura 21: Distancias de hamming para la base 'IGoR' restringida a los valores de N_g=47 y N_g=46

En las Figuras 19, 20 y 21 se muestra la evolución de la gráfica de las distancias de hamming en el entorno de los 50 gaps permitidos. Se representan cómo varían las distancias de hamming desde una base 'IGoR' con un valor N_g=51 a una base 'IGoR' con N_g=46. Se obtiene una gráfica final en la cual aparecen los 2 picos que se observan en la base 'learning'.

Se comparan, a continuación, las distancias de hamming entre la base 'learning' y la base 'IGoR' con $N_g = 40$. El resultado se muestra en la Figura 22.



(a) Distribución de gaps. Base 'learning' (b) Distribución de gaps. Base 'IGoR'

Figura 22: Distancias de hamming para la base 'learning' y para la base 'IGoR' con N_g=40

En la Figura 22 (b), para la base 'IGoR' con un valor de $N_g = 40$, aparecen 2 picos en las distancias de hamming, como ocurre para la Figura 22 (a) donde se ha empleado la base 'learning'.

En definitiva, se observa cómo una vez se ha restringido la base 'IGoR' a un valor de N_g=40, nos hemos quedado con un conjunto de secuencias que se parecen a las secuencias 'learning' en cuanto a la distribución del número de gaps y las distancias de hamming. Esto supone, quedarnos con un conjunto de secuencias más 'completas' con las cuáles se obtienen unos mejores resultados de la

capacidad predictiva.

3.3. Clasificación usando la base 'OAS'

Tras los análisis efectuados con la base 'IGoR', se procede a evaluar la capacidad predictiva del modelo utilizando esta vez una base de datos experimental extraída del repositorio *Observed Antibody Space* 'OAS' [6]. Esta base cuenta con un total de 23251 secuencias de cadenas VH. Se evalúa, en primer lugar, los valores de Ω y λ óptimos, tal y como se hizo en el apartado 3.1. Para estimar el valor de Ω óptimo, en la Figura 23 se representa la norma de la covarianza pesada ($\|\bar{C}\|$) en función de Ω, haciendo uso de un código en Python [8]. Se considera el valor óptimo de Ω aquel que maximiza el valor de $\|\bar{C}\|$, obtenido para Ω=0.508.

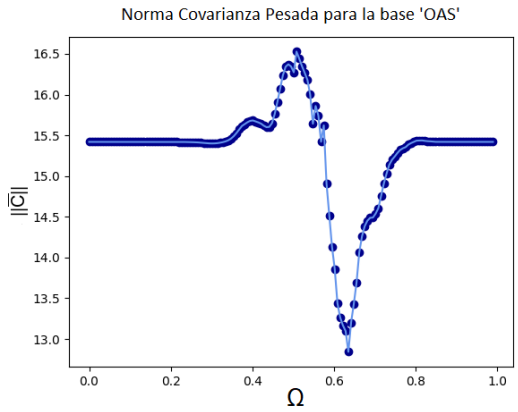


Figura 23: Norma de la covarianza pesada en función de Ω para la base 'OAS'. El valor de Ω óptimo será aquel que maximice la norma de la covarianza pesada

Se evalúa, ahora, el parámetro λ. Se sigue el mismo procedimiento que en las secciones 3.1 y 3.2 anteriores. El resultado se presenta en la Figura 24 donde el valor óptimo viene dado por $\lambda = 0.136$.

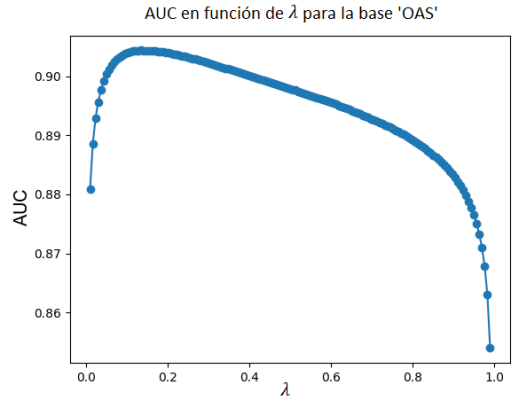


Figura 24: AUC en función de λ. El valor de λ se estima como aquel que es capaz de maximizar el valor de AUC

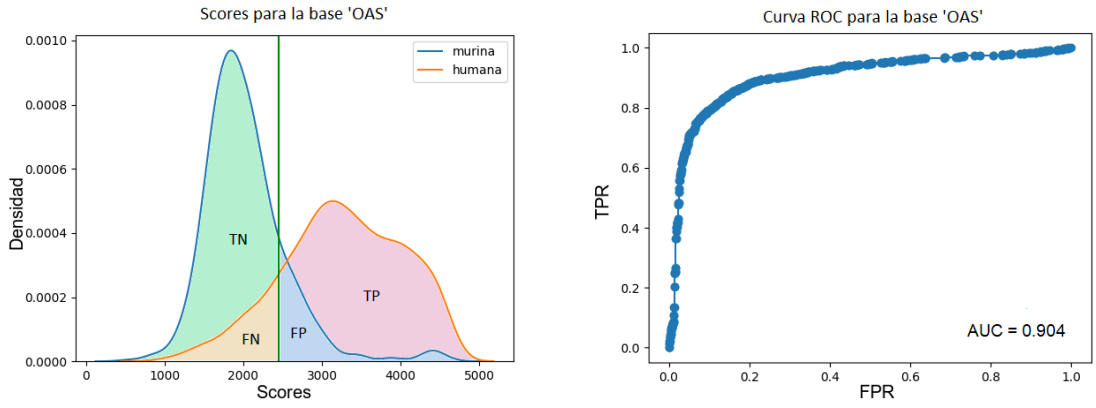
A continuación, se trata de optimizar los parámetros Ω y λ utilizando otro método. Como ya se hizo en la sección 3.1, se realiza un barrido de dichos parámetros, calculando el valor de AUC. De nuevo, se obtiene que AUC se reduce para valores de λ extremos, especialmente para valores de λ cercanos a 1. Además, se obtiene que AUC es ligeramente más alta para valores de Ω entre 0.5 y 0.6 y a su vez para valores de λ entre 0.1 y 0.3. En este rango se encontraban los valores de Ω y λ considerados como óptimos usando el criterio que maximiza la norma de la covarianza pesada. Sin embargo, no se ha podido estimar unos nuevos valores fiables de Ω y λ en los que poder confiar, por tanto, se procede a evaluar la capacidad predictiva del modelo usando los valores anteriores Ω=0.508 y λ=0.136.

Siguiendo el mismo método que en las secciones anteriores, se evalúa la capacidad predictiva. Se establece como valor umbral U = 2450.57 y se obtiene que el área bajo la curva ROC es AUC = 0.904. Esto significa que existe un 90.4 % de probabilidad de que dados dos anticuerpos, uno humano y el otro murino, el modelo los clasifique correctamente.

En la Tabla 5 se reflejan los valores concretos de la capacidad predictiva para la base de datos 'OAS' y en la Figura 25 se muestran las gráficas de *scores* y la curva ROC obtenida.

M_{humana}	M_{murina}	TP	FP	TN	FN	AUC	YI
1388	1379	1187	217	1162	201	0.904	0.698

Tabla 5: Valores ROC para la clasificación usando la base 'OAS'. M_{humana} corresponde al número de secuencias humanas de la base 'test', M_{murina} simboliza el número de secuencias murinas de la base 'test', TP expresa los verdaderos positivos, FP los falsos positivos, TN los verdaderos negativos, FN los falsos negativos, AUC es el área bajo la curva ROC y finalmente YI simboliza el índice de Youden

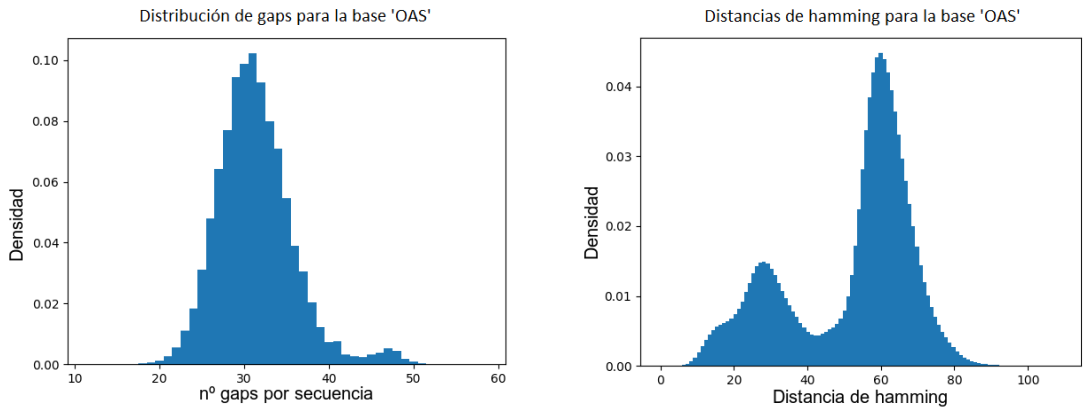


(a) *Scores* (b) Curva ROC

Figura 25: Capacidad predictiva para la base de datos 'OAS'. En el panel izquierdo se muestra la distribución de *scores* para los anticuerpos humanos y murinos. La línea azul representa los anticuerpos murinos, la línea naranja representa los anticuerpos humanos y la línea verde vertical establece el valor umbral. En el panel derecho, se presenta la curva ROC correspondiente

Estudiando la capacidad predictiva obtenida tanto con la base de datos 'learning' como con la base 'OAS' se observa que la capacidad predictiva es ligeramente mejor para la base de datos 'learning' empleada en 3.1, con la que se obtuvo que AUC=0.923 y YI= 0.746.

Con el objetivo de estudiar las diferencias entre los elencos de secuencias de la base 'learning', la base 'IGoR' y la base 'OAS', se muestra en la Figura 26 la distribución de número de gaps y las distancias de hamming para la base 'OAS'.



(a) Distribución de gaps

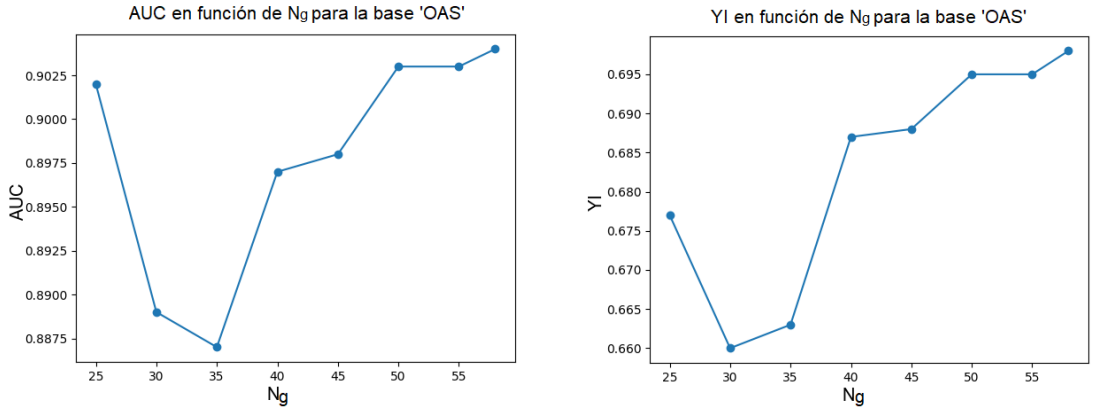
(b) Distancias de hamming

Figura 26: Distribución de gaps (izda.) y distancias de hamming (dcha.) para la base 'OAS'

Si se comparan la Figura 26(a) donde aparece la distribución de gaps para la base 'OAS' con la Figura 17 (a) donde se muestra esa misma distribución para la base 'learning', se observa que las secuencias que contienen en torno a 30 gaps son las que conforman la mayoría en ambos archivos de datos. Se puede decir que ambas bases de datos están conformadas por secuencias similarmente completas en cuanto al número de aminoácidos que contienen. Además, comparando la Figura 26(b) donde aparecen las distancias de hamming de la base 'OAS' con la Figura 18 (a) que muestra estas distancias para la base 'learning' se aprecia cierta similitud entre las distancias de hamming para ambos elencos de secuencias ya que en ambos casos aparecen dos picos.

En definitiva, existe una similitud en cuanto a la distribucción de gaps y las distancias de hamming de la base 'learning' y la base 'OAS' que no se da en la base 'IGoR' inicial conformada por las 40000 secuencias. Se recuerda la Figura 8, donde se mostraba que la mayoría de secuencias que conforman la base 'IGoR' contenían en torno a 50 gaps, así como la Figura 18 (b) donde aparecían 4 picos para las distancias de hamming de la base 'IGoR'. Sin embargo, restringiendo dicha base a valores en torno a N_g=40, la capacidad predictiva mejora y esto supone que las gráficas correspondientes a la distribución de gaps (Figura 17(b)), y las distancias de hamming (Figura 22(b)) se comiencen a asemejar a las obtenidas para la base 'learning' y la base 'OAS'.

Finalmente, se ha analizado el comportamiento de la capacidad predictiva en función de N_g para la base 'OAS'. Se muestran 8 puntos empezando desde N_g=25 de 5 en 5 hasta alcanzar el valor N_g=55 y añadiendo un valor final para N_g=58 que corresponde a la base de datos 'OAS' completa. El resultado de este estudio se muestra en la Figura 27.



(a) AUC en función de nº gaps filtrados

(b) IY en función de nº gaps filtrados

Figura 27: Estudio de la capacidad predictiva para la base 'OAS', en función de N_g

El comportamiento en la Figura 27 es completamente diferente a lo que se mostraba en la Figura 15. Para la base 'OAS' reducir el número de gaps permitidos por secuencia supone la disminución de AUC y del índice de Youden (YI). Se produce, además, un decaimiento en la capacidad predictiva para la zona en torno a 30 gaps permitidos por secuencia como ya vimos que sucedía con la base 'IGoR'.

Tras lo visto durante la sección, se sintetiza en la Tabla 6 los valores extraídos de la capacidad predictiva para cada una de las base de datos. Para el caso de la base 'IGoR' se muestran los valores obtenidos empleando N_g=20, pues con esta restricción se obtenían los mejores resultados de la capacidad predictiva con la base dada.

Base	M	AUC	YI
learning	1309	0.923	0.746
IGoR	1998	0.895	0.676
OAS	23251	0.904	0.698

Tabla 6: Recopilación de la capacidad predictiva obtenida para las bases 'learning', 'IGoR' y 'OAS'

4. Conclusiones

En este trabajo se ha utilizado el modelo desarrollado en [1] y se ha analizado si empleando distintas bases de datos los resultados obtenidos mejoran. Se recuerda que se han empleado las cadenas VH de la base 'learning', la base 'IGoR' y la base 'OAS'. Tras la presentación de los resultados para cada una de estas bases, se extraen las siguientes conclusiones:

• En un primer análisis, se ha verificado que con la base 'learning' que ya se utilizó en [1], el modelo clasificador de anticuerpos humanos y murinos es muy bueno. Se confirma esto al obtener una curva ROC con un valor AUC = 0.923. La selección de anticuerpos que se hizo para conformar esta base de datos fue, por tanto, muy buena.

• Posteriormente, se ha utilizado la base de datos generada con 'IGoR'. Un análisis exhaustivo de esta base de datos nos desveló que la mayoría de las secuencias generadas mediante 'IGoR' contenían un elevado porcentaje de gaps. Se infirió que el programa 'IGoR' crea secuencias incompletas que la herramienta ANARCI debe completar con un número de gaps muy elevado. Cabe destacar que en el conjunto de 40000 secuencias aparecen 5 secuencias conformadas por todo gaps, estas secuencias son capaces de alterar completamente el comportamiento de ciertos parámetros como la norma de la covarianza pesada $\|\bar{C}\|$.

Además, se ha confirmado que un número alto de gaps en la mayoría de secuencias de la base 'IGoR' empeora la capacidad predictiva del modelo. Esto se debe a que las secuencias 'IGoR' conformadas por más gaps contienen menos información y por tanto son secuencias de peor calidad.

Es por ello necesario aplicar un filtrado para reducir el número de gaps permitidos por secuencia (N_g) y poder trabajar con un elenco de secuencias 'IGoR' 'fiable'. Es necesario aplicar un filtrado bastante restrictivo con el valor de N_g para obtener buenos resultados de la capacidad predictiva. Sin embargo, el resultado hallado, AUC ≈ 0.895 para N_g=20, no logra mejorar el resultado obtenido para la base 'learning'.

• Finalmente, se emplea la base que se ha denominado 'OAS'. Con esta base de datos las predicciones obtenidas son buenas. Obtenemos una curva ROC con un valor de AUC = 0.904. Sin embargo, las predicciones son incapaces de mejorar los resultados obtenidos para la base de datos 'learning'. La base 'OAS' cuenta con $\approx$ 22000 secuencias, mientras la base 'learning' cuenta con $\approx$ 1000 secuencias, lo que evidencia que una base datos con más secuencias no conlleva a una mejora de las predicciones del modelo. Un filtrado preciso de las secuencias empleadas es más efectivo para obtener mejores resultados.

Cabe destacar dos resultados desconcertantes a desarrollar en el futuro que se han encontrado en el proceso de análisis:

- La capacidad predictiva empeora si reduzco el valor de N_g en la base de datos experimental 'OAS' al contrario de lo que ocurre para la base 'IGoR'.

- Tanto en la base de datos 'OAS' como en la base de datos 'IGoR' con la que hemos trabajado existe un rango de valores de N_g en torno a 30 para las cuales la capacidad predictiva parece empeorar ligeramente. Aunque es cierto que se trata de una pequeña fluctuación en los valores de AUC e YI, la razón o motivo de esta evidencia nos resulta de momento desconocido.

Referencias

[1] Clavero-Álvarez, A., Di Mambo, T., Pérez-Gaviro, S., Magnani, M., Bruscolini, P.*Humanization of Antibodies using a Statistical Interference Approach.* Sci Rep 8, 14820 (2018). https://doi.org/10.1038/s41598-018-32986-y

[2] Badassi, C., Zamparo, M., Feinauer, C., et al. *Fast and Accurate Multivariate Gaussian Modeling of Protein Families: Predicting Residue Contacts and Protein-Interaction Partners.* PLoS ONE 9(3): e92721 (2014). https://doi.org/10.1371/journal.pone.0092721

[3] Rendón-Anaya, M., Alagón, A. *Mecanismos moleculares de diversificación de inmuglobulinas.* REB 27(1): 19-29 (2008)

[4] Dunbar, J., Deane, C.M. *ANARCI: Antigen receptor numbering and receptor classification.* Bioinforma, 32, 298-300 (2016). https://doi.org/10.1093/bioinformatics/btv552

[5] *IGoR: Inference and Generation of Repertoires.* https://github.com/qmarcou/IGoR

[6] *OAS: Observed Antibody Space.* https://opig.stats.ox.ac.uk/webapps/oas/

[7] Luna Cerralbo, D. *Validación y refinamiento de un modelo estadístico para la clasificaición y humanización de anticuerpos.* TFG Universidad de Zaragoza (2020)

[8] Luna Cerralbo, D. *Códigos relacionados con el Modelo Gaussiano Multivariante.* https://github.com/DLunaUNIZAR/MG_Model

[9] Luna Cerralbo, D. *Librería de python desarrollada para trabajar en el área de la biotecnología.* https://github.com/DLunaUNIZAR/BioPyCustom

5. Anexos

5.1. Curvas ROC

Hemos empleado a lo largo del trabajo las denominadas curvas ROC y el área AUC bajo este tipo de curvas para valorar la capacidad predictiva del modelo según el tipo de base de datos empleada. Se va a explicar con detalle en qué consisten. Las curvas ROC son una representación gráfica, que muestran si un modelo es óptimo clasificando elementos en dos posibles categorías. En nuestro caso, dado un anticuerpo, catalogarlo como humano o como murino.

Se muestra un ejemplo gráfico:

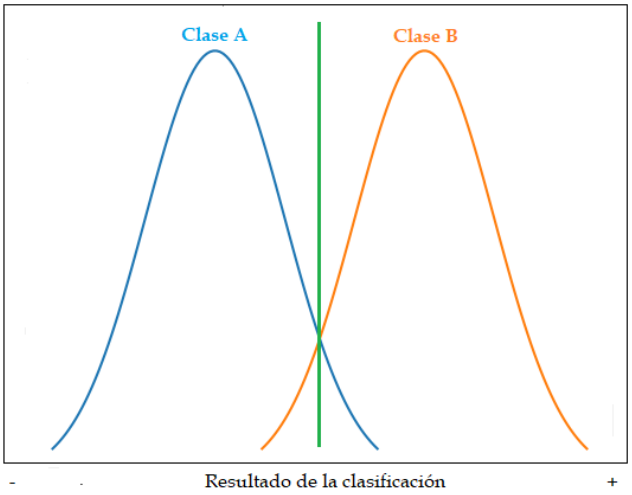


Figura 28: Ejemplo de clasificación de elementos

La línea en azul representa a los elementos que pertenecen a la clase A mientras que la línea naranja representa a los elementos que pertenencen a la clase B. Ahora debemos elegir un valor en donde establecemos el corte o un valor umbral, en el dibujo la línea verde, por encima de la cual predecimos a todos los elementos como clase B y por debajo de la cual predecimos a todos los elementos como clase A.

Aquellos elementos que pertenecen a la clase B y están por encima del umbral serán 'verdaderos positivos' (TP) y los elementos de clase A que están por encima del umbral será 'falsos positivos' (FP), ya que se han clasificado de manera errónea.

A su vez los elementos de clase A por debajo del umbral serán 'verdaderos negativos' (TN) y, los elementos de clase B por debajo del umbral serán 'falsos negativos' (FN), y se han predicho incorrectamente.

Diferentes valores del umbral, supondrán una diferente predicción clasificatoria. Esto es, a cada valor del umbral, le corresponde una proporción diferente de falsos positivos y verdaderos negativos y por tanto, cada valor umbral determina un punto distinto en el plano FPR-TPR, siendo FPR la tasa de falsos positivos:

$$FPR = \frac{FP}{N} \tag{14}$$

donde N es el número total de elementos clase A. Y TPR es la tasa de verdaderos positivos:

$$TPR = \frac{TP}{P} \tag{15}$$

siendo P el número total de elementos clase B.

En consecuencia, en el plano FPR-TPR se dibuja la denominada curva ROC, como puede verse en la siguiente Figura 29.

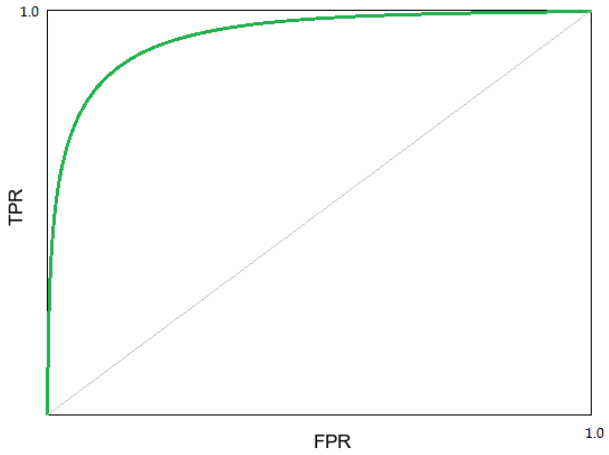


Figura 29: Ejemplo de una curva ROC

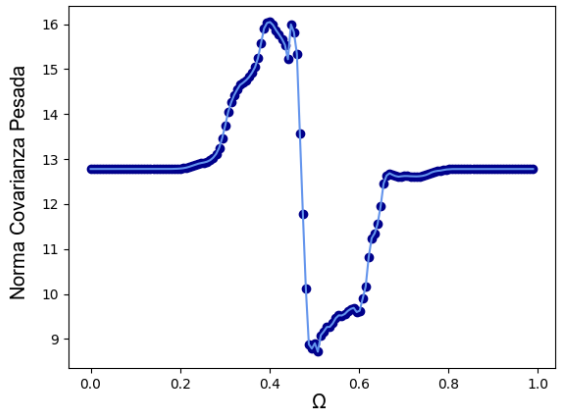
El área bajo estas curvas ROC se denomina *AUC* (*area under the curve*). Cuánto mayor es el valor de esta área, mejor es el modelo clasificatorio. Finalmente, el valor del umbral óptimo se puede determinar como el punto de la curva ROC que maximiza el índice de Youden, que viene dado por la fórmula:

$$Y = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1 \tag{16}$$

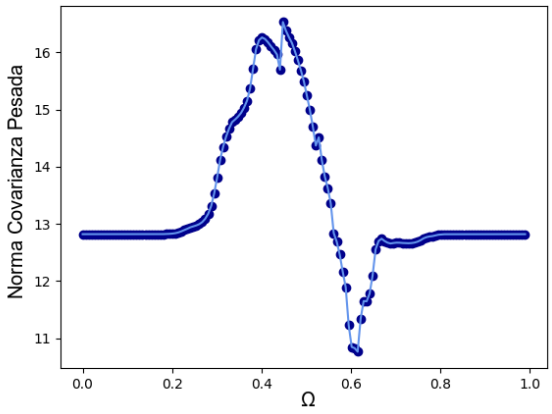
Un índice de Youden mayor supone un mejor modelo clasificatorio, como ocurre con el área bajo la curva ROC (AUC).

5.2. Norma de la covarianza pesada para la base 'IGoR'

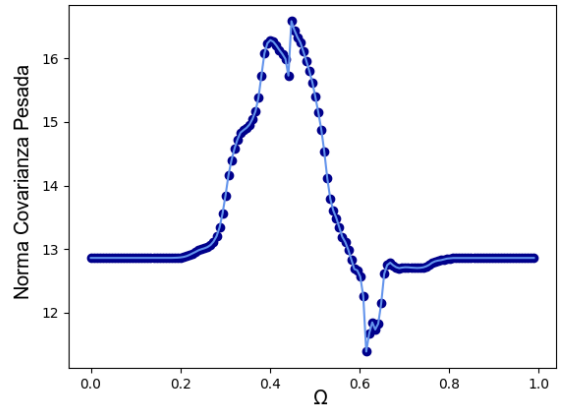
Se adjuntan en este apartado la evolución de la gráfica de la norma de la covarianza pesada en función de Ω para la base 'IGoR' en el rango entre los valores N_g=83 y N_g=40. En concreto, en la Figura 30, se muestran las gráficas para los casos de $N_g = 75$, $N_g = 65$, $N_g = 55$, y $N_g = 45$.



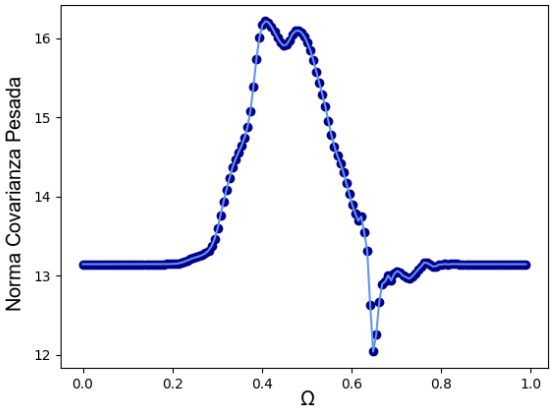
(a) Norma de la covarianza pesada para N_g=75



(b) Norma de la covarianza pesada para N_g=65



(c) Norma de la covarianza pesada para N_g=55



(d) Norma de la covarianza pesada para N_g=45

Figura 30: Evolución de la gráfica de la norma de la covarianza pesada en función de Ω para una serie de valores entre N_g=83 y N_g=40

5.3. Código para obtener las gráficas de la distribución de gaps por secuencia

Para este código y los posteriores nos apoyaremos en la libería BioPYCustom [9], muy útil para trabajar computacionalmente en problemas aplicados a la biología.

```
1
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import sys
5 import os
6 from typing import Tuple
7
8 from BioPyCustom.read_write_files import read_fasta_file, read_seq_file
```

```
from BioPyCustom.clases import Data_base, extract_Data_base
from BioPyCustom.read_write_files import write_fasta_file
from BioPyCustom.input_data import transform_list_type, transform_variable_type


###FUNCIONES####

def check_input(input_data):
  db_path=input_data[1]
  #Comprobamos que el fichero de secuencias efectivamente existe:
  if not os.path.exists(db_path):
    print("Archivo de secuencias no encontrado.")
    exit()
  return db_path

###C DIGO###
db_path = check_input(input_data = sys.argv)
db = read_fasta_file(fasta_file_path = db_path)

i=0
gaps=np.zeros(db.n_seqs)

#Recorremos cada secuencia contando el n mero de gaps que guardamos en un vector
    llamado gaps
for seq_class in db.seqs_list:
    gaps[i]=seq_class.seq.count("-")
    i=i+1

#Graficamos en un histograma
max=int(np.amax(gaps))
min=int(np.amin(gaps))
plt.hist(gaps, bins=(max-min)+1, range = (min-0.5, max+0.5), density=True)
plt.xlabel("n   gaps por secuencia")
plt.ylabel("Densidad")
plt.show()
```

5.4. Función para obtener valores ROC para la clasificación

Se modifica uno de los códigos proporcionados en [8] donde se halla el valor de AUC en función de lambda y se añade una función en el programa para el cálculo del parámetro del índice de Youden (YI), así como de otros parámetros de relevancia como los verdaderos positivos (TP), los falsos positivos (FP), los verdaderos negativos (TN) y los falsos negativos (FN):

```
def get_Y_par metros(tpr:np.ndarray,fpr:np.ndarray):
  Y=(tpr+(1-fpr))-1
  index=np.argmax(Y)
  tnr=1-fpr
  fnr=1-tpr
  Y=Y[index]
  tp=tpr[index]*human_test_db.n_seqs
  fp=fpr[index]*murine_test_db.n_seqs
```

```
  tn=tnr[index]*murine_test_db.n_seqs
  fn=fnr[index]*human_test_db.n_seqs
  return Y, tp, fp, tn, fn
```

5.5. Código para obtener las gráficas del número de secuencias de una base de datos en función de N_g

```
import pandas as pd
import numpy as np
import sys
import os
from sklearn import metrics
import matplotlib.pyplot as plt
from typing import Tuple
from BioPyCustom.read_write_files import read_fasta_file, read_seq_file,
    write_fasta_file
from BioPyCustom.clases import Data_base, random_sample
from BioPyCustom.MG_model.General import from_db_to_cov_mean, uniform_prior,
    posterior, sigma_inv_logdet, Prob_in_model, load_ref_data
from BioPyCustom.clases import add_two_dbs, extract_Data_base
from BioPyCustom.change_format import amin_db_str_to_bin
from BioPyCustom.input_data import cast_input_str_to_list, transform_list_type



def check_input(input_data:list)-> Tuple [str, int, int]:
  """
  Checkea si los valores de input del programa son validos.

  Parameters
  ----------
  input_data: list
    Contiene filtro_inf y filtro_sup

  Returns
  -------
  filtro_inf: int
    Valor inferior de gaps filtrados
  filtro_sup: int
    Valor superior de gaps filtrados

  """
  if len(input_data) != 4:
    print("\nInput data no valido.\n")
    print("Usage: python barrido_gaps.py seq_path filtro_inf filtro_sup\n")
    print("Base de datos con la que vamos a trabajar")
    print("filtro_inf")
    print("filtro_sup")
    exit()

```

```
  numerical_input = cast_input_str_to_list(input_data = input_data)
  db_path, filtro_inf, filtro_sup = transform_list_type(values_list = input_data,
    type_list = [str,int,int])

  return db_path, filtro_inf, filtro_sup

db_path, filtro_inf, filtro_sup = check_input(sys.argv)
db = read_fasta_file(fasta_file_path = db_path)


for gaps in range(filtro_inf, filtro_sup):
  seq=0
  for seq_class in db.seqs_list:
    if (seq_class.seq.count("-")<=gaps):
      seq=seq+1


  with open ("n_seq_Ng.txt", "a") as file:
      file.write('%s'%gaps)
      file.write("\t")
      file.write('%s'%seq)
      file.write("\n")

#Graficamos los valores

x,y= np.loadtxt('n_seq_Ng.txt',usecols=[0,1], unpack=True)

plt.plot(x,y)
plt.scatter(x,y)
plt.xlabel('Ng')
plt.ylabel('n   de secuencias')
plt.show()
```

5.6. Código para obtener las gráficas de las distancias de hamming

```

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sys
import os
from BioPyCustom.read_write_files import read_fasta_file, read_seq_file
from BioPyCustom.clases import Data_base, extract_Data_base
from BioPyCustom.read_write_files import write_fasta_file
from BioPyCustom.change_format import amin_db_to_str_matrix, amin_str_mat_to_uint8
from BioPyCustom.bio_statistics import hamming_dist_from_num_mat


#leer archivo fasta con todas las secuencias
db_path="C:/Users/Usuario/OneDrive/Escritorio/tfg/Codigos/MG_Model-master/MG_Model-
    master/Databases/VH/exthuman_jointVH_AHo_final.fasta"
```

```
db= read_fasta_file(fasta_file_path = db_path)

#Pasar las secuencias a formato ndarray
matriz=amin_str_mat_to_uint8(amin_db_to_str_matrix(data_base=db))
distancia=hamming_dist_from_num_mat(num_mat=matriz)

l_vect = int((db.n_seqs-1)*db.n_seqs/2)
array=np.zeros(l_vect)

k=0

#Guardar las distancias en un array
for i in range(db.n_seqs-1):
  for j in range(i+1,db.n_seqs):
    array[k]=distancia[i][j]
    k=k+1

#Representaci n distancias de hamming
max=int(np.amax(array))
min=int(np.amin(array))
plt.hist(array, bins=(max-min)+1, range = (min-0.5, max+0.5), density = True)
plt.xlabel("Distancia de hamming")
plt.ylabel("Densidad")
plt.show()
```