



23 DE FEBRERO DE 2021

CUADERNO PERSONAL DE ACTIVIDADES:
ANÁLISIS BIOINFORMÁTICO Y REVISIÓN
BIBLIOGRÁFICA DEL DOMINIO DE UNIÓN
AL RECEPTOR DEL SARS-COV-2 EN
COMPLEJO CON EL CR3022 FAB (6YLA)

GRADO BIOTECNOLOGÍA

JAVIER21 VILLEGAS SALMERÓN
UNIVERSIDAD DE GRANADA
Javier821@correo.ugr.es



Abstract

La pandemia provocada por el SARS-CoV-2 ha hecho temblar los cimientos del mundo que conocemos y ha puesto en jaque a millones de personas. Por esta razón, en la asignatura de Ingeniería de proteínas del grado en Biotecnología de la UGR se nos ha asignado una proteína o varias, todas pertenecientes al virus provocador de la enfermedad conocida como COVID-19.

Con este trabajo quiero poner mi granito de arena en la lucha contra la enfermedad al realizar una revisión bibliográfica y a su vez un análisis bioinformático de los ficheros entregados por el profesor. A mí me fueron entregado varios de ellos, todos ellos correspondientes a cristalizaciones del RBD en unión a distintos anticuerpos, y yo he elegido para mi estudio el correspondiente al que interactúa con el anticuerpo monoclonal, o mAB, CR3022, con nombre de fichero 6yla.

He realizado esta elección porque me parecía la más interesante de las que he visto y porque un trabajo de todos hubiese sido demasiado extenso, por lo que he preferido enfocarme en una.

Este trabajo, junto con los códigos y ejecutables de los programas empleados, serán subidos a GitHub, una plataforma que he aprendido a utilizar a lo largo del curso en paralelo con la asignatura y que resulta muy útil debido a que permite a varias personas trabajar con un mismo código a la vez, lo que me puede resultar muy útil en el futuro, y para que esté a disposición de todo aquel que quiera consultar la información que apporto.

Índice

- 1. Introducción**
- 2. Bases generales SARS-CoV-2**
 - 2.1. Proteína S**
 - 2.2. ECA2**
 - 2.3. Replicación Viral**
- 3. Respuesta inmunitaria y desarrollo de estrategias contra el SARS-Cov-2**
 - 3.1. Respuesta del sistema inmune**
 - 3.2. Mecanismos de evasión de la respuesta inmune**
 - 3.3. Táctica contra el SARS-CoV-2: mAB CR3022**
- 4. Actividades**
 - 4.1. Bases de datos secuenciales y estructurales**
 - 4.1.1. Detección de formatos de ficheros proteicos y extracción de la secuencia primaria**
 - 4.1.2. Desarrollo de la función CargarPDB**
 - 4.2. Visualización de proteínas**
 - 4.2.1. Ficheros de estructuras**
 - 4.2.2. Desarrollo de la función EscribirPDB**
 - 4.3. Predicción estructural y rediseño de proteínas**
 - 4.3.1. Diagrama de Ramachandran**
 - 4.3.2. Coordenadas transformadas y construcción de un estereodiagrama**
 - 4.3.3. Alinear_Z**
 - 4.3.4. Cálculo de RMSD**
 - 4.3.5. Mutagénesis dirigida: creación de una mutación Phe-Tyr**
 - 4.3.6. Creación de un programa de predicción de enlaces disulfuro**
 - 4.3.7. Perfil hidrofóbico y anfipatía axial**

1. Introducción

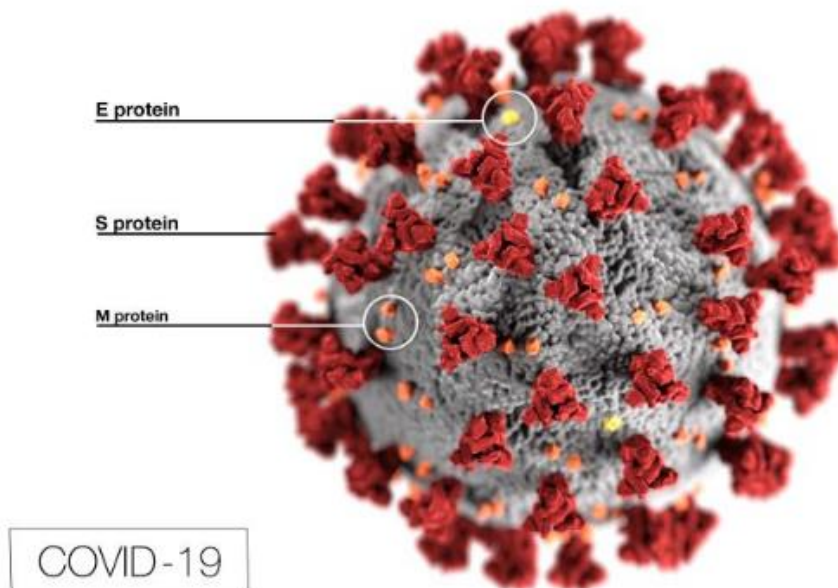
Los coronavirus son virus envueltos cuyo genoma consiste en una única molécula de RNA monocatenario positivo o +ssRNA. Pertenecen a una gran familia de virus (Coronaviridae) que infectan aves y varios mamíferos, y pertenecen a los Nidovirales, denominados así debido al aspecto similar a una corona solar conferido por sus glicoproteínas de superficie. La familia Coronaviridae se clasifica en cuatro géneros: alfacoronavirus, betacoronavirus, deltacoronavirus y gammacoronavirus. Estos virus causan enfermedades respiratorias, gastrointestinales, neurológicas y hepáticas en varias especies, entre ellas humanos, y son capaces de mutar con rapidez, recombinarse y transmitirse de una especie a otra (zoonosis).

Los coronavirus fueron reconocidos como causantes de serias infecciones respiratorias e intestinales después del brote del “síndrome respiratorio agudo severo” (SARS), cuyo agente etiológico emergió en China en 2002.

El coronavirus de tipo 2 causante del síndrome respiratorio agudo severo, causante de la enfermedad por coronavirus conocida como COVID-19, o SARS-CoV-2 es un tipo de coronavirus cuya expansión mundial provocó la pandemia de COVID-19, en la que todavía nos encontramos.

A mí me ha tocado estudiar la estructura cristalina del dominio de unión al receptor del SARS-CoV-2 en complejo con distintos anticuerpos, y yo he elegido la que forma con el anticuerpo CR3022 Fab (6YLA) a la hora de realizar las actividades.

En este trabajo realizaré una pequeña revisión de este virus que tanto ha dado que hablar en el último año, con el fin de esclarecer un poco los aspectos genéticos, moleculares y la naturaleza del SARS-2, tanto desde un punto de vista general como centrándome en los aspectos más característicos del mismo.



Estructura general del coronavirus SARS-CoV2. Imagen: Alissa Eckert, MS; Dan Higgins, MAMS, Centers for Disease Control and Prevention.

2. Bases generales SARS-COV-2

El SARS-CoV-2, nombrado así debido a que su secuencia genética es similar a la del SARS-CoV, es un betacoronavirus de 60 a 140 nm de diámetro, que posee una envoltura y una nucleocápside helicoidal formada por +ssRNA con cerca de 30,000 pares de bases, siendo el RNA más largo descrito en un virus.

La estructura del virión consiste principalmente en una nucleocápside, que protege al material genético viral, y en una envoltura externa. En la nucleocápside, el genoma viral está asociado con la proteína de la nucleocápside (N), la cual, se halla fosforilada e insertada dentro de la bicapa de fosfolípidos de la envoltura externa. En cuanto a la envoltura externa, allí se encuentran proteínas estructurales principales, denominadas proteína Spike (S), proteína de membrana (M) y proteína de envoltura, además de otras proteínas accesorias, como la proteína hemaglutinina esterasa.

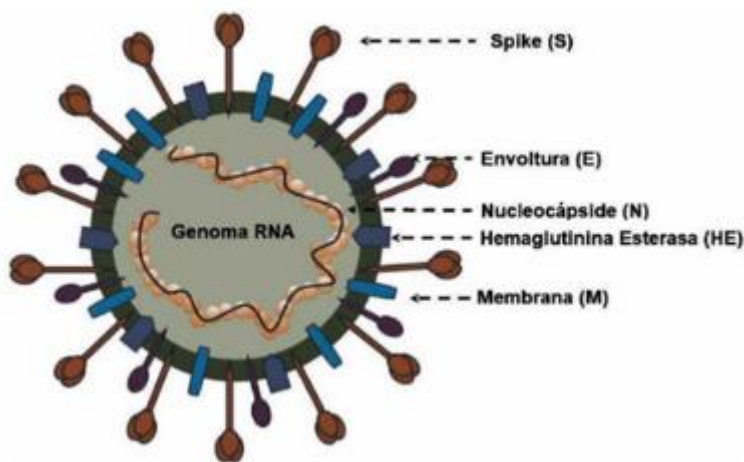


Figura 2. Podemos observar un esquema de la estructura del SARS-CoV-2 con sus principales componentes estructurales. ((Pastrian-Soto, Gabriel. (2020).

Dos tercios de su material genético (ORF1a Y ORF1b), más cerca del extremo 5', codifican para el gen de la replicasa viral. Este gen será traducido al inicio de la infección en dos poliproteínas de gran tamaño, pp1a y pp1ab. Estas posteriormente serán procesadas para generar 16 proteínas no estructurales (nsps), cuya mayoría resultan necesarias para el proceso de replicación y otras con funciones aún desconocidas; y del tercio restante se sintetiza RNA subgenómico que codifica proteínas estructurales de envoltura (E), membrana (M), nucleocápside (N) y espícula (S); junto con proteínas accesorias entre cuyas funciones destaca la evasión de la respuesta inmune del huésped.

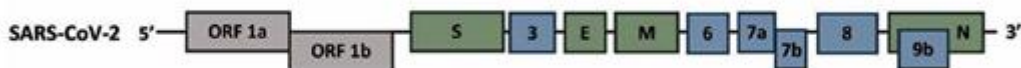


Fig 3. Organización genética de SARS-CoV-2. Esquema del genoma RNA monocatenario de polaridad positiva (+ssRNA) de SARS-CoV-2. Hacia el extremo 5'. Se codifica el gen de la replicasa viral por medio de ORF1a y ORF1b para la producción de pp1a y pp1ab. Hacia el extremo 3', se codifican los genes de las 4 proteínas estructurales principales, en verde, y las de las proteínas accesorias. (Pastrian-Soto, Gabriel. (2020).

A continuación, voy a explicar más en profundidad dos estructuras fundamentales en la infección de SARS-CoV-2:

2.1 Proteína S

Posiblemente la proteína más famosa del SARS-CoV-2, la proteína de la espícula, spike o S es una proteína de fusión tipo I, altamente glucosilada, con tres heterodímeros S₁/S₂. En S₁ se encuentra el dominio de unión al receptor (RBD, receptor binding domain) que se va a unir al receptor celular ACE2 y es la parte más variable del genoma viral; mientras que S₂ es una subunidad altamente conservada, que media la fusión con la membrana celular y hace posible su ingreso a las células. Debido a que realiza una función indispensable, la proteína S es el blanco principal de los anticuerpos neutralizantes utilizados en las vacunas.

El dominio RBD (aminoácidos 1030-1651 de la proteína S) tiene dos conformaciones principales, la acostada y la erguida. Mediante estudios de criomicroscopía electrónica se ha conseguido demostrar que en el SARS-CoV-2 está más tiempo en la forma acostada que en la erguida. Este detalle puede ser muy importante ya que permitiría al virus esconderse de los Ac que ataquen a este dominio, permitiéndole evadir de forma más eficaz al sistema inmune.

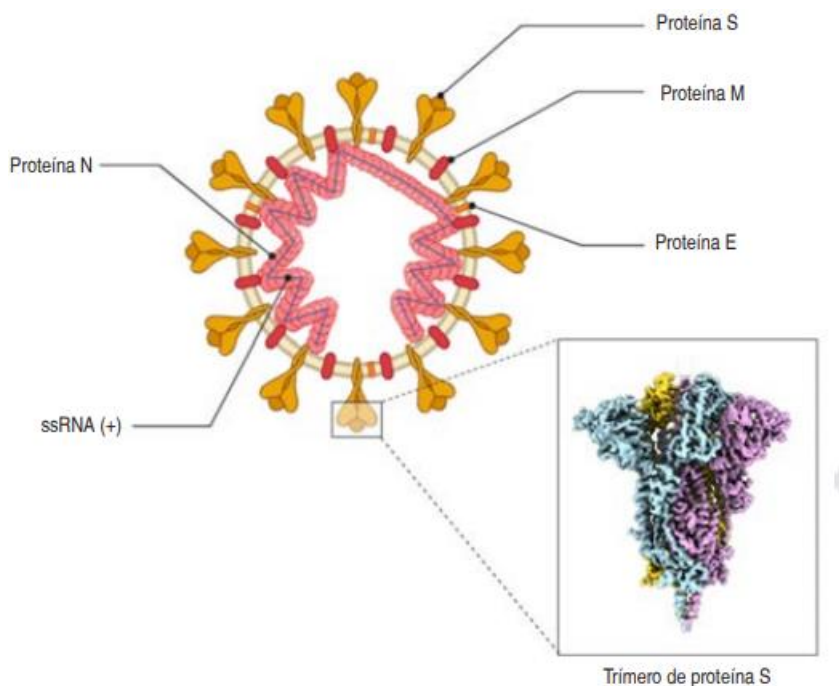


Fig 4. Otra representación de la estructura del SARS-CoV-2 donde se parecía mejor la estructura del trimero de proteína S.



Fig 5. Podemos observar el estado de pre-fusión de la proteína S con todos los RBDs en la conformación hacia abajo. La membrana viral estaría en la parte inferior de la imagen. Todo S1 y S2 se muestra en amarillo, excepto el RBD, que se muestra en gris, con el epítipo CR3022 coloreado en verde. (Cell Host Microbe. 2020 Sep 9; 28(3): 445–454.e6.)

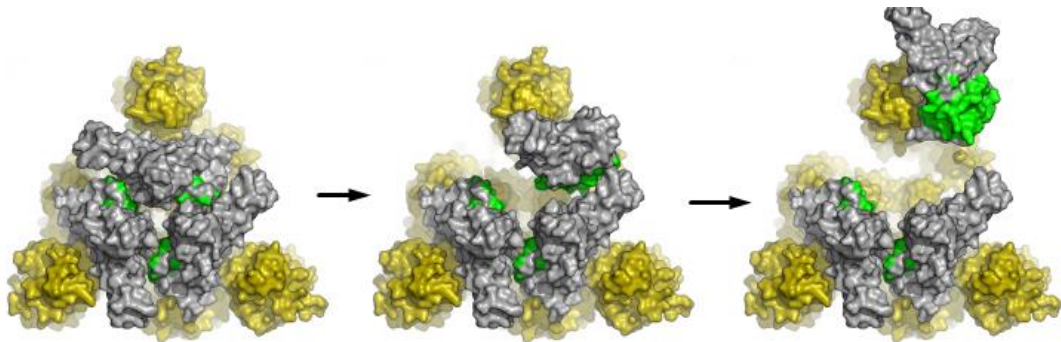


Fig 6. El trimero visto desde arriba. Todos los RBDs abajo en la primera estructura, un RBD arriba en la segunda, y un RBD girado en la última para permitir el acceso al CR3022 (Cell Host Microbe. 2020 Sep 9; 28(3): 445–454.e6.)

2.2 Enzima convertidora de angiotensina 2

La enzima convertidora de angiotensina 2, ECA2, y su proteína homóloga ECA, participan en la regulación de la presión arterial. La ECA2 es una proteína de membrana que se encuentra presente de manera abundante en las células alveolares del sistema respiratorio, especialmente en los neumocitos tipo II, y en humanos también se encuentra en otros tejidos como el corazón o los riñones. El SARS-CoV-2 utiliza esta enzima como receptor para poder infectar a las células que la expresan en su superficie.

ECA2 actúa como un receptor funcional tanto para el SARS-CoV-1 como para el SARS-CoV-2, uniéndose a este último con una afinidad de 10 a 20 veces mayor (KD de ~15 nM), lo que posiblemente contribuye a su facilidad de transmisión. Hay un 73% de identidad de secuencia entre los RBD del SARS-CoV-1 y el SARS-CoV-2. Cuando ACE2 se bloquea, mantiene el RBD "arriba", desestabilizando el "manto" de S1 y posiblemente favoreciendo la conversión a una forma post-fusión en la que la subunidad S2, a través de cambios conformacionales masivos, impulsa su dominio de fusión hacia arriba para enganchar con la membrana del huésped, desechando S1 en el proceso (Los estudios estructurales de la RBD en complejo con ACE2 (Lan et al., 2020; Wang et al., 2020; Yan et al., 2020) muestran que es reconocida por el dominio peptidasa extracelular (PD) de ACE2 a través de interacciones principalmente polares. Esto nos refuerza la teoría de que la proteína S es un candidato atractivo tanto para el desarrollo de vacunas como para la inmunoterapia.

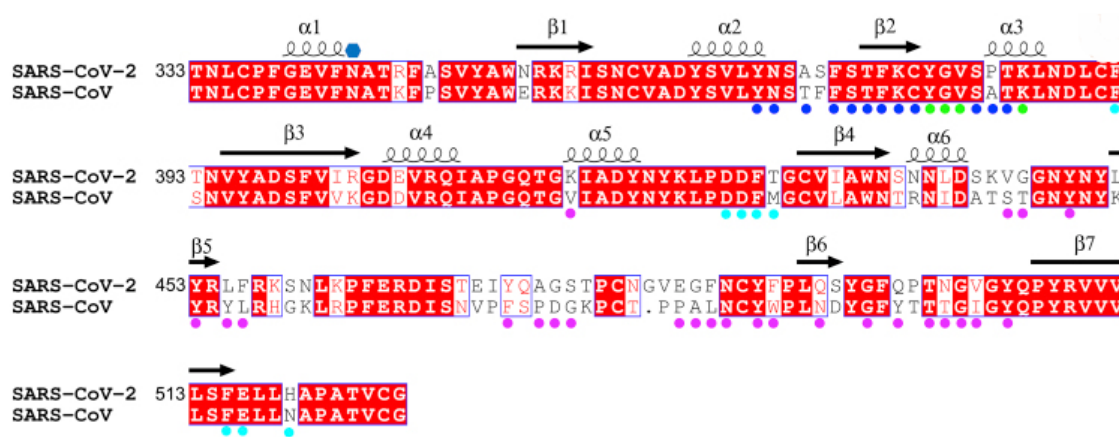


Fig 7. En esta imagen podemos observar una alineación de secuencias entre las RBD de SARS-CoV(-1) y SARS-CoV-2. Los números de los residuos son los del RBD del SARS-CoV-2. Los aminoácidos conservados tienen un fondo rojo, las estructuras secundarias están etiquetadas en la parte superior de la secuencia y el sitio de glicosilación está marcado con un hexágono azul. Los residuos implicados en la unión al receptor están marcados con puntos magenta. Los puntos azules marcan los residuos implicados en las interacciones con la cadena pesada del CR3022 (V_H), los puntos cian marcan los residuos que interactúan con la cadena ligera del CR3022 (V_L), y los puntos verdes marcan los que interactúan con ambas cadenas. Esto sirve como una forma gráfica y fácil de apreciar las similitudes entre ambas secuencias, lo que nos explica que cerca del 73% de sean idéntico entre ellas.

A mí me ha tocado estudiar la estructura cristalina del RBD del SARS-CoV-2 en complejo con el Fab CR3022, el cual solamente está expuesto cuando el RBD se une con ECA2.

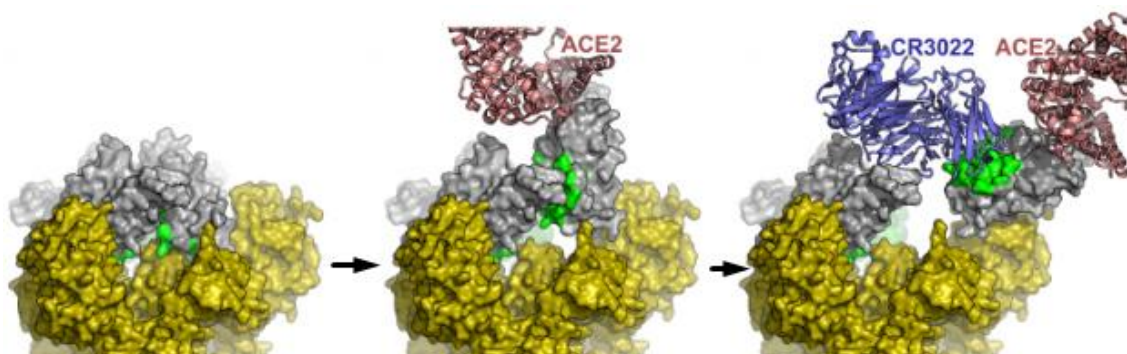


Fig 8. En la siguiente sucesión de imágenes se muestra como la región de unión CR3022, en color verde en la proteína, está inaccesible en la forma pre-fusión de la proteína S, y como una vez se da la unión con ACE2 esta región sufre una modificación que permite que el epítipo conformacional pueda ser detectado por CR3022. (Cell Host Microbe. 2020 Sep 9; 28(3): 445–454.e6.)

A pesar de esto, se ha reportado que el SARS-CoV puede ingresar a las células que no expresan o expresan con baja frecuencia esta enzima, como los macrófagos alveolares, por el contrario, no hay evidencia de infección de células endoteliales que sí expresan ECA2, por lo que podemos deducir que probablemente posea otros mecanismos de infección no descritos, como el uso de correceptores. Uno de estos mecanismos parece estar relacionado con el tipo de proteasa que escinde la proteína S y permite que el virus se internalice.

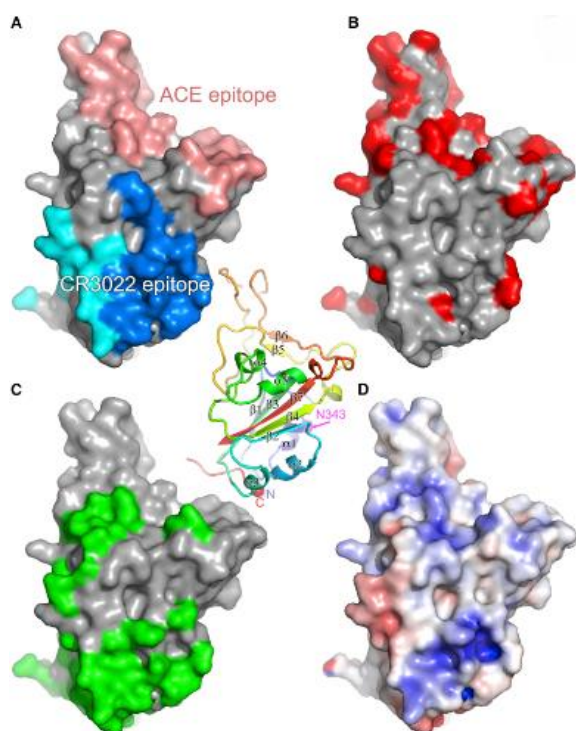


Fig 9. Propiedades superficiales del RBD del SARS-CoV-2: El panel central es una representación cartográfica coloreada en azul para la terminación N y en rojo para la terminación C; la vista es la misma que para (A)-(D). La estructura secundaria está etiquetada junto con el residuo glicosilado N343 (en magenta) y la posición de los extremos del dominio (N y C). En la imagen A vemos una representación de la superficie del RBD, con la zona accesible al disolvente enterrada por la unión del receptor ACE2 coloreada en salmón y la enterrada por el CR3022 (cadena pesada en azul y cadena ligera en cian). En la imagen B, vemos diferencias de secuencia mostradas en rojo entre los RBD del SARS-CoV y del SARS-CoV-2, mapeadas en la superficie del RBD del SARS-CoV-2. En la C, vemos la superficie enterrada en la conformación de pre-fusión de la espiga mostrada en verde. Por último, en la D vemos la superficie electrostática de SARS-CoV-2 RBD contorneada a ± 5 T/e (rojo, negativo; azul, positivo).

2.3 Replicación Viral

Durante la infección, el coronavirus entra en los neumocitos tipo II y otras células mediante la unión de la proteína S con el receptor ECA2. La proteína S es escindida por una proteasa (TMPRSS2 o CatB/L) en S₁ y S₂ y S₂' que separa las dos subunidades y produce un cambio de conformación para activar su estado de fusión. Una vez dentro de las células, el genoma viral se libera al citoplasma y los ribosomas del huésped inician la traducción del gen de la replicasa, que consiste en dos open reading frame o ORF que codifican para dos poliproteínas. Las poliproteínas pp1a y pp1ab son escindidas por dos proteasas virales, 3CLpro y PLpro en 16 proteínas no estructurales, nsp1-nsp16, que conforman el complejo viral de replicación-transcripción (RTC).

Los nsps forman vesículas de doble membrana a partir del retículo endoplasmático rugoso, RER, en donde tiene lugar la replicación viral. Dentro de estas vesículas, una RNA polimerasa dependiente de RNA genera moldes de RNA de cadena negativa que utiliza para replicar genomas y transcribe a nueve RNA subgenómicos, a partir de los cuales se sintetizan las proteínas estructurales. Una vez sintetizadas, las proteínas E, M y S se asocian al retículo endoplasmático y N forma la nucleocápside al unirse al RNA viral. Los viriones se ensamblan en el compartimiento intermedio retículo endoplasmático-Golgi (ERGIC) de donde salen en vesículas que migran hacia la membrana y son liberados de la célula infectada mediante exocitosis.

A diferencia de otros virus RNA, los coronavirus tienen una exonucleasa (EXON), encargada de mantener la fidelidad de la transcripción de su genoma y evitar la acumulación de mutaciones. Este control de calidad de la transcripción le hace resistente a la ribavirina y posiblemente a otros antivirales.

Fernández-Camargo DA y col. *Biología del SARS-CoV-2*

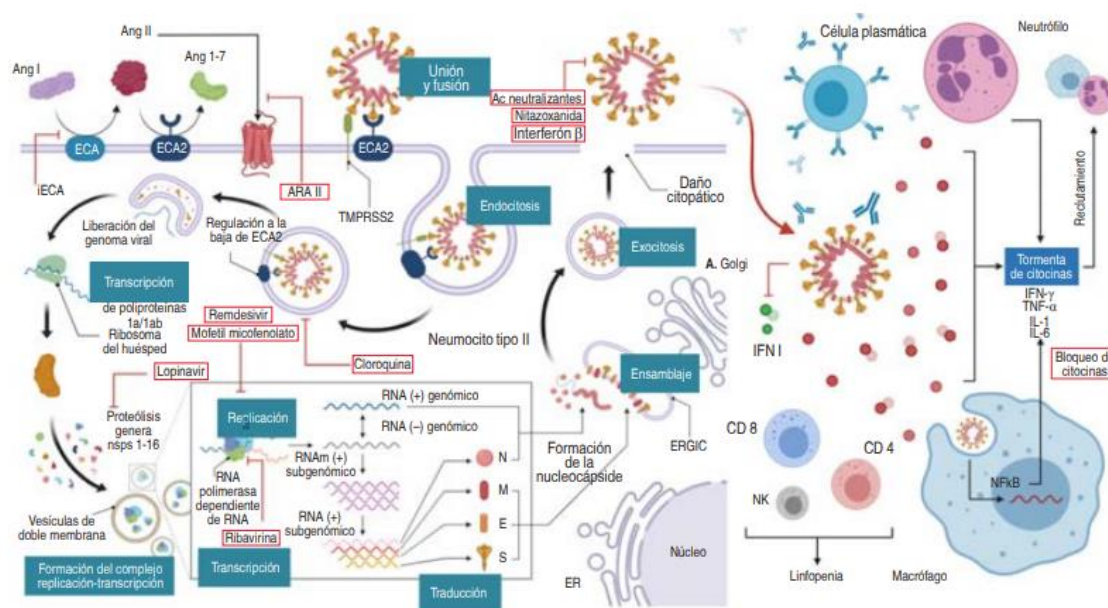


Fig 10. Gráfico donde se representa el proceso de internalización del SARS-CoV-2 y su replicación.

3. Respuesta inmunitaria y desarrollo de estrategias contra el SARS-CoV-2

3.1 Respuesta del sistema inmune

Dentro de la respuesta del sistema inmune podemos diferenciar:

Inmunidad innata: después de la entrada del virus a las células, el sistema inmune innato monta una respuesta inmediata e inespecífica en contra del virión al reconocer el RNA viral. El material genético del SARS-CoV-2, al igual que el de otros virus, es un patrón molecular asociado con patógeno o PAMP, el cual es reconocido por los receptores de reconocimiento de patrones como TLR-3 o TLR-7 que se encuentran en las células presentadoras de antígenos. Esto desencadenan cascadas de señalización que llevan a la producción de interferones, IFN, de tipo I y a la activación del factor de transcripción NF κ B. Los IFN de tipo I a través de la vía de señalización de STAT inician la transcripción de proteínas antivirales codificadas en genes estimulados por IFN. Estas, junto con otros mediadores de inflamación liberados como consecuencia del daño citopático, reclutan macrófagos y neutrófilos, lo que resulta en una hiperproducción de citocinas proinflamatorias y especies reactivas de oxígeno que favorecen la vasodilatación y fuga vascular, que en conjunto causan hipoxia.

No existe evidencia directa del papel fisiopatológico de los mediadores de inflamación en COVID-19, pero si se ha visto que el grado de respuesta inflamatoria se asocia con la gravedad de la enfermedad. Por lo tanto, una respuesta inmune excesiva o mal regulada podría ser el punto central de la fisiopatología en COVID-19, y la causa de que los pacientes presenten síndrome de insuficiencia respiratoria aguda, choque séptico o falla orgánica múltiple que han provocado el colapso del sistema sanitario. Poder entender estos mecanismos y aprender a controlarlos/interpretarlos ayudaría enormemente a el control de la pandemia.

Inmunidad adaptativa: la respuesta celular en pacientes con COVID-19 está mediada principalmente por linfocitos T CD8+, que son los responsables de la destrucción directa de las células infectadas. Por otro lado, los linfocitos T, los CD4+, desempeñan también un importante papel en el control del virus al incidir en la generación de la respuesta humoral, ya que la respuesta celular intensa se relaciona directamente con una mayor producción de anticuerpos neutralizantes contra el virus.

La respuesta humoral, mediante la producción de anticuerpos neutralizantes, tiene un papel importante en el control de la infección y también en la prevención de una reinfección. Estos reducen la infectividad del virus al unirse a algunas proteínas virales de superficie y así prevenir la unión al virus con su receptor y, por consiguiente, su internalización. También promueven la lisis del virus mediante opsonización o activación del complemento. Esto evidencia la necesidad que había por encontrar una vacuna que provocase una inmunidad adaptativa ante el virus sin necesidad de contraer la enfermedad, ya que de esta forma evitamos que continúe infectando a persona y protegemos también a aquellos inmunodeprimidos que son más vulnerables ante infecciones.

Tras la recuperación de la infección, se asume que las personas son inmunes a una reinfección a corto plazo, como sucede con cepas estacionales de coronavirus, en donde es muy poco probable que se vuelva a infectar con la misma cepa en esa temporada o la siguiente. Aún así, se han reportado casos de personas que han vuelto a contraer la enfermedad, por lo que debe estudiarse cuánto dura la inmunidad producida, o si por ejemplo han sido contagiadas por una nueva cepa.

Esta línea de estudio también es muy importante cuando lo aplicamos a las nuevas vacunas que se utilizan actualmente, pues todavía no sabemos cuánto dura la inmunidad que producen, lo que levanta incógnitas sobre su efectividad a largo plazo. Aún así, e incluso con el poco tiempo que se les lleva dando uso, su efectividad para combatir la pandemia a corto plazo está quedando demostrada y solo queda ver cómo se comportará en el futuro.

3.2 Mecanismo de evasión de la respuesta inmune

Una de las características de los coronavirus es su capacidad de evadir a respuesta inmune, principalmente mediante la inhibición o retraso en la síntesis de IFN tipo I a través de nsp1 y nsp3, lo que permite una mayor replicación viral y daño pulmonar, así como la inducción de apoptosis de células T, generando linfopenia y, por lo tanto, una respuesta inmune mal controlada. Además de esto, evitan que su RNA sea reconocido al replicarse dentro de vesículas de doble membrana que no tienen receptores de reconocimiento de patrones. El hecho de que pacientes asintomáticos con COVID-19 transmitan el virus puede ser un indicador de una respuesta retardada del sistema inmune innato para el control viral.

La respuesta inmune que se genera tras la infección es un factor clave en la enfermedad, ya que se puede derivar en un control eficiente del virus o en la generación de una respuesta mal regulada en la que hay un exceso en la producción de citocinas proinflamatorias asociado con una respuesta celular y humoral subóptima, lo que se relaciona con mayor daño pulmonar y por consiguiente con una mayor mortalidad de los pacientes.

Esto ha generado una serie de preocupaciones respecto a la enfermedad pues, aunque no tiene una mortalidad tan alta como otras, no llegamos a comprender la totalidad de efectos secundarios a corto o largo plazo que puede tener en nuestro cuerpo, por lo que se han estado estudiando métodos para prevenir la enfermedad, como las vacunas de Moderna o Pfizer, o para tratarla y reducir sus síntomas. Además, debido a su alta tasa de propagación es una enfermedad que puede colapsar fácilmente los servicios sanitarios, lo que hace más grave su efecto como pandemia y apremia más que nunca a la sociedad para buscar una solución que permita controlarla.

Aunque actualmente ya ha comenzado la vacunación contra la COVID-19 en varios países del mundo con las vacunas anteriormente mencionadas, esto no quiero decir que se deba dejar de lado la investigación de métodos distintos para prevenir/tratar la enfermedad, ya que la acumulación de conocimiento podría darnos nuevas tácticas para combatir a este y a otros virus.

Ahora voy a hablar un poco sobre una de las tácticas usadas para combatir el SARS-CoV-2:

Anticuerpo CR3022 Fab

Este anticuerpo se une al epítipo conformacional de la estructura cristalina que me ha tocado a mí estudiar. Varios grupos han caracterizado potentes anticuerpos dirigidos a la proteína S del coronavirus que neutralizan eficazmente el SARS-CoV in vitro e in vivo. Según el informe de *Tian y cols(2020)*, el anticuerpo fab CR3022, obtenido de un paciente convaleciente infectado por el SARS-CoV, se une de forma potente al dominio de unión al receptor (RBD) de la proteína S del SARS-CoV-2, y por lo tanto representa un importante anticuerpo monoclonal o mAb candidato con potencial como molécula terapéutica, solo o en combinación con otros posibles candidatos para tratar el COVID-19.

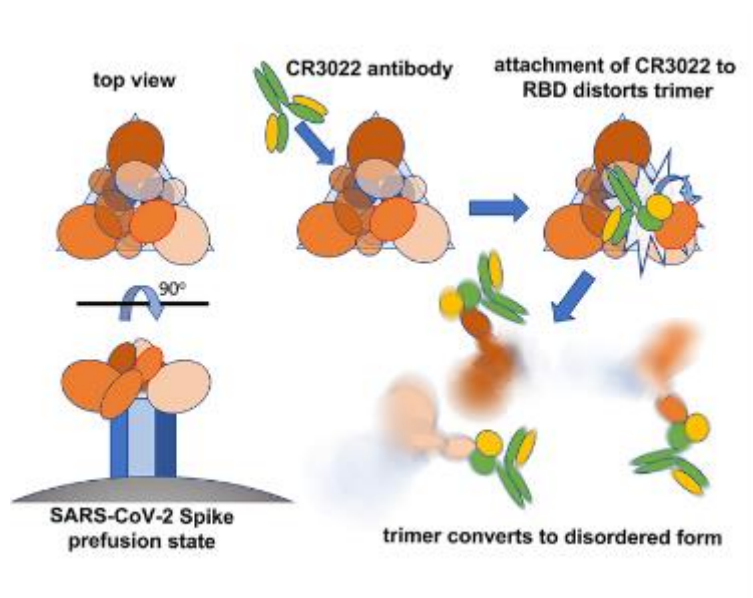


Fig 11. Mecanismo de acción del mAb CR3022. Se observa como el Ac se une al epítipo del trímero RBD, lo que provoca la desestabilización de la proteína S.

El fragmento de unión al antígeno, o Fab, es una región de los anticuerpos, o Ab, que es la que se une a los antígenos, o más concretamente a los determinantes antigénicos o epítomos, en nuestro caso con el epítipo CR3022, que está altamente conservado, estabiliza la estructura del complejo de unión y es inaccesible en la spike pre-fusión, lo que sugiere que la unión de CR3022 facilita la conversión al estado de post-fusión incompetente para la fusión.

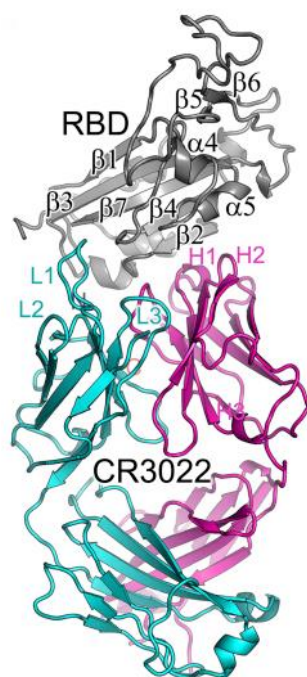


Fig 12. En la imagen observamos un diagrama que muestra la estructura del complejo RBD/CR3022 con el RBD mostrado en gris, visualizado en YASARA, la cadena pesada del CR3022 en magenta y la cadena ligera en cian. Las CDR1-3 de la cadena pesada están etiquetadas como H1-H3 y las CDR1-3 de la cadena ligera como L1-L3 (donde son visibles).

El análisis de microscopía electrónica criogénica (cryo-EM) utilizado en la investigación realizada por *Jiandong Huo* y cols. confirma que la incubación de la espiga con CR3022 Fab conduce a la destrucción del trímero de prefusión. La presentación de este epítipo críptico en una vacuna basada en RBD podría centrar ventajosamente las respuestas inmunitarias. Los aglutinantes contra epítipo podrían ser útiles terapéuticamente si pudiésemos conseguir que trabajasen sinergia con un anticuerpo que bloquee la fijación del receptor.

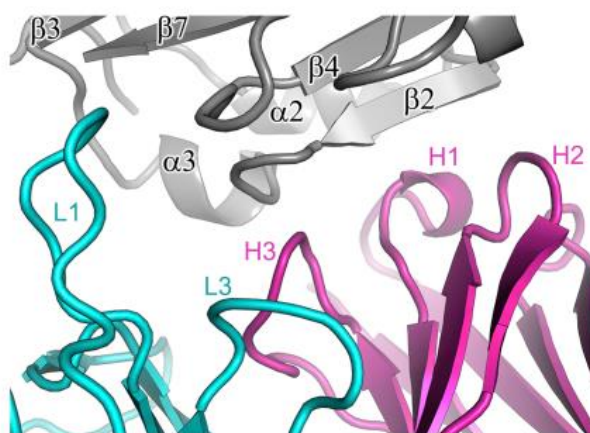


Fig 13. Imagen de la unión antígeno-anticuerpo en representación cartoon obtenido en YASARA (*Jiandong Huo* y cols)

La razón de la conservación del epítipo CR3022 se aclara en el contexto de la estructura S completa previa a la fusión, donde el epítipo es inaccesible (Figura 8). Cuando el RBD está en la configuración descendente, el epítipo CR3022 está empaquetado estrechamente contra otro RBD del trímero y el dominio N-terminal (NTD) del protómero vecino. En la estructura de la forma de pre-fusión de la espiga trimérica, la mayoría de los RBDs están hacia abajo, aunque presumiblemente estocásticamente uno podría estar hacia arriba.

La estructura de un complejo SARS-CoV-1 con el ectodominio ACE2 muestra que esta configuración hacia arriba es competente para unirse al receptor y que hay una familia de orientaciones hacia arriba con ángulos de bisagra significativamente diferentes. Sin embargo, el epítipo CR3022 sigue siendo en gran medida inaccesible incluso en la configuración hacia arriba. La modelización de la rotación del RBD necesaria para permitir la interacción de Fab en el contexto del trímero de espiga utilizada en el trabajo del grupo de investigación muestra que se requeriría una rotación correspondiente a una declinación $>60^\circ$ más del eje vertical central, aunque esto podría mitigarse en parte por movimientos más complejos del RBD, y si más de un RBD está en la configuración ascendente entonces este requisito se relajaría un poco.

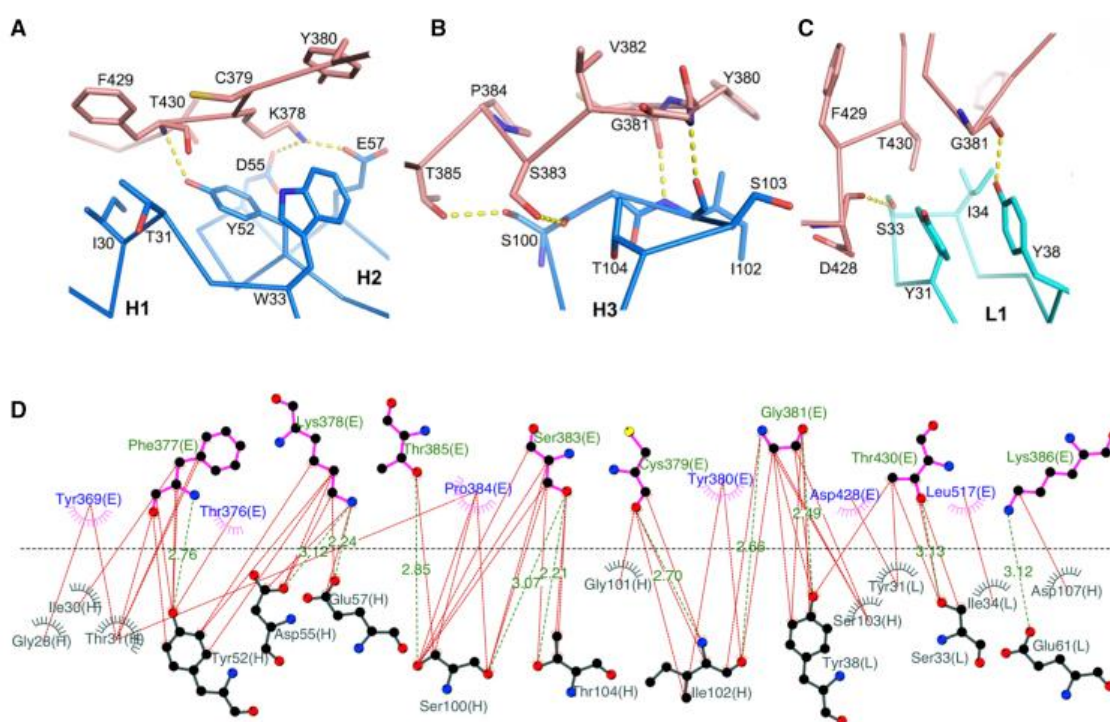


Fig 14. En esta imagen tenemos representado más en detalle los contactos entre el RBD y el mAb CR3022. En las imágenes A y B, tenemos los contactos de la RBD con las CDR1 (H1) y CDR2 (H2) de la cadena pesada del CR3022 (A), y con la CDR3 (H3) (B). En la imagen C, tenemos las interacciones entre el RBD y la cadena ligera CDR1 (L1). Las columnas vertebrales de la cadena principal se muestran como palos más finos y las cadenas laterales como palos gruesos (RBD, salmón; cadena pesada, azul; cadena ligera, cian). Los palos amarillos rotos representan enlaces de hidrógeno o puentes salinos. Por último, en la imagen D tenemos la representación en Ligplot de las interacciones (Laskowski and Swindells, 2011).

Debido a que el bloqueo del estado ascendente por parte de los anticuerpos bloqueadores de receptores desestabiliza el estado de pre-fusión, la unión del CR3022 presumiblemente introduce una mayor desestabilización, llevando a una conversión prematura al estado de post-fusión, lo que inactiva el virus. El CR3022 y los anticuerpos bloqueadores de la ACE2 pueden unirse independientemente, pero ambos inducen una conformación hacia arriba, lo que presumiblemente explica la sinergia observada entre la unión en los dos sitios.

Algunos informes (Yuan et al.,2020) indican que la utilización de mAB CR3022 no neutraliza el SARS-CoV-2, aunque esto podría deberse a que el grupo remueve la disolución con Ac/virus una vez se da la adsorción en las células indicadoras antes de la incubación, lo que impediría que se llegará a formar la forma pre-fusión que deja al descubierto a nuestro epítipo. De todas formas, el mecanismo de neutralización de CR3022 parece no ser bien detectado por todos los grupos de investigación lo que pone un poco en duda la rentabilidad a la hora de utilizarlo en métodos de control del virus, aunque los resultados obtenidos por los grupos que si han podido medirlo son muy alentadores. Además, estos anticuerpos pueden ser producidos en plantas como la *Nicotiana benthamina*, lo que facilitaría mucho su producción y rebajaría el costo de posibles tratamientos que lo utilicen.

4. Actividades

Una vez terminada la revisión bibliográfica, produce a realizar el resto de actividades del CPA. Cuando se asignaron las proteínas para realizar el trabajo a mi se me asignaron distintos ficheros que corresponden a estructuras cristalinas del RBD del SARS-CoV-2 en complejo con distintos anticuerpos, y yo he elegido el fichero 6yla, que se corresponde con la interacción con el mAb CR3022 para realizar las actividades del CPA debido a que es el que más interesante me ha parecido.

4.1 Bases de datos secuenciales y estructurales

4.1.1 Detección de formatos de ficheros proteicos y extracción de la secuencia primaria

En esta actividad se nos pide que creamos un diagrama de flujo que nos permita identificar si un fichero tiene formato EMBL, UniProt, GenBank u PDB. Yo lo he realizado el siguiente a través de la página web Canvas:



Fig 15. Diagrama de flujo que nos permite identificar un fichero y asignarle un formato determinado.

Que si lo pasamos a nuestro entorno de programación quedaría así:

```
//Usando bucles if detectamos el formato del documento que cargamos en
//el memo 1, y luego usamos la ecuación que tenemos en biotools para extraer la
//la secuencia
formato:='Formato no detectado';
if (copy(documento[0],0,6)= 'HEADER')
then formato:= 'PDB';

if (copy(documento[0],0,2)='ID') and (copy(documento[1],0,2)='XX')
then formato:= 'EMBL';

if (copy(documento[0],0,2)='ID') and (copy(documento[1],0,2)='AC')
then formato:= 'UniProt';

if (copy(documento[0],0,5)='LOCUS')
then formato:= 'GenBank';
```

Fig 16. Fragmento de código correspondiente al button 2 con el captión ‘Detectar formato’ de la aplicación ProgramaDiscriminar.

Siguiendo las bases planteadas por este diagrama, he procedido a crear un programa informático que implemente un código que nos permita realizar la diferenciación entre distintos tipos de formatos de fichero:

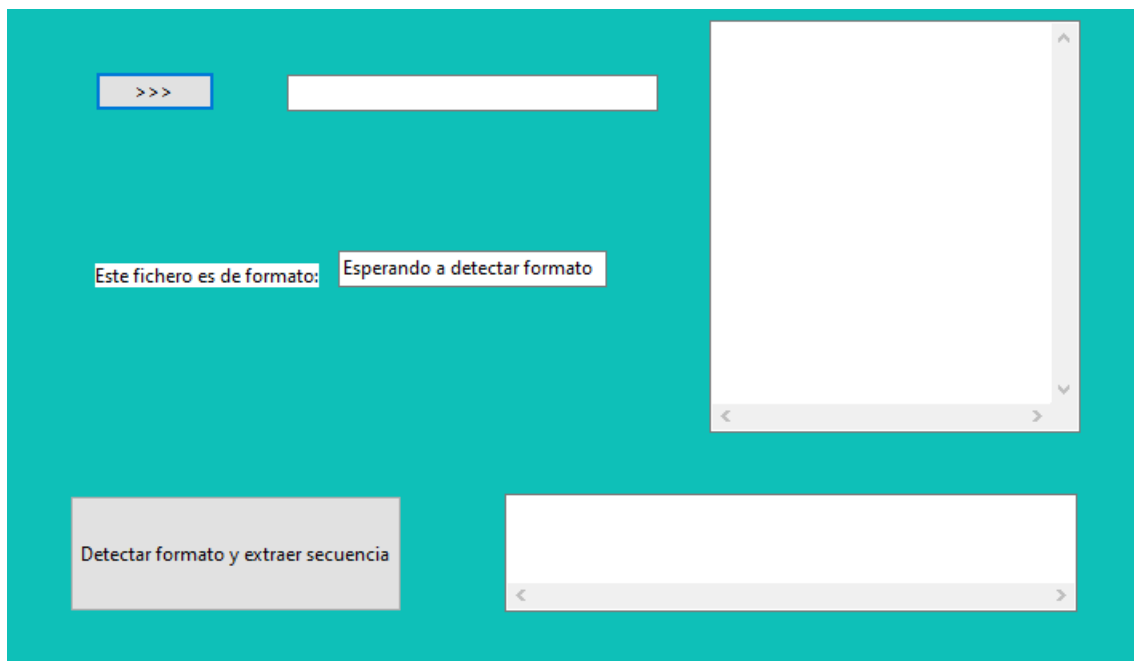


Fig 17. Interfaz de la aplicación ProgramaDiscriminar, que nos permite cargar un fichero, detectar su formato y extraer su secuencia.

Este programa nos permite detectar el formato del fichero y, a su vez, extraer la secuencia en códigos de una letra:

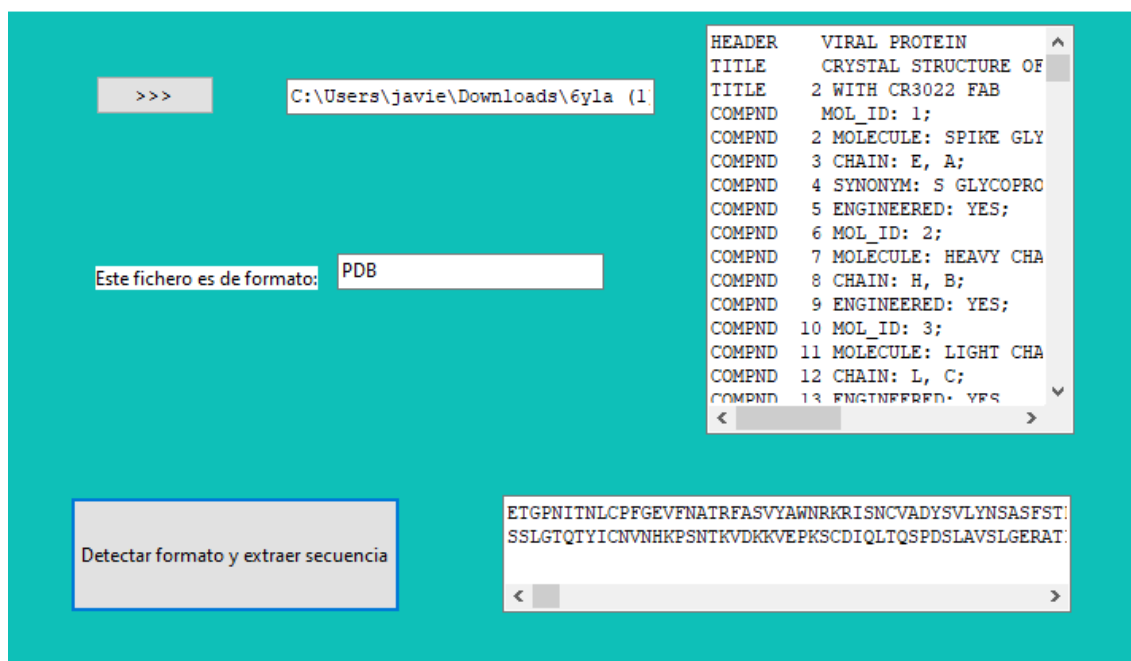


Fig 18. En esta imagen se puede observar el resultado obtenido al utilizar mi aplicación con mi fichero asignado, (6yla).

4.1.2 Desarrollo de la función CargarPDB

Se nos pide que desarrollemos una función de parsing, que denominamos CargarPDB, y que nos permita representar adecuadamente los datos estructurales relevantes de una proteína en nuestro entorno de programación, que en nuestro caso es Pascal. Esto es algo esencial a la hora de realizar cualquier análisis proteico que realicemos a continuación, por lo que extraer esta información de un fichero nos permitirá realizar diferentes cálculos con ellos.

Esta función la realizamos junto con el profesor a lo largo del curso y ha sido una de las más importantes de la asignatura pues es parte de la base que han utilizado el resto de aplicaciones desarrolladas.

Esta función tenía que ser capaz de leer los datos desde un fichero PDB y generar un tipo de datos especial, el TPDB, que funciona a modo de cajón donde se guardan los átomos, residuos y subunidades de una manera ordenada con toda la información relevante de cada uno como el número de átomo, su tipo, en qué subunidad está, el residuo al que pertenece, ...

Con este objetivo, TPDB está conformado por otros subtipos especial (TatomPDB, TResiduoPDB y TsubunidadPDB, cada uno con sus propias peculiaridades que permiten acceder de forma rápida y eficiente a la información.

```

□ Tpunto = record //almacena las coordenadas X, Y, Z de un átomo
  X, Y, Z: real;
end;

□ TatomPDB = record //átomo de la prot
  NumAtom: integer;
  ID: string;
  Residuo: string; //tipo de residuo (O,N, CO...)
  Subunidad: char;
  NumRes: integer; //número de residuo
  coor: Tpunto;
  AltLoc: char; //posición alternativa
  Temp: real;
end;

TPuntos = array of Tpunto;

□ TResiduoPDB = record //almacena varios datos sobre un residuo
  NumRes: integer;
  Subunidad: char;
  ID3: string;
  ID1: char; //denominación de una letra
  Atm1, AtmN: integer;
  N, CA, C, O: integer; //apunta a la ficha de estos átomos del residuo
  phi, psi: real;
end;

TPDB = record //estructura de datos que almacena toda la información útil sobre una proteína
  header: string; //primera línea del archivo PDB que da info de la prot
  atm: array of TatomPDB; //conjunto de átomos
  res: array of TResiduoPDB; // array de los residuos
  sub: array of TsubunidadPDB;
  subs, secuencia: string; //todas las letras que denominan cada subunidad como 'ABCD'
  //Secuencia primaria de la proteína KMY...
  NumFichas, NumResiduos: integer; //como a veces no ponen todos los átomos
  //vamos a contar nosotros los átomos que haya
  NumSubunidades: integer; // número total de átomos y de residuos

  AtomIndex: array of integer;

```

Fig 19. Código de los distintos subtipos utilizados por la función CargarPDB junto con sus respectivos comentarios que explican su función. Sacado de nuestra función biotools.

Mi versión de esta función es básicamente la misma que tienen mis compañeros y no he hecho variaciones en cuanto a su funcionamiento, aunque se puede resumir en que consiste en rellenar la información de los diferentes subtipos que forman el tipo TPDB con solo leer una vez el archivo PDB. La información sobre su funcionamiento se puede ver en los comentarios de biotools.

4.2 Visualización de proteínas

4.2.1 Ficheros de estructuras

Como ya he comentado anteriormente, a mi me fueron asignados varios ficheros de estructuras correspondientes a distintas interacciones y yo he decidido enfocarme más en el fichero 6yla. A lo largo de la revisión bibliográfica ya he hablado bastante de su estructura y función, por lo que para esta actividad remito a la información utilizada anteriormente pues sería redundante volver a explicarlo todo.

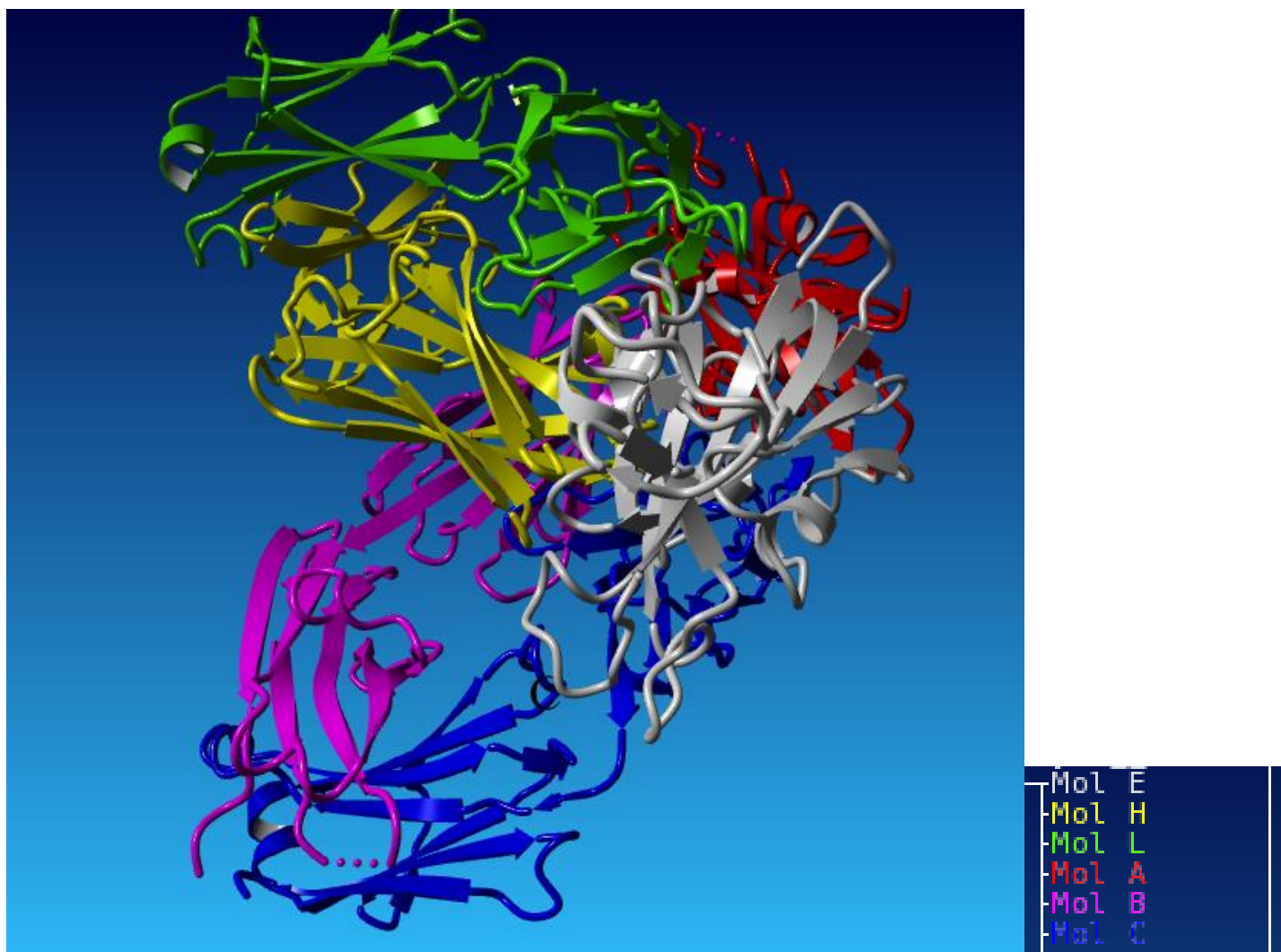


Fig 20. Estructura del fichero 6yla, donde podemos observar las distintas subunidades coloreadas.

4.2.3 Desarrollo de la función EscribirPDB

La función WritePDB, o EscribirPDB en mi caso, es una función que realiza una función opuesta a CargarPDB. Esta función nos permite reescribir una línea ATOM de un fichero PDB a partir de un tipo TPDB, y de esta forma recrear las líneas correspondientes a los átomos, sin cambiar su estructura con lo que permitimos que luego pueda ser usado por un programa de visualización de imágenes como YASARA.

```
function writePDB(atm: TAtomPDB): string;
var
  k: integer;
  pdb: TStrings;
  linea, numatm, numre, coordenadaX, coordenadaY, coordenadaZ, temp: String;
  s: string;
begin
  //Lo primero que hacemos es obtener las cadenas del número de átomo, número de residuo y las coordenadas con el
  //formato de tres decimales que nos interesa.
  numatm:= inttostr(atm.Numatom);
  numre := inttostr(atm.NumRes);
  coordenadaX := formatfloat('0.000', atm.coor.X);
  coordenadaY := formatfloat('0.000', atm.coor.Y);
  coordenadaZ := formatfloat('0.000', atm.coor.Z);
  temp := formatfloat('0.00', atm.Temp);

  linea:= 'ATOM'+ ' ' + indder(5, numatm.Length, numatm) + ' ' + indizq(3, atm.ID.Length, atm.ID) +
  ' ' + atm.Residuo + ' ' + atm.subunidad + ' ' + indder(3, numre.Length, numre) +
  ' ' + indder(7, coordenadaX.Length, coordenadaX) + ' ' +
  + indder(7, coordenadaY.Length, coordenadaY) + ' ' +
  indder(7, coordenadaZ.Length, coordenadaZ) + ' 1.00 ' + indder(5, temp.Length, temp);
  //Una vez que todo está listo, se encuadra en la línea con los espacios adecuados.
  //Las coordenadas tienen una longitud máxima de
  // 7 porque hay que tener en cuenta el posible signo - que tengan.
  result:= linea;
end;
```

Fig 20. Código correspondiente a la función writePDB que posteriormente utilizo en mi función EscribirPdb. Estraida de mi librería biotools.

Para el desarrollo de la función writePDB era muy importante tener cuidado con la indentación, lo que me dio problemas en un principio. Cuando me vi atascado decidí buscar información en internet y encontré la siguiente página: <https://webprogramacion.com/156/pascal/alinear-cadena-por-la-derecha.aspx>

De la que saqué las siguientes funciones que posteriormente he utilizado. Lo que me demuestra, una vez más, lo importante de colaborar entre programadores ya que nos permite solucionar unos problemas que podemos tener todos.

```
function IndDer(maxlong, long: integer; var s: string): string;
const
  blanco=' ';
var
  k:integer;
begin
  if maxlong <> long then
    for k:=1 to maxlong-long do insert( blanco,s,k);
    result:= s;
  end;

function IndIzq(maxlong, long: integer; var s: string): string;
const
  blanco=' ';
var
  k:integer;
begin
  if maxlong <> long then for k:=1 to maxlong-long do
  begin
    s:= s+blanco;
  end;
  result:= s;
end;
```

Fig 21. Código utilizado para realizar la indentación necesaria para la función writePDB. Código extraído de mi biotools.

Una vez hecho esto se nos pide escribir un programa que genere un fichero PDB que contenga solo los carbonos alfa de la proteína para posteriormente visualizarlos:

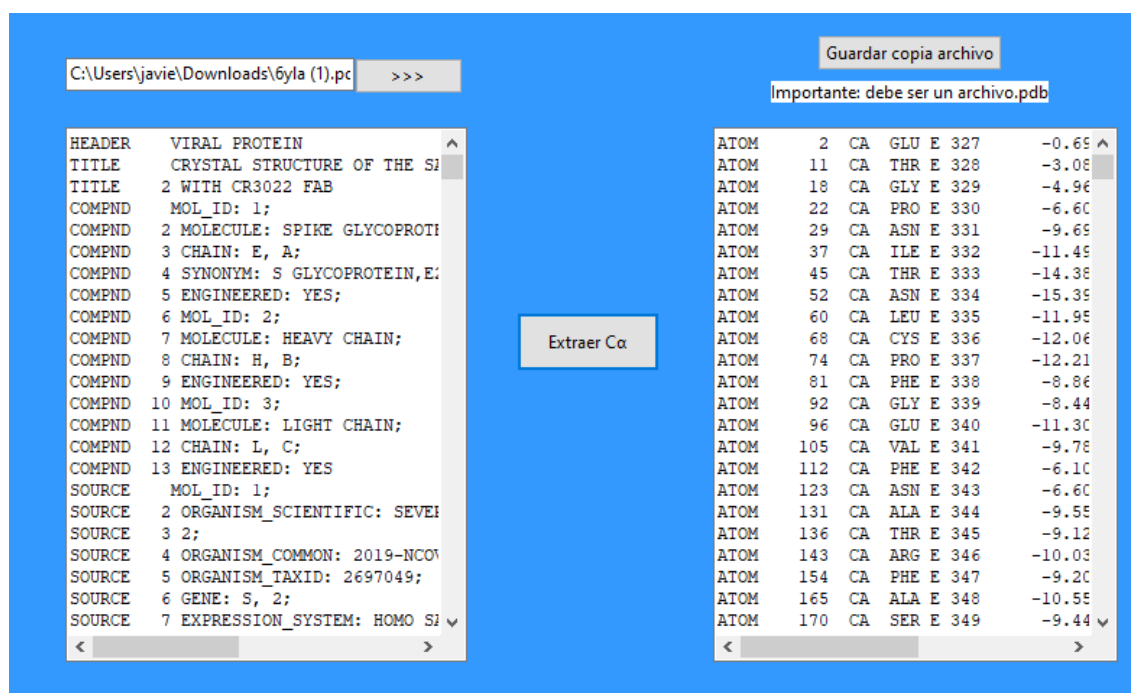


Fig 22. Interfaz de mi aplicación EscribirPDB. Se pueden apreciar un primer botón que permite cargar la proteína, un segundo botón con el que extraemos los carbonos alfa del fichero y un tercer botón que nos permite guardar el fichero generado.

Este programa nos permite cargar un fichero utilizando el mismo código que hemos usado para las demás aplicaciones y después nos permite guardar el fichero con solo los CA, que posteriormente deberíamos de poder visualizar en otros programas.

```

procedure TForm1.Button2Click(Sender: TObject); //Botón cuya función es extraer los CA del fichero
var
  j: integer;
begin
  memo2.visible:= false;
  for j:= 1 to p.NumFichas do
  begin //Si hay 'CA' guardamos sus valores
    if p.atm[j].ID= 'CA' then memo2.Lines.Add(writePDB(p.atm[j]));
    end;
  memo2.Visible:= true;
end;

```

Fig 23. Líneas de código correspondientes al button 2 que extrae los carbonos alfa del fichero. Código extraído de *EscribirPDB_unit1*.

En mi caso he utilizado YASARA para comprobar la proteína y, efectivamente y como podemos observar en las siguientes fotos, ha realizado bien su función pues ha cogido solo los CA de la proteína y no ha cambiado su estructura:

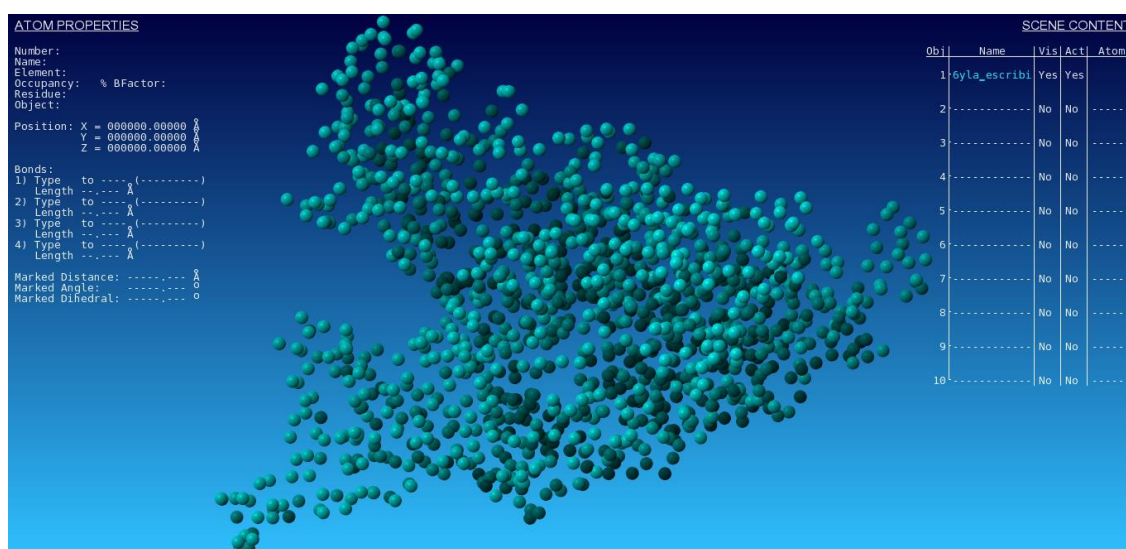


Fig 24. Representación en YASARA de los carbonos alfa extraídos por el programa *EscribirPDB*.

En la fig 24 tenemos representados en YASARA los átomos CA que aparecen en el fichero que ha creado *EscribirPDB*, mientras que en la fig 25 podemos observar el esqueleto covalente de los átomos coloreados en función de su temperatura en RasMol. El valor de temperatura o factor-B es un parámetro de cada átomo que describe el grado de extensión de la densidad electrónica, de modo que es indicativo de si la movilidad del átomo es estática o dinámica. En RasMol, los átomos con valores de temperatura bajos se colorean en azul, mientras que los valores altos se colorean en rojo.

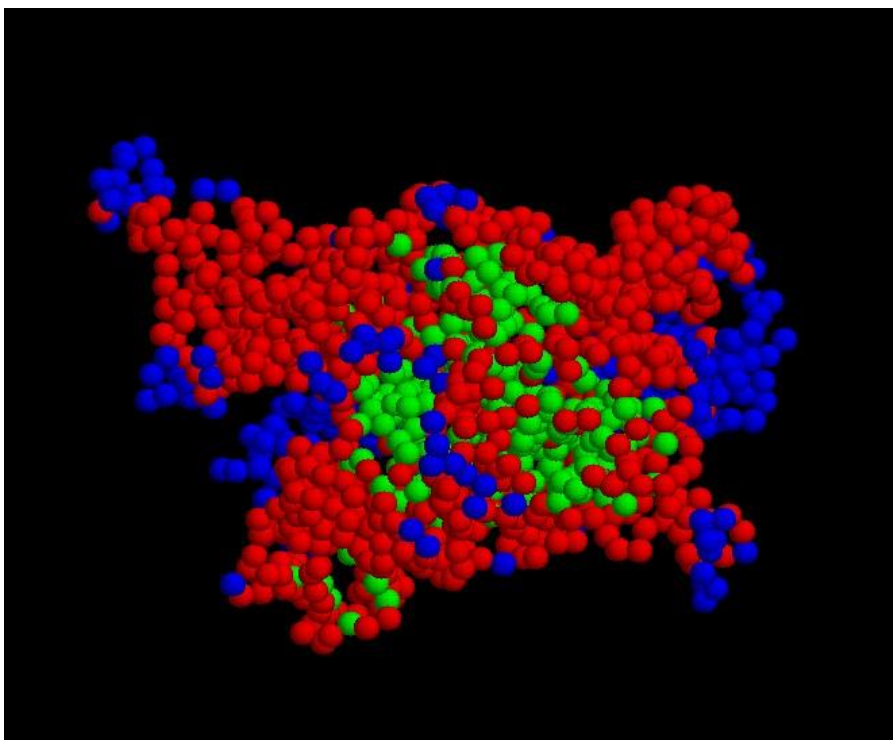


Fig 25. Representación en RasMol de los carbonos alfa obtenidos por mi aplicación EscribirPDB y coloreados en función de su valor de temperatura.

Tal y como se puede observar en la representación en RasMol de mi proteína, una gran parte de la misma, y la más exterior, posee un valor de temperatura bastante alto, lo que tiene sentido si atendemos al hecho de que es un epítipo conformacional que aparece en la proteína spike pre-fusión, por lo que debe poseer átomos muy dinámicos. Además, estos átomos con mayor valor de temperatura se encuentran en la parte más exterior de la estructura, lo que nos indica que esta es la zona que más tiende a reorganizarse a la hora del cambio estructural.

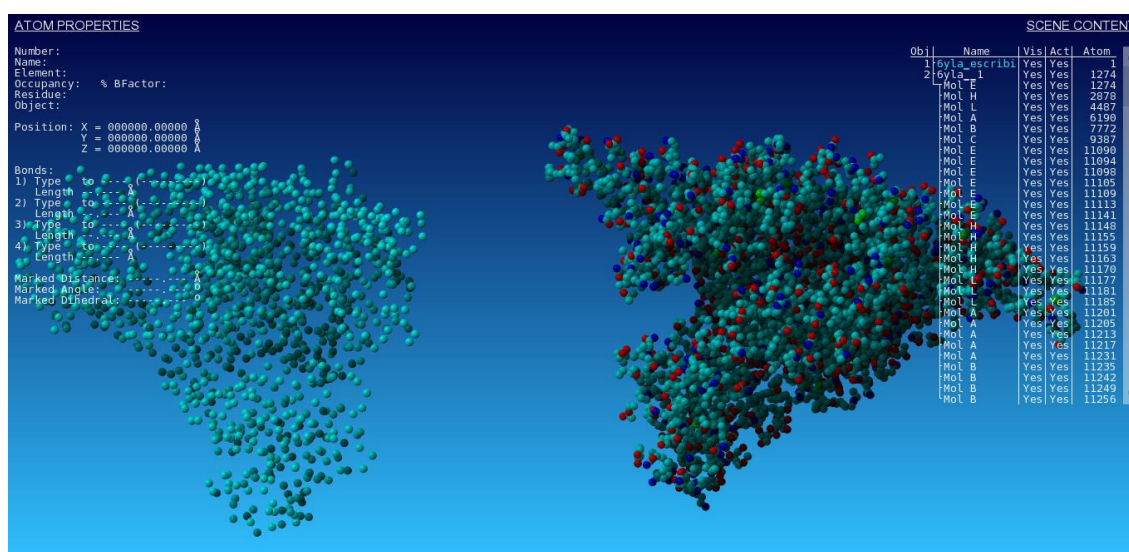


Fig 26. Representación en YASARA de mi fichero obtenido de escribirPDB, en la izquierda, con el fichero 6yla, en la derecha, donde se puede observar mejor la extracción de estos carbonos alfa.

4.3 Predicción estructural y resiñeño de proteínas

4.3.1 Diagrama de Ramachandran

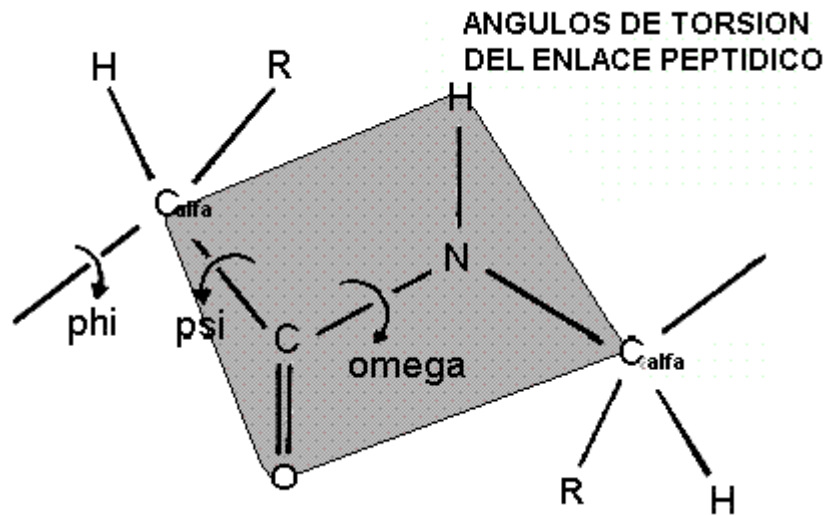


Figura 27. Se muestran los tres ángulos de torsión principales del enlace peptídico o de un polipéptido: phi (F), psi (Y), y omega (W).

En cada AA, el Ca puede establecer dos ángulos de torsión con los planos de los dos enlaces peptídicos contiguos: los llamados ϕ (phi) y ψ (psi) :el ángulo ϕ es la rotación en torno al enlace Ca-N y el ángulo ψ es la rotación en torno al enlace C-Ca.

En la estructura de un péptido o de una proteína, cada AA presenta un valor de phi y otro de psi determinados. Es el tamaño físico de los átomos o grupos de átomos presentes en las cadenas laterales el que determina los posibles valores que pueden adoptar los ángulos phi y psi. Conociendo todos los valores de psi y de phi podríamos determinar la estructura de una proteína.

Si representamos los valores de phi y de psi de cada aminoácido obtenemos la gráfica de Ramachandran, en la que se distinguen tres regiones con valores “permitidos” de los ángulos de torsión ya que no dan lugar a impedimentos estéricos en la estructura.

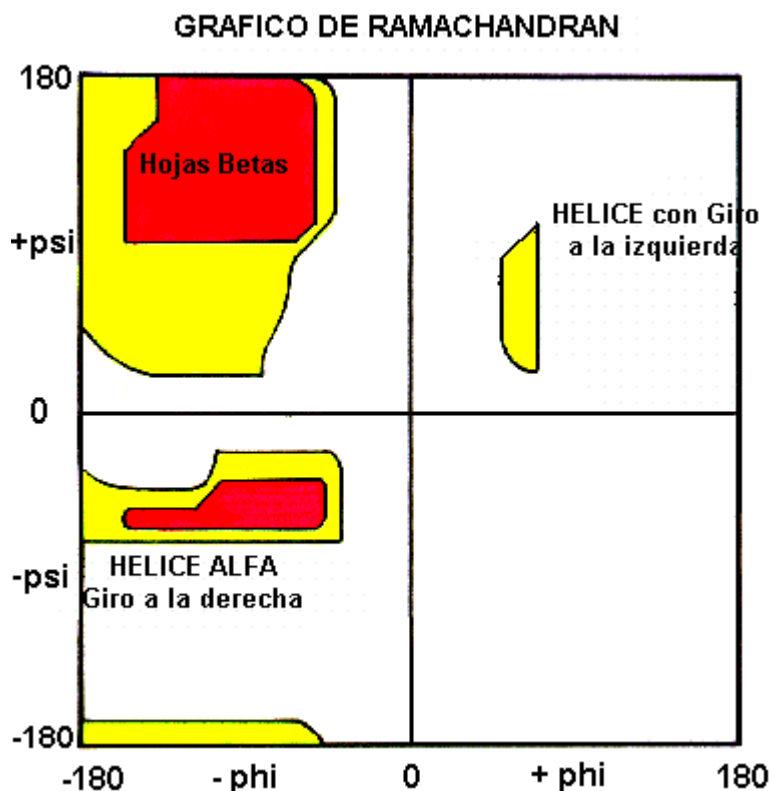


Fig 28. Las regiones coloreadas corresponden a los dos tipos de elementos estructurales mayoritarios presentes en las proteínas: la estructura beta, en el cuadrante superior izquierdo, y la alfa-hélice, en los cuadrantes superior derecho e inferior izquierdo.

Para realizar este cálculo nosotros hemos definido en nuestra matriz de datos TResiduosPDB los parámetros phi y psi, dentro de la librería biotools. Con estos parámetros y valiéndonos de otras funciones anteriormente definidas en la librería como la función torsión, que calcula el valor del ángulo de torsión entre cuatro puntos, hemos desarrollado un programa que nos calcula y representa el Ramachandran Plot:

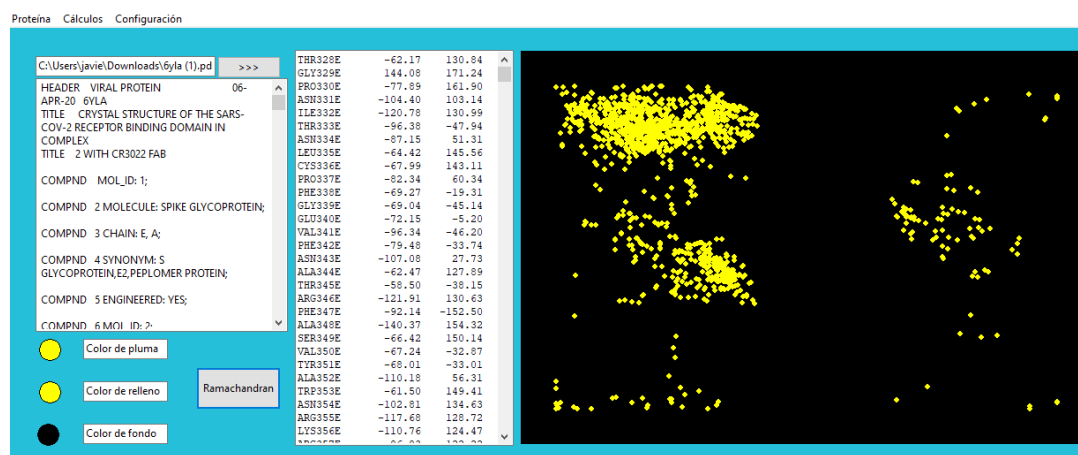


Fig 29. Interfaz de mi aplicación Ramachandran, donde nos permite cargar el fichero, elegir los colores que queremos utilizar y calculas el Ramachandran plot. El que aparece representado es el correspondiente a mi fichero 6yla. Podemos observar la gran cantidad de lámina beta que posee la proteína, lo que se corresponde con la información aportada por el PDB.

Una característica que ofrece el programa es que podemos seleccionar el color del perímetro del círculo con el que se representa cada punto (cpluma), así como el relleno de estos puntos (clrelleno). Como la función plotXY que usamos ofrece la opción de no borrar el gráfico anterior esto puede utilizarse para representar solapados varios gráficos de Ramachandran correspondientes con distintas proteínas.

Respecto al código, la gran mayoría es idéntico al desarrollado durante las sesiones de clase, salvo por el nombre de algunas variables, por lo que no veo necesario añadirlo aquí. Así que en caso de necesidad de revisar el código está todo en la Ramachandran_unit1.

Para comprobar si los valores de phi y de psi que calcula nuestro programa son correctos los he comprobado utilizando la página MolProbity, que ha sido elaborada por la Universidad de Duke, cuyos resultados aparecen en la tabla siguiente, y efectivamente son idénticos a los calculados por el programa.

```
Starting job
=====
Validating inputs
Inputs OK
Individual residues info:
Residue name, type, SS, (phi, psi), Z
THR   General L ( -62.17, 130.84) 1.2231
GLY    Gly L ( 144.08, 171.24) -0.7460
PRO   cis-Pro L ( -77.89, 161.90) 1.2425
ASN   General L (-104.40, 103.14) -0.4029
ILE   Ile/Val L (-120.78, 130.99) 1.4937
THR   General L ( -96.38, -47.94) -1.2836
ASN   General L ( -87.15,  51.31) -0.8633
LEU   General L ( -64.42, 145.56) 1.1618
CYS   pre-Pro L ( -67.99, 143.11) 2.0104
PRO   trans-Pro L ( -82.34,  60.34) -1.2793
PHE   General H ( -69.27, -19.31) -1.0752
GLY    Gly H ( -69.04, -45.14) -0.2588
GLU   General H ( -72.15,  -5.20) -1.2648
VAL   Ile/Val H ( -96.34, -46.20) -1.2508
PHE   General H ( -79.48, -33.74) -1.0707
ASN   General H (-107.08,  27.73) -0.8811
ALA   General L ( -62.47, 127.89) -0.3340
THR   General L ( -58.50, -38.15)  0.8709
ARG   General L (-121.91, 130.63) -0.6616
PHE   General L ( -92.14, -152.50) -1.5488
ALA   General L (-140.37, 154.32) -0.5872
SER   General L ( -66.42, 150.14)  0.4780
VAL   Ile/Val L ( -67.24, -32.87) -0.4163
TYR   General L ( -68.01, -33.01) -0.0090
ALA   General L (-110.18,  56.31) -0.9797
TRP   General S ( -61.50, 149.41) -1.5100
ASN   General S (-102.81, 134.63)  1.0129
ARG   General S (-117.68, 128.72)  1.3751
LYS   General S (-110.76, 124.47)  1.2339
```

Fig 30. Valores de phi y pshi calculados por la página MolProbity respecto a mi fichero 6yla. De esta forma comprobamos que los resultados obtenidos por mi aplicación son correctos

4.3.2 Coordenadas transformadas y construcción de un estereodiagrama

A lo largo del curso hemos creado unas funciones que nos permiten girar/mover la estructura proteica para poder representarla y moverla. Estas funciones son: traslacion, que nos permite realizar una traslación de las coordenadas X, Y, Z de un punto, giroOX (que nos permite calcular las nuevas coordenadas X, Y, Z de un punto cuando se realiza un giro sobre el eje OX), giroOY (que permite calcular las nuevas coordenadas X, Y, Z de un punto cuando se realiza un giro sobre el eje OY) y giroOZ (que permite calcular las nuevas coordenadas X, Y, Z de un punto cuando se realiza un giro sobre el eje OZ).

```

procedure GiroOY (rad: real; var p: TPDB); overload;
var
    seno, coseno:real;
    j:integer;
begin
    seno:= sin(rad);
    coseno:= cos(rad);
    for j:= 1 to p.NumFichas do
    begin
        p.atm[j].coor.X:= p.atm[j].coor.X*coseno + p.atm[j].coor.Z*seno;
        p.atm[j].coor.Y:= p.atm[j].coor.Y;
        p.atm[j].coor.Z:= -p.atm[j].coor.X*seno +p.atm[j].coor.Z*coseno;
    end;
end;

```

Fig 31. Fragmento de código correspondiente al GiroOY utilizado en el desarrollo de la aplicación. Extraído de mi biotools.

En esta actividad he utilizado la función giroOY para crear un estereodiagrama, que consiste en dos imágenes representadas una al lado de la otra que muestran el mismo objeto, pero girado 5 °, de modo que cuando se solapan ambas imágenes este se percibe como un único objeto tridimensional. Por lo tanto, he desarrollado un programa que coge un fichero PDB, calcula el número de Phe que posee y permite al usuario elegir respecto a cuál quiere realizar el estereodiagrama.

```

numphe:=0; //Calculamos el número de Phe que tiene la proteína para luego dar al usuario
//la oportunidad de usar la que quiera
for j:=0 to p.NumFichas do
begin
    if p.res[j].ID3='PHE' then numphe:=numphe+1;
end;
if numphe<>0 then spinedit1.maxvalue:=numphe;
spinedit1.visible:=true; //Hacemos visible el spinedit con el valor máximo de nuestra proteína
labell.visible:=true;
end;

```

Fig 32. Fragmento de código correspondiente al bucle utilizado para calcular el número de Phe de nuestra proteína y cómo asocia ese valor a un maxvalue de un spinedit que permite al usuario elegir a qué Phe quiere realizar el estereodiagrama.

El resto del código se encuentra en biotools, pero se resumen en el uso de distintos bucles if y for para buscar la Phe elegida por el usuario, almacenar sus valores y calcularles el giro para posteriormente representarlo. Lo he comprobado con la Phe 1 y Phe 5, que aparecen representados en las siguientes imágenes:

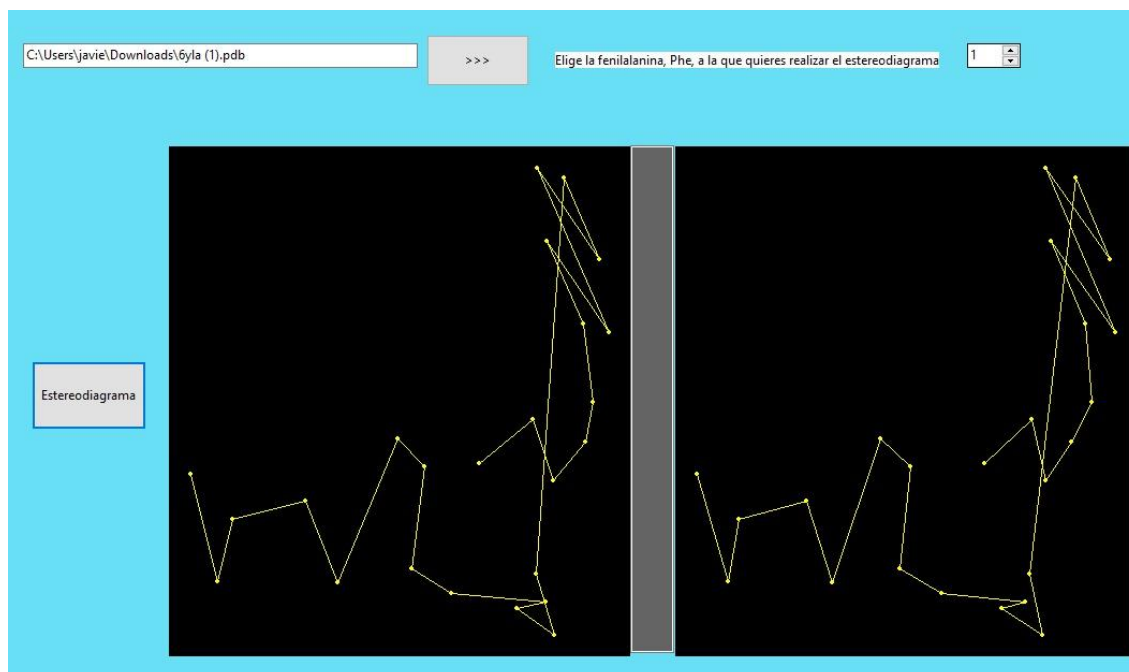


Fig 33. Imagen correspondiente al estereodiagrama de la Phe1 de nuestro fichero 6yla. Se pueden apreciar que la imagen de la derecha está girada 5 grados respecto a la otra.

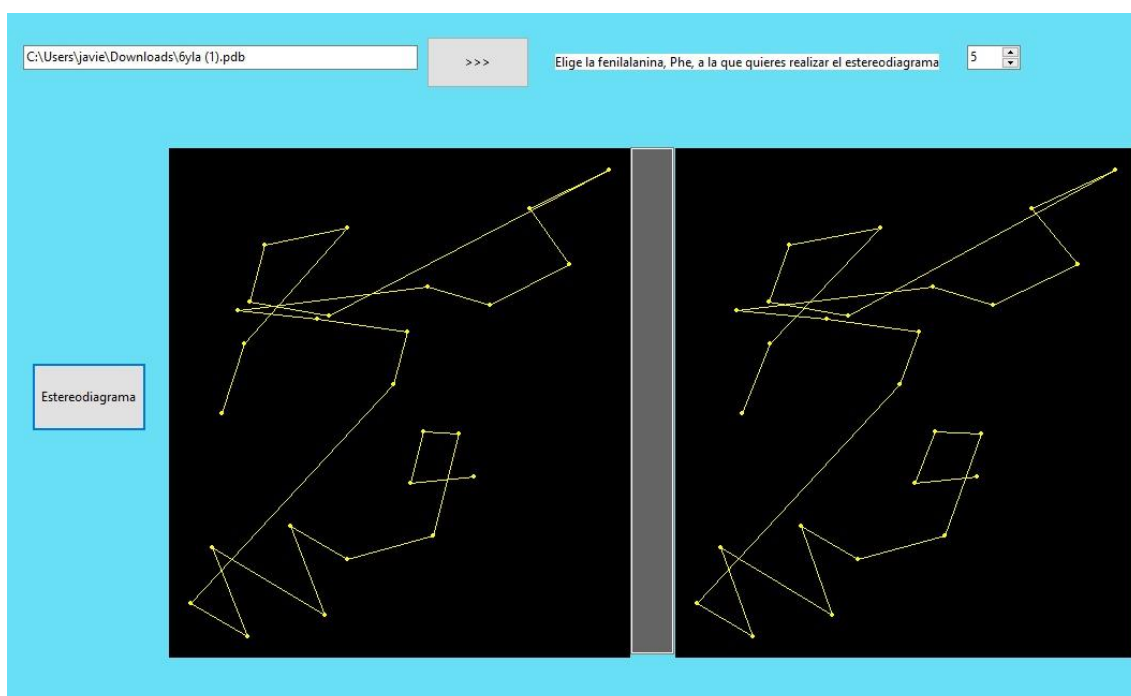


Fig 34. Imagen correspondiente al estereodiagrama de la Phe5 de nuestro fichero 6yla. Se pueden apreciar que la imagen de la derecha está girada 5 grados respecto a la otra.

4.3.3 Alinear Z

Para realizar esta actividad hemos desarrollado un programa denominado AlinearZ que nos permite representar los 10 primeros carbonos alfa de una proteína en un eje cartesiano donde consideramos el eje Z como el perpendicular a este, de modo que solo se representan las coordenadas X e Y de cada carbono alfa.

A continuación, realizamos una serie de transformaciones conformadas por traslaciones y giros que termina superponiendo el primer carbono alfa y el décimo en nuestro plano anterior. Las transformaciones que hemos realizado las podemos resumir en:

1. Los carbonos α se desplazan hasta que las coordenadas atómicas del primer carbono α son (0,0,0).
2. A continuación, rotamos la estructura respecto al eje OX hasta que el décimo átomo coincida con la trayectoria del eje X, siempre considerando el eje Z como perpendicular al plano, o dicho de otra forma, hasta que el valor de su coordenada Y sea 0.
3. Rotamos entorno al eje OY hasta que el décimo carbono α coincida con la trayectoria del eje Y cuando consideramos el eje Z como perpendicular al plano, o dicho de otra forma, cuando el valor de su coordenada X también es 0.
4. Giramos el número de grados necesarios.
5. Hacemos el inverso de los pasos 3, 2 y 1.

Siguiendo estos pasos conseguiríamos alinear ambos carbonos α . Los cálculos matriciales que seguimos en clase para realizar las funciones aparecen en la siguiente imagen:

1.- Traslación del punto P1 al origen

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_1 & -y_1 & -z_1 & 1 \end{bmatrix}$$

4.- Rotación sobre OZ un ángulo θ :

$$\begin{bmatrix} x'' & y'' & z'' & 1 \end{bmatrix} = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.- Rotación sobre OX hasta que el eje esté contenido en el plano XZ:

$a = x_2 - x_1$ $b = y_2 - y_1$ $c = z_2 - z_1$
 $\frac{b}{d_1} = \sin \phi$ $\frac{c}{d_1} = \cos \phi$
 $d_1 = \sqrt{b^2 + c^2}$

$$\begin{bmatrix} x'' & y'' & z'' & 1 \end{bmatrix} = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d_1 & b/d_1 & 0 \\ 0 & -b/d_1 & c/d_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.- Rotación inversa del paso 3:

$$\begin{bmatrix} x''' & y''' & z''' & 1 \end{bmatrix} = \begin{bmatrix} x'' & y'' & z'' & 1 \end{bmatrix} \begin{bmatrix} d_1/d_2 & 0 & -a/d_2 & 0 \\ 0 & 1 & 0 & 0 \\ a/d_2 & 0 & d_1/d_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6.- Rotación inversa del paso 2:

$$\begin{bmatrix} x^{(4)} & y^{(4)} & z^{(4)} & 1 \end{bmatrix} = \begin{bmatrix} x''' & y''' & z''' & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d_1 & -b/d_1 & 0 \\ 0 & b/d_1 & c/d_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.- Rotación sobre YO hasta que el eje coincida con OZ (ángulo α):

$\frac{a}{d_2} = \sin \alpha$
 $\frac{d_1}{d_2} = \cos \alpha$
 $d_2 = \sqrt{a^2 + d_1^2} = \sqrt{a^2 + b^2 + c^2}$

$$\begin{bmatrix} x^{(4)} & y^{(4)} & z^{(4)} & 1 \end{bmatrix} = \begin{bmatrix} x''' & y''' & z''' & 1 \end{bmatrix} \begin{bmatrix} d_1/d_2 & 0 & a/d_2 & 0 \\ 0 & 1 & 0 & 0 \\ -a/d_2 & 0 & d_1/d_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7.- Traslación inversa del paso 1:

$$\begin{bmatrix} x^{(5)} & y^{(5)} & z^{(5)} & 1 \end{bmatrix} = \begin{bmatrix} x^{(4)} & y^{(4)} & z^{(4)} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_1 & y_1 & z_1 & 1 \end{bmatrix}$$

MATRIZ DE TRANSFORMACIÓN:

$$T = [Trasl(-x_1, -y_1, -z_1)] [Rot_{OX}(\phi)] [Rot_{OY}(-\alpha)] [Rot_{OZ}(\theta)] [Rot_{OX}(-\phi)] [Rot_{OZ}(\alpha)] [Trasl(x_1, y_1, z_1)]$$

Fig 35. Esquema con los distintos cálculos utilizados en la realización del programa alinearZ. Extraída de las diapositivas aportadas en clase.

Cabe destacar que utilizando la matriz de transformación que aparece en la esquina inferior derecha podríamos calcular la transformación de cualquier conjunto de átomos directamente, por lo que resultar ser una herramienta muy potente ya que simplifica mucho los cálculos.

Una vez explicado todo esto procedemos a desarrollar el programa y ha utilizarlo en nuestra proteína de interés, 6yla, dándonos el siguiente resultado:

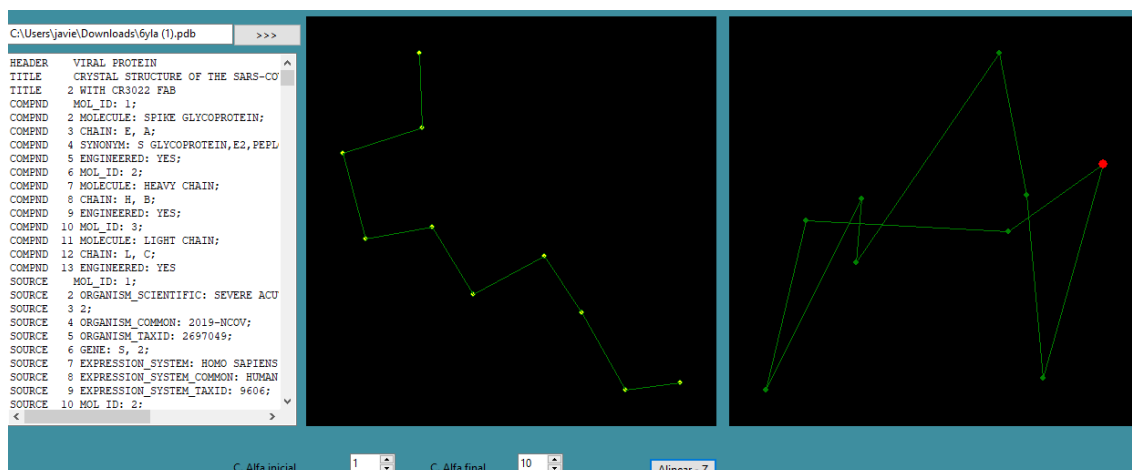


Fig 36. Podemos apreciar la interfaz de mi aplicación AlinearZ, donde podemos observar que a través de dos spinedits podemos elegir qué carbonos queremos tener en cuenta en el cálculo. El punto rojo en la imagen a la derecha del programa corresponde a los carbonos que han sido alineados.

Posteriormente he utilizado YASARA para comprobar si la representación era correcta.

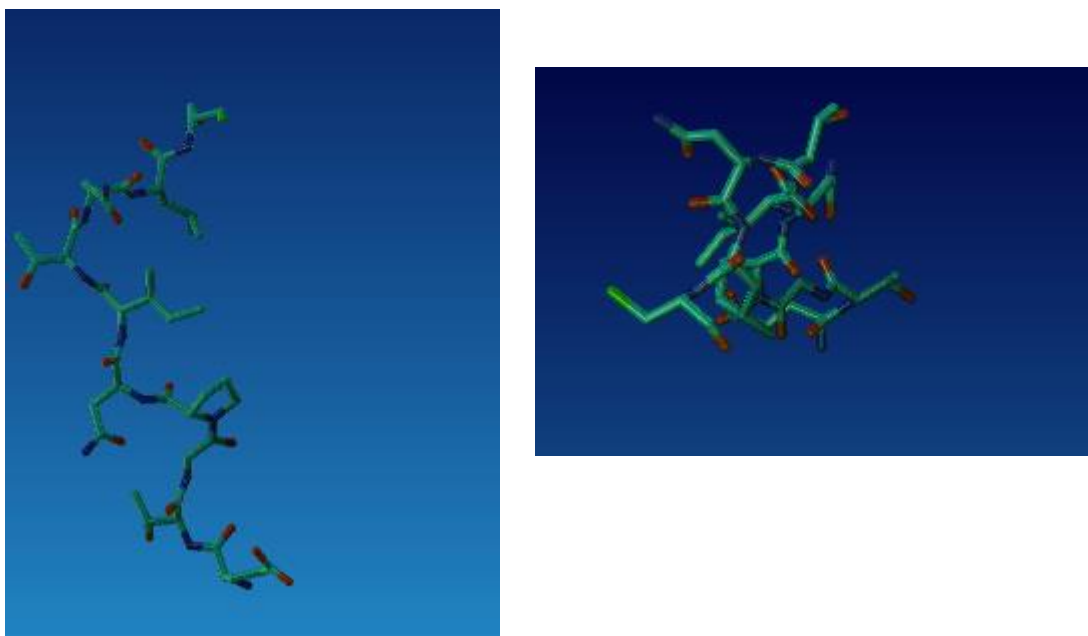


Fig 37. Imágenes obtenidas de YASARA. Las diferencias se deben a que el ángulo desde el que se mira la proteína no es el mismo, pero se puede apreciar que son los mismos carbonos, sobre todo en la primera imagen, y que el giro se ha realizado correctamente.

4.3.4 Cálculo de RMSD

La siguiente actividad consiste en desarrollar un programa que calcule el valor RMSD correspondiente a las tres primeras cisteínas de nuestra proteína asignada.

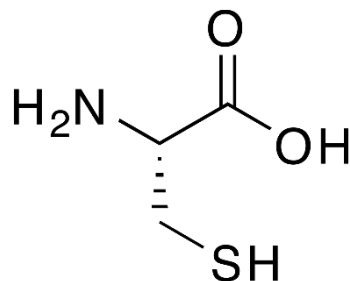


Fig 38. Estructura de la cisteína

El RMSD o desviación cuadrática media es capaz de medir las diferencias entre posiciones equivalentes con el objetivo de establecer cuánto se parecen o diferencian entre sí dos determinadas características. Por lo tanto, si calculamos el RMSD de nuestras cisteínas podremos calcular y deducir de un vistazo las divergencias entre ellas, lo que nos demuestra el potencial de esta técnica, y obtener una idea de su similitud desde un punto de vista geométrico.

Para hacer esto el programa debe escanear la proteína en busca de las tres primeras cisteínas, calcular las matrices de distancias internas entre todos los átomos que las componen y comparar dichas matrices entre sí empleando la siguiente fórmula:

$$RMSD = \sqrt{\frac{\sum_{i,j}^n (Dist1_{i-j} - Dist2_{i-j})^2}{n}}$$

Donde $Dist1_{i-j}$ es la distancia entre los elementos i y j en el primer residuo que se está empleando en la comparación, $Dist2_{i-j}$ es la distancia entre los mismos elementos del segundo residuo, y n es el número total de elementos que se tienen en cuenta en la comparación. Yo he decidido hacer un nuevo type denominado TMatrizRMSD, que son unas matrices de datos 6x6, donde he ido metiendo las distancias entre todos los átomos de una molécula de cisteína, para luego poder calcular el valor RMSD más fácilmente. Es interesante recalcar que la diagonal de dicha matriz debe ser siempre 0 ya que la distancia de un átomo consigo mismo no puede variar, por lo que en caso de que esto no sea así nos está indicando que algo va mal.

A continuación, he añadido un pantallazo de mi aplicación CalcularRMSD, dónde se pueden observar los valores de RMSD para nuestras cisteínas:

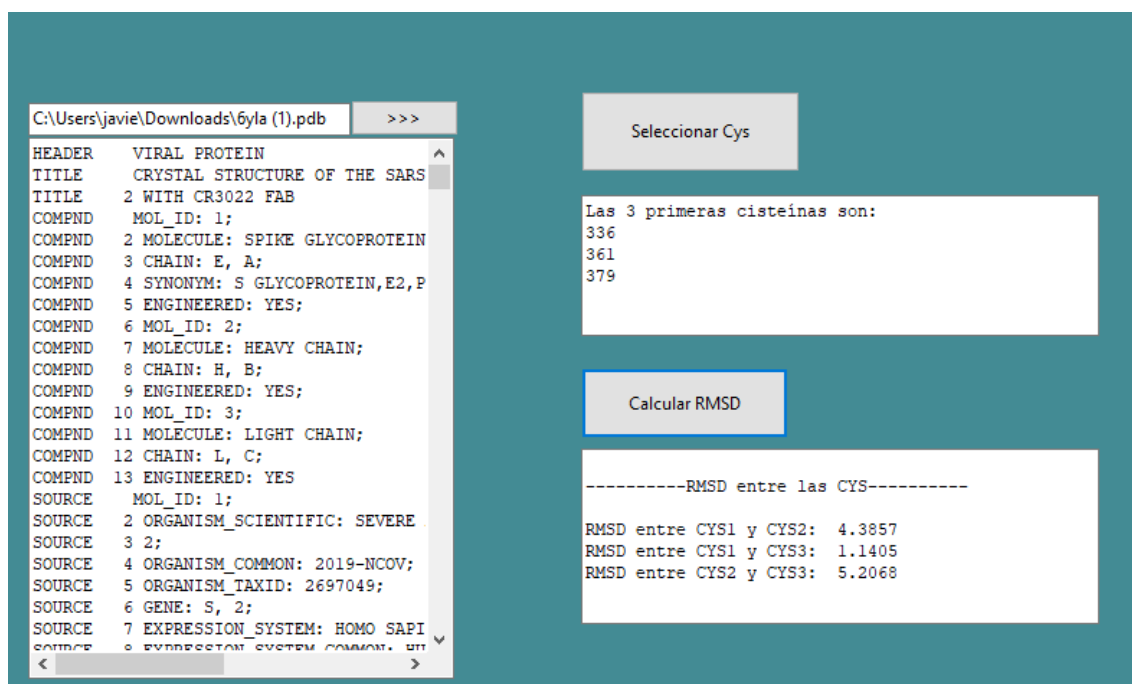


Fig 39. Interfaz de mi aplicación CalculoRMSD, donde podemos observar un primer botón que nos permite cargar el fichero, un segundo botón que nos selecciona las 3 primeras cisteínas del fichero y un tercer botón con el que hacemos el cálculo de RMSD. En la imagen el programa ha sido utilizado con mi fichero 6yla.

```

var
k,n,j:integer;
begin
n:=0;
j:=0;
memo2.lines.add('Las 3 primeras cisteinas son: ');
memo3.lines.add('');
for k:=0 to p.NumResiduos-1 do
begin
if (p.res[k].ID3='CYS') and (n<>3) then
begin
n:=n+1;
if n=1 then
begin
cys1:=p.res[k].NumRes; //Nos va a servir para almacenar la posición de las cisteinas
memo2.lines.add(inttostr(p.res[k].NumRes));
end;
if n=2 then
begin
cys2:=p.res[k].NumRes; //Nos va a servir para almacenar la posición de las cisteinas
memo2.lines.add(inttostr(p.res[k].NumRes));
end;
if n=3 then
begin
cys3:=p.res[k].NumRes; //Nos va a servir para almacenar la posición de las cisteinas
memo2.lines.add(inttostr(p.res[k].NumRes));
end;
end;
end;
end;
end;

```

Fig 40. Código utilizado por el button2 de la aplicación con el que buscamos y guardamos los valores de las tres primeras cisteínas del fichero.

El código referente al button3 es más largo y lo dejo en la unit1, pero se basa en bucles para ir rellenando las matrices y posteriormente calcular el RMSD con la ecuación anterior. Según la fig 39, la primera y la tercera cisteínas son mucho más parecidas entre sí que la segunda con cualquiera de las otras dos, debido a que sus valores RMSD son mucho más parecidos entre sí, lo que nos informa de que sus entornos deben ser parecidos.

4.3.5 Mutagénesis dirigida: creación de una mutación Phe-Tyr

En esta actividad nos han pedido rediseñar la proteína asignada, reemplazando una fenilalanina cualquiera por una tirosina. Para efectuar esta mutación dirigida se han tenido en cuenta las características estructurales de estos aminoácidos. Una fenilalanina se diferencia de una tirosina en que la tirosina posee un grupo -OH en la posición para- del anillo aromático, por lo que para realizar esta sustitución lo único que debe hacerse es añadir este grupo funcional a la fenilalanina correspondiente.

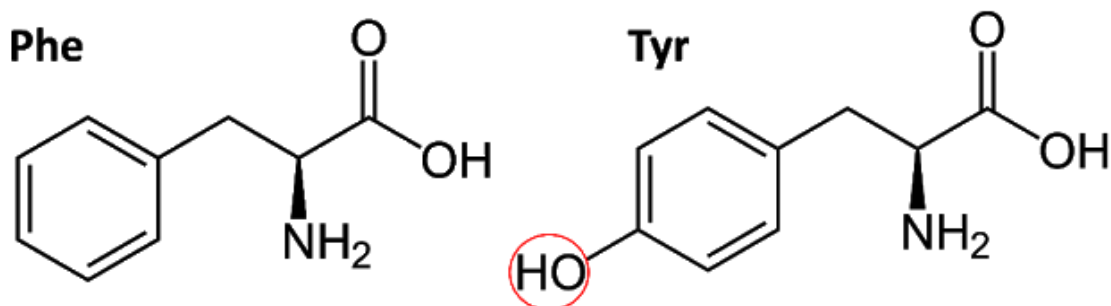


Fig 41. Fenilalanina a la izquierda y tirosina a la derecha. El grupo hidroxilo de la tirosina se ha resaltado con una circunferencia de color rojo. Se puede apreciar cómo la única diferencia en su secuencia es el OH.

Como los átomos de hidrógeno no aparecen normalmente en los ficheros *.pdb hemos considerado al grupo -OH como una única esfera de diámetro superior a un átomo de oxígeno, pero inferior que el espacio ocupado por un grupo -OH. Para saber dónde colocar este nuevo grupo funcional, primero calculamos la distancia media entre los grupos -OH de algunas tirosinas de las proteínas y el carbono ζ correspondiente, que es el átomo de carbono del anillo aromático al que se enlaza el grupo -OH. Una vez que se conoce la distancia a la que se encuentra el grupo -OH es necesario saber en qué posición ubicarlo. Para ello, se ha considerado que su posición se encontrará en el mismo plano que el plano del anillo aromático.

He seleccionado tres tirosinas al azar de mi fichero para a partir de ellos obtener el módulo promedio, lo que nos permite calcular el vector unitario a partir de nuestra fenilalanina. Posteriormente, he obtenido el vector de \vec{v} de interés. Lo que nos permite conocer las coordenadas del grupo OH introducido.

$$\|m\| = \vec{O} - \vec{CZ}$$

$$\vec{D} = \vec{CZ} - \vec{CB}$$

$$\vec{U} = \frac{D}{\|D\|}$$

$$\vec{V} = \vec{U} * \|m\|$$

$$\vec{O} = \vec{CZ} + \vec{v}$$

Todos los cálculos los he simplificado en un Excel y el resultado ha sido el siguiente:

	TYR ref.	Coor X	Coor Y	Coor z	Módulos
TYR1(E380)	Cz	2.481	37.104	26.082	
	O	3.215	37.044	24.921	1.37487345
TYR 2(H198)	Cz	42.328	30.672	73.951	
	O	41.578	30.417	75.047	1.35230951
TYR 3(C192)	Cz	51.882	5.960	9.714	
	O	51.151	6.555	8.712	1.37564167
Media =					1.36760821

Fenilalanina	Mutación			
PHE F (B170)	TYR F (170)	Coor X	Coor Y	Coor Z
	CZ (v)	33.781	10.602	17.296
	CB (v)	35.362	11.950	21.034
	D (v)	-1.581	-1.348	-3.738
	V unitario	-0.3696862	-0.3152037	-0.8740588
	V (v)	-0.5055859	-0.4310751	-1.19537
	O (v)	33.275	10.171	16.101

ModuloD = 4.27660017

Yo he realizado la mutación en la PHE B170, introduciendo los valores del O que he calculado en Excel. A continuación, he metido la mutación en mi proteína asignada y he procedido a introducirla en YASARA para visualizarla:

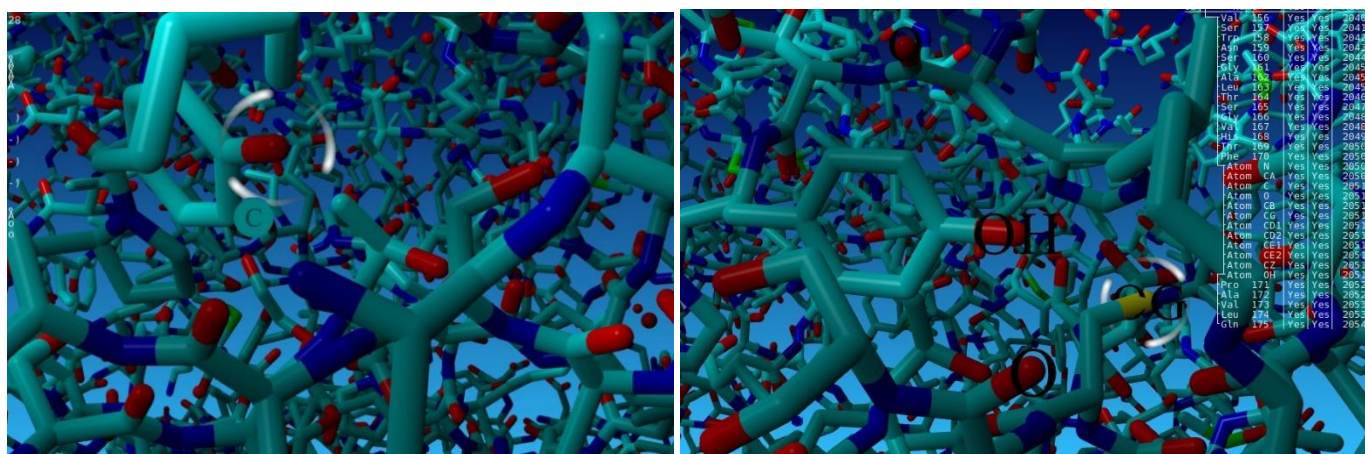


Fig 41. En las imágenes superiores podemos observar a mi proteína mutada, siendo el OH introducido el que aparece marcado (circulo blanco).

La fenilalanina es un aminoácido apolar, mientras que la tirosina es polar como consecuencia de tener un grupo hidroxilo. La fenilalanina mutada está en el interior de la proteína, por lo que esta mutación puede provocar cambios en la conformación de la

proteína, dado que el residuo de tirosina polar tendería a ubicarse en la superficie de la proteína en contacto con el disolvente acuoso. Como podemos observar, mi proteína se encuentra rodeada por otros átomos como O y C, lo que podría provocar algún impedimento de tipo estérico también, aunque también podría generar enlaces de hidrógeno con estos mismos átomos que estabilizase la estructura. Todo depende de la distancia entre estos y del espacio que haya entre mi TYR mutada y otras moléculas cercanas.

A pesar de esto, esta predicción computacional no nos puede garantizar que esto ocurra, tendría que llevarse a cabo esta mutagénesis dirigida en un laboratorio y experimentos que permitieran corroborarlo.

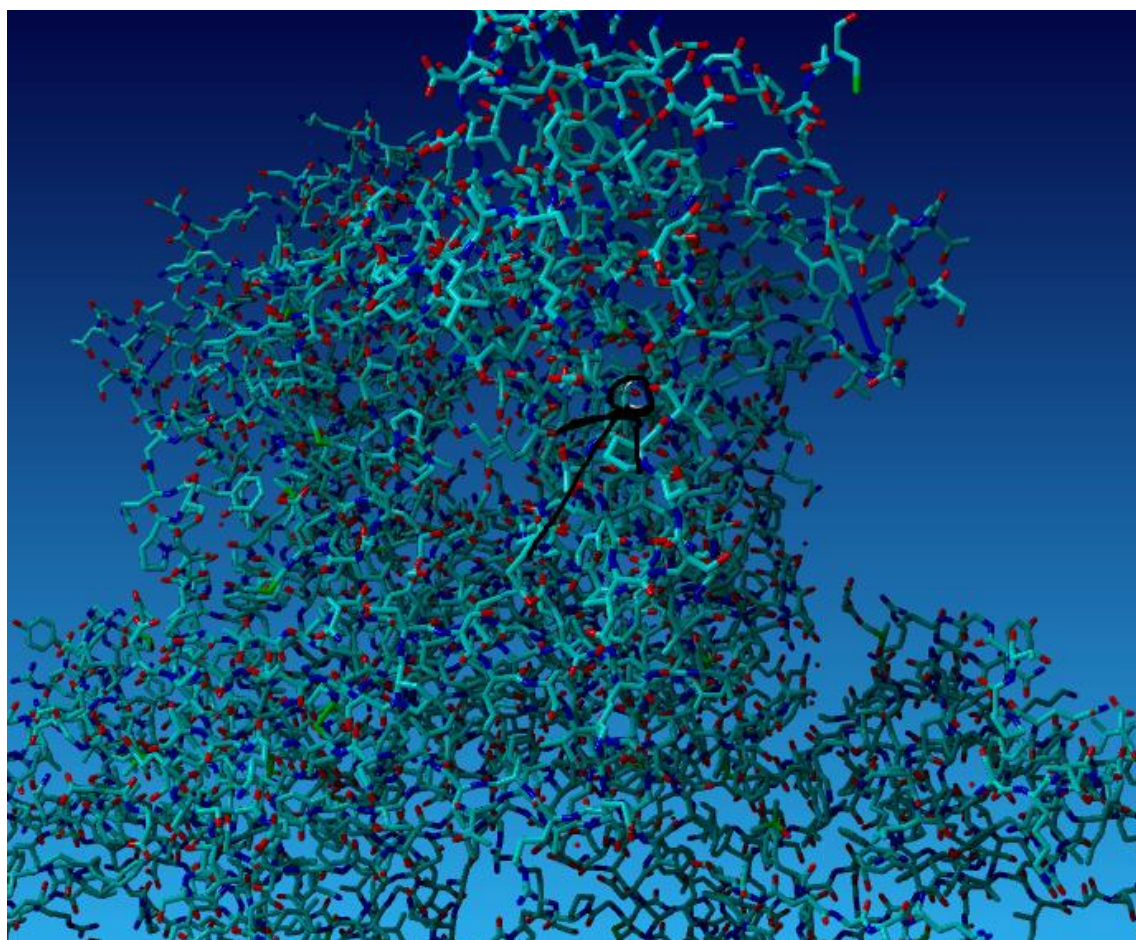


Fig 42. Representación de la mutación en el conjunto de la proteína. La flecha indica nuestro -OH que convierte la PHE en una TYR.

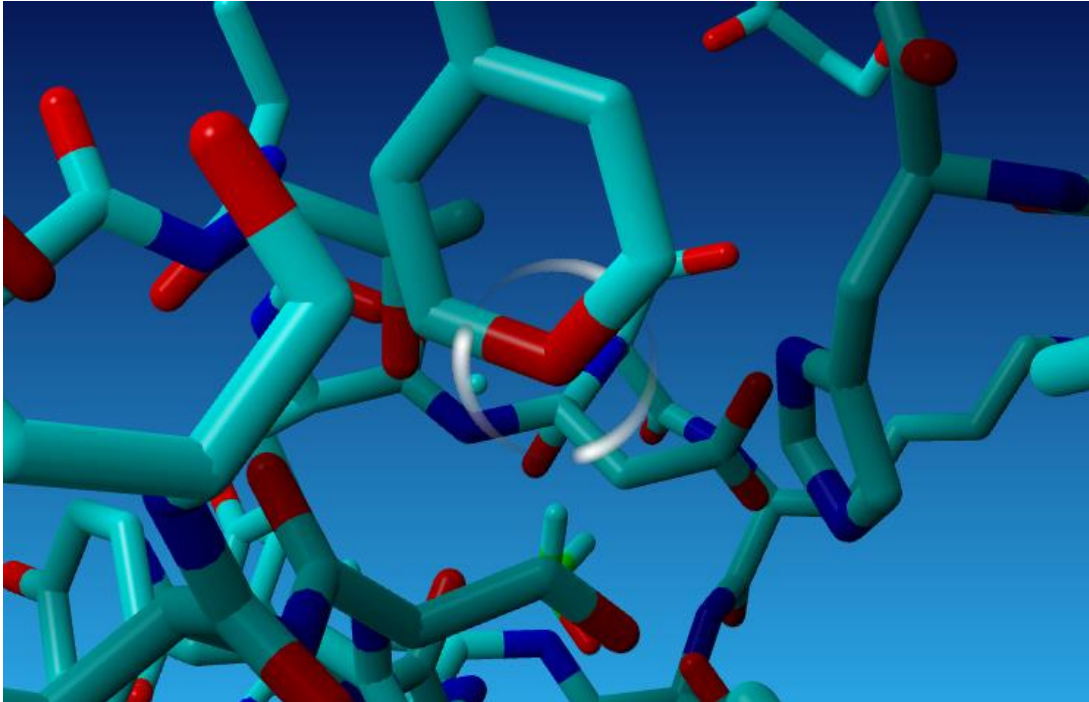


Fig 43. Aquí tenemos representada nuestra proteína sin la mutación, por lo tanto, es nuestra PHE silvestre.

4.3.6 Creación de un programa de predicción de enlaces disulfuro

En esta actividad se nos ha pedido que desarrollemos una aplicación informática que sea capaz de predecir los enlaces disulfuro de una proteína. Para hacer esto, debíamos desarrollar un programa que escanease la proteína asignada en busca de cisteína, calcular las distancias entre los átomos de azufre de todas ellas y determinar qué pares de cisteínas son susceptibles de formar puentes disulfuro.

La aplicación que hemos desarrollado se llama enlaces_SS. Por defecto, el programa selecciona una distancia umbral de 1.8 Armstrong para determinar si se puede formar un puente disulfuro o no. Yo he seleccionado una distancia umbral de 2.1 Armstrong para realizar el calculo y he obtenido lo siguiente:

The application interface displays the following data:

Left Pane (PDB File: 6yla (1).pdb):

Residue 1	Residue 2	Distance (Å)
1 AD5 4 ALA C 159	LEU C 160	
2 AD5 4 LYS C 151	VAL C 156	
3 AD5 4 VAL C 197	THR C 203	
4 AD5 4 VAL C 211	ASN C 216	
1 CYS E 336	CYS E 361	
2 CYS E 379	CYS E 432	
3 CYS E 391	CYS E 525	
4 CYS E 480	CYS E 488	
5 CYS H 22	CYS H 96	
6 CYS H 144	CYS H 200	
7 CYS L 23	CYS L 94	
8 CYS L 140	CYS L 200	
9 CYS A 336	CYS A 361	
10 CYS A 379	CYS A 432	
11 CYS A 391	CYS A 525	
12 CYS A 480	CYS A 488	
13 CYS B 22	CYS B 96	
14 CYS B 144	CYS B 200	
15 CYS C 23	CYS C 94	
16 CYS C 140	CYS C 200	

Right Pane (Predicted Disulfide Bonds):

Cisteína-1	Cisteína-2	Distancia
CYS336E	CYS361E	2.03
CYS379E	CYS432E	2.03
CYS391E	CYS525E	2.03
CYS480E	CYS488E	2.03
CYS22H	CYS96H	2.03
CYS144H	CYS200H	2.03
CYS23L	CYS94L	2.03
CYS140L	CYS200L	2.03
CYS336A	CYS361A	2.03
CYS379A	CYS432A	2.03
CYS391A	CYS525A	2.03
CYS480A	CYS488A	2.03
CYS22B	CYS96B	2.03
CYS144B	CYS200B	2.03
CYS23C	CYS94C	2.03
CYS140C	CYS200C	2.03

Bottom Controls:

- Umbral: 2.10
- Calcular

Fig 44. Interfaz de mi aplicación enlacesSS. El fichero cargado y los resultados corresponden a mi fichero 6yla.

Podemos observar que nuestra proteína asignada presentaría una gran cantidad de enlaces disulfuro, lo que tendría sentido si tenemos en cuenta que este tipo de enlace estabiliza la forma plegada de una proteína y la nuestra es un epítipo conformacional que estabiliza la proteína spike prefusión, por lo que es importante que sea muy estable. Observando los datos del PDB que aparecen en el memo de la izquierda podemos ver que los enlaces calculados por nuestro programa coinciden con los que nos daban en un principio, por lo que podríamos deducir que nuestro programa es bastante fiable.

Esta herramienta predictiva va a funcionar solo en el caso de que se utilice un valor de umbral razonable, como 2,05 Å o 2,10 Å. Podemos seleccionar un valor de umbral absurdamente grande, por ejemplo, 10 Å y posiblemente nos aparezca algún enlace calculado por el programa, pero no es posible que dos átomos a una distancia de 10 Å estén enlazados covalentemente.

De todas formas, he realizado un muestreo con otros ficheros para ver si salían bien los enlaces, y lo hacen, por lo que podemos dar por correctos los resultados de nuestra aplicación:

The figure displays two screenshots of a software application used for predicting disulfide bonds. Both screenshots show a list of disulfide bonds with columns for Cysteine-1, Cysteine-2, and Distance.

Left Screenshot (CD curso 2020-2):

Cysteine-1	Cysteine-2	Distance
CYS 336	CYS 361	2.03
CYS 379	CYS 432	2.04
CYS 391	CYS 525	2.03
CYS 480	CYS 488	2.03
CYS L 23	CYS L 88	2.04
CYS L 136	CYS L 196	2.03
CYS H 22	CYS H 92	2.03
CYS H 141	CYS H 197	2.03
CYS Z 336	CYS Z 361	2.03
CYS Z 379	CYS Z 432	2.04
CYS Z 391	CYS Z 525	2.03
CYS Z 480	CYS Z 488	2.03
CYS Y 23	CYS Y 88	2.03
CYS Y 136	CYS Y 196	2.03
CYS X 22	CYS X 92	2.03
CYS X 141	CYS X 197	2.03

Right Screenshot (6xc2.pdb):

Cysteine-1	Cysteine-2	Distance
CYS 336A	CYS 361A	2.03
CYS 379A	CYS 432A	2.04
CYS 391A	CYS 525A	2.03
CYS 480A	CYS 488A	2.03
CYS 23L	CYS 88L	2.04
CYS 136L	CYS 196L	2.03
CYS 22H	CYS 92H	2.03
CYS 141H	CYS 197H	2.03
CYS 336Z	CYS 361Z	2.03
CYS 379Z	CYS 432Z	2.04
CYS 391Z	CYS 525Z	2.03
CYS 480Z	CYS 488Z	2.03
CYS 23Y	CYS 88Y	2.03
CYS 136Y	CYS 196Y	2.03
CYS 22X	CYS 92X	2.03
CYS 141X	CYS 197X	2.03

Fig 45. Podemos observar cómo el programa calcula correctamente el número de enlaces disulfuro en otros ficheros utilizados a lo largo del curso.

4.3.7 Perfil hidrofóbico y anfipatía axial

Para esta última actividad se nos pidió que creásemos una aplicación que permita calcular la anfipatía de hidrofobicidad de un segmento de secuencia peptídica mediante el cálculo de momentos de Eisenberg y mediante el espectro de potencias de Fourier de Stroud. Esta aplicación es muy interesante ya que la hidrofobicidad es un factor esencial cuando hablamos de la estabilidad de las proteínas, siendo el factor que equilibra la balanza hacia la conformación proteica.

En nuestra aplicación se generan 3 gráficas, una de hidrofobicidad y dos de anfipatía o lateralización de la hidrofobicidad:

- Perfil hidrofóbico según el formalismo de Kyte-Doolittle: representa valor de hidrofobicidad frente a número de residuo, pero empleando una ventana para reducir el ruido de la señal de hidrofobicidad.
- Perfil anfipático según Eisenberg: procede de la noción física de momento hidrofóbico, que son vectores perpendiculares a la estructura cuya magnitud hace referencia a la hidrofobicidad de los residuos. Se calcula con la siguiente fórmula:

$$\mu_H = \sqrt{\left\{ \left[\sum_n H_n \sin(\delta n) \right]^2 + \left[\sum_n H_n \cos(\delta n) \right]^2 \right\}}$$

Donde μ_H es el momento hidrofóbico; H_n , la hidrofobicidad residual; δ , la distancia angular; y n el número de residuo.

- Perfil anfipático según Stroud: procede de un fundamento matemático basado en transformadas de Fourier, que permite descomponer cualquier grafo en senos y cosenos.

Representamos el Power Spectrum, o la amplitud frente a la frecuencia, a partir del cual se podrá extraer información secundaria ya que cada tipo tendría un ritmo distinto. Por poner un ejemplo, una alfa-hélice tiene un pico en torno a 100 grados o una lámina beta tiene picos en torno a 0 y 100 grados. Se calcula con la siguiente fórmula:

$$I(k, v) = \left[\sum_{j=k-n}^{k+n} [h_j - \bar{h}(k)] \sin(jv) \right]^2 + \left[\sum_{j=k-n}^{k+n} [h_j - \bar{h}(k)] \cos(jv) \right]^2$$

(Donde I es el espectro de potencias, h_j la hidrofobicidad residual, (k) , la media de la hidrofobicidad de los residuos del segmento, j , el número de residuo y v la distancia angular)

Con todo esto en mente, pasamos a desarrollar la aplicación perfiles, la cual nos calcula tanto los perfiles hidrofóbicos como los anfipáticos. Aplicándolo sobre nuestra proteína obtenemos:

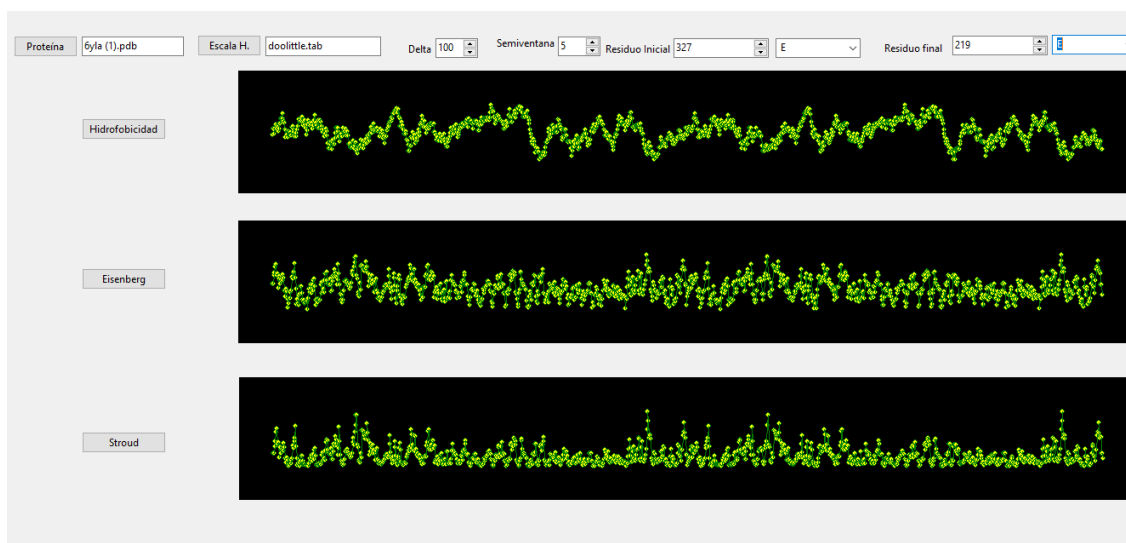


Fig 45. Podemos observar nuestra aplicación, que nos permite elegir el fichero y la escala H. con la que queremos hacer los cálculos, a la vez que la semiventana y los residuos inicial y final de las subunidades a tratar. El fichero utilizado en la imagen es el 6yla.

Los datos obtenidos en las gráficas son similares, lo que nos confirma que nuestro programa funciona bien. La forma del gráfico del momento de Eisenberg y el espectro de potencias de Fourier de Stroud es bastante similar, esto nos demuestra que ambas aproximaciones para calcular la anfipatía axial son equivalentes. Hay que tener en cuenta que los valores dependen mucho de la semiventana que utilicemos, por lo que hay que tener precaución con los resultados que obtengamos.

Bibliografía

- “Biología del SARC-COV-2”- Dheni Aidé Fernández-Camargo, Luis Eduardo Morales-Buenrostro, Área de Nefrología del Trasplante, Departamento de Nefrología y Metabolismo Mineral, Instituto Nacional de Ciencias Médicas y Nutrición «Salvador Zubirán». Ciudad de México, México.
- Yuan M., Wu N.C., Zhu X., Lee C.D., So R.T.Y., Lv H., Mok C.K.P., Wilson I.A. A highly conserved cryptic epitope in the receptor binding domains of SARS-CoV-2 and SARS-CoV. *Science*. 2020;**368**:630–633.
- ter Meulen J., van den Brink E.N., Poon L.L.M., Marissen W.E., Leung C.S.W., Cox F., Cheung C.Y., Bakker A.Q., Bogaards J.A., van Deventer E. Human monoclonal antibody combination against SARS coronavirus: synergy and coverage of escape mutants. *PLoS Med*. 2006;**3**:e237.
- Perlman S., Dandekar A.A. Immunopathogenesis of coronavirus infections: implications for SARS. *Nat. Rev. Immunol*. 2005;**5**:917–927.
- Rattanapisit, K., Shanmugaraj, B., Manopwisedjaroen, S. *et al*. Rapid production of SARS-CoV-2 receptor binding domain (RBD) and spike specific monoclonal antibody CR3022 in *Nicotiana benthamiana*. *Sci Rep* **10**, 17698 (2020).
- Pastrian-Soto, Gabriel. (2020). Bases Genéticas y Moleculares del COVID-19 (SARS-CoV-2). Mecanismos de Patogénesis y de Respuesta Inmune. *International journal of odontostomatology*, 14(3), 331-337.
- Fernández-Camargo DA, Morales-Buenrostro LE. Biología del SARS-CoV-2. *Rev Mex Traspl*. 2020;9(Suppl: 2):139-148. doi:10.35366/94503.
- “Neutralization of SARS-CoV-2 by Destruction of the Prefusion Spike-
“<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7303615/>