

Análisis de la estructura cristalina del súper complejo de proteínas no estructurales del SARS-CoV: el hexadecámero 2AHM

Ana Robles Fernández
Cuaderno Personal de Actividades (CPA)

Grado en Biotecnología
Asignatura de Ingeniería de Proteínas

Curso académico 2020/2021



UNIVERSIDAD DE GRANADA

ÍNDICE

BLOQUE I. General. Revisión bibliográfica.

- 1.1.** Revisión bibliográfica de la proteína asignada: el hexadecámero.
- 1.2.** Análisis mediante visor de proteínas 3D de elementos tratados en el bloque I. IActividad 4 incluida a lo largo de todo el apartado 1.

BLOQUE II. Bases de datos secuenciales y estructurales.

- 2.1. Actividad 2.** Diagrama de flujo de decisión y aplicación que permita la discriminación del tipo de fichero y obtención de secuencia.
- 2.2. Actividad 3.** Función de Parsing de ficheros: CargarPDB.

BLOQUE III. Visualización de proteínas.

- 3.2. Actividad 5.** WritePDB.

BLOQUE IV. Predicción estructural y rediseño de proteínas.

- 4.1. Actividad 6.** Cálculo de distancias y diagrama de Ramachandran.
- 4.2. Actividad 7.** Desarrollo de un Esterodiagrama.
- 4.3. Actividad 8.** Alineación de carbonos sobre el eje Z.
- 4.4. Actividad 9.** Cálculo del RMSD a las 3 primeras cisteínas de la proteína asignada.
- 4.5. Actividad 10.** Rediseño estructural de la proteína. El mutante Muto1.
- 4.6. Actividad 11.** Predicción y asignación de enlaces disulfuro.
- 4.7. Actividad 12.** Cálculo de la hidrofobicidad de un segmento. Anfipatía.
- 4.8. Actividad complementaria.**

Bibliografía y referencias.

BLOQUE I. General. Revisión bibliográfica

1.1. Generalidades

Orthocoronavirinae, conocido comúnmente como coronavirus, es una subfamilia de virus de ARN monocatenario positivos y policistrónico perteneciente a la familia Coronaviridae, la cual se subdivide genotípica y serológicamente en los géneros Alfacoronavirus, Betacoronavirus, Gammacoronavirus. Deltacoronavirus, que pertenecen al orden de los Nidovirales (usan un conjunto anidado de ARNm para su replicación), un grupo diverso de virus filogenéticamente similares cuya ascendencia común se dedujo a partir de los principios comunes que subyacen a la organización y expresión de su genoma, y de la conservación de un conjunto de dominios de replicasas centrales, incluidas las enzimas clave de síntesis de ARN, una nucleocápside de simetría helicoidal, con envoltura, cuyos viriones pueden medir entre aproximadamente 50 y 200 nm de diámetro.

Su material genético es el de mayor tamaño dentro de los virus de ARN conocidos hasta la fecha, con genomas que van desde los 26 a 32 kilobases, y se cree que su expansión fue facilitada por la adquisición de nuevas funcionalidades enzimáticas que contrarrestaron la alta frecuencia de errores de las ARN polimerasas virales. Se los llama coronavirus por la corona de puntas que se ve alrededor de la superficie del virus.

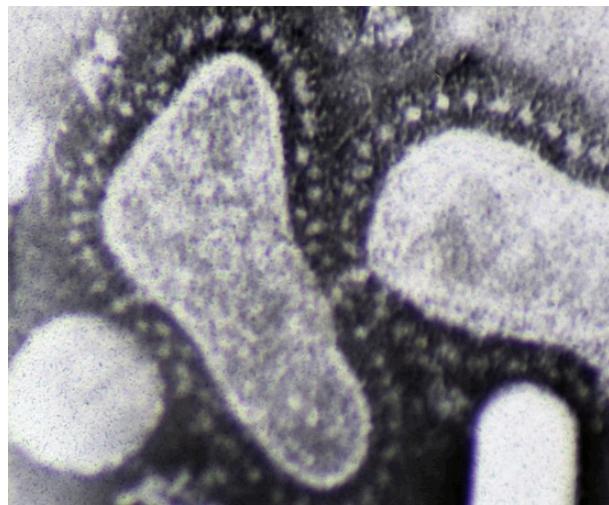
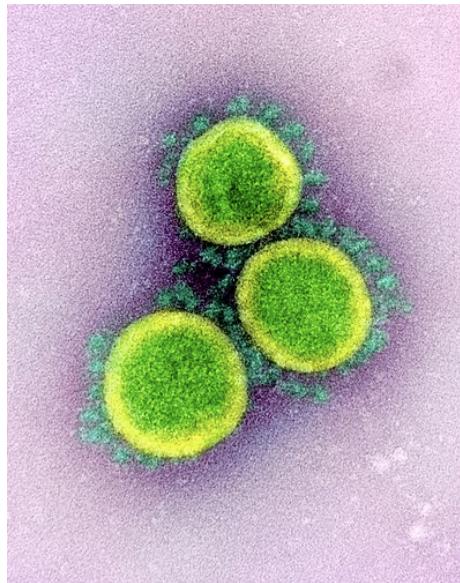


Figura 2. Micrografía electrónica de transmisión de partículas del virus SARS-CoV-2, aisladas de un paciente. Imagen capturada y coloreada en la Instalación de Investigación Integrada (IRF) del NIAID en Fort Detrick, Maryland. Crédito: NIAID.

Figura 3. Micrografía electrónica de dos coronavirus. Autor: [Dr Graham Beards](#)

1.2. Origen, evolución y transmisión de los coronavirus.

La familia de los coronavirus representa patógenos humanos y animales que pueden causar infecciones zoonóticas letales, tal como la que azota en nuestros días: el SARS-CoV2. Usualmente se trata de enfermedades respiratorias y digestivas, como SARS y MERS. En concreto, los alfacoronavirus y los betacoronavirus pueden suponer una alta carga viral de enfermedad para el ganado; estos virus incluyen el virus de la gastroenteritis transmisible porcina, el virus de la diarrea entérica porcina (PEDV) y el recientemente aparecido coronavirus del síndrome de la diarrea aguda porcina (SADS-CoV). Según las bases de datos de secuencias actuales, todos los coronavirus humanos tienen un origen animal: El SARS-CoV, el MERS-CoV, el HCoV-NL63 y el HCoV-229E se consideran originarios de los murciélagos; el HCoV-OC43 y el HKU1 probablemente se originaron en los roedores. Estas infecciones no eran consideradas muy patogénicas en humanos hasta que tuvo lugar el brote del síndrome respiratorio agudo severo (severe acute respiratory syndrome, SARS) en 2002 y 2003 en la provincia de Guandong, en China, debido a que antes de este hecho los coronavirus que circulaban entre la población causaban infecciones suaves en aquellas personas con trastornos de inmunodeficiencia. Diez años después de este brote de SARS, otro coronavirus altamente patógeno, el síndrome respiratorio de coronavirus de Medio Oriente (MERS-CoV) emergió en los países de Medio Oriente.

Los SARS coronavirus usan a la **enzima conversora de angiotensina 2 (ACE2)** como receptor, e infecta primariamente a las células epiteliales ciliadas de los bronquios y a los pneumocitos de tipo II, mientras que el MERS-CoV usa a la dipeptidil-peptidasa 4 (DPP4 o CD26) como receptor, e infecta al epitelio no ciliado de los bronquios, y de igual manera que el anterior, a los pneumocitos de tipo II.

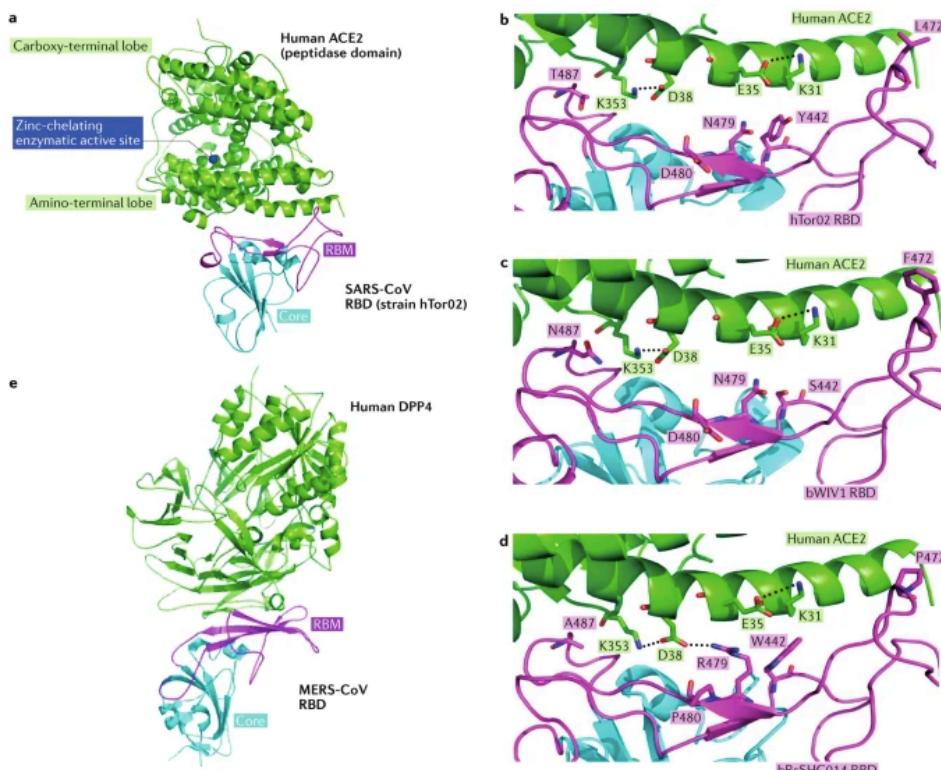


Figura 4. Reconocimiento por receptores por el SARS-CoV y MERS-CoV, indicándose aquellos aminoácidos que se encuentran implicados en la misma.

Se cree que SARS-CoV y MERS-CoV se transmitieron directamente a los seres humanos desde civetas presentes en mercados y camellos dromedarios, respectivamente. Además, ambos virus se originaron en murciélagos, pasando posteriormente por infecciones zoonóticas a otros mamíferos.

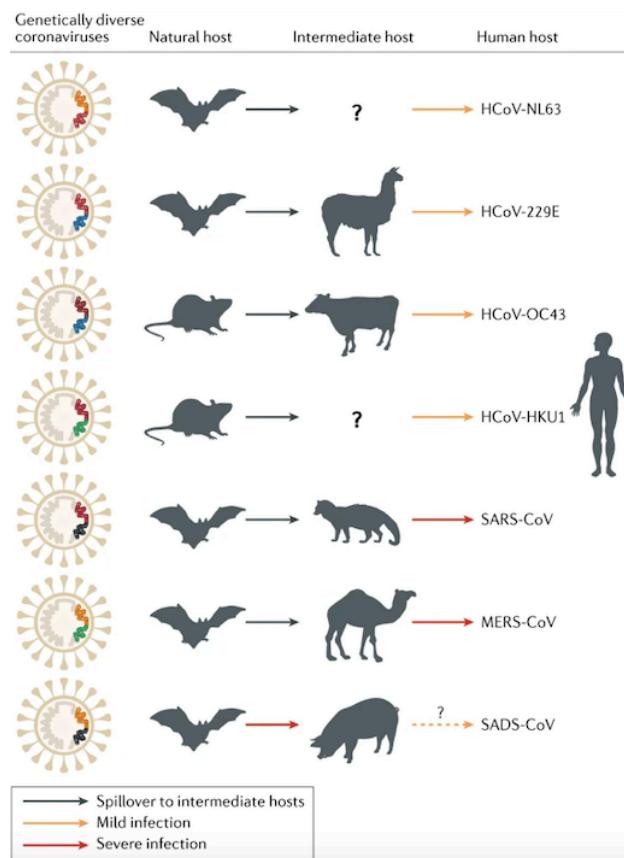


Figura 5. Esquema sobre las relaciones de transmisión zoonótica de los coronavirus. Las flechas grises implican contagio hacia hospedadores intermedios. Las flechas amarillas indican infecciones leves, y las flechas rojas, infecciones de mayor severidad.

1.3. Análisis filogenético de los coronavirus.

La secuenciación de genomas de SARS-CoV-2 en personas infectadas por todo el mundo ha permitido el desarrollo de un árbol filogenético de sus diferentes cepas, además de las relaciones de homología estructurales/secuenciales que presenta con otros virus de la familia de los Coronaviridae extraídos de numerosos repositorios.

Los virus analizados que se distribuían en el momento del estudio a lo largo del globo presentan una alta identidad de secuencias, con un 99.98% de similitud. Además, los mismos se compararon con los coronavirus más cercanos presentes en otras especies, como el del pangolín o el murciélagos, con los cuales presentan un 92% y un 98% de homología, respectivamente.

Tras las primeras infecciones, el virus se dividió en dos macro-haplogrupos, el A que constituye aquel de afectación mundial, y el B, con afectación principalmente en Asia, y se ha demostrado que la cepa original es la A. Estos macro-haplogrupos a su vez se dividen en sub-haplogrupos, y todos ellos presentan mutaciones puntuales en posiciones características que los diferencian del resto. De esta forma, el virus muta a un ritmo de aproximadamente 1-2 mutaciones por mes, y cuando acumula más de 2 mutaciones no existentes en la cepa original, las cepas pasan a considerarse novedosas.

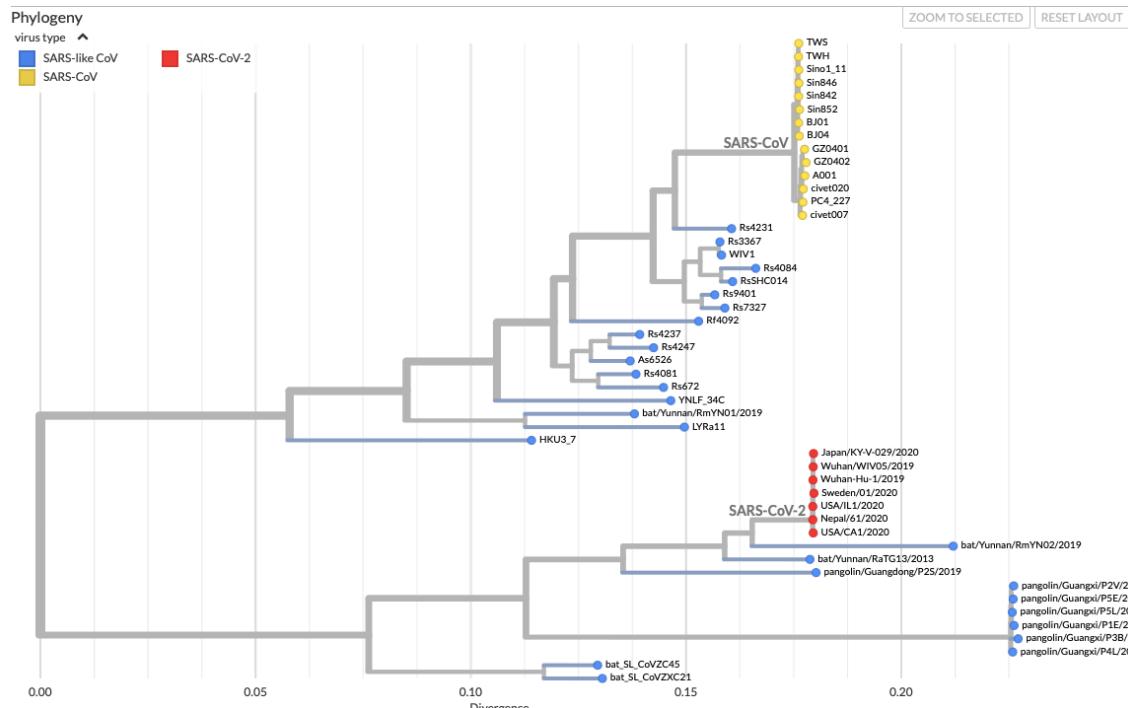


Figura 1. Árbol filogenético rectangular que representa las relaciones existentes entre los virus SARS-like CoV, los SARS-CoV y SARS-CoV2 [13].

A su vez, existen mutaciones más ventajosas que otras, las cuales van a determinar la efectividad en factores como la mayor transmisibilidad de la cepa. Por ejemplo, se cree que la mutación de transversión A23403T, la cual genera un cambio de aminoácido del aspártico a una glicina, en la posición 614 de la proteína Spike (S), puede aportar una ventaja en la transmisibilidad, debido a que puede alterar la interacción entre los dominios de la proteína Spike. Esto explicaría la rápida propagación por Europa de dicha cepa, además de la selección por tanto de sub-haplogrupos más virulentos.

A día de hoy, podemos hablar de las siguientes variantes o cepas, aunque en un futuro pueda haber muchas más:

- **Cluster 5**, originada en Dinamarca.
- **501.V2**, originada en Sudáfrica.
- **B.1.207**, originada en Brasil.
- **VUI-2020/2021**, procedente de Reino Unido.
- **D614G**, con origen en Malasia.

Estas nuevas cepas actualmente están siendo objeto de estudio, para la evaluación principalmente de aquellas mutaciones que puedan afectar a variantes de patogenicidad, debido a que cambios relevantes podrían implicar la falta de efectividad de las vacunas desarrolladas hasta la fecha.

1.4. El SARS-CoV2

Como he mencionado anteriormente, en el año 2019 hubo un nuevo brote de CoVID, pasando de su aparición y expansión en la República Popular de China como una infección respiratoria grave, a causar una pandemia de magnitudes mundiales que continua hasta nuestros días.

Teniendo en cuenta las relaciones filogenéticas y estructuras genómicas que posee, SARS-CoV-2 pertenece al género de los Betacoronavirus.

El SARS-CoV-2 presenta una similitud genética del 97% con el SARS-CoV y un 50% con el MERS-CoV, y aunque presente una tasa de transmisión mayor que estos dos últimos, su tasa de mortalidad es menor (de un 3.4%), lo que hace que tenga una menor patogenicidad.

En cuanto a su estructura, podemos destacar gracias a las imágenes obtenidas por microscopía electrónica de transmisión la presencia de una corona solar que, como he mencionado anteriormente, es la que le da el nombre de “coronavirus”. Esta partícula vírica tiene una morfología esférica de un diámetro de entre unos 60 y 140 nanómetros, junto con espigas o “spikes” de unos 8 a 12 nanómetros de longitud. El virión consiste básicamente en una nucelocápside que se encarga de proteger al material genético que contiene, y una envoltura externa.

En la envoltura externa, tenemos proteínas estructurales principales denominadas como proteínas Spike (S), la proteína de membrana (M) y la proteína de envoltura (E), además de algunas otras proteínas accesorias como la hemaglutinina esterasa (HE), proteína 3, proteína 7a, entre otras.

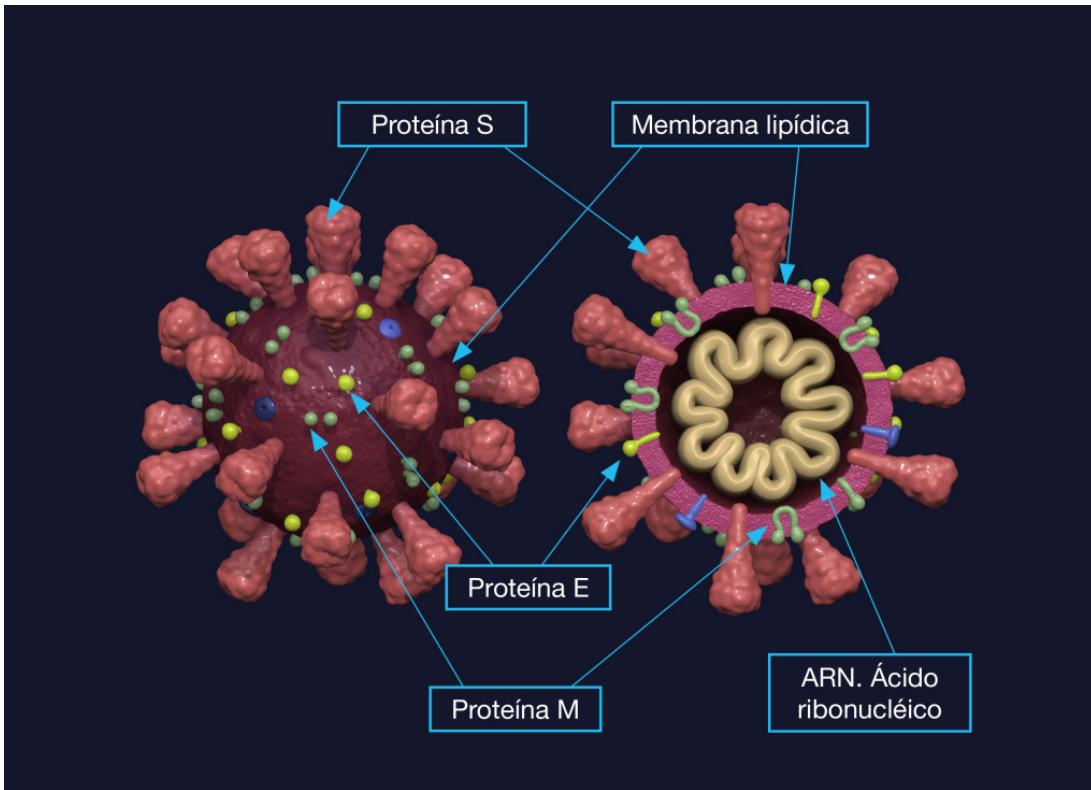


Figura 6. Estructura del virión SARS-CoV-2.

En cuanto a la nucleocápside, el genoma viral se encuentra anclado a la proteína de la nucleocápside (N), la cual está fosforilada y se localiza dentro de la bicapa de fosfolípidos de la envoltura externa.

Cada una de estas proteínas tiene funcionalidades distintas:

- La proteína S facilita la unión del virus al receptor de la célula huésped.
- La proteína de membrana M ayuda a mantener la curvatura de la membrana y la unión con la nucleocápside.
- La proteína E tiene un papel fundamental en el ensamblaje y la liberación del virus.
- La proteína N forma parte de la nucleocápside al unirse al material genético viral.
- La proteína HE, aunque no aparece en todos los coronavirus, posee una actividad esterasa que facilita la entrada del virus en la célula huésped, y además ayuda en su propagación.

Con respecto a su material genético, posee las características generales del genoma de los Coronavirus, teniendo una única cadena de ARN monocatenario positiva, la cual se asemeja estructuralmente al ARNm de las células eucariotas debido a la presencia de un CAP o capuchón metilado en el extremo 5' y una cola poliadenilada (poli-A) en el extremo 3'. A pesar de estas similitudes, este genoma contiene 6 marcos abiertos de lectura, algo que no ocurre en nuestro ARNm.

El genoma se divide en tres regiones de tamaño similar. Las dos primeras están constituidas por dos ORF, el 1a y el 1b, los cuales van a ser traducidos al principio de la infección por la maquinaria celular, dando lugar a las poliproteínas pp1a y

pp1ab, que se transcriben directamente como si de un ARNm se tratase, y que constituyen el gen de la replicasa viral. Lo especial de estas dos poliproteínas es que posteriormente van a ser degradadas por enzimas proteolíticas como la quimiotripsina (3CLpro), codificada por el genoma vírico, una proteasa principal (Mpro) y otras proteasas similares a la papaína, para dar lugar a 16 proteínas no estructurales o nsps, que se enumeran desde la nsps1 hasta la nsps16, las cuales en su mayoría están encargadas de la transcripción y replicación el ARN mensajero subgenómico.

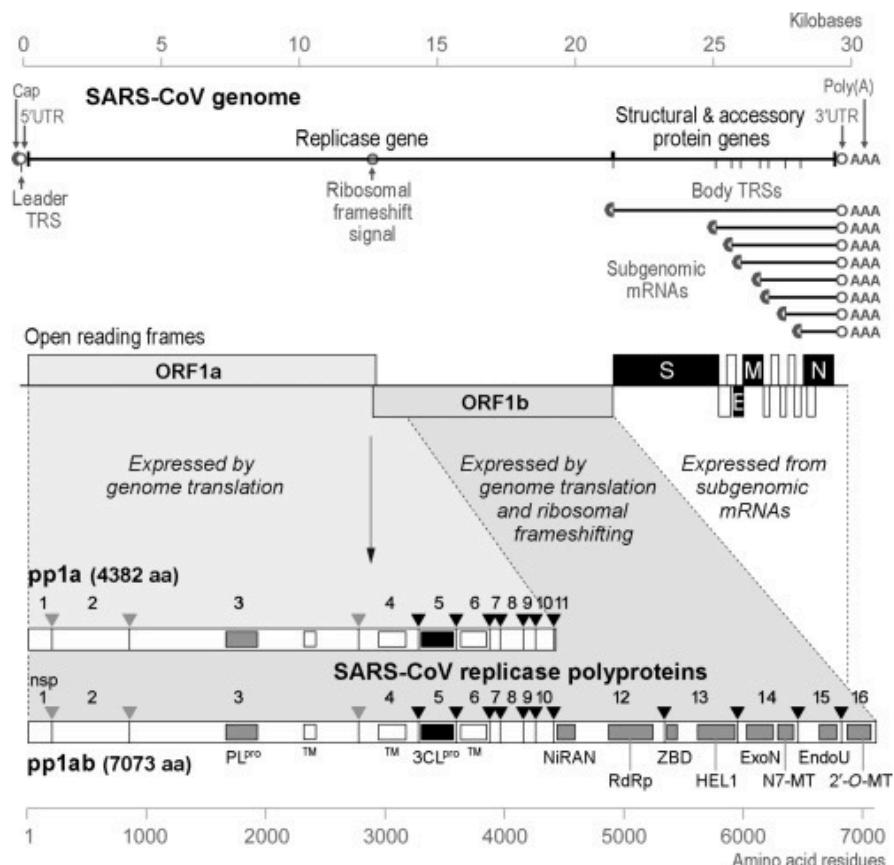


Figura 7. Organización del material genómico del SARS-CoV-2.

Como se puede observar en la figura 7, la otra tercera parte del genoma del coronavirus está destinada a albergar los genes necesarios para la traducción de las proteínas S, M, E, N, las principales proteínas estructurales, y las proteínas accesorias, expresadas a partir de los sgRNA o ARN subgenómico. Estas proteínas estructurales migrarán primero al RE y luego al Aparato de Golgi, para posteriormente ser ensambladas junto con la nucleocápside y así originar nuevas partículas del virus que se van a exportar a la membrana en forma de vesículas, esperando a ser liberadas para continuar la infección. Además, como se puede observar en la imagen, el genoma posee elementos reguladores de ARN, usados para expresar los genes aguas abajo del gen de la replicasa. Una vez liberadas de la pp1a y la pp1ab tras la proteólisis, la mayoría de las nsps de los CoVs estudiadas hasta ahora se ensamblan en un complejo ribonucleoproteico unido a la membrana que impulsa la síntesis de diferentes formas de ARN viral y que

usualmente se denomina complejo de replicación y transcripción (RTC). Asociadas en este complejo, las proteínas no estructurales van a desarrollar funciones enzimáticas, del tipo proteasa, endorribonucleasa, metiltransferasa o exorribonucleasa, todas ellas relacionadas con los mecanismos de traducción y transcripción del ARN viral. Sin embargo, hay algunas de ellas cuya función particular no está muy clara (nsps6, nsps7 o nsps8), y se cree que podrían estar relacionadas con la desregulación de la respuesta inmune en el huésped.

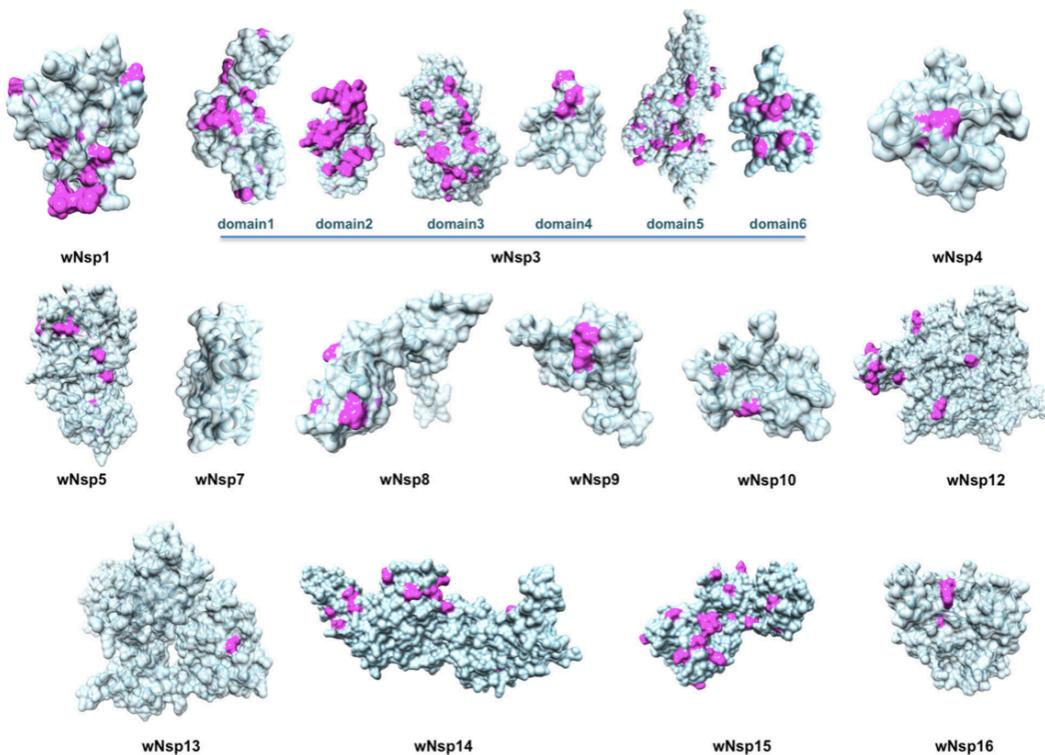


Figura 8. Las proteínas no estructurales o nsps.

El ancestro común de las replicasas de los nidovirus no está únicamente reflejado en los dominios nucleares de la replicasa, sino también en esta síntesis de un ARN mensajero subgenómico (sg) empleado en la expresión de genes localizados aguas abajo del ORF1b. Estas proteínas a su vez son fundamentales a la hora de interferir en la respuesta inmune del individuo infectado.

El hexadecámero nsp7-nsp8, 2AHM

Como se ha explicado anteriormente, estas poliproteínas 1a y 1ab son procesadas por las proteasas, para dar lugar finalmente a proteínas no estructurales maduras (nsps), la mayoría con actividad replicasa, que se ensamblan para dar lugar a la maquinaria viral de replicación y transcripción asociada a la membrana, la cual es obviamente fundamental para el mantenimiento del ciclo de infección de los virus en cuestión. Así, tras la formación de este complejo y la aparición de una serie de factores celulares específicos, dicha maquinaria es capaz de sintetizar

no solo ARN monocatenario sino una serie de ARN mensajeros subgenómicos. Estos ARNm subgenómicos están destinados a la expresión de ORFs aguas abajo del orf1ab, codificando una gran cantidad de proteínas accesorias y estructurales.

Por tanto, la maquinaria empleada para la transcripción y replicación del coronavirus incluye múltiples complejos formados por proteínas no estructurales codificadas por el propio genoma vírico. En este caso, estamos centrándonos en el súper complejo compuesto por proteínas no estructurales: el hexadecámero nsp7-nsp8, de forma que la estructura que forman es la primera que permite observar interacciones entre las replicasas del coronavirus, y que ofrece una nueva visión acerca de la sofisticada maquinaria de replicación de esta familia de virus y su maquinaria de transcripción a nivel atómico.

En el extremo 3' parte del ORF1a, el espacio de unas 1.7 kb que separa a la secuencia codificadora de nsps6 y el ORF1a/1b codifica un set de cuatro subunidades pequeñas, las cuales están nombradas desde nsps7 hasta nsps10. Estas proteínas, pese a estar altamente conservadas entre los miembros de la familia de los Coronavirinae, parecen carecer de funciones enzimáticas. En su lugar, parecen haber surgido como factores de diferentes interacciones y moduladores de las enzimas nucleares codificadas por el ORF1b, como la nsps12 (con actividad polimerasa), la nsps14 (una exorribonucleasa) y la nsps16 (una ribosa 2'-O-metil transferasa). Más allá de esto muchas de ellas han demostrado interaccionar con el ARN.

Estas subunidades se localizan por tanto en la región perinuclear de las células infectadas, donde se acumulan los orgánulos membranosos de replicación de los CoVs.

El complejo proteico hexadecamérico (nsp7-nsp8)

La proteína que me ha sido asignada para el desarrollo de este proyecto es el hexadecámero 2AHM formado por las proteínas no estructurales nsps7 y nsps8, que a su vez pertenecen al completo de replicación y transcripción (RTC) del SARS-CoV-2.

Se trata de una proteína viral perteneciente al SARS-CoV-2 y que se encuentra implicada en la replicación del genoma del mismo. Como la información que aparece en la base de datos de proteínas data del año 2006, probablemente también se encuentre en otros virus de la familia del SARS-CoV, como el SARS-CoV-1, ya que además se menciona que dicha proteína no ha presentado mutaciones desde que se describió por primera vez su estructura (PDB).

El complejo que forman presenta un cilindro hueco con una morfología externa de “tipo donut”, en la cual existe un canal central que está conformado principalmente por aminoácidos con cadenas laterales cargadas positivamente, el cual se cree que está implicado en las interacciones con el ARN de doble cadena. La superficie exterior de la estructura está predominantemente formada por aminoácidos de carga negativa.

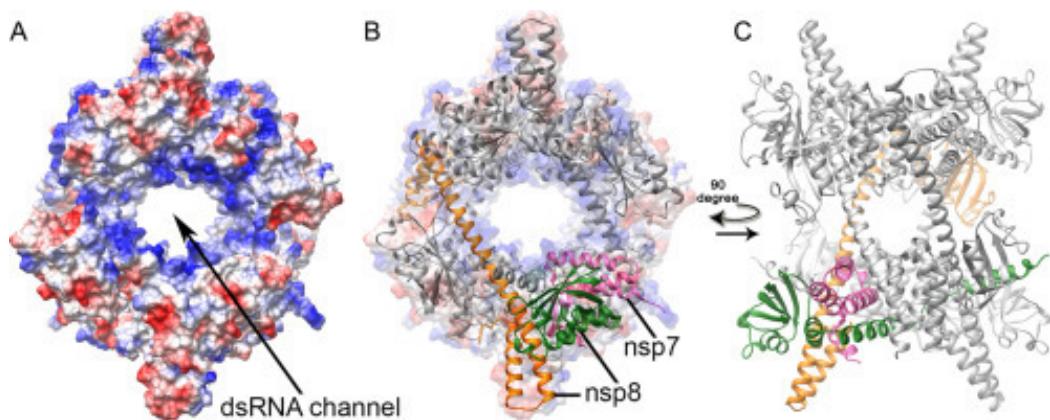


Figura 1. Estructura cristalina del hexadecámero nsp7-nsp8 (2AHM). En A, observamos una representación del complejo coloreado en función de sus cargas, siendo las regiones en rojo aquellas con carga negativa, y las azules con una carga positiva. En B y C, se representa a la proteína nsp7, en rosa, y a las dos conformaciones posibles que adopta nsp8, en naranja y verde.

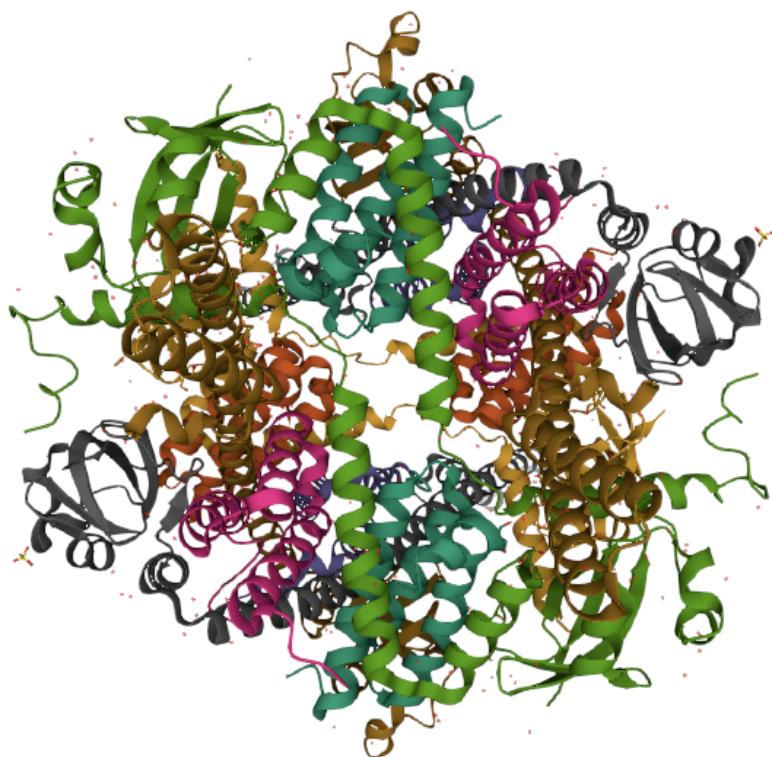


Figura 2. Imagen de la estructura terciaria del complejo/hexadecámero nsp7-nsp8. Imagen extraída del visor 3D de PDB.

La proteína nsp7

Nsp7 es una proteína de unos 83 aminoácidos formada por cuatro hélices alfa, las cuales presentan orientación espacial y posición de las unas con respecto a las otras bastante diferentes. Esto sugiere que la conformación de esta proteína depende en gran medida de la interacción con nsps8 (en concreto, en cuanto a la cuarta hélice). Así, se compone de un núcleo central formado por un ramillete helicoidal o helical bundle (HB) N-terminal, con las hélices HB1, HB2 y HB3, las cuales contienen los residuos del 5 al 26, del 30 al 47 y del 49 al 68, respectivamente, formando una bobina enrollada anti paralela con 3 hebras y con un paso superhelical a la derecha. Además, tenemos una pequeña hélice que sale un poco de este ramillete que interactúa entre sí, la hélice corta HTC, que contiene los residuos del 70 al 78.

Los cuatro monómeros de nsp7 se superponen bien cuando se encuentran en una unidad asimétrica. Sin embargo, HTC individualmente conserva cierta movilidad. La interacción en el ramillete entre HB1 y HB3 se da gracias a la presencia de interacciones entre residuos hidrófobos.

Además, se ha observado que HB1, 2 y 3 están altamente conservadas en otros virus de la familia del coronavirus, mientras que la secuencia de HTC está menos conservada. Esto por ejemplo podría deberse al rol tan importante en la interacción con nsps8 que tiene este ramillete helicoidal.

Se considera que nsp7 tiene una gran importancia en el proceso de replicación del virus, aunque el impacto de las mutaciones puntuales en diversos experimentos fue menor de lo previsto sobre la base de la caracterización bioquímica de las propiedades de unión al ARN de los complejos proteicos que contienen nsp7 in vitro.

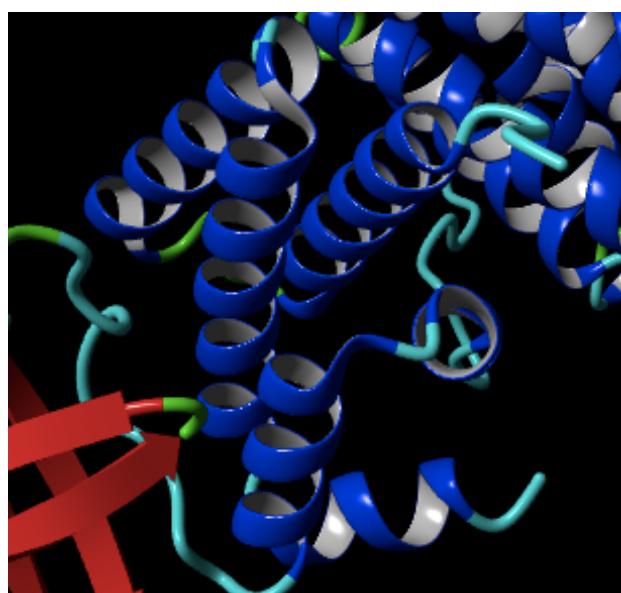


Figura 3. La proteína no estructural nsp7. Imagen extraída del visor YASARA.

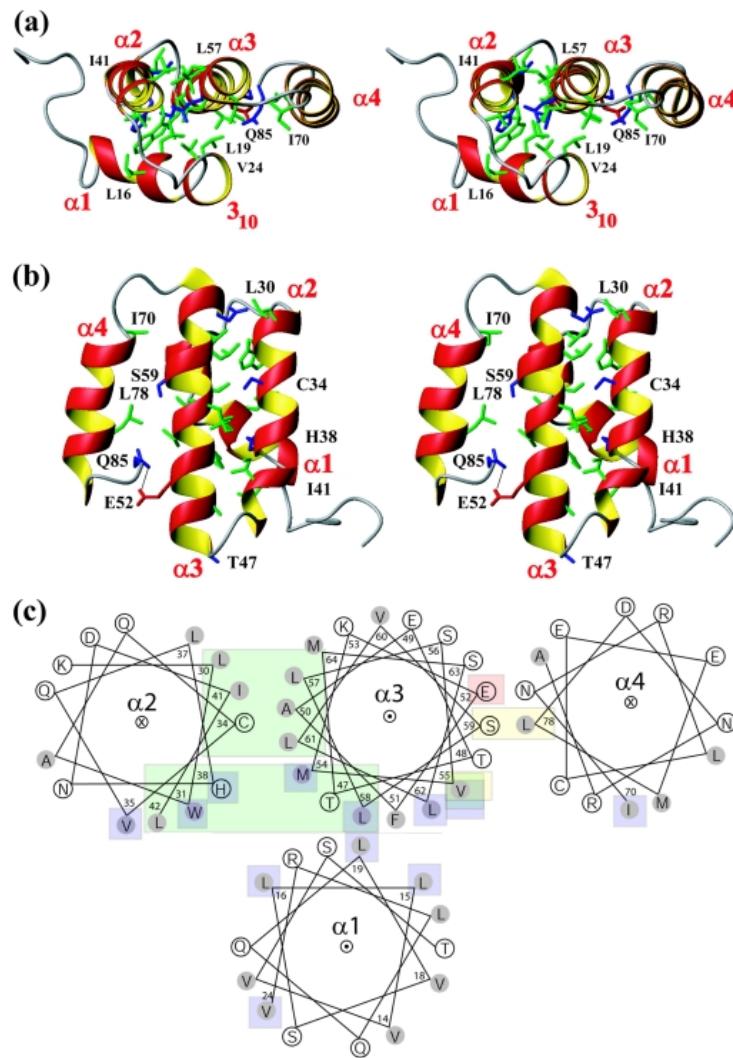


Figura 4. Representación de las hélices de nsp7 a) Desde arriba, marcándose el extremo N terminal y marcando aquellos residuos implicados en la interacción entre todas ellas, b) Al igual que en la figura 3, pero de nuevo marcando aquellos aminoácidos implicados en su interacción, y c) una imagen esquemática desde arriba en la cual se indican las zonas de interacción entre todas ellas.

La proteína nsp8

Los cuatro monómeros de nsp8 en una unidad asimétrica adoptan dos conformaciones bien diferenciadas:

- nsp8I, que serían las cadenas G y H, tendrían una estructura descrita en la bibliografía como de tipo ‘golf-club’, formada por un dominio ‘en flecha’ N-terminal y un dominio C-terminal ‘en cabeza’, formado por los residuos del 6 al 104 el primero y del 105 al 196 el segundo. Este dominio en punta de flecha contendría tres hélices, denominadas NH-1, NH-2 y NH-3, siendo una de ellas de una notable longitud (NH-3). Otras tres hélices alfa (CH-1, CH-2 y CH-3) y 7 láminas beta (Beta1-7) compondrían el dominio en cabeza, formando un

plegamiento alfa/beta. Estas 7 láminas beta forman un barril beta, con dos láminas beta antiparalelas empaquetadas ortogonalmente. Más de la mitad de los residuos del dominio en cabeza C-terminal son hidrofóbicos, de modo que el dominio al completo tiene un núcleo de alto carácter hidrofóbico.

- nsp8II, formado por las cadenas de la E a la F, representan de la misma forma un ‘golf-club’ con un eje doblado. A pesar de que su dominio en cabeza sea muy similar al del nsp8I, la hélice NH3 se dobla en dos hélices más pequeñas, denominadas NH3alfa y NH3beta, unidas por una cola denominada C3. El resto de la estructura es muy similar a nsp8I.

Múltiples análisis de alineamientos de secuencia correspondientes a las proteínas nsp8 revelan el alto grado de conservación de las mismas, estando el dominio N-terminal está más conservado que el C-terminal. Este hecho sugiere que el dominio N-terminal puede tener un gran papel en la interacción con otras moléculas y ensamblaje del complejo.

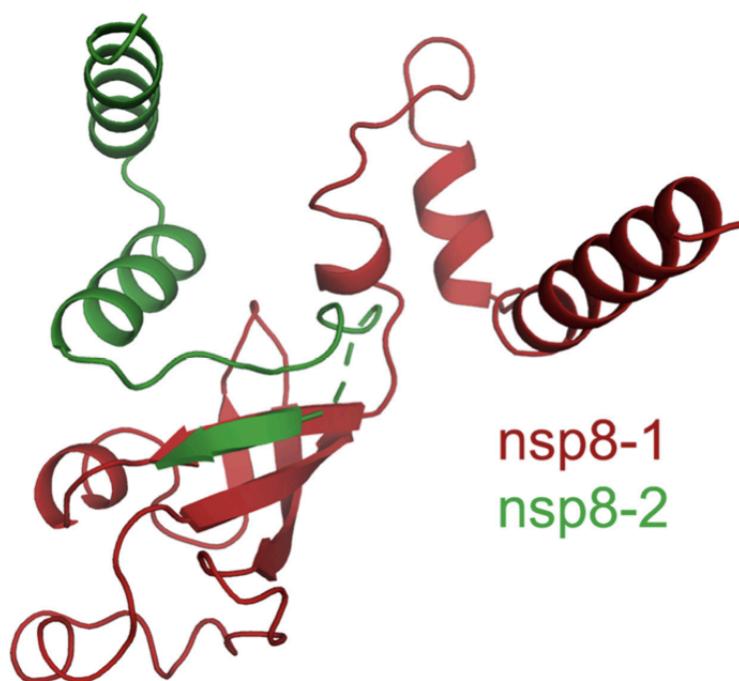


Figura 5. Imagen que representa las dos conformaciones posibles en las que se encuentra nsp8.

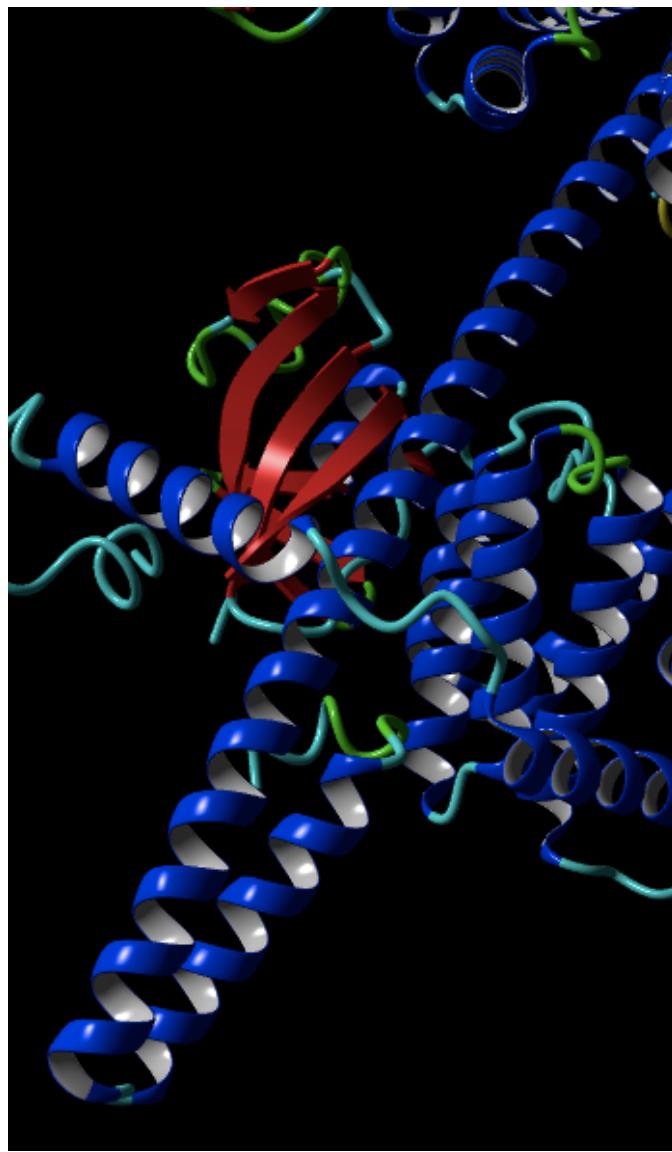


Figura 6. La proteína nsp8, extraída del visor 3D YASARA.

Interacciones entre nsp7 y nsp8

Nsp8I y nsp8II interactúan estrechamente con nsp7, para dar lugar a un área enterrada de unos 1400 Å, formando dos tipos de heterodímeros: D1 y D2, respectivamente. Nsp8I y nsp8II interactúan con nsp7 con los mismos sitios de interacción, lo cual nos sugiere que el cambio conformacional que sufre nsp8 no ocurre con motivo de la unión con nsp7.

Cuando se produce la interacción entre las cadenas, estas lo hacen en relación 4:4, es decir, hay 4 cadenas de interacción en nsp7, A, B, C, y D (cadenas ligeras), que interactúan con cuatro cadenas de nsp8, E, F, G y H (cadenas pesadas), por cada unidad asimétrica.

Los residuos de la unión entre de nsp8 y nsp7 forman un corazón hidrófobo. Además, se forman puentes de hidrógeno entre las principales cadenas de nsp8 y nsp7 en diferentes puntos de unión, lo cual confiere especial estabilidad a la molécula en su conformación nativa. A su vez, una mayor estabilidad repercutirá en la posterior funcionalidad del complejo.

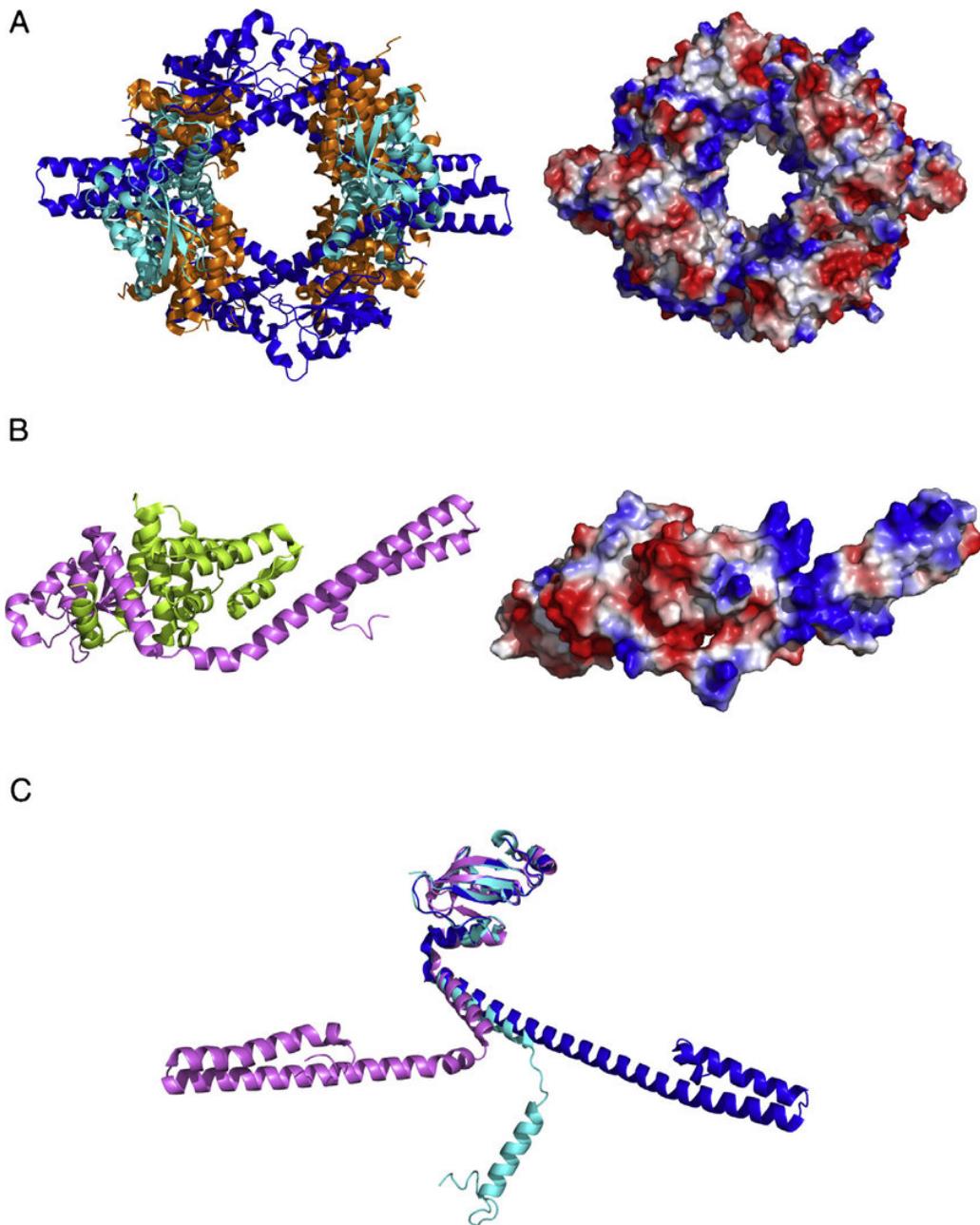
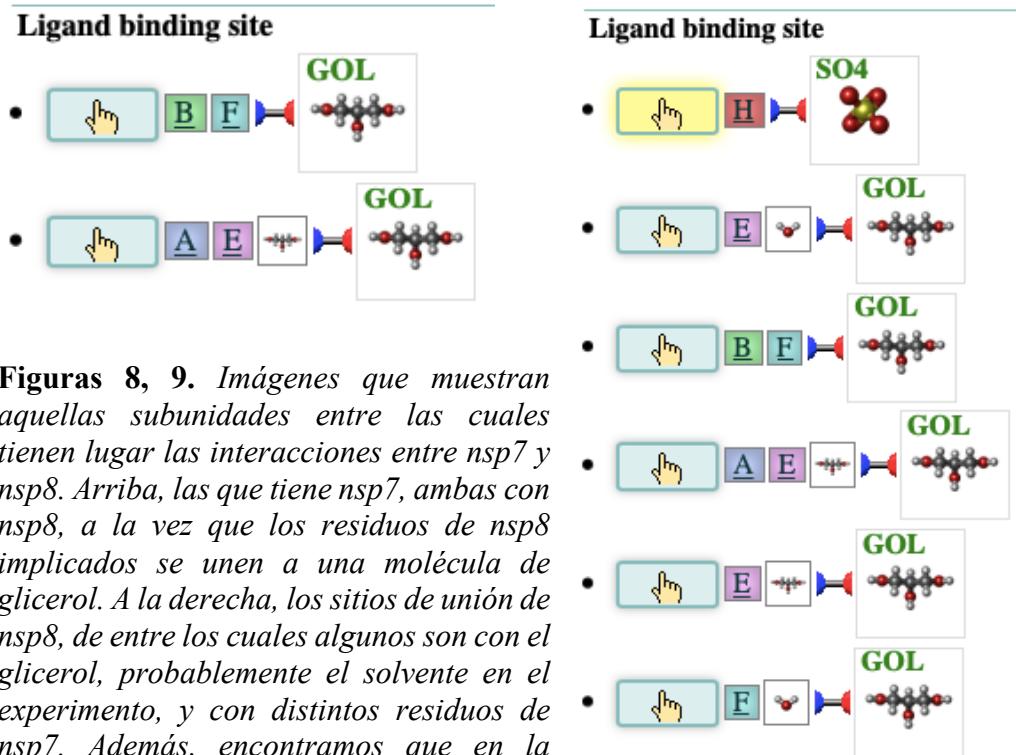


Figura 7. (A) Estructura del hexadecámero nsp7-nsp8 del SARS-CoV. A la izquierda, nsp7 y las dos conformaciones de nsp8 están coloreadas en naranja, cián y azul oscuro, respectivamente. A la derecha, tenemos la representación de colores en función del potencial electrostático (azul: carga positiva, rojo: carga negativa). (B) A la derecha, estructura del heterodímero nsp8/nsp7. Dos moléculas nsp7 (en verde) están asociadas a una molécula de nsp8 (en rosa). A la izquierda, la superficie coloreada en función del potencial electrostático. (C) Superposición del dominio C-terminal de las dos conformaciones de la molécula

nsp8 (en cián y azul oscuro) con *FCoV* en rosa. El extremo N-terminal es muy flexible, incluso en asociación con *nsp7*.



Figuras 8, 9. Imágenes que muestran aquellas subunidades entre las cuales tienen lugar las interacciones entre *nsp7* y *nsp8*. Arriba, las que tiene *nsp7*, ambas con *nsp8*, a la vez que los residuos de *nsp8* implicados se unen a una molécula de glicerol. A la derecha, los sitios de unión de *nsp8*, de entre los cuales algunos son con el glicerol, probablemente el solvente en el experimento, y con distintos residuos de *nsp7*. Además, encontramos que en la subunidad *H* se dan uniones a un ión sulfato.

Ensamblaje y arquitectura del súper complejo

Los heterodímeros D1 y D2 pueden dimerizar para formar los heterotetrámeros T1 y T2. De esta forma, las interacciones entre T1 y T2 permiten la construcción total del súper complejo hexadecamérico.

La arquitectura de este complejo hexadecamérico *nsp7-nsp8* es única entre todos los complejos macromoleculares que contienen dos tipos de proteínas o subunidades descritos hasta la fecha. En otras estructuras, los multímeros homólogos de un tipo siempre se apilan en aquellos con un tipo diferente, de forma que van formando capas simétricas. La ausencia de cualquiera de estos tipos de proteína afecta marcadamente a la forma global del complejo. Dos ejemplos podrían ser los del complejo multienzimático Rubisco y el complejo GroEL-GroES-ATP. Por otro lado, la relación existente entre los multímeros *nsp7* y *nsp8* en el súper complejo no es de apilamiento, sino de enlazamiento cruzado. *Nsp8I* y *nsp8II* constituyen el marco del supercomplejo, a modo de ladrillos, y *nsp7* estabiliza y rellena la configuración, como el cemento que une a dichos ladrillos. La pérdida de *nsp7* no cambiaría especialmente la forma de la estructura. Por tanto, podemos concluir que el hexadecámero *nsp7-nsp8* demuestra un nuevo modo de arquitectura proteica en complejos macromoleculares grandes nunca antes estudiado.

Interacción con el ARN de doble cadena

Las propiedades electroestáticas y las dimensiones específicas del complejo nsp7-nsp8 demuestran que su rol principal en el período de replicación e infección del virus es el de la unión de ácidos nucleicos. El canal interno está provisto de un potencial positivo, mientras que la superficie exterior del cilindro que conforman se encuentra cargada negativamente. Esta distribución parcial de las cargas asegura que el esqueleto de ácidos nucleicos pueda pasar a través del canal sin repulsiones electrostáticas, así como otras proteínas de unión al ADN/ARN. Es más, el canal central del súper complejo tiene un diámetro interno medio de unos 30 Å, lo que permite la acomodación perfecta del dúplex ADN/ARN.

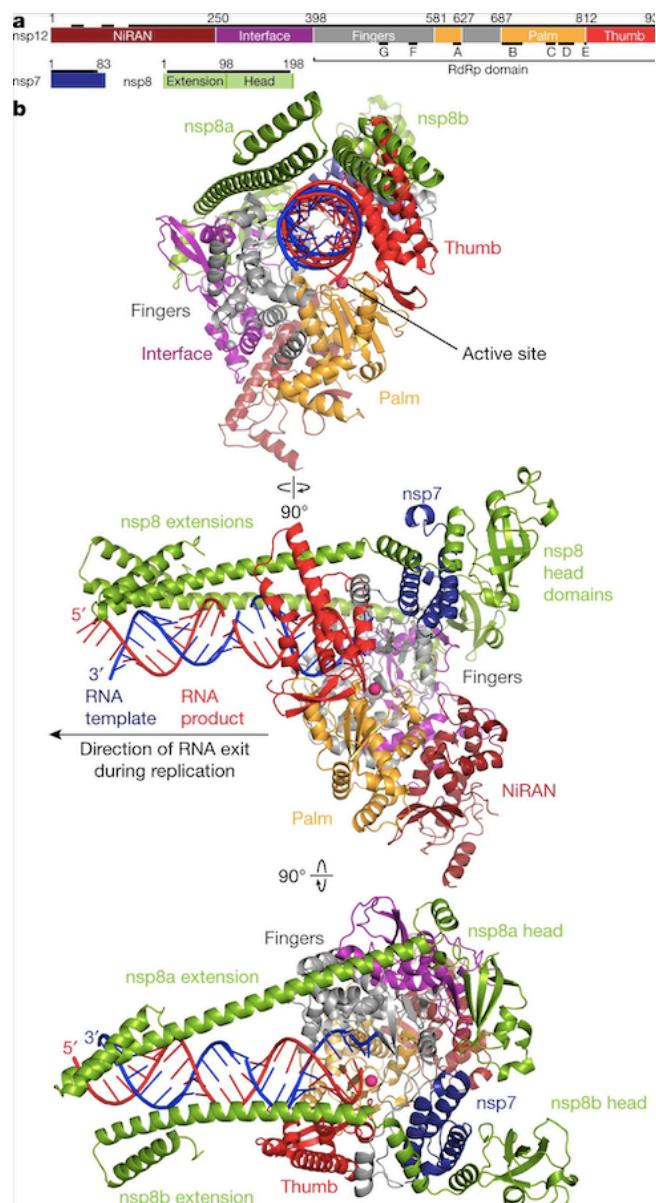


Figura 11. Estructura de dominio de las subunidades nsp12, nsp8 y nsp7 de RdRp. En nsp12, se representan los motivos de secuencia conservados A-G16.

Las regiones incluidas en la estructura se indican con barras negras. b, Tres vistas de la estructura, relacionadas por rotaciones de 90° (arriba, vista trasera; centro, vista lateral; abajo, vista superior). Código de colores para nsp12 (NiRAN, interfaz, dedos, palma y pulgar), nsp8, nsp7, plantilla de ARN (azul) y producto de ARN (rojo) utilizados en todo el proceso. La esfera magenta representa un ion metálico modelado en el sitio activo.

Se ha aceptado que la replicación del genoma del coronavirus ocurre en el citoplasma de las células infectadas, y que el ADN no está implicado en dicho proceso. Por tanto, se requiere de un ARN de doble cadena que actúe como intermediario en la replicación genómica de todos los coronavirus durante el proceso de síntesis del ARN. La estructura hueca cilíndrica del hexadecámero nos sugiere que su función es rodear y estabilizar al ARN de doble cadena, sujetando así las hebras nacientes molde juntas para facilitar una transcripción y replicación eficientes. Todo este procedimiento ocurriría cuando nsp7-nsp8 se encuentran en un estado de interacción con la proteína nsp12 o RdRp, encargada de la replicación del ARN del virus. Además, tras algunos experimentos en los cuales se introducían mutaciones del hexadecámero para evaluar su función, se demostró que los mutantes de nsp8 tenían mayor afinidad por el ARN de doble cadena que por el ADN bicatenario, lo que sugiere que el ARN bicatenario es un compañero natural en la unión al complejo nsp7-nsp8.

Teniendo en cuenta todo lo anterior, el hexadecámero podría ser un factor que se une y sigue a la proteína nsp12, confiriendo una alta procesividad para la replicación eficiente del genoma extremadamente grande del coronavirus.

La co-localización de nsp7, nsp8, nsp9 y nsp10 en diferentes experimentos con MVH proporcionan una evidencia muy clara de las interacciones de la misma con el virus.

Los experimentos analíticos con ultra centrifugación indican que nsp8 interactúa con nsp9, una proteína de unión al ARN monocatenario. Además, el desorden en las regiones N-terminales de nsp8 parece disminuir sobre la adición de nsp9 a nsp8. En las bases de la estructura del complejo nsp7-nsp8, el sitio de unión más probable a nsp9 sería en la región formada por los 50 residuos situados en la región N-terminal de nsp8II, la cual está localizada en la entrada del canal y que tiene una alta flexibilidad con pérdida de densidad electrónica.

Por tanto, se considera que la función del dímero nsp9 podría ser proteger del procesamiento de nucleasas de las hebras nacientes y molde recién desenrolladas que emergen del canal del complejo nsp7-nsp8, que aún no han formado una estructura secundaria estable.

1.7. Mecanismos de patogénesis del SARS-CoV-2

En cuanto al mecanismo de infección, el SARS-CoV-2 se une a la superficie celular gracias a la proteína spike (S) del virus, y el receptor de la enzima convertidora de angiotensina (ACE-2), de igual modo que ocurría en el caso del

SARS-CoV. La ACE-2 es una enzima localizada en las células del epitelio pulmonar bajo, en el corazón, en los riñones, la vejiga, el esófago e intestino, y que está encargada de contribuir a la regulación de la presión arterial mediante la transformación de angiotensina I en angiotensina. En concreto, en los pulmones la encontramos en las células alveolares de tipo II.

La proteína spike posee 2 subunidades, la S1 y la S2, de modo que la subunidad S1 es aquella encargada de efectuar la unión a ACE-2, mientras que S2 se encarga de la fusión entre ambas membranas en contacto. Una vez se produce la unión de las membranas, la proteína S va a ser escindida por una proteasa, en dos posiciones distintas, lo que va a permitir la entrada definitiva del virus en la célula. Una vez esto ocurre, la nucleocápside se libera y permite la salida del RNA genómico.

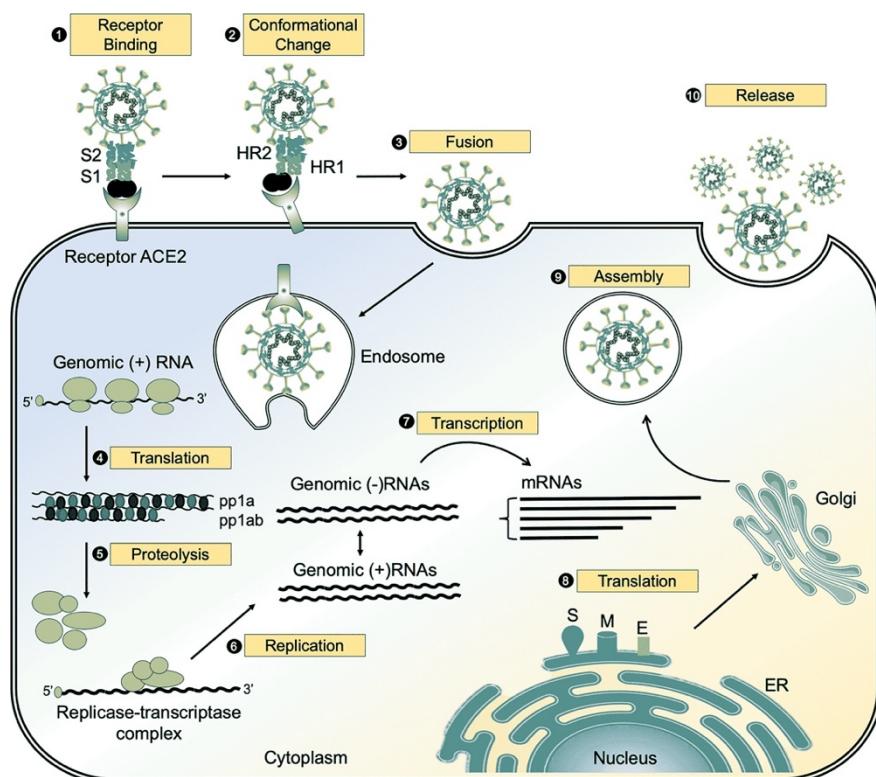


Figura 1. Proceso infectivo del SARS-CoV-2.

1.7. Respuesta inmune frente al SARS-CoV-2

En cuanto a la respuesta inmune del organismo frente al SARS-CoV-2, encontramos en primer lugar, la respuesta humoral y, en segundo lugar, la respuesta celular.

Para hacer frente al patógeno mediante los mecanismos de inmunidad innata, es necesario que se produzca una detección por parte de los receptores de reconocimiento de patrones, en concreto TLRs, los cuales van a encargarse de reconocer los PAMPs, o los patrones moleculares asociados a patógenos, presentes en el virus. Estos PAMP, en el caso del CoV-2, están asociados a sus ácidos nucleicos, de modo que cuando la proteína S se une al receptor ACE2

presente en la célula huésped y se fusiona con la membrana de dicha célula, se forma un endosoma gracias al cual el virus consigue ingresar en la célula. De esta forma, una vez que el virus se encuentra en este compartimento, los receptores tipo toll que se encuentran allí (TLR3, TLR7, TLR8 y TLR9) reconocen dichos PAMPs, lo que activa varias vías de señalización y varios factores de transcripción, como el factor nuclear kappa B (NFkB), la proteína activadora AP-1, y el IRF7 o factor regulador del interferón 7 y el del 3. Todas estas están encargadas de promover la liberación de numerosas citoquinas y quimioquinas encargadas de desencadenar el proceso inflamatorio. Además, se estimula al interferón de tipo I (INF- α /INF- β), encargados de suprimir la replicación del virus.

Por otro lado, está la respuesta secundaria o celular, en la cual es fundamental el papel de los linfocitos T a la hora del reconocimiento del SARS-CoV-2, y además en la destrucción de aquellas células afectadas, que como sabemos son especialmente las células pulmonares del individuo infectado. Además, se ha observado que usualmente al comienzo de la infección, se suele producir una mayor síntesis de IgM, y en fases más tardías, aumenta la proporción de IgG. Dado que se demostró con el SARS-CoV que aquellos determinantes antigenicos de reconocimiento por parte de las células de la respuesta adaptativa estaban localizados en las proteínas estructurales principales (S, M, N y E), estos podrían ser posibles epítopos también en el caso del SARS-CoV-2 frente a los cuales dirigir tratamientos terapéuticos o vacunas. De hecho, se ha demostrado que la infección por SARS-CoV-2 estimula la producción de IgG contra la proteína N.

1.8. Estrategias terapéuticas contra el SARS-CoV-2 dirigidas al complejo.

Pese a que actualmente se están desarrollando novedosas técnicas de ataque al SARS-CoV-2, tanto por el lado de vacunas de ADN y ARN, las cuales son muy novedosas, como de fármacos dirigidos a ciertas proteínas estructurales del mismo (como, por ejemplo, la proteína Spike que interviene en la internalización del virus en la célula infectada), no se han descrito todavía ningún fármaco que ataque directamente al complejo hexadecamérico o a alguna de estas dos proteínas estructurales. Sin embargo, y no dejando atrás la complejidad que presenta el diseño de alguno de estos fármacos contra ciertas estructuras, me aventuro a decir que en el futuro diseño de alguna estrategia terapéutica sería interesante plantear el bloqueo de la acción de estas proteínas no estructurales. Como he comentado anteriormente, son fundamentales en la unión a la proteína nsp12 o RdRp, la cual está encargada principalmente de la transcripción del material genético del virus, potenciando la actividad de la misma. De esta forma, si se impidiese de alguna forma la formación del complejo nsp7-nsp8-nsp12, podríamos interferir en la replicación del material genético viral.

BLOQUE II: Bases de datos secuenciales y estructurales

Actividad 2. Desarrollo de un diagrama de decisión

Para el desarrollo de esta actividad, resulta fundamental el diseño de un diagrama de decisión que, en primer lugar y sin necesidad del uso de ninguna aplicación, nos permita determinar los pasos a seguir a la hora de reconocer la identidad de un fichero cualquiera en función de su contenido.

Para ello, he desarrollado la siguiente plantilla, en la cual resulta más visual entender cuáles serían los eventos determinantes a la hora del reconocimiento del tipo de fichero ante el que nos encontramos:

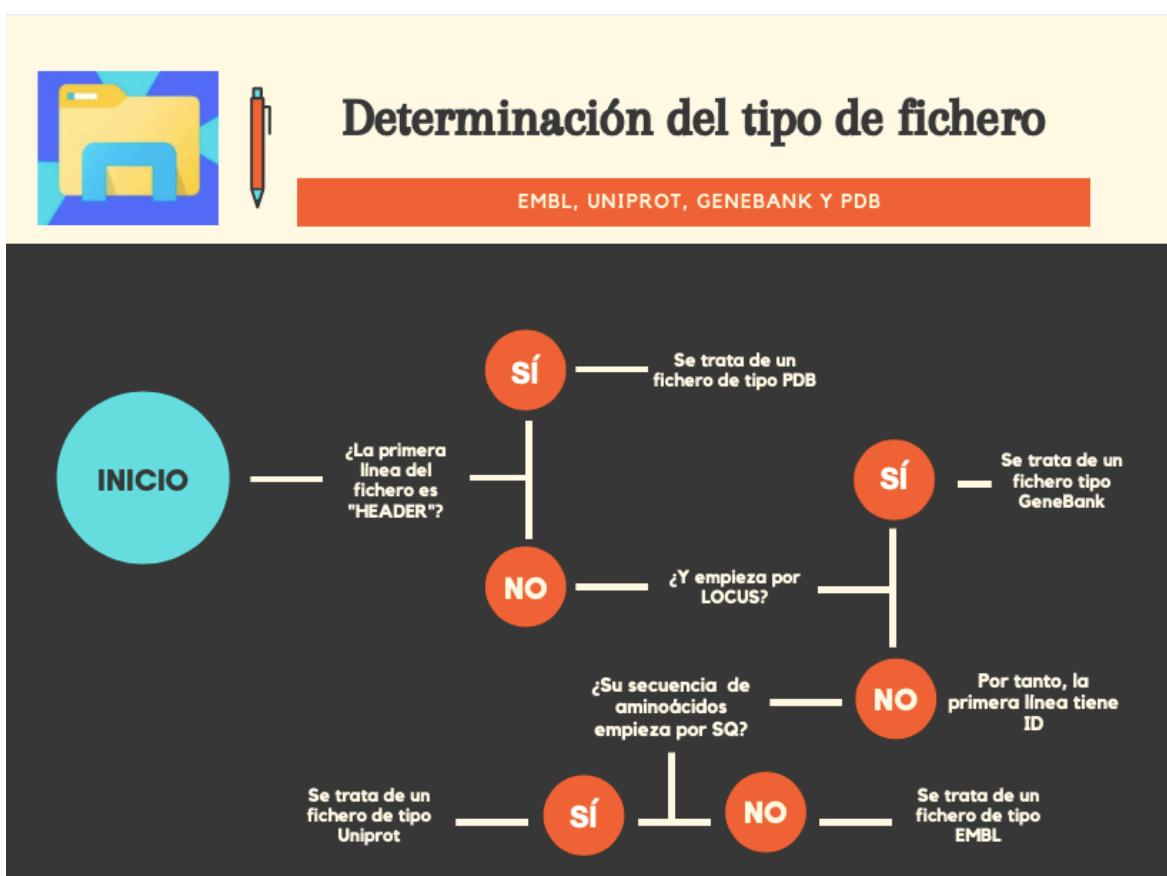


Figura 1. Diagrama de decisión entre formatos EMBL, UniProt, Genbank y PDB. Diseño extraído desde www.canva.com/

Adicionalmente, resulta necesario el desarrollo de una aplicación que nos permita diferenciar entre el tipo de fichero PDB, Uniprot y Genbank haciendo uso de un código que reconozca ciertas estructuras presentes en el texto de dicho fichero y que nos permita extraer la secuencia de aminoácidos de 1 letra de la proteína asignada, en este caso, del hexadecámero.

Para ello, he diseñado una aplicación nueva en el entorno Lazarus, la cual se denomina “Diagrama”, en la cual he añadido al Form1 un primer botón que permite la carga del fichero sea del tipo que sea en el Memo1.

He añadido otro botón (2), el cual mediante el uso de un condicional if triple, va a detectar el tipo de fichero del cual se trata, analizando la cadena que existe en la primera posición de la primera línea del Memo1, que hace claramente diferenciable a cada uno de los ficheros que vamos a tratar de distinguir: Uniprot, PDB y Genbank. Una vez determinado el tipo gracias a los condicionales, el nombre o etiqueta del tipo de archivo se va a presentar en el Edit2. Por último, en dicho botón se incluye la llamada a una función creada en mi librería biotools, la cual se denomina ExSecuencia, la cual nos permite extraer la secuencia de aminoácidos con el código de 1 letra del fichero sea del tipo que sea.

Esta función recibe como argumentos el fichero y el tipo, ambas variables de tipo string, y también devuelve un string, que vendría a ser la secuencia de aminoácidos. Para ello, esta función a grandes rasgos se encarga de ejecutar un bucle for distinto en función del tipo de fichero ante el que nos encontramos, y con el va a recorrer las distintas líneas del memo, reconocer determinadas peculiaridades de escritura existentes en cada tipo de fichero, las que van a ser asignadas como el inicio y el final de la secuencia, y va a extraer de dichas líneas o filas la secuencia de aminoácidos en cuestión.

Como se puede observar en el código, al final se trata de ir añadiendo filas a la variable “secuencia”, la cual se va a encargar de ir almacenando las sucesivas filas con secuencias.

Cabe destacar que, en el caso de que el tipo de fichero sea ‘PDB’, hacemos directamente uso de la función CargarPDB definida en nuestra librería, obteniendo la secuencia con el simple comando p.Secuencia, definido para el desarrollo de otras aplicaciones.

Los resultados que pueden observarse tras la ejecución del proyecto y usando como ejemplo nuestro hexadecámero se muestran a continuación:

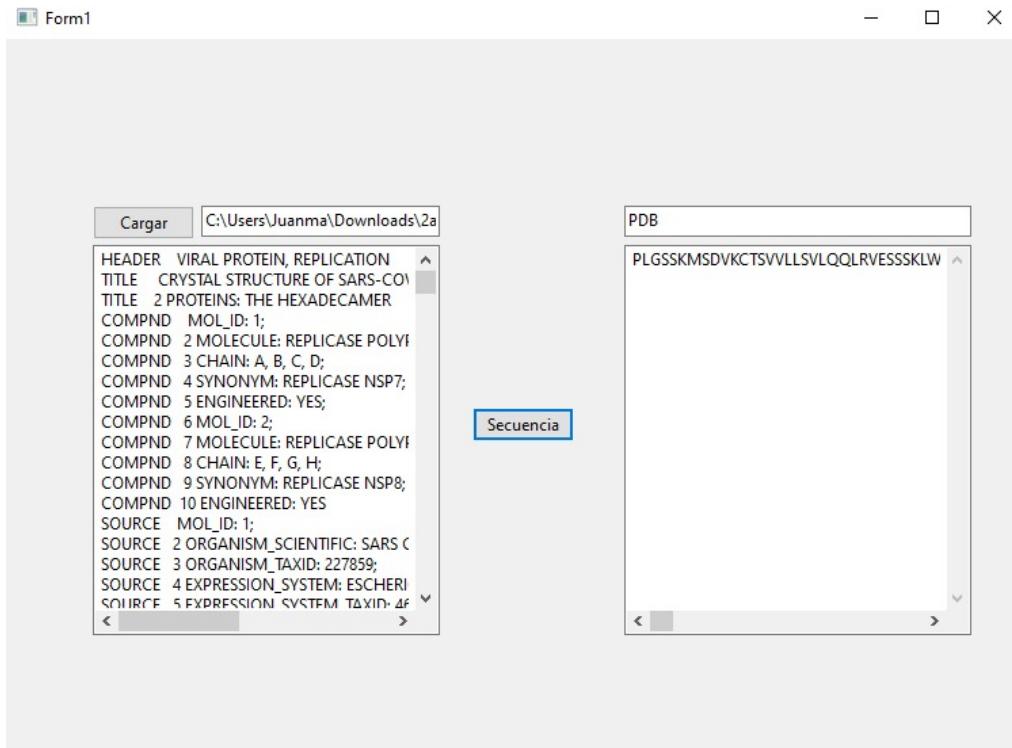


Figura 2. Imagen tomada de la actividad “Diagrama” en el entorno Lazarus.

Usando como ejemplo un texto en formato UniProt, obtenemos lo siguiente:

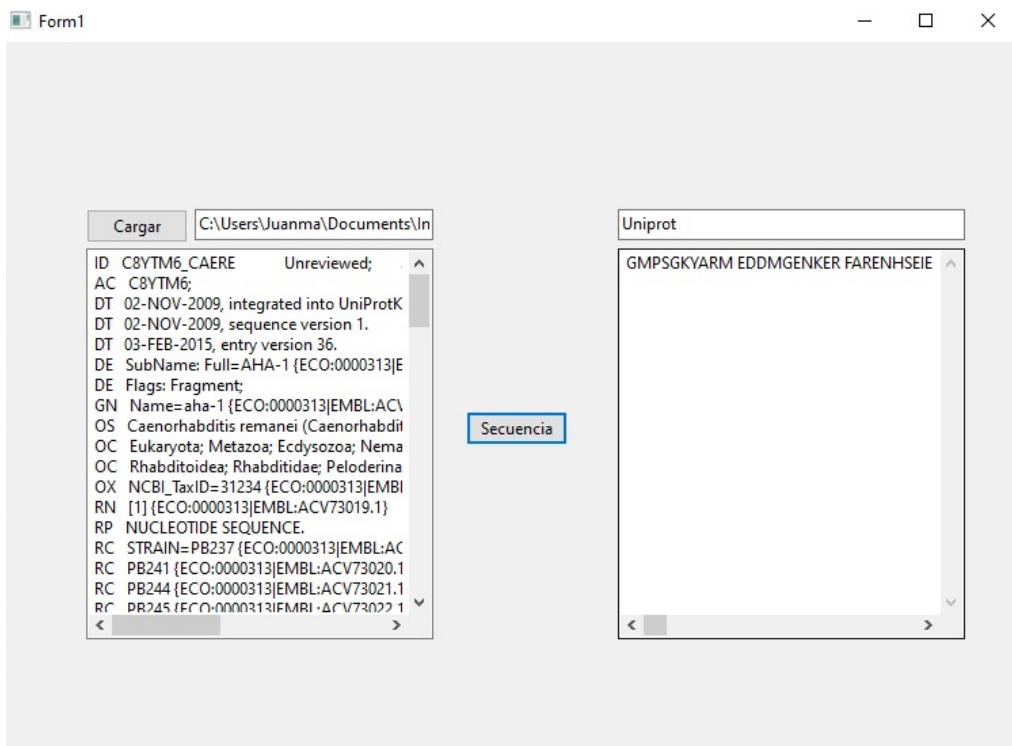


Figura 3. Resultado de la aplicación “Diagrama” en el entorno Lazarus.

Como podemos comprobar en las figuras, la actividad funciona adecuadamente, permitiéndonos extraer la secuencia de aminoácidos de una letra una vez detectado el tipo de archivo introducido en el programa.

Actividad 3. Función de parsing de ficheros: CargarPDB

En esta actividad ha sido necesario programar una función, la cual hemos denominado **CargarPDB**, la cual nos permite integrar, por ejemplo, en un Memo que designemos, todos aquellos datos relevantes a extraer a la hora de hacer uso de un fichero PDB para el desarrollo de actividades sucesivas.

Para ello, como bien indica en el enunciado, esta función se encarga de leer los datos desde un fichero PDB y los almacena en una estructura matricial del tipo TPDB, definido a su vez en nuestra librería biotools, que va a contener los campos relativos a la identificación de los átomos (ID, número de átomo, residuo y subunidad a la que pertenecen, las coordenadas, una localización alternativa...), el residuo (su ID de 3 letras, su ID de 1 letra, el número de residuo, la subunidad a la que pertenece, la secuencia...) y la subunidad (su ID, el átomo inicial y el residuo inicial...) a la que pertenecen.

Dado que se trata de una función compleja, en la que además hemos añadido el cálculo de phi y psi para cada residuo calculado gracias al uso de las coordenadas cristalográficas, y la cual se usa en numerosas actividades consecutivas, por no decir en todas, para obtener una visión más específica de la misma se debe revisar la librería biotools, en la cual se puede leer el código y además se incluyen los numerosos comentarios que he ido anotando a lo largo del desarrollo de la asignatura y de la función en sí misma.

Sin embargo, para hacernos una idea básica de qué es lo que hace esta función, adjunto a continuación un Memo en el cual se ha cargado el resultado de la misma:

```
Cargar C:\Users\Juanma\Downloads\2a
HEADER VIRAL PROTEIN, REPLICATION
TITLE CRYSTAL STRUCTURE OF SARS-COV2
TITLE 2 PROTEINS: THE HEXADECAMER
COMPND MOL_ID: 1;
COMPND 2 MOLECULE: REPLICASE POLYF
COMPND 3 CHAIN: A, B, C, D;
COMPND 4 SYNONYM: REPLICASE NSP7;
COMPND 5 ENGINEERED: YES;
COMPND 6 MOL_ID: 2;
COMPND 7 MOLECULE: REPLICASE POLYF
COMPND 8 CHAIN: E, F, G, H;
COMPND 9 SYNONYM: REPLICASE NSP8;
COMPND 10 ENGINEERED: YES
SOURCE MOL_ID: 1;
SOURCE 2 ORGANISM_SCIENTIFIC: SARS-CoV-2
SOURCE 3 ORGANISM_TAXID: 227859;
SOURCE 4 EXPRESSION_SYSTEM: ESCHERICHIA_CE
SOURCE 5 EXPRESSION_SYSTEM_TAXID: 462
```

Figura 1. Resultado de la función “CargarPDB” en el entorno Lazarus.

BLOQUE III. *Visualización de proteínas*

Actividad 5. Reconstrucción de líneas de texto: WritePDB

En esta actividad se requería del desarrollo de una función denominada WritePDB, la cual nos permitiese reconstruir una línea de texto con el formato correcto de una línea ATOM de un PDB.

Para el desarrollo de la misma, he creado una unidad nueva y además he desarrollado la función WritePDB dentro de nuestra librería de funciones biotools.

En la unidad, en primer lugar, he diseñado un botón 1 el cual nos permite cargar en un Memo el fichero PDB de la proteína con todos los datos de tipo TPDB. Además, existe un botón 2, al cual he denominado Write PDB, de modo que, al pulsarlo, se va a llamar a la función WritePDB existente en el biotools añadido al proyecto.

En cuanto a la función **WritePDB**, es una función que acepta como argumentos variables atm del tipo **TAtomPDB**, la cual nos va a devolver un string que va a consistir en una fila que suma todos los valores de interés para la línea ATOM que vamos a mostrar en el Memo2. Para ello, tenemos que incluir en dicha fila el número de átomo, el número del residuo, las coordenadas del mismo (X, Y, Z) y el factor de temperatura. Dado que estos valores son números reales/enteros, para incluirlos en la línea susodicha debemos convertirlos en cadena. Sin embargo, es necesario tener en cuenta que las líneas de TAtom requieren de un riguroso cuidado tanto del encolumnado como del número de decimales de los valores, en tanto que necesita de 3 valores decimales exactos, incluidos aquellos 0 que no son significativos. Para ello, en la función se hace uso del **formatfloat**, que nos permite fijar el número de decimales que se van a incluir cuando pasemos el valor real a string. Adicionalmente, debido a la necesidad de establecer el encolumnado específico y correcto, he hecho uso de dos funciones a las cuales he denominado EncDcha y EncIzq, las cuales he encontrado en una página web de internet y he incluido también en la librería biotools.

Tras la llamada a la función **WritePDB**, he hecho uso de un bucle con **Memo2.Lines.Add()** para ir añadiendo las líneas ATOM cada vez que el ID del átomo del residuo en el que nos encontramos sea un carbono alfa (CA). Por último, he añadido un **TSaveDialog**, gracias al cual el usuario, en este caso yo, va a poder guardar las líneas ATOM como un archivo que se va a emplear luego para el siguiente paso de la actividad.

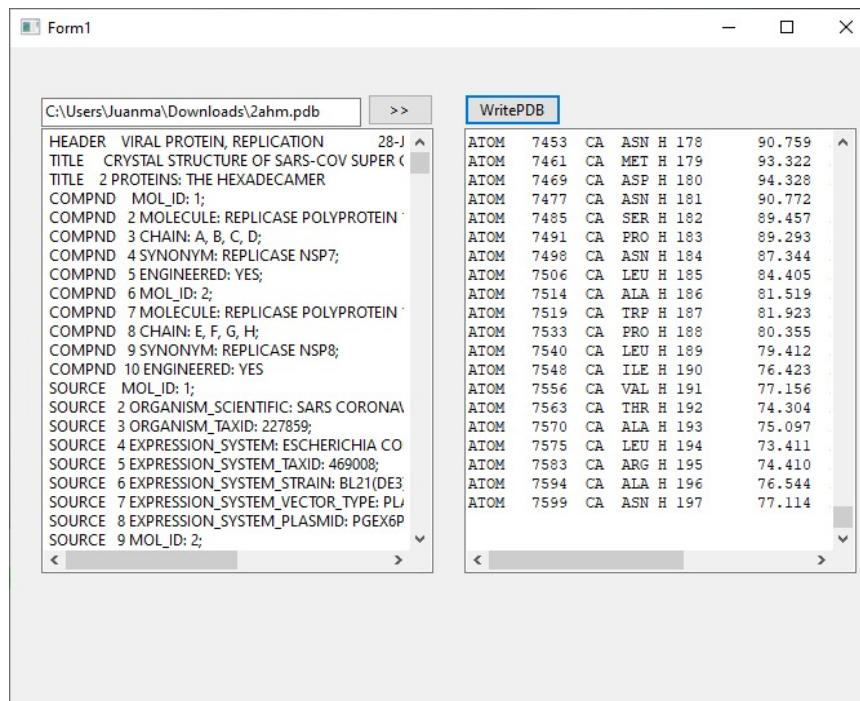


Figura 1. Prueba del funcionamiento de la función WritePDB.

Como se puede observar, el encolumnado en Courier New en el Memo es perfecto, lo cual nos indica que las funciones de EncDcha y EncIzq funcionan adecuadamente. También podemos apreciar que en cada línea ATOM tenemos todos los datos de interés que en un principio queríamos obtener. En el segundo Memo podemos movernos gracias a las ScrollBars insertadas y acceder a básicamente cualquier línea ATOM de los C alfa que deseemos. Además, con motivo de guardar dicho archivo, y como he explicado anteriormente, se ha añadido un TSaveDialog, cuya función se ejecuta justo después de que termine de cargarse el segundo memo, y que tiene una pinta tal que:

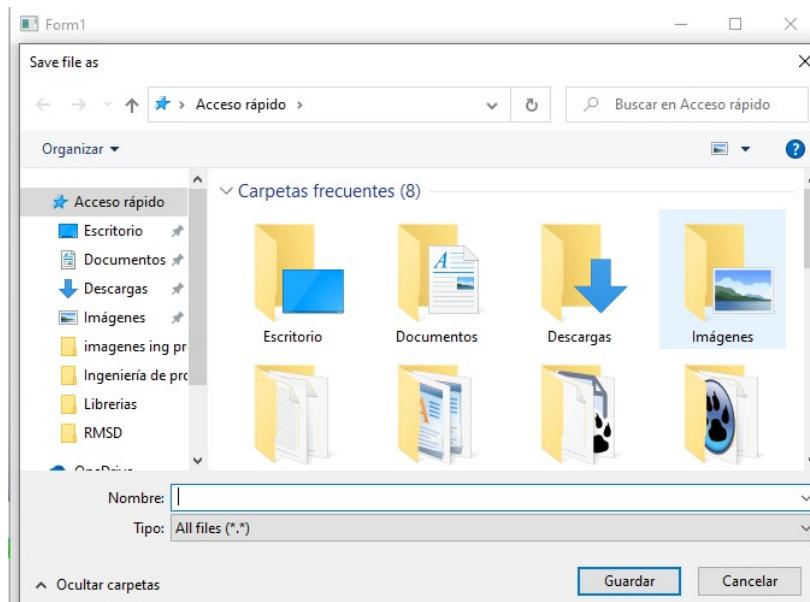
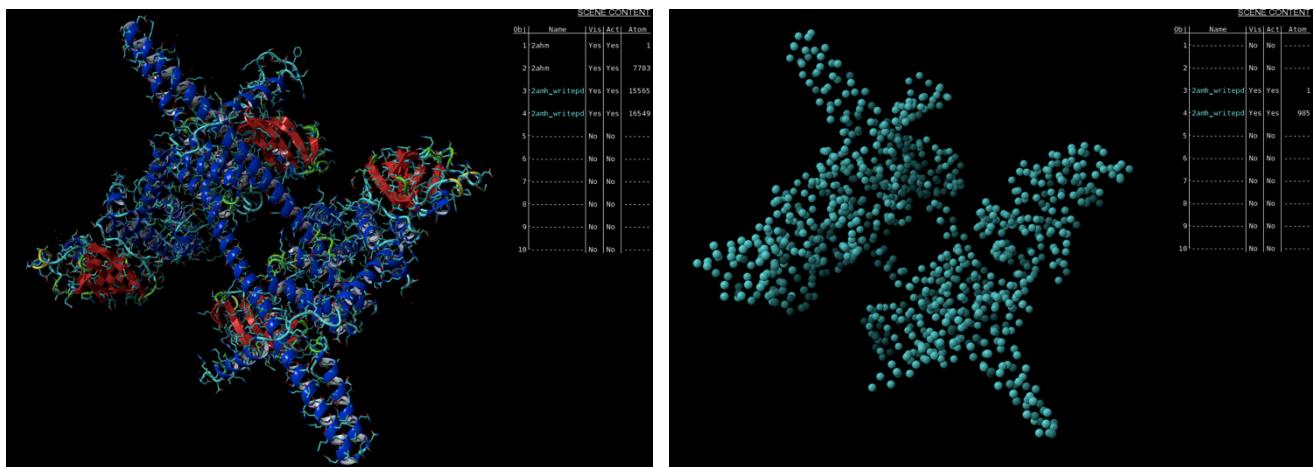


Figura 2. Muestra del funcionamiento del TSaveDialog.

Una vez hemos hecho una copia de nuestro PDB y posteriormente guardado el nuevo PDB de la proteína que únicamente contiene las líneas de ATOM de los carbonos alfa de la proteína, hacemos la prueba de abrir con el visor YASARA dicho fichero PDB.

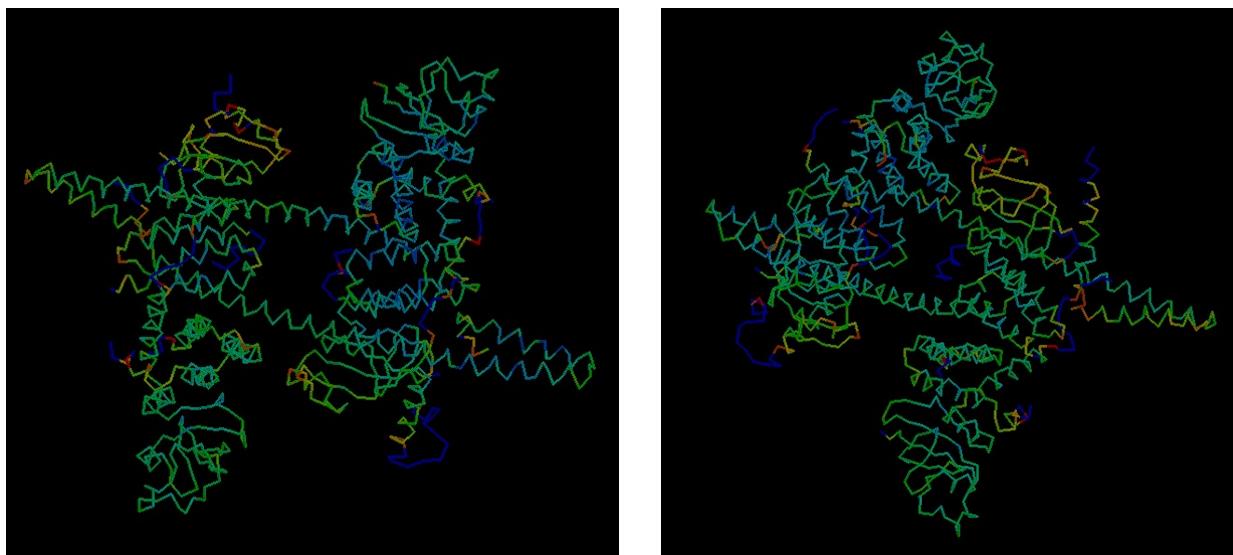


Figuras 3, 4. En la primera imagen (izquierda), se observa la apariencia del hexadecámero a partir de su fichero PDB inicial. En la segunda imagen (derecha), vemos la carga del fichero WritePDB.

Como se puede concluir a través de las imágenes, la función lleva a cabo su cometido razonablemente, debido a que la representación de la misma presenta al esqueleto de carbonos alfa, y este tiene a su vez una estructura visual a grandes rasgos que se corresponde con la que los mismos describen en la proteína con todos sus átomos representados al completo desde el primer PDB.

El factor de temperatura o factor B es un factor que describe el desplazamiento de las posiciones atómicas con respecto a un valor de referencia que es la media, de modo que aquellas zonas que tienen mayores valores de factor B se corresponden con regiones que tienen una alta flexibilidad en su estructura tridimensional. En este caso el programa usado para el análisis del factor de temperatura es RASMOL, en el cual el factor de temperatura elevado está asociado al color rojo, y un factor de temperatura bajo está asociado al color azul, presentándose colores graduales entre ambos para los factores que son intermedios.

Como sabemos, todas las proteínas tienen cierto grado de flexibilidad inherente el cual les permite llevar a cabo su actividad catalítica gracias a la interacción con otras moléculas o proteínas presentes, y en la mayoría de las ocasiones esta interacción tiene lugar en aquellas regiones que tienen un mayor grado de flexibilidad, aunque esto no ocurre en el 100% de los casos. Sin embargo, este hecho hace que resulte muy interesante el estudio de la distribución de factores de temperatura en la estructura tridimensional de una proteína.



Figuras 5, 6. Imágenes desde distintos puntos de referencia del esqueleto peptídico de 2AHM, representado por colores en función del factor de temperatura.

Atendiendo a las figuras anteriormente presentadas, podemos observar los diferentes perfiles en función del factor de temperatura. Tras recoger aquellos residuos que representan las regiones rojas, es decir, aquellos con un factor de temperatura mayor, he encontrado en común que la mayoría de ellos se encuentran en la subunidad G, y son: Met 179, Ser 175, Asp 168 y Gln 162. Si bien es cierto que hay otros residuos sueltos en otros extremos del hexadecámero con factores de temperatura altos, estos al encontrarse agrupados pueden darnos información importante acerca de la funcionalidad de la proteína. Por ejemplo, esta región debido a sus factores de temperatura sería una región más flexible, la cual podría estar implicada en el proceso de contacto en una interacción con el ARN del virus.

Para comprobar si esto es así, es necesario llevar a cabo un análisis de la bibliografía existente, y comparar si dicha región coincide con algún centro regulador o catalítico. En un principio, podría tratarse debido a su localización y a la subunidad en la que se da, de aquella región que contacta con nsp12 (RdPd), de modo que al tratarse de un contacto interproteico, sería fundamental que la flexibilidad en esta región sea notablemente mayor que en el resto del complejo proteico.

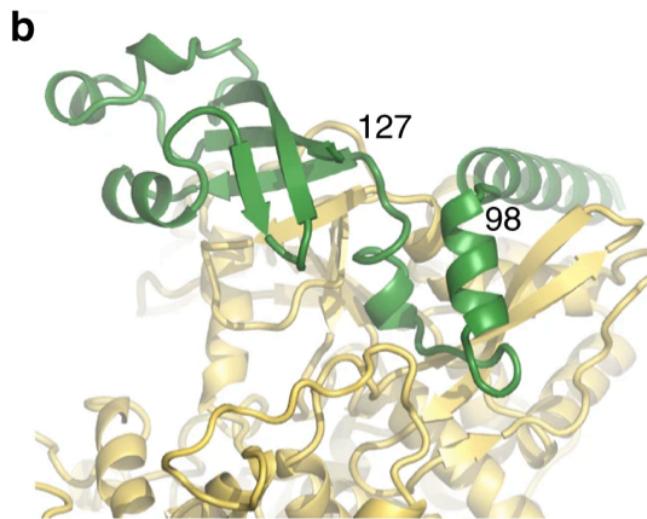


Figura 7. Una segunda subunidad nsp8 también contacta directamente con nsp12 utilizando una conformación única de los aminoácidos 98-127 de nsp8 [17].

BLOQUE IV. Predicción estructural y rediseño de proteínas

Actividad 6. Cálculo de distancias interatómicas. Ramachandran

El objetivo de esta actividad consiste en la creación de una serie de funciones en nuestra librería que permitan la automatización del cálculo de distancias entre átomos de nuestra proteína, cálculo de ángulos de enlace y de ángulos de torsión, a partir de las coordenadas de nuestra proteína obtenidas desde un fichero PDB. Todo ello se emplea en la construcción del famoso Diagrama de Ramachandran, un diagrama creado por el científico hindú del mismo nombre, que lo diseñó como representación los ángulos psi (ψ) frente a los ángulos phi (ϕ) de una proteína determinada, con el fin de llevar a cabo el análisis y visualización de las regiones espaciales energéticamente permitidas para los ángulos diedros del esqueleto de una proteína, considerando a los átomos como esferas o bolas rígidas, con dimensiones correspondientes a su radio de Van der Waals.

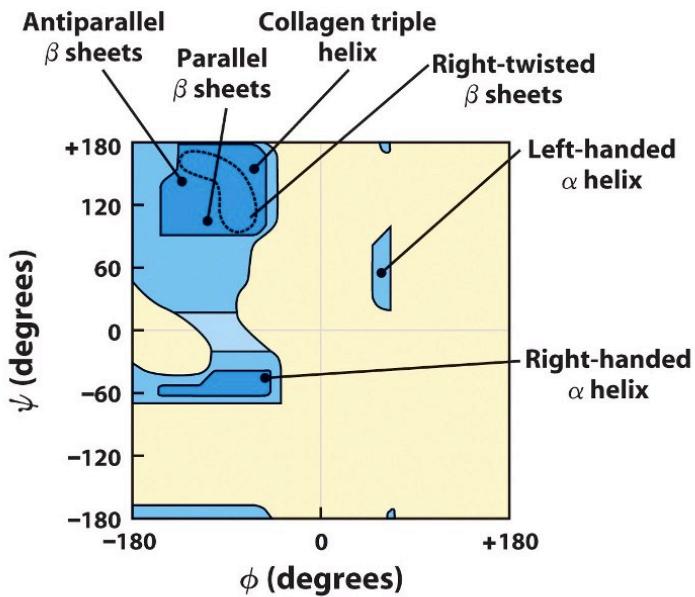


Figura1. Diagrama de Ramachandran, extraído de wikimedia.

Esta representación bidimensional nos da una idea sobre la estructura secundaria de la cual se compone nuestra proteína en cuanto a que la distribución diferencial de puntos con diferentes ángulos psi y phi en la misma está asociada a su vez a las diferentes estructuras secundarias: como podemos ver en la imagen, hay regiones en las cuales una mayor densidad de puntos implica la presencia de hélices alfa dextrógiras, levóginas, láminas beta paralelas, antiparalelas, otros tipos de hélices alfa... De modo que podemos hacernos una idea bastante clara de las posiciones más estables en cuanto a estructura secundaria en las cuales se va a organizar nuestra proteína en las tres dimensiones espaciales, cuando se encuentra en su conformación nativa. Además, si encontramos numerosos átomos los cuales se salen de dichas regiones de estabilidad estructural, podemos determinar que la estructura cristalográfica incluida en el PDB no es la mejor, habiendo zonas por tanto prohibidas, con estructuras de gran inestabilidad, las cuales serían rechazadas.

Para el desarrollo del diagrama, como hemos visto anteriormente, resulta necesaria la determinación del ángulo diedro, el cual se define como cada una de las dos partes del espacio delimitadas por dos semiplanos que parten de una arista común. Estos dos semiplanos coincidentes estarían definidos en este caso por los 4 átomos del esqueleto covalente de la proteína: N, C alfa, C carbonílico y el N del residuo siguiente. Sólo tenemos 3 tipos de átomos que determinan el enlace covalente, por lo que sólo hay 3 ángulos relevantes a la hora de determinar la estructura de la proteína: ψ , ω y ϕ . De esta forma, tendríamos definidos al ángulo ϕ como aquel ángulo de torsión entre el N y el C alfa, ψ como el existente entre el C alfa y el C carbonílico, y ω como aquel definido entre el C carbonílico y el siguiente N.

Los ángulos de las cadenas laterales, por tanto, presentan poca relevancia en la estructura secundaria o movimiento geométrico, aunque luego puedan afectar a la estructura terciaria o cuaternaria de la proteína.

Adicionalmente, la rotación es bastante limitada, debido a que el enlace peptídico tiene un fuerte carácter covalente doble (se trata de un enlace muy fuerte y rígido). Esto provoca que el ángulo ω esté bastante limitado en cuanto a sus valores, y pierda relevancia en este análisis, llevándose la restante los ángulos ψ y ϕ , como he mencionado anteriormente, especialmente en cuanto a la estructura secundaria de la proteína.

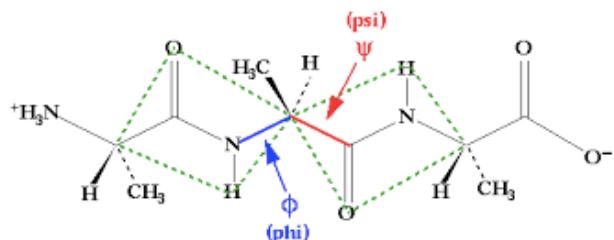
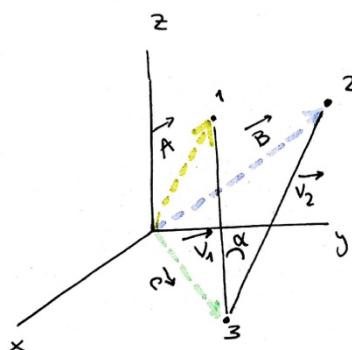


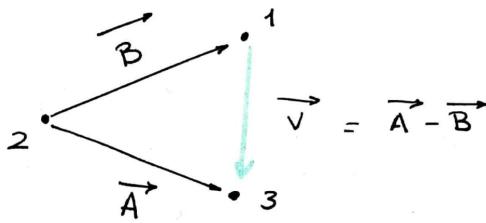
Figura2. Ángulos ψ y ϕ del esqueleto peptídico de una proteína.

Cuando hacemos uso de nuestro fichero de estructuras obtenido a partir de CargarPDB, tenemos acceso a las coordenadas cristalográficas presentes en mi proteína. Sin embargo, estas distancias son aquellas que van desde el origen en el cual confluyen las tres dimensiones del espacio. Es decir, tenemos los vectores de posición de los átomos/residuos de la proteína, pero con respecto al plano. No obstante, desconocemos los vectores que unen a los átomos entre sí, es decir, desconocemos la distancia existente entre 2 átomos cuando la referencia son ellos mismos, más allá del origen de los planos.



Es decir, en este plano conoceríamos los vectores A, B y C, pero no conoceríamos ni v_1 ni v_2 .

A pesar de desconocer dicho valor, podríamos obtenerlo fácilmente a partir de las relaciones matemáticas y en concreto vectoriales que podemos definir como, por ejemplo, el concepto del vector diferencia.



Según esto, sabido dos vectores de dirección entre el punto 1 y 2 y el punto 2 y 3, podemos conocer las componentes del vector entre el punto 1 y 3 definiendo el vector diferencia, resultante de la diferencia entre los vectores conocidos.

A su vez, para el cálculo de un ángulo entre dos puntos, se requiere de 2 vectores y 3 puntos, de modo que hacemos uso de la ecuación del producto vectorial (que luego definimos como función en biotools, prodVectorial), en la cual el resultado de la multiplicación de dos vectores se obtiene a partir de la multiplicación de sus módulos por el coseno del ángulo que se forma entre ellos, y de ahí obtenemos el ángulo.

$$\vec{u} \cdot \vec{v} = |u| \cdot |v| \cdot \cos\alpha$$

$$\alpha = \arccos \frac{\vec{u} \cdot \vec{v}}{|u| \cdot |v|}$$

Teniendo en cuenta la ecuación anterior, y la determinación de la figura del vector diferencia, podemos hacer uso de los vectores conocidos del plano para extraer el ángulo, dado que estos se relacionan por los vectores diferencia con aquellos que son desconocidos.

Por tanto, resulta completamente fundamental para el cálculo de ángulos diedros interatómicos, los cuales se van a emplear posteriormente, la elaboración de una serie de funciones en nuestra librería biotools que nos permitan integrar el cálculo de todas aquellas variables, tales como el ángulo de enlace, el módulo de los vectores, la suma vectorial, las diferencias entre vectores y el producto escalar de un vector dado. Para consultarlas, solo se tiene que acceder a la librería biotools y revisarlas con los comentarios que allí aparecen.

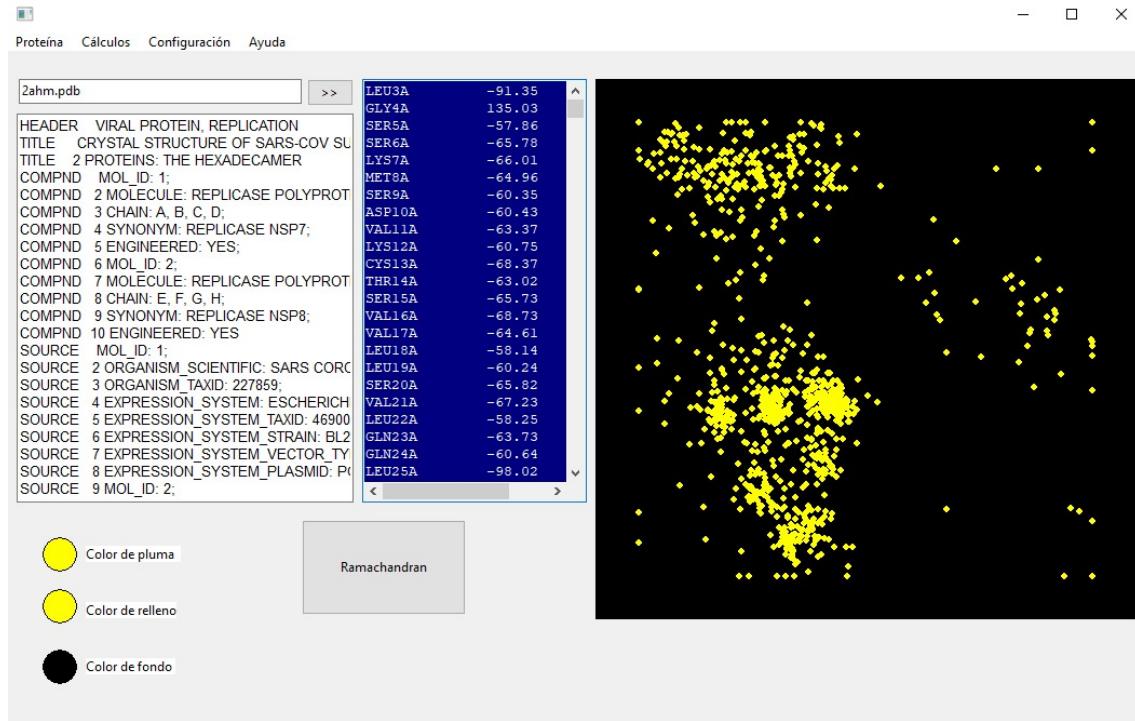


Figura 1. Resultados obtenidos con la aplicación de Ramachandran en Lazarus.

Como podemos observar en el diagrama en cuestión, tenemos una distribución de puntos bastante marcada, la cual nos muestra la gran cantidad de hélices alfa dextrógiros que presenta la estructura., además de algunas láminas beta. Gracias al uso del visor en YASARA, podemos confirmar que esto es así.

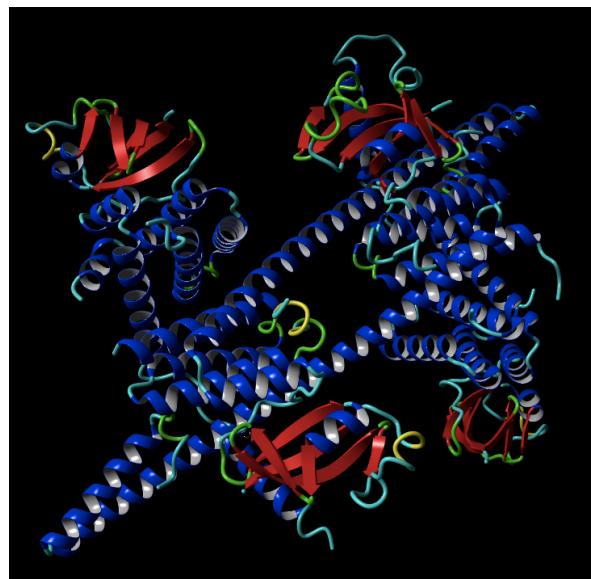


Figura 2. Imagen global de las estructuras secundarias observadas en la estructura de 2AHM, extraído de YASARA.

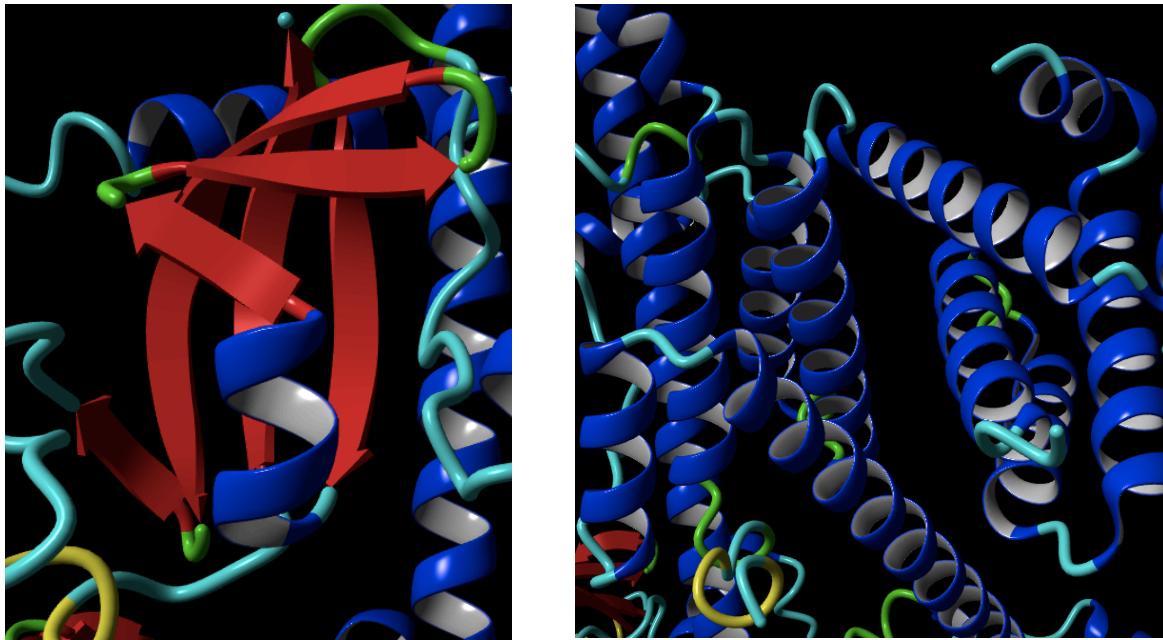


Figura 3, 4. Detalles de las diferentes estructuras secundarias de 2AHM en el visor YASARA. A la izquierda, un manojo de hélices beta antiparalelas. A la derecha, un manojo de hélices alfa dextrógiros.

Además, resulta necesario comprobar que el cálculo de los ángulos phi y psi realizados por la aplicación de Ramachandran funcionan correctamente. Para ello, es necesario comparar entre dichos valores y aquellos que pueden calcularse gracias al visor YASARA.

En primer lugar, he calculado los ángulos ϕ y ψ para un residuo de cisteína, en concreto el número 37 de la subunidad D, el cual pertenece a una alfa-hélice.

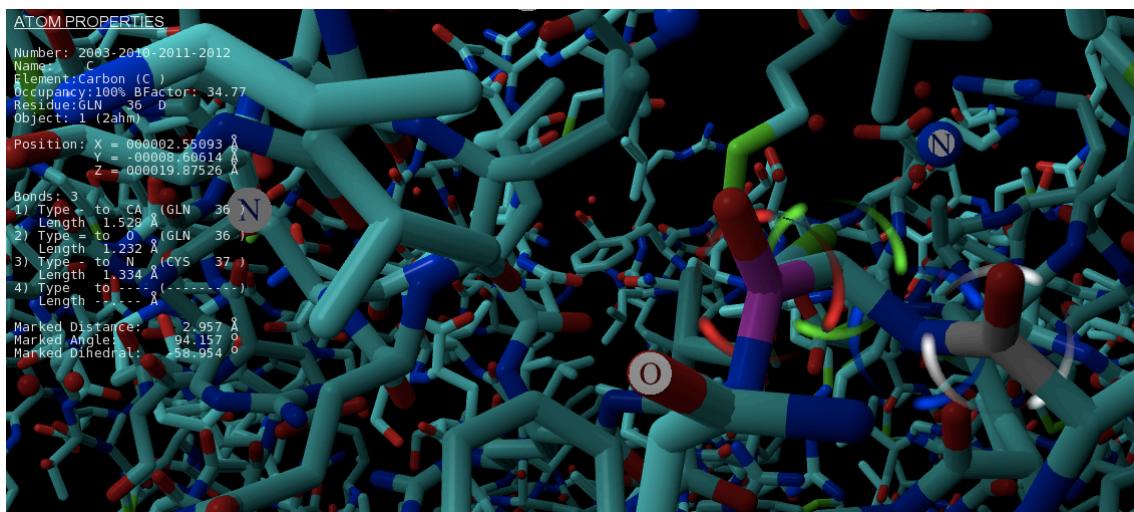


Figura 5. Cálculo de ángulos diedros de un residuo. Ángulo ϕ .

Si nos fijamos en el ángulo diedro que aparece en la imagen, tiene un valor de -58.954° .

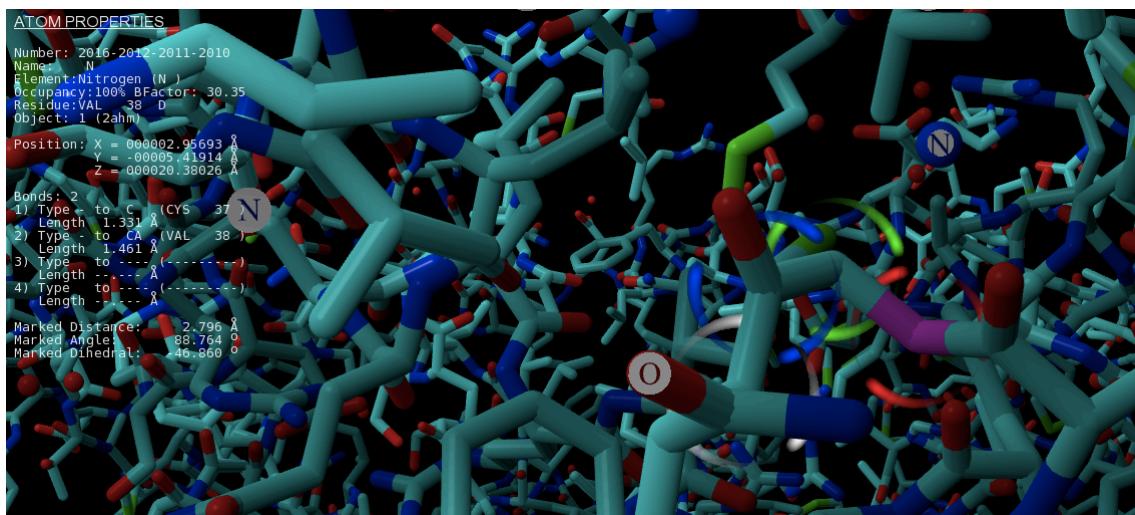


Figura 6. Cálculo de ángulos diedros de un residuo. Ángulo ψ .

En esta imagen, se recoge que el ángulo tiene un valor de -46.860° .

Teniendo ambos valores, determinamos que estos son correctos debido a que los mismos son valores aceptados como posiciones estables dentro de un diagrama de Ramachandran “original”. Si nos fijamos en el diagrama obtenido haciendo uso de la aplicación, podemos afirmar que existen puntos recogidos en la región que comprende las hélices alfa, y los ángulos descritos por este residuo/en esta hélice están recogidos. Por tanto, se puede decir que la aplicación funciona correctamente.

A continuación, he repetido dicho procedimiento de cálculo para el residuo de valina 191 de la subunidad G, perteneciente a una lámina beta.

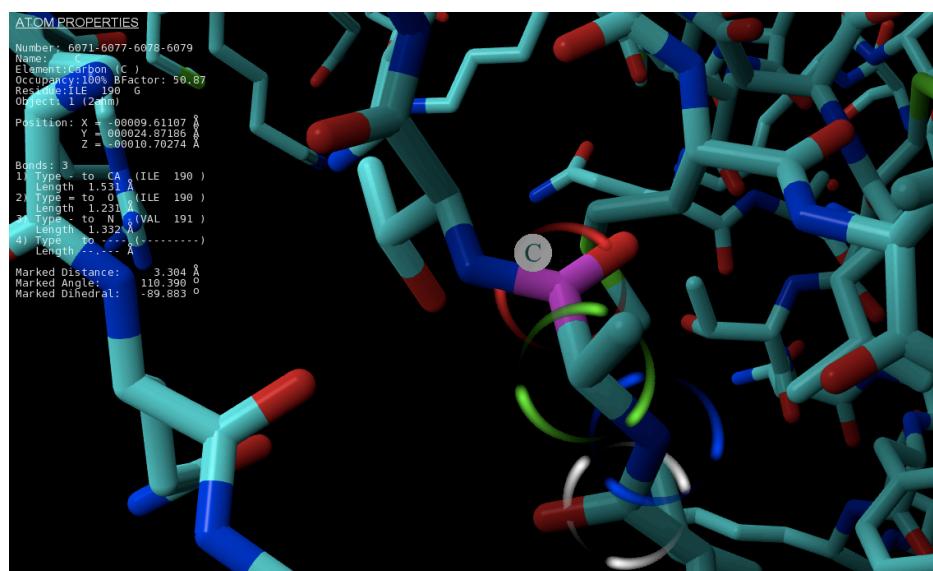


Figura 7. Cálculo de ángulos diedros de un residuo. Ángulo ϕ .

Tras seleccionar los ángulos que conforman el ángulo ϕ , se puede obtener el ángulo diedro, con un valor de -89.883° .

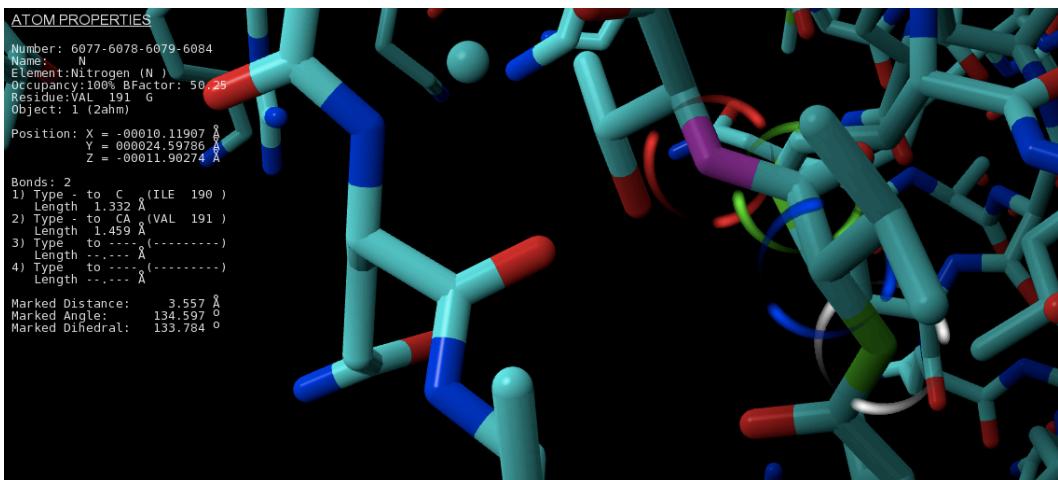


Figura 8. Cálculo de ángulos diedros de un residuo. Ángulo ψ .

Repetiendo el proceso para el ángulo ψ , obtenemos un valor de 133.784° . Teniendo en cuenta estos valores, si nos fijamos en un diagrama de Ramachandran, podemos ver que corresponden a láminas beta antiparalelas. Si hacemos referencia a nuestro diagrama de Ramachandran, encontramos que dichos valores para los ángulos están contemplados en el mismo, de modo que también podemos confirmar que el cálculo de ángulos diedros y del diagrama de Ramachandran en conjunto funcionan correctamente.

Cabe mencionar, como último apunte, que en nuestro diagrama aparecen numerosos puntos que se encuentran alejados de las zonas permitidas, es decir, en zonas prohibidas para la estabilidad de estructuras secundarias.

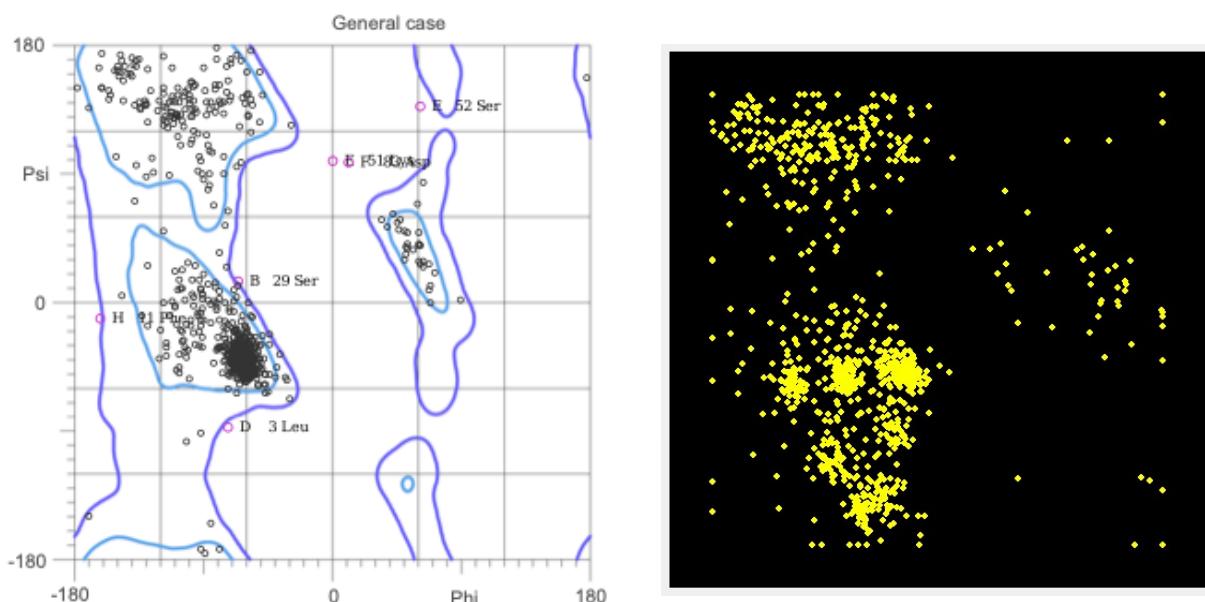


Figura 9. Comparación del Diagrama de Ramachandran para 2AHM. Imagen obtenida de MolProbity (izq), resultado de la aplicación (drcha).

Al comparar dicho diagrama realizado por MolProbity, vemos que, en cuanto a la distribución general de ángulos en el diagrama, esta es muy similar a la obtenida por la aplicación desarrollada. Sin embargo, al no estar representadas las coordenadas en nuestro gráfico, resulta difícil decir la escala del mismo, y por tanto resulta también complicado discriminar ciertos puntos, por lo que en general se puede decir que es una representación bastante fiable del original.

```
Residue name, type, SS, (phi, psi), z
LEU General L (-91.35, -24.61) -0.4630
GLY Gly L (135.03, -159.64) -0.7407
SER General H (-57.86, 176.34) -0.9704
SER General H (-65.78, -41.50) 1.3414
LYS General H (-66.01, -42.15) 0.8496
MET General H (-64.96, -40.95) 1.1756
SER General H (-60.35, -44.61) 1.2203
ASP General H (-60.43, -43.86) 1.2231
VAL Ile/Val H (-63.37, -48.15) 0.4519
LYS General H (-60.75, -41.48) 1.3412
```

Figura 10. Cálculo de los ángulos diedros para los 10 primeros residuos de 2AHM. Extraído de MolProbity.

-91.35	-24.61
135.03	-159.64
-57.86	176.34
-65.78	-41.50
-66.01	-42.15
-64.96	-40.95
-60.35	-44.61
-60.43	-43.86
-63.37	-48.15
-60.75	-41.48

Figura 11. Ángulos phi y psi extraídos del Memo2 de la actividad “Ramachandran” en Lazarus.

Por último, he incluido una tabla en la cual se muestran el cálculo de los ángulos phi y psi para los 10 primeros residuos de la proteína 2AHM. Como se demuestra, los ángulos obtenidos corresponden con valores aceptados para hélices alfa dextrógiros (desde el 2° al 10°). Los valores de los dos primeros parecen pertenecer a regiones menos definidas, lo cual puede ser debido a que usualmente los primeros residuos que se obtienen en las estructuras cristalográficas son aquellos que tienen menor importancia para la estructura

y función, y suelen ser los peor conseguidos por motivo de inestabilidad en dichas regiones marginales. Además, atendiendo a los resultados obtenidos con la actividad programada en Lazarus, podemos ver que son idénticos a los extraídos desde MolProbity, con lo que podemos una vez más confirmar que la aplicación funciona excelentemente.

Actividad 7. Esterodiagrama

La estructura tridimensional de las proteínas es una de las áreas más complejas de análisis de las mismas, y la que congrega la mayor cantidad de recursos para su determinación y estudio, precisamente debido a que en gran medida la funcionalidad de cada proteína depende precisamente de dicha estructura tridimensional.

El ojo humano es capaz de construir imágenes en tres dimensiones por el simple hecho de que entre nuestros ojos existe una separación específica que hace que dos imágenes hacia el mismo objeto lleguen al cerebro a través de nuestros nervios ópticos para ser procesadas, de modo que el objeto en cuestión es el mismo en ambas imágenes, pero este se encuentra girado unos 5° con respecto al del otro ojo. La superposición imaginaria de estas dos imágenes que no se encuentran en el mismo ángulo la una con respecto a la otra lleva a la creación de las imágenes 3D.

Es por esto por lo que, para el desarrollo de una aplicación que nos permita visualizar la estructura tridimensional de una proteína, bastaría con representar dicha proteína en dos ventanas, de modo que la representación de una de las ventanas mantendría los valores X, Y, Z provenientes del fichero PDB, mientras que en la segunda ventana se haría una representación de la proteína con todas y cada una de sus coordenadas giradas unos 5°, intentando imitar las condiciones bajo las cuales nuestro cerebro es capaz de integrar la información en 3 dimensiones, llevando todo este procedimiento a la creación de lo que conocemos como **Esterodiagrama**. De esta forma, si superpusiésemos ambas ventanas, obtendríamos la visión 3D de la proteína.

En esta actividad he buscado recrear exactamente dicho efecto, creando una nueva aplicación en la cual he añadido un botón denominado “Proteína”, el cual se encarga de la carga en el Memo1 del fichero PDB con la información de nuestra proteína, además de un Edit1 en el cual aparece el nombre de la misma. Posteriormente, he añadido un botón denominado “Esterodiagrama”, el cual se encarga de recorrer todas las fichas de la proteína en cuestión, hasta que encuentra a una fenilalanina. Una vez la reconoce por su código de 3 letras, su ID3, extrae sus coordenadas X, Y y Z y las introduce en una matriz denominada Ci. Esto se repite para los dos residuos contiguos.

Una vez almacenados en dicha matriz las coordenadas, hay un bucle que recorre desde la posición (0,0) de la misma hasta la posición más alta de la matriz, almacenando en otra matriz dinámica previamente dimensionada denominada datos, los valores X e Y de Ci, para la representación por medio de la función plot en el 1^{er} plotXY, básicamente para eliminar el eje Z en la representación, y que esta se haga en dos dimensiones. Posteriormente, se

aplica la función GiroOY a la matriz Ci, debido a que dicha función actúa “tridimensionalmente”, para crear Cf, que contiene a las coordenadas giradas 5° (considerando que 0.087 radianes se corresponden con 5°, ya que GiroOY está diseñada en radianes). Posteriormente, se repite el bucle en el cual se añade a la matriz “datos” las coordenadas X e Y, esta vez giradas, y se representan de nuevo en el TImage2, obteniendo así en cada TImage la “imitación” de lo que verían cada uno de nuestros ojos. Si se superpusieran dichas ventanas, obtendríamos la visión tridimensional de este fragmento de la proteína.

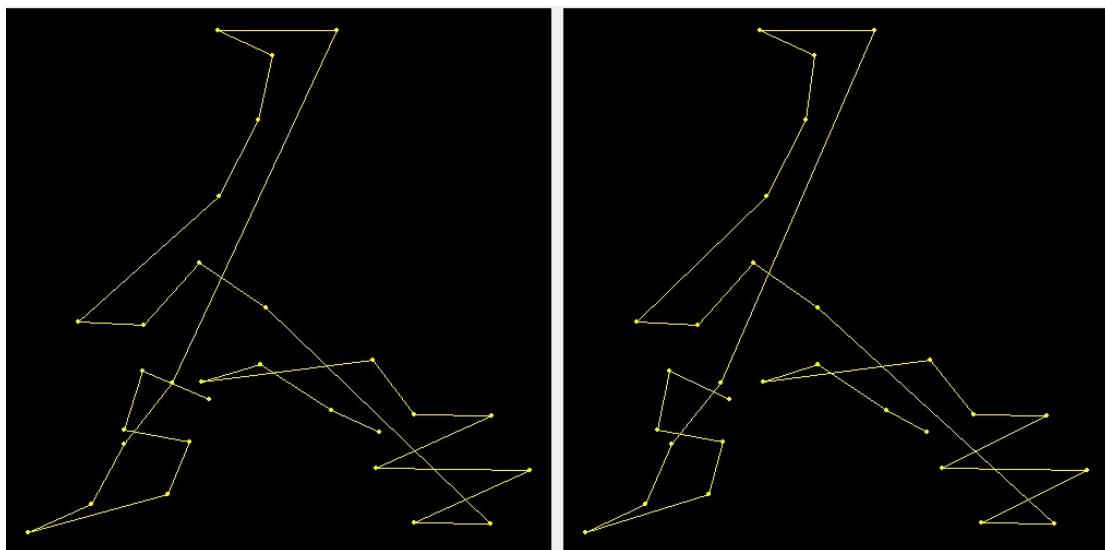


Figura1. Resultados obtenidos tras el empleo de la aplicación de Esterodiagrama en Lazarus.

Como podemos observar, pese a la sutileza del giro, es posible percibir ciertas diferencias en la orientación espacial de los átomos y el dibujo que describen. Si pudiésemos superponer dichas imágenes, tendríamos este segmento en visión 3D.

Actividad 8. Alinear el eje Z

En esta actividad, se requiere de la proyección en el eje cartesiano de las posiciones de los 10 primeros carbonos alfa de la proteína asignada, asumiendo como Z el eje perpendicular al plano del papel. A continuación, se pide que se deduzca la transformación necesaria para que los puntos 1º y último se superpongan en la nueva proyección.

Para el desarrollo de esta actividad, he hecho uso de la creación de una unidad nueva, la cual se denomina “AlinearZ”, en la cual tenemos un Form con 2 botones. El primero, como usualmente solemos hacer, está encargado de la carga del fichero PDB en el Memo1. Por otro lado, definimos a CA1 y CAN como el número de residuo inicial y el número de residuo final de la proteína,

y además, incluimos 2 SpinEdit que nos permiten seleccionar el rango de valores disponibles para el uso de la función que está acotado entre el primer y el último residuo de la proteína.

El segundo botón, denominado Alinear Z, básicamente asigna los valores seleccionados en los SpinEdit al residuo inicial y final que se quieren alinear (CA1 y CAn). Posteriormente, hacemos uso de un método de control de errores que nos permite invertir la identidad numérica de estas dos variables en caso de que la segunda sea menor que la primera. Una vez controlado esto, y habiendo creado las variables CAI (los carbonos alfa a representar en el primer TImage, previos a hacer uso de la función de alineación), CAT (que va a almacenar las coordenadas de los carbonos alfa y nos va a permitir llevar a cabo la representación a la alineación) y datos (un tipo TTablaDatos que nos permite recoger las coordenadas X e Y de los carbonos para la representación), que son básicamente arrays, y dimensionarlas. Después, se hace uso de un bucle for que permite recorrer el rango de residuos seleccionado y obtener las coordenadas del átomo C alfa de cada uno de los residuos, y almacenarlos en CAI.

Una vez hecho esto, creamos otro bucle que nos permita recorrer desde el valor más bajo al valor más alto de CAI y obtener del mismo las coordenadas X e Y, que van a ser introducidas en datos, excluyendo por tanto los datos de las coordenadas del eje Z. Estos datos se representan en el primer plot.

Tras esto, aplicamos la función **alinear_ejeZ** a CAI, función de biotools, que hace uso a su vez de las funciones de traslación, giroOX y giroOY, convirtiéndose estos TPuntos en CAT, y repetimos el proceso anterior con las nuevas coordenadas, obteniendo una representación de los átomos alineados.

Tras ejecutar la aplicación con el hexadecámero, obtenemos lo siguiente:

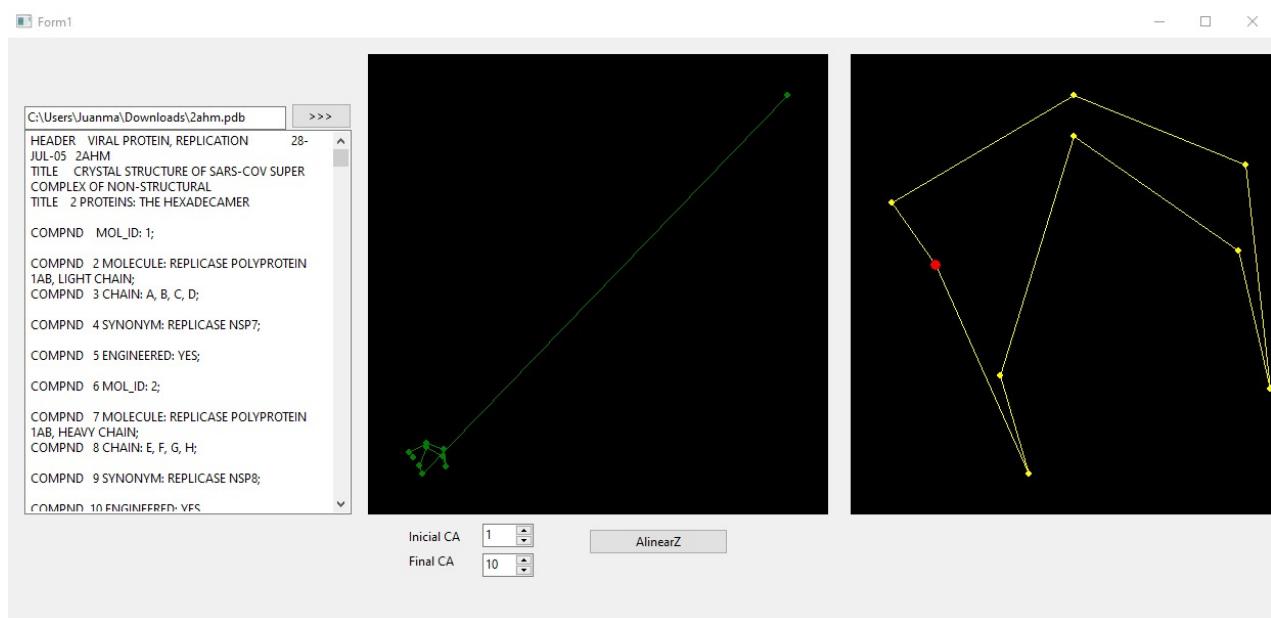


Figura1. Funcionamiento de la función *Alinear_ejeZ*.

En el gráfico TImage2, en el cual representamos CAT, observamos un punto rojo que marcaría aquel punto en el que están alineados/superpuestos los carbonos alfa, asumiendo el eje z como perpendicular al plano del papel. Si hacemos un poco de uso de la visión espacial, podemos observar que, efectivamente, la proteína se encuentra girada y alineada con el eje Z, por lo cual comprobamos que el programa funciona.

Actividad 9. Cálculo del RMSD

En esta actividad se requiere de la creación de una aplicación que permita llevar a cabo el cálculo del valor de RMSD de las 3 primeras cisteínas de nuestra proteína.

Para llevar a cabo este cálculo, y dado a la gran cantidad de cuentas repetitivas que implica, he creado una aplicación nueva en el entorno Lazarus, la cual se denomina “CalculoRMSD”.

Como pequeña introducción, el RMSD es la desviación cuadrática media de las posiciones atómicas, un valor que mide la distancia promedio entre los átomos de las proteínas superpuestas en las estructuras básicas de superposición de proteínas, y el cual en un experimento de caracterización estructural puede darnos información acerca de la validez del modelo estructural extraído.

En el desarrollo de esta aplicación, cabe destacar el uso de un Form1 en el cual he incluido varios botones con funciones diferentes:

En primer lugar, y como acostumbro a hacer, he creado un **botón 1º** el cual nos ayuda a cargar el archivo PDB, haciendo uso de la función de biotools **CargarPDB**, de forma que cargo el contenido del fichero en un Memo.

Para continuar, he desarrollado un **segundo botón**, el cual se encarga de recorrer los residuos de la proteína, identificar y almacenar en un segundo Memo aquellos residuos que sean una cisteína. En concreto, y gracias a la construcción del **bucle for**, siempre se van a almacenar las 3 primeras cisteínas de la proteína en cuestión, debido a que se añade como condiciones que cada nueva cisteína tenga un número de residuo superior al anterior, asegurándome así de que son consecutivas. Una vez estas han sido “reconocidas”, se presentan sus valores en un Memo hacia el cual está dirigida esta información.

Por último, he desarrollado un **botón 3**, el cual nos permite llevar a cabo el cálculo del RMSD a grandes rasgos. A pequeña escala, el botón consiste en un amasijo de bucles anidados y matrices de datos, los cuales permiten realizar el cálculo de distancias entre todos los átomos que forman parte de cada uno de los residuos de cisteína en cuestión, haciendo uso de la función *distancia* existente en nuestro biotools. Básicamente, consiste en introducir en una matriz de filas h y columnas k los valores resultantes del cálculo de distancias entre cada uno de los átomos de la cisteína con el resto de átomos que componen el residuo (para ello hacemos uso de las variables j e i, que

hacen referencia a los átomos del residuo en los bucles). De esta forma, pasaremos a la función *distancia* valores TPunto con los cuales pueda realizar la operación.

Una vez realizado el cálculo de las distancias internas de cada una de los residuos de cisteína, es necesario realizar la sumatoria de las distancias de cada residuo y hacer uso de la fórmula del cálculo del RMSD, la cual va a aplicarse 3 veces: una entre la primera y la segunda cisteína (RMSD1), otra entre la segunda y tercera cisteína (RMSD2) y una última vez entre la primera y la tercera (RMSD3). Una explicación con mayor detalle del código puede encontrarse en los comentarios añadidos a la unidad RMSDunit1.

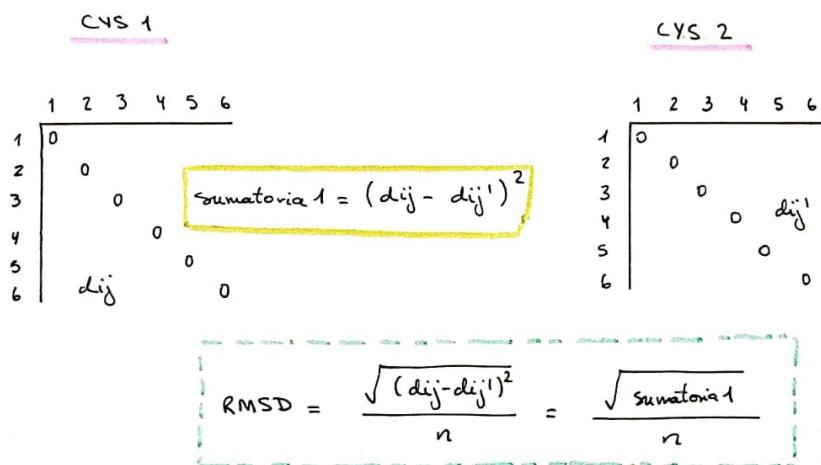


Figura1. Planteamiento del cálculo de RMSD.

En la anterior imagen se muestra una idea global sobre lo expuesto anteriormente en cuanto a la forma de llevar a cabo la construcción del botón 3.

Una vez finalizado el código, ponemos a prueba el mismo con el hexadecámero, obteniéndose los siguientes resultados:

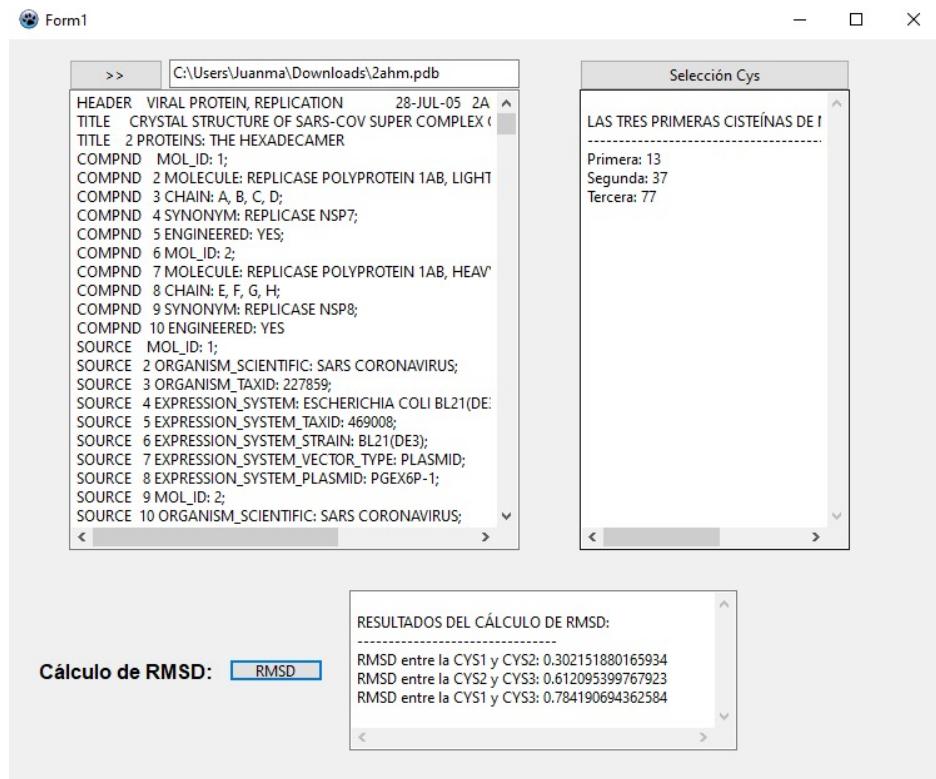


Figura2. Resultados del cálculo de RMSD por la aplicación CalculoRMSD.

Atendiendo a dichos resultados, y teniendo en cuenta la propiedad medida en el cálculo de RMSD, podemos afirmar que, debido a los bajos niveles del mismo entre estos 3 residuos, el modelo cristalográfico que proporciona el PDB es bastante fiable, puesto que la desviación cuadrática media de los átomos es muy baja, es decir, el modelo se acerca bastante a la realidad de la estructura tridimensional o nativa de la proteína. Además, los valores de RMSD más altos se encuentran entre la cisteína 2 y la cisteína 3, y entre la cisteína 1 y la cisteína 3. Esto puede significar que la cisteína 3 es aquella que tiene una desviación mayor de entre las tres, contribuyendo con una componente mayor a la desviación cuadrática media.

Actividad 10. Mutación con modificación de un aminoácido

En esta actividad se nos plantea el análisis del efecto de una mutación en uno de los residuos de nuestra proteína. En concreto, se trata de una mutación que afecta a una fenilalanina de nuestra elección presente en la secuencia, la cual pasa a ser una tirosina (PHE – TYR).

De una forma algo más visual, podemos observar que, al fin y al cabo, una tirosina no es más que una fenilalanina a la que se ha añadido un grupo OH, ningún otro cambio sucede en el aminoácido:

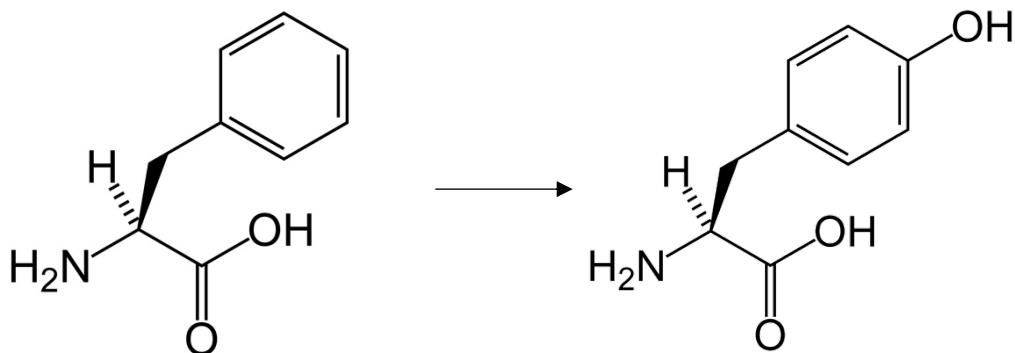


Figura1. Comparación entre una fenilalanina y una tirosina.

Sin embargo, pese a la aparente insignificancia de este cambio, como ya sabemos, el mismo puede tener consecuencias dramáticas en el entorno del aminoácido. Y es que, dependiendo del lugar de la proteína en la cual se sitúe esta fenilalanina inicial, la mutación puede implicar cambios en las interacciones con el medio al que se ve expuesta la superficie de la proteína en la cual dicho residuo se encuentra, o afectar a interacciones internas de la misma, debido al evidente cambio en la hidrofobicidad del residuo, que pasa de ser hidrófobo a ser hidrofílico. El objeto de esta actividad es, por tanto, en primer lugar, la determinación y caracterización en el fichero PDB de los parámetros que definen al nuevo grupo introducido en la mutación Mut01, y, en segundo lugar, la visualización en una aplicación del tipo YASARA que nos permita entender los cambios que se van a producir en el entorno del residuo, si es que los hay, debido a que esto dependerá de la posición y situación espacial del residuo previo a la mutación.

En este caso, el cambio o mutación se ha realizado en la fenilalanina 54 de la subunidad A del hexadecámero en cuestión.

Para el primer paso, es necesario determinar las coordenadas del grupo OH en el plano, lo cual es posible gracias a las relaciones matemáticas directas que existen entre el vector que va desde el CZ terminal del anillo aromático al grupo OH y otros vectores pertenecientes a la fenilalanina que se incluyen en el plano. El vector CZ-OH posee, como todo vector que se precie, módulo, sentido y dirección. Para obtener el **módulo** de dicho vector, se ha calculado la media de los módulos de los vectores CZ-OH de 3 tirosinas presentes en la proteína en posiciones azarosas, que nos permitan hacernos una idea de cuál es el valor usual de dicho módulo.

Sin embargo, el vector que necesitamos calcular, v, posee como todo vector dirección, módulo y sentido. En cuanto a la obtención de la dirección y el sentido del mismo, podemos inferirla a partir de la relación que existe entre el vector entre CZ-OH y el vector CB, considerando las condiciones de coplanaridad del anillo y los enlaces, de modo que obtenemos con ellos un vector, en este caso denominado D, existente entre dichos átomos (CB y CZ). A partir del mismo, calculamos su módulo y obtenemos su vector unitario y asumiendo que su dirección y sentido son los mismos que el del hipotético

vector CZ-OH (principio pre establecido de coplanaridad). Por último, para obtener el vector v, sólo sería necesario multiplicar el vector unitario de D por el módulo de v obtenido a partir de la media de módulos de los vectores preexistentes.

Una idea algo más visual de este razonamiento matemático se incluye a continuación:

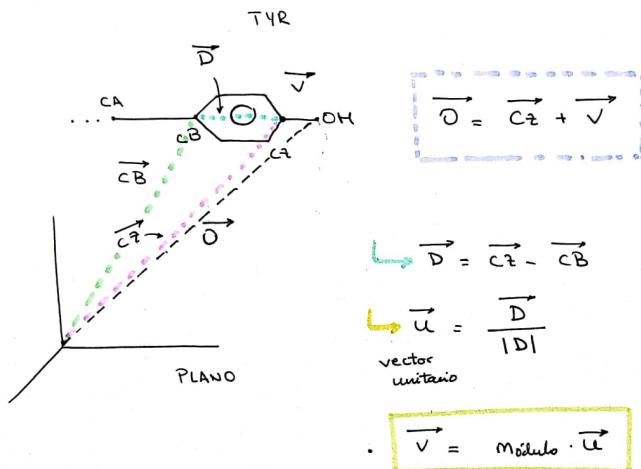


Figura 2. Planteamiento matemático.

Todo el procedimiento ha sido desarrollado en una hoja de cálculo de Microsoft Excel:

	TYR ref.	Coor X	Coor Y	Coor z	Módulos
TYR1	Cz	48.646	30.541	141.398	
	O	48.897	30.988	142.677	1.37791545
TYR 2	Cz	49.143	86.376	153.881	
	O	49.796	87.548	153.747	1.34831339
TYR 3	Cz	49.485	85.391	146.692	
	O	49.334	86.332	145.707	1.37058637
					Media = 1.36560507

Fenilalanina	Mutación	Coor X	Coor Y	Coor Z
PHE A 54	TYR A 54			
	CZ (v)	72.926	68.570	140.763
	CB (v)	70.801	71.161	138.073
	D (v)	2.125	-2.591	2.690
	V unitario	0.4945204	-0.6029658	0.62600464
	V (v)	0.67531956	-0.8234132	0.85487511
	O (v)	73.601	67.747	141.618

$$\text{ModuloD} = 4.29709274$$

Figura 3. Hoja de cálculo de Excel con todas las operaciones.

Las componentes del vector parecen tener sentido incluso previamente al tratamiento de los datos, debido a que son valores similares a los que tenemos en los vectores CZ y CB, y al estar considerando la coplanaridad, parecen resultados congruentes.

$$\overrightarrow{O} = (73.601x, 67.747y, 141.618z)$$

Una vez obtenidas las componentes del vector O, es necesario que las mismas sean añadidas a mano al fichero PDB, como último elemento en los átomos del residuo modificado de modo que, al cargar los datos en un visor de proteínas, en este caso YASARA, aparezca representado el nuevo átomo.

Tras revisar ambos ficheros, obtenemos como resultado:

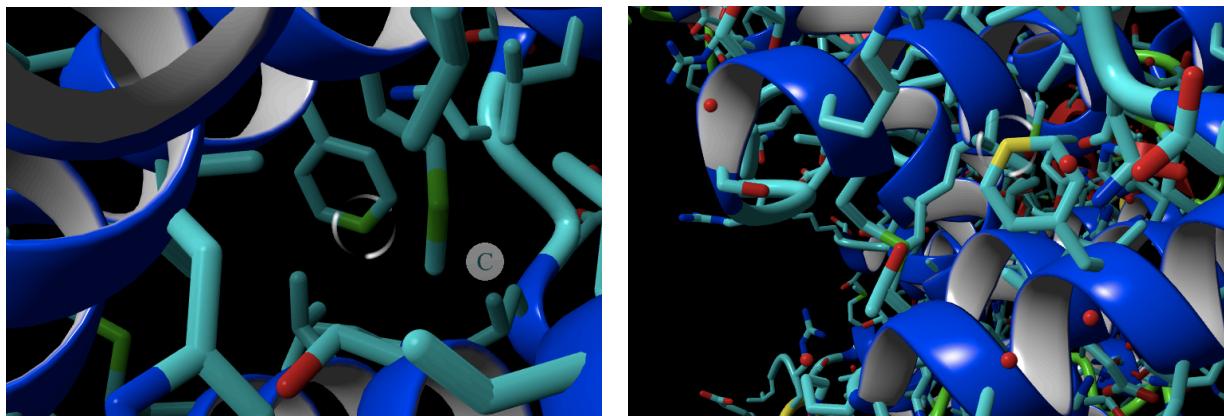


Figura 4. Capturas obtenidas desde YASARA sobre el átomo CZ de la PHE 54, subunidad A de 2AHM. No mutado.

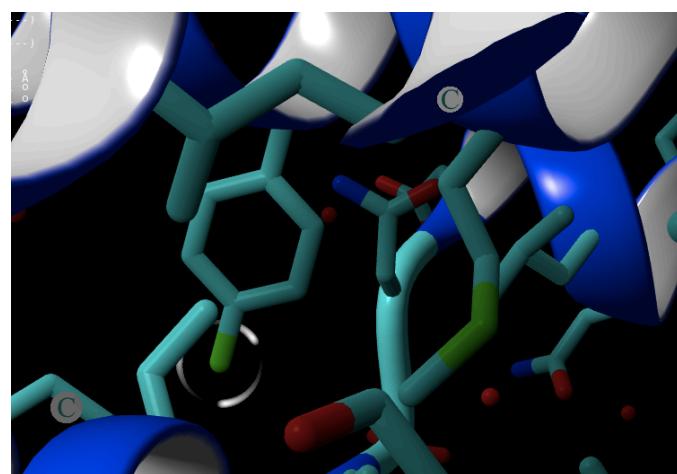


Figura 5. Capturas obtenidas desde el visor YASARA, sobre el OH introducido tras la mutación, 2AHM Mut01.

Como se aprecia en las figuras, la fenilalanina seleccionada se encuentra en una localización más bien externa, orientada hacia fuera de la hélice y de la proteína en general. Al cambiar un aminoácido que antes era apolar a uno polar, el cual se encuentra orientado más o menos hacia el solvente, pero también hacia una hélice colindante, puede ocurrir que se de una repulsión de cargas entre ambas hélices y estas se separen un poco entre sí. Además, por otro lado, podrían aumentar y favorecerse las interacciones electrostáticas entre los iones del disolvente en el cual se encuentra la proteína y esta región, y además cabría la posibilidad de que se diese la formación de puentes de hidrógeno con otros residuos colindantes o con las moléculas de agua, lo cual a su vez tendría especial impacto en la estabilidad de la proteína, que aumentaría.

Actividad 11. Predicción de enlaces disulfuro

En esta aplicación se nos pide el desarrollo o diseño de una aplicación la cual permita predecir o asignar los posibles enlaces disulfuro presentes en una proteína a partir de los datos estructurales que podemos obtener de la misma desde el PDB.

Como principio para ser capaces de asignar la presencia de enlaces disulfuro, debemos tener en cuenta ciertas premisas.

El enlace covalente, en primer lugar, se da únicamente entre los átomos de azufre presentes en los residuos de cisteína en las proteínas. Estos enlaces son enlaces covalentes fundamentales para la estructura de estas macromoléculas, y son de los pocos enlaces cruzados que permiten que la estructura tridimensional de las mismas no sea lineal. Es más, la estabilidad de la proteína depende de ello, pues el enlace covalente formado entre los átomos de azufre es altamente energético (con una Energía libre de Gibbs de unas 100 kcal/mol). La pregunta que nos planteamos es la siguiente: ¿qué cisteínas de nuestra proteína se encuentran formando un puente disulfuro? Y es ahí donde empieza el proceso de predicción.

Para esta predicción se formula la siguiente hipótesis: si las cisteínas en la estructura tridimensional se encuentran a una determinada distancia, entonces estarán formando enlaces disulfuro. A pesar de que también deberíamos tener en cuenta los ambientes oxidorreductores, sólo vamos a centrarnos en las distancias, debido a que poseemos las numerosas y poderosas funciones del cálculo de distancias interatómicas. Para todo esto, vamos a establecer un umbral de distancia, el cual recogerá un valor máximo y un valor mínimo a partir del cual las fuerzas que actúan son principalmente las de van der Waals o los enlaces covalentes. Este umbral para el grupo tiol se encuentra entre 2.0 y 2.5 angstroms, y nos va a permitir discriminar entre el enlace disulfuro y otro tipo de interacciones interatómicas e interaminoacídicas.

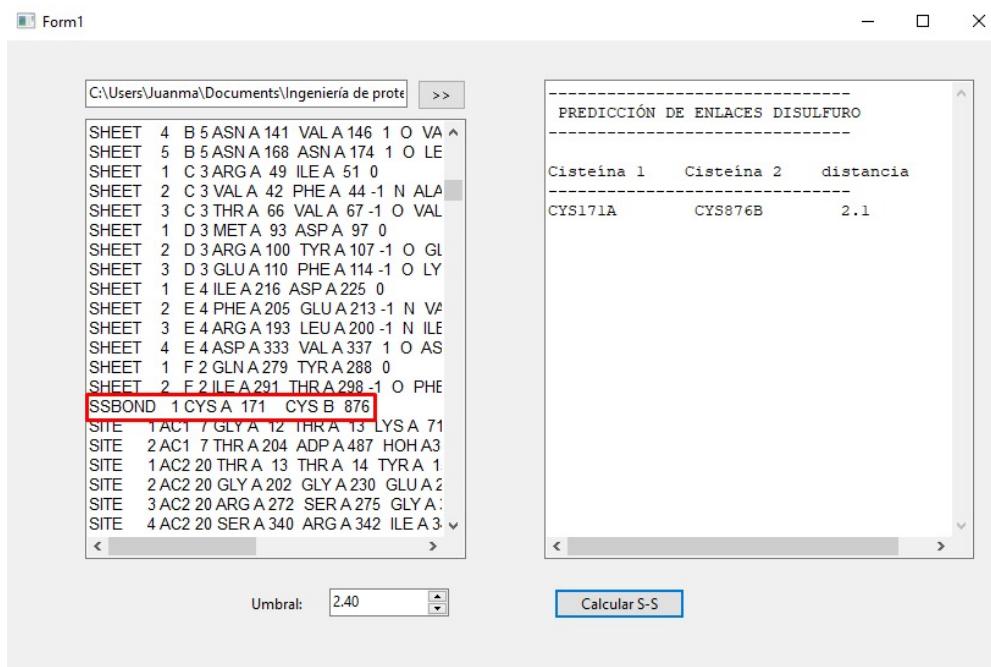
En mi caso he creado una nueva aplicación de Lázarus, la cual se denomina Enlaces_SS, en cuya unidad incluyo, como usualmente, un botón 1 que permite cargar el fichero con los datos PDB de la proteína en un Memo1. Además, añadimos un botón 2 en el que definimos en primer lugar como variables

“listaSG”, una lista que va a contener todos los enlaces, tabla (del tipo TTablaDatos), que va a crear una tabla o array con las distancias, y “numSG”, que va a funcionar en los procedimientos como. Un contador de grupos tioles SG, que son aquellos que nos interesan. Una vez dichas variables se encuentran definidas, gracias al uso de los procedimientos denominados Buscar_SG, Calcular_distancia, y Mostrar_resultados, todos ellos en la unidad principal del proyecto, llevamos a cabo el cálculo de las distancias entre grupos SG y cargamos los datos obtenidos al Memo2 que tenemos en el Form.

Cabe destacar como novedad el uso de un SpinEditFloat, que es lo mismo que un SpinEdit pero acepta decimales. Además, en el Memo llevamos a cabo el encolumnado haciendo uso de padright y padleft/de unas funciones de encolumnado encontradas en una página web de programación en Pascal: <https://webprogramacion.com/156/pascal/alinear-cadena-por-la-derecha.aspx>

Para evaluar la efectividad del programa, he usado el mismo para el cálculo de enlaces disulfuro de algunas proteínas, para luego comparar con la información bibliográfica de la misma y evaluar si tenemos falsos positivos o falsos negativos. En primer lugar, he empleado mi proteína asignada. Posteriormente, he aplicado el algoritmo a las proteínas ejemplo 2QWQ (1), 2ZUP (2), 9WGA (32), 1BHS (0), 1AFD (6), el hexadecámero, 2AHM (0), Y 1AHI (0), siendo los números entre paréntesis el número de enlaces disulfuro obtenido por la aplicación predictiva.

- Resultados para 2QWQ:



- Resultados para 9WGA:

Form1

C:\Users\Juanma\Documents\Ingeniería de prote			>>
SSBOND	12 CYS A	121 CYS A	126
SSBOND	13 CYS A	132 CYS A	147
SSBOND	14 CYS A	141 CYS A	153
SSBOND	15 CYS A	146 CYS A	160
SSBOND	16 CYS A	164 CYS A	169
SSBOND	17 CYS B	3 CYS B	18
SSBOND	18 CYS B	12 CYS B	24
SSBOND	19 CYS B	17 CYS B	31
SSBOND	20 CYS B	35 CYS B	40
SSBOND	21 CYS B	46 CYS B	61
SSBOND	22 CYS B	55 CYS B	67
SSBOND	23 CYS B	60 CYS B	74
SSBOND	24 CYS B	78 CYS B	83
SSBOND	25 CYS B	89 CYS B	104
SSBOND	26 CYS B	98 CYS B	110
SSBOND	27 CYS B	103 CYS B	117
SSBOND	28 CYS B	121 CYS B	126
SSBOND	29 CYS B	132 CYS B	147
SSBOND	30 CYS B	141 CYS B	153
SSBOND	31 CYS B	146 CYS B	160
SSBOND	32 CYS B	164 CYS B	169

Umbra: 2.40

CYS103A	CYS117A	2.0
CYS121A	CYS126A	2.1
CYS132A	CYS147A	2.1
CYS141A	CYS153A	2.0
CYS146A	CYS160A	2.0
CYS164A	CYS169A	2.0
CYS3B	CYS18B	2.0
CYS12B	CYS24B	2.0
CYS17B	CYS31B	2.0
CYS35B	CYS40B	2.1
CYS46B	CYS61B	2.1
CYS55B	CYS67B	2.0
CYS60B	CYS74B	2.0
CYS78B	CYS83B	2.0
CYS89B	CYS104B	2.0
CYS98B	CYS110B	2.0
CYS103B	CYS117B	2.0
CYS121B	CYS126B	2.0
CYS132B	CYS147B	2.0
CYS141B	CYS153B	2.0
CYS146B	CYS160B	2.0
CYS164B	CYS169B	2.0

- Resultados para 2ZUP:

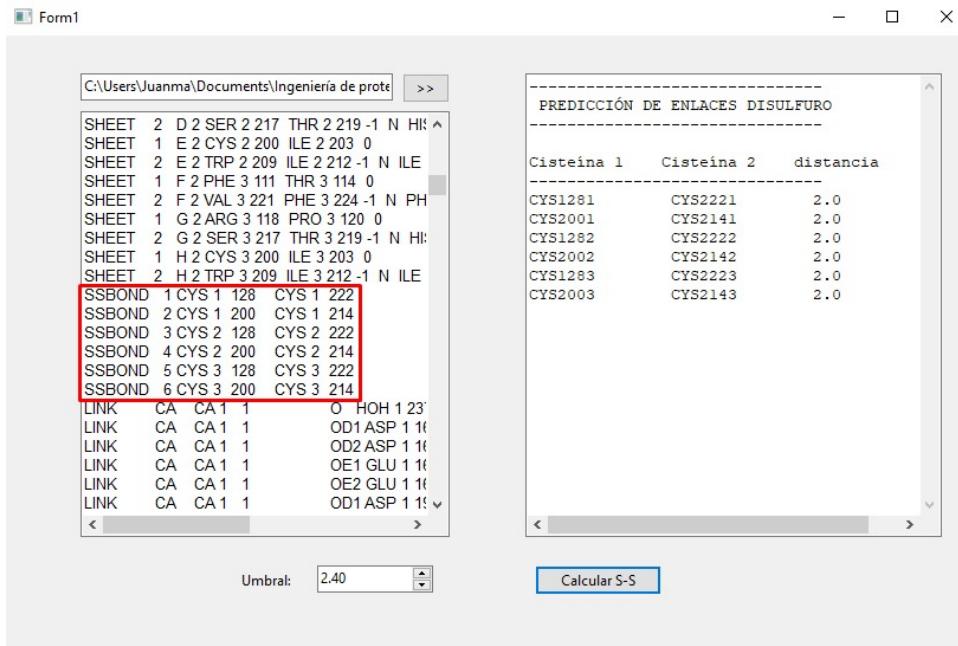
Form1

C:\Users\Juanma\Documents\Ingeniería de prote			>>
HELIX	13 13 SER B	80 LEU B	94 1
HELIX	14 14 PRO B	115 VAL B	120 1
HELIX	15 15 PRO B	143 ILE B	151 1
HELIX	16 16 ILE B	151 ILE B	162 1
SHEET	1 A 3 TYRA	9 THRA	11 0
SHEET	2 A 3 TYRA	159 LEU A	161-1 O GLU
SHEET	3 A 3 PHE	154 VAL A	155-1 O VA
SHEET	1 B 2 VAL A	22 GLU A	24 0
SHEET	2 B 2 META	56 LYS A	58 1 O THF
SSBOND	1 CYS A	30 CYS B	104
SSBOND	2 CYS B	41 CYS B	44
LINK	NE2 HIS A	41 ZN	ZNA 19
CISPEP	1 LEU A	82 GLY A	83 0
CISPEP	2 VAL A	150 PRO A	151 0
SITE	1 AC1 10 ALA B	29 GLN B	33 LYS B 3
SITE	2 AC1 10 CYS B	44 GLU B	47 ARG B
SITE	3 AC1 10 HIS B	91 LEU B	146
SITE	1 AC2 1 HIS A	41	
CRYST1	165.500 165.500	65.920 90.00	90.00
ORIGX1	1.000000 0.000000	0.000000	0.
ORIGX2	0.000000 1.000000	0.000000	0.

Umbra: 2.40

----- PREDICCIÓN DE ENLACES DISULFURO -----		
Cisteina 1	Cisteina 2	distancia
CYS30A	CYS104B	2.0
CYS41B	CYS44B	2.0

- Resultados para 1AFD:



Tras este muestreo, podemos confirmar que la aplicación funciona bastante bien, debido a que prácticamente todos los resultados se corresponden con los obtenidos en la caracterización estructural del PDB para cada una de las proteínas (verdaderos positivos). Además, efectivamente, para 1AHI y para 2AHM no existen enlaces disulfuro (verdaderos negativos), lo cual confirma el fichero PDB.

Como peculiaridad, cabe destacar que he aumentado el umbral debido a que prácticamente todas las distancias de enlace oscilan entre 2.0 y 2.1 angstroms, por lo que podemos deducir que nuestra hipótesis o premisa a la hora de la predicción estaba bien encaminada, ya que habíamos estipulado que las distancias para la formación de puentes disulfuro sería de entre 2.0 y 2.5 angstroms.

Actividad 12. Cálculo de la anfipatía e hidrofobicidad

La hidrofobicidad es la fuerza predominante en el plegamiento proteico. Se trata de una interacción por omisión, y consiste en la tendencia de las moléculas no polares a formar clatratos en una disolución acuosa para excluir a las moléculas de agua. Tal es su importancia que, a diferencia de el caso de enlaces iónicos y covalentes, en los cuales los cambios entálpicos justifican la estabilidad del enlace; en el caso de las moléculas no polares la estabilidad reside en los cambios de entropía. Esto es debido a que cuando dos moléculas se unen, el agua que rodeaba a las moléculas en dicho lugar de unión desaparece, lo que lleva a cambios de entropía, por ruptura de las estructuras o clatratos formados en dicha región.

$$\Delta G = \Delta H - T\Delta S$$

En el carácter hidrofóbico de una proteína, aquellos átomos que contribuyen en mayor medida en el esqueleto proteico son el N y el C=O, que a su vez son comunes en todos los aminoácidos que componen una proteína. Teniendo en cuenta esto, las principales diferencias de hidrofobicidad entre un residuo y otro no vienen determinadas por dichos componentes del esqueleto peptídico, sino que las determinan las cadenas laterales.

Para el estudio de la hidrofobicidad, se extrae con métodos estadísticos la posición de cada aminoácido de la proteína en cuanto a si se encuentra enterrado en el corazón hidrófobo de la misma o si se encuentra expuesto al solvente, dado a que este hecho va a tener consecuencias determinantes en cuanto a la estructura y funcionalidad posterior: para hacernos una idea, únicamente conociendo la secuencia, podríamos encontrar segmentos hidrofóbicos de la proteína y, por ejemplo, si encontramos una región en la cual la hidrofobicidad es mayor, podríamos desarrollar una hipótesis de que dicho fragmento es parte del dominio transmembrana de una proteína anclada a la membrana celular.

La medida de la hidrofobicidad es complicada, debido a que se debe evaluar a la misma en cuanto a las cadenas laterales únicamente. Es por esto por lo que necesitamos valernos de tablas o escalas de **hidrofobicidad**, que asignan un valor de hidrofobicidad a cada aminoácido, siendo una de las tablas más usadas la de Doolittle, aquella que vamos a emplear en el desarrollo de esta actividad.

Para la evaluación de la hidrofobicidad de nuestra proteína debemos construir un **perfil hidrofóbico** de la misma. Esto nos lleva al diseño de un algoritmo que sintetiza el uso de una escala de hidrofobicidad específica (en nuestro caso, la escala de Doolittle, aunque existen muchas más), y una maquinaria de suavizado del ruido para la gráfica del perfil hidrofóbico, denominada **suavizado de Savinsky-Golay**. Además, hacemos uso del método de **moving average**, de modo que en el gráfico se van abriendo pequeñas semiventanas que engloban a un residuo y los que se encuentran adyacentes (número impar de residuos, para permitir la existencia de un residuo central por cada semiventana), recorriendo y asociando picos de hidrofobicidades a residuos en pequeños segmentos de la proteína. Así, las semiventanas se van ampliando progresivamente hasta haber cubierto toda la proteína.

Este análisis resulta fundamental además debido a que, conociendo únicamente la secuencia de aminoácidos y haciendo uso de una tabla de hidrofobicidad, se puede hacer un estudio de la presencia de diferentes estructuras secundarias en la proteína (hélices α , láminas β ...), y de cuáles de ellas presentan residuos hidrofóbicos, todo gracias a la ritmidad en el patrón del perfil hidrofóbico que vamos a obtener. Todo en el sentido de que, si existen picos asociados a hélices o láminas, podremos inferir que dichas estructuras secundarias presentan un alto grado de hidrofobicidad.

- En conclusión, para el desarrollo de este algoritmo, necesitamos:
- La secuencia de aminoácidos de la proteína.

- La tabla de hidrofobicidad que vamos a emplear (que en el desarrollo lógico será un **TTablaDatos**), nuestro principal problema.
- Un tamaño de la ventana que va a consistir en un dial que cargamos en el lienzo.

Para tener una idea más clara de cómo se ha desarrollado esta actividad, puede revisarse el código en la aplicación, además de los comentarios que fui anotando sobre la función de cada uno de los botones cargados en el lienzo. Es de notar la presencia en esta actividad de una nueva estructura del Form, denominada **ComboBox**, la cual ofrece al usuario una variedad limitada de opciones a elegir, en este caso en referencia a la subunidad sobre la cual se quiere calcular el perfil hidrofóbico. Es decir, el usuario solo puede seleccionar como subunidad aquellas presentes en la proteína que se selecciona. También hemos incluido SpinEdits para modular el tamaño de la semiventana, lo cual tiene un papel fundamental en la posterior visión del gráfico.

Cabe destacar que siempre vamos a poder variar la tabla de hidrofobicidad a emplear a la hora del cálculo del perfil hidrofóbico, debido a la existencia de un botón que permite seleccionar la misma y cargarla usando la función **cargarEscala** definida en nuestra librería, que recibe como argumento a la escala en cuestión (tipo **TEscala** definido previamente).

Aplicando el algoritmo a la proteína asignada, dentro de la subunidad H, encontramos el siguiente gráfico:

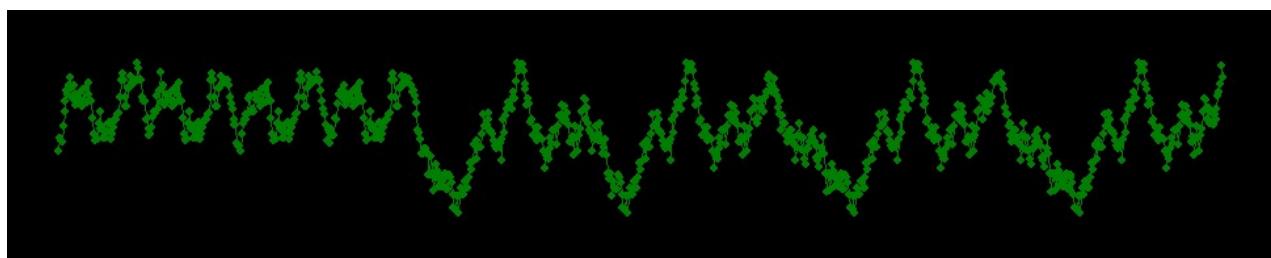


Figura 1. Resultado del cálculo del perfil hidrofóbico por la aplicación *ProfilesH*.

En esta **figura 1**, referida a la subunidad H del hexadecámero (del 1^{er} al último residuo), se nos presenta el perfil hidrofóbico haciendo uso de un tamaño de semiventana de 11. Sin embargo, tenemos la posibilidad de variar este tamaño de semiventana, de modo que la forma de nuestro perfil hidrofóbico variará drásticamente. Por ejemplo, un tamaño de semiventana de aproximadamente 2 nos permitirá evaluar la presencia de láminas β en la estructura de la proteína, y un tamaño de semiventana de 4 nos permitirá el análisis de la presencia de α hélices.

En términos generales, aquellas estructuras secundarias que presenten un alto grado de hidrofobicidad se encontrarán localizadas en el corazón hidrofóbico de la proteína, es decir, hacia el interior de la estructura globular de la misma, y aquellas que no lo hagan serán las que estén expuestas al solvente polar.

Por otro lado, debemos lidiar con el concepto de que la distribución de las hidrofobicidades en una estructura secundaria perteneciente a una proteína, como, por ejemplo, una hélice α , es desigual y puede estar lateralizada, de modo que existen regiones de las hélices con una alta carga hidrofóbica, y otras zonas con menor valor hidrofóbico, o incluso formadas por aminoácidos polares (parche hidrofóbico). Esto nos lleva al concepto de la **anfipatía**: la distribución lateral de la hidrofobicidad en un elemento secundario. Por tanto, la hidrofobicidad deja de ser un valor escalar, pasa a ser un vector, con dirección y sentido.

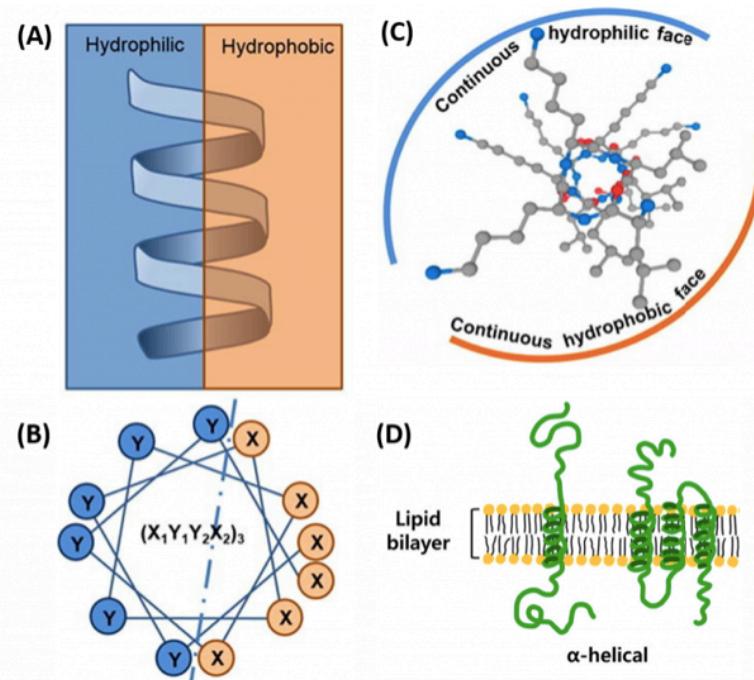


Figura 2.21 Posicionamiento de los grupos-R dentro de las estructuras de hélice alfa. Los grupos-R pueden posicionarse dentro de la hélice alfa para crear regiones anfipáticas dentro de la proteína, donde los residuos hidrofílicos se posicionan en un lado de la hélice y los hidrofóbicos en el otro, como se muestra en la vista lateral (A) o en la vista superior (B y C). Los grupos R también pueden ser totalmente hidrofóbicos dentro de las hélices alfa que abarcan la membrana plasmática, como se muestra en (D). Figura modificada desde: [Khara, J.S., et al. \(2017\) Acta Biomat 57:103-114](#) and [Ryu, H., et al. \(2019\) Int J Mol Sci 20 \(6\) 1437](#)

A lo largo del tiempo se han desarrollado diferentes modelos para la medida de la anfipatía de las proteínas, puesto que es una medida o cálculo fundamental en la predicción estructural y desarrollo de modelos de proteínas de interés: conociendo la secuencia de aminoácidos, y de la misma forma que anteriormente, usando una tabla de hidrofobicidades, podemos estudiar la presencia de diferentes tipos de estructuras secundarias en la proteína). Entre

ellas, podemos destacar dos modelos: el modelo de Eisenberg y el modelo de Stroud, los cuales con diferentes enfoques (físico y matemático, respectivamente), que al final confluyen en la misma conclusión, y que han contribuido al desarrollo de nuestro propio cálculo o algoritmo del cálculo de hidrofobicidades y anfipatía.

El modelo de Eisenberg, de forma resumida, es un modelo basado en el concepto físico del momento hidrofóbico, en el cual se veta del espacio tridimensional al eje z, de modo que se supone la hidrofobicidad de un segmento de proteína como la sumatoria de las hidrofobicidades de las cadenas laterales de los residuos que componen a dicha estructura secundaria. De este modo, orientando a, por ejemplo, una hélice α , cada una de las cadenas laterales se consideraría como un vector orientado hacia el exterior del cilindro que conforma la hélice, y cada uno de los vectores (con módulo, dirección y sentido) se encontraría distribuido con una separación angular característica de cada tipo de hélice (en el caso de las hélices α , la separación entre cada vector sería de unos 100°). Otro ejemplo sería el de la lámina beta, en la cual la separación entre los vectores sería de 180° , la máxima que se puede dar en una vuelta de la hélice que describe el cilindro imaginario que soporta dicha lógica. Así, en este modelo vamos a obtener un vector cuya magnitud de módulo va a corresponderse con la intensidad de la hidrofobicidad, y cuya dirección y sentido indican hacia dónde apunta dicha hidrofobicidad.

$$\mu_H = \sqrt{(\sum H_n \sin(\delta n))^2 + (\sum H_n \cos(\delta n))^2}$$

Ecuación del momento hidrofóbico de Eisenberg.

Como podemos observar en la ecuación, el primer término en ambos componentes de la suma es aquel referido a la hidrofobicidad asociada al residuo en el que nos encontramos, y el segundo término se encuentra asociado a la distancia angular entre los residuos, siendo la constante δ la que hace referencia al ángulo.

Por otro lado, encontramos el modelo de Stroud, el cual está basado en la parte real de la Transformada de Fourier, es decir, en el “Fourier Power Spectrum”. Basándonos en el principio básico de la Transformada de Fourier, que dicta que cualquier curva matemática que cumpla 3 requisitos (continuidad, derivabilidad y con un número n de derivadas consecutivas) se puede aproximar con una serie infinita de términos sinusoidales, es decir, podemos descomponer curvas en funciones de senos y cosenos, series de infinitos armónicos. Las representaciones de las curvas enfrentarían a la amplitud de la función armónica en el eje de coordenadas, mientras que en el eje de abscisas estaría representada la frecuencia de dicha función. De esta forma, el Power Spectrum representa en forma de puntos la amplitud frente a la frecuencia: lo que se conoce como el **espectro de potencias**, que asocia cada amplitud a una frecuencia.

Cuando tenemos curvas con muchísimos armónicos, dicho espectro va a representar muchos puntos.

Los picos en el espectro de potencias revelan la ritmicidad en una señal muy compleja. Usando esa ritmicidad, se puede inferir la estructura secundaria de la proteína. Las frecuencias que nos interesan están entre 0° y 180° , por el mismo motivo que en Eisenberg (hace falta más de 1 aminoácido para definir una vuelta de hélice).

- Si tenemos un pico de frecuencia/amplitud a 100° , es porque existe un ritmo de hidrofobicidad compatible con 100° de distancia angular, que es aquella distancia característica de las hélices alfa.
- Si tenemos un pico a 180° , sin embargo, estaremos hablando de una lámina beta.

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

Ecuación basada en la parte real de la Transformada de Fourier de Stroud

Como podemos observar, la ecuación se parece mucho a la descrita por Eisenberg, con excepción de Stroud resta a la hidrofobicidad de cada residuo la media de la hidrofobicidad de todos los residuos (que es un valor constante), lo cual tiene como finalidad eliminar el pico aberrante.

En la actividad o proyecto creado en el entorno Lázarus, se ha desarrollado un botón capaz de calcular el perfil de anfipatía de cualquier subunidad o conjunto de residuos que se quiera basándose en los modelos de Eisenberg y Stroud, de modo que se pueden hacer análisis de estructuras secundarias presentes en la proteína y la distribución de la hidrofobicidad.

Para nuestro hexadecámero:

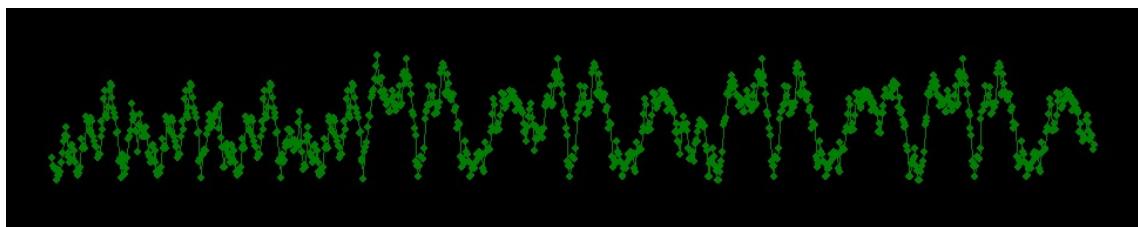


Figura2. Resultado del cálculo de anfipatía de Eisenberg para la subunidad H de 2AHM.

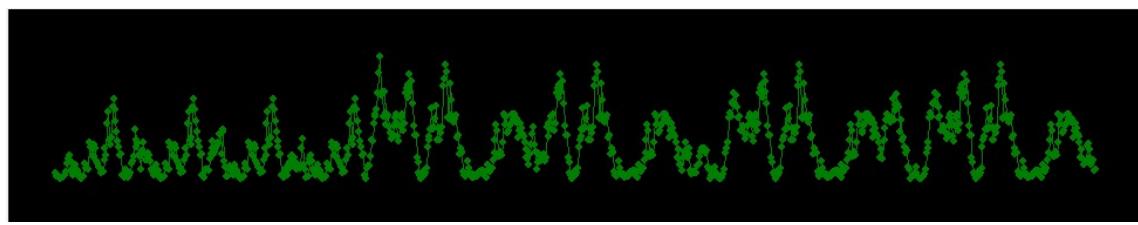


Figura3. Resultado de cálculo de anfipatía de Stroud para la subunidad H de 2AHM.

Como se puede apreciar, ambas representaciones, pese a no ser iguales, son muy parecidas.

Cabe destacar que la anfipatía y la hidrofobicidad no son conceptos equivalentes, es decir, no existe una relación directa entre ambas medidas: una estructura muy hidrofóbica no tiene por qué tener un valor de anfipatía elevado, ya que la anfipatía mide las diferencias de hidrofobicidad, no la hidrofobicidad en conjunto de nuestra proteína.

Actividad 13. Actividad complementaria.

En el intento de desarrollar algún otro procedimiento que nos permita hacer un avance en el estudio de nuestra proteína de interés, enfocado sobre todo a su secuencia nucleotídica, he desarrollado un programa muy básico en Python, en el cual hago uso de un par de funciones muy básicas para llevar a cabo la “traducción” de una secuencia de aminoácidos a proteína.

Para ello, en primer lugar, he creado una función denominada **leer_secuencia(inputfile)**, la cual permite leer y guardar el código de nucleótidos que tengamos previamente archivado en un fichero .txt

```
16
17     def leer_secuencia(inputfile):
18         '''Función que nos permite leer y cargar la secuencia desde el txt.
19         '''
20         with open(inputfile, "r") as f:
21             secuencia = f.read()
22
23         return secuencia
24
```

Figura 1. Imagen de la función `leer_secuencia()`. Extraído de la IDE Spyder.

```
22
23     def traduccion(secuencia):
24         """
25             Función que se encarga de llevar a cabo la traducción de una cadena
26             de nucleótidos a una cadena de aminoácidos.
27         """
28         tabla = {
29             'ATA': 'I', 'ATC': 'I', 'ATT': 'I', 'ATG': 'M',
30             'ACA': 'T', 'ACC': 'T', 'ACG': 'T', 'ACT': 'T',
31             'AAC': 'N', 'AAT': 'N', 'AAA': 'K', 'AAG': 'K',
32             'AGC': 'S', 'AGT': 'S', 'AGA': 'R', 'AGG': 'R',
33             'CTA': 'L', 'CTC': 'L', 'CTG': 'L', 'CTT': 'L',
34             'CCA': 'P', 'CCC': 'P', 'CCG': 'P', 'CCT': 'P',
35             'CAC': 'H', 'CAT': 'H', 'CAA': 'Q', 'CAG': 'Q',
36             'CGA': 'R', 'CGC': 'R', 'CGG': 'R', 'CGT': 'R',
37             'GTA': 'V', 'GTC': 'V', 'GTG': 'V', 'GTT': 'V',
38             'GCA': 'A', 'GCC': 'A', 'GCG': 'A', 'GCT': 'A',
39             'GAC': 'D', 'GAT': 'D', 'GAA': 'E', 'GAG': 'E',
40             'GGA': 'G', 'GGC': 'G', 'GGG': 'G', 'GGT': 'G',
41             'TCA': 'S', 'TCC': 'S', 'TCG': 'S', 'TCT': 'S',
42             'TTC': 'F', 'TTT': 'F', 'TTA': 'L', 'TTG': 'L',
43             'TAC': 'Y', 'TAT': 'Y', 'TAA': '—', 'TAG': '—',
44             'TGC': 'C', 'TGT': 'C', 'TGA': '—', 'TGG': 'W'}
45
46         proteina = "" #Inicializamos la variable proteina como una cadena vacía.
47
48         pattern = re.compile('((?:A|C|T|G){3})', re.IGNORECASE )
49
50         for codon in pattern.findall(secuencia):
51             print(codon.upper())
52
53             proteina += tabla[codon]
54
55         return proteina
```

Figura 2. Imagen de la función traducción(). Extraído de la IDE Spyder.

Por otro lado, he creado otra función, denominada **traducción()**, la cual se va a encargar como su propio nombre indica de la traducción de nuestra secuencia de ADN a la secuencia de aminoácidos de una letra de nuestra proteína de interés. Para ello, he definido lo que se conoce como un patrón en Python, el cual me permite a partir de una cadena con caracteres extraños, seleccionar/agrupar mis caracteres de interés y, además, en conjuntos de 3 letras, para generar codones. De esta forma, de entre todos los caracteres del string extraído desde el txt, los que me interesarán son A, T, G y C, las bases nitrogenadas de nuestros nucleótidos.

De esta forma, la función va a recibir como argumento la secuencia (string) que devuelve la función anterior. Una vez reciba esa secuencia, y haciendo uso de un diccionario (llamado “tabla”) en el cual he incluido cada uno de los codones del código genético, la función va a hacer uso del patrón para conseguir codones de 3 letras. De esta forma, se van a recorrer todas las letras del código de ácidos nucleicos, y se van a ir añadiendo a la cadena vacía “**proteína**” las letras que se corresponden con cada codón, en orden.

Por último, dicha función devuelve la cadena “proteína”, con la secuencia de aminoácidos en cuestión traducida.

De esta forma, para hacer funcionar dichas funciones, lo único que tenemos que hacer es descargar del NCBI la secuencia en formato FASTA de nuestra proteína, y guardar la misma en un fichero .txt, en un directorio que nos permita acceder a ella a través de python, ya que de otro modo no la reconocerá. En mi caso, al no disponer el NCBI de la secuencia de nucleótidos de mi proteína, he aplicado esta actividad a la proteína ejemplo por excelencia 1AHI, obteniendo los siguientes resultados:

```
In [120]: runfile('/Users/anaroblesfernandez/Desktop/Python /TraducionSecuencia.py', wdir='/Users/anaroblesfernandez/Desktop/Python ')
In [121]: leer_secuencia('SecuenciaNTs1AHI.txt')
Out[121]: '\nCCCTTTAAAGAGGGGACGGGGTGTTCACATAGGGGATCACACTCTCAGCCTCCCT
\nGGGAAAGTCTATGCCAGGGTACTGGAGGGAGAATACGGCGATAGTAGAACCTCGGATTAGGAGGAAC
\nAGTGTGGTTTCTGCCTGGGGCGTGGAAACACTGGACCACTATACCCCTACAGGGTCTGGAGGGTTCT
\nATGGGAGTTGCCAACCAATCCACATGTGCTTGTGGATTGGAGAAGGCATTGACTGTGTCCCTCGC
\nGGCGGCTGTGGAGGGTGTCCGGGAGTGGGGTCCGGGGCTCTTGCTAAGGGCTGTCCGGTCCCTGT
\nACGACCGAACGAGGAGCTGGCTCGCATTCGGCAGTAAGTCAGACT'

In [121]:
In [122]: secuencia
Out[122]: 'CCCTTTAAAGAGGGGACGGGGTGTTCACATAGGGGATCACACTCTCAGCCTCCCT
GGGAAAGTCTATGCCAGGGTACTGGAGGGAGAATACGGCGATAGTAGAACCTCGGATTAGGAGGAAC
AGTGTGGTTTCTGCCTGGGGCGTGGAAACACTGGACCACTATACCCCTACAGGGTCTGGAGGGTTCT
ATGGGAGTTGCCAACCAATCCACATGTGCTTGTGGATTGGAGAAGGCATTGACTGTGTCCCTCGC
GGCGGCTGTGGAGGGTGTCCGGGAGTGGGGTCCGGGGCTCTTGCTAAGGGCTGTCCGGTCCCTGT
ACGACCGAACGAGGAGCTGGCTCGCATTCGGCAGTAAGTCAGACT'

In [123]: traducion(secuencia)
CCC
TCT
TTT
TAA
GAA
GGG
GGA
```

```
In [123]: traduccion(secuencia)
CCC
TCT
TTT
TAA
GAA
GGG
GGA
CCG
GAG
GGT
GTG
TTC
CAA
CTA
TAG
GGG
GAT
CAC
ACT
TCT
CAG
CCT
CCC
GGG
AAA
GTC
TAT
GCC
```

```
TTC
CTA
AGG
GCT
GTC
CGG
TCC
CTG
ACG
ACC
GAA
GCA
GGA
GCT
TGG
TTC
GCA
TTG
CCG
GCA
GTA
AGT
CAG
ACT
Out[123]: '
PSF_EGGPEGVFQL_GDHTSQPPGVYARVLERRIRPIVEPRIQEESVVFRAVEHWTSIIPSTGCWRVMGVCPTNPHVLCFGEGIRLCPSGGLWRVLREYGVI
PLLAQAVRSLTEAGAWFALPAVSQT'
In [124]:
```

Figuras 3, 4 y 5. Demostración de la ejecución del programa diseñado. Como se puede observar, al ejecutar la función traducción(), obtenemos una lista con los codones agrupados en orden, y finalmente la salida de la función, que es la cadena de aminoácidos identificados mediante su ID de una letra.

Pese a que este programa se diseñó en un principio con la intención de realizar la traducción de nuestra proteína de interés, que es el hexadecámero, me ha sido imposible encontrar su secuencia de nucleótidos en el NCBI y en Genbank, por lo que los ejemplos están realizados con la proteína 1AHI ejemplo. Sin embargo, lo interesante del programa es que este serviría para cualquier proteína cuyo genoma esté accesible y se guarde previamente en un .txt, y permite obtener la secuencia de aminoácidos de una forma distinta a la descrita en la actividad 2.

BIBLIOGRAFÍA, CITAS Y REFERENCIAS

- 1.** Brun-Buisson C. SARS: the challenge of emerging pathogens to the intensivist. *Intensive Care Med.* 2003 Jun;29(6):861-2. doi: 10.1007/s00134-003-1823-y. PMID: 12858876; PMCID: PMC7080195.
- 2.** Woo, P. C. et al. Discovery of seven novel mammalian and avian coronaviruses in the genus deltacoronavirus supports bat coronaviruses as the gene source of alphacoronavirus and betacoronavirus and avian coronaviruses as the gene source of gammacoronavirus and deltacoronavirus. *J. Virol.* **86**, 3995–4008 (2012).
- 3.** Cui, J., Li, F. & Shi, ZL. Origin and evolution of pathogenic coronaviruses. *Nat Rev Microbiol* **17**, 181–192 (2019).
- 4.** PASTRIAN-SOTO, Gabriel. Bases Genéticas y Moleculares del COVID-19 (SARS-CoV-2). Mecanismos de Patogénesis y de Respuesta Inmune. *Int. J. Odontostomat.* [online]. 2020, vol.14, n.3 [citado 2021-02-20], pp.331-337.
- 5.** Snijder EJ, Decroly E, Ziebuhr J. The Nonstructural Proteins Directing Coronavirus RNA Synthesis and Processing. *Adv Virus Res.* 2016;96:59-126. doi:10.1016/bs.aivir.2016.08.008
- 6.** Adedeji AO, Singh K, Calcaterra NE, et al. Severe acute respiratory syndrome coronavirus replication inhibitor that interferes with the nucleic acid unwinding of the viral helicase. *Antimicrob Agents Chemother.* 2012;56(9):4718-4728. doi:10.1128/AAC.00957-12
- 7.** Kumar P, Gunalan V, Liu B, et al. The nonstructural protein 8 (nsp8) of the SARS coronavirus interacts with its ORF6 accessory protein. *Virology.* 2007;366(2):293-303. doi:10.1016/j.virol.2007.04.029
- 8.** Adedeji AO, Singh K, Calcaterra NE, et al. Severe acute respiratory syndrome coronavirus replication inhibitor that interferes with the nucleic acid unwinding of the viral helicase. *Antimicrob Agents Chemother.* 2012;56(9):4718-4728. doi:10.1128/AAC.00957-12
- 9.** PASTRIAN, S. G. Bases genéticas y moleculares del COVID-19 (SARS-CoV-2). Mecanismos de patogénesis y de respuesta inmune. *Int. J. Odontostomat.*,14(3):331-337, 2020.
- 10.** Ahn DG, Shin HJ, Kim MH, Lee S, Kim HS, Myoung J, Kim BT, Kim SJ. Current Status of Epidemiology, Diagnosis, Therapeutics, and Vaccines for Novel Coronavirus Disease 2019 (COVID-19). *J Microbiol Biotechnol.* 2020 Mar 28;30(3):313-324. doi: 10.4014/jmb.2003.03011. PMID: 32238757.
- 11.** Li, S., Li, S., Disoma, C., Zheng, R., Zhou, M., Razzaq, A., Liu, P., Zhou, Y., Dong, Z., Du, A., Peng, J., Hu, L., Huang, J., Feng, P., Jiang, T. and Xia, Z. (2021), SARS-CoV-2: Mechanism of infection and emerging technologies for future prospects. *Rev Med Virol* e2168.

- 12.** Teilum, K., Olsen, J. G. & Kragelund, B. B. Functional aspects of protein flexibility. *Cell. Mol. Life Sci.* **66**,2231–47 (2009).
- 13.** Bedford, Trevor; Hodcroft, Emma. [«Phylogeny of SARS-like betacoronaviruses including novel coronavirus SARS-CoV-2»](#).
- 14.** Raj R. Analysis of non-structural proteins, NSPs of SARS-CoV-2 as targets for computational drug designing. *Biochem Biophys Rep.* 2020 Dec 11;25:100847. doi: 10.1016/j.bbrep.2020.100847. PMID: 33364445; PMCID: PMC7750489.
- 15.** Peti W, Johnson MA, Herrmann T, Neuman BW, Buchmeier MJ, Nelson M, Joseph J, Page R, Stevens RC, Kuhn P, Wüthrich K. Structural genomics of the severe acute respiratory syndrome coronavirus: nuclear magnetic resonance structure of the protein nsP7. *J Virol.* 2005 Oct;79(20):12905-13. doi: 10.1128/JVI.79.20.12905-12913.2005. PMID: 16188992; PMCID: PMC1235862.
- 16.** Sevajol, Marion & Subissi, Lorenzo & Decroly, Etienne & Canard, Bruno & Imbert, Isabelle. (2014). Insights into RNA synthesis, capping, and proofreading mechanisms of SARS-coronavirus. *Virus research.* 194. 10.1016/j.virusres.2014.10.008.
- 17.** Kirchdoerfer, R.N., Ward, A.B. Structure of the SARS-CoV nsp12 polymerase bound to nsp7 and nsp8 co-factors. *Nat Commun* **10**, 2342 (2019).
<https://doi.org/10.1038/s41467-019-10280-3>