

ACTIVIDAD PRÁCTICA 2.3

Modelización estructural de interacciones proteína-proteína (PPI)

En los ejercicios que se proponen a continuación se estudiará como caso modelo el complejo formado entre las proteínas **savinasa** de la bacteria *B. lentus* y **BASI** de cebada (Micheelsen et al. 2008). BASI (por sus siglas en inglés: *Barley α-Amylase/Subtilisin Inhibitor*) es un inhibidor que se encuentra en la semilla de cebada y la protege de las proteasas secretadas por patógenos, como la savinasa, mediante la formación de un complejo de alta afinidad. Este complejo entre savinasa y BASI fue uno de los casos propuestos ("Target 32") en el experimento CAPRI (Janin et al. 2003), para el que los grupos participantes fueron invitados a enviar predicciones antes de que la estructura del complejo se hiciera pública (Pons et al. 2010).

La información de partida disponible durante el experimento CAPRI fue la siguiente:

- Estructura individual de savinasa: código **PDB 1svn**
- Estructura de BASI en complejo con α-amilasa: código **PDB 1ava** (cadena C)

Dado que ya está disponible la estructura cristalográfica del complejo savinasa/BASI (código **PDB 3bx1**, cadenas A y C), podemos usar este caso para evaluar la capacidad predictiva de diferentes métodos computacionales en condiciones lo más realistas posibles.

1. INCLUSIÓN DE RESTRICCIONES EN DOCKING: PYDOCK

Nota: En ocasiones, se puede incluir en los cálculos de *docking* información adicional sobre residuos potencialmente relevantes para la interacción, a partir de datos obtenidos de la literatura. En pyDock, es posible usar el módulo "*dockrst*" para definir restricciones de distancia basadas en dichos residuos, y evaluar el cumplimiento de las mismas para cada una de las orientaciones de *docking* previamente generadas. Esta información será complementaria a la puntuación energética previamente calculada con el módulo "*dockser*". De esta forma, el ranking final de las orientaciones de *docking* se hará de acuerdo a la combinación de puntuación energética y restricciones de distancia.

1.1. Identificación de residuos potencialmente relevantes para la interacción

Nota: Obviamente, el punto crítico es la identificación de información relevante para re-evaluar los modelos de docking. Esta información no es fácil de obtener, y aunque se encuentren datos en la literatura, muchas veces es ambigua. En el caso de la interacción entre savinasa y BASI, hay varios artículos que muestran datos funcionales y mutacionales de residuos que podrían ser relevantes para la interacción. A continuación, se han seleccionado textos extraídos de varios artículos. Es importante saber interpretar correctamente la información para poder definir residuos potencialmente relevantes para la interacción y así poder usarlos posteriormente como restricciones en *docking*.

Información relevante extraída de bibliografía:

En el artículo donde se describe la estructura de rayos X de la savinasa (Betzel et al 1992) se puede extraer la siguiente información:

(...)

The catalytic triad of residues is situated at the carboxy-terminal end of β II-1 (Asp32), the amino-terminal end of α C (His64) and the amino-terminal end of α F (Ser221).

(...)

Por otro lado, en otro artículo (Nielsen et al 2004) aparece la siguiente información sobre BASI:

(...)

Sequence alignment suggests that the protease binding site involves a loop between β -strands 5 and 6 which consists of the residues 85–96 (BASI numbering).

(...)

Y en otro artículo más reciente (Bønsager et al 2005) se reporta lo siguiente:

(...)

Mutants in the Putative Savinase-binding Site. Y87A BASI increased K_i and K_D 3.5- and 11-fold, respectively, whereas T89A, S93A, and E95Q, located in the same region of BASI (...), showed little change in affinity as judged from both the K_i and K_D values (...). The residue Tyr87 thus may be involved, albeit not critically, in inhibition of savinase.

Nota: Con esta información, se preparará una lista de residuos de savinasa y/o BASI potencialmente relevantes para su interacción.

1.2. Inclusión de restricciones de distancia en pyDock

Los residuos que se decidan usar para definir restricciones de distancia se pueden incluir como una nueva línea en el fichero inicial (`T32.ini`), en el campo del receptor y/o del ligando. Por ejemplo, si se encontrase como hipotético residuo relevante la arginina 36 de la cadena A, se indicaría de esta forma:

`Restr = A.Arg.36`

Nota: El campo `restr` indica a pyDock que se definirán restricciones de distancia. Cada restricción será definida mediante el nombre de cadena correspondiente al fichero PDB usado en el *docking* (deberá ser la misma que la definida en "`newmol`"), el código de 3 letras (con la primera en mayúscula) del aminoácido que corresponda, y el número de residuo tal como aparece en el fichero PDB usado en el *docking*. Si se usan varios residuos de restricción, se deben separar mediante comas, sin espacios, como en el siguiente ejemplo:

```
[receptor]
pdb      = 1svn.pdb
mol      = A
newmol   = A
restr    = A.Hid.147,A.Arg.36
```

Importante:

¡Cuidado! Este es un ejemplo hipotético para ilustrar cómo hay que incluir los residuos de restricción. En el ejercicio actual, habrá que incluir los residuos que se han identificado para receptor y/o ligando en el fichero `T32.ini` (no es necesario definir residuos para las dos proteínas; si no se encuentran residuos fiables para una de las proteínas, es preferible no incluirlos).

Una vez definidos los residuos que se establecerán como restricciones, se lanzará el comando siguiente para evaluar las diferentes orientaciones de *docking*:

```
>pyDock3 T32 dockrst > dockrst.log &
(el cálculo llevará solo unos minutos)
```

Nota: Se obtendrán dos ficheros: `T32.rst` y `T32.eneRST`. El fichero `T32.rst` mostrará el grado de cumplimiento de las restricciones definidas en `T32.ini` para todas las orientaciones de *docking* descritas en `T32.rot` (que en este caso serán las 200 orientaciones previamente seleccionadas al azar).

Importante:

El fichero `T32.eneRST` se ha generado a partir del fichero de energías `T32.ene` previamente calculado mediante el módulo "`dockser`" y el fichero de restricciones `T32.rst`. La puntuación final (columna Total) será la combinación de la puntuación energética y la puntuación en base al cumplimiento de las restricciones. Las orientaciones de docking se ordenarán de acuerdo a esta nueva puntuación combinada y su ranking se reflejará en la última columna RANK. Es interesante analizar si la inclusión de restricciones ha mejorado los resultados de docking (en un caso como este,

en el que conocemos la estructura de referencia, podemos observar por ejemplo si hay más orientaciones de baja RMSD entre las 10 mejores del ranking que antes de aplicar las restricciones).

Nota: Es importante asegurarse que el fichero `T32.rot` existente en el directorio de trabajo antes de lanzar el módulo "`dockrst`" es el que corresponde al fichero `T32.ene`, ya que este módulo usará la información de ambos ficheros.

Importante:

Una restricción de distancia definida a partir de un posible residuo de interacción se considera cumplida cuando el centro de coordenadas de su cadena lateral se encuentra a menos de 6 Å de distancia de cualquier átomo no-hidrógeno de la otra proteína, en una determinada orientación de *docking*.

Para cada orientación de *docking*, el porcentaje de restricciones cumplidas se convierte a una pseudo-energía (multiplicando por -1.0) y se añade a la puntuación energética para obtener la puntuación total.