

## ACTIVIDAD PRÁCTICA 2.5

### 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  $\alpha$ -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  $\alpha$ -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. PREDICCIÓN DE REGIONES DE INTERACCIÓN

### 1.1. Predicción de regiones de interacción mediante docking

A partir de los modelos obtenidos con pyDock para el caso de estudio actual (interacción savinasa/BASI, ver punto 3) se pueden predecir regiones de interacción en las proteínas que interaccionan, mediante el módulo "*patch*" (Fernández-Recio et al 2004; Grosdidier et al 2008). Este módulo identifica los residuos más frecuentes en las zonas de interacción de los 100 modelos con mejor puntuación de pyDock.

#### Importante:

En el caso actual, dado que no se ha aplicado pyDock a todas las orientaciones de *docking* por cuestión de tiempo, sino solo a 200 orientaciones seleccionadas al azar (ver punto 3.5), los resultados de este módulo no serán óptimos, pero es suficiente para entender su funcionamiento.

#### 1.1.1. Lanzamiento del módulo de predicción de regiones de interacción "*patch*" de pyDock

```
>pyDock3 T32 patch
```

**Nota:** Los ficheros resultantes `T32.recNIP` y `T32.ligNIP` contienen la lista de los residuos de receptor y ligando, respectivamente, con sus correspondientes valores de *NIP* (por sus siglas en inglés: *Normalized Interface Propensity*). Los ficheros `T32_rec.pdb.nip` y `T32_lig.pdb.nip` son ficheros PDB en los que la columna de factores de temperatura (o *B-factors*) ha sido sustituida por los valores *NIP*, de forma que dichos valores puedan visualizarse fácilmente en la mayor parte de programas de gráficos moleculares.

#### Importante:

Los valores *NIP* (*Normalized Interface Propensity*) para cada residuo representan su frecuencia en las zonas de interacción de las 100 orientaciones de *docking* con mejor puntuación energética obtenidas por pyDock.

Si un residuo tiene  $NIP = 0$ , éste se encuentra en las zonas de interacción de las 100 primeras orientaciones de *docking* con la frecuencia esperada al azar.

Si un residuo tiene  $NIP < 0$ , dicho residuo aparece en las zonas de interacción de las 100 primeras orientaciones de *docking* con menor frecuencia de la esperada al azar.

Si un residuo tiene  $NIP > 0.2$ , éste aparece en las zonas de interacción de las 100 primeras orientaciones de *docking* con una frecuencia relevante, e indica que está en una posible zona de interacción.

### 1.1.2. Visualización de los resultados del módulo de “patch” con ICM:

```
icm>read pdb "T32_rec.pdb.nip" # lee el fichero del receptor con los valores NIP)
icm>display a_1.1 # muestra la primera molécula
icm>color a_1.1/* Bfactor(a_1.1/*) # colorea los residuos de la molécula de acuerdo a los valores NIP almacenados en la columna de B-factors
```

**Nota:** Los residuos con valores altos de *NIP* aparecerán en rojo, y especialmente aquellos que tengan *NIP* > 0.2 predicen una posible región de interacción con la otra proteína.

En este momento se podrá evaluar si los residuos usados como restricciones en el *docking* están en la superficie predicha (mostrando dichos residuos en ‘CPK’, por ejemplo). En este caso, dado que se conoce la estructura de referencia, será posible también comprobar si la superficie predicha coincide con la superficie de interacción real del complejo (también se puede hacer lo mismo con el ligando).

**Nota:** En los casos en los que la proteína de interés esté compuesta por más de una cadena, se incluirían todas en el análisis visual.

### 1.2. Predicción de regiones de interacción mediante análisis de la energía de desolvatación: método ODA

**Nota:** Otro método para predecir posibles regiones de interacción en una proteína es el método ODA (por sus siglas en inglés: *Optimal Desolvation Areas*), que se basa en el análisis de regiones de la superficie de una proteína que tengan desolvatación favorable cuando se unan hipotéticamente a otra proteína (Fernández-Recio et al 2005). Esta puede ser una información adicional de interés en cualquier proyecto de *docking*.

Para lanzar los cálculos de ODA desde pyDock, para savinasa y BASI, respectivamente ejecuta los siguientes comandos:

```
>pyDock3 1svn_atom.pdb oda
>pyDock3 1ava_atom_C.pdb oda
```

**Nota:** Los ficheros resultantes 1svn\_atom.pdb.oda.ODAtab y 1ava\_atom\_C.pdb.oda.ODAtab contienen la lista de los residuos de savinasa y BASI, respectivamente, con sus correspondientes valores de ODA. Los ficheros 1svn\_atom.pdb.oda y 1ava\_atom\_C.pdb.oda son ficheros PDB en los que la columna de factores de temperatura (o *B-factors*) ha sido sustituida por los valores ODA, de forma que dichos valores puedan visualizarse fácilmente en la mayor parte de programas de gráficos moleculares. Las regiones con valor de ODA < -10.0 predicen regiones de interacción a proteínas.

#### 1.2.1. Visualización de los resultados del método ODA con ICM:

```
icm>read pdb "1svn_atom.pdb.oda" # lee el fichero de savinasa con los valores ODA
```

```
icm>display a_1.1
icm>color a_1.1/* -Bfactor(a_1.1/*) # colorea los residuos de acuerdo a los
valores de ODA
icm>read pdb "lava_atom_C.pdb.oda" # idem. para BASI
icm>display a_2.1
icm>color a_2.1/* -Bfactor(a_2.1/*)
```

**Nota:** En este caso se multiplica el valor del *B-factor* por -1.0 con el objetivo de que los valores más negativos de ODA (es decir, los más interesantes) se muestren en rojo.