**Descripción del funcionamiento de la aplicación**

La aplicación consiste en una interfaz gráfica de usuario la cual determina de manera automática las constantes cinéticas de unión tres proteínas modelo, **H-IgG**, **PSA**, **Anti-RBD**. Estas constantes son calculadas para determinar la afinidad con sus respectivos anticuerpos.

El funcionamiento de la aplicación se describe en los siguientes 4 pasos:

**Primer paso:**

El usuario carga un archivo **.xlsx** o **.csv**, utilizando el botón de **Load data**, con las señales **Rtp** las cuales fueron previamente invertidas empleando la ecuación:

**Rtp = 1 – Rt**,

en donde:

**Rt** son las señales originales de las mediciones de apagamiento de fluorescencia causada por la interacción con el óxido de grafeno. Estas señales se normalizan dividiendo , donde *Ii* es la intensidad de fluorescencia inicial e *If* es la intensidad de fluorescencia obtenida en cada tiempo de lectura.

Enseguida, la aplicación muestra en el widget de la izquierda un análisis estadístico desplegando el promedio (**mean**), la desviación estándar (**std**), el valor mínimo (**min**), el cuartil **Q1** (**25%**), el cuartil **Q2** (**50%**), el cuartil **Q3** (**75%**) y el valor máximo (**max**) de cada una de las mediciones **Rtp**. Además, la aplicación genera un archivo de texto dentro de la carpeta **output** con esta información estadística de las señales.

La Figura 1 muestra cómo se debe organizar la información que utiliza la aplicación. En la primera columna se debe introducir las mediciones de los tiempos en segundos. Luego, las siguientes columnas contienen las mediciones y puede haber ***m*** mediciones, la aplicación las detectará automáticamente sin necesidad de especificar. La primera fila es ignorada por la aplicación, pero se recomienda poner el nombre de las columnas, de tiempo y de las mediciones para tener una mejor documentación de los datos. La penúltima fila contiene información de las concentraciones de los analitos en unidades Molares. Y la última fila contiene las concentraciones del analito usado. El archivo puede contener ***n*** reglones, la aplicación los detectará automáticamente sin necesidad de especificar. Por lo tanto, cada columna incluye los valores de señal para cada medición de tiempo y la concentración de analito respectiva.

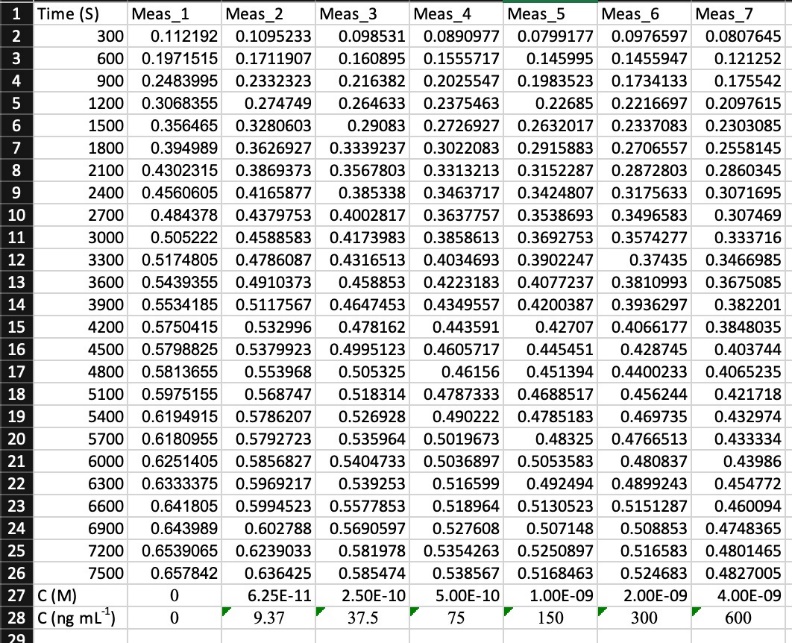


Figura 1. Organización de la información para la determinación de las constantes cinéticas de unión.

**Segundo paso:**

En el siguiente paso, cuando el usuario oprime el botón **Calculate**, la aplicación calcula las curvas de calibración de las señales **Rtp**, o encuentra un modelo matemático que ajusta las mediciones de las señales. Dada la naturaleza exponencial de las señales preprocesadas **Rtp**, se utilizó la ecuación de asociación de una fase ,o one phase association equation (Motulsky and Christopoulos, 2004) en inglés:

en donde:

son las señales **Rtp** expresadas en unidades arbitrarias.

son los tiempos en minutos de las señales **Rtp**,

es el valor de cuando esta tiende al infinito, expresada en las mismas unidades que ,

es la constante de velocidad, expresada como recíproco de las unidades de tiempo ,

es el valor de en el tiempo 0 expresada en las mismas unidades que las señales **Rtp**.

La **Figura 2** muestra un ejemplo de una curva de calibración utilizando el modelo de asociación de una fase con los parámetros y variables previamente mencionados.

Chart, line chart

Description automatically generated

**Figura 2.** Curva calibración para una señal **Rtp**. La gráfica naranja discontinua pertenece a una señal **Rtp**. La gráfica naranja continua corresponde a la curva de calibración de la señal **Rpt** obtenida utilizando el modelo de asociación de una fase. La gráfica verde es el **Plateau** de la señal.

El algoritmo de optimización que utiliza la aplicación para el ajuste de las curvas y determinación de las variables , y es el de **Lavender-Marquardt** (Gavin, 2019), implementado en la librería de SciPy (Virtanen et al., 2020), el cual ajusta de manera iterativa estas variables con el objetivo de reducir la función de error, en este caso mínimos cuadrados:

en donde:

son los argumentos, las variables , y , de función de asociación de una fase . Y es el número de parámetros de las mediciones.

Una vez que se calcularon estas variables, la aplicación despliega en el widget de la derecha una figura con las curvas de calibración en línea continua y en línea discontinua las señales **Rtp** y guarda una imagen .png dentro de la carpeta output. Además, la aplicación despliega en el widget de la izquierda los valores obtenidos de la ecuación de asociación de una fase y la métrica **R2** de las predicciones con respecto a las señales **Rtp** utilizando la librería de scikit-learn (Pedregosa et al., 2011), enseguida la aplicación guarda estos valores en el archivo de texto.

**Tercer paso:**

Una vez que se obtuvieron las variables , y para todas las señales, el siguiente paso la aplicación calcula las variables y .

En donde:

es un vector con las diferencias entre y de las señales y

es valor máximo del vector .

Calculados de la siguiente manera:

y

**.**

En seguida, la aplicación calcula las constantes cinéticas de unión de la siguiente forma:

en donde:

es la constante de asociación (M-1 s-1),

es la constante de disociación (s-1),

es la constante de disociación en equilibrio (M),

es la concentración del analito (M).

**Cuarto paso:**

La aplicación calcula el **error absoluto** para determinar el **error relativo** de las mediciones utilizando las fórmulas:

En donde:

es el error absoluto de las mediciones,

es el error relativo de las mediciones,

**A** son las mediciones,

es el promedio de las mediciones **A**,

es el número de mediciones.

El **error relativo** de las mediciones es expresado entre paréntesis utilizando un símbolo de más menos **±** en la interfaz gráfica.

Por último, esta información es desplegada en el widget izquierdo y guardada en el archivo de texto, junto con el tiempo en segundos que requirió el proceso de determinación de las constantes cinéticas de unión.

**Bibliografía**

Gavin, H.P., 2019. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems.

Motulsky, H., Christopoulos, A., 2004. Fitting Models to Biological Data Using Linear and Nonlinear Regression: A Practical Guide to Curve Fitting. Oxford University Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. J. Mach. Learn. Res. 12, 2825–2830.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, \.Ilhan, Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, 2020. Nat. Methods 17, 261–272.