# Fourier self-deconvolution

Análisis de espectros utilizando la deconvolución de picos mediante ajustes gaussianos. Se utilizan métodos de ajuste de curvas de SciPy para descomponer un espectro en sus componentes gaussianos, aunque también cabe la posibilidad de realizar el ajuste mediante curvas lorentzianas.

## Importación de librerías

**import** numpy **as** pnp

**import** matplotlib**.**pyplot **as** plt

**from** scipy**.**optimize **import** curve\_fit

**from** scipy**.**signal **import** find\_peaks**,** peak\_widths

## Importación y preparación de datos

x **=** wl\_recortado

y **=** data\_detrend**[**0**,:]**

Aquí, *x* y *y* representan los datos del espectro que serán analizados. x podría ser la longitud de onda y *y* la absorbancia o intensidad. Se elije el primero de los datos para comprobar el funcionamiento del algoritmo, luego se debe modificar para aplicarlo a todos los datos.

## Búsqueda de Picos

peaks**,** \_ **=** find\_peaks**(**y**,** distance**=**20**)**

peak\_centers **=** x**[**peaks**]**

Utiliza *find*\_*peaks()* para identificar los picos en los datos. *distance=20* indica que cada pico debe estar al menos a 20 puntos de distancia del siguiente. El valor de *distance* se debe modificar según la aplicación.

## Visualización de Picos

# Graficamos los resultados

plt**.**figure**(**figsize**=(**10**,** 6**))**

plt**.**plot**(**x**,** y**,** label**=**'Espectro Original'**)**

# Graficamos las líneas verticales en la posición del pico y las líneas horizontales representando el ancho

**for** center **in** peak\_centers**:**

plt**.**plot**([**center**,** center**],** **[**0**,** np**.**max**(**y**)],** 'k--'**,** label**=**'Centro del Pico'**)** # Línea vertical para el centro del pico

plt**.**title**(**"Análisis de Picos en el Espectro"**)**

plt**.**xlabel**(**"Número de onda (cm-1)"**)**

plt**.**ylabel**(**"Absorbancia"**)**

plt**.**legend**()**

plt**.**show**()**

**Gráfico, Gráfico de líneas

Descripción generada automáticamente**Genera un gráfico del espectro original con líneas verticales en los centros de los picos identificados por la función find\_peaks().

## Funciones para Ajuste de Curvas

**def** gaussian**(**x**,** amp**,** mu**,** sig**):**

fun\_val **=** amp **\*** np**.**exp**(-**np**.**power**(**x **-** mu**,** 2.**)** **/** **(**2 **\*** np**.**power**(**sig**,** 2.**)))**

**return** fun\_val

Esta función define el modelo gaussiano para un solo pico (Gaussian).

## Función General para Múltiples Picos

# Función general para múltiples picos

**def** multi\_gaussian**(**x**,** **\***params**):**

y **=** np**.**zeros\_like**(**x**)**

**for** i **in** range**(**0**,** len**(**params**),** 3**):**

amp**,** mu**,** sig **=** params**[**i**],** params**[**i**+**1**],** params**[**i**+**2**]**

y **+=** gaussian**(**x**,** amp**,** mu**,** sig**)**

**return** y

Una función más general para ajustar múltiples picos gaussianos. Permite mayor flexibilidad en el número de picos que se van a emplear en la *deconvolución*.

## Preparación para Ajuste de Múltiples Picos

# Número de picos

num\_peaks **=** len**(**peak\_centers**)**

# Definir la histéresis o factor de tolerancia para los límites

hysteresis\_amp **=** 0.05 # 5% de histéresis para la amplitud

hysteresis\_center **=** 0.1 # 10% de histéresis para el centro

# Genera valores iniciales basados en los centros de picos

initial\_guesses **=** np**.**zeros**(**num\_peaks **\*** 3**)**

# Generar límites inferiores y superiores

bounds\_lower **=** np**.**zeros\_like**(**initial\_guesses**)**

bounds\_upper **=** np**.**zeros\_like**(**initial\_guesses**)**

**for** i**,** peak **in** enumerate**(**peak\_centers**):**

idx**,** \_ **=** encontrar\_cercano**(**wl\_recortado**,** peak**)**

initial\_guesses**[**3**\***i**]** **=** y**[**idx**]**

initial\_guesses**[**3**\***i**+**1**]** **=** x**[**idx**]**

initial\_guesses**[**3**\***i**+**2**]** **=** 8

bounds\_lower**[**3**\***i**]** **=** initial\_guesses**[**3**\***i**]** **\*** **(**1 **-** hysteresis\_amp**)**

bounds\_lower**[**3**\***i**+**1**]** **=** initial\_guesses**[**3**\***i**+**1**]** **\*** **(**1 **-** hysteresis\_center**)**

bounds\_lower**[**3**\***i**+**2**]** **=** 2

bounds\_upper**[**3**\***i**]** **=** initial\_guesses**[**3**\***i**]** **\*** **(**1 **+** hysteresis\_amp**)**

bounds\_upper**[**3**\***i**+**1**]** **=** initial\_guesses**[**3**\***i**+**1**]** **\*** **(**1 **+** hysteresis\_center**)**

bounds\_upper**[**3**\***i**+**2**]** **=** 15

Genera conjeturas iniciales y límites para el ajuste de múltiples picos basándose en los centros de los picos encontrados.

## Ajuste y Visualización de Múltiples Picos

# Encontrar y ajustar los parámetros óptimos

popt**,** pcov **=** curve\_fit**(**multi\_gaussian**,** x**,** y**,** p0**=**initial\_guesses**,** bounds**=(**bounds\_lower**,** bounds\_upper**))**

# Calcular el ajuste y los componentes individuales

yfit **=** multi\_gaussian**(**x**,** **\***popt**)**

components **=** **[**gaussian**(**x**,** **\***popt**[**i**:**i**+**3**])** **for** i **in** range**(**0**,** len**(**popt**),** 3**)]**

# Plotear el espectro y los componentes

plt**.**figure**(**figsize**=(**12**,** 7**))**

plt**.**plot**(**x**,** y**,** '-o'**,** label**=**'Data'**)**

**for** i**,** component **in** enumerate**(**components**):**

plt**.**plot**(**x**,** component**,** label**=**f'Peak {i**+**1}'**)**

plt**.**plot**(**x**,** yfit**,** linewidth**=**2**,** label**=**'Total Fit'**)**

plt**.**xlabel**(**'Wavelength (nm)'**)**

plt**.**ylabel**(**'Absorbance'**)**

plt**.**title**(**'Peak Deconvolution'**)**

plt**.**legend**()**

plt**.**show**()**

Gráfico, Gráfico de líneas, Histograma

Descripción generada automáticamente