

基于最小二乘拟合的复合光谱分峰模型

摘要

PCR（聚合酶链式反应）是一种用于放大扩增特定 DNA 片段的分子生物学技术，近年来广泛应用于核酸检测，可以大大提高检测的准确性和应用性。荧光定量 PCR 技术在 PCR 基础上引入了荧光基团，可以实时检测产物中的荧光信号并进行定量分析。实际在荧光信号的检测过程中，不同荧光基团产生的荧光具有不同的光谱范围，需要对荧光光谱进行区分。实现复合光谱分峰的方法，从采集的复合光谱曲线中分离不同荧光基团对应的光谱曲线，能够在保证灵敏度的同时对光谱进行区分，对于核酸检测技术及荧光定量 PCR 反应技术都具有重要意义。

针对问题一：先对附件 1 中的三组数据分别绘制双峰复合光谱信号强度与波长的散点图，结合数据与图像便可粗略估计各自对应的单峰光谱波长、峰高及半峰宽数值，从而大致确定各单峰曲线的位置和参数。再分析荧光光谱的形成过程，可得理想状态下单峰光谱的峰强与波长满足一维高斯分布，故采用高斯函数对单峰光谱进行拟合。对于每一个双峰复合光谱，将其对应的两个单峰的预估波长、峰高及半峰宽数值代入高斯函数，便可得到两个较为粗略的单峰分峰。再利用最小二乘法原理，使两个预估分峰的叠加图与实际复合图之间的误差平方和达到最小，便可较为准确地实现双峰复合光谱的分峰，进而求出分峰后的独立光谱曲线与原始光谱曲线的线性相关系数。

针对问题二：问题二要在问题一的基础上探究峰高比对双峰分峰模型的影响。可以在满足瑞利判据的前提下构建虚拟数据，对给定的波长和半峰宽，令峰高比从 1.000 以 0.333 步长遍历至 9.000，大致模拟实际光谱峰高比范围。将构建的虚拟数据输入问题一的双峰分峰模型，比较拟合图与数据图的线性相关系数与相对误差，从而分析峰高比对双峰分峰效果的影响。

针对问题三：此问相较于问题一仅发生了峰数变化，可尝试套用双峰分峰思路建模。注意到多峰复合光谱的重叠度较双峰更高，分析数据图像得到的波长、峰高及半峰宽估计值较真实值的偏差会更大，但初始估计值的偏差可以通过最小二乘法的迭代步数抵消，误差的平方和仍会达到最小。故仍然可对附件 2 中的八组数据分别绘制复合光谱信号强度与波长的散点图，粗略估计各自对应的单峰光谱波长、峰高及半峰宽。再用高斯函数拟合各单峰曲线，使用最小二乘法迭代至拟合叠加图与实际复合图的误差平方和最小，便可较为准确地实现多峰复合光谱的分峰，进而求出分峰后的独立光谱曲线与原始光谱曲线的线性相关系数。

针对问题四：要建立非对称复合光谱的分峰模型。利用题目数据绘图可知，非对称复合光谱各峰重叠度极大，曲线形状较不规则，无法用前三问的方法直接从原数据中预测各分峰的位置及参数。鉴于理想状态下单峰光谱的峰强与波长满足一维高斯分布，不妨假设非对称光谱可由两个半峰宽不同的高斯曲线各取一半而拼成，进而对非对称曲线实现参数化描述。采用小波分析与二次微分相结合的方法处理数据，降低数据噪声并提取分峰参数的估计值，再用最小二乘拟合的方法逼近期望输出，实现非对称复合光谱的分峰。

关键词：高斯函数；最小二乘；小波分析；二次微分

1 问题重述

PCR（聚合酶链式反应）是一种用于放大扩增特定 DNA 片段的分子生物学技术，近年来广泛应用于核酸检测，可以大大提高检测的准确性和应用性。荧光定量 PCR 技术在 PCR 基础上引入了荧光基团，可以实时检测产物中的荧光信号，从而通过标准曲线对未知模板进行定量分析。实际在荧光信号的检测过程中，不同荧光基团产生的荧光具有不同的光谱范围，需要对荧光光谱进行区分。然而传统的区分方法使用了窄带滤光片，其透过光谱范围约为 15-20nm 较窄，会降低荧光信号的强度，从而降低检测的灵敏度。

所以，实现复合光谱分峰的方法，从采集的复合光谱曲线中分离不同荧光基团对应的光谱曲线，能够在保证灵敏度的同时对光谱进行区分，对于核酸检测技术及荧光定量 PCR 反应技术都具有重要意义。

- 问题 1：利用附件 1 中的数据建立双峰复合光谱的分峰模型，给出分峰后各峰的中心波长、高度及半峰宽，并计算得到的独立光谱曲线和原始曲线的线性相关系数。
- 问题 2：分析在问题 1 的双峰分峰模型中，峰高比对所建立的分峰模型的影响。除使用赛题提供数据外，可使用虚拟数据仿真协助分析。
- 问题 3：利用附件 2 中三峰和四峰叠加的光谱曲线及数据，建立多个光谱峰叠加的复合光谱曲线模型，将多峰复合光谱曲线拆分为独立光谱曲线（最多四个峰），并给出分离后各峰的中心波长、高度及半峰宽，与其相应的准单色光谱进行比较。
- 问题 4：实际检测中的荧光光谱曲线多数是非对称的光谱曲线。利用附件 3 中实际的检测数据尝试建立非对称光谱曲线分峰的数学模型，将复合光谱曲线拟合拆分为对应的独立光谱曲线。

2 问题分析

2.1 问题一的分析

问题一要建立双峰复合光谱的分峰模型，并给出各峰的中心波长、高度及半峰宽。先对附件 1 中的三组数据分别绘制双峰复合光谱信号强度与波长的散点图，结合数据与图像便可粗略估计各自对应的单峰光谱波长、峰高及半峰宽数值，从而大致确定各单峰曲线的位置和参数。再分析荧光光谱的形成过程，可得理想状态下单峰光谱的峰强与波长满足一维高斯分布，故采用高斯函数对单峰光谱进行拟合。对于每一个双峰复合光谱，将其对应的两个单峰的预估波长、峰高及半峰宽数值代入高斯函数，便可得到两个较为粗略的单峰分峰。再利用最小二乘法原理，使两个预估分峰的叠加图与实际复合图之间的误差平方和达到最小，便可较为准确地实现双峰复合光谱的分峰，进而求出分峰后的独立光谱曲线与原始光谱曲线的线性相关系数。

2.2 问题二的分析

问题二要在问题一的基础上探究峰高比对双峰分峰模型的影响。可以在满足瑞利判据的前提下构建虚拟数据，对给定的波长和半峰宽，令峰高比从 1.000 以 0.333 步长遍历至 9.000，大致模拟

实际光谱峰高比范围。将构建的虚拟数据输入问题一的双峰分峰模型，比较拟合图与数据图的线性相关系数与相对误差，从而分析峰高比对双峰分峰效果的影响。

2.3 问题三的分析

问题三要建立三峰和四峰复合光谱的分峰模型，并给出各峰的中心波长、高度及半峰宽。此问相较于问题一仅发生了峰数变化，可尝试套用双峰分峰思路建模。注意到多峰复合光谱的重叠度较双峰更高，分析数据图像得到的波长、峰高及半峰宽估计值较真实值的偏差会更大，但初始估计值的偏差可以通过最小二乘法的迭代步数抵消，误差的平方和仍会达到最小。故仍然可对附件 2 中的八组数据分别绘制复合光谱信号强度与波长的散点图，粗略估计各自对应的单峰光谱波长、峰高及半峰宽。再用高斯函数拟合各单峰曲线，使用最小二乘法迭代至拟合叠加图与实际复合图的误差平方和最小，便可较为准确地实现多峰复合光谱的分峰，进而求出分峰后的独立光谱曲线与原始光谱曲线的线性相关系数。

2.4 问题四的分析

问题四要建立非对称复合光谱的分峰模型。利用题目数据绘图可知，非对称复合光谱各峰重叠度极大，曲线形状较不规则，无法用前三问的方法直接从原数据中预测各分峰的位置及参数。鉴于理想状态下单峰光谱的峰强与波长满足一维高斯分布，不妨假设非对称光谱可由两个半峰宽不同的高斯曲线各取一半而拼成，进而对非对称曲线实现参数化描述。采用小波分析与二次微分相结合的方法处理数据，降低数据噪声并提取分峰参数的估计值，再用最小二乘拟合的方法逼近期望输出，实现非对称复合光谱的分峰。

3 模型假设

- 假设一：光谱宽度由荧光分子不规则运动导致的多普勒效应造成。
- 假设二：非对称光谱由两个半峰宽不同的高斯曲线各取一半拼成。

4 符号说明

符号	解释
y_i	第 i 组双峰复合数据预估峰值
x_i	第 i 组双峰复合数据预估中心波长
c	真空中的光速 (m/s)
m	一个荧光分子的质量 (kg)
T	温度 (K)
k	玻尔兹曼常量 (J/K)

表 1: 符号说明 (注：未申明的变量以其在出现出的具体说明为准)

5 问题一模型的建立与求解

5.1 数据处理

5.1.1 预估峰高及中心波长

利用附件 1 中的三组数据分别绘制散点图，如图 1 所示。

为了得到峰强的初步估计值，可使用 python 中的 `signal.find_peaks` 函数在峰强数据中寻找极值点。考虑到数据受噪声影响会导致极值点较多，但光谱图像形状规则曲线清晰，表明噪声对光谱峰信号的干扰远小于峰体本身的信号强度，故可对极值点的峰强范围做限制，用程序筛选出双峰复合光谱波峰与波谷处对应的极值点并在图上用星形标记（详见附录 A.1 代码）。将第 i 组的复合光谱两峰峰强分别记作 y_{1i}, y_{2i} ，以之作为分峰的峰强估计值，再使用数据索引得到 y_{1i}, y_{2i} 对应的波长 x_{1i}, x_{2i} ($x_{1i} < x_{2i}$)，作为两个分峰的中心波长估计值。

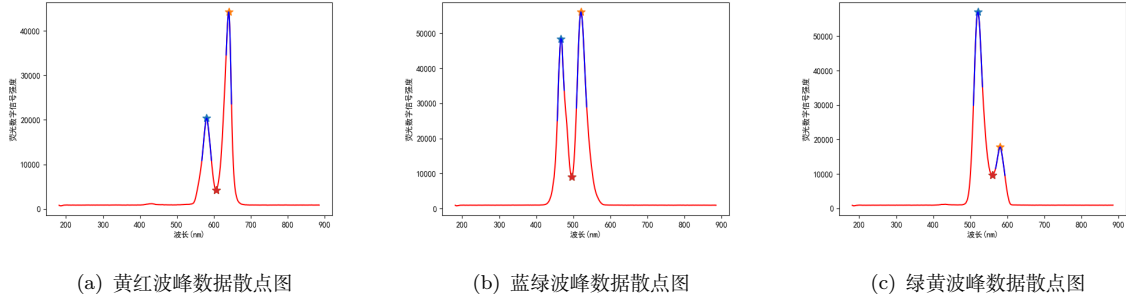


图 1: 附件 1 双峰光谱数据散点图

5.1.2 预估半峰宽

观测图 1 可知，波长小于 400nm 或大于 700nm 范围内三组数据的信号强度均较为平坦，相较于波峰的信号强度变化不大。故对第 i 组前 100 个数据的信号强度取平均数 y_{0i} ，表示峰底的信号强度。再将两个波峰的信号强度分别与峰底的信号强度取平均值，便可得到两个分峰的半峰高 y_{3i}, y_{4i} 。继而检索信号强度高于半峰高的数据点，提取其对应波长的最小值与最大值，记为 x_{3i}, x_{4i} ，则两峰的预测半峰宽 $\Delta x_{1i}, \Delta x_{2i}$ 可表示为：

$$\begin{cases} \Delta x_{1i} = 2(x_{1i} - x_{3i}) & \text{对应横坐标范围}[x_{3i}, x_{3i} + \Delta x_{1i}] \\ \Delta x_{2i} = 2(x_{4i} - x_{2i}) & \text{对应横坐标范围}[x_{4i} - \Delta x_{2i}, x_{2i}] \end{cases} \quad (5.1.1)$$

位于半峰宽横坐标范围内的点在图 1 中用蓝色表示。

5.1.3 预估数据整合

初步估计得到的分峰波长、峰高及半峰宽如下表所示：

参数名	黄红	蓝绿	绿黄
第 1 个波峰的强度	20,393.84	48,481.82	57,039.24
第 2 个波峰的强度	44,217.84	56,080.16	17,803.51
第 1 个波谷的强度	4,287.12	9,010.17	9,644.65
第 1 个峰的半峰宽 (nm)	26.12	18.64	24.64
第 2 个峰的半峰宽 (nm)	14.66	31.50	25.30
第 1 个波峰的中心波长 (nm)	580.86	466.24	520.84
第 2 个波峰的中心波长 (nm)	640.77	520.49	580.86

表 2: 问题一分峰参数估计值

5.2 模型建立

5.2.1 高斯函数拟合单峰光谱

由假设一，理想状态下光谱宽度由荧光分子不规则运动导致的多普勒效应造成，设所有荧光分子在各自参考系中发出光的波长为 λ_0 ，某时刻一荧光分子以速度 v 沿与观测者成 θ 角方向做无规则运动，则根据多普勒效应，接收波长 λ 可表示为：

$$\lambda = \lambda_0 \left(1 - \frac{v}{c} \cos\theta\right) \left(1 - \frac{v^2}{c^2}\right)^{-\frac{1}{2}} \quad (5.2.1)$$

荧光分子无规则运动的速度远低于光速，故可以忽略相对论效应，不妨设 $v_1 = v \cos\theta$ ，则：

$$\lambda = \lambda_0 \left(1 - \frac{v}{c} \cos\theta\right) = \lambda_0 \left(1 - \frac{v_1}{c}\right) \quad (5.2.2)$$

且荧光分子服从麦克斯韦速度分布律：

$$f(v_1) dv_1 = \left(\frac{m}{2\pi kT}\right)^{\frac{1}{2}} \cdot e^{-\frac{mv_1^2}{2kT}} dv_1 \quad (5.2.3)$$

记 $\Delta\lambda = \lambda - \lambda_0$ ，则：

$$\frac{\Delta\lambda}{\lambda_0} = -\frac{v_1}{c} \quad (5.2.4)$$

将式 (5.2.4) 代入 (5.2.3)：

$$f(\Delta\lambda) d\lambda = \left(\frac{mc^2}{2\pi kT\lambda_0^2}\right)^{\frac{1}{2}} \cdot e^{-\frac{mc^2}{2kT}\left(\frac{\Delta\lambda}{\lambda_0}\right)^2} d\lambda \quad (5.2.5)$$

整理可得：

$$f(\Delta\lambda) d\lambda = \frac{1}{\sqrt{2\pi}} \left(\sqrt{\frac{kT\lambda_0^2}{mc^2}}\right)^{-1} \cdot e^{-\frac{(\lambda-\lambda_0)^2}{2\left(\sqrt{\frac{kT\lambda_0^2}{mc^2}}\right)^2}} d\lambda \quad (5.2.6)$$

发出 $\lambda_0 + \Delta\lambda$ 波长光的荧光分子满足上述分布，而高斯函数形如：

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad a = \frac{1}{\sigma\sqrt{2\pi}}, b = \mu, c = \sigma \quad (5.2.7)$$

形式相同，故可用高斯函数拟合单峰光谱。

5.2.2 最小二乘拟合复合光谱

用高斯型函数模拟的单个谱峰形如:

$$g(\lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\lambda-\mu)^2}{2\sigma^2}} = a e^{-\frac{(\lambda-\mu)^2}{2\sigma^2}},$$

其中 $g(\lambda)$ 为关于光谱波长 λ 的函数, μ, a 分别为谱线的中心波长和峰高, σ 为半峰宽的 $\frac{1}{2\sqrt{\ln 2}}$. 设区间 $[a, b]$ 内任一点以整数 i ($i=1, 2, \dots, N$) 表示, 则以 i 点为中心的高斯函数记作:

$$g_i(\lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\lambda-i)^2}{2\sigma_i^2}} = a_i e^{-\frac{(\lambda-i)^2}{2\sigma_i^2}}$$

则 N 峰叠加拟合谱图可表示为:

$$f(\lambda) = \sum_{i=1}^N g_i(\lambda) = \sum_{i=1}^N a_i e^{-\frac{(\lambda-i)^2}{2\sigma_i^2}} \quad (i = 1, 2, \dots, N)$$

设实际图谱为 $f^*(\lambda)$, 则 $f(\lambda)$ 与 $f^*(\lambda)$ 的误差平方和函数定义为:

$$E = \frac{1}{2} \sum_{i=1}^N [f(\lambda) - f^*(\lambda)]^2$$

寻找最小二乘法意义下的最优拟合函数 $f(\lambda)$, 实质上是确定函数 E 的极小值问题。

5.3 模型求解

5.3.1 求解方法

E 是关于 (a_i, σ_i) ($i = 1, 2, \dots, N$) 的多元非线性函数, 用解析法直接求解极值存在困难, 但可以通过数值迭代法进行。数值迭代法分为两类: 一类是导数法, 该方法以目标函数导数的负方向作为下一次迭代方向; 另一类是直接法, 该方法一般是在目标函数的导数难以得到的情况下, 直接利用目标函数而不利用其导数来确定下一次迭代方向。一般而言, 前者的计算效率比后者高 [1]。本文目标函数 E 的导数存在并较易得到, 故采用导数法。对 E 关于 a_i, σ_i ($i = 1, 2, \dots, N$) 分别求偏导, 得到:

$$\begin{cases} \frac{\partial E}{\partial a_i} = \sum_{i=1}^N [f(\lambda) - f^*(\lambda)] e^{-\frac{(\lambda-i)^2}{2\sigma_i^2}} \\ \frac{\partial E}{\partial \sigma_i} = \sum_{j=1}^N [f(\lambda) - f^*(\lambda)] e^{-\frac{(j-i)^2}{2\sigma_i^2}} \frac{a_i(\lambda-i)^2}{\sigma_i^3} \end{cases}$$

为了保证迭代收敛的稳定性, 引入惯性因子 α 后的调整公式如下:

$$\begin{cases} \Delta a_i(t+1) = -p_a \frac{\partial E}{\partial a_i} + \alpha_a \Delta a_i(t), \\ \Delta \sigma_i(t+1) = -p_\sigma \frac{\partial E}{\partial \sigma_i} + \alpha_\sigma \Delta \sigma_i(t), \end{cases}$$

$\Delta a_i(t)$ 和 $\Delta a_i(t+1)$ 分别为 a_i 第 t 次和 $t+1$ 次迭代调整量, p_a 和 α_a 分别为其步长和惯性因子; $\Delta \sigma_i(t)$ 和 $\Delta \sigma_i(t+1)$ 分别为 σ_i 第 t 次和 $t+1$ 次迭代调整量, p_σ 和 α_σ 分别为其步长和惯性因子。
[1] (算法实现详见附录 A.2)

5.3.2 结果及分析

拟合得到的单峰与原始单峰线性相关系数如上表，线性相关系数均超过 0.95，分峰效果较好

组别	线性相关系数	
黄红	0.996352008517681(黄)	0.959712648696846(红)
蓝绿	0.981953296003524(蓝)	0.980693907202059(绿)
绿黄	0.989571798229459(绿)	0.980693907202059(黄)

表 3: 问题一双峰分峰线性相关系数

6 问题二模型的建立与求解

6.1 数据处理

6.1.1 构建虚拟数据

对给定的波长和半峰宽，在满足瑞利判据的前提下，令峰高比从 1.000 以 0.333 步长遍历至 9.000，生成 50 组虚拟的单峰光谱数据，叠加得到 25 组双峰复合光谱数据。(程序与数据参见附录 B.1)

6.1.2 预估曲线参数

预估双峰复合曲线的中心波长、峰高及半峰宽仍采用 5.1 的处理方法。绘制复合光谱数据散点图如图 3，整合得到的预估数值见下表：

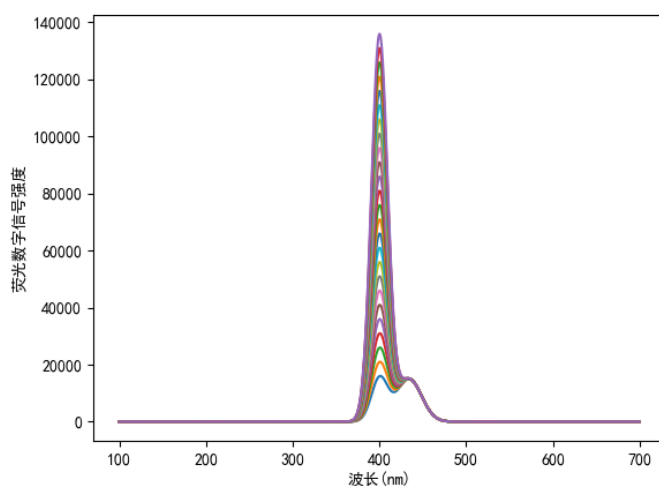
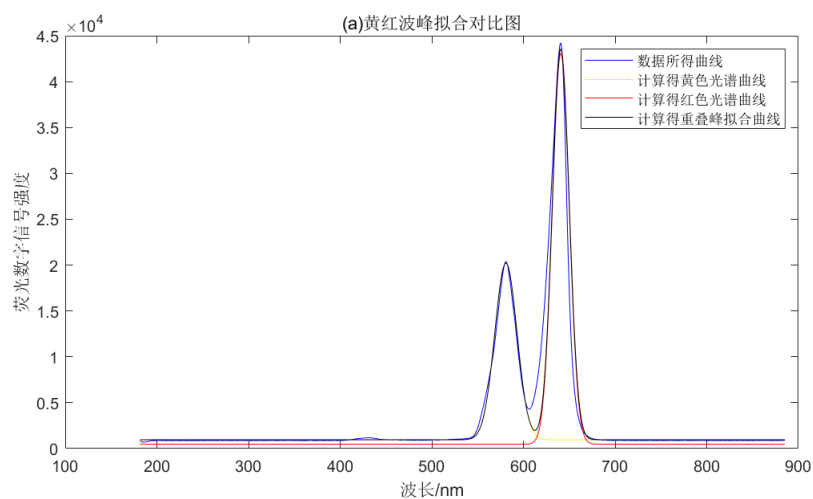
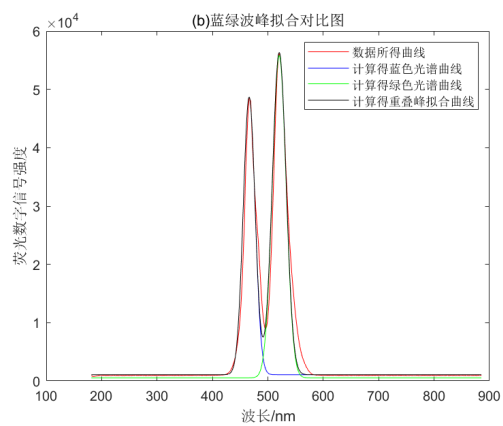


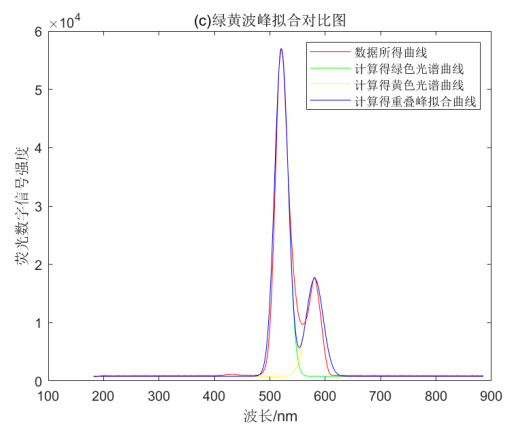
图 3: 问题二虚拟数据峰高比 1-9 散点图



(a) 黄红波峰数据散点图



(b) 蓝绿波峰数据散点图



(c) 绿黄波峰数据散点图

图 2: 附件 1 双峰拟合对比图

注：半峰宽与中心波长单位为纳米 (nm)

峰高比	波峰 1 峰强	波峰 2 峰强	峰 1 半峰宽	峰 2 半峰宽	峰 1 中心波长	峰 2 中心波长
1.000	16,013.01	15,017.55	22.80	33.60	401.20	433.60
1.333	20,991.78	15,025.32	21.60	33.60	400.60	433.60
1.667	25,981.48	15,033.08	21.60	33.60	400.60	433.60
2.000	30,971.18	15,040.85	22.80	33.60	400.60	433.60
2.333	35,960.88	15,048.62	22.80	33.60	400.60	433.60
2.667	40,950.58	15,056.39	22.80	33.60	400.60	433.60
3.000	45,940.28	15,064.15	22.80	33.60	400.60	433.60
3.333	50,937.50	15,071.92	21.60	33.60	400.00	433.60
3.667	55,937.50	15,079.69	21.60	33.60	400.00	433.60
4.000	60,937.50	15,087.46	21.60	33.60	400.00	433.60
4.333	65,937.50	15,095.22	21.60	33.60	400.00	433.60
4.667	70,937.50	15,102.99	21.60	33.60	400.00	433.60
5.000	75,937.50	15,110.76	21.60	33.60	400.00	433.60
5.333	80,937.50	15,120.32	21.60	34.80	400.00	433.00
5.667	85,937.50	15,130.08	21.60	34.80	400.00	433.00
6.000	90,937.50	15,139.85	21.60	34.80	400.00	433.00
6.333	95,937.50	15,149.61	21.60	34.80	400.00	433.00
6.667	100,937.50	15,159.38	21.60	34.80	400.00	433.00
7.000	105,937.50	15,169.14	21.60	34.80	400.00	433.00
7.333	110,937.50	15,178.91	21.60	34.80	400.00	433.00
7.667	115,937.50	15,189.41	21.60	36.00	400.00	432.40
8.000	120,937.50	15,201.63	21.60	36.00	400.00	432.40
8.333	125,937.50	15,213.86	21.60	36.00	400.00	432.40
8.667	130,937.50	15,226.09	21.60	36.00	400.00	432.40
9.000	135,937.50	15,238.52	21.60	37.20	400.00	431.80

表 4: 问题二虚拟分峰参数估计值

6.2 模型建立与求解

使用 5.2,5.3 中的最小二乘拟合模型，得出线性相关系数，相对误差如下表所示：

峰高比	复合峰相关系数	峰 1 相关系数	峰 2 相关系数	相对误差
1.000	0.999671354	0.998875278	0.999783442	0.000586865
1.333	0.999561478	0.998866441	0.999573098	0.000785398
1.667	0.999480277	0.998874813	0.999425651	0.000934852
2.000	0.99942004	0.998877857	0.999252767	0.001048149
2.333	0.999919491	0.999969415	0.999357757	0.000145843
2.667	0.999935198	0.999976571	0.999357506	0.000117902
3.000	0.999946464	0.999983288	0.999382956	9.79582E-05
3.333	0.999955443	0.999986436	0.999382982	8.18016E-05
3.667	0.999962401	0.999988774	0.999383011	6.92287E-05
4.000	0.999967886	0.999990556	0.999383041	5.92804E-05
4.333	0.999972276	0.999991946	0.999383071	5.12899E-05
4.667	0.99997584	0.999993051	0.999383101	4.47848E-05
5.000	0.999977549	0.999994605	0.999400705	4.18223E-05
5.333	0.99998012	0.999995374	0.99940071	3.70869E-05
5.667	0.999935222	0.999979045	0.997394604	0.000120417
6.000	0.999941874	0.999981103	0.99737909	0.000108163
6.333	0.999947599	0.999983038	0.99737913	9.76204E-05
6.667	0.999952526	0.99998469	0.997379172	8.85338E-05
7.000	0.99995653	0.999985583	0.997306931	8.1072E-05
7.333	0.999960275	0.999986867	0.997306982	7.41515E-05
7.667	0.999920726	0.999974266	0.993979483	0.000148172
8.000	0.99992703	0.999976369	0.993980259	0.00013649
8.333	0.999932615	0.999978224	0.993981035	0.000126129
8.667	0.999889894	0.999963688	0.98833575	0.000205733
9.000	0.999847926	0.999951322	0.981786848	0.000284283

表 5: 问题二拟合误差

随着峰高比的增大，拟合的线性相关系数越低，相对误差越大，分峰效果越差。

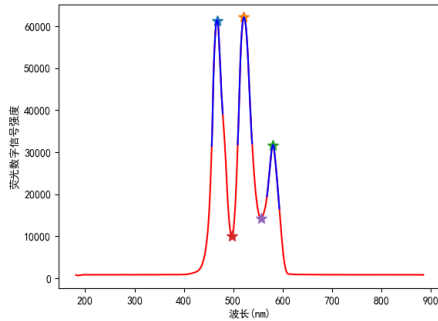
7 问题三模型的建立与求解

7.1 数据处理

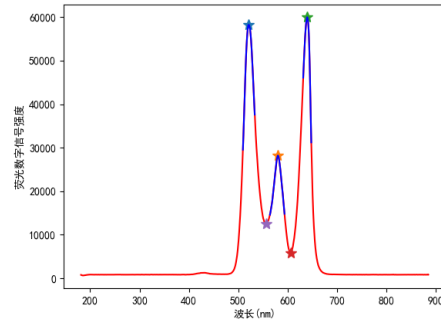
7.1.1 三峰复合光谱分峰模型

利用附件 2 中的两组数据分别绘制散点图，如图 4 所示。

光谱图象形状清晰曲线规则，推测噪声对光谱峰信号的干扰较小，仍使用`signal.find_peaks`函数寻找峰强极值点的方法预测各分峰峰高及中心波长，筛选出波峰波谷处的极值点并在图上用星形标记（详见附录 C.1 代码）。用波峰处极值点的信号强度作为各分峰的预估峰强，检索波峰对应的荧光波长作为各分峰的预估中心波长，得到峰强与中心波长的粗略估计值。



(a) 黄红波峰数据散点图



(b) 蓝绿波峰数据散点图

图 4: 附件 2 三峰光谱数据散点图

受到三峰复合光谱两个波谷的限制，中间波峰的半峰宽可能无法用 5.1.2 的方法估计得出。随着复合光谱峰数的增加，需要建立更普适的半峰宽估计方法。若多峰复合光谱由多个对称的单峰光谱叠加形成，且重叠度相对不高，则可使用以下流程方法预估半峰宽：

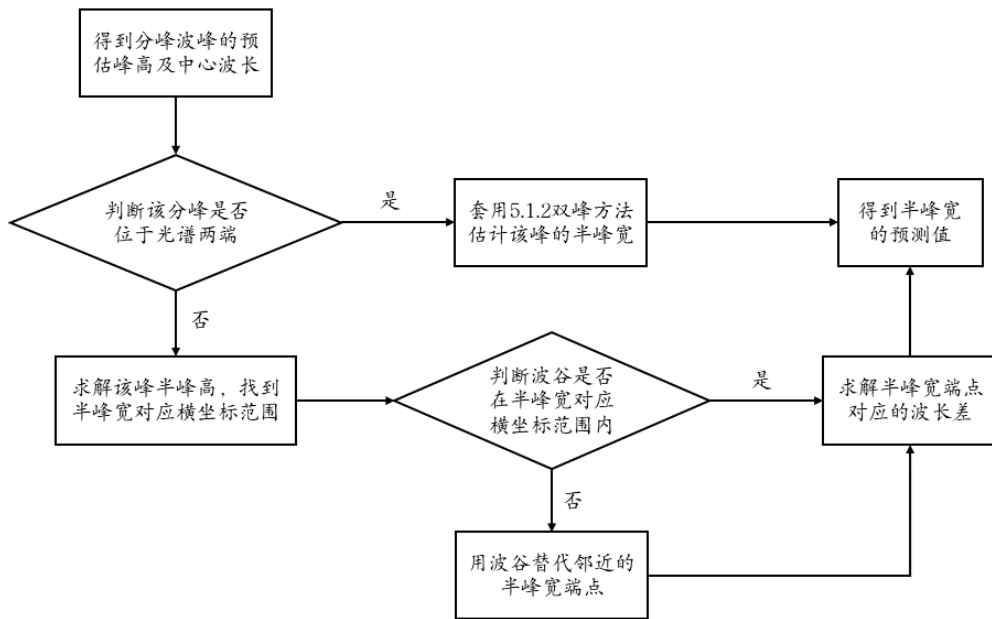


图 5: 多峰复合光谱预估分峰半峰宽流程图

图四的数据中, 各波谷均位于分峰的预估半峰宽之外, 只需求出半峰宽端点对应的波长即可对单峰的半峰宽做出估计。位于半峰宽横坐标范围内的点在图 4 中用蓝色表示, 初步估计得到的分峰波长、峰高及半峰宽整合如下表所示:

参数名	蓝绿黄	绿黄红
第 1 个波峰的强度	61,357.43	58,223.72
第 2 个波峰的强度	62,113.91	28,198.34
第 3 个波峰的强度	31,745.64	59,963.55
第 1 个波谷的强度	10,049.51	5,845.72
第 2 个波谷的强度	14,315.76	12,475.11
第 1 个峰的半峰宽 (nm)	22.22	23.92
第 2 个峰的半峰宽 (nm)	24.62	32.32
第 3 个峰的半峰宽 (nm)	24.76	16.06
第 1 个波峰的中心波长 (nm)	467.67	521.54
第 2 个波峰的中心波长 (nm)	521.54	580.51
第 3 个波峰的中心波长 (nm)	580.86	639.76

表 6: 问题三三峰复合光谱分峰参数估计值

7.1.2 四峰复合光谱分峰模型

利用附件 2 中的两组数据分别绘制散点图，如图 7 所示。

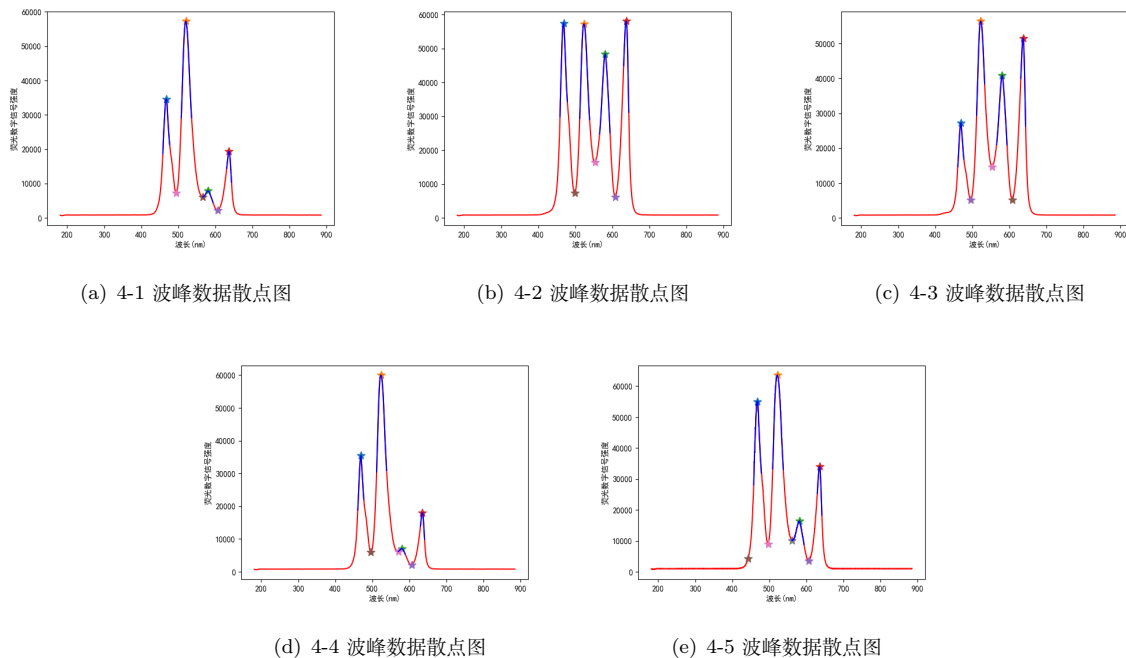


图 6: 附件 2 四峰光谱数据散点图

使用 7.1.1 中多峰复合光谱预估单峰参数的方法，用 `signal.find_peaks` 函数筛选出波峰波谷处的极值点并在图上用星形标记（详见附录 C.1 代码），以波峰处极值点的信号强度作为各分峰的预估峰强，通过检索得到各单峰的预估中心波长，再按图 5 的方法估得四峰复合光谱各分峰的半峰宽。位于半峰宽横坐标范围内的点在图 6 中用蓝色表示，初步估计得到的分峰波长、峰高及半峰宽整合如下表所示：

参数名	4-1	4-2	4-3	4-4	4-5
第 1 个波峰的强度	34,590.02	57,540.05	27,313.05	35,453.46	55,134.35
第 2 个波峰的强度	57,311.73	57,278.40	56,431.14	60,009.35	63,571.38
第 3 个波峰的强度	7,878.60	48,458.84	40,906.30	7,071.15	16,456.57
第 4 个波峰的强度	19,374.09	58,078.25	51,583.81	18,018.08	34,027.23
第 1 个波谷的强度	2,172.69	6,134.56	5,258.98	2,045.34	3,572.87
第 2 个波谷的强度	6,098.64	7,315.81	5,265.13	2,045.34	4,359.29
第 3 个波谷的强度	7,206.42	16,418.38	14,728.22	6,202.04	9,087.52
第 1 个峰的半峰宽 (nm)	18.64	20.08	16.46	16.46	20.78
第 2 个峰的半峰宽 (nm)	24.64	23.18	21.78	23.88	25.32
第 3 个峰的半峰宽 (nm)	27.50	30.26	30.26	18.56	41.30
第 4 个峰的半峰宽 (nm)	12.7	13.40	12.72	18.56	12.08
第 1 个波峰的中心波长 (nm)	467.67	468.75	469.10	469.10	468.39
第 2 个波峰的中心波长 (nm)	520.49	523.64	522.59	523.64	521.89
第 3 个波峰的中心波长 (nm)	580.51	580.86	580.86	579.49	581.89
第 4 个波峰的中心波长 (nm)	637.09	637.42	637.42	635.41	635.75

表 7: 问题三四峰复合光谱分峰参数估计值

7.2 模型的建立与求解

由 5.2.2 可知，有限个单峰光谱的复合均可用最小二乘拟合逼近，故仍使用最小二乘拟合模型，拟合图像及线性相关系数如下：

7.2.1 三峰复合光谱分峰模型

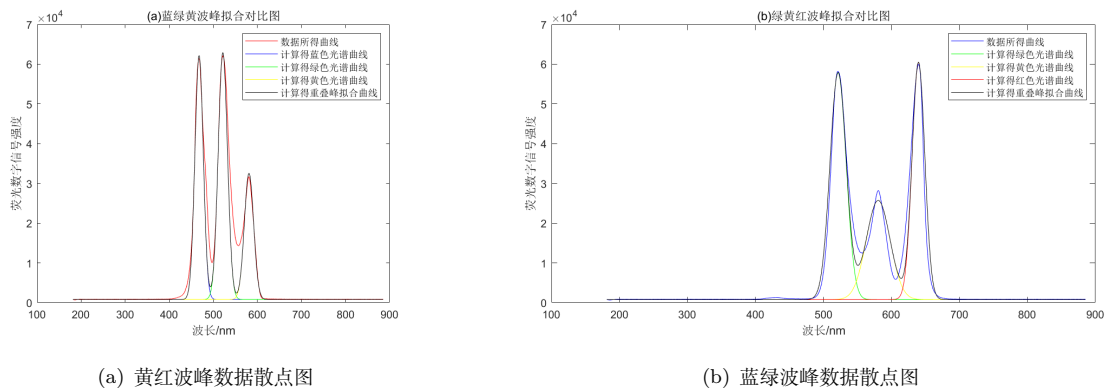


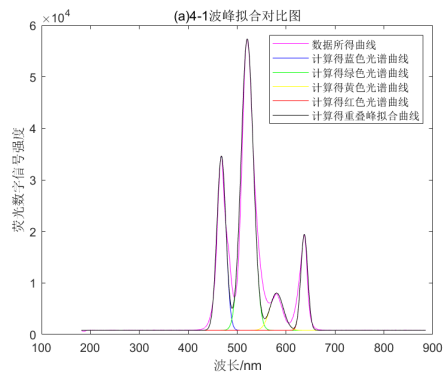
图 7: 附件 2 三峰拟合对比图

组别	线性相关系数		
蓝绿黄	0.991671208495265(蓝)	0.985592034458935(绿)	0.994994411962427(黄)
绿黄红	0.985601173075390(绿)	0.960119254105144(黄)	0.977446030640229(红)

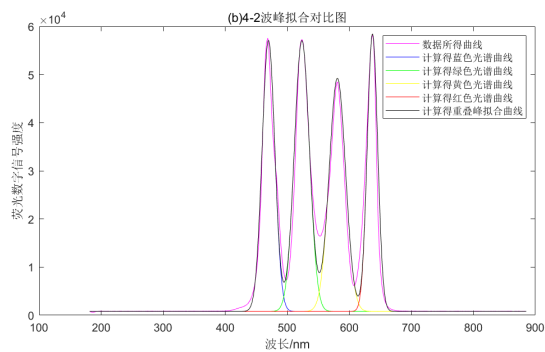
表 7: 问题三三峰分峰线性相关系数

拟合得到的单峰与原始单峰线性相关系数如上表，线性相关系数均超过 0.96，分峰效果较好。

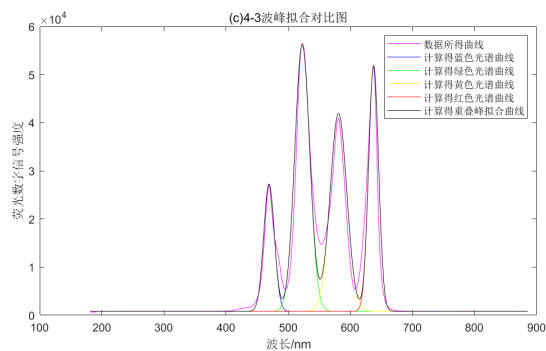
7.2.2 四峰复合光谱分峰模型



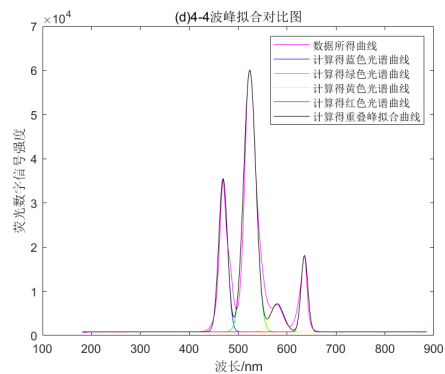
(a) 4-1 波峰数据散点图



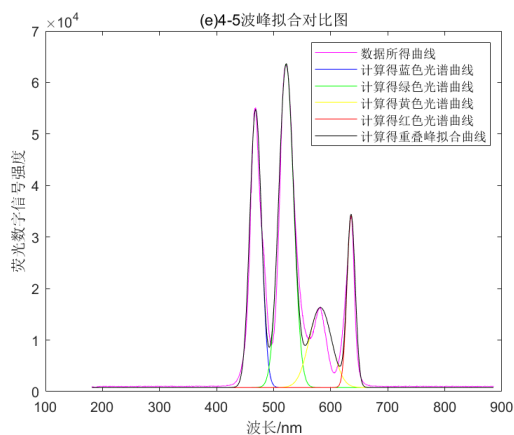
(b) 4-2 波峰数据散点图



(c) 4-3 波峰数据散点图



(d) 4-4 波峰数据散点图



(e) 4-5 波峰数据散点图

图 8: 附件 2 四峰拟合对比图

组别	线性相关系数			
4-1	0.981890272339212	0.988361262512770	0.993303698735636	0.946078922744599
4-2	0.993051688216376	0.988145148099476	0.993585659587470	0.961451936899554
4-3	0.977870586000329	0.984178088708726	0.992467781010006	0.967302559133360
4-4	0.979902760110044	0.988860682237004	0.993567689721240	0.953985389100263
4-5	0.990709862908278	0.989096003689477	0.935506690590112	0.971845763983573

表 8: 问题三三峰分峰线性相关系数

拟合得到的单峰与原始单峰线性相关系数如上表，线性相关系数均超过 0.93，分峰效果尚可。

8 问题四模型的建立与求解

8.1 数据处理

8.2 模型建立与求解

最小二乘拟合模型，得出拟合图像及线性相关系数如下（程序详见附件）

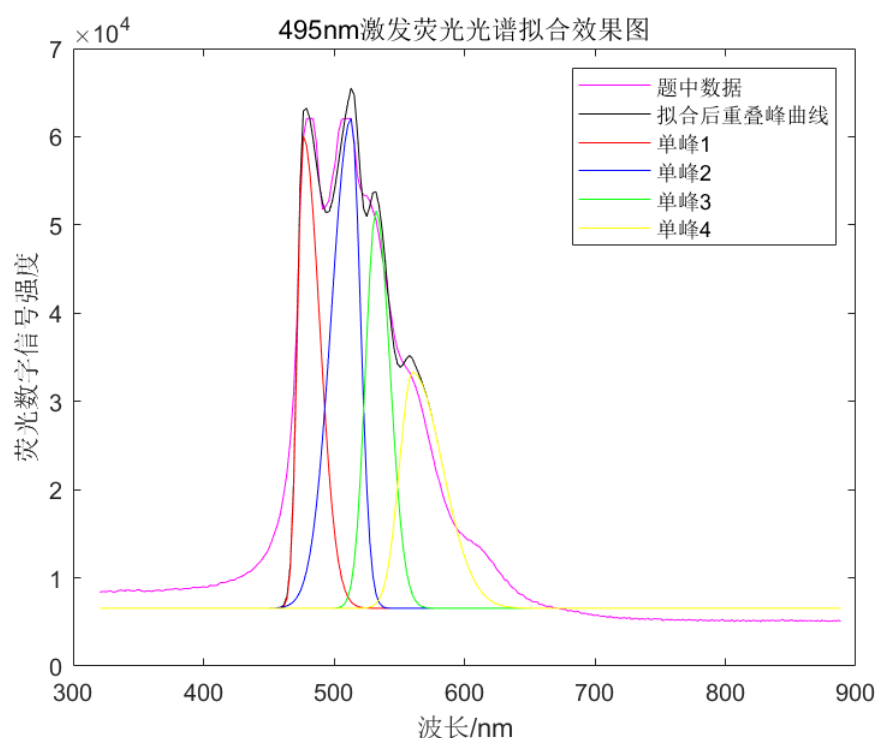


图 9: 问题二虚拟数据峰高比 1-9 散点图

线性相关系数为 0.984060259416489，拟合程度较好

9 参考文献

- [1] 胡耀垓, 张晓星, 赵正予, 冯波. 光谱重叠峰的曲线拟合解析策略与实现 [J]. 信息系统工程. 2012.5. 第 35 卷第 5 期. 76-82

附录

A 问题一相关数据及代码

A.1 数据处理 python 代码

```
1      #中心波长and峰强and半波宽
2  from optparse import Values
3  from turtle import distance
4  from scipy.signal import argrelextrema
5  from numpy import *
6  from pandas import *
7  from matplotlib.pyplot import *
8  from scipy import signal
9  for filenames in ['黄红','蓝绿','绿黄']:
10     a=read_excel(filenames+'.xlsx')
11     print(filenames)
12     b=a['波长 (nm) ']
13     c=a['荧光数字信号强度']
14     d=-c
15     excelarray=[]
16     #方法1
17     num_peak_3=signal.find_peaks(c,distance=100)
18     num_peak_4=signal.find_peaks(d,distance=100)
19     peak=[]
20     leastpeak=[]
21     dictionary={}
22     plot(b,c,'r',label='original values')
23     #极大值
24     for ii in range(len(num_peak_3[0])):
25         if c[num_peak_3[0][ii]]>5000:          #剔除坏点
26             plot(b[num_peak_3[0][ii]],c[num_peak_3[0][ii]], '*',markersize=10)
27             print('峰值的波长',b[num_peak_3[0][ii]], 'nm');print('峰值的荧光数字信号强度',c[←
                num_peak_3[0][ii]])
28             peak.append(num_peak_3[0][ii])      #波峰的索引
29             excelarray.append(['第{}个波峰的强度'.format(peak.index(num_peak_3[0][ii])+1),c[←
                num_peak_3[0][ii]])]
30             excelarray.append(['第{}个波峰的中心波长'.format(peak.index(num_peak_3[0][ii])+1),b[←
                num_peak_3[0][ii]])]
31     #极小值
32     for ii in range(len(num_peak_4[0])):
33         if d[num_peak_4[0][ii]]>-10000:        #剔除坏点
34             dictionary[d[num_peak_4[0][ii]]=num_peak_4[0][ii]          #用-极小值做键，索引做值
35             leastpeak.append(d[num_peak_4[0][ii]])
36     least_min=[min(leastpeak)]
37     plot(b[dictionary[least_min[0]]],-least_min[0], '*',markersize=10)
38     for i in range(len(least_min)):
39         plot(b[dictionary[least_min[i]]],-least_min[i], '*',markersize=10)
40         print('峰谷的波长',b[dictionary[least_min[i]]], 'nm');print('峰谷的荧光数字信号强度',-←
                least_min[i])
41         excelarray.append(['第{}个波谷的强度'.format(i+1),-least_min[i]])
```

```

42     c_base=c[0:100].mean()
43
44     for i in 0,1:
45         height=c[peak[i]]-c_base
46
47         xs=[x for x in range(len(c)) if (c[x]-c_base)>(height/2) ]
48         min_b_distance=abs(min(b[x] for x in xs)-b[peak[i]]);max_b_distance=abs(max(b[x] for x in xs)-b[peak[i]])
49         if min_b_distance <= max_b_distance: #取到正确的半波宽的对应横坐标 (波长) x
50             xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= min_b_distance]
51             bxs=[b[x] for x in xs]
52             cxs=[c[x] for x in xs]
53             # plot(min(b[xs]))
54             plot(bxs,cxs,'b-')
55             lambda_b=2*min_b_distance #根据高斯函数对称性得到的半波宽
56             print('第',i+1,'个峰的复合图像上的半峰宽为: ',lambda_b,'nm')
57         else:
58             xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= max_b_distance]
59             bxs=[b[x] for x in xs]
60             cxs=[c[x] for x in xs]
61             plot(bxs,cxs,'b-')
62             # plot(min(b[xs]),min(c[xs]),'x');plot(max(b[xs]),min(c[xs]),'x')
63             lambda_b=2*max_b_distance
64             print('第',i+1,'个峰的复合图像上的半峰宽为: ',lambda_b,'nm')
65             excelarray.append(['第{}个峰的半峰宽'.format(i+1),lambda_b])
66     rcParams['font.sans-serif']=['SimHei']
67     rcParams['axes.unicode_minus']=False
68     xlabel('波长(nm)')
69     ylabel('荧光数字信号强度')
70     name=filenames+'波峰特征值提取.png'
71     savefig(name)
72     df=DataFrame(excelarray,columns=['index','values'])
73     df.to_excel(filenames+'的特征值数据.xlsx')
74     show()

```

B 问题二相关数据及代码

B.1 数据处理 python 代码

(1) 创建单峰虚拟数据

```

1  #单峰
2  from cmath import sqrt
3  import numpy as np
4  import pandas as pd
5  import matplotlib.pyplot as plt
6  import openpyxl
7
8  def gauss(x,a,b,c): #只有峰强和波长是变量

```

```

9     y=a*np.exp(-(x-b)**2/(2*c**2))
10     return y
11 c1=22/(2*np.sqrt(np.log(4)))
12 c2=34/(2*np.sqrt(np.log(4)))
13 x=np.arange(100,700,0.6)
14 b1=400;b2=434
15 a1=np.arange(15000,140000,5000)
16 a2=15000
17 arrayt=[]
18
19 for i in a1:
20     y=[]
21     arrayt=[]
22     for j in x:
23         y.append(gauss(j,i,b1,c1))
24     for t in range(len(x)):
25         arrayt.append([x[t],float(y[t])])
26
27     df=pd.DataFrame(arrayt,columns=['波长 (nm) ','荧光数字信号强度'])
28     df.to_excel('峰高比{:.3f}的单峰1.xlsx'.format(i/a2),sheet_name='单峰1')
29     plt.plot(x,y)
30
31
32 for i in a1:
33     y=[]
34     arrayt=[]
35     for j in x:
36         y.append(gauss(j,a2,b2,c2))
37     for t in range(len(x)):
38         arrayt.append([x[t],float(y[t])])
39
40     df=pd.DataFrame(arrayt,columns=['波长 (nm) ','荧光数字信号强度'])
41     df.to_excel('峰高比{:.3f}的单峰2.xlsx'.format(i/a2),sheet_name='单峰2')
42     plt.plot(x,y)

```

(2) 创建重叠峰峰数据

```

1  from cmath import sqrt
2  import numpy as np
3  import pandas as pd
4  import matplotlib.pyplot as plt
5  import openpyxl
6
7  def gauss(x,a,b,c):          #只有峰强和波长是变量
8      y=a*np.exp(-(x-b)**2/(2*c**2))
9      return y
10 c1=22/(2*np.sqrt(np.log(4)))
11 c2=34/(2*np.sqrt(np.log(4)))
12 x=np.arange(100,700,0.6)
13 b1=400;b2=434
14 a1=np.arange(15000,140000,5000)
15 a2=15000

```

```

16 arrayt=[]
17
18 for i in a1:
19     y=[]
20     arrayt=[]
21     for j in x:
22         y.append(gauss(j,i,b1,c1)+gauss(j,a2,b2,c2))
23     for t in range(len(x)):
24         arrayt.append([x[t],float(y[t])])
25
26     df=pd.DataFrame(arrayt,columns=['波长 (nm) ','荧光数字信号强度'])
27     df.to_excel('峰高比{:.3f}.xlsx'.format(i/a2),sheet_name='复合光谱')
28     plt.plot(x,y)
29 plt.rcParams['font.sans-serif']=['SimHei']
30 plt.xlabel('波长(nm)')
31 plt.ylabel('荧光数字信号强度')
32 plt.savefig('峰高比1_to_9作图.png')
33 plt.show()

```

(3) 预估中心波长，峰高及半峰宽

```

1 from turtle import distance
2 from scipy.signal import argrelextrema
3 from numpy import *
4 from pandas import *
5 from matplotlib.pyplot import *
6 from scipy import signal
7 for filenames in ['峰高比1.000.xlsx','峰高比1.333.xlsx','峰高比1.667.xlsx','峰高比2.000.xlsx','峰高比2.333.xlsx','峰高比2.667.xlsx','峰高比3.000.xlsx','峰高比3.333.xlsx','峰高比3.667.xlsx','峰高比4.000.xlsx','峰高比4.333.xlsx','峰高比4.667.xlsx','峰高比5.000.xlsx','峰高比5.333.xlsx','峰高比5.667.xlsx','峰高比6.000.xlsx','峰高比6.333.xlsx','峰高比6.667.xlsx','峰高比7.000.xlsx','峰高比7.333.xlsx','峰高比7.667.xlsx','峰高比8.000.xlsx','峰高比8.333.xlsx','峰高比8.667.xlsx','峰高比9.000.xlsx']:
8
9     a=read_excel(filenames)
10    print(filenames)
11    b=a['波长 (nm) ']
12    c=a['荧光数字信号强度']
13    d=-c
14    excelarray=[]
15    #方法5
16    num_peak_3=signal.find_peaks(c,distance=50)
17    num_peak_4=signal.find_peaks(d,distance=50)
18    peak=[]
19    leastpeak=[]
20    dictionary={}
21    # plot(b,c,'r',label='original values')
22    #极大值
23    for ii in range(len(num_peak_3[0])):

```

```

23     if c[num_peak_3[0][ii]]>5000:                #剔除坏点
24         # plot(b[num_peak_3[0][ii]],c[num_peak_3[0][ii]], '*', markersize=10)
25         print('峰值的波长',b[num_peak_3[0][ii]], 'nm');print('峰值的荧光数字信号强度',c[←
                                     num_peak_3[0][ii]])
26         peak.append(num_peak_3[0][ii])          #波峰的索引
27         excelarray.append(['第{}个波峰的强度'.format(peak.index(num_peak_3[0][ii])+1),c[←
                                     num_peak_3[0][ii]])
28         excelarray.append(['第{}个波峰的中心波长'.format(peak.index(num_peak_3[0][ii])+1),b[←
                                     num_peak_3[0][ii]])
29 #极小值
30 for ii in range(len(num_peak_4[0])):
31     if d[num_peak_4[0][ii]]>-20000:                #剔除坏点
32         dictionary[d[num_peak_4[0][ii]]=num_peak_4[0][ii]        #用-极小值做键，索引做值
33         leastpeak.append(d[num_peak_4[0][ii]])
34     least_min=[min(leastpeak)]
35     # plot(b[dictionary[least_min]],-least_min, '*', markersize=10)
36     for i in range(len(least_min)):
37         plot(b[dictionary[least_min[i]],-least_min[i], '*', markersize=10)
38         print('峰谷的波长',b[dictionary[least_min[i]], 'nm');print('峰谷的荧光数字信号强度',-←
                                     least_min[i])
39         excelarray.append(['第{}个波谷的强度'.format(i+1),-least_min[i]])
40     c_base=c[0:100].mean()
41
42     for i in 0,1:
43         height=c[peak[i]]-c_base
44
45         xs=[x for x in range(len(c)) if (c[x]-c_base)>(height/2) ]
46         min_b_distance=abs(min(b[x] for x in xs)-b[peak[i]]);max_b_distance=abs(max(b[x] for x in ←
                                     xs)-b[peak[i]])
47         if min_b_distance <= max_b_distance:        #取到正确的半波宽的对应横坐标（波长）x
48             xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= min_b_distance]
49             bxs=[b[x] for x in xs]
50             cxs=[c[x] for x in xs]
51             plot(bxs,cxs, 'g-')
52             lambda_b=2*min_b_distance
53             print('第',i+1,'个峰的复合图像上的半波宽为： ',lambda_b,'nm')
54         else:
55             xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= max_b_distance]
56             bxs=[b[x] for x in xs]
57             cxs=[c[x] for x in xs]
58             plot(bxs,cxs, 'b-')
59             lambda_b=2*max_b_distance
60             print('第',i+1,'个峰的复合图像上的半波宽为： ',lambda_b,'nm')
61         excelarray.append(['第{}个峰的半峰宽'.format(i+1),lambda_b])
62     df=DataFrame(excelarray, columns=['index', 'values'])
63     df.to_excel(filename+'的特征值数据.xlsx')

```

C 问题三相关数据及代码

C.1 数据处理 python 代码

(1) 三峰复合光谱分峰

```
1 from fileinput import filename
2 from turtle import distance
3 from unicodedata import name
4 from scipy.signal import argrelextrema
5 from numpy import *
6 from pandas import *
7 from matplotlib.pyplot import *
8 from scipy import signal
9 import heapq
10 for filenames in ['蓝绿黄','绿黄红']:
11     a=read_excel(filenames+'.xlsx')
12     print(filenames)
13     b=a['波长 (nm) ']
14
15     c=a['荧光数字信号强度']
16     d=-c
17     #方法
18     num_peak_3=signal.find_peaks(c,distance=50)
19     num_peak_4=signal.find_peaks(d,distance=20)
20     peak=[]
21     leastpeak=[]
22     excelarray=[]
23     dictionary={}
24     plot(b,c,'r',label='original values')
25     #极大值
26     for ii in range(len(num_peak_3[0])):
27         if c[num_peak_3[0][ii]]>20000: #剔除坏点
28             plot(b[num_peak_3[0][ii]],c[num_peak_3[0][ii]], '*',markersize=10)
29             print('峰值的波长',b[num_peak_3[0][ii]], 'nm');print('峰值的荧光数字信号强度',c[←
30                                     num_peak_3[0][ii]])
31             peak.append(num_peak_3[0][ii]) #波峰的索引
32             excelarray.append(['第{}个波峰的强度'.format(peak.index(num_peak_3[0][ii])+1),c[←
33                                     num_peak_3[0][ii]])]
34             excelarray.append(['第{}个波峰的中心波长'.format(peak.index(num_peak_3[0][ii])+1),b[←
35                                     num_peak_3[0][ii]])]
36
37     #极小值
38     for ii in range(len(num_peak_4[0])):
39         if d[num_peak_4[0][ii]]<=-5000 : #剔除坏点
40             dictionary[d[num_peak_4[0][ii]]=num_peak_4[0][ii] #用-极小值做键，索引做值
41             leastpeak.append(d[num_peak_4[0][ii]])
42     least_min=heapq.nlargest(2,leastpeak)
43     for i in range(len(least_min)):
44         plot(b[dictionary[least_min[i]]],-least_min[i], '*',markersize=10)
45         print('峰谷的波长',b[dictionary[least_min[i]]], 'nm');print('峰谷的荧光数字信号强度',-←
46                                     least_min[i])
47         excelarray.append(['第{}个波谷的强度'.format(i+1),-least_min[i]])
48     c_base=c[1:100].mean()
```

```

44     for i in range(len(peak)):
45         height=c[peak[i]]-c_base
46
47         xs=[x for x in range(len(c)) if (c[x]-c_base)>(height/2) ]
48         min_b_distance=min(abs(min(b[x] for x in xs)-b[peak[i]]),abs(b[peak[i]]-b[dictionary[←
                                     least_min[0]]]),abs(b[peak[i]]-b[←
                                     dictionary[least_min[1]]]))
49         max_b_distance=min(abs(max(b[x] for x in xs)-b[peak[i]]),abs(b[peak[i]]-b[dictionary[←
                                     least_min[0]]]),abs(b[peak[i]]-b[←
                                     dictionary[least_min[1]]]))
50         if min_b_distance <= max_b_distance:           #取到正确的半波宽的对应横坐标 (波长) x
51             xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= min_b_distance]
52             bxs=[b[x] for x in xs]
53             cxs=[c[x] for x in xs]
54             plot(bxs,cxs,'b-')
55             t=min(min_b_distance,abs(min(bxs)-b[peak[i]]))
56             lambda_b=2*t
57             print('第',i+1,'个峰的复合图像上的半波宽为: ',lambda_b,'nm')
58         else:
59             xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= max_b_distance]
60             bxs=[b[x] for x in xs]
61             cxs=[c[x] for x in xs]
62             plot(bxs,cxs,'b-')
63             t=min(max_b_distance,abs(min(bxs)-b[peak[i]]))
64             lambda_b=2*t
65             print('第',i+1,'个峰的复合图像上的半波宽为: ',lambda_b,'nm')
66             excelarray.append(['第{}个峰的半峰宽'.format(i+1),lambda_b])
67         rcParams['font.sans-serif']='SimHei'
68         rcParams['axes.unicode_minus']=False
69
70         xlabel('波长(nm)')
71         ylabel('荧光数字信号强度')
72         name=filenames + '的波峰特征值提取.png'
73         savefig(name)
74
75         df=DataFrame(excelarray,columns=['index','values'])
76         df.to_excel(filenames+'的特征值数据.xlsx')
77         show()

```

(2) 四峰复合光谱分峰

```

1  from turtle import distance
2  from scipy.signal import argrelextrema
3  from numpy import *
4  from pandas import *
5  from matplotlib.pyplot import *
6  from scipy import signal
7  import heapq
8  for filenames in ['4-1','4-2','4-3','4-4']:
9      a=read_excel(filenames+'.xlsx')
10     print(filenames)
11     b=a['波长 (nm) ']

```



```

12 c=a['荧光数字信号强度']
13 d=-c
14 excelarray=[]
15 #方法1
16 num_peak_3=signal.find_peaks(c,distance=50)
17 num_peak_4=signal.find_peaks(d,distance=20)
18 peak=[]
19 leastpeak=[]
20 dictionary={}
21 plot(b,c,'r',label='original values')
22 #极大值
23 for ii in range(len(num_peak_3[0])):
24     if c[num_peak_3[0][ii]]>6000: #剔除坏点
25         plot(b[num_peak_3[0][ii]],c[num_peak_3[0][ii]], '*',markersize=10)
26         print('峰值的波长',b[num_peak_3[0][ii]], 'nm');print('峰值的荧光数字信号强度',c[←
                num_peak_3[0][ii]])
27         peak.append(num_peak_3[0][ii]) #波峰的索引
28         excelarray.append(['第{}个波峰的强度'.format(peak.index(num_peak_3[0][ii])+1),c[←
                num_peak_3[0][ii]])]
29         excelarray.append(['第{}个波峰的中心波长'.format(peak.index(num_peak_3[0][ii])+1),b[←
                num_peak_3[0][ii]])]
30 #极小值
31 for ii in range(len(num_peak_4[0])):
32     if d[num_peak_4[0][ii]]<=-2000 : #剔除坏点
33         dictionary[d[num_peak_4[0][ii]]=num_peak_4[0][ii] #用-极小值做键，索引做值
34         leastpeak.append(d[num_peak_4[0][ii]])
35 least_min=heapq.nlargest(3,leastpeak)
36 for i in range(len(least_min)):
37     plot(b[dictionary[least_min[i]],-least_min[i]], '*',markersize=10)
38     print('峰谷的波长',b[dictionary[least_min[i]]], 'nm');print('峰谷的荧光数字信号强度',-←
            least_min[i])
39     excelarray.append(['第{}个波谷的强度'.format(i+1),-least_min[i]])
40 c_base=c[1:100].mean()
41 for i in range(len(peak)):
42     height=c[peak[i]]-c_base
43
44     xs=[x for x in range(len(c)) if (c[x]-c_base)>(height/2) ]
45     min_b_distance=min(abs(min(b[x] for x in xs)-b[peak[i]]),abs(b[peak[i]]-b[dictionary[←
            least_min[0]]]),abs(b[peak[i]]-b[←
            dictionary[least_min[1]]]),abs(b[peak[i]←
            ]-b[dictionary[least_min[2]]]))
46     max_b_distance=min(abs(max(b[x] for x in xs)-b[peak[i]]),abs(b[peak[i]]-b[dictionary[←
            least_min[0]]]),abs(b[peak[i]]-b[←
            dictionary[least_min[1]]]),abs(b[peak[i]←
            ]-b[dictionary[least_min[2]]]))
47     if min_b_distance <= max_b_distance: #取到正确的半波宽的对应横坐标（波长）x
48         xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= min_b_distance]
49         bxs=[b[x] for x in xs]
50         cxs=[c[x] for x in xs]
51         plot(bxs,cxs,'b-')
52         t=min(min_b_distance,abs(min(bxs)-b[peak[i]]))
53         lambda_b=2*t
54         print('第',i+1,'个峰的复合图像上的半波宽为：',lambda_b,'nm')
55     else:

```

```

56         xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= max_b_distance]
57         bxs=[b[x] for x in xs]
58         cxs=[c[x] for x in xs]
59         plot(bxs,cxs,'b-')
60         t=min(max_b_distance,abs(min(bxs)-b[peak[i]]))
61         lambda_b=2*t
62         print('第',i+1,'个峰的复合图像上的半波宽为:',lambda_b,'nm')
63         excelarray.append(['第{}个峰的半峰宽'.format(i+1),lambda_b])
64         rcParams['font.sans-serif']='SimHei'
65         xlabel('波长(nm)')
66         ylabel('荧光数字信号强度')
67         name=filenames + '的波峰特征值提取.png'
68         savefig(name)
69         df=DataFrame(excelarray,columns=['index','values'])
70         df.to_excel(filenames+'的特征值数据.xlsx')
71         show()
72 a=read_excel('4-5.xlsx')
73 print('4-5.xlsx')
74 b=a['波长 (nm) ']
75 c=a['荧光数字信号强度']
76 d=-c
77 #取得特征值
78 num_peak_3=sig.find_peaks(c,distance=100)
79 num_peak_4=sig.find_peaks(d,distance=50)
80 peak=[]
81 leastpeak=[]
82 excelarray=[]
83 dictionary={}
84 plot(b,c,'r',label='original values')
85 #极大值
86 for ii in range(len(num_peak_3[0])):
87     if c[num_peak_3[0][ii]]>6000: #剔除坏点
88         plot(b[num_peak_3[0][ii]],c[num_peak_3[0][ii]], '*',markersize=10)
89         print('峰值的波长',b[num_peak_3[0][ii]], 'nm');print('峰值的荧光数字信号强度',c[num_peak_3[0][ii]])
90         peak.append(num_peak_3[0][ii]) #波峰的索引
91         excelarray.append(['第{}个波峰的强度'.format(peak.index(num_peak_3[0][ii])+1),c[num_peak_3[0][ii]]])
92         excelarray.append(['第{}个波峰的中心波长'.format(peak.index(num_peak_3[0][ii])+1),b[num_peak_3[0][ii]]])
93 #极小值
94 for ii in range(len(num_peak_4[0])):
95     if d[num_peak_4[0][ii]]<-2000 and b[num_peak_4[0][ii]]>min(b[peak]): #剔除坏点
96         dictionary[d[num_peak_4[0][ii]]]=num_peak_4[0][ii] #用-极小值做键,索引做值
97         leastpeak.append(d[num_peak_4[0][ii]])
98 least_min=heapq.nlargest(3,leastpeak)
99 for i in range(len(least_min)):
100     plot(b[dictionary[least_min[i]]],-least_min[i], '*',markersize=10)
101     print('峰谷的波长',b[dictionary[least_min[i]]], 'nm');print('峰谷的荧光数字信号强度',-least_min[i])
102     excelarray.append(['第{}个波谷的强度'.format(i+1),-least_min[i]])
103 c_base=c[1:100].mean()
104 for i in range(len(peak)):
105     height=c[peak[i]]-c_base

```

```

106
107     xs=[x for x in range(len(c)) if (c[x]-c_base)>(height/2) ]
108     min_b_distance=min(abs(min(b[x] for x in xs)-b[peak[i]]),abs(b[peak[i]]-b[dictionary[least_min←
[0]]]),abs(b[peak[i]]-b[dictionary[least_min←
[1]]]),abs(b[peak[i]]-b[dictionary[least_min←
[2]]]))
109     max_b_distance=min(abs(max(b[x] for x in xs)-b[peak[i]]),abs(b[peak[i]]-b[dictionary[least_min←
[0]]]),abs(b[peak[i]]-b[dictionary[least_min←
[1]]]),abs(b[peak[i]]-b[dictionary[least_min←
[2]]]))
110     if min_b_distance <= max_b_distance:           #取到正确的半波宽的对应横坐标（波长）x
111         xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= min_b_distance]
112         bxs=[b[x] for x in xs]
113         cxs=[c[x] for x in xs]
114         plot(bxs,cxs,'b-')
115         t=min(min_b_distance,abs(min(bxs)-b[peak[i]]))
116         lambda_b=2*t
117         print('第',i+1,'个峰的复合图像上的半波宽为：',lambda_b,'nm')
118     else:
119         xs=[x for x in xs if abs(b[x]-b[peak[i]]) <= max_b_distance]
120         bxs=[b[x] for x in xs]
121         cxs=[c[x] for x in xs]
122         plot(bxs,cxs,'b-')
123         t=min(max_b_distance,abs(min(bxs)-b[peak[i]]))
124         lambda_b=2*t
125         print('第',i+1,'个峰的复合图像上的半波宽为：',lambda_b,'nm')
126         excelarray.append(['第{}个峰的半峰宽'.format(i+1),lambda_b])
127 rcParams['font.sans-serif']='SimHei'
128 xlabel('波长(nm)')
129 ylabel('荧光数字信号强度')
130 savefig('4-5的波峰特征值提取.png')
131 df=DataFrame(excelarray,columns=['index','values'])
132 df.to_excel('4-5的特征值数据.xlsx')
133 show()

```

D 问题四相关数据及代码

D.1 数据处理 python 代码

```

1 from turtle import distance
2 from scipy.signal import argrelextrema
3 from numpy import *
4 from pandas import *
5 from matplotlib.pyplot import *
6 from scipy import signal
7 import heapq
8
9 a=read_excel('partial_twice.xlsx')
10 b=a['波长']

```

```

11 c=a['二次处理后的峰强']
12 d=-c
13 #方法1
14 num_peak_3=signal.find_peaks(c,distance=5)
15 num_peak_4=signal.find_peaks(d,distance=5)
16 peak=[]
17 leastpeak=[]
18 dictionary={}
19 excelarray=[]
20 new_peaklist=[]
21 plot(b,c,'r',label='original values')
22 #极大值
23 for ii in range(len(num_peak_3[0])):
24     if c[num_peak_3[0][ii]]>40: #剔除坏点
25         new_peaklist.append(num_peak_3[0][ii])
26         plot(b[num_peak_3[0][ii]],c[num_peak_3[0][ii]], '*',markersize=10)
27         print('峰值{}的波长'.format(new_peaklist.index(num_peak_3[0][ii])+1),b[num_peak_3[0][ii]],←
                'nm');print('峰值的荧光数字信号强度',c[←
                num_peak_3[0][ii]])
28         peak.append(num_peak_3[0][ii]) #波峰的索引
29         excelarray.append(['第{}个波峰的强度'.format(peak.index(num_peak_3[0][ii])+1),c[num_peak_3[←
                0][ii]])]
30         excelarray.append(['第{}个波峰的中心波长'.format(peak.index(num_peak_3[0][ii])+1),b[←
                num_peak_3[0][ii]])]
31 #极小值
32 for ii in range(len(num_peak_4[0])):
33     if d[num_peak_4[0][ii]]>-50 and b[num_peak_4[0][ii]]>min(b[peak]): #剔除坏点
34         dictionary[d[num_peak_4[0][ii]]]=num_peak_4[0][ii] #用-极小值做键,索引做值
35         leastpeak.append(d[num_peak_4[0][ii]])
36 least_min=heapq.nlargest(5,leastpeak) #取到三个波谷
37 for i in range(len(least_min)):
38
39     plot(b[dictionary[least_min[i]]],-least_min[i], '*',markersize=10)
40     print('第{}个峰谷的波长'.format(i+1),b[dictionary[least_min[i]]], 'nm');print('峰谷的荧光数←
                字信号强度',-least_min[i])
41     excelarray.append(['第{}个波谷的强度'.format(i+1),-least_min[i]])
42     excelarray.append(['第{}个波谷的中心波长'.format(i+1),b[dictionary[least_min[i]]]])
43 xlabel('波长')
44 ylabel('二次微分处理后的信号强度')
45 title('复合光谱信号强度函数的二阶微分')
46 rcParams['font.sans-serif']=['SimHei']
47 rcParams['axes.unicode_minus']=False
48 df=DataFrame(excelarray,columns=['index','values'])
49 df.to_excel('partial_twice'+的极大值极小值数据.xlsx')
50 show()

```

E 也可以插入 matlab 代码

1 %论文1-熵权值求权重

```

2      A=[58 38 14 8 57 10
3      50 45 11 9 52 12
4      42 47 8 12 50 15
5      45 42 12 15 46 16
6      47 44 13 10 49 13];%统计数据A 5x6
7      [m,n]=size(A);
8      h =max(A);%每一列最大值
9      H= repmat(h,m,1);%向量h排列mx1的矩阵
10     Mij=A./H;%隶属度矩阵
11
12
13     A2=sum(Mij);%Mij矩阵每一列求和
14     A22= repmat(A2,m,1);%向量A2排列mx1的矩阵
15     Qij=Mij./A22;%隶属度矩阵每一列标准化
16
17
18     Fj=-1/log(5)*Qij.*log(Qij);
19     fj=sum(Fj);%熵值
20     dj=1-fj;%差异系数
21     v=dj/sum(dj)%权重系数
22     B=Qij*v';%最终评价
23     disp(B)%显示结果

```
