

批量画图

1 功能解释

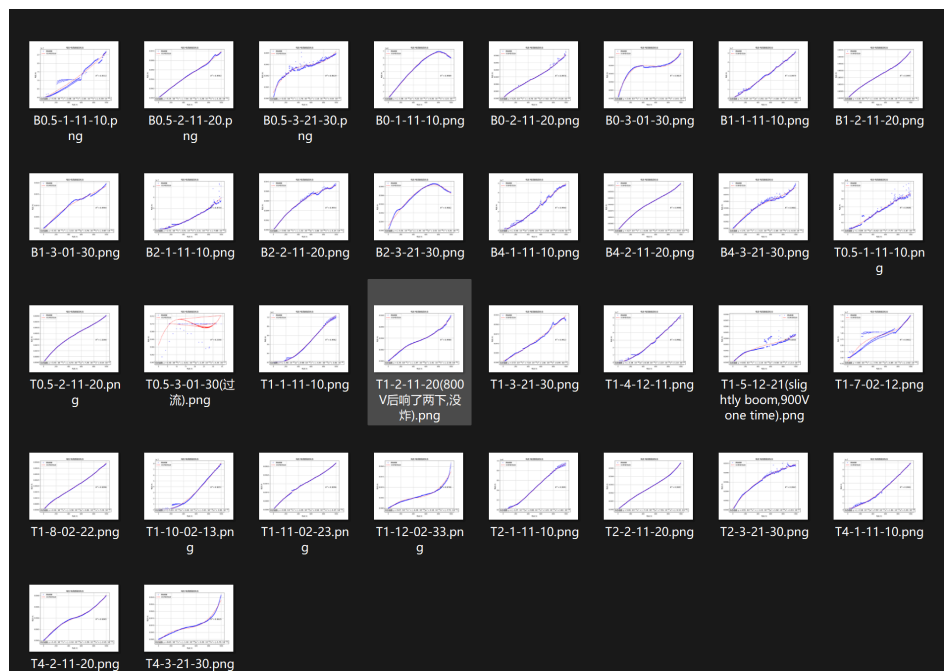
- 1 `batch_analyze_data('./datas_learn/old_34', './datas_learn/old_34/old34_img', line_style='--', line_color='r', connect_points=False)`
 - 将 old_34 目录下的所有xlsx表格都读取
 - 并以原数据画图 (`connect_points=False` 表示不用线连接原数据点. 只在图中描点)
 - 图中附带多项式拟合曲线(`line_style='--', line_color='r'`表示拟合曲线为虚线,红色)
 - 并将所有图像存储与old_34文件夹中的old34_img中(程序会自动创建)
 - `batch_analyze_data` 函数中引用了函数 `analyze_data_no_display` (`no_display` 后缀表示画图, 但不会弹出窗口让你看)
 - `analyze_data_no_display` 函数是 读取原数据描点画图 + 根据数据进行多项式曲线拟合 的二合一包装
 - 注意: 该函数不能实现迭代拟合, 因为我初步考虑是, 这个画图阶段只是初步展示, 后续再加入迭代拟合
 - 简单的迭代拟合已经实现, 不过对于一些奇葩数据效果实在不好

2 效果展示

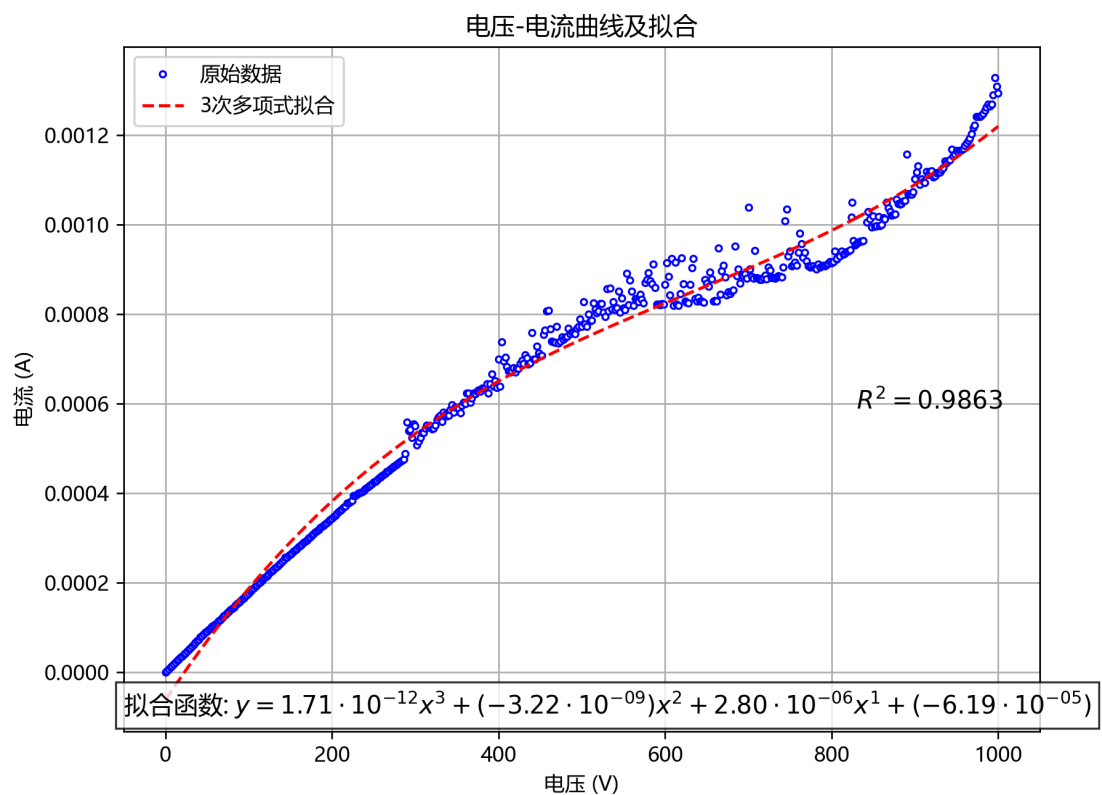
- 对于代码
 - 1 `batch_analyze_data('./datas_learn/old_34', './datas_learn/old_34/old34_img', line_style='--', line_color='r', connect_points=False)`
- 原数据(old_34文件夹)

名称	修改日期	类型	大小
B0.5-2-11-20.xlsx	1/23 星期四 15:18	Microsoft Excel 工...	26 KB
B0.5-3-21-30.xlsx	1/23 星期四 15:19	Microsoft Excel 工...	26 KB
B0-1-11-10.xlsx	1/23 星期四 15:09	Microsoft Excel 工...	26 KB
B0-2-11-20.xlsx	1/23 星期四 15:11	Microsoft Excel 工...	26 KB
B0-3-01-30.xlsx	1/23 星期四 15:12	Microsoft Excel 工...	26 KB
B1-1-11-10.xlsx	1/23 星期四 15:21	Microsoft Excel 工...	26 KB
B1-2-11-20.xlsx	1/23 星期四 15:23	Microsoft Excel 工...	26 KB
B1-3-01-30.xlsx	1/23 星期四 15:25	Microsoft Excel 工...	26 KB
B2-1-11-10.xlsx	1/23 星期四 15:28	Microsoft Excel 工...	26 KB
B2-2-11-20.xlsx	1/23 星期四 15:30	Microsoft Excel 工...	26 KB
B2-3-21-30.xlsx	1/23 星期四 15:31	Microsoft Excel 工...	26 KB
B4-1-11-10.xlsx	1/23 星期四 15:34	Microsoft Excel 工...	26 KB
B4-2-11-20.xlsx	1/23 星期四 15:35	Microsoft Excel 工...	26 KB
B4-3-21-30.xlsx	1/23 星期四 15:37	Microsoft Excel 工...	26 KB
T0.5-1-11-10.xlsx	1/23 星期四 14:27	Microsoft Excel 工...	26 KB
T0.5-2-11-20.xlsx	1/23 星期四 14:30	Microsoft Excel 工...	26 KB
T0.5-3-01-30(过流).xlsx	1/23 星期四 14:31	Microsoft Excel 工...	24 KB
T1-1-11-10.xlsx	1/23 星期四 14:35	Microsoft Excel 工...	26 KB
T1-2-11-20(800V后躺了两下,没炸).xlsx	1/23 星期四 14:37	Microsoft Excel 工...	26 KB
T1-3-21-30.xlsx	1/23 星期四 14:39	Microsoft Excel 工...	26 KB
T1-4-12-11.xlsx	1/23 星期四 14:42	Microsoft Excel 工...	26 KB
T1-5-12-21(slightly boom,900V one time).xlsx	1/23 星期四 14:44	Microsoft Excel 工...	26 KB
T1-7-02-12.xlsx	1/23 星期四 14:49	Microsoft Excel 工...	26 KB

- 运行程序后, old_34文件夹中的 old34_img文件夹:



- 单个图片展示:(dpi=240) --- 由于渲染pdf有宽度限制, 所以可能图片不是原始大小



3 全代码

```

1 import os
2 import pandas as pd
3 import numpy as np
4 import matplotlib
5 import matplotlib.pyplot as plt
6
7

```

```

8 # 设置中文字体, 解决字体显示问题
9 matplotlib.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 例如使用微软雅黑 (SimHei)
10 matplotlib.rcParams['axes.unicode_minus'] = False # 解决负号显示问题
11
12
13 def analyze_data_no_display(file_path, degree=3, show_equation=True, show_r_squared=True,
14                             line_style='-', line_color='b',
15                             line_width=1.5, connect_points=True):
16     """
17     与 analyze_data 功能相同, 但不显示图像。
18     """
19     # 读取 Excel 文件
20     xls = pd.ExcelFile(file_path)
21     df = xls.parse(sheet_name=0)
22
23     # 提取 C35 到 D536 的数据 (假设列名在第 35 行开始)
24     df_filtered = df.iloc[34:536, [2, 3]].dropna() # 选择 C 和 D 列 (0-based 索引)
25
26     # 重新命名列名
27     df_filtered.columns = ["电压 (V)", "电流 (A)"]
28
29     # 转换数据为数值类型 (防止字符串干扰)
30     df_filtered = df_filtered.astype(float)
31
32     # 提取自变量 (电压) 和因变量 (电流)
33     x = df_filtered["电压 (V)"]
34     y = df_filtered["电流 (A)"]
35
36     # 多项式拟合函数
37     def polynomial_fit(x, y, degree=3):
38         p = np.polyfit(x, y, degree) # 拟合多项式
39         poly = np.poly1d(p)
40         y_fit = poly(x)
41         return p, y_fit
42
43     # 调用多项式拟合
44     p, y_fit = polynomial_fit(x, y, degree)
45
46     # 准备拟合函数的字符串表达式 (优化科学计数法显示)
47     equation_terms = []
48     for i, coef in enumerate(p):
49         power = degree - i
50         # 将python中 "e数字" 的科学计数法格式更改为latex形式
51         coef_str = f"{coef:.2e}".replace("e", "\\cdot 10^{") + ")"
52         # 处理负号和小数点
53         if coef < 0:
54             coef_str = f"({coef_str})"
55         # 添加 x 的幂次
56         if power == 0:
57             term = coef_str
58         else:
59             term = f"{coef_str}x^{{{power}}}"
60         equation_terms.append(term) # 使用 {{}} 包裹幂次 防止出现x~10的情况出现 (LaTeX规范: x^{10})
61
62     # 拼接拟合函数表达式

```

```

62     equation_str = " + ".join(equation_terms)
63
64     # 计算  $R^2$  (决定系数)
65     ss_residual = np.sum((y - y_fit) ** 2) # 残差平方和
66     ss_total = np.sum((y - np.mean(y)) ** 2) # 总平方和
67     r_squared = 1 - (ss_residual / ss_total)
68
69     # 绘制数据和拟合曲线
70     plt.figure(figsize=(8, 6))
71
72     # 设置原始数据点的样式和连接线的样式
73     if connect_points:
74         plt.plot(x, y, marker='o', linestyle='-', color='b', label="原始数据",
75 markersize=3,
76                 markerfacecolor='white', markeredgewidth=1, linewidth=2) # 数据点为白
77     else:
78         plt.plot(x, y, marker='o', linestyle='', color='b', label="原始数据",
79 markersize=3,
80                 markerfacecolor='white', markeredgewidth=1) # 数据点为白色, 不连接
81
82     # 绘制拟合曲线
83     plt.plot(x, y_fit, linestyle=line_style, color=line_color, label=f"{degree}次多项式拟
84 合", linewidth=line_width)
85
86     # 当 show_equation == True, 在图形中添加拟合函数的表达式 (使用 LaTeX 样式)
87     if show_equation:
88         plt.text(0.0, 0.06, f"拟合函数:  $y = {equation\_str}$ ",
89 transform=plt.gca().transAxes, fontsize=12,
90                 verticalalignment='top', ha='left', bbox=dict(facecolor='white',
91 alpha=0.8))
92
93     # 当 show_r_squared == True, 在图形中添加  $R^2$  值, 使用 LaTeX 样式
94     if show_r_squared:
95         plt.text(0.8, 0.5, f" $R^2 = {r\_squared:.4f}$ ", transform=plt.gca().transAxes,
96 fontsize=12,
97                 verticalalignment='top', ha='left')
98
99     # 图表设置
100     plt.xlabel("电压 (V)")
101     plt.ylabel("电流 (A)")
102     plt.title("电压-电流曲线及拟合")
103     plt.legend()
104     plt.grid(True)
105
106     # 返回图像对象
107     return plt.gcf()
108
109 def batch_analyze_data(folder_path, output_folder, degree=3, show_equation=True,
110 show_r_squared=True, line_style='-', line_color='b',
111                        line_width=1.5, connect_points=False):
112     """
113     批量处理文件夹中的 Excel 文件, 进行多项式拟合并保存图像。
114
115     参数:
116     - folder_path: 包含 Excel 文件的文件夹路径。
117 
```

```

110     - output_folder: 保存图像的文件夹路径。
111     - degree: 多项式拟合的次数, 默认3。
112     - show_equation: 是否在图中显示拟合函数表达式, 默认显示。
113     - show_r_squared: 是否显示决定系数 $R^2$ , 默认显示。
114     - line_style: 曲线的线型, 默认为 '-' (实线)。
115     - line_color: 曲线的颜色, 默认为 'b' (蓝色)。
116     - line_width: 曲线的线宽, 默认为 1.5。
117     - connect_points: 是否连接原始数据点, 默认不连接。
118     """
119     # 创建输出文件夹
120     if not os.path.exists(output_folder):
121         os.makedirs(output_folder)
122
123     # 遍历文件夹中的所有 Excel 文件
124     for file_name in os.listdir(folder_path):
125         if file_name.endswith('.xlsx') or file_name.endswith('.xls'):
126             file_path = os.path.join(folder_path, file_name)
127             # 调用 analyze_data_no_display 函数生成图像
128             fig = analyze_data_no_display(file_path, degree, show_equation,
show_r_squared, line_style, line_color,
129                                         line_width, connect_points)
130
131             # 保存图像
132             output_file_path = os.path.join(output_folder, f"{os.path.splitext(file_name)
[0]}.png")
133             fig.savefig(output_file_path, dpi=240)
134             plt.close(fig)
135             print(f"已保存图像: {output_file_path}")
136             print('分析完成.')
137
138     # 示例调用
139     # batch_analyze_data('./datas', './datas/datas_img')
140     batch_analyze_data('./datas_learn/old_34', './datas_learn/old_34/old34_img',
line_style='--', line_color='r', connect_points=False)
141
142

```