浙江大学量子化学实验

量子化学计算中的红外光谱

实
验
报
告

参加学生：叶青杨（3210100360）

指导老师：刘迎春

浙江大学化学实验教学中心

2023 年 09 月 27 日

量子化学计算中的红外光谱

叶青杨 (3210100360), 求化 2101，指导教师：刘迎春

1 实验目的

1.1 对方法和基组优化参数

1.2 计算指定物质的 IR 并分析

2 实验步骤

本次实验进行的操作和计算如下：

Table 1 展示了共对 36 个组合完成的 38 次计算。

extract.py(见附录，下同) 用于批量提取 log 文件中的能量信息至 csv 表格中。

Figure 1、Figure 2、Figure 3 由 draw.py 生成，展示了不同化合物、方法和基组计算出的能量的对比（单位为 Hartree）。

Figure 4、Figure 5、Figure 6 的图来自物理化学数据库和计算得到的 log 文件，展示了使用 Semi-empirical PM6 以及 DFT B3LYP 6-311G 计算出的 IR 以及和标准实验谱的对比。

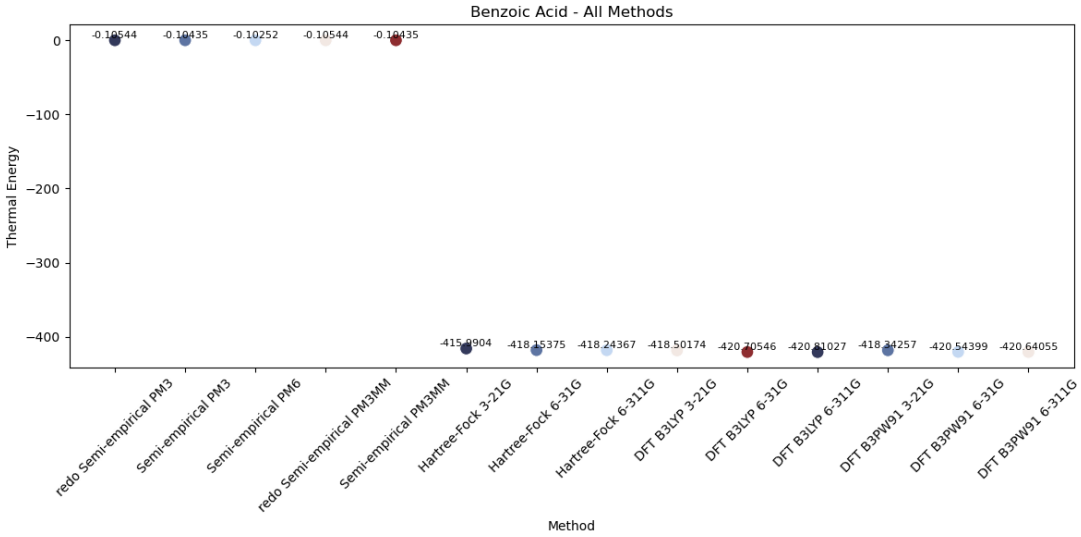
脚本 trans.sh 用于在服务器上批量把 chk 文件处理为 fchk 文件。

3 实验结果

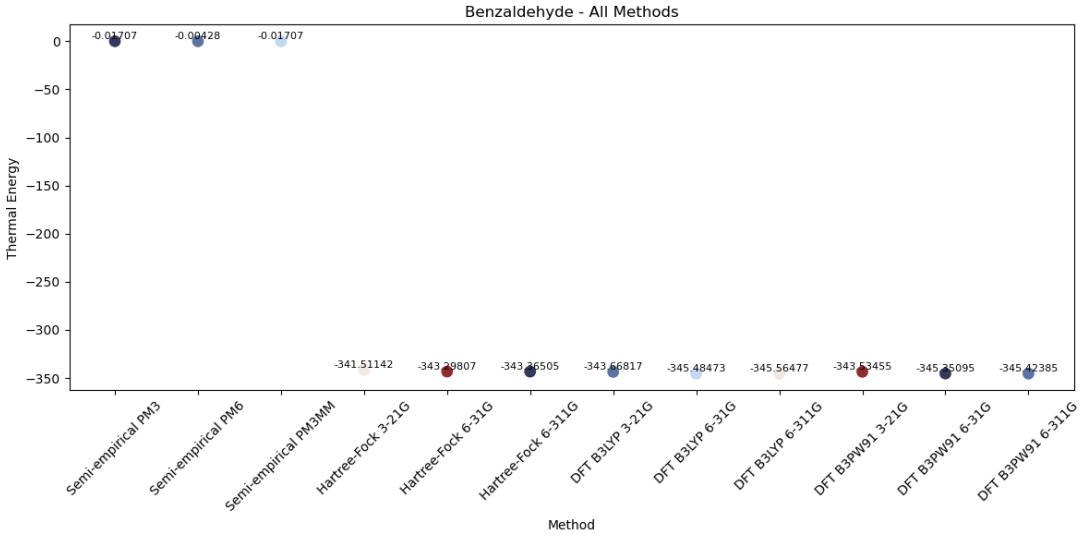
Molecule	Method	Basis Set/Note
苯甲酸	Semi-empirical	PM3 redo PM3 (Initial guess symmetric) PM6 PM3MM redo PM3MM (Initial guess symmetric)
	Hartree-Fock	3-21G 6-31G 6-311G
	DFT B3LYP	3-21G 6-31G 6-311G
	DFT B3PW91	3-21G 6-31G 6-311G
苯甲醛	Semi-empirical	PM3 PM6 PM3MM
	Hartree-Fock	3-21G 6-31G 6-311G
	DFT B3LYP	3-21G 6-31G 6-311G
	DFT B3PW91	3-21G 6-31G 6-311G
苯甲醇	Semi-empirical	PM3 PM6 PM3MM
	Hartree-Fock	3-21G 6-31G 6-311G
	DFT B3LYP	3-21G 6-31G 6-311G
	DFT B3PW91	3-21G 6-31G 6-311G

Table 1: 各种分子的计算方法和基组总结

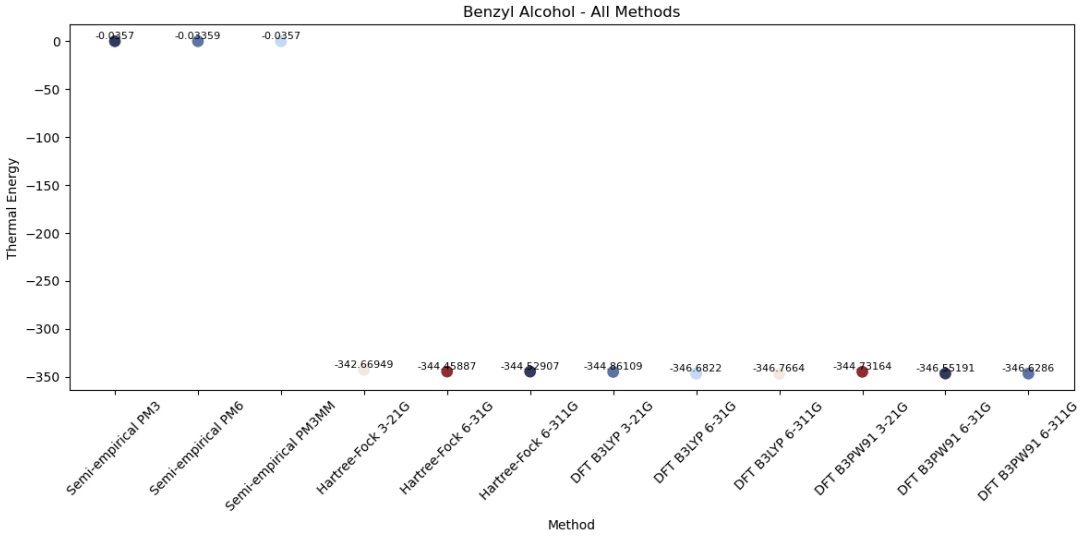
Figure 1: 所有的能量对比图



(a) 苯甲酸

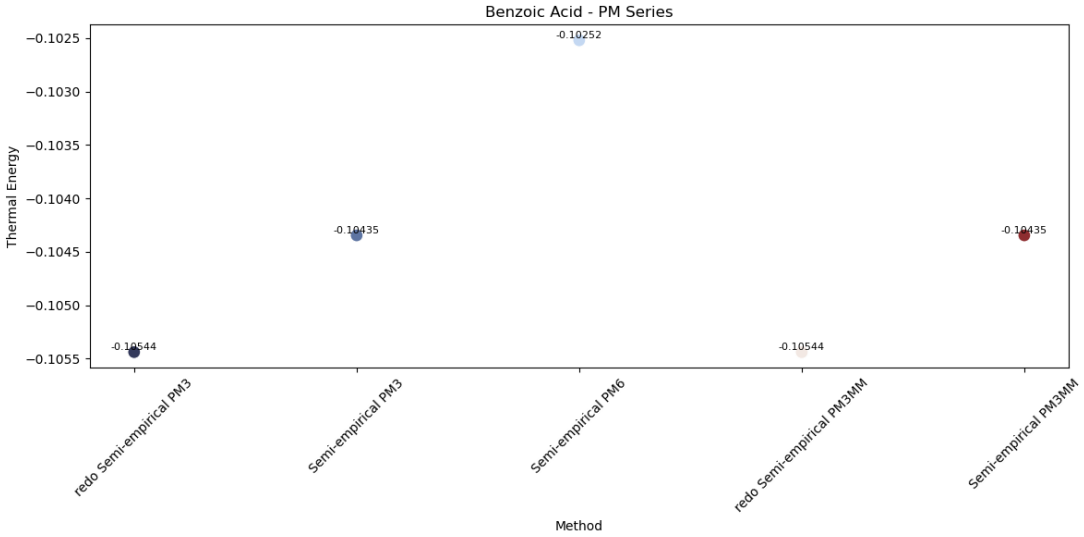


(b) 苯甲醛

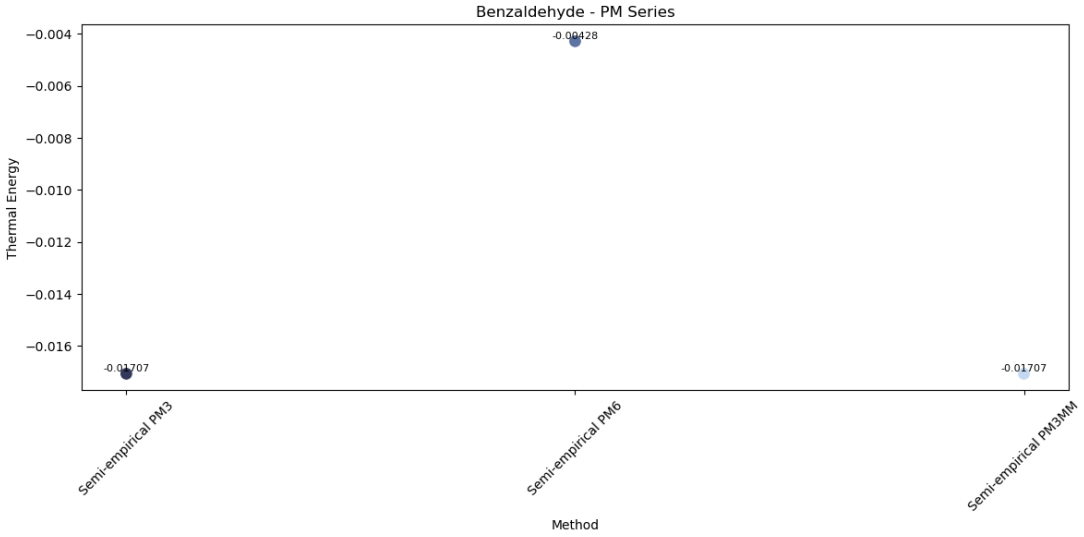


(c) 苯甲醇

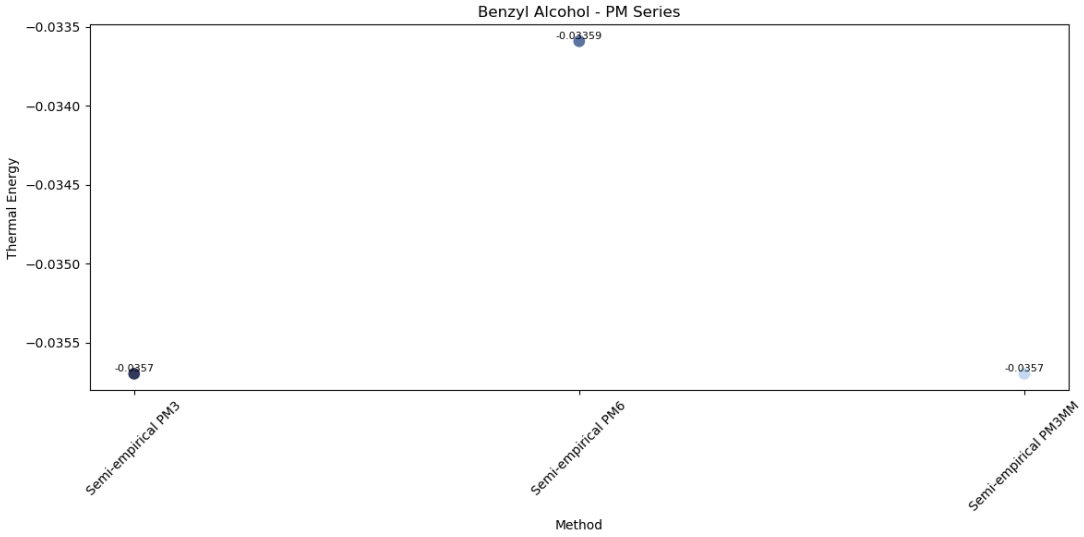
Figure 2: Semi-empirical 方法的能量对比图



(a) 苯甲酸

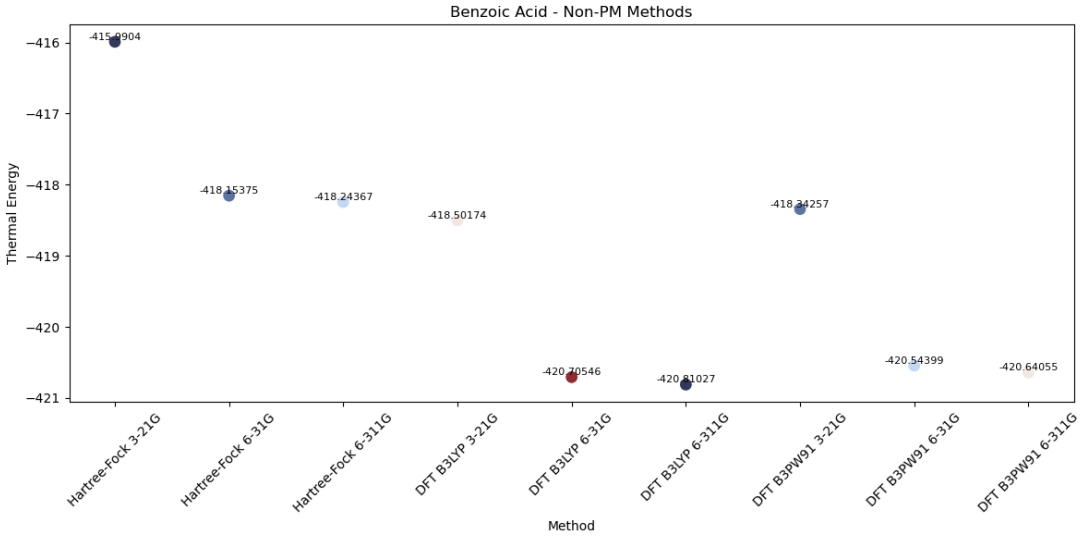


(b) 苯甲醛

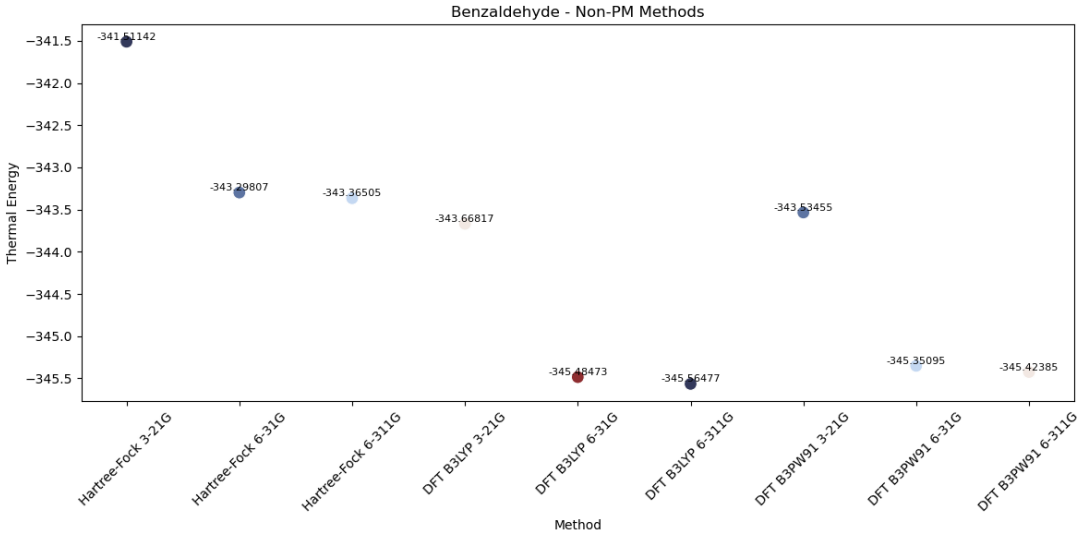


(c) 苯甲醇

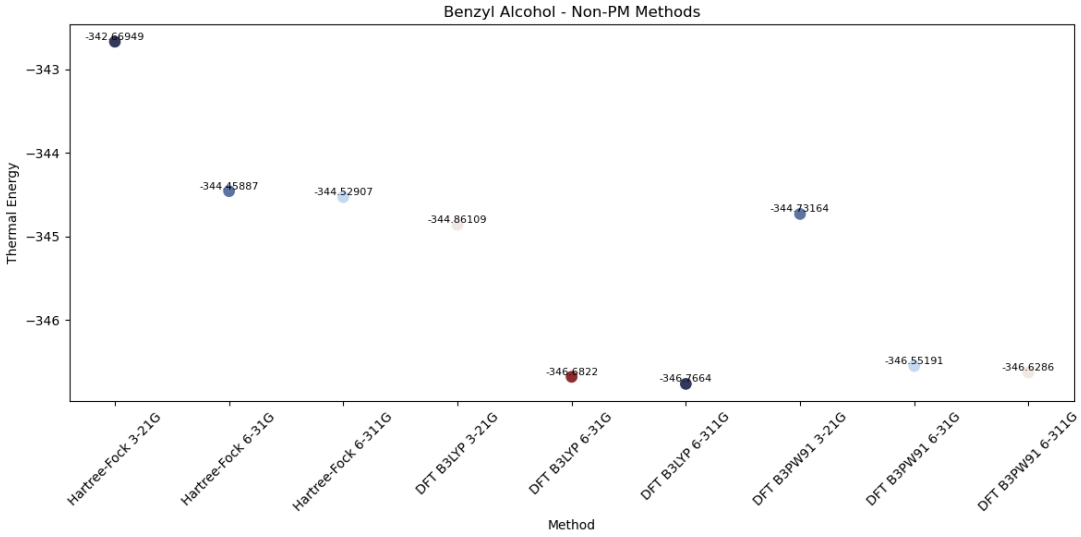
Figure 3: Semi-empirical 方法以外的能量对比图



(a) 苯甲酸

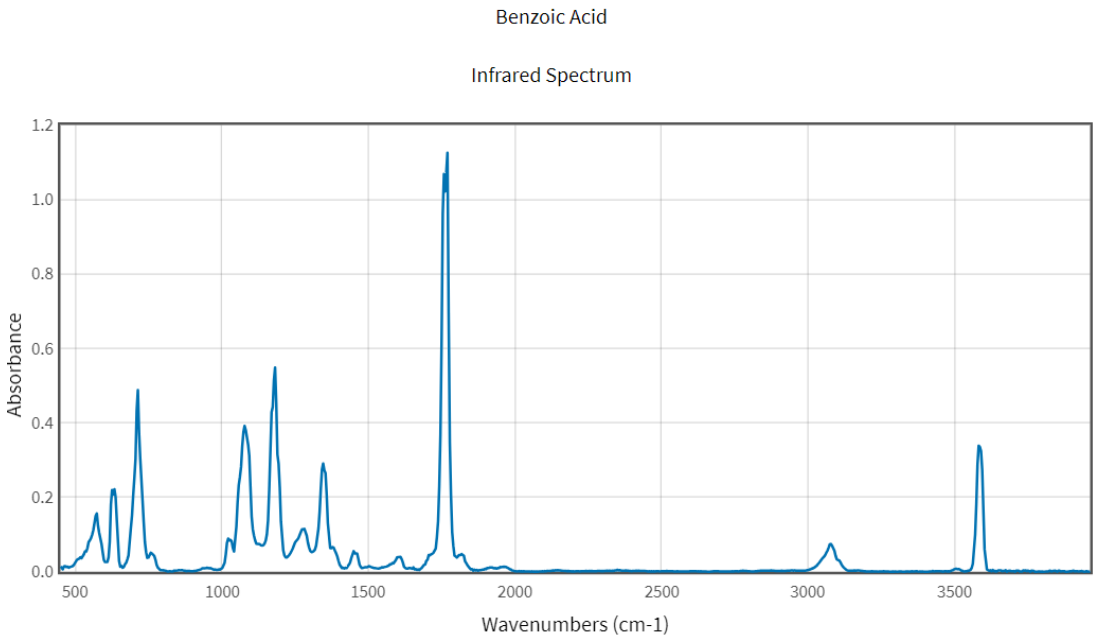


(b) 苯甲醛

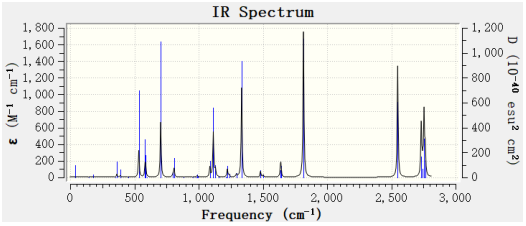


(c) 苯甲醇

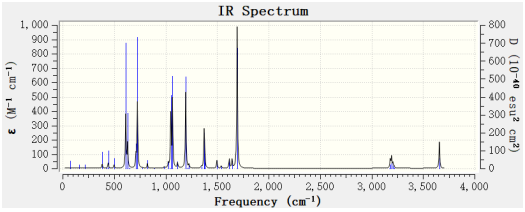
Figure 4: 苯甲酸 IR 图



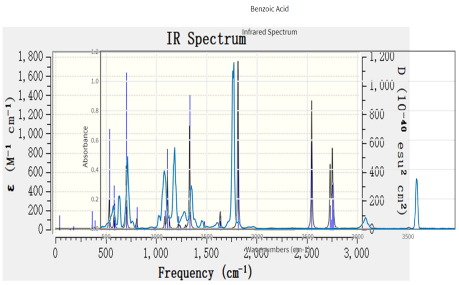
(a) 实验数据



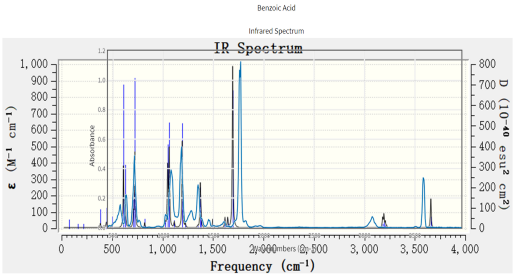
(b) 采用 Semi-empirical PM6



(c) 采用 DFT B3LYP 6-311G

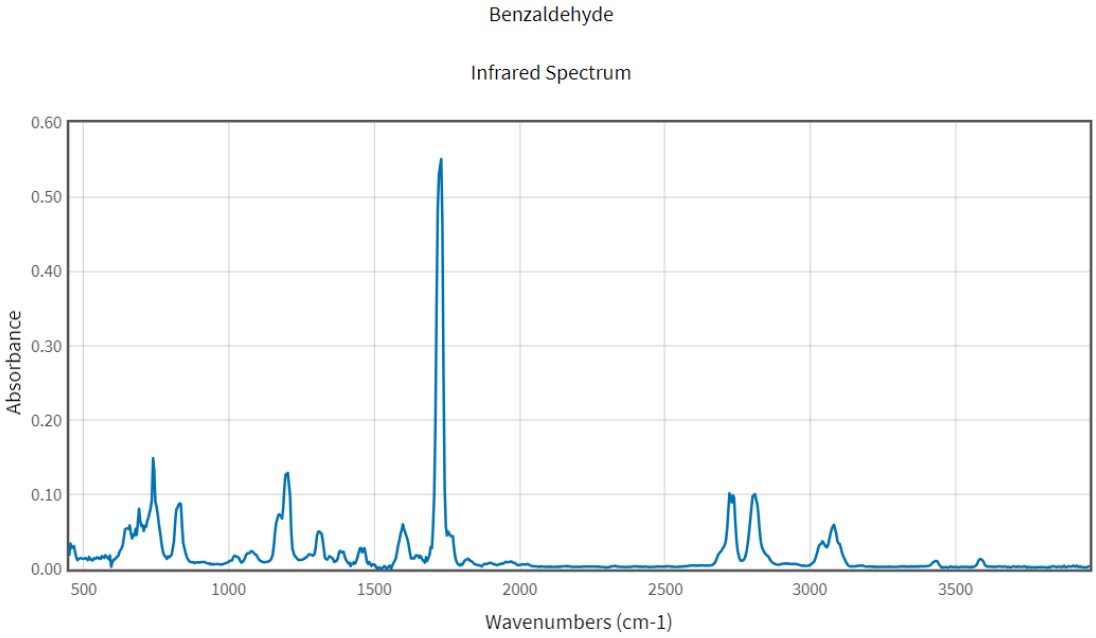


(d) Semi-empirical PM6 方法与实验数据对比

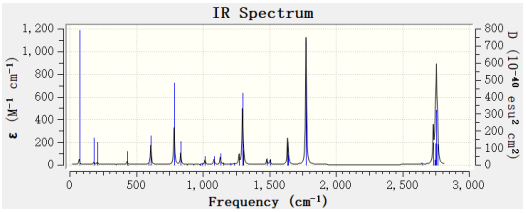


(e) DFT B3LYP 6-311G 方法与实验数据对比

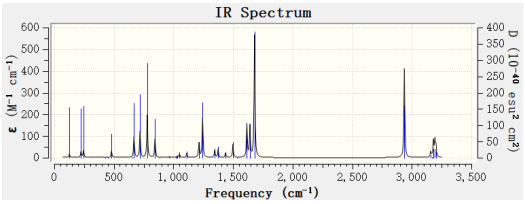
Figure 5: 苯甲醛 IR 图



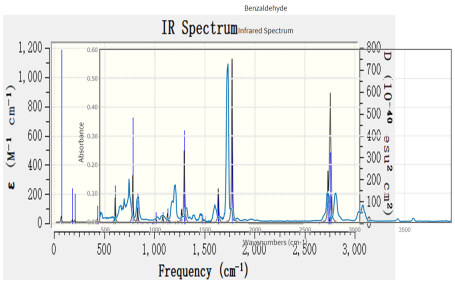
(a) 实验数据



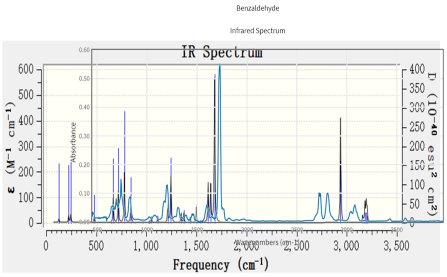
(b) 采用 Semi-empirical PM6



(c) 采用 DFT B3LYP 6-311G

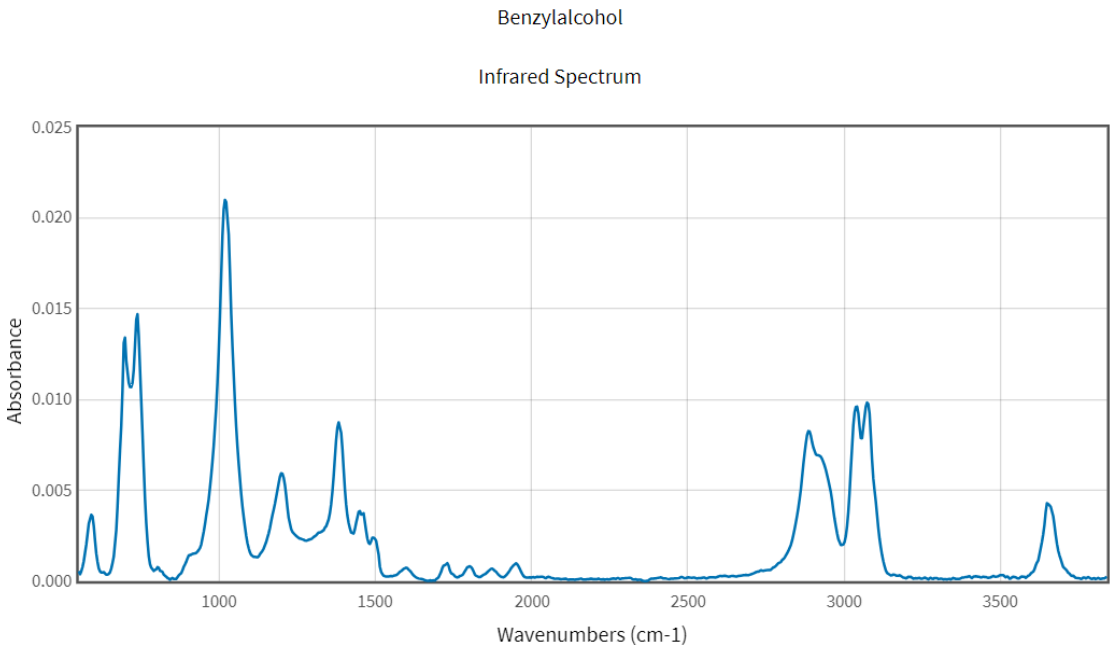


(d) Semi-empirical PM6 与实验数据对比

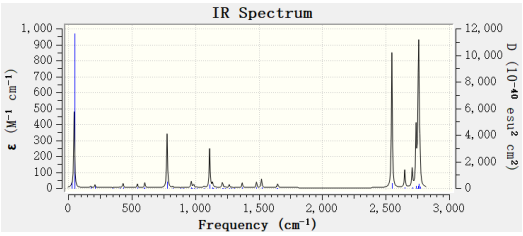


(e) DFT B3LYP 6-311G 与实验数据对比

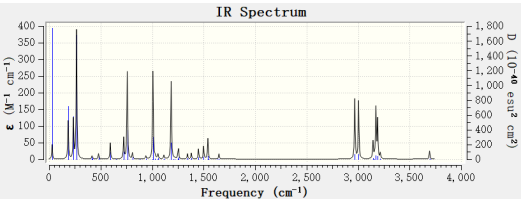
Figure 6: 苯甲醇 IR 图



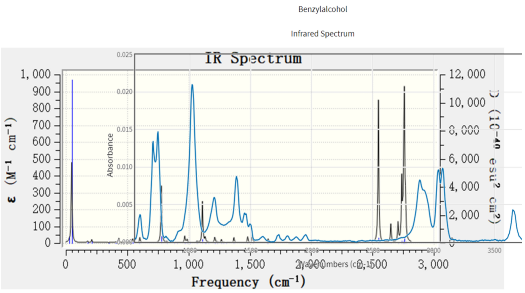
(a) 实验数据



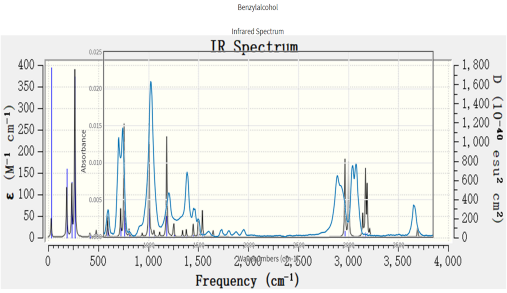
(b) 采用 Semi-empirical PM6



(c) 采用 DFT B3LYP 6-311G



(d) Semi-empirical PM6 与实验数据对比



(e) DFT B3LYP 6-311G 与实验数据对比

4 实验分析与结论

通过以上图表，我们不难发现，由于初猜的对称性，PM3 与 PM3MM 会导致分子结构不能收敛到最低点，这个问题在切换到 PM6 或者改变初猜的原子坐标（在 redo 中完成了修改）后可以得到解决。

由于 Semi-empirical 的方法不能得到分子绝对能量，故除了所有数据的对比图 (Figure 1) 外，我们分别给出了仅包含 Semi-empirical 和不包含 Semi-empirical 的

能量对比图 (Figure 2、Figure 3) 很明显在 redo 后分子相对能量得到了下降，这和我们的预期是相符合的。由于 Semi-empirical 法得到的能量差不大，我们可以选择较稳定的 PM6 开展后续红外光谱分析。

观察图表，我们发现，在不同方法中，Hartree-Fock 总是得到较高的能量，而能量总是从 3-21G、6-31G、6-311G 递减，而 B3LYP 总是低于 B3PW91。DFT B3LYP 6-311G 总是得到最低的能量。因此我们选择 DFT B3LYP 6-311G 作为后续红外光谱分析的第二个方法。

在对红外光谱的对比分析中，我们也可以看出，DFT B3LYP 6-311G 得到的红外光谱相比 Semi-empirical PM6，会更接近实验的红外光谱。而且显然前者在精细结构上要优于后者，大部分的峰的位置也更与实验接近，尽管两者在总体上与实验数据都有着不小的差距，还不足以作为准确的判据。

本次实验的绝大部分重复操作均使用了各类脚本完成，可以用作后续大批量计算工作的参考。

5 附录

```bash
# trans.sh
for chk_file in *.chk; do
    base_name="${chk_file%.chk}"
    fchk_file="${base_name}.fchk"
    formchk "$chk_file" "$fchk_file"
done
```

```python
# extract.py,用于从log文件中提取能量信息
import os
import re

def extract_thermal_energy_from_log(log_file):
    with open(log_file, 'r') as file:
        content = file.read()
        match = re.search(r'\\HF=([-+]?[0-9]*\.?[0-9]+)', content)
        if match:
            return match.group(1)
    return None

def main():
    data = []  # 列表用于存储提取的数据

    for file in os.listdir('.'):
        if file.endswith('.log'):
            energy = extract_thermal_energy_from_log(file)
            if energy:
                # 将.log从文件名中移除
                filename_without_extension = os.path.splitext(file)[0]
                data.append([filename_without_extension, energy])

    # 写入CSV文件
    with open('output.csv', 'w') as outfile:
        outfile.write("Filename,Thermal Energy\n")  # CSV头部
        for line in data:
            outfile.write(f"{line[0]},{line[1]}\n")
    print("Current working directory:", os.getcwd())
```

```
if __name__ == "__main__":
    main()
```

```
# draw.py,用于将能量信息可视化
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the CSV data
data = pd.read_csv('output.csv')

# Extract the compound identifier
data['Compound'] = data['Filename'].str.extract(r'(\d+-\d+)')

# Define the color palette
colors = {
    'color1': (51/255, 57/255, 91/255),
    'color2': (93/255, 116/255, 162/255),
    'color3': (196/255, 216/255, 242/255),
    'color4': (242/255, 232/255, 227/255),
    'color5': (142/255, 45/255, 48/255)
}
color_list = list(colors.values())

# Mapping of compound methods to their names
method_mapping = {
    '1-1-1': 'Semi-empirical PM3',
    '1-1-1-1': 'redo Semi-empirical PM3',
    '1-1-2': 'Semi-empirical PM6',
    '1-1-3': 'Semi-empirical PM3MM',
    '1-1-3-1': 'redo Semi-empirical PM3MM',
    '1-2-1': 'Hartree-Fock 3-21G',
    '1-2-2': 'Hartree-Fock 6-31G',
    '1-2-3': 'Hartree-Fock 6-311G',
    '1-3-1': 'DFT B3LYP 3-21G',
    '1-3-2': 'DFT B3LYP 6-31G',
    '1-3-3': 'DFT B3LYP 6-311G',
    '1-4-1': 'DFT B3PW91 3-21G',
    '1-4-2': 'DFT B3PW91 6-31G',
    '1-4-3': 'DFT B3PW91 6-311G',
    '2-1-1': 'Semi-empirical PM3',
    '2-1-2': 'Semi-empirical PM6',
    '2-1-3': 'Semi-empirical PM3MM',
    '2-2-1': 'Hartree-Fock 3-21G',
    '2-2-2': 'Hartree-Fock 6-31G',
    '2-2-3': 'Hartree-Fock 6-311G',
    '2-3-1': 'DFT B3LYP 3-21G',
    '2-3-2': 'DFT B3LYP 6-31G',
    '2-3-3': 'DFT B3LYP 6-311G',
    '2-4-1': 'DFT B3PW91 3-21G',
    '2-4-2': 'DFT B3PW91 6-31G',
    '2-4-3': 'DFT B3PW91 6-311G',
    '3-1-1': 'Semi-empirical PM3',
    '3-1-2': 'Semi-empirical PM6',
    '3-1-3': 'Semi-empirical PM3MM',
    '3-2-1': 'Hartree-Fock 3-21G',
    '3-2-2': 'Hartree-Fock 6-31G',
    '3-2-3': 'Hartree-Fock 6-311G',
    '3-3-1': 'DFT B3LYP 3-21G',
    '3-3-2': 'DFT B3LYP 6-31G',
    '3-3-3': 'DFT B3LYP 6-311G',
    '3-4-1': 'DFT B3PW91 3-21G',
    '3-4-2': 'DFT B3PW91 6-31G',
```

```
    '3-4-3': 'DFT B3PW91 6-311G'
}
# Apply the mapping to the data
data['Method Name'] = data['Filename'].map(method_mapping)

# Map the Chinese compound names to English for visualization
compound_name_mapping = {
    "苯甲酸": "Benzoic Acid",
    "苯甲醛": "Benzaldehyde",
    "苯甲醇": "Benzyl Alcohol"
}

compound_names = ["苯甲酸", "苯甲醛", "苯甲醇"]
compound_ids = ['1-1', '2-1', '3-1']

# Plot scatter plots for each compound without PM series methods
for compound_name, compound_id in zip(compound_names, compound_ids):
    subset = data[data['Compound'].str.startswith(compound_id.split('-')[0])
        & ~data['Filename'].str.endswith(('1-1', '1-1-1', '1-2', '1-3', '
        1-1-3-1'))]
    plt.figure(figsize=(12, 6))
    sns.scatterplot(x='Method Name', y='Thermal Energy', data=subset, hue='
        Method Name', palette=color_list, s=100, legend=None)
    plt.title(f"{compound_name_mapping[compound_name]} - Non-PM Methods")
    plt.ylabel("Thermal Energy")
    plt.xlabel("Method")
    plt.xticks(rotation=45)
        # Adding data labels
    for _, row in subset.iterrows():
        plt.text(row['Method Name'], row['Thermal Energy'], round(row['
            Thermal Energy'], 5), fontsize=8, ha='center', va='bottom')
    plt.tight_layout()
    plt.show()

# Plot scatter plots for each compound in PM series
for compound_name, compound_id in zip(compound_names, compound_ids):
    subset = data[data['Filename'].str.startswith(compound_id) & data['
        Filename'].str.endswith(('1', '1-1', '2', '3', '3-1'))]
    plt.figure(figsize=(12, 6))
    sns.scatterplot(x='Method Name', y='Thermal Energy', data=subset, hue='
        Method Name', palette=color_list, s=100, legend=None)
    plt.title(f"{compound_name_mapping[compound_name]} - PM Series")
    plt.ylabel("Thermal Energy")
    plt.xlabel("Method")
    plt.xticks(rotation=45)
        # Adding data labels
    for _, row in subset.iterrows():
        plt.text(row['Method Name'], row['Thermal Energy'], round(row['
            Thermal Energy'], 5), fontsize=8, ha='center', va='bottom')
    plt.tight_layout()
    plt.show()

# Plot scatter plots for each compound with all methods
for compound_name, compound_id in zip(compound_names, compound_ids):
    subset = data[data['Compound'].str.startswith(compound_id.split('-')[0])]
    plt.figure(figsize=(12, 6))
    sns.scatterplot(x='Method Name', y='Thermal Energy', data=subset, hue='
        Method Name', palette=color_list, s=100, legend=None)
    plt.title(f"{compound_name_mapping[compound_name]} - All Methods")
    plt.ylabel("Thermal Energy")
    plt.xlabel("Method")
    plt.xticks(rotation=45)    # Adding data labels
    for _, row in subset.iterrows():
```

```
        plt.text(row['Method Name'], row['Thermal Energy'], round(row['
            Thermal Energy'], 5), fontsize=8, ha='center', va='bottom')

    plt.tight_layout()
    plt.show()
```