数据解释（前100行示例）

文件内容为基因突变矩阵，格式如下：

- **行**：基因名称（如 `UBE2Q2`，`CHMP1B`），可能包含假基因或非编码RNA（如 `PSMA2P1`）。
- **列**：样本ID（如 `TCGA-3M-AB46-01`），每个样本对应一个癌症患者的肿瘤样本。
- **值**：大部分为 `0` 或 `1`，表示该基因在样本中是否发生突变（`1` 表示突变，`0` 表示未突变）。

示例数据片段：

```
sample    TCGA-3M-AB46-01  TCGA-3M-AB47-01  ...  TCGA-ZQ-A9CR-01
UBE2Q2    0                0                ...  0
CHMP1B    0                0                ...  0
...
REM1      0                0                ...  0
```

数据预处理步骤

1. 数据清洗：去噪与缺失值填充

目标：确保数据质量，处理异常值和缺失值。

步骤：

1. **检查数据格式**：
   - 确认所有值为 `0` 或 `1`，无其他异常值（如字符串、负数）。
   - 若存在非 `0/1` 的值（如 `NA` 或空值），需标记为缺失值。
2. **处理缺失值**：
   - **策略**：基因突变数据通常以 `0` 表示未突变，缺失值可填充为 `0`（假设未检测到突变）。
   - **代码示例（Python）**：

     ```python
     import pandas as pd
     df = pd.read_csv("STAD_mc3_gene_level.txt", sep="\t", index_col=0)
     df.fillna(0, inplace=True)  # 填充缺失值为0
     ```

3. **去噪**：
   - 若某基因在所有样本中均为 `0`（无突变），可删除该基因（对分析无贡献）。
   - **代码示例**：

     ```python
     df = df.loc[df.sum(axis=1) > 0]  # 删除全0行
     ```

---

2. 标准化

目标：调整数据分布，便于后续分析（如机器学习）。

注意事项：

- **二进制数据（0/1）**：通常不需要标准化。若需整合其他连续型组学数据（如表达量），需单独标准化后再整合。

若需标准化（非必要步骤）：

- **归一化（Min-Max Scaling）**：

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df_normalized = pd.DataFrame(scaler.fit_transform(df),
columns=df.columns, index=df.index)
```

---

3. 数据整合：多模态数据对齐

目标：将不同组学数据（如突变、表达量、甲基化）按样本ID对齐。

步骤：

1. **读取其他组学数据**：
   - 假设另有表达量数据文件 `STAD_expression.txt`，格式与突变数据类似。

```
df_expression = pd.read_csv("STAD_expression.txt", sep="\t", index_col=0)
```

2. **对齐样本ID**：
   - 提取共有的样本ID，并排序确保一致性。

```
common_samples = df.columns.intersection(df_expression.columns)
df_mutation_aligned = df[common_samples]
df_expression_aligned = df_expression[common_samples]
```

3. **生成多模态矩阵**：
   - 按行（基因）合并突变和表达量数据，或按列（样本）拼接不同组学特征。

```
# 按行合并（假设行名一致）
df_multimodal = pd.concat([df_mutation_aligned, df_expression_aligned],
axis=0)

# 按列合并（添加组学类型前缀）
df_mutation_aligned.columns = ["Mutation_" + col for col in
df_mutation_aligned.columns]
df_expression_aligned.columns = ["Expression_" + col for col in
df_expression_aligned.columns]
df_multimodal = pd.concat([df_mutation_aligned.T,
df_expression_aligned.T], axis=1)
```

最终输出

- **清洗后数据**：无缺失值、低质量基因已过滤。
- **标准化数据**（可选）：归一化后的连续型数据。
- **多模态矩阵**：对齐样本ID的整合数据，可直接用于下游分析（如生存分析、聚类）。

通过以上步骤，您可以将基因突变数据与其他组学数据整合，构建可用于多模态机器学习或统计分析的输入矩阵。