Mixed_Visualization

November 13, 2023

[ ]:
```
import pandas as pd
import numpy as np

import seaborn as sns

import thermogram_utilities

import warnings
warnings.filterwarnings("ignore")

mixed = pd.read_excel("mixed_thermograms.xlsx")
```

[ ]:
```
mixed.groupby(["DiseaseGroup"])["code"].value_counts()
```

[ ]:
```
DiseaseGroup             code
ALS                      ALS      12
Anti-CCP                 CCP      10
CIN                      CIN      67
Centromere               CEN      10
Chromatin / Ribo-P / Sm  CHR      10
Diabetes                 DBT      33
Early RA                 ERA      10
Endometrial              END       8
Gammopathy               GAM       4
Heart                    HRT      18
Jo-1 (polymyositis)      JOF      25
Lung                     LUN      50
Lupus                    SLE     299
                         LUF      50
                         LUP       2
Lyme                     LYM      10
Melanoma                 MEL       7
Multiple sclerosis       MSC      20
Myocardial infarction    MCI      20
Normal                   NML     122
Ovarian                  OVA      12
Pelvic mass              PEL      16
Phenytoin                PHN       6
```

```
Rheumatoid arthritis     RAF      18
                         RAA      11
Ro52                     ROF      10
Scl-70                   S70       9
Scleroderma              SCL      50
Uterine                  UTR       2
Name: count, dtype: int64
```

[ ]:
```
mixed["code"].value_counts()
```

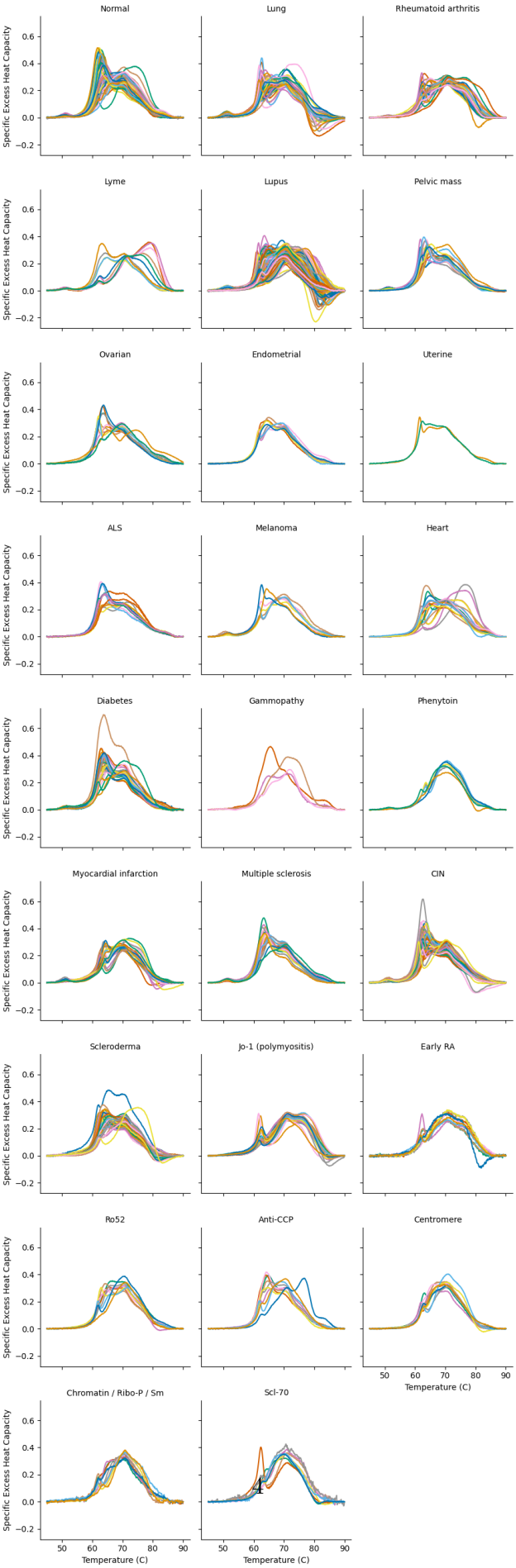
[ ]:
```
code
SLE    299
NML    122
CIN     67
LUN     50
SCL     50
LUF     50
DBT     33
JOF     25
MCI     20
MSC     20
RAF     18
HRT     18
PEL     16
OVA     12
ALS     12
RAA     11
CEN     10
CCP     10
ROF     10
ERA     10
CHR     10
LYM     10
S70      9
END      8
MEL      7
PHN      6
GAM      4
UTR      2
LUP      2
Name: count, dtype: int64
```

[ ]:
```
mixed_long = pd.melt(mixed, id_vars=["DiseaseGroup", "sampleID", "code"],␣
 ↪var_name="temp", value_name="dsp" )

mixed_long["temp"] = mixed_long["temp"].str.replace("T", "")
mixed_long["temp"] = mixed_long["temp"].astype(float)
```

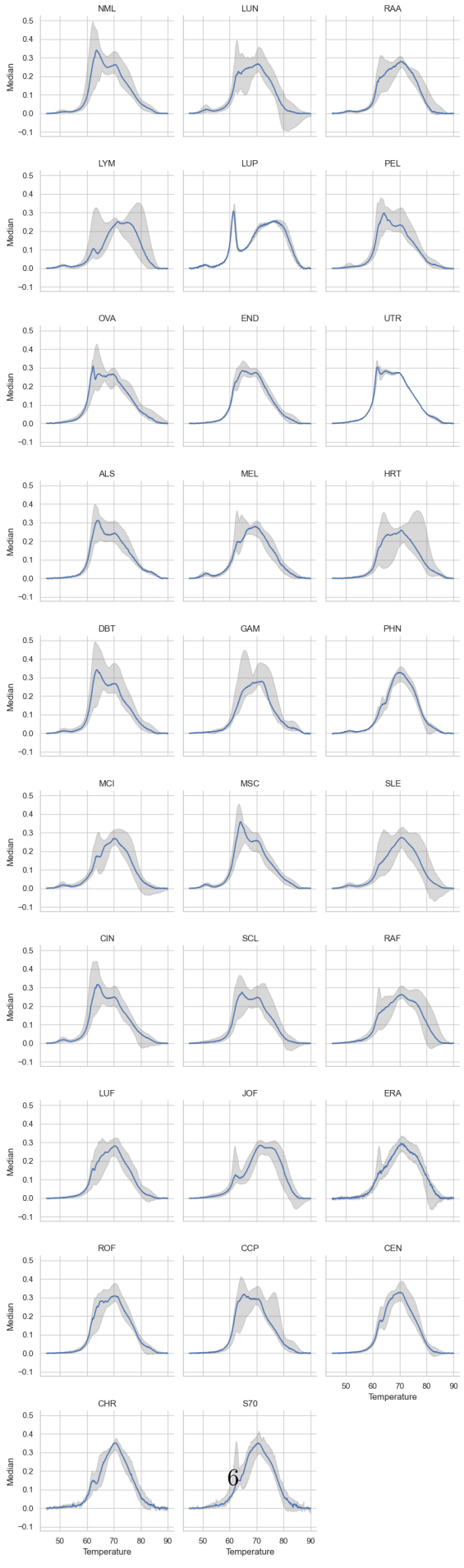
[ ]:
```
median_df = thermogram_utilities.median_curve(mixed_long, "code", "temp", "dsp")
```

[ ]:
```
g = sns.FacetGrid(mixed_long, col="DiseaseGroup", col_wrap= 3, hue="sampleID",␣
↪palette = 'colorblind')
g.map_dataframe(sns.lineplot, x="temp", y="dsp")
g.set_axis_labels("Temperature (C)", "Specific Excess Heat Capacity")
g.set_titles(col_template="{col_name}")
```

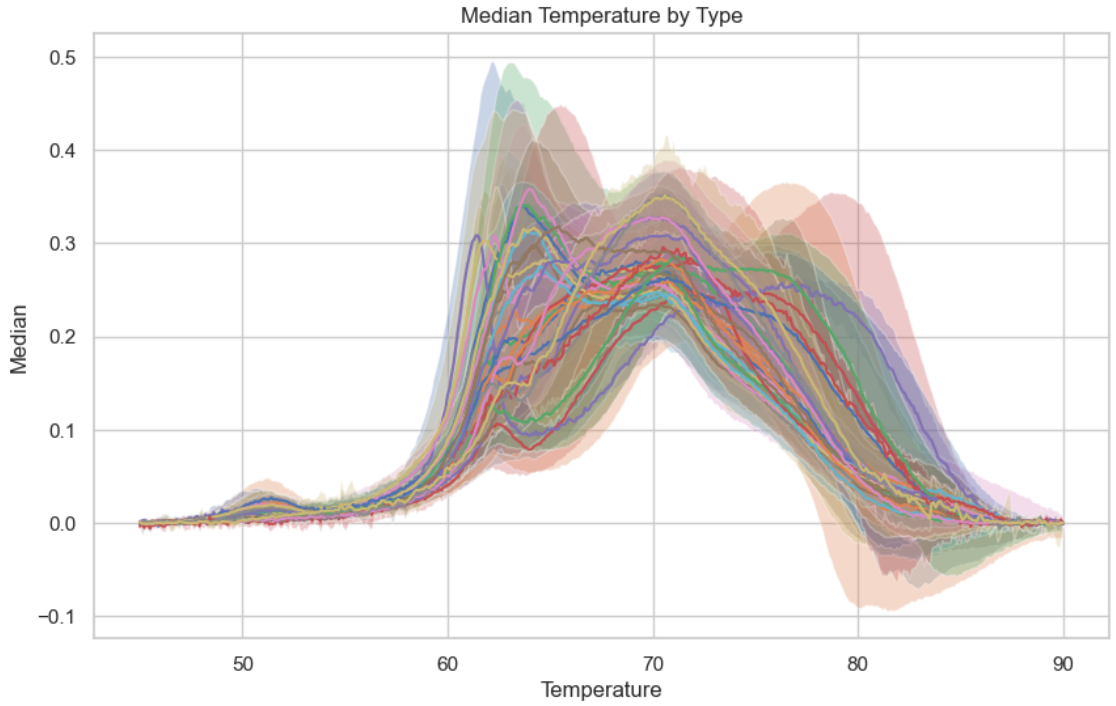
[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1dfbcc73eb0>
```


Normal
Lung
Rheumatoid arthritis
Lyme
Lupus
Pelvic mass
Ovarian
Endometrial
Uterine
ALS
Melanoma
Heart
Diabetes
Gammopathy
Phenytoin
Myocardial infarction
Multiple sclerosis
CIN
Scleroderma
Jo-1 (polymyositis)
Early RA
Ro52
Anti-CCP
Centromere
Chromatin / Ribo-P / Sm
Scl-70
Specific Excess Heat Capacity
Temperature (C)

[ ]:
```
g = sns.FacetGrid(median_df, col="type", col_wrap= 3, palette = 'colorblind')


g.set_axis_labels("Temperature", "Median")
g.set_titles(col_template="{col_name}")

# Define a custom function to add the ribbon to each panel
def add_ribbon(**kwargs):
    ax = plt.gca()
    panel_data = kwargs.pop("data")  # Get the data specific to the current␣
 ↪panel

    sns.lineplot(data=panel_data, x='temperature', y='median', ax=ax,␣
 ↪palette='colorblind')
    ax.fill_between(panel_data['temperature'], panel_data['lower_q'],␣
 ↪panel_data['upper_q'], alpha=0.3, color='gray')

# Use FacetGrid.map_dataframe to apply the custom function to each panel
g.map_dataframe(add_ribbon)

# Show the plot
plt.show()
```


NML
LUN
RAA
LYM
LUP
PEL
OVA
END
UTR
ALS
MEL
HRT
DBT
GAM
PHN
MCI
MSC
SLE
CIN
SCL
RAF
LUF
JOF
ERA
ROF
CCP
CEN
CHR
S70
Median
Temperature
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
50
60
70
80
90

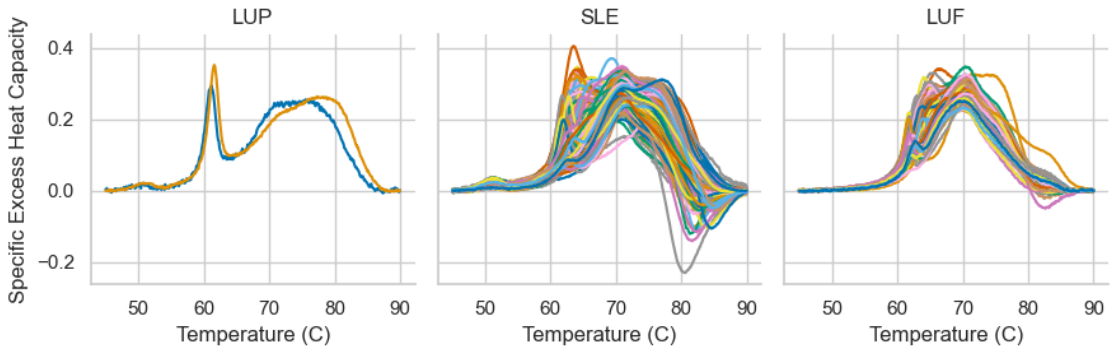
[ ]:
```
# Assuming 'median_df' is your DataFrame
# Ensure that the 'temperature' column is numeric and that 'lower_q' and␣
 ↪'upper_q' columns are present

# Set the style
sns.set(style="whitegrid")

# Create a figure and axis
fig, ax = plt.subplots(figsize=(10, 6))

# Define the groups based on 'type'
groups = median_df['type'].unique()

# Plot each line and add ribbon
for group in groups:
    group_data = median_df[median_df['type'] == group].
 ↪sort_values(by='temperature')
    sns.lineplot(data=group_data, x='temperature', y='median', label=group,␣
 ↪legend=False)
    ax.fill_between(group_data['temperature'], group_data['lower_q'],␣
 ↪group_data['upper_q'], alpha=0.3)

# Set labels and title
ax.set(xlabel='Temperature', ylabel='Median', title='Median Temperature by␣
 ↪Type')

# Show the plot without legend
plt.show()
```



[ ]:
```
mixed_multiple = mixed_long[(mixed_long["DiseaseGroup"] == "Lupus") |␣
↪(mixed_long["DiseaseGroup"] == "Rheumatoid arthritis")]
```

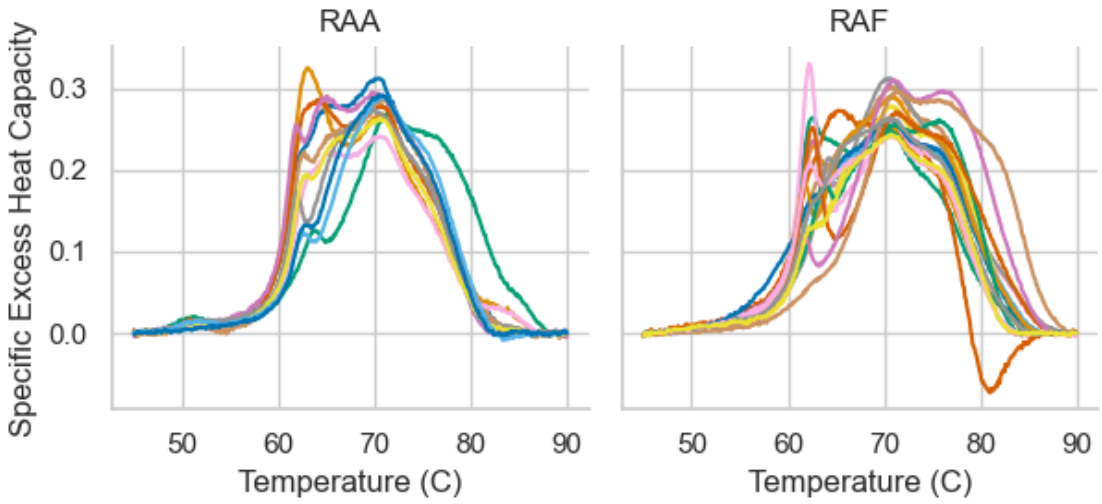
[ ]:
```
graph_df = mixed_multiple[mixed_multiple["DiseaseGroup"] == "Lupus"]

g = sns.FacetGrid(graph_df, col="code", col_wrap= 3, hue="sampleID", palette =␣
↪'colorblind')
g.map_dataframe(sns.lineplot, x="temp", y="dsp")
g.set_axis_labels("Temperature (C)", "Specific Excess Heat Capacity")
g.set_titles(col_template="{col_name}")
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1dfc21e90c0>
```



[ ]:
```
graph_df = mixed_multiple[mixed_multiple["DiseaseGroup"] == "Rheumatoid␣
↪arthritis"]

g = sns.FacetGrid(graph_df, col="code", col_wrap= 3, hue="sampleID", palette =␣
↪'colorblind')
g.map_dataframe(sns.lineplot, x="temp", y="dsp")
g.set_axis_labels("Temperature (C)", "Specific Excess Heat Capacity")
g.set_titles(col_template="{col_name}")
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x1dfca0337f0>
```



[ ]:
```
C:\Users\avery\.jupyter nbconvert -- to pdf Mixed_Visualization.ipynb
```