In [1]:

```
import os
import json
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from extract_features import moving_window

from sklearn.metrics import davies_bouldin_score
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn import svm

from sklearn.model_selection import cross_val_score

from matplotlib import pyplot as plt

RAND_SEED = 42
```

In [84]:

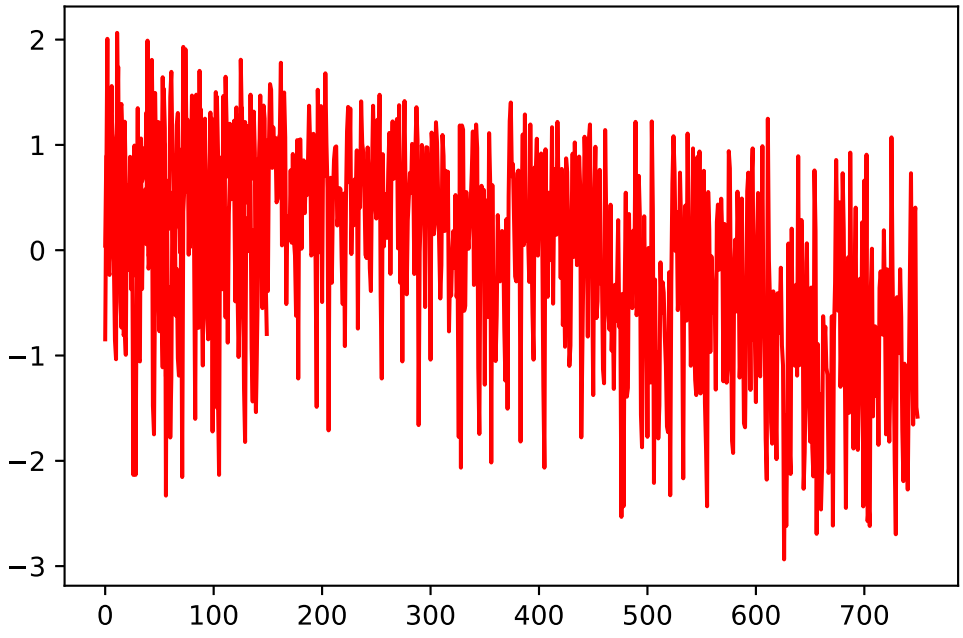
```
# compare windowed to full
# notice the downward slope

features_26 = np.load("data/junipersun/FA_data/features_train_2.6_0.1.npy")
features_26.shape
plt.plot(features_26[0, :, 0], 'r-')

features_30 = np.load("data/junipersun/FA_data/features_train_3.0_0.1.npy")
features_30.shape
plt.plot(features_30[0, :, 0], 'r-')
```

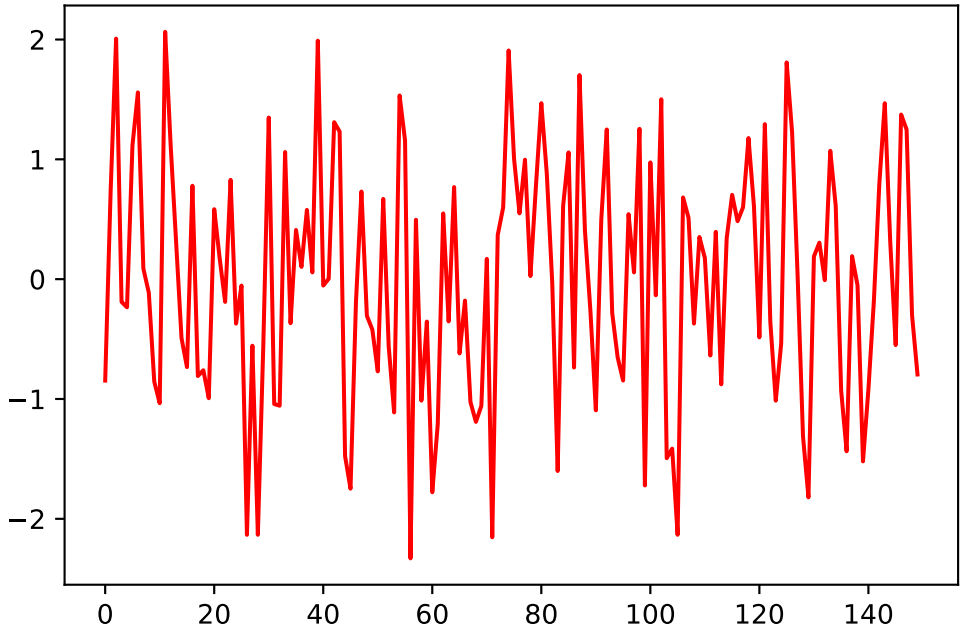
Out[84]:

```
[<matplotlib.lines.Line2D at 0x7f5837d61f10>]
```



In [82]:

Out[82]: [<matplotlib.lines.Line2D at 0x7f58368fad30>]



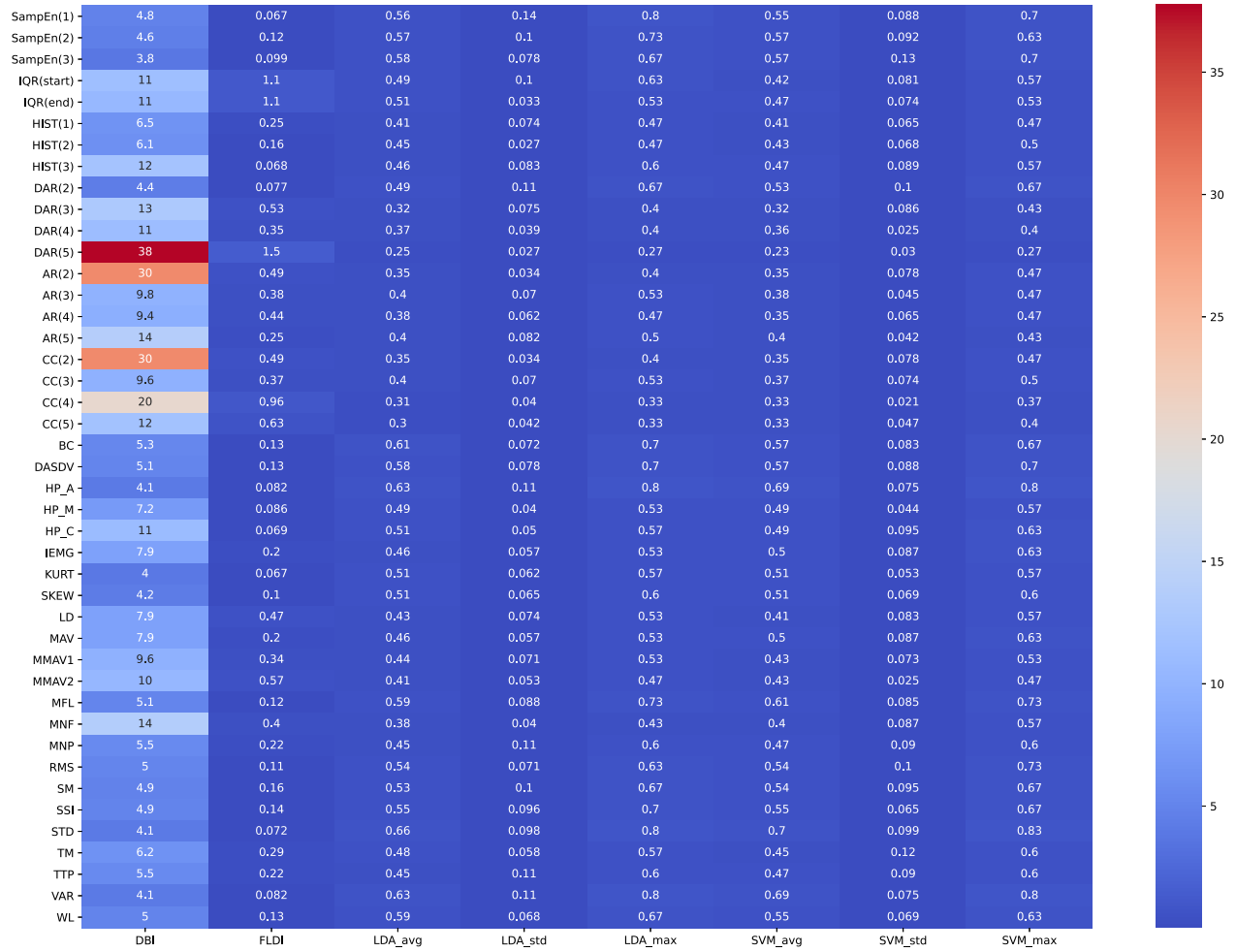
In [69]:

```
wli
wii
path_to_data = "data/junipersun/"

FA_results = pd.read_csv(os.path.join(path_to_data, "FA_results/feature_analysis

plt.figure()
plt.subplots(figsize=(20,15))
sns.heatmap(FA_results, annot=True, cmap='coolwarm')
plt.show()
```

<Figure size 432x288 with 0 Axes>



In [48]:

```
wli = 3.0
wii = 0.1
path_to_data = "data/junipersun/"

# load meta data
with open(os.path.join(path_to_data, "info.txt")) as json_file:
    meta_info = json.load(json_file)

# load gestures labels, subject labels
y_g = np.load(os.path.join(path_to_data, "split/y_g_train.npy"))

# load features
features = np.load(os.path.join(
    path_to_data, "FA_data/features_train_{}_{}.npy".format(round(wli, 2), round

# load feature labels
with open(os.path.join(path_to_data, "FA_data/feature_labels_train_{}_{}.txt".fo
    feature_labels = [line.rstrip('\n') for line in f]
print(feature_labels)

features = np.nan_to_num(features)

# format gesture labels
n_windows = features.shape[1] / y_g.shape[0]
y_g_w = np.repeat(y_g, n_windows)
```

```
['SampEn(1)', 'SampEn(2)', 'SampEn(3)', 'IQR(start)', 'IQR(end)', 'HIST(1)', 'HI
ST(2)', 'HIST(3)', 'DAR(2)', 'DAR(3)', 'DAR(4)', 'DAR(5)', 'AR(2)', 'AR(3)', 'AR
```

```
(4)', 'AR(5)', 'CC(2)', 'CC(3)', 'CC(4)', 'CC(5)', 'BC', 'DASDV', 'HP_A', 'HP_
M', 'HP_C', 'IEMG', 'KURT', 'SKEW', 'LD', 'MAV', 'MMAV1', 'MMAV2', 'MFL', 'MNF',
'MNP', 'RMS', 'SM', 'SSI', 'STD', 'TM', 'TTP', 'VAR', 'WL']
```

In [49]:

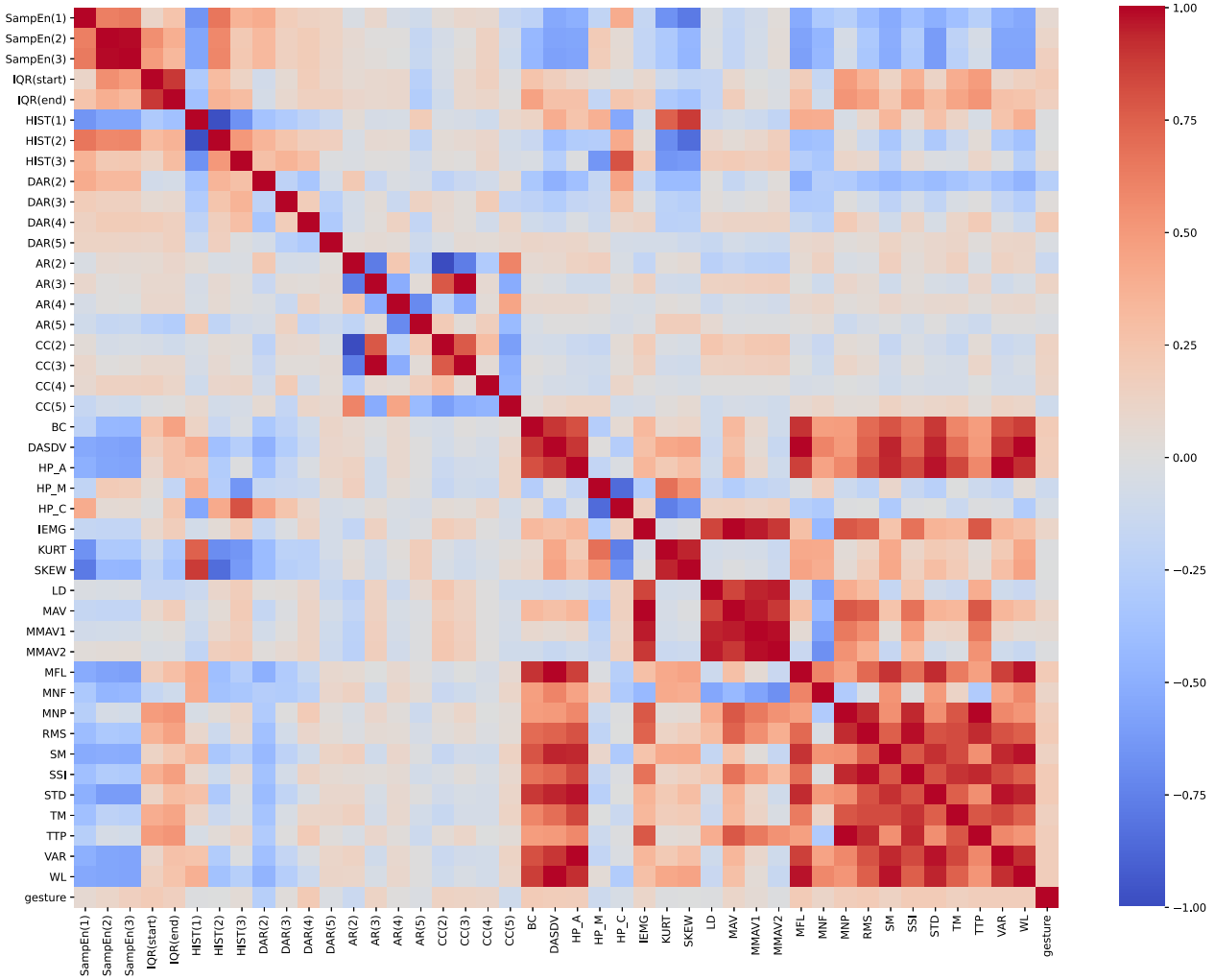
```
c = 0 # channel
features_c = np.squeeze(features[:, :, c]).T
print("features_c.shape", features_c.shape)
df_features_c = pd.DataFrame(features_c, columns=feature_labels)
df_features_c.loc[:, 'gesture'] = y_g_w
df_features_c = df_features_c.sort_values(by='gesture')
```

```
features_c.shape (150, 43)
```

In [70]:

```
plt.figure()
plt.subplots(figsize=(20,15))
sns.heatmap(df_features_c.corr(), annot=False, cmap='coolwarm')
plt.show()
```

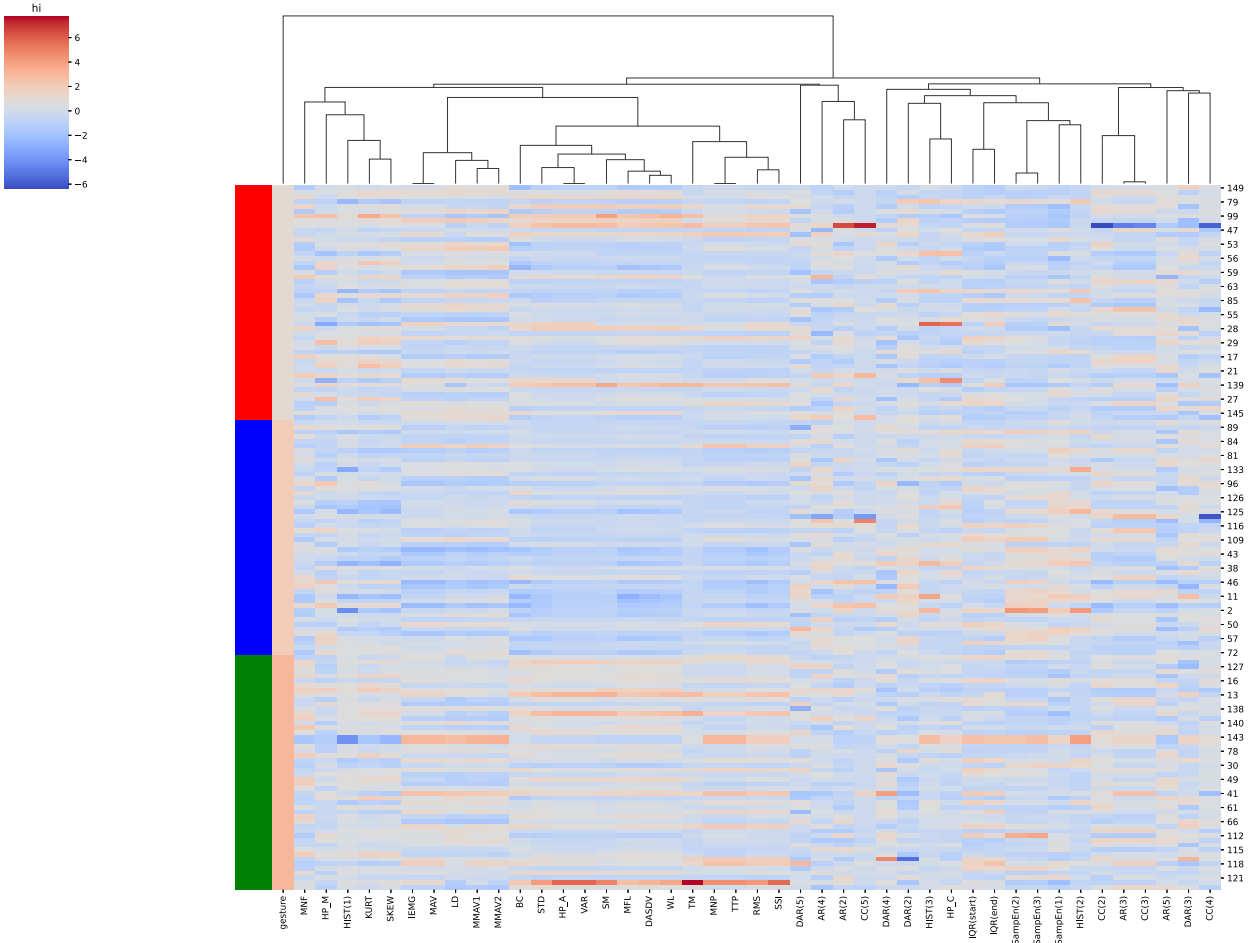
```
<Figure size 432x288 with 0 Axes>
```



In [71]:

```
lut = dict(zip([1, 2, 3], "rbg"))
row_colors = [lut[y] for y in df_features_c['gesture']]

sns.clustermap(df_features_c, row_cluster=False, cmap='coolwarm', row_colors=rov
plt.show()
```



In [63]:

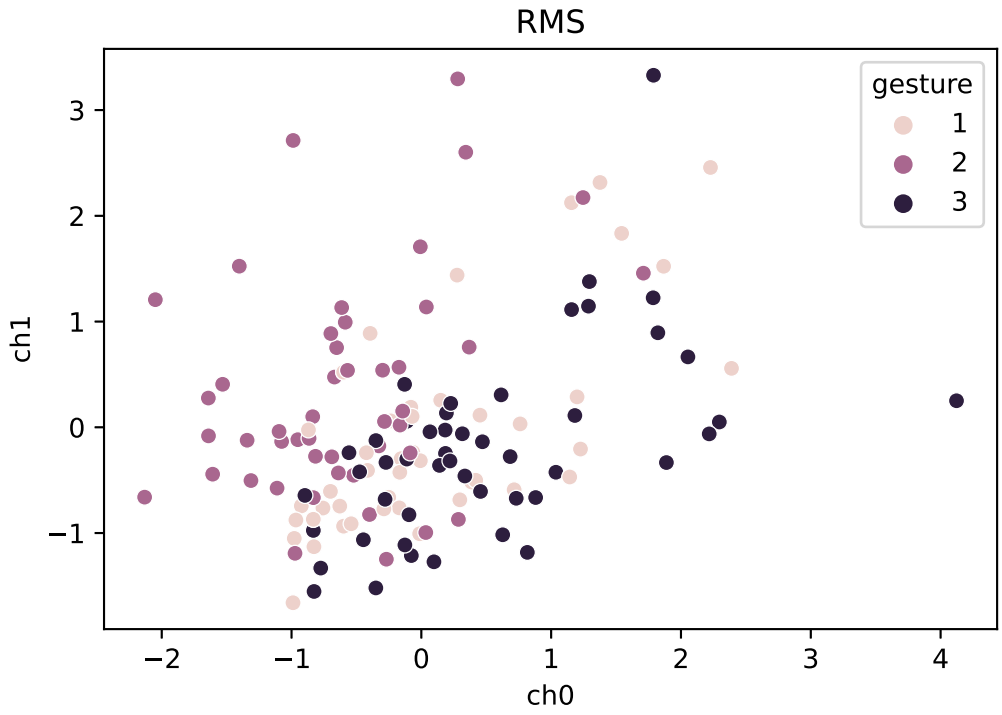
```
f = 35
print(feature_labels[f])
single_df = pd.DataFrame(features[f, :, :], columns=['ch0', 'ch1', 'ch2'])
single_df.loc[:, 'gesture'] = y_g_w

plt.figure()
plt.title(feature_labels[f])
# plt.subplots(figsize=(20,15))
sns.scatterplot(data=single_df, x="ch0", y="ch1", hue='gesture')
```

```
RMS
```

Out[63]:

```
<AxesSubplot:title={'center':'RMS'}, xlabel='ch0', ylabel='ch1'>
```


RMS
gesture
1
2
3
ch1
ch0
3
2
1
0
−1
−2
−1
0
1
2
3
4