contingency_table

February 17, 2025

```
[52]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import seaborn as sns
      import plotly.express as px

      from scipy.cluster.hierarchy import linkage, dendrogram, fcluster

      import pickle as pkl
```

```
[53]: with open('dense_rank_distance_matrix_experts_BSU1_BSU2_df.pkl', 'rb') as f:
          dense_rank_distance_matrix_experts_BSU1_BSU2_df = pkl.load(f)

      with open('kmeans_rank_distance_matrix_experts_BSU1_BSU2_df.pkl', 'rb') as f:
          kmeans_rank_distance_matrix_experts_BSU1_BSU2_df = pkl.load(f)
```

```
[54]: with open('dense_rank_linkage_matrix_experts_BSU1_BSU2.pkl', 'rb') as f:
          dense_rank_linkage_matrix_experts_BSU1_BSU2 = pkl.load(f)

      with open('kmeans_rank_linkage_matrix_experts_BSU1_BSU2.pkl', 'rb') as f:
          kmeans_rank_linkage_matrix_experts_BSU1_BSU2 = pkl.load(f)
```

1 Cluster Assignments

1.1 Clusters for 30 trials

```
[55]: # Perform hierarchical clustering on columns (trials/audio samples) for both␣
        ↪distance matrices
      linkage_matrix_dense_audio =␣
        ↪linkage(dense_rank_distance_matrix_experts_BSU1_BSU2_df.T, method='ward')
      linkage_matrix_kmeans_audio =␣
        ↪linkage(kmeans_rank_distance_matrix_experts_BSU1_BSU2_df.T, method='ward')

      # Extract 5 clusters from both hierarchical trees
      num_clusters_trials = 5
      dense_audio_clusters = fcluster(linkage_matrix_dense_audio,␣
        ↪num_clusters_trials, criterion='maxclust')
```

```
kmeans_audio_clusters = fcluster(linkage_matrix_kmeans_audio,␣
 ↪num_clusters_trials, criterion='maxclust')

# Create DataFrames mapping audio samples to their clusters
dense_audio_cluster_df = pd.DataFrame({
    'Audio Sample': dense_rank_distance_matrix_experts_BSU1_BSU2_df.columns,
    'Dense Cluster': dense_audio_clusters
})

kmeans_audio_cluster_df = pd.DataFrame({
    'Audio Sample': kmeans_rank_distance_matrix_experts_BSU1_BSU2_df.columns,
    'KMeans Cluster': kmeans_audio_clusters
})

# Merge to align the clusters from both methods
merged_audio_clusters = dense_audio_cluster_df.merge(kmeans_audio_cluster_df,␣
 ↪on="Audio Sample")

# Build a contingency table comparing the two clustering results
contingency_table_trials = pd.crosstab(merged_audio_clusters['Dense Cluster'],␣
 ↪merged_audio_clusters['KMeans Cluster'])

# Display the contingency table
contingency_table_trials
```

[55]:

```
KMeans Cluster  1  2  3  4  5
Dense Cluster
1               1  0  2  0  2
2               1  0  3  3  2
3               0  1  0  1  1
4               2  1  2  2  0
5               0  1  2  1  2
```

1.2 Clusters for 42 Subjects

[56]:

```
# Perform hierarchical clustering on rows (subjects) for both distance matrices
linkage_matrix_dense_subjects =␣
 ↪linkage(dense_rank_distance_matrix_experts_BSU1_BSU2_df, method='ward')
linkage_matrix_kmeans_subjects =␣
 ↪linkage(kmeans_rank_distance_matrix_experts_BSU1_BSU2_df, method='ward')

# Extract 3 clusters from both hierarchical trees
num_clusters_subjects = 3
dense_subject_clusters = fcluster(linkage_matrix_dense_subjects,␣
 ↪num_clusters_subjects, criterion='maxclust')
kmeans_subject_clusters = fcluster(linkage_matrix_kmeans_subjects,␣
 ↪num_clusters_subjects, criterion='maxclust')
```

```
# Create DataFrames mapping subjects to their clusters
dense_subject_cluster_df = pd.DataFrame({
    'Subject': dense_rank_distance_matrix_experts_BSU1_BSU2_df.index,
    'Dense Cluster': dense_subject_clusters
})

kmeans_subject_cluster_df = pd.DataFrame({
    'Subject': kmeans_rank_distance_matrix_experts_BSU1_BSU2_df.index,
    'KMeans Cluster': kmeans_subject_clusters
})

# Merge to align clusters from both methods
merged_subject_clusters = dense_subject_cluster_df.
 ↪merge(kmeans_subject_cluster_df, on="Subject")

# Build a contingency table for subjects
contingency_table_subjects = pd.crosstab(merged_subject_clusters['Dense␣
 ↪Cluster'], merged_subject_clusters['KMeans Cluster'])

# Display the contingency table
contingency_table_subjects
```

```
[56]: KMeans Cluster   1  2   3
      Dense Cluster
      1                1  2  14
      2                9  0   5
      3               10  0   1
```