TECHNISCHE
UNIVERSITÄT
DARMSTADT

Clinical Applications of Brain Imaging, Stimulation, and Modeling

Special Exercise 3

Talha Bhatti (2667250)

Contents

List of Figures

1 Neural Correlates of the Risk for Schizophrenia and Bipolar Disorder: A Meta-Analytic Overview

Schizophrenia (SCZ) and bipolar disorder (BD) are two major psychiatric disorders that are often characterized by overlapping symptoms and clinical/genetic profiles. A crucial step in developing early biomarkers for SCZ and BD patients and employing preventive strategies involve the understanding of neural basis of these disorders. One promising approach is to study individuals who are genetically at risk but do not yet exhibit symptoms as this can reveal early neural alterations linked to vulnerability. This essay explores one such meta-analysis by Cattarinussi et al. 2022 about the structural and functional MRI studies focusing on unaffected first-degree relatives (RELs) of SCZ and BD patients. RELs provide researchers a chance to study intermediate phenotypes, heritable traits often associated with disease risk, without interfering factors like medication use.

1.1 Study Inclusion, Methods, and Populations

1.1.1 Neuroimaging Modalities and Paradigms

Cattarinussi et al. 2022 used 230 studies in their meta-analytsis consisting of structural MRI and task-based fMRI, excluding modalities such as resting-state fMRI or diffusion imaging. In partiuclar, they analyzed 26 VBM studies and 81 task fMRI studies, including cognitive, emotional and reward paradigms. Some ROI-based volumetric studies were also included, especially the ones focusing on subcortical regions like the hippocampus, amygdala, striatum and thalamus. The exclusion criteria made sure to omit the studies lacking stereotaxic coordinates, those with subjects at clinical high risk and non-MRI/fMRI modalities.

1.1.2 Demographics and Sample Characteristics

Following are some of the key characteristics of the meta-analysis data:

- The data comes from 6274 SCZ-RELs, 1900 BD-RELs and 10,789 healthy controls.
- Participants' age range varies widely with mean age over 18 years.
- Relatives are often first-degree relatives, siblings or offspring.

1.1.3 Meta-Analytic Techniques

The meta-analytic technqiue used by authors is coordinate-based meta-analysis (CBMA) utilizing Activation Likelihood Estimation (ALE) implemented in GingerALE software, along with volume-based meta-analyses (VMA) conducted using the Jamovi and R statistical environments.

- ALE identifies consistent neural activation or structural differences across studies based on peak coordinates.
- VMA quantifies standardized mean differences in volume between groups.

The also adjusted the statistical thresholds for sample size and applied corrections for multiple comparisons using family-wise error rates.

1.2 Key Findings

1.2.1 Common Abnormalities in Both Disorders

One of the key findings of the study was the reduced thalamic volume in both SCZ-RELs and BD-RELs compared to HCs, which suggests that thalamic alterations may represent a shared intermediate phenotype, tied to general risk of psychosis in the individual. This finding also supports prior work, highlighting the involvement of thalamus in cognitive and affective processing disruptions, which is common in both disorders.

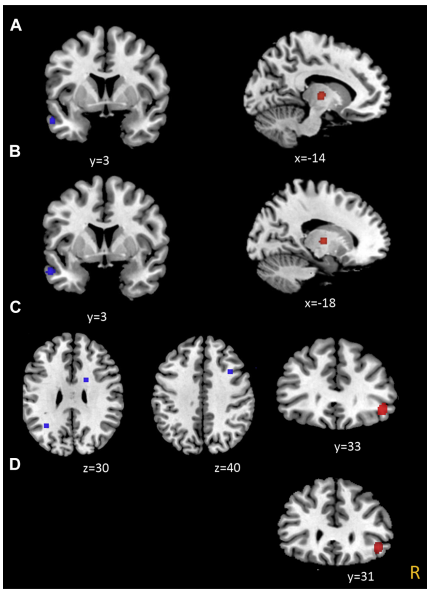

Figure 1: Coordinate-based meta-analysis of structural differences in SCZ-RELs and BD-RELs

1.2.2 Schizophrenia Relatives: Corticostriatal-Thalamic Alterations

In SCZ-RELs, structural and functional anomalies converged in the dorsolateral prefrontal cortex (DLPFC), particularly the Brodmann area (BA). This region is critical for working memory, planning and executive function, some of the functions which are consistently impaired in SCZ. Increased activity in the DLPFC during cognitive tasks may be due to some compensatory mechanism where the brain uses additional resources to perform the task. Additionally, the study notes the reduction in the middle temporal gyrus, hippocampus, striatum and global gray matter volume (GMV), which align with some already-known deficits in SCZ patients.

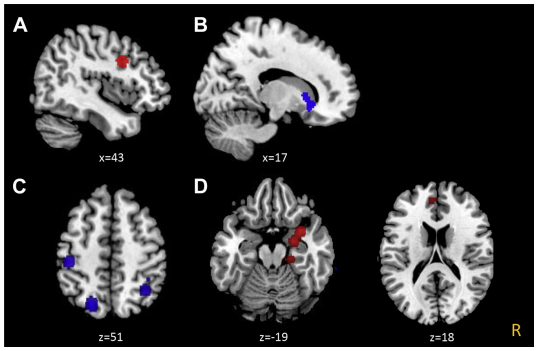

Figure 2: Functional CBMA of SCZ-RELs during cognitive tasks

These findings indicate disruptions in corticostriatal-thalamic circuits, long associated with deficits in salience attribution, goal-directed behavior and cognitive control in SCZ. Decreased hippocampal volume in SCZ-RELs compared to BD-RELs further supports this pattern.

1.2.3 Bipolar Disorder Relatives: Thalamocortical and Limbic Dysregulation

On the other hand, BD-RELs showed increased activation in limbic and prefrontal area, particularly in the parahippocampal gyrus (PHG) and ventrolateral prefrontal cortex (VLPFC), the regions which are generally involved in emotion regulation and reward processing. Structural analysis also showed an increase in the VLPFC volume along with reduced GMV in the temporoparietal junction and dorsomedial prefrontal cortex. These abnormalities align with emotion regulation difficulties observed in both BD patients and their unaffected relatives.

An interesting thing to note here is that the BD type I relatives shared some features with SCZ-RELs, such as DLPFC hyperactivation during cognitive tasks, which sheds lights on shared genetic vulnerabilities.

1.3 Discussion

1.3.1 Functional Relevance of Affected Regions

The DLPFC is important for many executive functions and is commonly affected in patients and at-risk populations. Its dysfunction may be an important biomarker for cognitive symptoms of SCZ. On the other hands, VLPFC regulates emotional responses and social cognition. Structural enlargement and hyperactivation in this region may be attributed to maladaptive emotional control strategies which possibly contributes to BD symptoms.

The thalamus, which is the central relay hub between subcortical areas and cortex, is reduced in both groups. Its disruption may represent a fundamental vulnerability factor in psychosis related disorders.

1.3.2 Reasons for Inconsistent Findings

Despite some robust trends, the authors acknowledge some significant inconsistency across studies. Some of the common factors which contribute to these consistencies are variations in MRI protocols, task paradigms, sample demographics (age, kinship, comorbidities) and analytical strategies. Some of the common mitigation strategies for consistent results include meta-analytic subgrouping, outlier removal and control for sample size.

1.3.3 Genetic and Environmental Risk Factors

Although this meta-analysis centered on neural correlates, its results are consistent with genetic studies showing that SCZ and BP share a considerable amount of polygenic risk. The study didn't concentrate on environmental or lifestyle risk factors, but the authors point out that intermediate phenotypes like brain structure are generally less influenced by outside factors than behavioral symptoms.

1.4 Clinical Implications and Future Work

This meta-analysis builds on the previous works by including both structural and functional modalities in one framework. Compared to prior ENIGMA studies de Zwarte et al. 2019, which focused on ROI-based structural volumes, this study adds voxel-level resolution and functional activation data. The finding of reduced hippocampal volume in SCZ-RELs is different from ENIGMA results, perhaps due to the different inclusion criteria and analysis techniques.

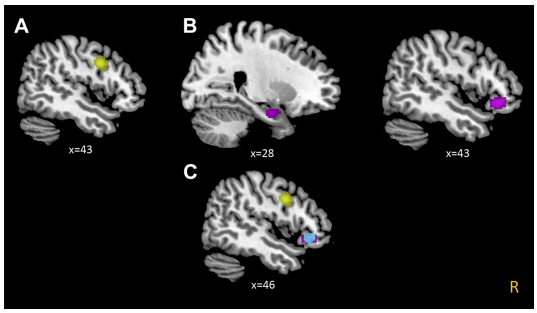

Figure 3: All-effects meta-analysis showing convergent alterations in SCZ-RELs and BD-RELs

Identification of intermediate phenotypes like DLPFC dysfunction or thalamic volume reduction may eventually assist in early detection strategies or preventive interventions. But Cattarinussi et al. 2022 emphasize on how the clinical translation is not possible still, given the current lack of specificity and the modest effect sizes of these alterations.

The authors also mention the importance of tracking RELs in the future to differentiate between individuals who actually develop the disorders. Future work could also explore

developmental trajectories, particularly since BD with early onset or psychotic features may follow different neurodevelopmental paths.

1.5 Conclusion

Cattarinussi et al. 2022 have compiled a comprehensive analysis of neural alterations which are associated with familial risk for SCZ and BD. Their findings suggest both shared and disorder-specific intermediate phenotypes. These alterations are likely caused by both genetical and developmental neurological conditions. Although it is difficult to apply these finding to clinical practices currently, this meta-analysis is a step forward in our understanding of how the structural and functional changes in brain relate to psychiatric risk. It also lays important groundwork for future research into these biomarkers.

References

Cattarinussi, Giulia et al. (2022). "Neural Correlates of the Risk for Schizophrenia and Bipolar Disorder: A Meta-analysis of Structural and Functional Neuroimaging Studies". In: *Biological Psychiatry* 92.5. Copy Number Variants and Related Risk Mechanisms, pp. 375–384. ISSN: 0006-3223. DOI: https://doi.org/10.1016/j.biopsych.2022.02.960. URL: https://www.sciencedirect.com/science/article/pii/S000632232201068X.

de Zwarte, Sonja M.C. et al. (2019). "The Association Between Familial Risk and Brain Abnormalities Is Disease Specific: An ENIGMA-Relatives Study of Schizophrenia and Bipolar Disorder". In: *Biological Psychiatry* 86.7. Schizophrenia: Genomics to Therapeutics, pp. 545–556. ISSN: 0006-3223. DOI: https://doi.org/10.1016/j.biopsych.2019.03.985. URL: https://www.sciencedirect.com/science/article/pii/S0006322319314374.

2 Python Exercises

2.1 Task 1

```
import numpy as np
import seaborn as sb
import scipy.stats as stats
import matplotlib.pyplot as plt
import pandas as pd

# functional network indices
networks = np.array(
      [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  1,  1,  1,  2,  2,
        2,  5,  5,  5,  5,  5,  5,  5,  5,  6,  6,  6,  6,  6,  6,  6,  6,
        6,  6,  6,  6,  6,  7,  7,  7,  3,  3,  3,  3,  3,  3,  4,  4,  4,
        4,  4,  8,  8,  9,  9,  9,  9, 13, 13, 13, 13, 13, 13, 13, 14, 14,
       14, 14, 10, 10, 10, 10, 10, 10, 10, 10, 11, 11, 11, 11, 11, 11, 11,
       11, 12, 12, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16,  0,  0,
        0,  0,  0,  0,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  2,  2,  2,
        5,  5,  5,  5,  5,  5,  6,  6,  6,  6,  7,  7,  7,  3,  3,  3,  3,
        3,  3,  4,  4,  4,  4,  4,  8,  8,  8,  8,  9,  9,  9,  9, 13, 13,
       13, 13, 13, 13, 13, 14, 14, 14, 14, 14, 14, 10, 10, 10, 10,
       10, 10, 10, 10, 10, 10, 10, 11, 11, 11, 11, 11, 11, 11, 12, 12, 12,
       12, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16])

# functional network names
netw_names = np.array(
    ['ContA', 'ContB', 'ContC',
     'DefaultA', 'DefaultB', 'DefaultC',
     'DorsAttnA', 'DorsAttnB',
     'LimbicA_TempPole', 'LimbicB_OFC',
     'SalVentAttnA', 'SalVentAttnB',
     'SomMotA', 'SomMotB',
     'TempPar',
     'VisCent', 'VisPeri'])

#%% Task 1: Compare graph strength for two populations  (7 points)

"""
Load the edge data from all 52 subjects.
The first 25 subjects have been diagnosed with a condition, let's say it is
    MCI,
   the other 27 are healthy controls.
For each frequency band:
  Compute the graph strength (mean edge strength) for each subject.
  Plot the distribution of subjects' values in each group with a violinplot (
    with dots).
  Run an appropiate test to see if the distributions of values
    are significantly different between the two groups.

"""

bands = ['delta', 'theta', 'alpha', 'beta', 'low-gamma', 'high-gamma']
total_subjects = 52
```

```
group_labels = ['MCI']*25 + ['Healthy']*27
results = []

for band in bands:
    subject_strength = []

    for subject_idx in range(total_subjects):
        file_path = f"iPLV/{band}/S{subject_idx}.npy"
        matrix = np.load(file_path)
        strength = matrix[np.triu_indices_from(matrix, k=1)].mean() #self
    connections exluded
        subject_strength.append(strength)
    df_band = pd.DataFrame({
        'SubjectID': np.arange(total_subjects),
        'Group': group_labels,
        'Strength': subject_strength,
        'Band': band
    })
    results.append((df_band))

df_all = pd.concat(results, axis=0, ignore_index=True)

for band in bands:
    df_band = df_all[df_all['Band']==band]
    plt.figure(figsize=(6,6))
    sb.violinplot(data=df_band, x='Group', y='Strength', cut=0, inner=None)
    sb.stripplot(data=df_band, x='Group', y='Strength', color='black', size=4,
     alpha=0.7)
    plt.title(f'Graph Strength by Group for {band.capitalize()} Band')
    plt.ylabel('Mean Edge Strength')
    plt.xlabel('Group')
    plt.tight_layout()
    plt.show()

    mci_values = df_band[df_band['Group']=='MCI']['Strength']
    hc_values = df_band[df_band['Group']=='Healthy']['Strength']
    stat, p_val = stats.mannwhitneyu(mci_values, hc_values, alternative='two-
    sided')
    print(f"{band.capitalize()} Band:")
    print(f"  Mann-Whitney U: statistic={stat}, p-value={p_val:.5f}")
    if p_val < 0.05:
        print("Significant difference between MCI and Healthy groups")
    else:
        print("No significant difference")
```

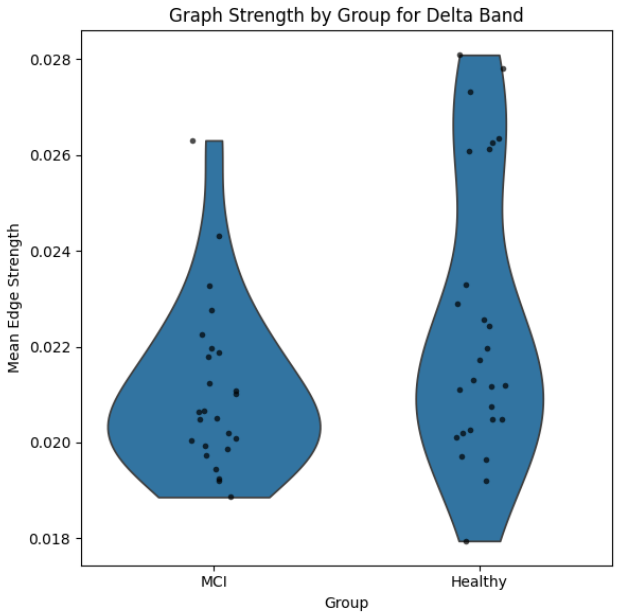

Figure 4: Graph Strength by Group for Delta Band

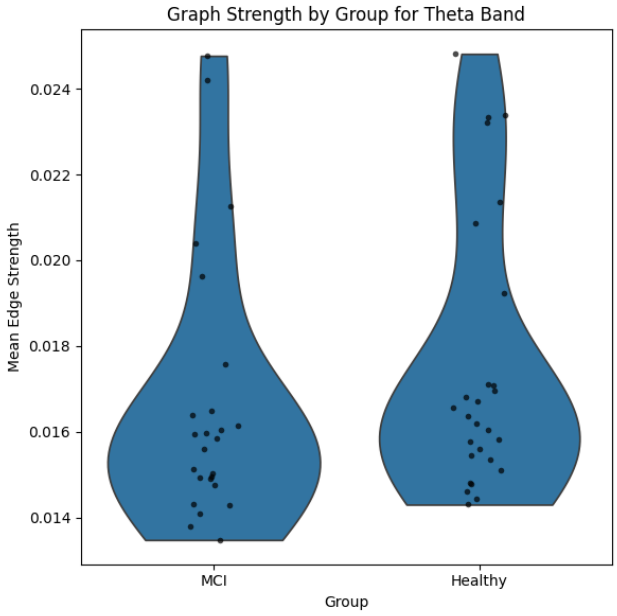

Figure 5: Graph Strength by Group for Theta Band

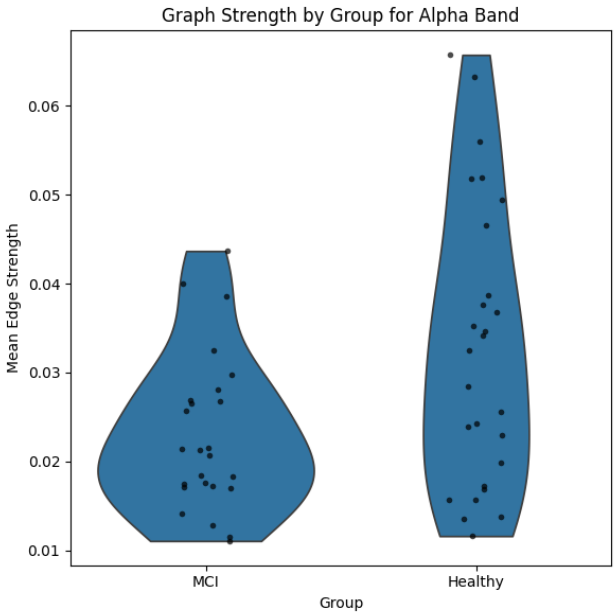

Figure 6: Graph Strength by Group for Alpha Band

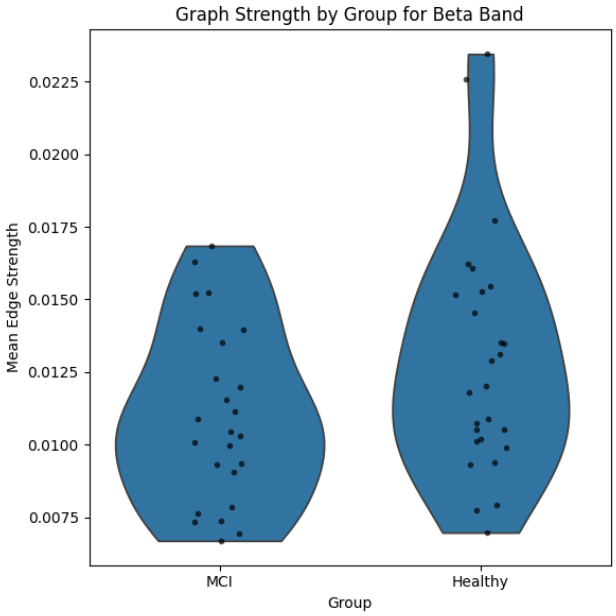

Figure 7: Graph Strength by Group for Beta Band

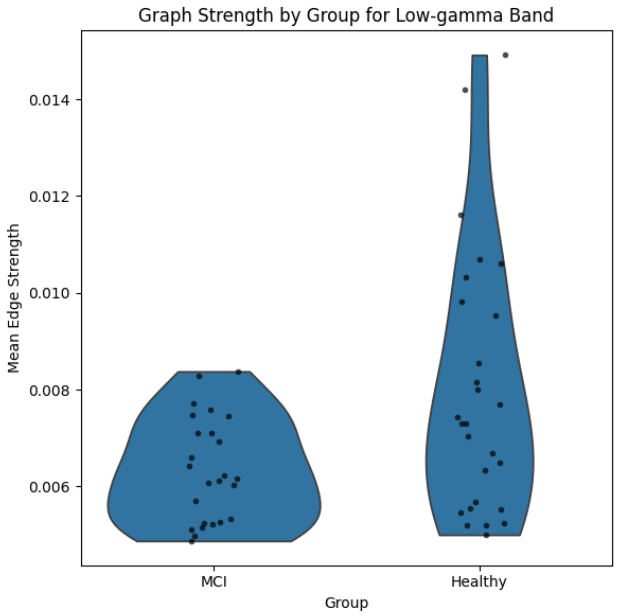

Figure 8: Graph Strength by Group for Low-gamma Band

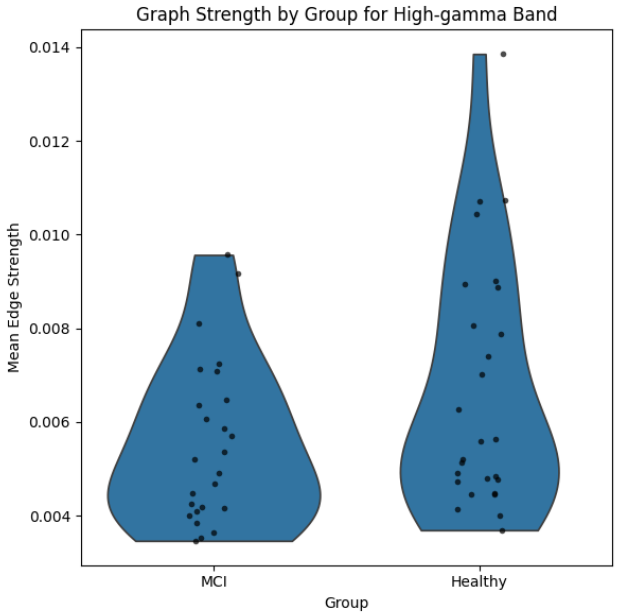

Figure 9: Graph Strength by Group for Higfh-gamma Band

```
Delta Band:
  Mann-Whitney U: statistic=242.5, p-value=0.08348
No significant difference
Theta Band:
  Mann-Whitney U: statistic=259.0, p-value=0.15313
```

```
No significant difference
Alpha Band:
  Mann-Whitney U: statistic=229.0, p-value=0.04792
Significant difference between MCI and Healthy groups
Beta Band:
  Mann-Whitney U: statistic=247.0, p-value=0.09929
No significant difference
Low-gamma Band:
  Mann-Whitney U: statistic=215.0, p-value=0.02546
Significant difference between MCI and Healthy groups
High-gamma Band:
  Mann-Whitney U: statistic=256.0, p-value=0.13794
No significant difference
```

2.2 Task 2

```python
#%% Task 2: Compare node strengths in functional networks  (8 points)

"""
Compute the node strength for each subject and frequency band.
For each frequency band and functional network:
    Compute the average node strength for this network.
    Run a test to see if values differ significantly between groups.
    Make a heatmap plot that shows the significant differences.
    Label axes for frequency bands and networks, center z-axis on 0.

"""
node_strength_data = []
for band in bands:
    for subject_idx in range(total_subjects):
        file_path = f"iPLV/{band}/S{subject_idx}.npy"
        matrix = np.load(file_path)
        np.fill_diagonal(matrix, 0)
        node_strengths = matrix.sum(axis=1)
        for node_idx, n_str in enumerate(node_strengths):
            node_strength_data.append({
                'SubjectID': subject_idx,
                'Group': group_labels[subject_idx],
                'Band': band,
                'Node': node_idx,
                'NodeStrength': n_str
            })

df_node_strength = pd.DataFrame(node_strength_data)
# compute average node strength for each band
average_strength_data = []

for (band, subject_id), sub_df in df_node_strength.groupby(['Band', 'SubjectID
    ']):
    group_label = sub_df['Group'].iloc[0]  # MCI or Healthy
```

```
    for net_id in range(len(netw_names)):
        # find which rows in sub_df correspond to net_id
        mask = (networks == net_id)
        network_nodes = sub_df[sub_df['Node'].isin(np.where(mask)[0])]
        if len(network_nodes) > 0:
            avg_str = network_nodes['NodeStrength'].mean()
        else:
            avg_str = np.nan
        average_strength_data.append({
            'SubjectID': subject_id,
            'Group': group_label,
            'Band': band,
            'NetworkID': net_id,
            'NetworkName': netw_names[net_id],
            'AvgNodeStrength': avg_str
        })

df_network_strength = pd.DataFrame(average_strength_data)

# comparing groups for each freq band and network
n_bands = len(bands)
n_networks = len(netw_names)
diff_matrix = np.zeros((n_bands, n_networks))
pval_matrix = np.ones((n_bands, n_networks))
for b_idx, band in enumerate(bands):
    for net_id in range(n_networks):
        df_sub = df_network_strength[
            (df_network_strength['Band'] == band) &
            (df_network_strength['NetworkID'] == net_id)
            ]
        mci_vals = df_sub[df_sub['Group'] == 'MCI']['AvgNodeStrength']
        hc_vals = df_sub[df_sub['Group'] == 'Healthy']['AvgNodeStrength']
        # compute difference in means
        mean_mci = np.mean(mci_vals)
        mean_hc = np.mean(hc_vals)
        diff_matrix[b_idx, net_id] = mean_mci - mean_hc

        # Mann-Whitney test
        if len(mci_vals) > 0 and len(hc_vals) > 0:
            stat, pval = stats.mannwhitneyu(mci_vals, hc_vals, alternative='
    two-sided')
            pval_matrix[b_idx, net_id] = pval
        else:
            pval_matrix[b_idx, net_id] = np.nan

annot_matrix = np.empty_like(pval_matrix, dtype=object)
annot_matrix[:] = ''
annot_matrix[pval_matrix < 0.05] = '*'

plt.figure(figsize=(12, 5))
ax = sb.heatmap(
    diff_matrix,
    center=0,
    cmap='bwr',
```

```
    annot=annot_matrix,
    fmt='',
    xticklabels=netw_names,
    yticklabels=bands,
    cbar_kws={'label': 'Difference in Mean Node Strength (MCI - Healthy)'}
)
ax.set_xlabel('Functional Network')
ax.set_ylabel('Frequency Band')
ax.set_title('Differences in Node Strength by Network and Band')
plt.tight_layout()
plt.show()
```

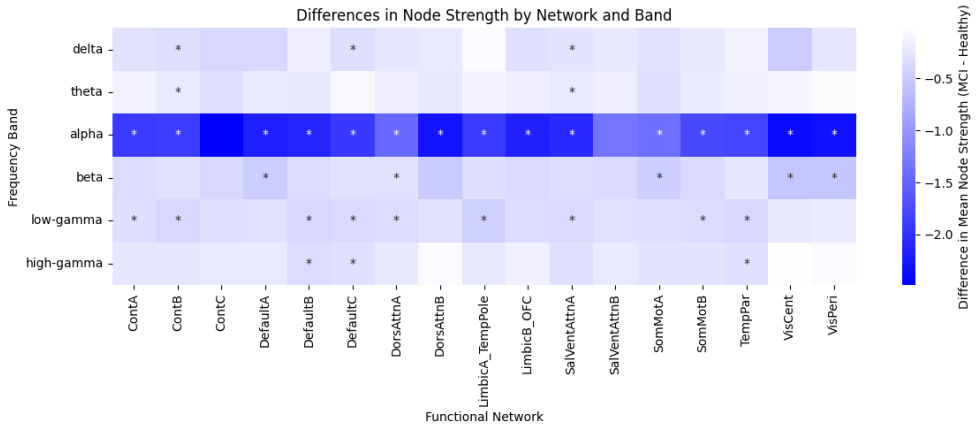

Figure 10: Differences in Node Strength by Network and Band

2.3 Task 3

```
#%% Task 3: Get number of significant edges. ( 10 points)

"""

For each frequency band and edge, test if the edge strength values differ
  significantly between groups.
For each frequency band, get the number of significant edges.
In which band(s) is the number of significant edges higher
   than would be expected by chance?
Is the overal number of significant edges higher than expected by chance?

"""

adjacency_data = {band: [] for band in bands}
test_file = f"iPLV/{bands[0]}/S0.npy"
temp_matrix = np.load(test_file)
N = temp_matrix.shape[0]
for band in bands:
    for subject_idx in range(total_subjects):
        file_path = f"iPLV/{band}/S{subject_idx}.npy"
        mat = np.load(file_path)
```

```
        adjacency_data[band].append(mat)

# for each band  M a n n Whitney  U (MCI vs Healthy) test for each edge
alpha = 0.05
band_significant_counts = {}
overall_significant_count = 0
total_edges_all_bands = 0

for band in bands:
    n_significant = 0
    # upper triangle edges
    total_edges = N * (N - 1) // 2
    for i in range(N):
        for j in range(i + 1, N):
            # gather MCI vs Healthy edge values
            edge_vals_mci = []
            edge_vals_hc = []
            for s_idx in range(total_subjects):
                edge_val = adjacency_data[band][s_idx][i, j]
                if group_labels[s_idx] == 'MCI':
                    edge_vals_mci.append(edge_val)
                else:
                    edge_vals_hc.append(edge_val)
            if len(edge_vals_mci) > 0 and len(edge_vals_hc) > 0:
                stat, pval = stats.mannwhitneyu(edge_vals_mci, edge_vals_hc,
    alternative='two-sided')
                if pval < alpha:
                    n_significant += 1

    band_significant_counts[band] = n_significant
    overall_significant_count += n_significant
    total_edges_all_bands += total_edges
    print(f"Band '{band}': {n_significant} significant edges out of {
    total_edges} (p < 0.05).")

# binomial test to check significant edges in each band

print("\nChecking if significant edges in each band")
for band in bands:
    count_sig = band_significant_counts[band]
    n_edges = N * (N - 1) // 2
    pval_binom = stats.binomtest(count_sig, n_edges, 0.05, alternative='
    greater')
    print(f"Band '{band}': {count_sig} / {n_edges} edges, "
          f"binomial test p-value={pval_binom.pvalue:.5f}. "
          f"\nNumber of significant edges higher than would be expected by
    chance? {pval_binom.pvalue < 0.05}")


print(f"Total significant edges across all bands = {overall_significant_count
    }, out of {total_edges_all_bands} total edges tested")
overall_pval_binom = stats.binomtest(overall_significant_count,
    total_edges_all_bands,0.05, alternative='greater')
print(f"Overall binomial test p-value={overall_pval_binom.pvalue:.5f}. "
```

```
      f"\nOverall number of significant edges higher than expected by chance?
    {overall_pval_binom.pvalue < 0.05}")
```

```
Band 'delta': 975 significant edges out of 19900 (p < 0.05).
Band 'theta': 961 significant edges out of 19900 (p < 0.05).
Band 'alpha': 2976 significant edges out of 19900 (p < 0.05).
Band 'beta': 1356 significant edges out of 19900 (p < 0.05).
Band 'low-gamma': 1957 significant edges out of 19900 (p < 0.05).
Band 'high-gamma': 1738 significant edges out of 19900 (p < 0.05).

Checking if significant edges in each band
Band 'delta': 975 / 19900 edges, binomial test p-value=0.74668.
Number of significant edges higher than would be expected by chance? False
Band 'theta': 961 / 19900 edges, binomial test p-value=0.86938.
Number of significant edges higher than would be expected by chance? False
Band 'alpha': 2976 / 19900 edges, binomial test p-value=0.00000.
Number of significant edges higher than would be expected by chance? True
Band 'beta': 1356 / 19900 edges, binomial test p-value=0.00000.
Number of significant edges higher than would be expected by chance? True
Band 'low-gamma': 1957 / 19900 edges, binomial test p-value=0.00000.
Number of significant edges higher than would be expected by chance? True
Band 'high-gamma': 1738 / 19900 edges, binomial test p-value=0.00000.
Number of significant edges higher than would be expected by chance? True
Total significant edges across all bands = 9963, out of 119400 total edges tested
Overall binomial test p-value=0.00000.
Overall number of significant edges higher than expected by chance? True
```

2.4 Task 4

```
#%% Task 4: Connected spiking-neuron model (18 points)

"""
Now, we will pick up the Ihzikevich model that we also used in Ex. 6 and
    simulate a neuronal population consisting of a large number (N=1000)
    of interacting neurons. We will also add weights between the neurons,
    and random external input.

The model will run for 1000 time points of 1 msec each.

The 1000 neurons will be partially of excitatory (E) and inhibitory (I) type.
  Choose a ratio E/I that is likely to produce oscillation-like behaviour.
  (Check the lectures!)

The E and I neurons can have different values for a,b,c,d parameters.
All E neurons will have the same value, and all I neurons will
   have the same value for each parameter.
```

```
Both the initial values and the updated values at each time point will
  be calculated as before, only for each neuron separately.

The 1000x1000 weight matrix W will represent the effect that each neuron has
    on all the others.
  (We assume that each neuron is connected to each)
For each neuron, the weights for inputs coming from the E-neurons should be
    drawn from a uniform [0,0.5] distribution
   and for inputs coming from the I-neurons drawn from a uniform [-1,0]
    distribution.

The equation for the external input current as well the
   equation that adds the currents from other cells are already in the code.

Write to a variable whenever a neuron is firing: which neuron and which time
    point.
Then make a plot of all neurons firing over time where each firing is a small
    dot.
Time should be on x-axis, neurons on y-axis.

The output should show oscillation-like behaviour in the collective firing
   after a short while!

If you don't get oscillation-like firing, run again a few times.
If not, check your code if it updates values correctly.
You can also try to change the parameters.

What is the approximate oscillation frequency in your plot?
    (may vary from run to run)

"""

## given parameters

simulation_time = 1000
dt = 1
# Excitatory neurons                Inhibitory neurons
Ne = 800 ;                            Ni = 200
N_tot = Ne + Ni
exc_indices = np.arange(0, Ne)
inh_indices = np.arange(Ne, N_tot)

a1 = .02    ;                         a2 = .025
b1 = .25    ;                         b2 = .34
c1 = -60    ;                         c2 = -65
d1 = 6      ;                         d2 = 2

v0    = -65.5
v_thr =  35.5


# create arrays for parameters
a = np.zeros(N_tot)
```

```
b = np.zeros(N_tot)
c = np.zeros(N_tot)
d = np.zeros(N_tot)
a[exc_indices] = a1;  a[inh_indices] = a2
b[exc_indices] = b1;  b[inh_indices] = b2
c[exc_indices] = c1;  c[inh_indices] = c2
d[exc_indices] = d1;  d[inh_indices] = d2

# initial values of v and u for each neuron
v = np.full(N_tot, v0, dtype=float)
u = b * v

# set the matrix W of all-to-all synaptic weights
W = np.zeros((N_tot, N_tot), dtype=float)
for idx in exc_indices:
    W[:, idx] = np.random.uniform(0, 0.5, size=N_tot)
for idx in inh_indices:
    W[:, idx] = np.random.uniform(-1, 0, size=N_tot)


# initializations

spike_times = []  # (t, neuron_index)
example_inh_neuron = Ne  # to record membrane voltage of 800th index
    inhibitory neuron
v_record = np.zeros(simulation_time)

# loop through simulation time
for t in range(simulation_time):

    I = np.hstack((5*np.random.randn(Ne),2*np.random.randn(Ni)))
    # check which neurons will fire an action potential
    fired = np.where(v >= v_thr)[0]
    if len(fired) > 0:
        for neuron_id in fired:
            spike_times.append((t, neuron_id))
        # update membrane variables for neurons that fired
        v[fired] = c[fired]
        u[fired] += d[fired]
    # add currents from neurons that fired
    if len(fired) > 0:
        I += np.sum(W[:,fired], axis=1)
  # update membrane voltage
    v += (0.04 * v ** 2 + 5 * v + 140 - u + I) * dt
    u += (a * (b * v - u)) * dt
    v_record[t] = v[example_inh_neuron]
spike_times = np.array(spike_times)
mean_firing_rate = len(spike_times) / (N_tot * (simulation_time/1000))

print("Mean firing rate: %.2f Hz" % mean_firing_rate)

plt.figure(figsize=(10,5))
plt.scatter(spike_times[:,0], spike_times[:,1], s=2, marker='.', color='k')
plt.title('Population Firing Raster Plot')
```

```
plt.xlabel('Time (ms)')
plt.ylabel('Neuron')
plt.xlim([0, simulation_time])
plt.ylim([-1, N_tot+1])
plt.tight_layout()
plt.show()

# fast spiking voltage of inhibitory neuron
plt.figure(figsize=(10,5))
time_array = np.arange(simulation_time)
plt.plot(time_array, v_record, color='b')
plt.title('Inhibitory Neuron Voltage Trace')
plt.xlabel('Time (ms)')
plt.ylabel('mV')
plt.ylim([-90, 50])
plt.grid(True)
plt.tight_layout()
plt.show()
```

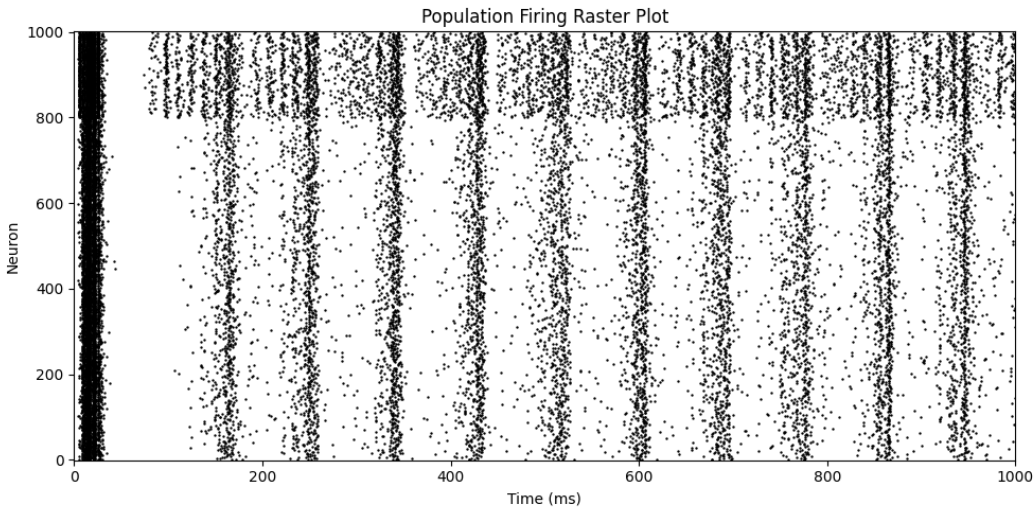

Figure 11: Population Firing Raster Plot

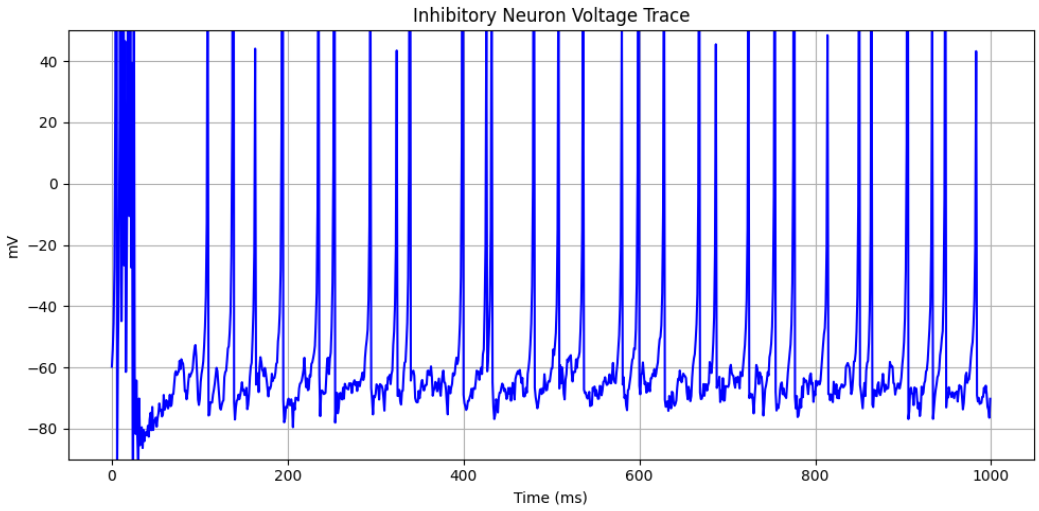

Figure 12: Inhibitory Neuron Voltage Trace

```
Mean firing rate: 20.55 Hz
```

2.5 Task 5

```python
#%% Task 5: Alter firing frequency in the model (7 points)

"""
Now, change the parameters of the model so that you get approximately the
   double firing frequency as before in your firing pattern.

"""
spike_times = []
for t in range(simulation_time):
    I = np.hstack((10*np.random.randn(Ne),2*np.random.randn(Ni))) # doubled
    the excitatory input
    fired = np.where(v >= v_thr)[0]
    if len(fired) > 0:
        for neuron_id in fired:
            spike_times.append((t, neuron_id))
        v[fired] = c[fired]
        u[fired] += d[fired]
    if len(fired) > 0:
        I += np.sum(W[:,fired], axis=1)
    v += (0.04 * v ** 2 + 5 * v + 140 - u + I) * dt
    u += (a * (b * v - u)) * dt

spike_times = np.array(spike_times)
mean_firing_rate = len(spike_times) / (N_tot * (simulation_time/1000))
print("Mean firing rate with double excitatory input: %.2f Hz" %
    mean_firing_rate)

plt.figure(figsize=(10,5))
plt.scatter(spike_times[:,0], spike_times[:,1], s=2, marker='.', color='k')
```

```
plt.title('Population Firing Raster Plot (doubled excitatory input)')
plt.xlabel('Time (ms)')
plt.ylabel('Neuron')
plt.xlim([0, simulation_time])
plt.ylim([-1, N_tot+1])
plt.tight_layout()
plt.show()
```

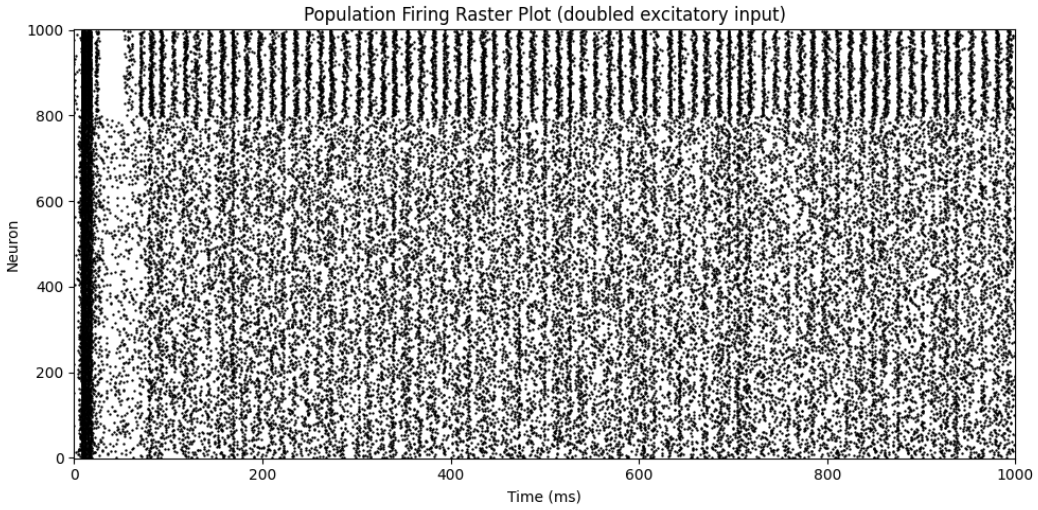

Figure 13: Population Firing Raster Plot (doubled excitatory input)

```
Mean firing rate with double excitatory input: 40.18 Hz
```