Internship Summary

盛晓宇

1 Brain Network Analysis

在这一部分, 我的主要工作是分析大脑功能连接在不同刺激下的结构变换, 主要的步骤总结为以下这些:

1. **Data Processing:** 陈婕学姐给我的数据包含了四组神经调控数据 (sham、tdcs、tacs6hz、tacs10hz), 每组数据有三个时间点 (调控前 s_1, 调控中 s_2 以及调控后 s_3). 一共有 20 个被试, 每个被试都有一个 360×360 的矩阵. 每个被试的功能连接矩阵有两种 format(基于相关性系数 FC, 在 FC 基础上进行费雪变换的 ZFC).
   我这里主要是针对 FC 进行了测试, 对于每个功能连接矩阵的 thresholding 我选择了 0.3 和 0.5 两种方式, 即低于 threshold 的 edge weight 调整为 0, 高于 threshold 的 edge weight 保持不变. 这里 threshold=0.3, 0.5 是因为在这边文章里 [LS20], 作者说这两个 threshold 的 represtativeness 最好.

2. **Graph Properties Analysis:** 针对每一个功能连接矩阵, 我主要分析的以下的 graph properties:

   - *degree centrality:* The number of neighbours a node has. It helps identify highly connected regions in the brain.
   - *betweenness centrality:* Measures the extent to which a node lies on the shortest path between other nodes. It identifies critical nodes for information or stimulation flow.
   - *eigenvector centrality:* Assigns relative scores to nodes based on their connections to high-scoring nodes. It identifies influential nodes, which may help pinpoint regions that influence brain activity.
   - *clustering coefficient:* Indicates the degree to which nodes tend to cluster together. It measures local connectivity and can reveal changes in regional connectivity patterns in response to stimuli.
   - *modularity:* Quantifies the strength of division of a network into modules (communities). High modularity suggests strong community structure, which could indicate functional segregation of brain regions under different conditions. (This concept is similar to the definition of the clustering coefficient, but the clustering coefficient focuses on local community structures, modularity assesses global community divisions.)

- rich club coefficient: Measures the tendency of high-degree nodes to form tightly interconnected communities. It identifies nodes that may play a crucial role in network resilience and information processing.
- participation coefficient: Assesses how well a node connects with different modules. It indicates the diversity of a node’s connections across the network, useful for understanding how different brain regions integrate information.
- connected components: Counts the number of connected subgraphs. This can reveal how brain regions fragment or connect under different stimuli.
- minimum spanning tree: A tree that spans all nodes with the minimum possible total edge weight. It is used to identify the most efficient pathways in the brain’s structural network (if the graph is disconnected, then it will compute a forest with the minimum possible total edge weight.)
- maximum spanning tree: Similar to the minimum spanning tree but maximises total edge weight, emphasising the strongest connections in the brain network.
- cut vertices: Nodes whose removal increases the number of connected components. Identifying these nodes can help determine critical points in the brain network that, when disrupted, fragment the network.
- biconnected components: Subgraphs where removing any single node does not disconnect the network. These components represent robust regions in the brain network that maintain connectivity despite node removal, important for understanding network resilience.

其中前 7 个是 network neuroscience 这个领域文章里经常提及的 properties, 后面 5 个 properties 是我自己加的一些可能比较能反应大脑结构变换的 properties.

3. **Results:** 针对不同的刺激类型, 我计算了每个 graph property 相同刺激时间点的 T-Statistic across 20 个被试 (有些 graph perproty 是 node-wise, 如 centrality, 这种情况下该 property 的值取的是所以 node 的 average). 以下为 threshold=0.3 时比较有 statistical significance 的 results:

   - *sham vs tacs6 (s1, s2, s3):*
     - degree centrality: -1.51, -2.63, -2.06
     - modularity: -1.08, -1.97, -1.38
     - rich club coefficient: 0.93, 3.45, 1.43
     - participation coefficient: 0.7, -0.7, -0.36
     - minimum spanning tree: 0.66, 1.51, 2.28
     - connected component: 1.79, 0.85, 2.12

- cut vertices: 2.74, 1.74, 1.51
  - biconnected component: 2.67, 1.78, 1.84
- *sham vs tacs10 (s1, s2, s3):*
  - degree centrality: -0.039, -0.95, -1.16
  - modularity: -0.17, -0.88, -0.61
  - rich club coefficient: 1.23, 0.51, 1.4
  - participation coefficient: 0.47, -0.6, -0.89
  - minimum spanning tree: 0.18, 1.53, 0.98
  - connected component: 0.47, -0.16, 1.53
  - cut vertices: 0.68, 0.80, 1.56
  - biconnected component: 0.61, 0.75, 1.72
- *sham vs tdcs (s1, s2, s3):*
  - degree centrality: -0.7, -0.81, -1.57
  - modularity: -0.6, -0.46, -0.67
  - rich club coefficient: 0.2, 1.66, 1.81
  - participation coefficient: 1.54, -0.35, -0.61
  - minimum spanning tree: 0.95, 1.97, 3.63
  - connected component: 0.63, 0.18, 1.50
  - cut vertices: 1.03, 0.76, 1.82
  - biconnected component: 0.89, 0.82, 1.98

It is worth mentioning that some graph properties are highly sensitive to the threshold. For example, connected components, cut vertices, and biconnected components may show no variation across all graphs if the graphs are too dense.

2 Network Control Theory

在这一部分, 我主要是根据这个领域的论文和 nctpy 这个 package 去 implement 一个 linear model 去模拟 brain dynamics. 在给定 initial 和 target state 的情况下, 去对每个刺激节点所需的 energy 进行排序, 以求找到所需 energy 比较小的刺激节点完成 state transition(from initial state to target state)

- **Linear model:** if the system is continuous:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

if discrete:

$$x(t+1) = Ax(t) + Bu(t)$$

where $x(t)$ is a $N \times 1$ vector that represents the brain state at a given time, and N is the number of ROI (node). A is the structural connectivity

matrix. $u(t)$ is a $N \times 1$ vector that represents the input required to control the system. B is a $N \times N$ input matrix whose diagonal entries select the regions that will receive input, and this set of selected regions is referred to as the control set.
Energy is defined as:

$$E = \int_0^T \|Bu(t)\|_2^2 dt$$

where T is the amount of time given to model to reach the target state.

- **Implementation:** 我的 implementation 主要 based on nctpy 这个 package. specially, 我 implement 以下这些 functions:
  - *compute_opt_control_set:* This function will compute and sort the optimal control set according to their energy.
    - Input:
      - A (NxN, numpy array): The adjacency matrix of structural connectivity.
      - x_0 (Nx1, numpy array): The initial state.
      - x_f (Nx1, numpy array): The target state.
      - rho (float): A mixing parameter that balances the importance of different terms in the control problem.
      - S (NxN, numpy array): A constraint matrix that specifies which nodes' states are constrained.
      - system (str, optional): Specifies whether the system is continuous or discrete.
      - c (float, optional): A scaling factor for normalising the adjacency matrix.
      - T (float, optional): The time horizon over which control is applied.
      - control_size (int, optional): The number of nodes in the control set.
      - numerical_threshold (float, optional): Threshold for numerical errors; control sets with errors above this threshold are discarded. Defaults to 1e-8.
    - Output: A list of dictionaries with 'combination (simulation points)', 'energy', and 'error'
      e.g. ['combination': comb, 'energy': energy, 'error': [error1, error2], ...], where error1 refer to inversion error, error2 refer to reconstruction error.

  这个 function 最主要的 component 来自于 nctpy 提供的一个 function *get_control_inputs,* which will give us the state trajectory x(t), control signals u(t) and numerical error (inversion and reconstruction error).

get_control_inputs 这个 function 主要是在 minimise 一个 cost function, mathematically,

$$\min_{\mathbf{u}} \int_0^T (\mathbf{x}_T - \mathbf{x}(t))' \, \mathbf{S} \, (\mathbf{x}_T - \mathbf{x}(t)) + \rho \mathbf{u}(t)' \mathbf{u} \, dt,$$
$$\text{s.t. } \dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \text{ and } \mathbf{x}(T) = \mathbf{x}_f$$

where $(\mathbf{x}_T - \mathbf{x}(t))' \, \mathbf{S} \, (\mathbf{x}_T - \mathbf{x}(t))$ is the distance between the state at time t and the final state x_T, so in this optimisation problem, it will constrain the state trajectories of a subset of nodes (specified by S) by preventing the system from travelling too far from the target state. $\rho \mathbf{u}(t)' \mathbf{u}$ constrains the amount of input used (external stimuli) to reach the target state.

- ComputeOptimizedControlEnergy: 以 data driven 的 manner 通过 perturb B 的 diagonal values(每一个 stimulation node 的 weight) 去计算一个完成 state transition 的 possible minimum energy. (第一个 function 里的 B 是 binary 的, 这个 function 里的 B 是 non-binary 的,B 的 value 可以理解为对应的 stimulation point 对完成给定 state transition 过程中 dynamics 影响力的大小). 这个 function 主要参考了 nctpy 里一个类似的 class, 它主要是在每一个 step 给 B.diagonal 一个 perturbed, 计算 energy 相对于 B.diagonal value 的 graident, 用类似 graident descent method 去不断的 update B.diagonal. 但 nctpy 提供的 class 需要提前 specify gradient steps, 我这里对它改进了下, 让它一直进行 graident descent 的 step 知道 energy 不再下降 (两个 consectuive steps 计算出来的 energy difference 小于 specified threshold)

- **Results:** 我这里主要测试了 3 组数据, high pain to low pain, initial state to resting state and initial state to initial state perturbed. 这里的 Initial state perturbed 是在 initial state 的基础上加入一点 noise, noise value is drawn from a normal distribution with $\mu = 0$, $\sigma = 0.1 \times mean(initial\ state)$, 并一讲所有结果都交给了阳洋学姐.
在对一个刺激节点的情况进行测试时, 遇到一个很重要的问题, 那就是一个刺激节点无法 properly 完成 state transition(i.e. inversion error 和 reconstruction error 都很大), 解决办法参考了这边文章 [Sti+18], 即对 non-stimulation point 对应的 B 的值 (原先是 0) assign a small random value, drawn from a normal distribution with $\mu = 0.0005$, $\sigma = 0.00005$, 并且 linear model 的参数设置为 rho = 0.2, T = 0.7, c = 4, 这种情况下我们计算的 error 最小. (所以的 implementtaion 和 results 都已经 clearly annotated, 并交给阳洋学姐了)

- **Next:** Obviously, 整个 linear model 主要是基于 brain network 的 structural connectivity, 所以 structural connecivity 的一些性质的可能对于 brain dynamics 有着很好的预测相应, 如已经被该领域论文里频繁提及的 average controllability which identifies brain areas that can steer the system into many easily reached states, and Modal controllability identifies

brain areas that can steer the system into difficult-to-reach states. 也许接下来可以结合大脑生物和化学层面的研究去探讨那些可能与大脑结构连接有相关性关系的生物和化学性质, 影响了大脑结构连接, 自然而然也会影响整个 brain dynamics.
目前的 linear model 还是有点"呆板"，即大脑每一个 time step 的 state $x(t)$ 都是由前一个 time step $x(t-1)$ 决定的。对于 group-level 数据可能有不错的模拟效果，但具体到 individual level 时就不一定了 (因为这种 $x(t)$ 和 $x(t-1)$ 的决定性使得模型在个体层面可能无法捕捉到大脑活动的随机性和复杂性). 我个人觉得，具体到 individual level，想要实现更好、更 robust 的模拟效果，可能在 model 里加入些 noise 会更好。一个 potential choice 是 stochastic linear model, where $x(t)$ 不再是一个 fixed value, 而是一个 random variable drawn from a normal distribution, but the mean and variance are still related to their linear model counterparts. For more details, see [Kam+22].

Bibliography

[Kam+22] Shunsuke Kamiya et al. "Optimal Control Costs of Brain State Transitions in Linear Stochastic Systems". In: *The Journal of Neuroscience* 43 (Nov. 2022), JN–RM. DOI: 10.1523/JNEUROSCI.1053-22.2022.

[LS20] Andrea Luppi and Emmanuel Stamatakis. "Combining network topology and information theory to construct representative brain networks". In: *Network Neuroscience* 5 (Sept. 2020), pp. 1–29. DOI: 10.1162/netn_a_00170.

[Sti+18] Jennifer Stiso et al. *White Matter Network Architecture Guides Direct Electrical Stimulation Through Optimal State Transitions*. 2018. arXiv: 1805.01260 [q-bio.NC]. URL: https://arxiv.org/abs/1805.01260.