Generating brain-like networks with distance-dependent connection probability

Introduction to Computational Neuroscience I, Charles University

Aitor Morales-Gregorio (aitor.morales-gregorio@matfyz.cuni.cz)

Introduction

Neuronal networks have a nonrandom structure, with small-world properties (high clustering, short average path length), and heavy-tailed weight and degree distributions. These properties are believed to endow brain networks with their remarkable information processing capabilities. However, how do such networks emerge in nature? What are the fundamental principles that lead to networks with such properties in the first place? Are they common across species?

One fundamental constraint is the distance between neurons. It is known that neurons with a shorter distance between them tend to have far more connections than far away neurons. Indeed, connection probability is known to follow an exponential distance rule (Ercsey-Ravasz et al. 2013).

Kaiser and Hilgetag's network generative algorithm

Two computational neuroscientists proposed a network generative algorithm based on distance dependence in 2004 (Kaiser and Hilgetag 2004). Their model starts with a single neuron in a 2D sheet with coordinates in the interval $[0,1]$. At each iteration a new neuron is placed within the sheet, and it is connected to the existing neurons following the probability given by:

$$P(u,v)=\beta \mathrm{e}^{-\alpha d(u,v)}$$

where $p(u,v)$ is the Euclidean distance between neurons u and v, α and β are scaling parameters controlling the spread of the distance-dependent probability.

If the new neuron did not establish any connection to the network it was removed, to avoid fragmentation of the network. The process of adding new neurons continued until a target number of neurons N was reached.

Note that, the algorithm is designed to create <u>unweighted and undirected networks</u>! The model produces networks are not best suited to be compared with weighted and directed brain-like networks.

Improved algorithm

Several aspects of Kaiser and Hilgetags' algorithm are not particularly biologically plausible, especially adding nodes one after the other. Petra Vértes and colleagues (Vértes et al. 2012) proposed some fundamental improvements, which we will study here.

First, the number of neurons N is fixed from the start, with each neuron being randomly positioned in a 2D sheet with coordinates in the interval $[0,1]$. The positions never change and are used to estimate the distances between neurons, as before.

At each iteration a new edge is added to the network based on the probability given by:

$$P(u,v) \propto \mathrm{e}^{-\alpha d(u,v)}$$

where the probability is calculated as a matrix and normalized such that $\Sigma P=1$. The iterative process stops when a target network density ρ is reached, for this project we will always use $\rho=0.1$.

Note that, multiple edges can be added between the same two neurons, leading to <u>weighted and directed networks</u>.

Tasks (useful programming skills: Python)

1. Reproduce the algorithm from Kaiser and Hilgetag:

   1.1. Write a function that takes in the parameters α, β and N, and returns the adjacency matrix of the network. *TIP: The algorithm can be fully written using only NumPy.*

   1.2. Plot the adjacency matrix and the network itself (with NetworkX) for $\alpha=1$, $\alpha=5$, and $\alpha=10$, with $\beta=0.5, N=100$.

   1.3. Perform a parameter scan with fixed parameters $\beta=1, N=100$, but changing $\alpha \in [0.1,100]$. Calculate the Clustering coefficient C, and Average Shortest Path lengths L, for all parameter values of α, and make a plot showing how they depend on α. *HINT: Your aim is to reproduce Figure 3a from Kaiser and Hilgetag's paper.*

   1.4. Based on the results from 1.3, which network parameters produce a network with small-world properties? Why?

2. Reproduce the Improved algorithm:

   2.1. Write a function that takes in the parameters α and N, and returns the adjacency matrix of the network. *Once again only NumPy is necessary.*

   - Optional: How fast is your algorithm? Measure speed as you increase N from 100 to 1000. Can you make the code faster?

   2.2. Plot the adjacency matrix and the network itself (with NetworkX) for $\alpha=5$, $\alpha=10$, and $\alpha=20$. With $N=100$ for all cases.

   2.3. Run a parameter scan for $\alpha \in [0.1,100]$ and plot the clustering and shortest path length as in 1.3. *Warning: This might be much slower! Reduce the number of points or improve the algorithm speed to gain better performance (Optional).*

2.4. Visualize the degree distribution and weight distribution for a network with $\alpha = 15, N = 100$. What do you observe? *Hint: use a log-scale for the weight distribution.*

3. Compare artificial networks with the connectivity in the mouse visual cortex (MiCroNS Consortium 2025).

- We provide a subset of 100 neurons as a numpy array: `mouse_V1_adjacency_matrix.npy`

3.1. Visualize the mouse network.

3.2. Measure clustering, path length, weight and degree distributions in the mouse, compare to your parameter scans (from both algorithms). Which one is better? Why? *Hint: Adjust the target density of your models to better match the experimental data.*

3.3. [Optional] Can you find the best parameters to match the mouse connectivity?

3.4. [Hard, Optional] Improve the algorithm further, by including the degree-preferential attachment as described in Vertes et al 2012. Can you now match the mouse connectivity? What else would you add?

4. Finalize your report with discussion of the results, conclusions, and speculate about future directions.

Optional: Feel free to perform further analysis on the networks, quantify modularity, centrality, rich clubs… You could discover something totally new!

References

Ercsey-Ravasz et al. (2013). "A Predictive Network Model of Cerebral Cortical Connectivity Based on a Distance Rule". Neuron 80(1):184–197.

Kaiser and Hilgetag (2004). "Spatial growth of real-world networks". Physical Review E 69(3)

Vértes et al. (2012). "Simple models of human brain functional networks". Proceedings of the National Academy of Sciences 109(15):5868–5873.

The MICrONS Consortium (2025). "Functional connectomics spanning multiple areas of mouse visual cortex". Nature 640:435-447. See also: www.microns-explorer.org