IEEE SciVis Contest 2023 - Dataset of Neuronal Network Simulations of the Human Brain

Tim Gerrits Fabian Czappa Divya Banesh Felix Wolf

Introduction

The IEEE SciVis Contest is held annually as part of the IEEE VIS Conference. It challenges participants within the visualization community to create innovative and effective state-of-the-art visualizations to analyze and understand complex scientific data. In 2023, the data described *neuronal network simulations of plasticity changes in the human brain* provided by a collaboration of the Cross-Sectional Group Parallelism and Performance, and the Cross-Sectional Group Visualization within the National High Performance Computing Center for Computational Engineering Science (NHR4CES). The results of the contest are available online and the data can be accessed here: `https://doi.org/10.5281/zenodo.10519411`.

This document describes both the contest background, rules, and tasks, as found on the contest website, as well as the data itself.

Contest Description

Background

The 2023 contest dealt with the simulation of activity within the human brain, especially the changes within its network structure. Current understanding suggests, that a large number of connections between neurons via synapses is constantly changing, leading to alteration of brain functionality. This includes positive effects such as the creation of memories [GGR22] or negative ones such as limitations caused by accidents or illnesses [BvO13]. To understand these effects on a neuronal level, scientists hope to artificially recreate networks that work similarly to the human brain with the help of simulation models. The data provided in this contest is based on clinical measurements of actual human brains as well as the Model of Structural Plasticity by Butz & Van Ooyen [BvO13] which can be coarsely summarized as follows:

The brain is modeled by a network of neurons. Each neuron is represented by a node with a number of properties such as location, calcium concentration, and target calcium concentration. It can further carry axonal ("plugs") and dendritic ("sockets") elements. If two neurons are spatially close and exhibit unused complementary elements, they have a probability to form a synapse, which is represented by a connection in the network through which electric energy can be passed from one neuron to another if it "fires" based on external input and the Izhikevich model [Izh03]. When a neuron fires, its calcium level increases only to decrease again over time. In this model, neurons aim to reach a certain target calcium concentration based on their current calcium level. To achieve this, it either seeks to increase or decrease the level of incoming electrical activity resulting in the addition or removal of elements, which, in turn, leads to the formation, rewiring, or deletion of synapses. Thus, in each simulation step, each neuron calculates its energy income and calcium level based on this model, followed by the addition or deletion of elements, leading to changes in the network structure. This can be repeated until the system has reached a stable configuration or stopped on demand.

Visualization can help to understand these processes (see, e.g., [NDPW+18]). Therefore, the goal of this contest is to enable domain scientists as well as simulation scientists to better evaluate their models,

compare, and explore the simulation results. Based on this, we have provided a list of visualization tasks for this contest.

Tasks

The contest was set out to look for powerful, state-of-the-art approaches that aim towards helping domain scientists to understand the changing structures within the human brain. This includes determining a simulation's state and questions about the similarities and dissimilarities of simulation outcomes. Some questions that researchers hope to answer are: Which parts of the brain are more active in firing and restructuring than others? Can it be determined when the neurons reached a steady state using the calcium concentration and the fluctuation in grown elements? Is that the case for all neurons or simulations simultaneously? How are still-alive neurons affected by the loss of neighboring neurons based on lesions? Does "learning" improve the connections of groups of neurons?

We give a possible list of tasks a submission could follow. This was, however, only a suggestion and could be altered and extended by the participants:

1. **Overview of data:** Create a basic framework that allows showing, exploring, and comparing the different datasets to gain a first insight into the data.
2. **Visualization of plasticity changes:** Create the functionality that allows analyzing when and where creation, rewiring, and deletion events happen. How many regions/neurons are part of such events? How is this related to calcium levels? How can it be shown, that a simulation has reached a steady state? How far is the impact of events such as learning or lesions?
3. **Ensemble visualization:** Find ways to indicate differences and similarities between the simulations. Where do the different simulations show similar or dissimilar behavior also compared to Tasks 1 and 2?
4. **Workflow:** Try to combine the functionality of the other tasks into a combined framework and interface that enables experts to interactively explore the data and investigate the questions mentioned above. Which tools help the users to mark, select, and analyze the data on different levels of complexity and transfer information among these levels?

Further information

Questions and remarks about the data could be send to the contest organizers via email or by using a dedicated google group.

Data Description

This data contains:

- 49.8 GB data
- 4 simulation ensemble members
- Each members consists of 50,000 neurons
- Each node collects information about 12 parameters
- The simulation is run for 1,000,000 simulations steps
- Every 100^{th} iteration step is sampled per node

Please find more details in the *readme.md*.

In-Detail Description

Four simulations have been run based on neuron locations given by the same underlying human brain data (MEG 146129) provided within the human connectome database and extended by additional neurons close to existing ones. In the future, we plan on using more accurate models but these are not yet open for free use. Technical details on the simulation process can be found in [CGW23] and [RBOH+18]. Each simulation produces several text files sorted in their corresponding folder as shown in Table 1.

Simulation	Conditions	Folder Name
1	0-calcium level at the start and equal target calcium level for all neurons, no existing connections	no-network
2	Lesion Simulation: Take the connection result of 1 and simulate with 10% neurons inactive (dead)	disable
3	Learning Simulation: Take the connection result of 1 and simulate learning (electric stimulation) of a specific area	stimulus
4	Take the connection result of 1 and use different target calcium levels for neurons	calcium

Table 1: Provided folder names and corresponding simulaiton descriptions.

Some of the simulations have additional files, e.g., `calcium_target.txt`, `disable.txt`, and `stimulus.txt` that include simulation-specific information.

As the simulation can be efficiently run on a cluster of computing nodes, MPI information is included within the files and names by a leading MPI Rank ID.

Each folder has the same setup explained in the following:

Network Layout

The base for the simulation is a neuronal network represented by point neurons. Each neuron has a Neuron ID that is unique to its MPI Rank ID. The combination of both is then used throughout the simulation and data.

The 3D-position of each neuron can be found in `positions/rank_0_positions.txt`. These are fixed and do not change. This file further contains labels explaining which part of the brain the neuron belongs to. Note: These areas are set manually and are not based on medical accurate regions within the human brain.

Ingoing as well as outgoing connections between nodes are sampled at different simulation steps and written in `network/<MPI_RANK_ID>_step_<STEP>_in_network.txt` and `network/<MPI_RANK_ID>_step_<STEP>_out_network.txt` respectively.

Per-Node Information

The information about individual nodes is saved to `*.csv` files within `monitors.zip`. Their naming convention is as follows: `<MPI_RANK_ID>_<NEURON_ID-1>.csv`. Each row within the file contains the per-node information at a sampled simulation step.

Note: In the original dataset, there was an error in the step parameter due to a bug in the writing function. The step number was reset to 0 every 30,000 iterations as discussed here. This has been fixed for the final publication.

Variable	Description
Step	Sampled simulation step
Fired	Boolean: Did the neuron fire within the last sample step
Fired Fraction	In Percent: Number of firings since the last sampling
x	Electric Activity
Secondary Variable	Inhibition variable used for the firing model of Izhikevich [Izh03]
Calcium	Current calcium level
Target Calcium	Target calcium level
Synaptic Input	Input electrical activity
Background Activity	Background noise electric activity input
Grown Axons	Number of currently grown axonal boutons
Connected Axons	Number of current outgoing connections
Grown Excitatory Dendrites	Number of currently grown dendrite spines for excitatory connections
Connected Excitatory Dendrites	Number of incoming excitatory connections

Table 2: Variables stored per node within the monitor files.

Table 2 shows all listed parameters.

Simulation Information

Each simulation run also saves some statistics as listed in Table 3.

File Name	Description
`neurons_overview.txt`	Statistical summary sampled simulation steps
`plasticity_changes.txt`	Summary of creation and deletions of connections
`stdcout.txt`	Program output
`timers.txt`	Timings

Table 3: Files storing information on statistics.

Matlab example for visualizing nodes and connections

This is a small hacked example of how to read node positions, incoming connections, and calcium value and plot the network in Matlab.

The code example should produce the outcome shown in Figure 1.

```
% Assuming the correct file locations have been set
%%% First we read in nodes and their locations
fileID = fopen(positionfilelocation,'r');

% Read number of nodes:
numOfNodes = fscanf(fileID,'# %d');
fgetl(fileID);

% Read min max values:
minMaxXYZ = fscanf(fileID,'%*s %*s %*s %f',[3,2]);
fgetl(fileID);
fgetl(fileID);

% Read positions, labels and type
data = textscan(fileID,'%*f %f %f %f %s');
fclose(fileID);

% Data containters
positions = [data{1} data{2} data{3}];
labels = [data{4}];
```

```
21
22 %%% Now read the incoming connections:
23 fileID = fopen(connectionInfilelocation,'r');
24 fgetl(fileID);fgetl(fileID);fgetl(fileID);fgetl(fileID);fgetl(fileID);
25 inconnect = textscan(fileID,'%f %f %f %f %f');
26 fclose(fileID);
27
28 %%% Read Calcium values:
29 fileID = fopen(calciumfilelocation,'r');
30 str = fgetl(fileID);
31 iterations = 0;
32 calcium = zeros(1,numOfNodes);
33
34 i = 1;
35 while(str ~= -1)
36     linedata = strsplit(str,';');
37     iterations(i) = sscanf(linedata{1},'#%d');
38     calcium(i,:) = str2double(linedata(2:end));
39     i = i+1;
40     str = fgetl(fileID);
41 end
42 fclose(fileID);
43
44 %%% Plot Nodes and incoming connections
45 f = figure
46 hold on
47 axis vis3d
48 % optional: set(f,'renderer','opengl');
49 view(60,8)
50
51 % plot ingoing connections:
52 xx = [];
53 for i=1:size(inconnect{1},1)
54     startnodeid = inconnect{2}(i);
55     endnodeid = inconnect{4}(i);
56     line = [positions(startnodeid,:);positions(endnodeid,:)];
57
58     xx(((i-1)*3)+1:((i-1)*3)+3,:) = [[line(:,1);NaN],[line(:,2);NaN],[line(:,3)
        ;NaN]];
59 end
60 h = plot3(xx(:,1),xx(:,2),xx(:,3),'-k');
61 h.Color(4)= 0.01;
62
63 % plot neuron locations
64 plot3(positions(:,1),positions(:,2),positions(:,3),'.','Color', "#80B3FF")
```

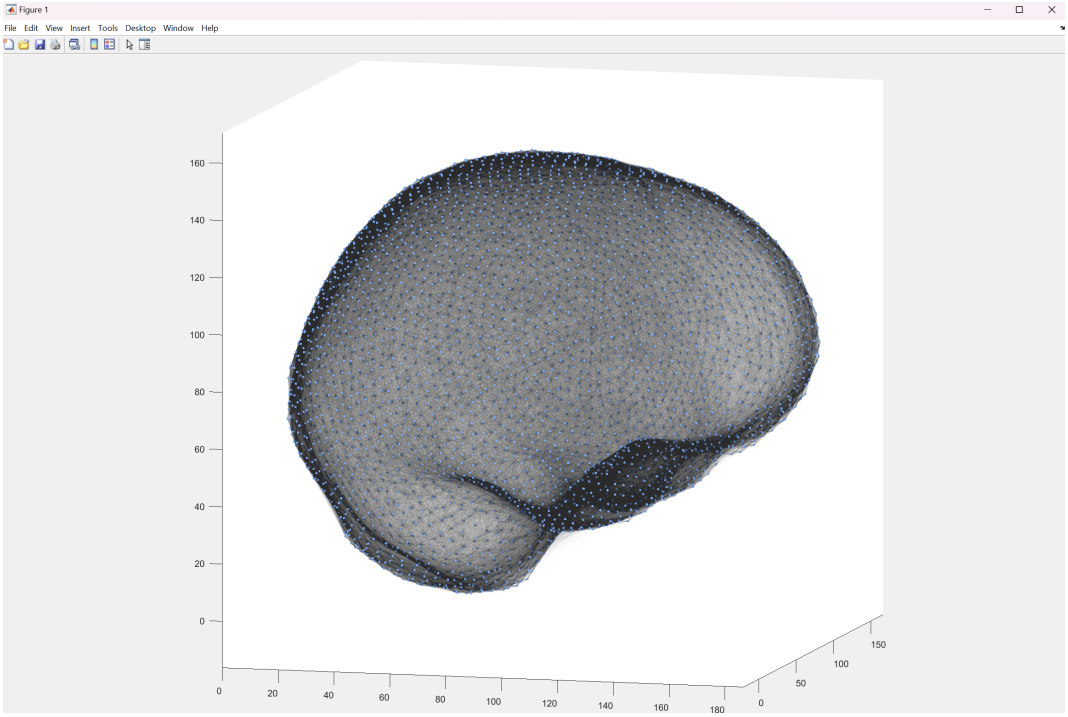

Figure 1: Simple 3D plot in Matlab of node positions and connections

Be warned: Due to the number of nodes, plotting might be very slow on some machines.

References

[BvO13] Markus Butz and Arjen van Ooyen. A simple rule for dendritic spine and axonal bouton formation can account for cortical reorganization after focal retinal lesions. *PLoS computational biology*, 9(10):e1003259, 2013.

[CGW23] Fabian Czappa, Alexander Geiß, and Felix Wolf. Simulating structural plasticity of the brain more scalable than expected. *Journal of Parallel and Distributed Computing*, 171:24–27, 2023.

[GGR22] Júlia V Gallinaro, Nebojša Gašparović, and Stefan Rotter. Homeostatic control of synaptic rewiring in recurrent networks induces the formation of stable memory engrams. *PLOS Computational Biology*, 18(2):e1009836, 2022.

[Izh03] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[NDPW+18] Christian Nowke, Sandra Diaz-Pier, Benjamin Weyers, Bernd Hentschel, Abigail Morrison, Torsten W Kuhlen, and Alexander Peyser. Toward rigorous parameterization of underconstrained neural network models through interactive visualization and steering of connectivity generation. *Frontiers in neuroinformatics*, 12:32, 2018.

[RBOH+18] Sebastian Rinke, Markus Butz-Ostendorf, Marc-André Hermanns, Mikaël Naveau, and Felix Wolf. A scalable algorithm for simulating the structural plasticity of the brain. *Journal of Parallel and Distributed Computing*, 120:251–266, 2018.