Benchmarking BioDynaMo

AUGUST 2021

AUTHOR(S):
Damla Gözük

SUPERVISOR(S):
Ahmad Hesam
Fons Rademakers

PROJECT SPECIFICATION

The idea of the BioDynaMo project is to create a general-purpose and high-performance simulation engine that can assist scientists to design an agent-based modeling tool for complex biological systems.

BioDynaMo is an open-source project that tackles problems in In Silico Simulations used in scientific research such as the usage of the hardware which results in compromising either on the resolution of the model or on simulation size and for specific use purposes for each case.

This project focuses on performance analysis with the help of tools such as Grafana and enhancing the benchmarking environment of the project. The idea is to make a set of benchmark simulations that allow us to:

1) See that BDM works after installation.
2) Show detailed performance of each test.
3) Track slowdowns, speedups, memory increase, memory decrease between versions.
4) Assign performance numbers to a user's machine in the form of BDMMARKS.

Some expected tasks are:

- Read BioDynaMo paper to understand what BioDynaMo is.
- Build BioDynaMo according to the instructions.
- To get to know Grafana.
- Familiarize with Google Benchmark.

ABSTRACT

The purpose of this project was to test and analyse the BioDynaMo project's performance. In order to achieve this, BioDynaMo uses its own benchmark tools. Benchmarking is highly important because it allows us to test the code performance and monitor the effect of the changes. It allows us to observe the project's speed, memory usage, and hardware efficiency.

This report consists of the *Introduction* section which explains the BioDynaMo project's purpose, where it is used with demonstrating some example cases, and some technical information; *Objectives* section which indicates the responsibilities; *Implementation Methods* section which shows the technologies used and how to approach the tasks; *Conclusion* section which summarizes the works done and challenges achieved; and finally *Future Improvements* section that identifies and demonstrates tasks needed for the future work.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. What is BioDynaMo

BioDynaMo is an open-source, high-performance, and modular agent-based simulation platform. Currently, most of the biological demos are made with In Silico Simulations. After slowing down Moore's Law and ending Dennard's Scaling, In Silico Simulations lost its effectiveness in terms of usage of the hardware which results in compromising either on the resolution of the model or on simulation size. Additionally, these simulations are made for specific use purposes for each case. To change the main software from one to another is very challenging and creates a buggy implementation design. Hence, BioDynaMo is shaped to tackle those problems in scientific research.

BioDynaMo has 5 system properties:

- It is agent-based. Agents are modelled as discrete objects and perform actions based on their current state, behaviour, and environment.
- It is a general-purpose platform. Instead of specializing in a single case, it can simulate various cases in different fields and does this in both modular and extensible ways.
- It is large-scale model support. Since biological systems consist of many agents, for example, the cerebral cortex has approximately 16 billion neurons, BioDynaMo is designed to overcome the limitations of scaling up the simulations with its modern hardware usage and memory efficiency.
- It is easily programmable. It provides common functionalities, and it has a modular and extensible design.
- Its quality is assured. BioDynaMo has over 280 automated tests which are continuously executed on all supported operating systems to ensure high code quality.

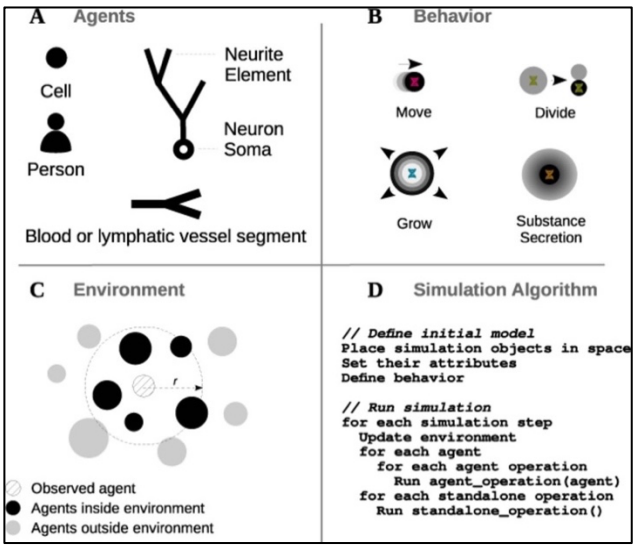

Figure 1: *BioDynaMo core simulation concepts. (A) Geometry, (B) Behavior, (C) Environment, (D) Simulation algorithm.*

1.2. Example Use Case

BioDynaMo can be used to model neurite growth of pyramidal cells using chemical cues. With the knowledge of measurement of pyramidal cells and dendrites in 3D space and some mathematical equations along with Gaussian distribution, we are able to follow the growth of pyramidal cells and simulate them.

The figure below demonstrates the process and the modelling of neurite growth of pyramidal cells.

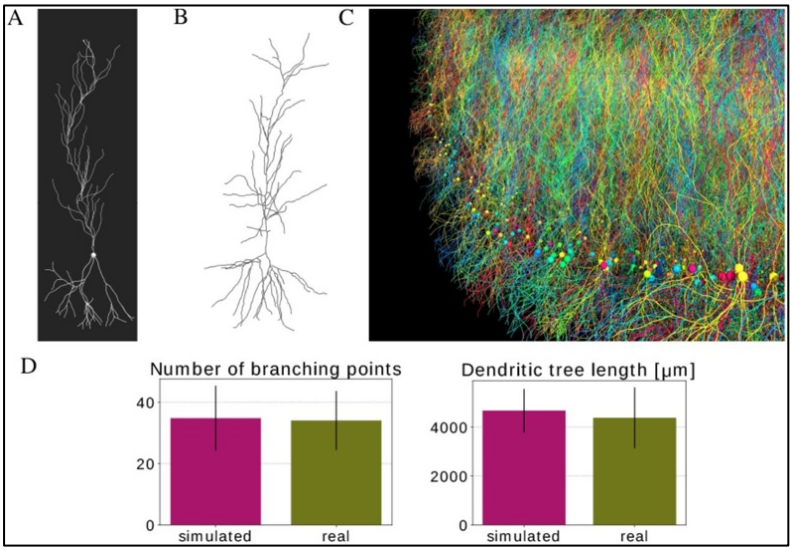

Figure 2: *Pyramid Cell Simulation. (A) Model of a straight long apical dendrite with a simple branching patter, (B) Real pyramidal cell morphology, (C) Large-scale simulation incorporating 5000 neurons, (D) Data between the comparison of the real version and the simulated version.*

1.3. Technical Background

BioDynaMo is written in C++. It also utilizes Python backends. In addition, BioDynaMo is available as easy to run Jupyter Notebooks. BioDynaMo uses the ROOT system and ROOT library which is a framework created at CERN and used for data analysis, perform simulations as well as restoring to the memory in case of failures and backing up the BioDynaMo files. The engine uses OpenMP to parallelize and accelerate the response time.

2. OBJECTIVES

My work in BioDynaMo mainly consists of enhancing the benchmarking environment of the project. This is highly important because it allows us to test the code performance and monitor the effect of the changes. It allows us to observe the project's speed, memory usage, and hardware efficiency,

For our responsibilities and workflow, we used Trello, a web-based, Kanban-style, list-making application, and made constant team meetings. Some objectives are given and tasks that I have worked on are indicated below as a Trello card.

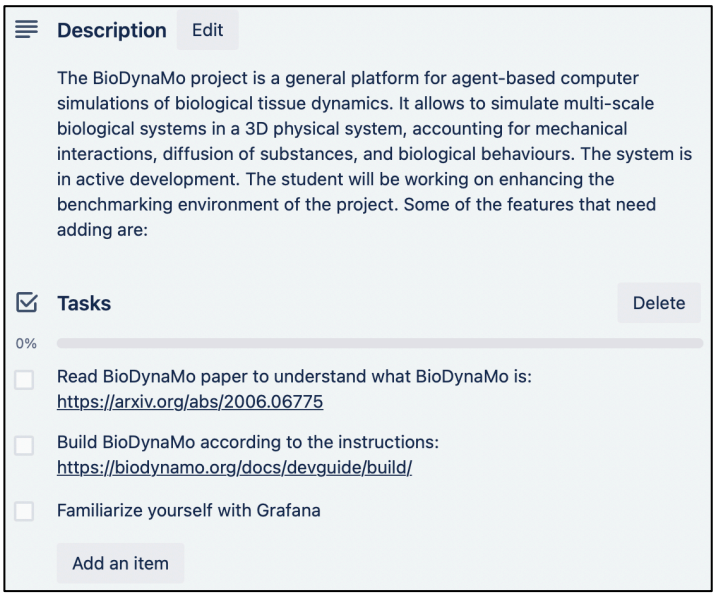

Figure 3: *Trello Workflow.*

3. IMPLEMENTATION

3.1. Technologies Used

BioDynaMo's benchmarking is similar to Google Benchmark. It is written in C++. BioDynaMo Team was inspired by the ROOTBench. The ROOT system is created at CERN and is used for data analysis. The ROOT team has its own benchmarking system called ROOTBench. It is categorized into 3 categories, performance monitoring primitives, visualization features, and data storage. Google Benchmark library has a significant role that provides multithreading and custom report generation. It has an extensive C++ library. BioDynaMo data is visualized with Grafana. It allows us to collect data from numerous sources and share data between all the team members. Grafana is a general-purpose dashboard and graph composer. It provides

many ways to visualize time series metrics and it supports database backends such as Influx DB.

For the development process, workflows, responsibilities, and tasks, the BioDynaMo team uses Trello. All the Information and to-dos are listed on Trello boards. Internal communication such as codes, images, files, etc. is done through Slack chat. Once in two weeks, all the team and interested scientists, professors, students gather for our scheduled meeting and discussed completed tasks, managed progress, decide future tasks, and exchange new ideas. To share and distribute the code, Git and GitHub are actively used.

3.2. Approach to the Task

In order to build BioDynaMo, after downloading necessary and required dependencies, we need to follow the directions given on the BioDynaMo website. After all the packages are downloaded, to build the project on MacBook, we need to do `cmake -GNinja..` .

```
git clone https://github.com/BioDynaMo/biodynamo.git
cd biodynamo

# Install the prerequisites
./prerequisites.sh all

# Create the build directory
mkdir build
cd build

# Build BioDynaMo
cmake ..
make

# (Optional) Installs the libraries
make install
```

Figure 4: *BioDynaMo build.*

But the version was not compatible with MacBook's new M1 chip. Hence, there were some ROOT issues to build the project and it couldn't be built.

```
-- Could NOT find ROOT (missing: ROOT_CONFIG_EXECUTABLE ROOTSYS
ROOT_VERSION ROOT_INCLUDE_DIR ROOT_LIBRARIES ROOT_LIBRARY_DIR)

FAILED: CMakeFiles/run-benchmarks and ninja: build stopped:
subcommand failed.
```

While trying to find a solution for this case, the first thought was to run gbenchmark based benchmarking environment for ROOT that integrates with Grafana. After communicating with the ROOT team, we gathered the following instructions to be able to run rootbench as continuous performance benchmarking testing:

- Need to create InfluxDB database, that has been done in CERN DBOD service: https://dbod.web.cern.ch/
- Need to have Grafana instance with connected Influx DB as a backend
- Need to have a possibility to be able to run benchmarks continuously and in an automatic way (e.g., via Jenkins, cron, etc.)
- To upload benchmark results data, generated as json files, https://github.com/root-project/rootbench/blob/master/tools/upload2influxdb.py script is used where all required parameters for DB connection are passed.

However, this did not work for our case and the error I got was `FAILED: CMakeFiles/run-benchmarks and ninja: build stopped: subcommand failed`. While trying for alternative solutions, the ROOT team helped us to solve this problem and now the project is buildable on MacBook M1 chips.

4. PERSONAL EXPERIENCE

Past 9 weeks at CERN as an Openlab Summer Student was spectacular. Even though it was online due to the COVID-19, Openlab Team, staff, organizers worked hard to assist us with our problems and made our time working at CERN as efficient as possible. BioDynaMo Team, the project I have participated in, is made of an international, interdisciplinary group of researchers who are passionate about using modern computer simulation techniques for promoting biomedicine and the health of society. Working in an international environment and working with people from different cultures enhance thinking beyond the horizons and improve perception. Furthermore, working with multiple people from various disciplines broadens the impact scope by increasing the exchange of ideas and contributes to better analyze outcomes.

In addition to our projects, we attended lectures almost daily with different topics varying from particle physics to deep learning in high energy physics that were given by notable professors and professionals. Summer lectures also had Q&A sessions where students were able to ask lecture-related topics or topics, they are curious about. Each lecture was outstanding however, for me, two topics were intriguing: Computer Security in 2021 by Stefan Lueders and Electronics, DAQ, and Triggers by Emilio Meschi.

Considering CERN Openlab 2021 was held online, we were not able to have physical meetings, activities, visits, etc. CERN Summer Student organizers appointed affiliated people for virtual visits where they would go on-site and show us how the device works as well as clarifying its reason and how it operates. That being said, we visited AD, LEIR, ALICE, LHCb, ATLAS, and NA61/SHINE points online. Both CERN openlab Team and Summer Student Team organized an online meeting where all of us, students would come together and play online games, talk about our projects at CERN or talk about trending topics at the time. Thus, I met wonderful people and made long-lasting friendships along with learning new ideas and new trends.

On the weekend of 21-22 August, CERN organized its annual hackathon, Webfest. This year's theme was science, society, sustainability. CERN Openlab Summer Students were highly encouraged to attend these Webfests. I formed a team submitted an idea which I thought would be a great idea for a better world related to wildfires across the globe called Wildfire Emergency Response: Prevent-Protect-Enforce (https://webfest.cern/node/341). The project mainly proposes solutions to Wildfire problems the world is facing a lot recently. It corresponds to decreasing the spread of the fires, lower the response time, shorten the building and infrastructure damages and most importantly, prevent animal and human deaths as much as possible. All weekend we worked hard and created a prototype app that achieved our objectives. On 23 August, our project was shortlisted among 29 other projects.

Towards the end of our time at CERN as an Openlab Summer Student, I had a chance to present my project, Biodynamo, and my work to other fellow Summer Students and colleagues. I gain more confidence as I was presenting, and I gain more knowledge about the details of the project while I was researching. Besides, I enhanced my know-how by listening to other summer students' projects.

5. CONCLUSION

Due to changes in the new MacBook's Arm-based M1 chip, we had a ROOT problem deploying the project and developing it. Our focus turned into the accessibility of the project codes for 2020 or above MacBook users. Eventually, we were able to overcome this issue and the project is now deployable and applicable.

Tracking and improvements in slowdowns/speedups/memory increase/decrease between versions.

Assignation of performance numbers to a user's machine in the form of BDMMARKS.

6. FUTURE IMPROVEMENTS

Following recommendations can be considered to improve BioDynaMo:

New techniques and algorithms would be developed to decrease CPU and GPU usage and to boost efficiency.

Routine and scheduled checks are needed for tracking down the slowdowns/speedups/memory increase/decrease between versions.

7. REFERENCES

1. *BioDynaMo: a general platform for scalable agent-based simulation Lukas Breitwieser, Ahmad Hesam, Jean de Montigny, Vasileios Vavourakis, Alexandros Iosif, Jack Jennings, Marcus Kaiser, Marco Manca, Alberto Di Meglio, Zaid Al-Ars, Fons Rademakers, Onur Mutlu, Roman Bauer bioRxiv 2020.06.08.139949;*

   doi: https://doi.org/10.1101/2020.06.08.139949

   Supplementary Information: https://zenodo.org/record/4501515

2. *BioDynaMo, https://biodynamo.org/*
3. *Root, https://root.cern/*
4. *Continuous Performance Benchmarking Framework for root Shadura, O., Vassilev, V., & Bockelman, B. P. EPJ Web of Conferences 2019. 214, 05003;*

   doi; https://doi.org/10.1051/epjconf/201921405003

5. *Grafana, https://grafana.com/*
6. *CMake, https://cmake.org*
7. *CERN Webfest, https://webfest.cern/event/webfest-2021*