BioDynamo Lab

September 2021, Geneva

AUTHOR(S):
Muhammad Hassan Ahmed

SUPERVISOR(S):
Ahmad Hesam

CERN Openlab Summer Student Report 2021

CERN openlab

PROJECT SPECIFICATION

The BioDynaMo project is a general platform for agent-based computer simulations of biological tissue dynamics. It can simulate multiscale biological systems in a 3D physical system, accounting for mechanical interactions, diffusion of substances, and biological behaviors. The system is in active development. The student will be working on enhancing the BioDynaMo Jupyter notebook dashboard environment. The project consists of tasks including the automation of the build process for Jupyter notebook Docker containers with the latest build of BioDynaMo. To transfer notebooks platform from classic Jupyter to JupyterLab. Add two new templates for developing and using BioDynaMo in the browser that will provide an option to run demo and tutorial notebooks.

ABSTRACT

BioDynaMo is an open-source, high-performance, and modular agent-based simulation platform. The main goal of this project is to make BioDynaMo more accessible to the public. Therefore, I created a new docker container and Github actions workflow to build and update the image on Docker Hub for each change to the BioDynaMo repository. The build process now installs all libraries and dependencies required by the BioDynaMo into a single Docker image; and then pushes them into Docker Hub. I also added two new pages to the BioDynaMo website (www.biodynamo.org). The pages give an overview of the available demos and notebooks. The displayed information is automatically extracted from the underlying code and helps to keep the codebase and website synchronized.

TABLE OF CONTENTS

1. INTRODUCTION

BioDynaMo is an open-source, high-performance, and modular agent-based simulation platform that allows users to create, run and visualize 3D agent-based simulations. It takes full advantage of modern parallel and distributed computing for agent-based modeling [1].

Simulated experiments are a crucial tool for researchers that work with complex biomedical systems to simulate the interaction between different agents. These simulations help in developing new medicines and biological understanding. However, most simulation platforms are specific to their intended purposes and cannot be extended further for new experiments. As advancements in single-threaded computing performance have slowed down over recent years, Multi-core processing and multi-threaded computer architectures have become commonplace. A need existed for a software platform to take full advantage of parallel computing and new hardware that existing solutions haven't utilized fully.

BioDynaMo started as a knowledge transfer project at CERN [2]. It tackles the problem of the absence of a standardized and high-performance platform that can handle in-silico biomedical experiments. It is general-purpose and can simulate various experiments in a modular way. Furthermore, it allows agent-based modeling of discrete objects that perform actions based on their current state, behavior, and the surrounding environment. BioDynaMo is easily programmable to help researchers transform their ideas into simulations quickly. BioDynaMo is thoroughly tested and follows a test-driven approach to its development.

This project aims to make BioDynaMo more accessible to anyone that wants to try out simulations in BioDynaMo without worrying about installing dependencies and other libraries to make it run. This report discusses the steps to make BioDynaMo more accessible by containerizing BioDynaMo and its dependencies, notebooks, and tutorials. The tutorials help in understanding how the platform works and its various use cases. Example demo and tutorials are made available to be executed in the browser using MyBinder. This report also discusses problems and difficulties faced during the process.

2. BUILD PROCESS

a. CONTAINERIZATION AND DOCKER:

The process of packaging software with just the operating system (OS) libraries and dependencies required to run the code is called Containerization [3]. It allows applications to be written once and run everywhere. Containerization creates a lightweight executable called a Container [4]. A container is independent of the host operating system. It is portable and can operate across platforms. Containerization allows for faster, reliable, and robust automated deployments.

Containers are made possible by process isolation and virtualization capabilities built into the Linux kernel. Containers are lightweight because they don't need an extra load of a hypervisor but run directly within the host machine's kernel.

Figure 1 illustrates the Docker container architecture; they share the same host kernel; they include only the OS processes and dependencies necessary to execute the code.

Docker is one of the most well-known open-source containerization platforms with integration for major cloud providers such as OpenShift, AWS, Azure, GCP, etc. Docker allows for efficient and predictable deployments [5]. Docker will help containerize BioDynaMo and deploy it on DockerHub, an online platform to host container images.

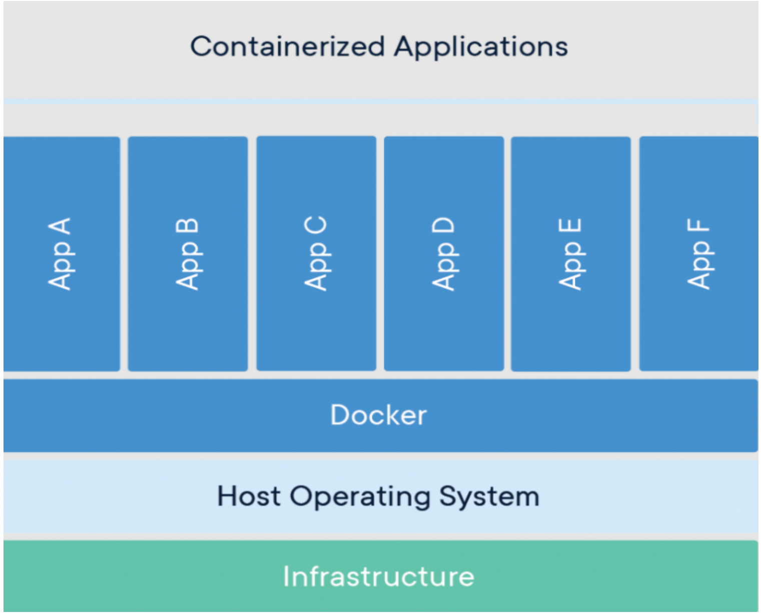


Figure 1. *Docker Container Architecture*

b. BIODYNAMO CONTAINER IMAGES

i. BASE DOCKER IMAGE

My work follows the work of another OpenLab Summer Student for the year 2020, Berina Bandić [6]. She created a Dockerfile; that downloaded a BioDynaMo build archive from the CERN website to produce a docker container image, illustrated in **Figure 2**. This process needed to be updated as it was dependent on manually uploading a build archive of BioDynaMo.

To ensure BioDynaMo container images are on par with the latest release on GitHub, I created GitHub Action scripts. The GitHub actions run whenever a change is pushed to the 'master' branch of the repository. The GitHub Action checkouts the latest code available in the main repository, installs it along with its dependencies in a container and then pushes that container to Dockerhub.

Figure 3, illustrates the described new workflow.

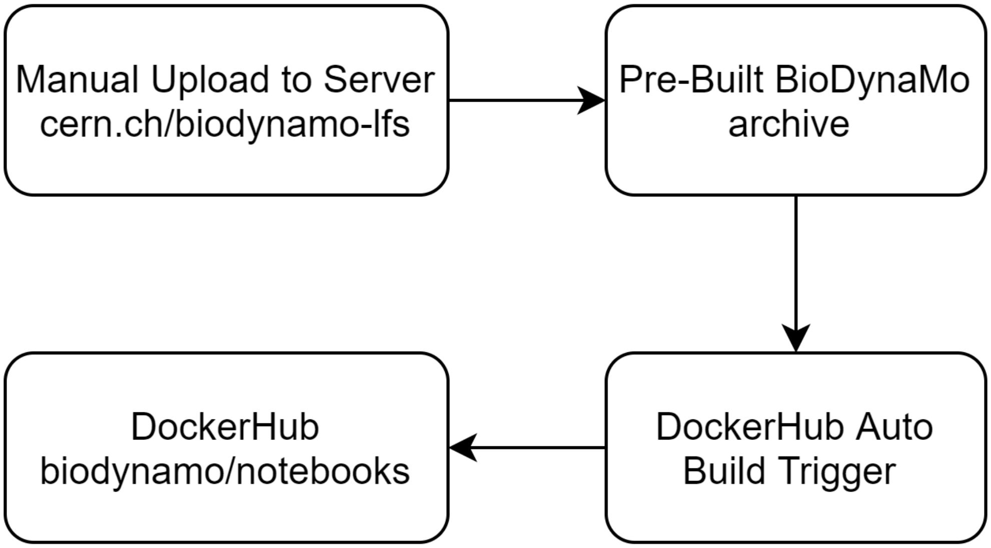


Figure 2. *Previously used BioDynaMo Base Container Build Process*

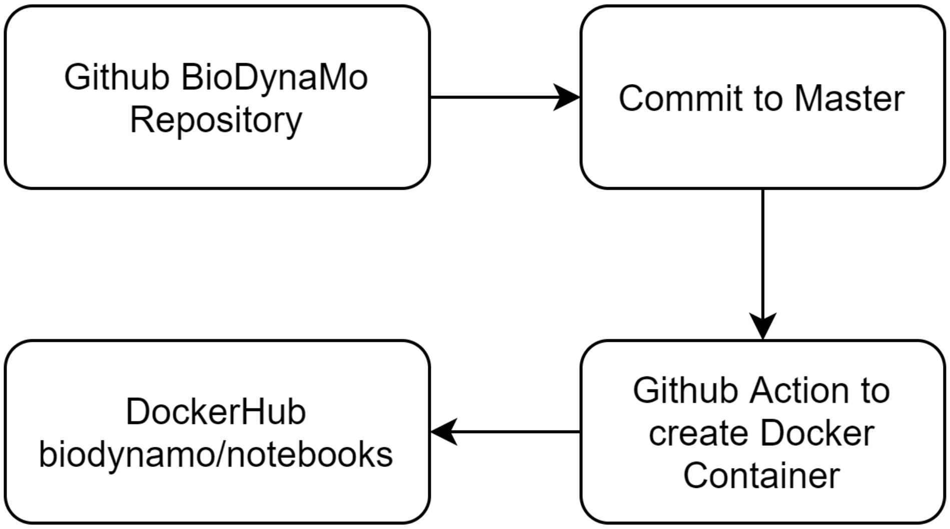

Github BioDynaMo Repository
Commit to Master
Github Action to create Docker Container
DockerHub biodynamo/notebooks

Figure 3. *New BioDynaMo Base Container Build Process*

ii. BIODYNAMO NOTEBOOKS DOCKER IMAGE

The BioDynaMo notebooks image contains all dependencies necessary to execute Jupyter notebooks utilizing BioDynaMo. This container uses the base image of the BioDynaMo container and installs additional dependencies and configurations to set up a Jupyter server.

The previous notebook container '**Dockerfile'** was hard-coded to a certain tag of the BioDynaMo base container.

The new workflow contains scripts to update the **Notebooks Docker image** with the tag for the latest BioDynaMo **Base Docker image**. The scripts are defined as **jobs** in the **.github/workflows/dockerhub.yml** file in the **binder-demo** [7] repository.

The **jobs** defined in the script are as follows:

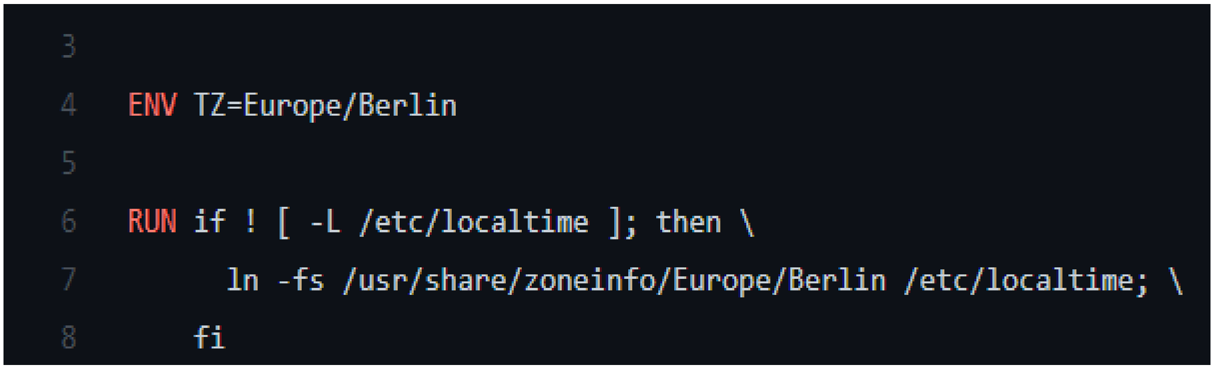
- **push_to_registry:**
  - Checkout the latest BioDynaMo repository.
  - Build a docker container from the **base image.**
  - Push container to DockerHub.
  - Extract tag of the built container and pass it to next job
- **update_notebooks_dockerfile:**
  - Update image tag of **biodynamo/notebooks** in docker file using the output of the previous job.

iii. TECHNICAL ISSUES AND SOLUTIONS

The **Base Dockerfile** is currently present in the **‘base-image’** folder, the **Notebooks Dockerfile** is present in the root directory of the binder-demo repository in BioDynaMo’s GitHub Organization.

Some of the dependencies failed during installation due to the timezone variable not being set in the environment.

The snippet in **Figure 4** sets the timezone before installing dependencies:



```

ENV TZ=Europe/Berlin

RUN if ! [ -L /etc/localtime ]; then \
      ln -fs /usr/share/zoneinfo/Europe/Berlin /etc/localtime; \
    fi
```

Figure 4. *snippet to set up timezone in Docker Container*

The Base Dockerfile performs the following operations:

- Set timezone variable
- Install prerequisites of BioDynaMo
- Create a build directory for BioDynaMo build, ‘/**build_dir’**.
- Clone BioDynaMo repository from GitHub
- Run **CMake** to Build BioDynaMo
- Source **thisbdm.sh** from the installation directory.

The **Notebooks Dockerfile** is used to run Jupyter Notebooks in “MyBinder”. To ensure reproducibility, the Dockerfile follows recommendations [8] from MyBinder to set up a Dockerfile.

The **Notebooks Dockerfile** is currently present in the root directory of the binder-demo repository in BioDynaMo’s GitHub Organization.

This Dockerfile performs the following operations:

- Use BioDynaMo base docker container as its base container. The base container has BioDynaMo built and sourced.
- Add another Linux user account to the container.
- Copy **jupyter_notebook_config.py** file to the container, it sets static paths for Jupyter.
- Copy notebooks from BioDynaMo build.
- Provide newly created user permissions to execute notebooks.
- Set “**/start.sh”** as an entry point to run the notebook server, it starts the Jupyter server when the container is executed.

3. WORKFLOW

The current BioDynaMo container process is dependent on the following platforms and tools:-

a. GITHUB ACTIONS

GitHub Actions are used to create workflows. They are event-driven, and we can automate every step of software development through them. Therefore, it automates the task by helping to set up commands after a specified event has occurred [9].

The GitHub Action is present at **.github/workflows/dockerhub.yml** of the binder-demo repository in BioDynaMo's GitHub Organization. It performs the following operations:

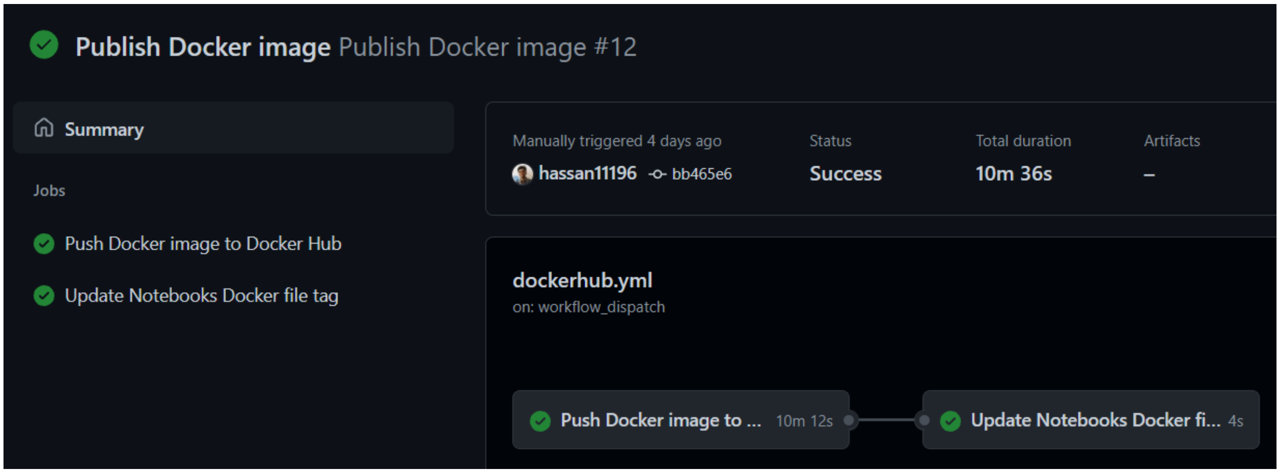
Job 1: Push Docker image to Docker Hub

- Checkout the current repository.
- Login to DockerHub using the "GitHub Secrets". The "**DOCKER_USERNAME**" and "**DOCKER_PASSWORD**" variables contain DockerHub credentials that need to be set up in the GitHub repository settings.
- Build the base docker container.
- Push the already built base docker container to DockerHub.
- SHA-Tag of the newly pushed base docker container is obtained as output.

Job 2: Update **Notebooks Dockerfile** tag

- Checkout the current repository.
- Use the output from the previous job and update the first line of Notebooks Dockerfile to reflect the tag of the latest base docker image pushed to DockerHub Registry.
- Commit change in Dockerfile to the repository.

Figure 5 is a screenshot from GitHub Actions, the line between the two jobs illustrates the dependency of Job 2 on Job 1.



Publish Docker image Publish Docker image #12

Jobs

- Push Docker image to Docker Hub
- Update Notebooks Docker file tag

Manually triggered 4 days ago	Status	Total duration	Artifacts
hassan11196 -o- bb465e6	Success	10m 36s	–

dockerhub.yml
on: workflow_dispatch

Push Docker image to ... 10m 12s
Update Notebooks Docker fi... 4s

Figure 5. *BioDynaMo GitHub Action Jobs*

b. DOCKER HUB

Docker Hub is a service provided by Docker for finding and sharing container images with your team. It is the world's largest repository of container images with an array of content sources including container community developers, open-source projects, and independent software vendors (ISV) building and distributing their code in containers [10].

The BioDynaMo base docker container is available from DockerHub.

It can be pulled by the following command:

```
$ docker pull biodynamo/notebooks
```

c. BIODYNAMO BUILD WORKFLOW

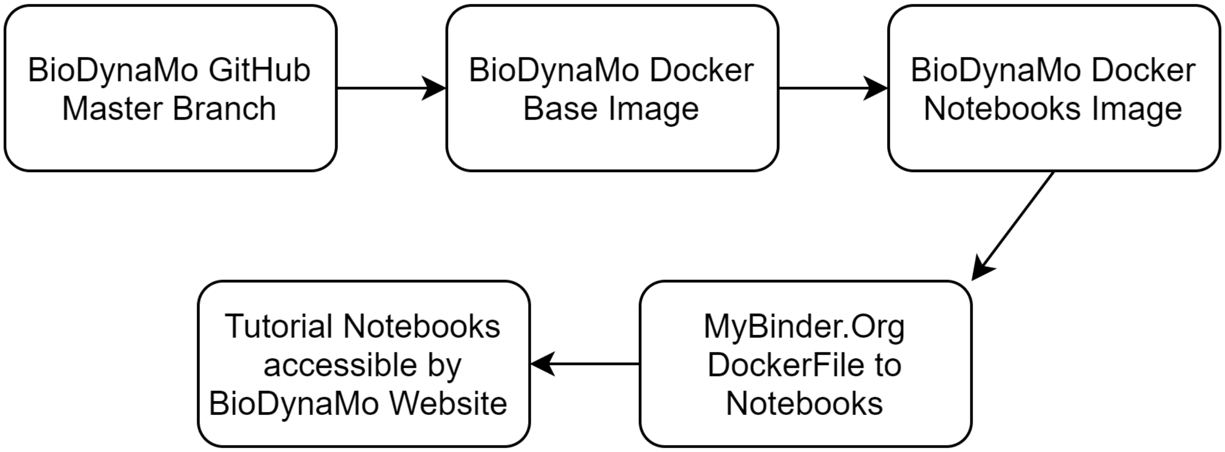

BioDynaMo GitHub Master Branch
BioDynaMo Docker Base Image
BioDynaMo Docker Notebooks Image
MyBinder.Org DockerFile to Notebooks
Tutorial Notebooks accessible by BioDynaMo Website

Figure 6. *BioDynaMo Container to Accessible Notebooks*

As illustrated in **Figure 6**, The complete process from BioDynaMo code to tutorials is as follows:

- Developers push code to the BioDynaMo GitHub repository master branch.
- GitHub actions build a base docker image and push it to Dockerhub.
- Users visits the BioDynaMo website and accesses tutorials.
- The user is redirected to MyBinder, which pulls the BioDynaMo container from Dockerhub and builds it.
- Users can execute BioDynaMo in the browser.

4. BIODYNAMO WEBSITE

BioDynaMo website hosts information about BioDynaMo, build, and development guides. In this project, I added two new pages on the website; to display information about the available demos and notebooks.

a. COMPILED DEMOS AND NOTEBOOK TUTORIALS

BioDynaMo supports complex simulations having multiple agents and behaviors. Depending on the number of agents and their interaction, we can use interpreted notebooks (interpreted notebooks are executed line by line by an interpreter) to experiment on multiple scenarios. However, there is a performance hit in interpreted mode. Compiled demos are faster and provide better performance but are less flexible. We have now published both demos and Interpreted notebooks on our website.

Users can execute and play with these demos and notebooks on a **JupyterLab** environment hosted by **MyBinder.Org**.

MyBinder:

MyBinder creates custom computing environments that can be shared and used by many remote users. We are using MyBinder to generate computing Environments using the Notebook Dockerfile that can run BioDynaMo Tutorial Notebooks [11].

JupyterLab:

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. It is helpful in a range of workflows in data science, scientific computing, and machine learning [12].

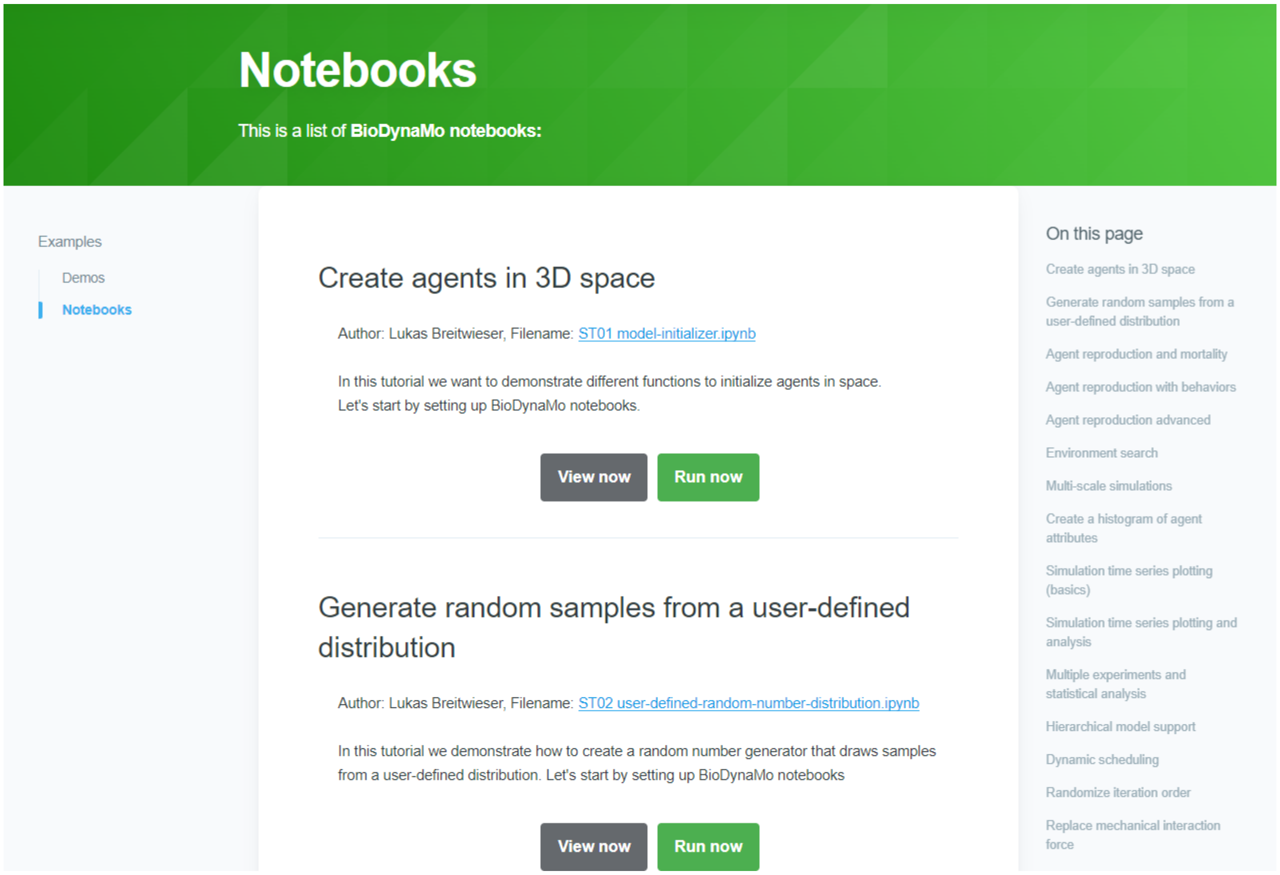
b. CHANGES TO BIODYNAMO WEBSITE

I created new pages to display BioDynaMo demos and tutorial notebooks:

The Interpreted Tutorials notebooks page is accessible by **/examples/notebooks** shown in **Figure 7.** The BioDynaMo demos page is accessible by **/examples/demos** shown in **Figure 8.**

A symlink is created in the **/content** folder of the website repository to access notebooks from BioDynaMo's main repository build. In the BioDynaMo main repository, the **/demo/** directory contains all demos; the **/build/notebook** directory contains all interpreted notebooks presented on the website.

While building the website, the Gatsby query accesses interpreted notebooks from the **/content/biodynamo/build/notebook,** and demo folders from **/content/biodynamo/demo.**



Notebooks

This is a list of **BioDynaMo notebooks:**

Create agents in 3D space

Author: Lukas Breitwieser, Filename: ST01 model-initializer.ipynb

In this tutorial we want to demonstrate different functions to initialize agents in space. Let's start by setting up BioDynaMo notebooks.

Generate random samples from a user-defined distribution

Author: Lukas Breitwieser, Filename: ST02 user-defined-random-number-distribution.ipynb

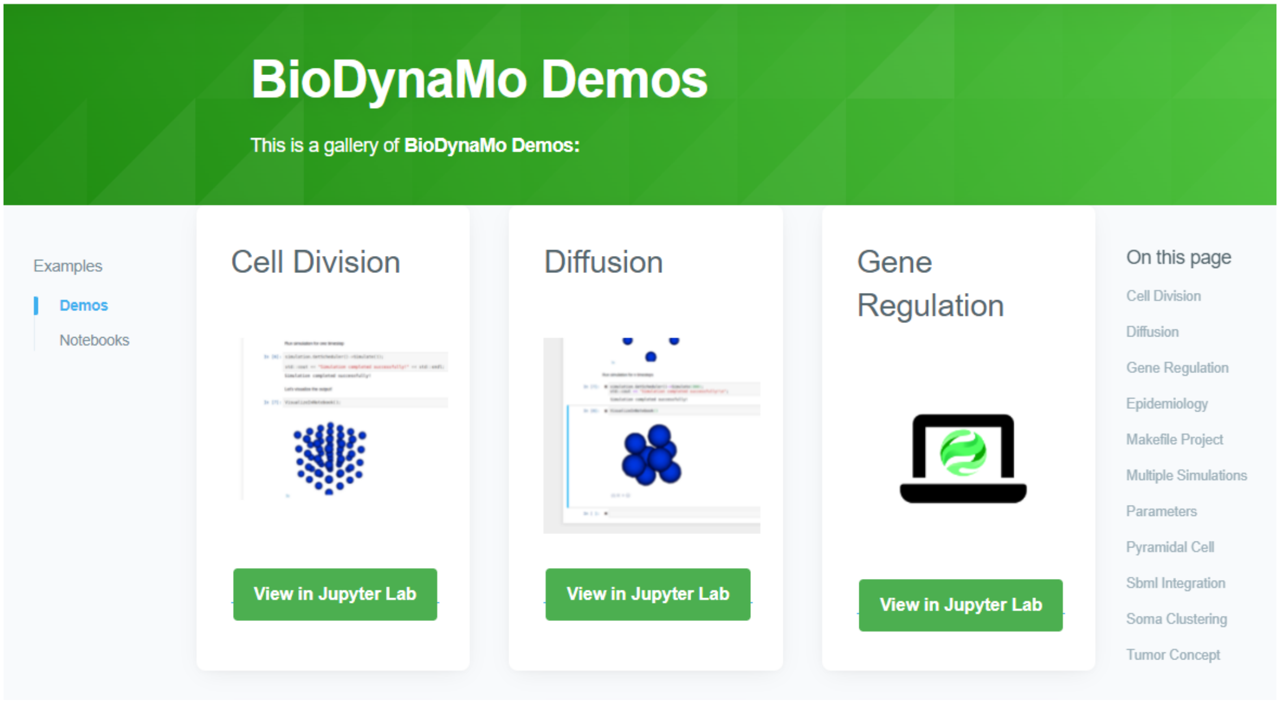
In this tutorial we demonstrate how to create a random number generator that draws samples from a user-defined distribution. Let's start by setting up BioDynaMo notebooks

On this page
Create agents in 3D space
Generate random samples from a user-defined distribution
Agent reproduction and mortality
Agent reproduction with behaviors
Agent reproduction advanced
Environment search
Multi-scale simulations
Create a histogram of agent attributes
Simulation time series plotting (basics)
Simulation time series plotting and analysis
Multiple experiments and statistical analysis
Hierarchical model support
Dynamic scheduling
Randomize iteration order
Replace mechanical interaction force

Figure 7. *BioDynaMo Interpreted Notebook Tutorials Page*

The Interpreted notebooks tutorials page is shown in **Figure 7**. The user can click the **"View Now"** button to view the notebook in the browser. To execute the notebook, the user can click the **"Run Now"** button. It will redirect to MyBinder and open the selected notebook.

Figure 10 shows MyBinder creating an environment from the BioDynaMo Notebook docker image. On successful environment creation, it redirects the user to the jupyter notebook, **Figure 11**.

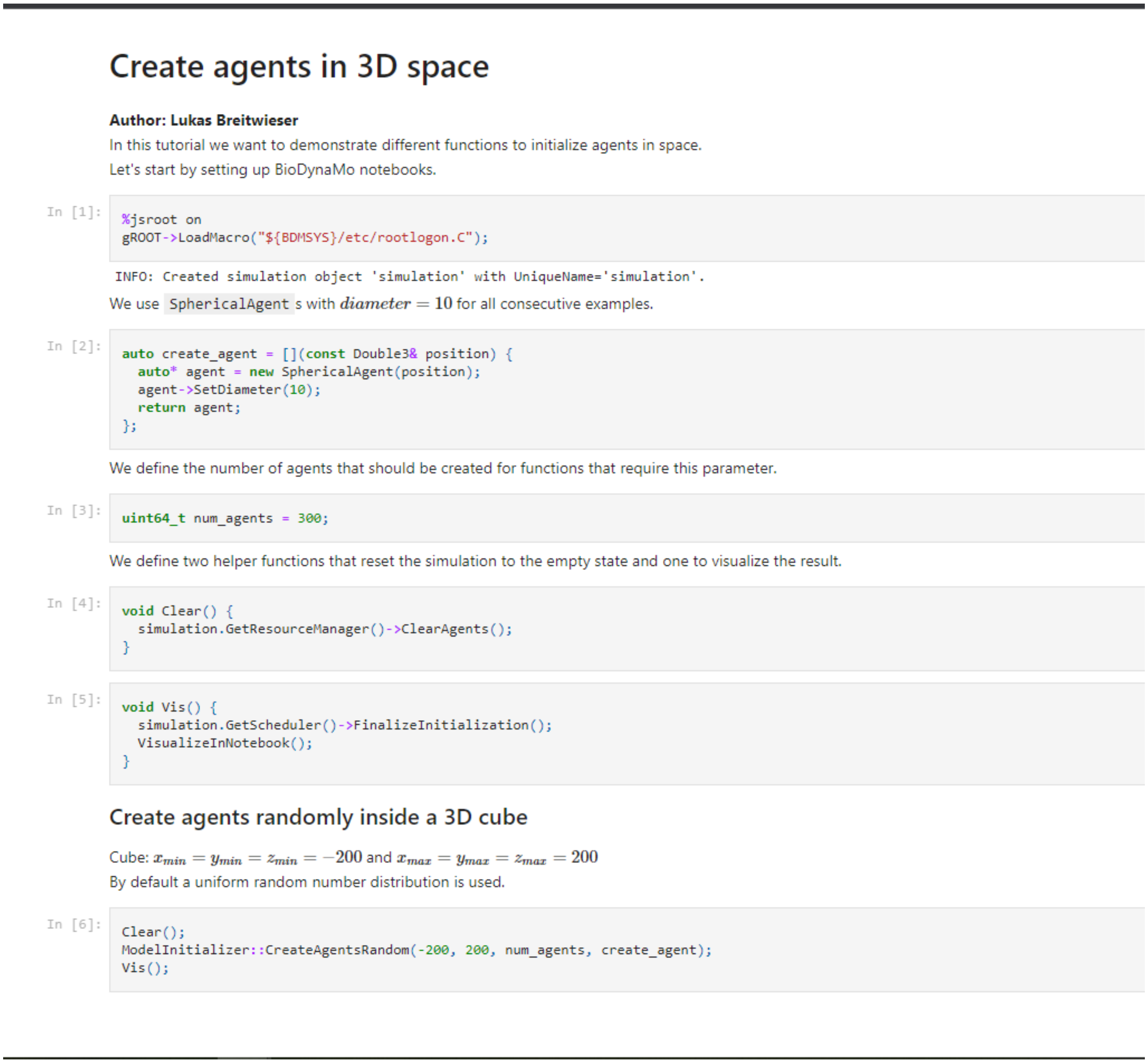


BioDynaMo Demos

This is a gallery of **BioDynaMo Demos:**

Figure 8. *BioDynaMo Demos Page*

Figure 8 shows the BioDynaMo demos page. On Clicking the "View on JupyterLab" button, the user is redirected to the JupyterLab folder of the respective notebook, as shown in **Figure 12.**



Create agents in 3D space

Author: Lukas Breitwieser

In this tutorial we want to demonstrate different functions to initialize agents in space.
Let's start by setting up BioDynaMo notebooks.

In [1]:
```
%jsroot on
gROOT->LoadMacro("${BDMSYS}/etc/rootlogon.C");
```

```
INFO: Created simulation object 'simulation' with UniqueName='simulation'.
```

We use `SphericalAgent` s with $diameter = 10$ for all consecutive examples.

In [2]:
```
auto create_agent = [](const Double3& position) {
  auto* agent = new SphericalAgent(position);
  agent->SetDiameter(10);
  return agent;
};
```

We define the number of agents that should be created for functions that require this parameter.

In [3]:
```
uint64_t num_agents = 300;
```

We define two helper functions that reset the simulation to the empty state and one to visualize the result.

In [4]:
```
void Clear() {
  simulation.GetResourceManager()->ClearAgents();
}
```

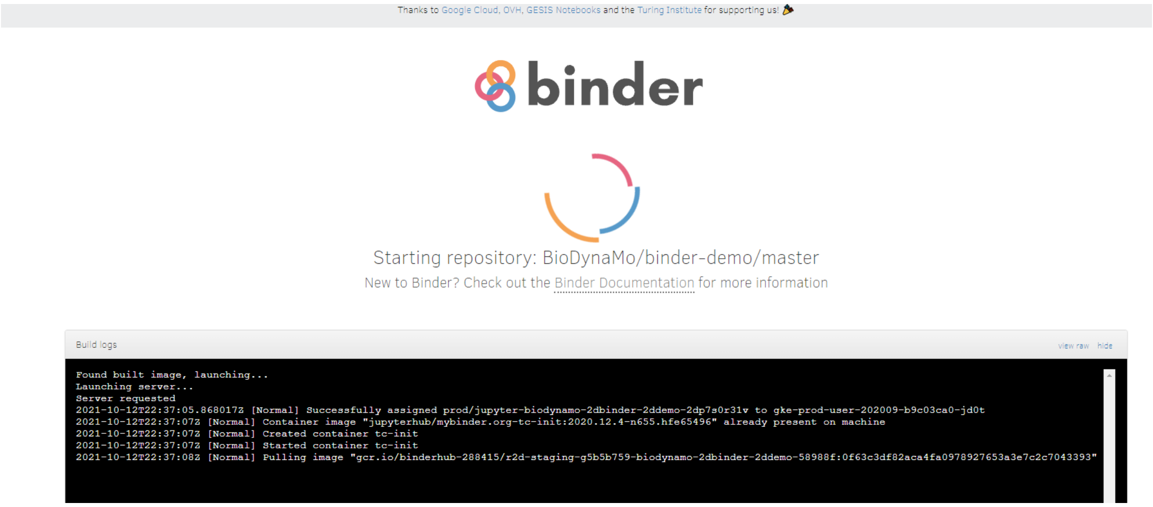
In [5]:
```
void Vis() {
  simulation.GetScheduler()->FinalizeInitialization();
  VisualizeInNotebook();
}
```

Create agents randomly inside a 3D cube

Cube: $x_{min} = y_{min} = z_{min} = -200$ and $x_{max} = y_{max} = z_{max} = 200$
By default a uniform random number distribution is used.

In [6]:
```
Clear();
ModelInitializer::CreateAgentsRandom(-200, 200, num_agents, create_agent);
Vis();
```

Figure 9. *Interpreted Notebook View on Website*



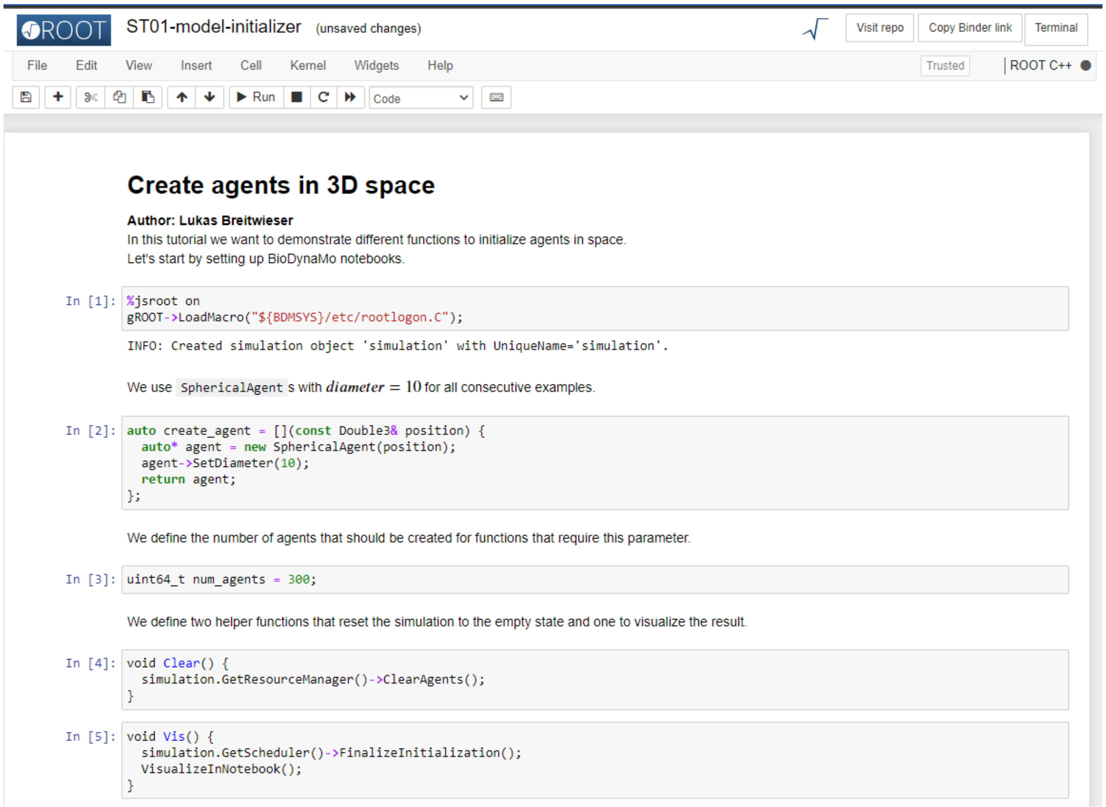
Thanks to Google Cloud, OVH, GESIS Notebooks and the Turing Institute for supporting us!
binder

Starting repository: BioDynaMo/binder-demo/master

New to Binder? Check out the Binder Documentation for more information

```
Found built image, launching...
Launching server...
Server requested
2021-10-12T22:37:05.868017Z [Normal] Successfully assigned prod/jupyter-biodynamo-2dbinder-2ddemo-2dp7s0r31v to gke-prod-user-202009-b9c03ca0-jd0t
2021-10-12T22:37:07Z [Normal] Container image "jupyterhub/mybinder.org-tc-init:2020.12.4-n655.hfe65496" already present on machine
2021-10-12T22:37:07Z [Normal] Created container tc-init
2021-10-12T22:37:07Z [Normal] Started container tc-init
2021-10-12T22:37:08Z [Normal] Pulling image "gcr.io/binderhub-288415/r2d-staging-g5b5b759-biodynamo-2dbinder-2ddemo-58988f:0f63c3df82aca4fa0978927653a3e7c2c7043393"
```

Figure 10. *MyBinder Creating Notebook Environment*



Create agents in 3D space

Author: Lukas Breitwieser

In this tutorial we want to demonstrate different functions to initialize agents in space.
Let's start by setting up BioDynaMo notebooks.

```
In [1]: %jsroot on
        gROOT->LoadMacro("${BDMSYS}/etc/rootlogon.C");

        INFO: Created simulation object 'simulation' with UniqueName='simulation'.
```

We use `SphericalAgent` s with $diameter = 10$ for all consecutive examples.

```
In [2]: auto create_agent = [](const Double3& position) {
          auto* agent = new SphericalAgent(position);
          agent->SetDiameter(10);
          return agent;
        };
```

We define the number of agents that should be created for functions that require this parameter.

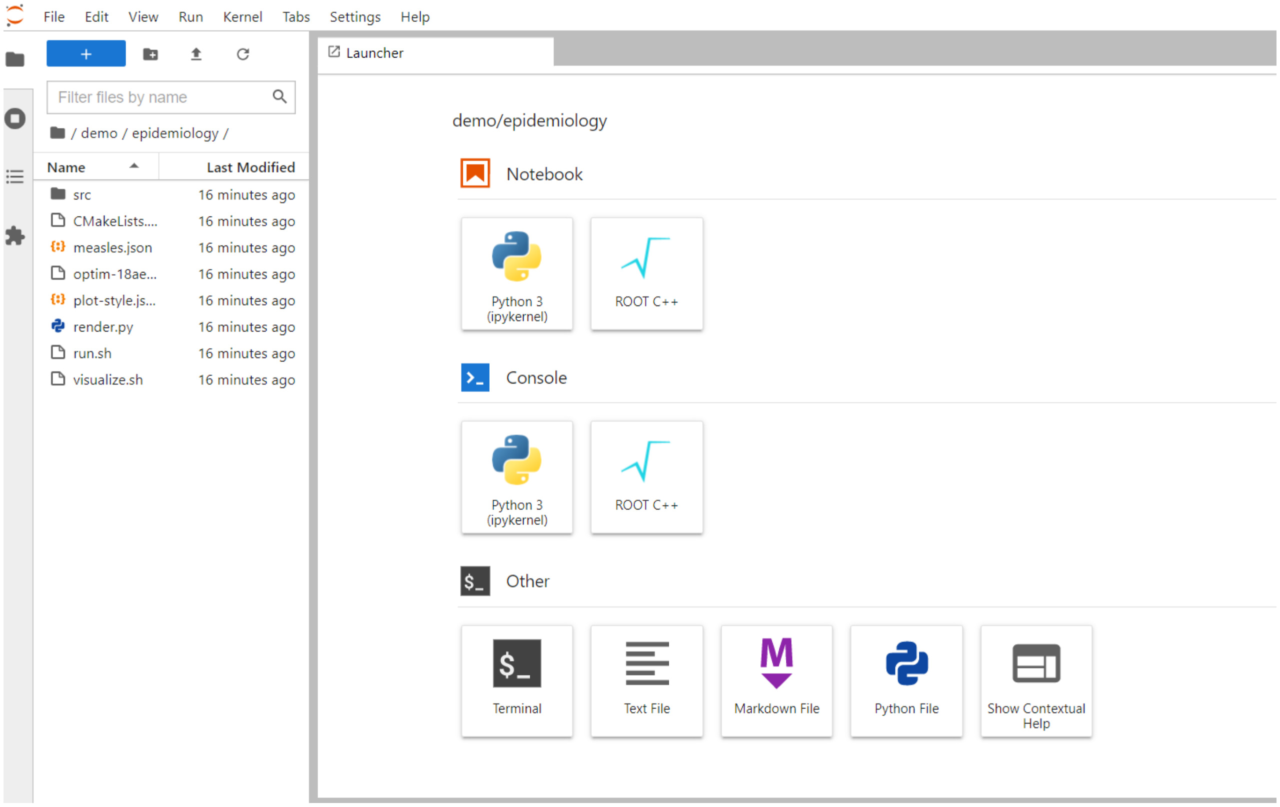
```
In [3]: uint64_t num_agents = 300;
```

We define two helper functions that reset the simulation to the empty state and one to visualize the result.

```
In [4]: void Clear() {
          simulation.GetResourceManager()->ClearAgents();
        }

In [5]: void Vis() {
          simulation.GetScheduler()->FinalizeInitialization();
          VisualizeInNotebook();
        }
```

Figure 11. *Interpreted Jupyter Notebook in MyBinder Environment*



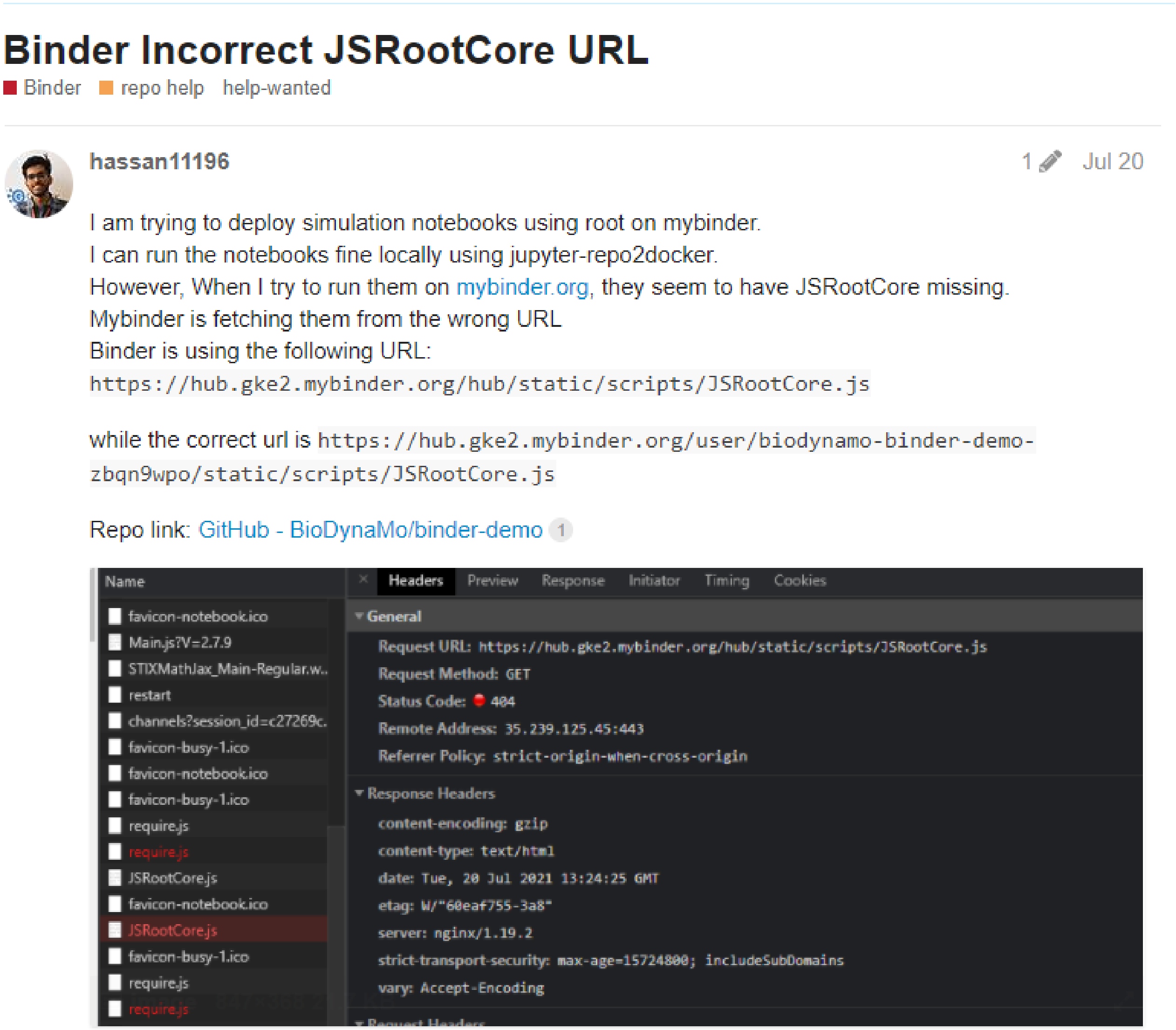
/ demo / epidemiology /

Name	Last Modified
src	16 minutes ago
CMakeLists....	16 minutes ago
measles.json	16 minutes ago
optim-18ae...	16 minutes ago
plot-style.js...	16 minutes ago
render.py	16 minutes ago
run.sh	16 minutes ago
visualize.sh	16 minutes ago

Figure 12. *BioDynaMo Demo Folder in JupyterLab*

5. FUTURE WORK

- All docker files and GitHub workflows need to be moved to the main BioDynaMo GitHub repository.
- Fix for visualizations failure on MyBinder notebooks due to MyBinder not finding JSRootCore **Figure 13.** I posted on the Jupyter Community Platform [13] regarding this problem, but the responses did not help me.



Binder Incorrect JSRootCore URL

🟥 Binder 🟧 repo help help-wanted

hassan11196 1 ✎ Jul 20

I am trying to deploy simulation notebooks using root on mybinder.
I can run the notebooks fine locally using jupyter-repo2docker.
However, When I try to run them on mybinder.org, they seem to have JSRootCore missing.
Mybinder is fetching them from the wrong URL
Binder is using the following URL:
`https://hub.gke2.mybinder.org/hub/static/scripts/JSRootCore.js`

while the correct url is `https://hub.gke2.mybinder.org/user/biodynamo-binder-demo-zbqn9wpo/static/scripts/JSRootCore.js`

Repo link: GitHub - BioDynaMo/binder-demo 1

```
favicon-notebook.ico
Main.js?V=2.7.9
STIXMathJax_Main-Regular.w..
restart
channels?session_id=c27269c.
favicon-busy-1.ico
favicon-notebook.ico
favicon-busy-1.ico
require.js
require.js
JSRootCore.js
favicon-notebook.ico
JSRootCore.js
favicon-busy-1.ico
require.js
require.js
```

```
▾ General
    Request URL: https://hub.gke2.mybinder.org/hub/static/scripts/JSRootCore.js
    Request Method: GET
    Status Code: ● 404
    Remote Address: 35.239.125.45:443
    Referrer Policy: strict-origin-when-cross-origin
▾ Response Headers
    content-encoding: gzip
    content-type: text/html
    date: Tue, 20 Jul 2021 13:24:25 GMT
    etag: W/"60eaf755-3a8"
    server: nginx/1.19.2
    strict-transport-security: max-age=15724800; includeSubDomains
    vary: Accept-Encoding
▾ Request Headers
```

Figure 13. *JSRootCore not found by MyBinder*

6. CONCLUSION

The process of building a docker container and publishing it to Docker Hub is fully automated. It can now be triggered on each push to master and all subsequent containers i.e. BioDynaMo base container and notebooks' containers will be updated. The BioDynaMo website is updated to make it easier to upload new notebooks and make them more accessible for the public.

7. REFERENCES

1. Breitwieser L, Hesam A, de Montigny J et al. BioDynaMo: a modular platform for high-performance agent-based simulation. Bioinformatics. 2021. doi:10.1093/bioinformatics/btab649
2. Rademakers F. Using BioDynaMo to Study COVID-19 Spread in Closed Spaces. HealthManagement. https://healthmanagement.org/c/healthmanagement/issuearticle/using-biodynamo-to-study-covid-19-spread-in-closed-spaces. Published 2021.
3. Containerization. Ibm.com. https://www.ibm.com/cloud/learn/containerization.
4. What is a Container? | Docker. https://www.docker.com/resources/what-container
5. What is Docker?. Ibm.com. https://www.ibm.com/cloud/learn/docker.
6. Berina Bandić, 2020. BioDynaMo Notebook Service. https://doi.org/10.5281/zenodo.4081577
7. Binder-Demo - https://github.com/BioDynaMo/binder-demo
8. Use a Dockerfile for your Binder repository - https://mybinder.readthedocs.io/en/latest/tutorials/dockerfile.html
9. GitHub Actions Documentation - GitHub Docs. GitHub Docs. https://docs.github.com/en/actions
10. Docker Hub. Hub.docker.com. https://hub.docker.com/.
11. MyBinder.Org - https://mybinder.org/.
12. Jupyter - https://jupyter.org/
13. Binder Incorrect JSRootCore URL- https://discourse.jupyter.org/t/binder-incorrect-jsrootcore-url/10046