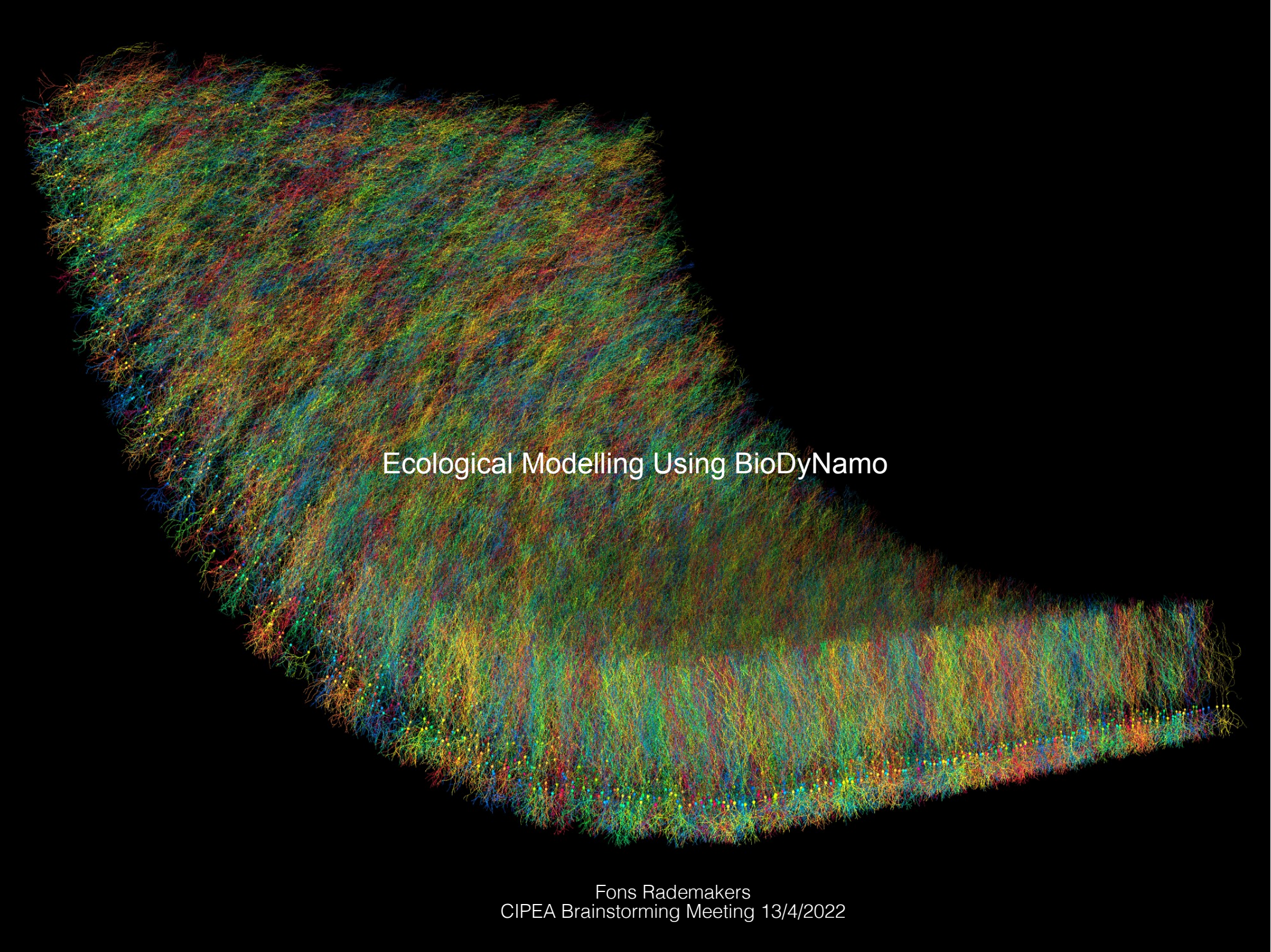
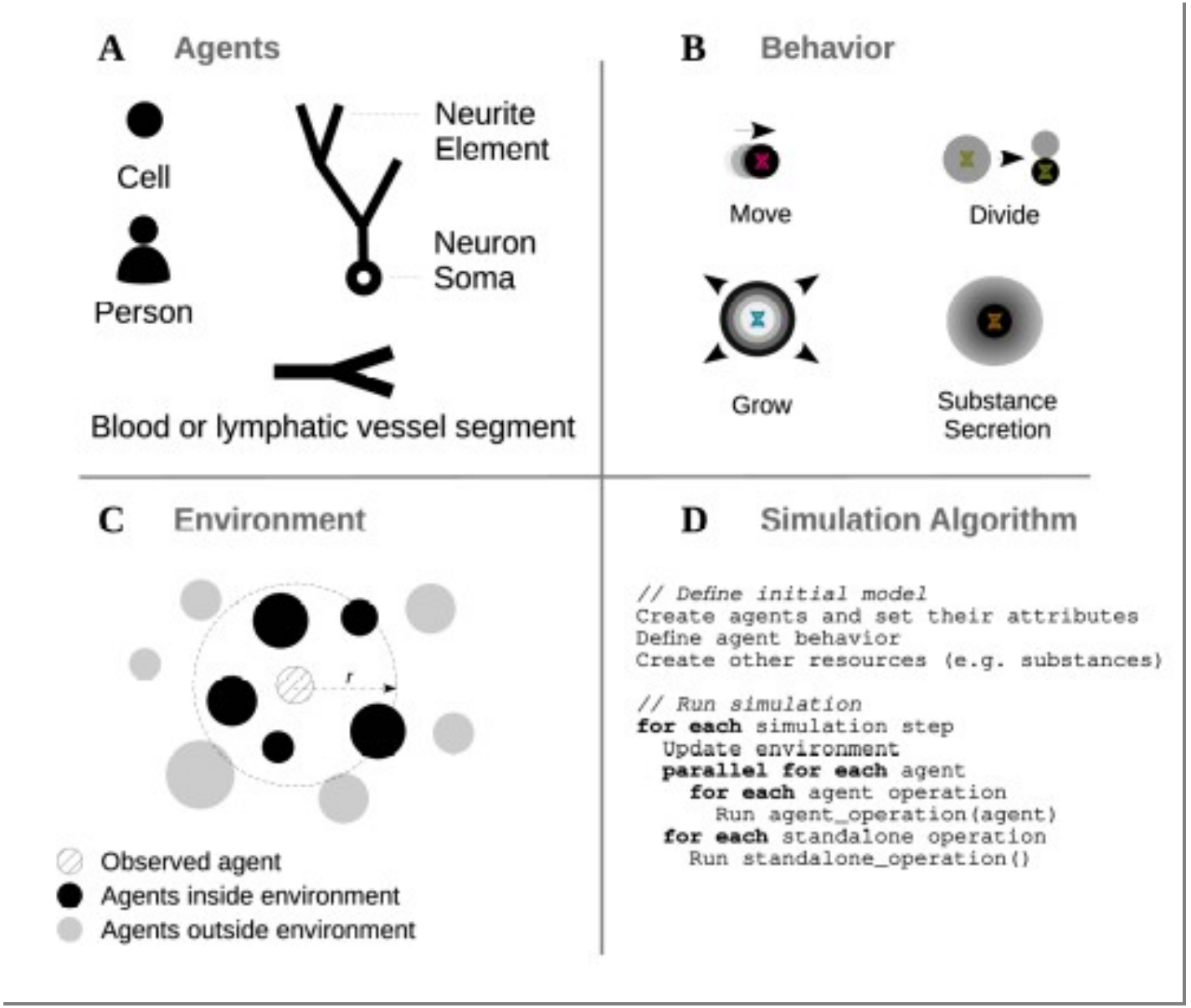


Fons Rademakers
CIPEA Brainstorming Meeting 13/4/2022

BioDynaMo is a **modular**, **high-performance** **agent-based** simulation platform written in C++

Agent-based simulation



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

Scientific Impact

Community Impact

I. Main BioDynaMo Article

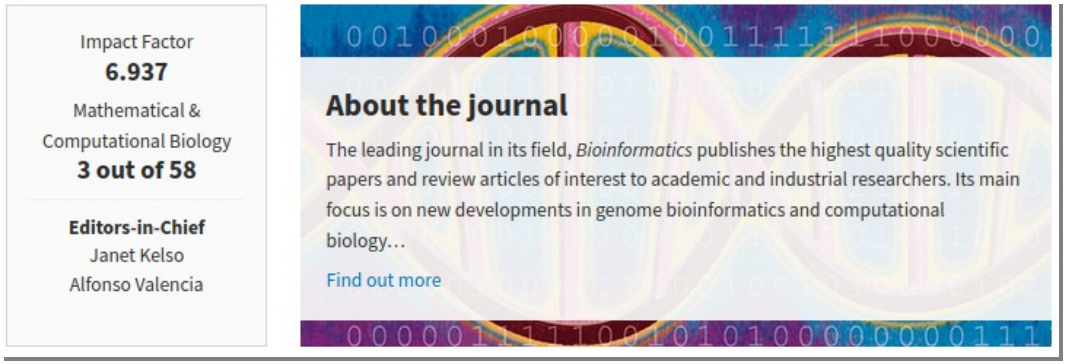
Bioinformatics, 2021, 1–8
doi: 10.1093/bioinformatics/btab649
Advance Access Publication Date: 16 September 2021
Original Paper

OXFORD

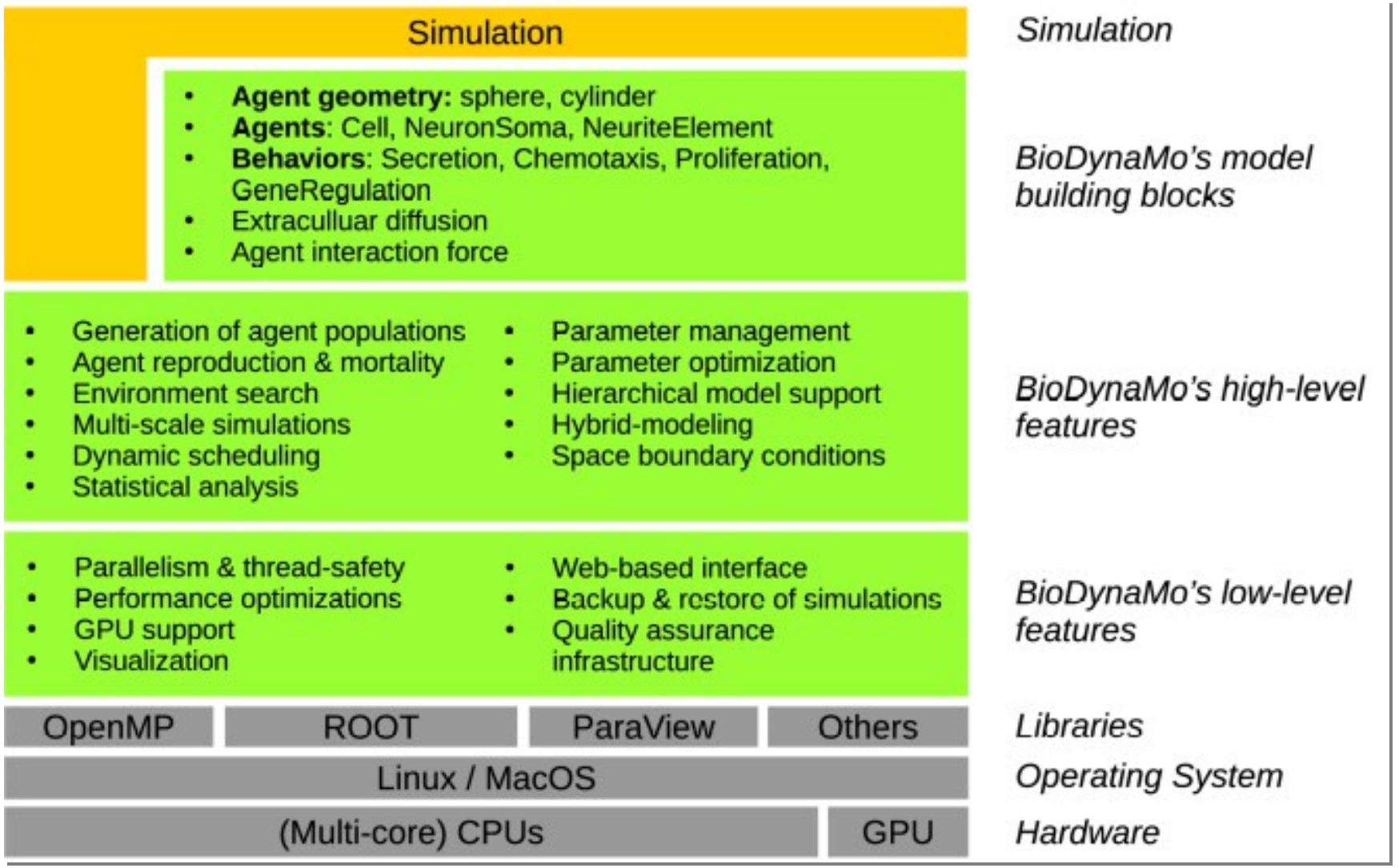
Systems biology

BioDynaMo: a modular platform for high-performance agent-based simulation

Bioinformatics Journal (Oxford)

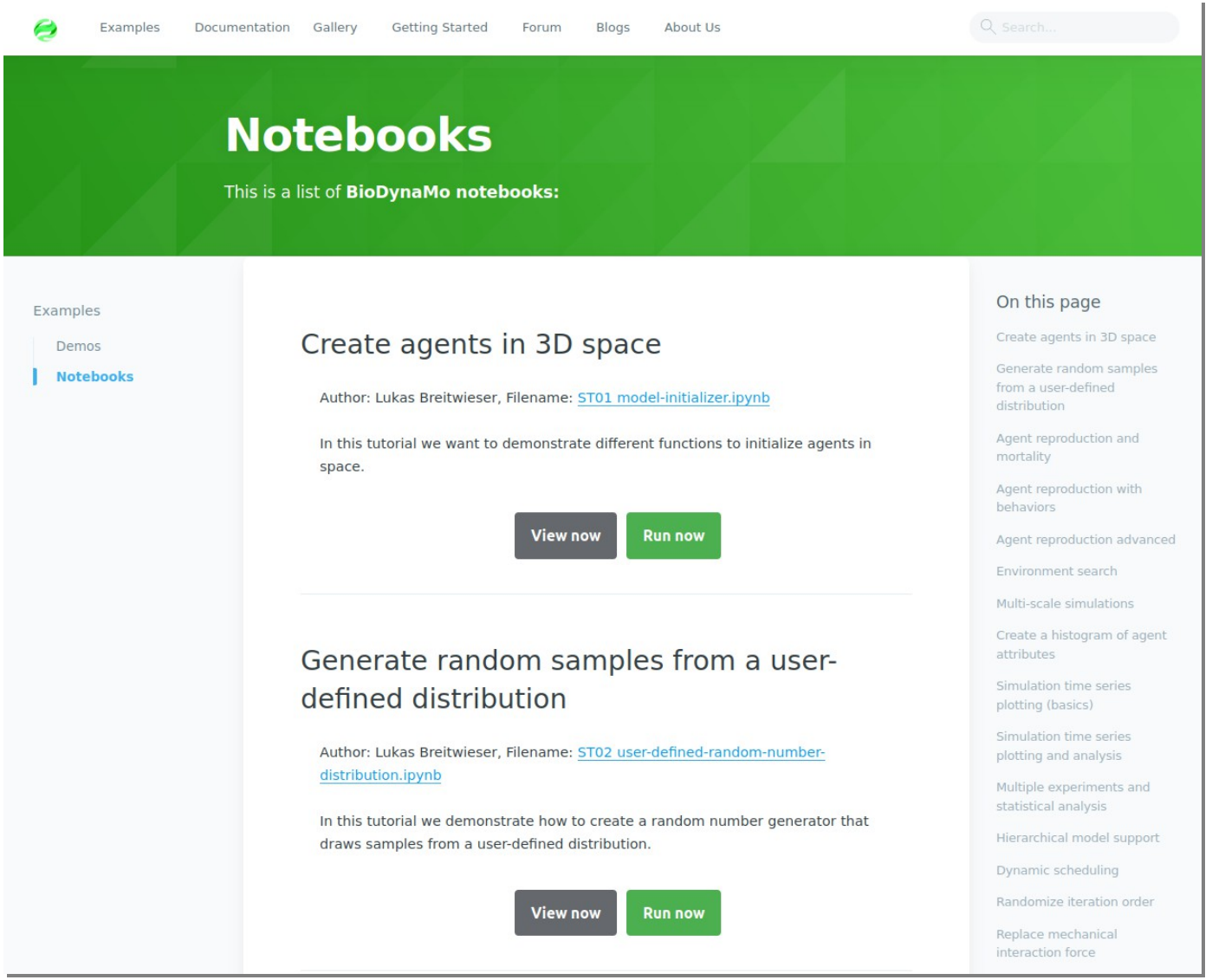


BioDynaMo overview



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

15 BioDynaMo Notebooks



https://biodynamo.org/examples/notebooks

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();
```

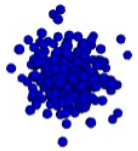
Create agents randomly inside a 3D cube using a gaussian distribution

Cube: $x_{min} = y_{min} = z_{min} = -200$ and $x_{max} = y_{max} = z_{max} = 200$

Gaussian: $\mu = 0, \sigma = 20$

Note the extra parameter rng passed to `CreateAgentsRandom`

In [7]:
```
Clear();
auto rng = simulation.GetRandom()->GetGausRng(0, 20);
ModelInitializer::CreateAgentsRandom(-200, 200, num_agents, create_agent, &rng);
Vis();
```



https://biodynamo.org/notebooks/ST01-model-initializer.html

Create a histogram of agent attributes

Author: Lukas Breitwieser

In this tutorial we will show how to create a histogram of all agent diameters in the simulation and fit a function to the data.

Let's start by setting up BioDynaMo notebooks.

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

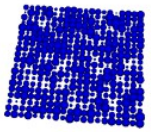
```
INFO: Created simulation object 'simulation' with UniqueName='simulation'.
```

We want to define a function that creates a cell at a certain position with diameters drawn from a gaussian distribution with $\mu = 20$ and $\sigma = 5$. The smallest diameter should be larger then 2.0.

In [2]:
```
simulation.GetResourceManager()->ClearAgents();
auto rng = simulation.GetRandom()->GetGausRng(20, 5);
auto create_cell = [&](const Double3& position) {
  Cell* cell = new Cell(position);
  double diameter = std::max(2.0, rng.Sample());
  cell->SetDiameter(diameter);
  return cell;
};
```

Now that we defined `create_cell` we can use it to create 400 cells on a plane with $z = 0$, $xmin = ymin = -200$, $xmax = ymax = 200$, and spacing = 20 in both dimensions.

In [3]:
```
auto f = [](const double* x, const double* params) { return 0.0; };
ModelInitializer::CreateAgentsOnSurface(f, {}, -200, 200, 20, -200, 200, 20,
                                        create_cell);
simulation.GetScheduler()->FinalizeInitialization();
VisualizeInNotebook(300, 300);
```



The next step is to create a histogram object with 100 bins in the interval [2, 40].

The second line creates a function which fills the histogram with the diameter of the given agent.

The third line calls the function `fill` for each agent, thus adding all diameters to the histogram.

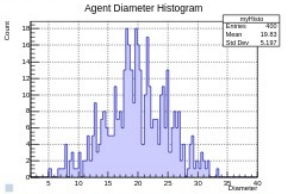
In [4]:
```
TH1F h("myHisto","Agent Diameter Histogram;Diameter;Count", 100, 2, 40);
auto fill = L2F([&](Agent* a, AgentHandle){ h.Fill(a->GetDiameter()); });
simulation.GetResourceManager()->ForEachAgent(fill);
```

Let's draw the final histogram.

Before we have to create a `TCanvas` object in order to display the result in this notebook.

We also modify the default color and create a grid.

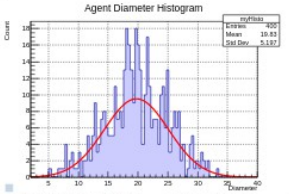
In [5]:
```
TCanvas c("", "", 400, 300);
h.SetFillColor(kBlue - 10);
c.SetGrid();
h.Draw();
c.Draw();
```



Finally, we can try to fit a function to the data in the histogram.

Since we drew samples from a gaussian random number generator when we created our cells, we expect that a gaussian will fit our data.

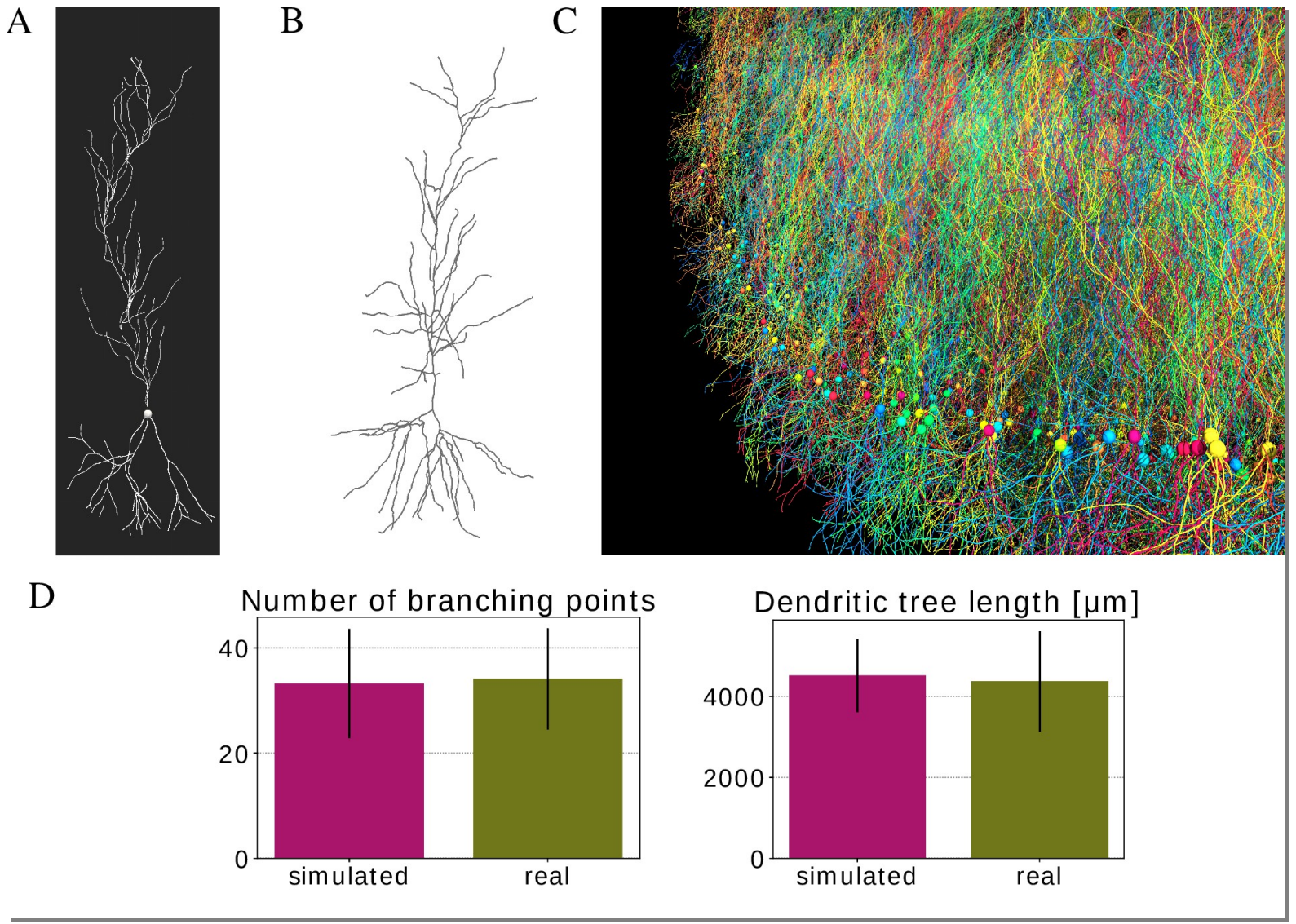
In [6]:
```
h.Fit("gaus", "S");
h.Draw();
c.Draw();
```




```
FCN=73.1872 FROM MIGRAD    STATUS=CONVERGED      78 CALLS          79 TOTAL
                     EDM=6.76682e-08     STRATEGY= 1      ERROR MATRIX ACCURATE
 EXT PARAMETER                                   STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  Constant     9.50544e+00   7.09907e-01   2.21048e-03   4.02840e-04
  2  Mean         1.97314e+01   3.23457e-01   1.36014e-03  -3.65230e-04
  3  Sigma        5.40166e+00   3.17423e-01   6.23484e-05   2.34242e-02
```

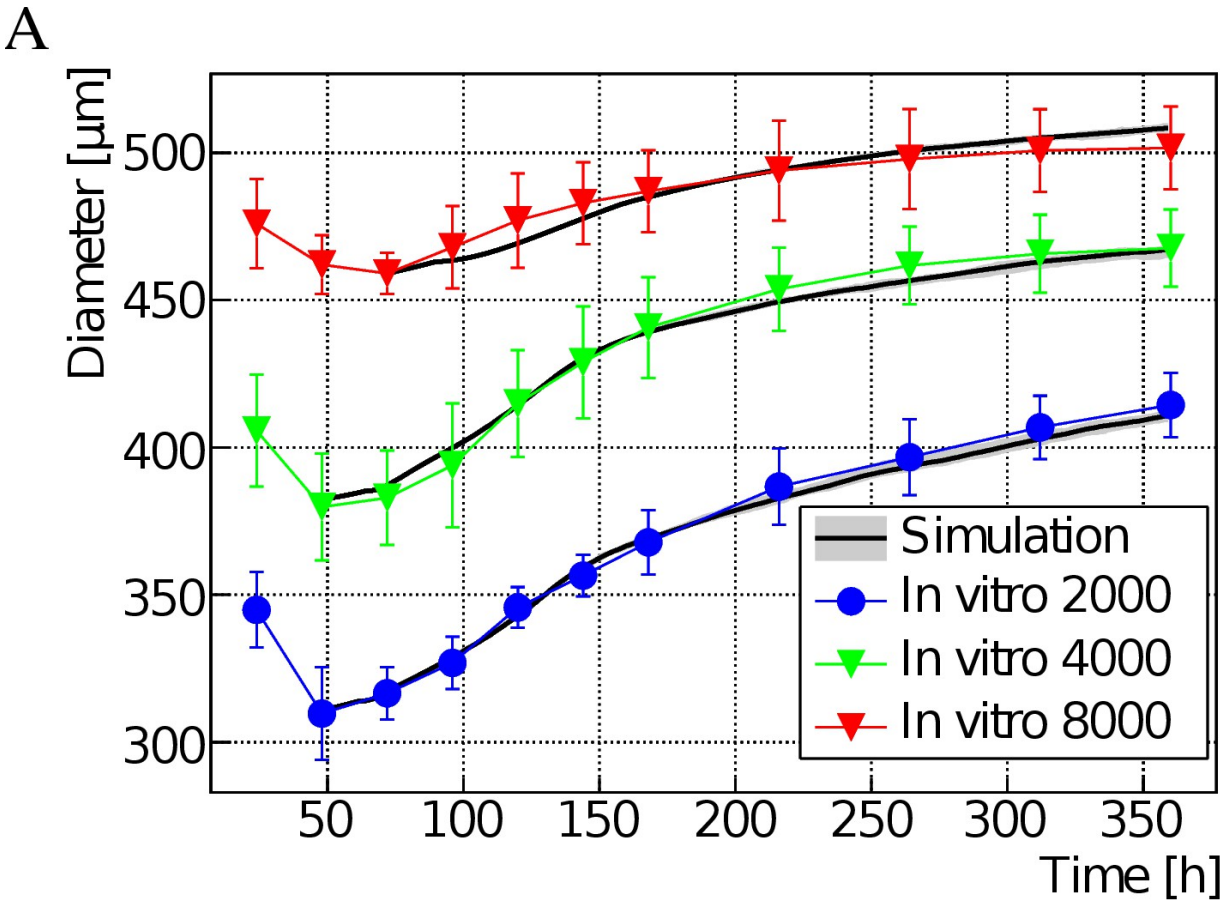
https://biodynamo.org/notebooks/ST08-histograms.html

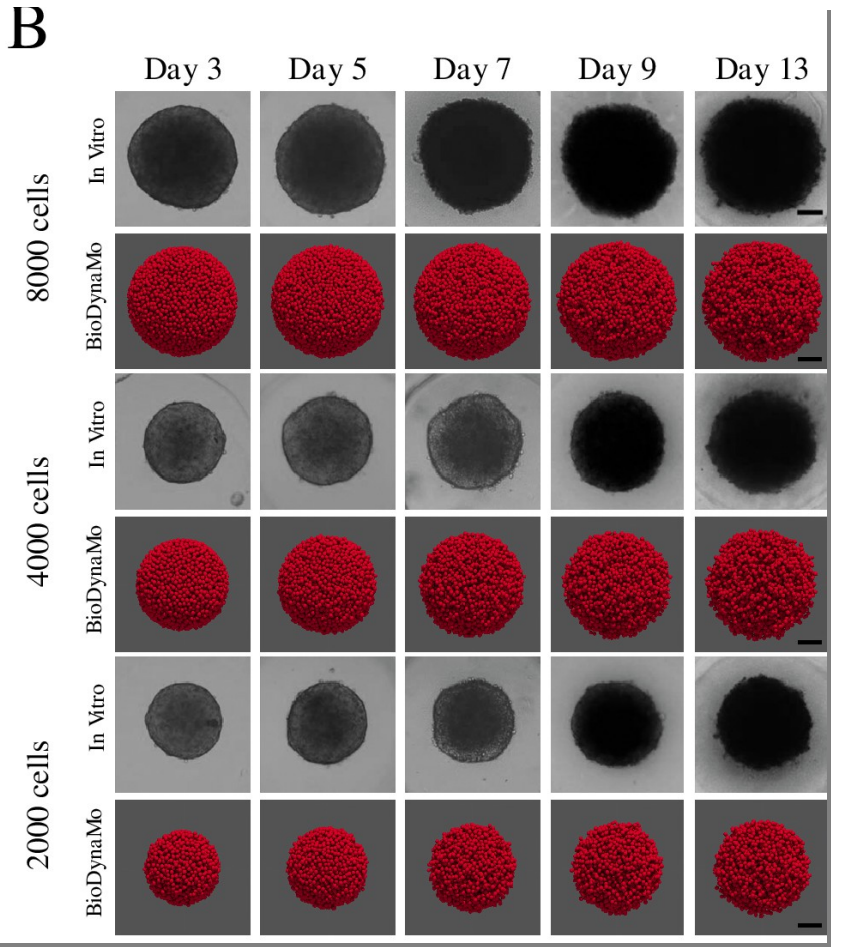
Neuroscience Use Case



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

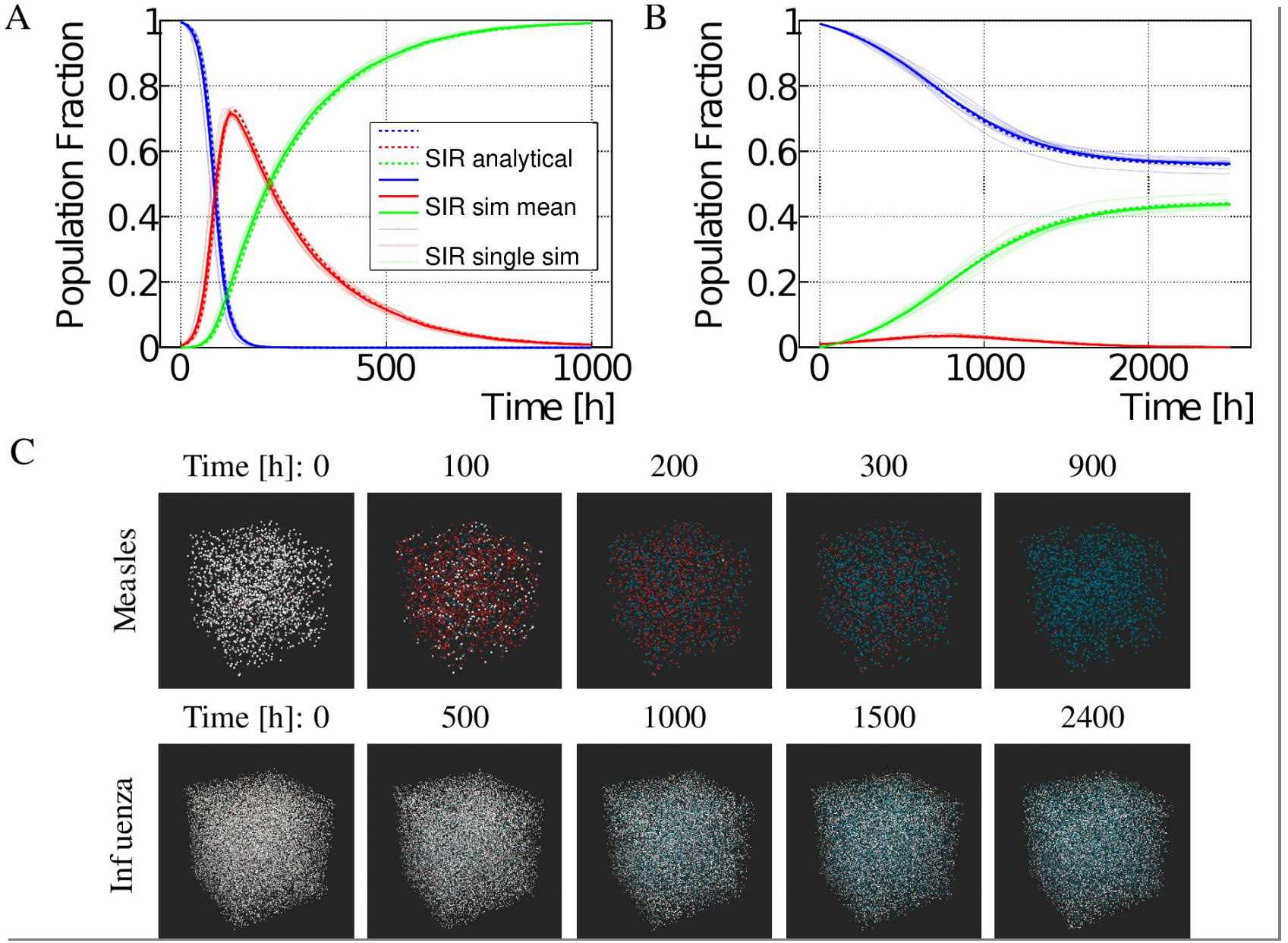
Oncology Use Case





Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

Epidemiology Use Case



Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

BioDynaMo Performance

Table 1. **Performance data.** The values in column “Agents" and “Diffusion volumes" are taken from the end of the simulation. Runtime measures the wall-clock time to simulate the number of iterations. It excludes the time for simulation setup and visualization. The entries in column “System” correspond to Supplementary File S1 Table 5. Supplementary File S1 Table 6 contains more detailed performance data.

Simulation	Agents	Diffusion volumes	Iterations	System	Physical CPUs	Runtime	Memory
Neuroscience use case							
Single (Figure 4A)	1 494	250	500	A	1	0.16 s	382 MB
Large-scale (Figure 4C)	9 054 740	65 536	500	A	72	36 s	6.02 GB
Very-large-scale	1 018 644 154	5 606 442	500	B	72	1 h 26 min	436 GB
Oncology use case (Figure 5)							
2000 initial cells	4 177	0	312	A	1	1.05 s	382 MB
Large-scale	1 000 3925	0	288	A	72	1 min 42 s	7.42 GB
Very-large-scale	986 054 868	0	288	B	72	6 h 21 min	604 GB
Epidemiology use case (Figure 6C)							
Measles	2 010	0	1000	A	1	0.53 s	381 MB
Seasonal Influenza	20 200	0	2500	A	1	16.41 s	383 MB
Large-scale (measles)	10 050 000	0	1000	A	72	59.19 s	5.87 GB
Very-large-scale (measles)	1 005 000 000	0	1000	B	72	2 h 0 min	495 GB

Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

Scientific Impact

Community Impact

Ongoing BioDynaMo Projects

- Spatial spread of HIV in Malawi with the UNIGE
- Retinal self-organization
- Radiation induced lung injury simulation
- Substitute in-vitro experiments with simulations
- Modelling COVID-19 spread in closed environments
- Simulation of financial markets
- Wealth distribution in societies
- Simulation of mosquito borne diseases

Agent-Based Modelling of Social-Ecological Systems

- Understanding Social-Ecological Systems (SES) is crucial to supporting the sustainable management of resources
- ABM is a valuable tool to achieve this because it can represent the behaviour and interactions of organisms, human actors and institutions
- ABM have already been widely used to study SES
- However, ABMs of SES are by their very nature complex and compute intensive

Example Ecological ABM's

- Interaction between human and the natural habitat of wildlife (road construction, deforestation, animal reproduction)
- Smart cities where traffic is based on ABM and derived predictions
- Modeling the spreading of wildfires in order to reduce large forest fires

Summary

- BioDynaMo is capable of simulating **billions of agents** in a **wide variety of research fields**.
- BioDynaMo is a generic Agent-Based Simulation toolkit well suite to also model SES
- We are looking forward to work together with domain experts on creating proof of concept social-ecological simulations

Questions?

For more see https://biodynamo.org