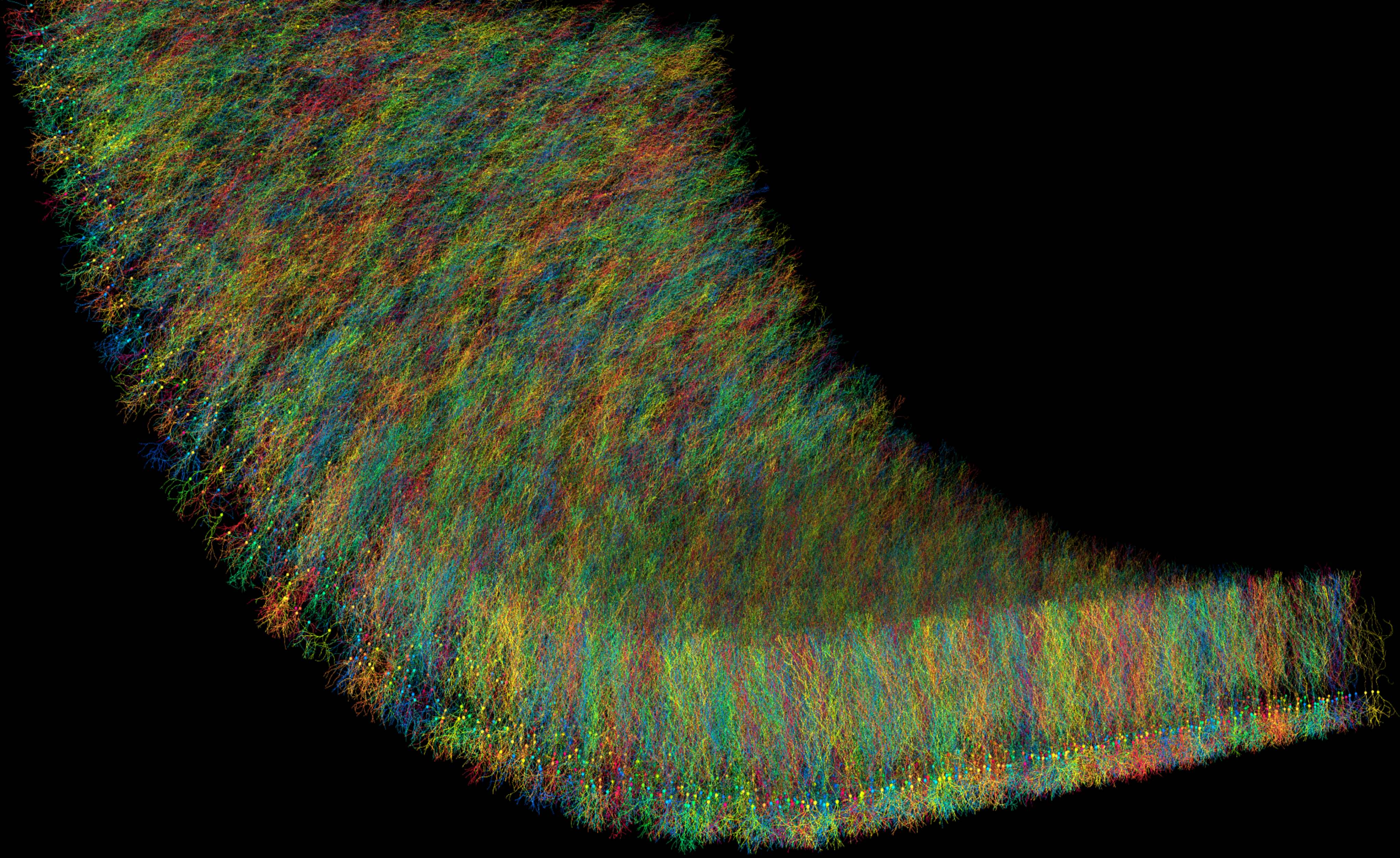
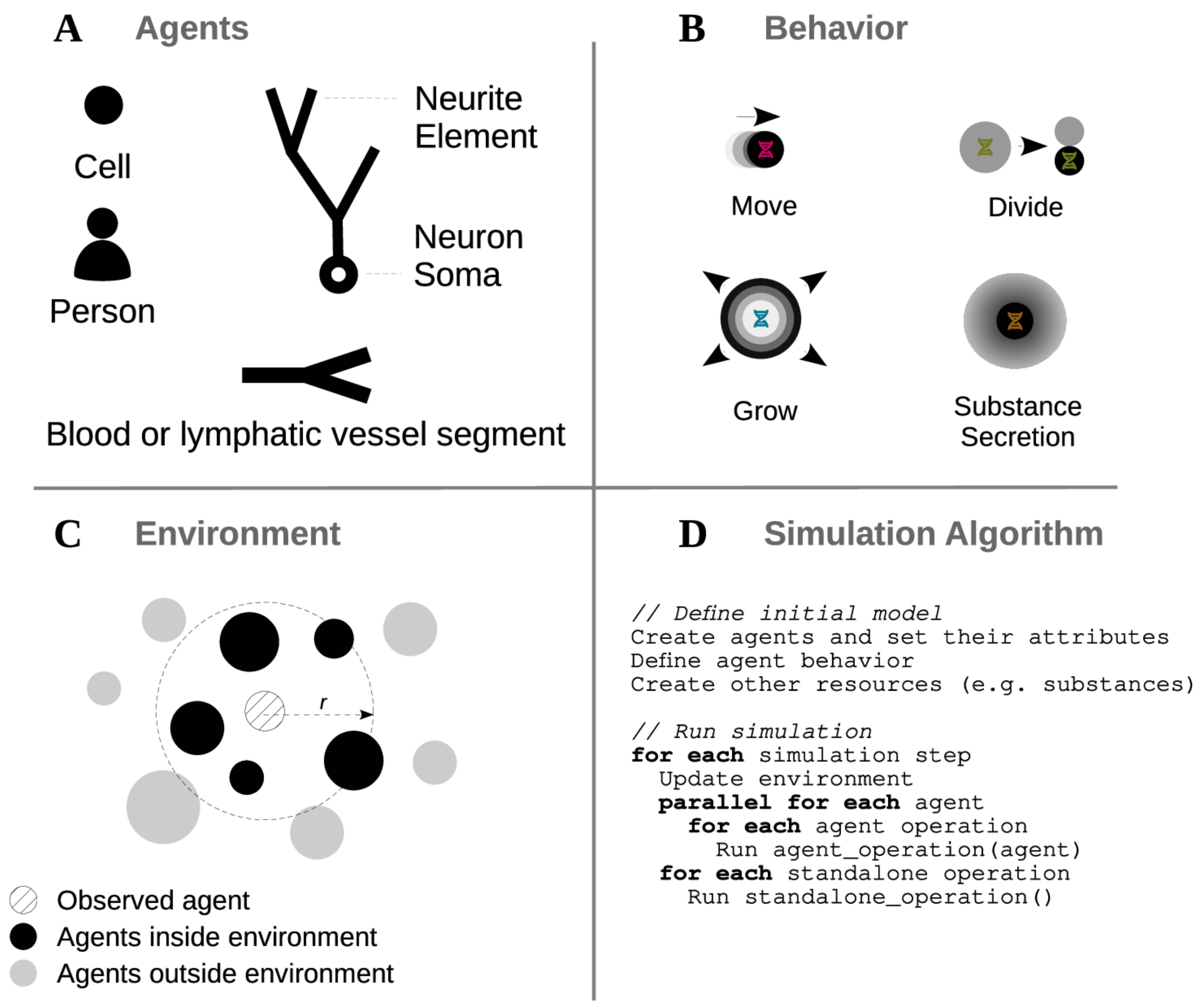


BioDynaMo

Lukas Breitwieser
Tobias Duswald
CERN BIC Screening Week 26.10.2021

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

Agent-based simulation 1/2




```
// Define initial model
Create agents and set their attributes
Define agent behavior
Create other resources (e.g. substances)

// Run simulation
for each simulation step
  Update environment
  parallel for each agent
    for each agent operation
      Run agent_operation(agent)
  for each standalone operation
    Run standalone_operation()
```


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

Agent-based simulation 2/2

- Also called individual-based modeling
- Bottom-up approach
  - Modeling the trees not the forest
- Characteristics
  - Local interaction
  - Emergent behavior

Agent-based simulation is very versatile

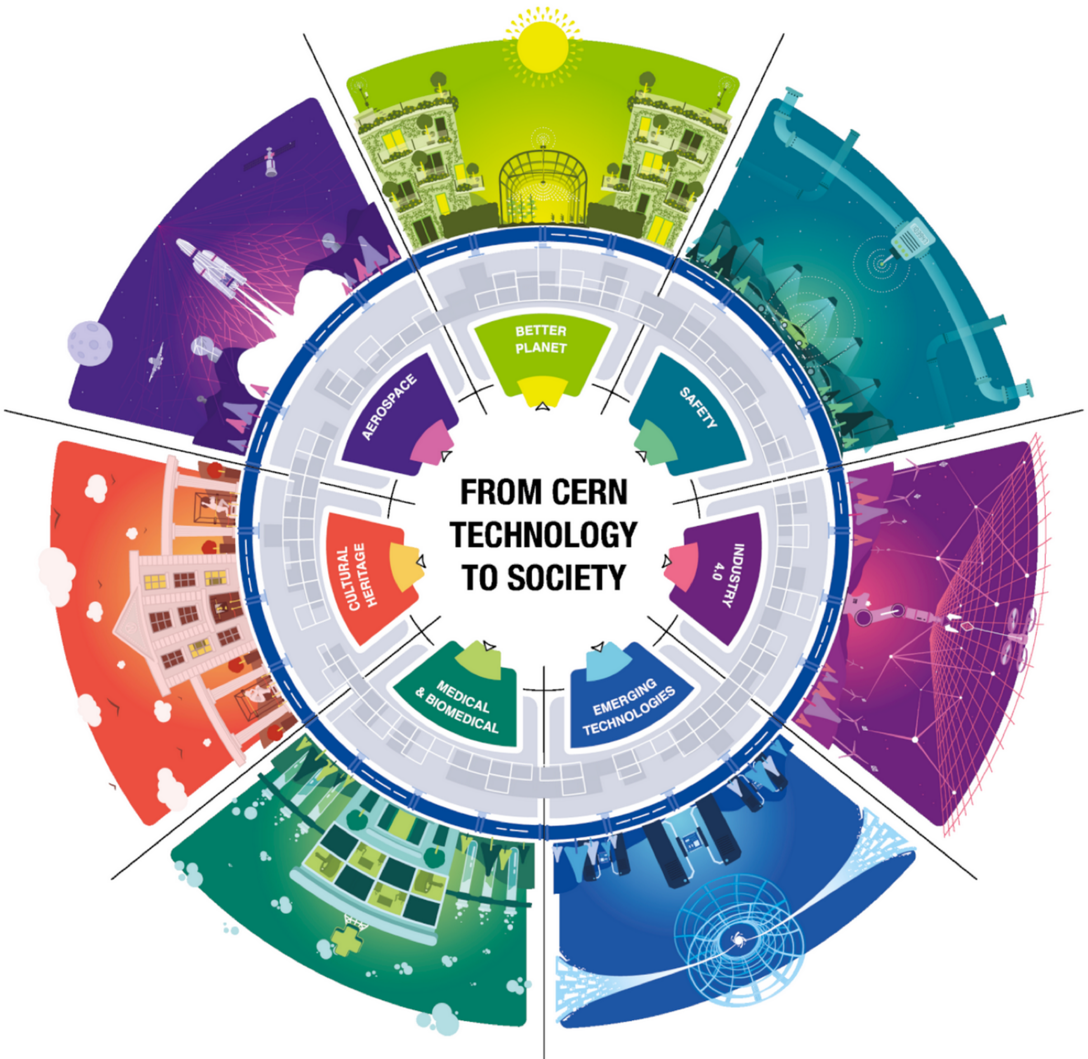
Table 1: A sample of recent agent-based applications

Application Area:	Agent-based Model Focus:
Agriculture	A spatial individual-based model prototype for assessing potential pesticide exposure of farm-workers conducting small-scale agricultural production (Leyk et al. 2009)
Air Traffic Control	Air traffic control to analyze control policies and performance of an air traffic management facility (Conway 2006)
Anthropology	Prehistoric settlement patterns and political consolidation in the Lake Titicaca basin of Peru and Bolivia (Griffin and Stanish 2007)
Biomedical Research	*The Basic Immune Simulator*, to study the interactions between innate and adaptive immunity (Folcik et al. 2007)
Crime Analysis	A realistic virtual urban environment, populated with virtual burglar agents (Malleson 2010)
Ecology	Investigate the trade-off between road avoidance and salt pool spatial memory in the movement behavior of moose (Grosman et al. 2011) Predator-prey relationships between transient killer whales and other marine mammals (Mock and Testa 2007)
Energy Analysis	A building occupant network energy consumption decision-making model (Chen et al. 2011) Application for the Smart Grid (Jackson 2010) Energy investment decision making (Tobias 2008) Oil refinery supply chain (Van Dam et al. 2008)
Epidemiology	Pandemic disease model accounting for individual behavior and demographics (Aleman et al. 2010) Global-scale agent model of disease transmission (Parker and Epstein 2011)
Evacuation	Tsunami evacuation using a modified form of Helbing's social-force model applied to agents (Puckett 2009)
Market Analysis	Consumer marketing model developed in collaboration with a Fortune 50 firm (North et al. 2009) Consumer airline market share (Kuhn et al. 2010) Simulation that models the possibilities for a future market in sub-orbital space tourism (Charania et al. 2006)
Social Networks	Model of email-based social networks, in which individuals establish, maintain and allow atrophy of links through contact-lists and emails (Menges et al. 2008)

Source: Macal and North (2014), https://doi.org/10.1109/WSC.2014.7019874

Why at CERN?

Knowledge Transfer!

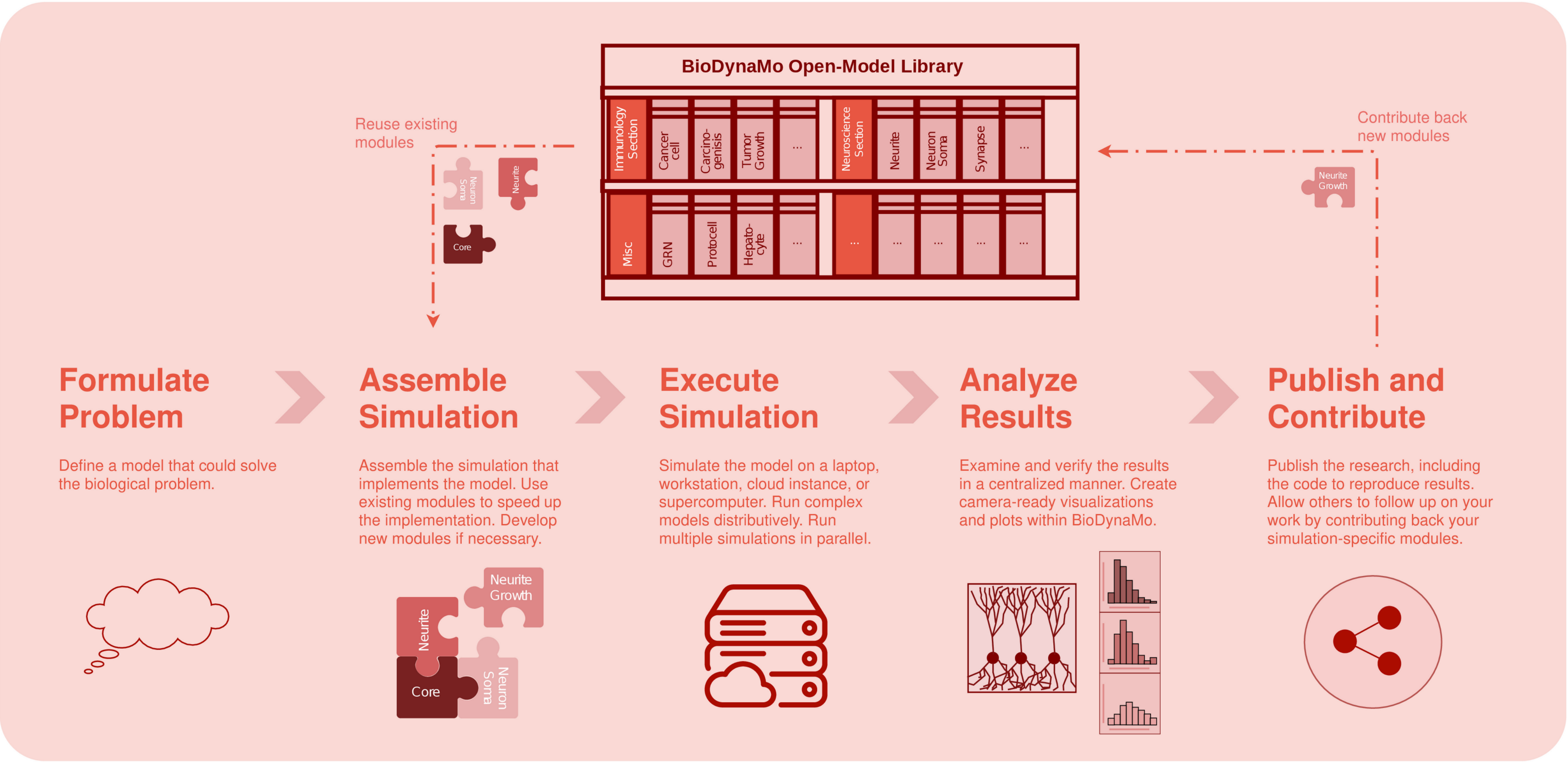


Source: https://kt.cern

BioDynaMo’s Goal

- Develop a new agent-based simulation platform that
  - supports **large-scale** biological simulations
  - supports different fields (e.g. neuroscience, oncology, immunology, …) with a **modular software design**
  - **hides the complexity** of parallel and distributed computing
  - promotes **reproducibility** of results

The envisioned platform



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

BioDynaMo overview

Simulation — *Simulation*

BioDynaMo's model building blocks

- **Agent geometry:** sphere, cylinder
- **Agents**: Cell, NeuronSoma, NeuriteElement
- **Behaviors**: Secretion, Chemotaxis, Proliferation, GeneRegulation
- Extracelluar diffusion
- Agent interaction force

BioDynaMo's high-level features

- Generation of agent populations
- Agent reproduction & mortality
- Environment search
- Multi-scale simulations
- Dynamic scheduling
- Statistical analysis
- Parameter management
- Parameter optimization
- Hierarchical model support
- Hybrid-modeling
- Space boundary conditions

BioDynaMo's low-level features

- Parallelism & thread-safety
- Performance optimizations
- GPU support
- Visualization
- Web-based interface
- Backup & restore of simulations
- Quality assurance infrastructure

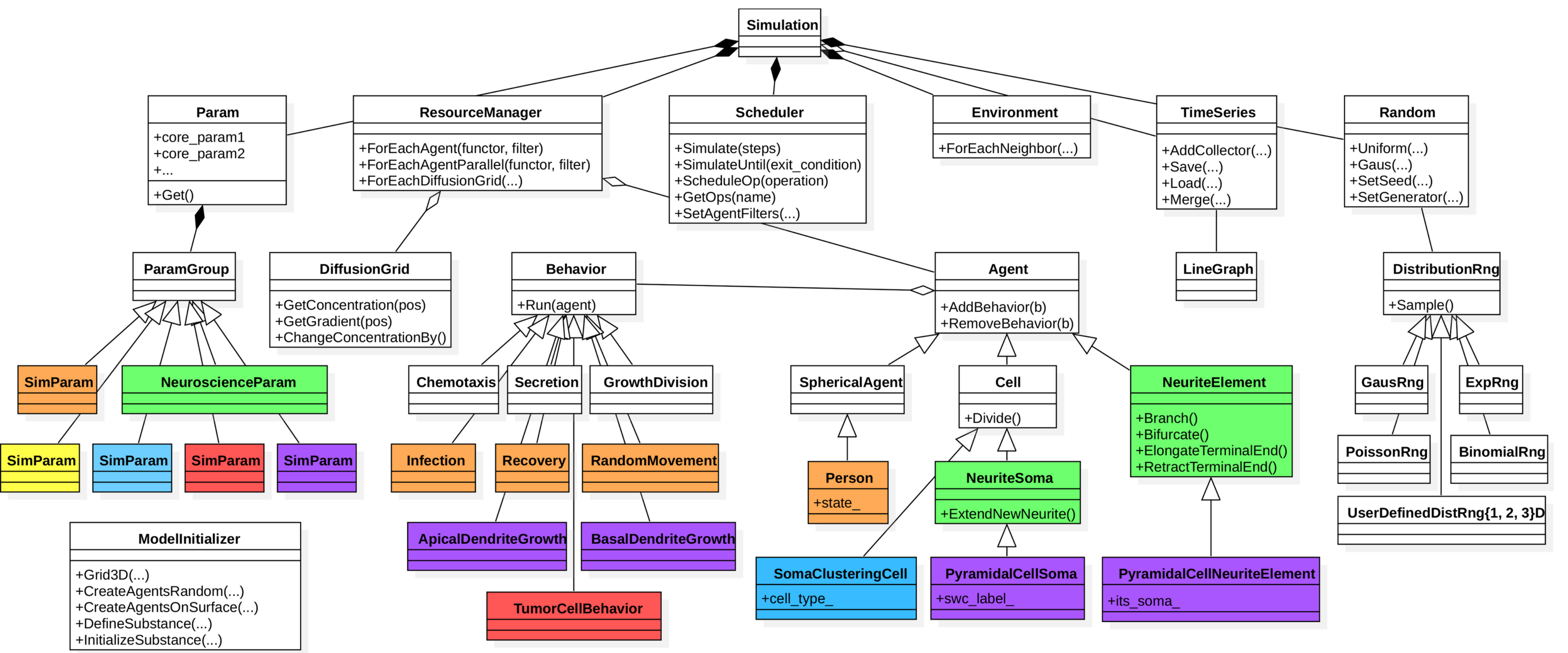
Libraries: OpenMP | ROOT | ParaView | Others

Operating System: Linux / MacOS

Hardware: (Multi-core) CPUs | GPU

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

BioDynaMo classes



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

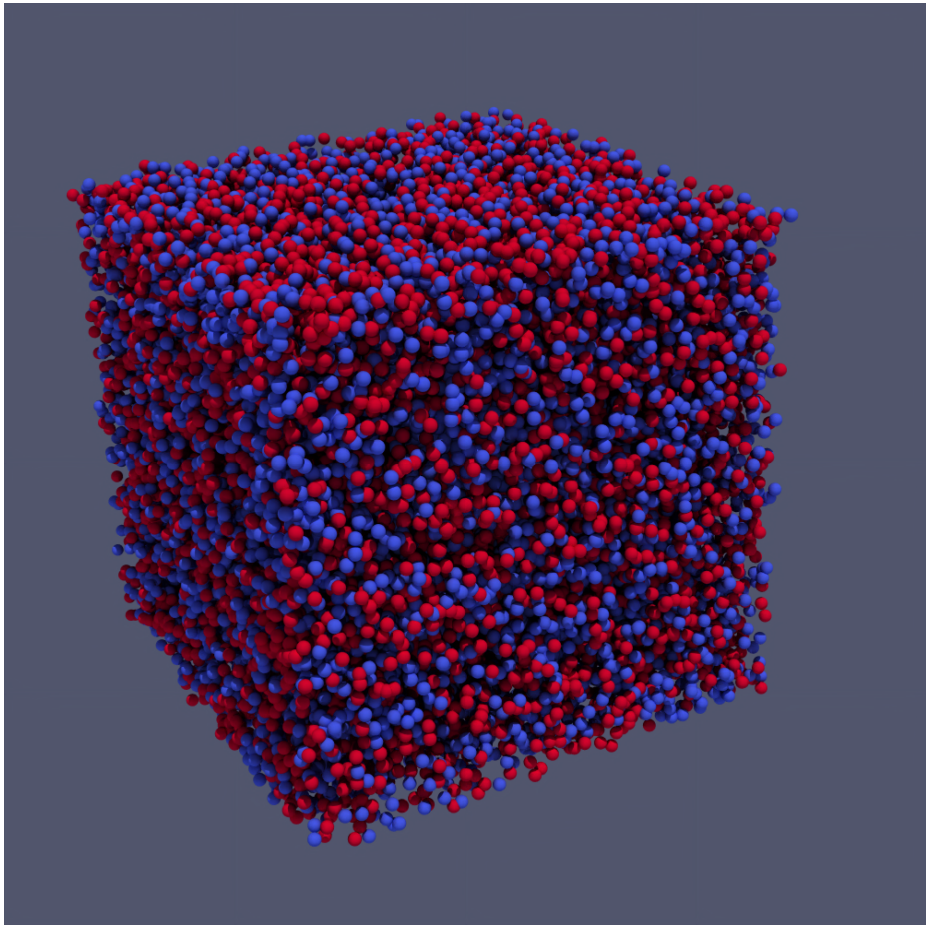
“Hello World” Simulation

```
#ifndef HELLO_WORLD_H_
#define HELLO_WORLD_H_

#include "biodynamo.h"

namespace bdm {

inline int Simulate(int argc, const char** argv) {
  Simulation simulation(argc, argv);

  // Define initial model - in this example: single cell at origin
  simulation.GetResourceManager()->AddAgent(new Cell(30));

  // Run simulation for one time step
  simulation.GetScheduler()->Simulate(1);
  return 0;
}

}  // namespace bdm

#endif  // HELLO_WORLD_H_
```

Demos

Cell clustering

- Two types of cells
  - red and blue
- Randomly distributed in 3D space
- Each cell has two behaviors
  - Secrete a substance into the extracellular matrix
  - Follow the concentration gradient (chemotaxis)



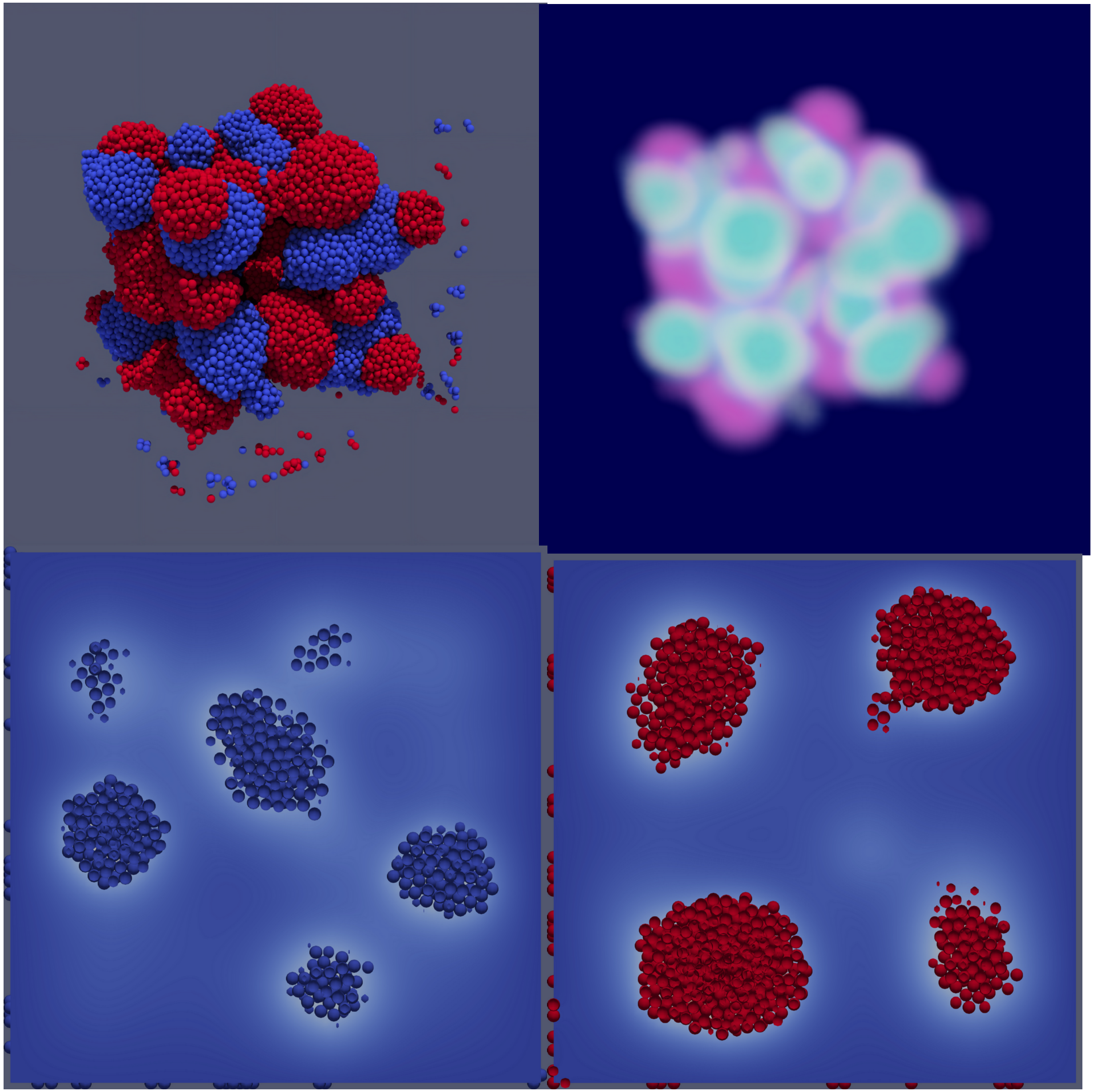
Cell clustering implementation

```
enum Substances { kSubstance0, kSubstance1 };

inline int Simulate(int argc, const char** argv) {
  Simulation simulation(argc, argv);

  // Define initial model
  auto* param = simulation.GetParam();
  int num_cells = 20000;

  // Define the substances that cells may secrete
  // Order: substance_name, diffusion_coefficient, decay_constant, resolution
  ModelInitializer::DefineSubstance(kSubstance0, "Substance_0", 0.5, 0.1, 20);
  ModelInitializer::DefineSubstance(kSubstance1, "Substance_1", 0.5, 0.1, 20);

  int cell_type = 1;
  std::string substance_name = "Substance_0";

  auto construct = [&cell_type, &substance_name](const Double3& position) {
    auto* cell = new MyCell(position, cell_type);
    cell->SetDiameter(10);
    cell->AddBehavior(new Secretion(substance_name));
    cell->AddBehavior(new Chemotaxis(substance_name, 5));
    return cell;
  };

  // Construct num_cells/2 cells of type 0
  ModelInitializer::CreateAgentsRandom(param->min_bound, param->max_bound,
                                       num_cells / 2, construct);
  // Construct num_cells/2 cells of type 1
  cell_type = -1;
  substance_name = "Substance_1";
  ModelInitializer::CreateAgentsRandom(param->min_bound, param->max_bound,
                                       num_cells / 2, construct);

  // Run simulation for N timesteps
  simulation.GetScheduler()->Simulate(1000);
  return 0;
}
```

Cell clustering video

https://www.youtube.com/watch?v=jlOk_Y3SUHo

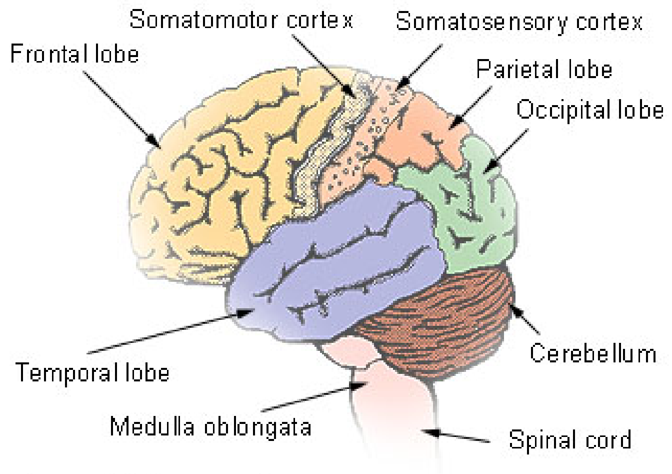
Cell clustering result



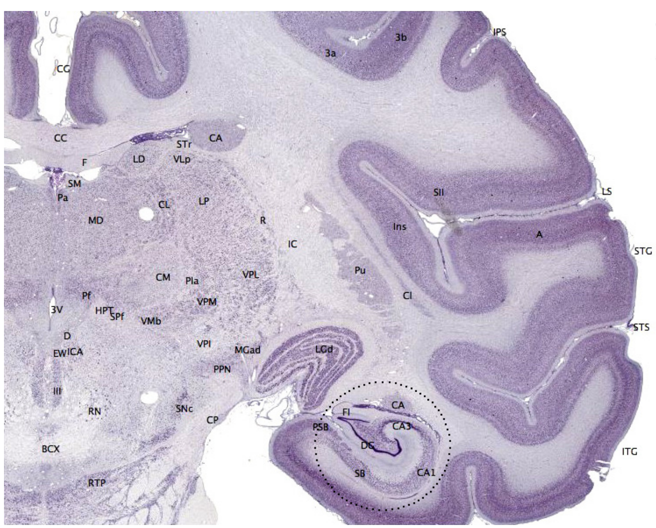
Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

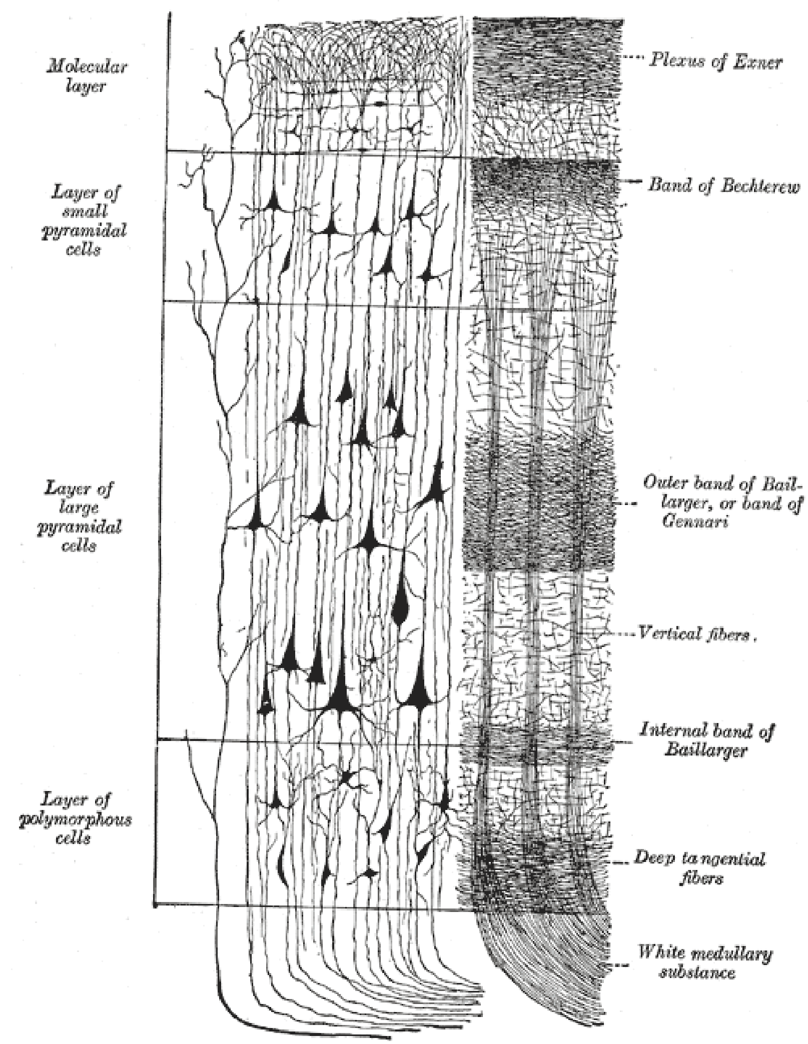
Neuroscience Use Case

Overview

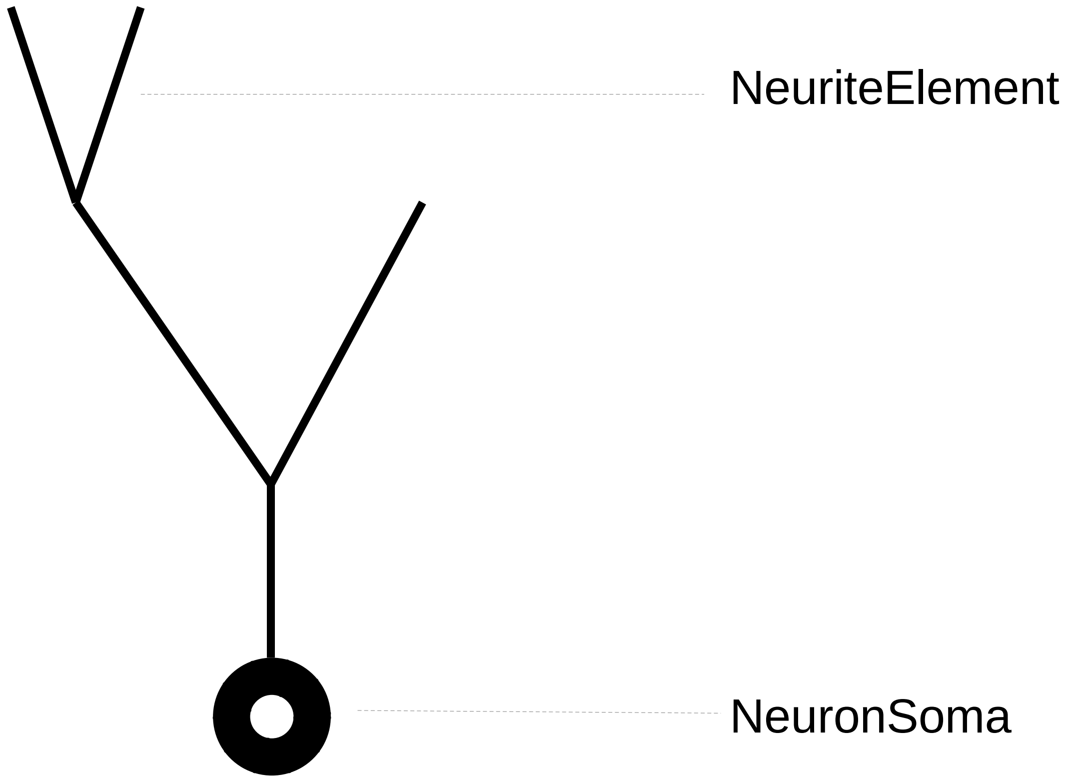


Lobes of the cerebrum





Model

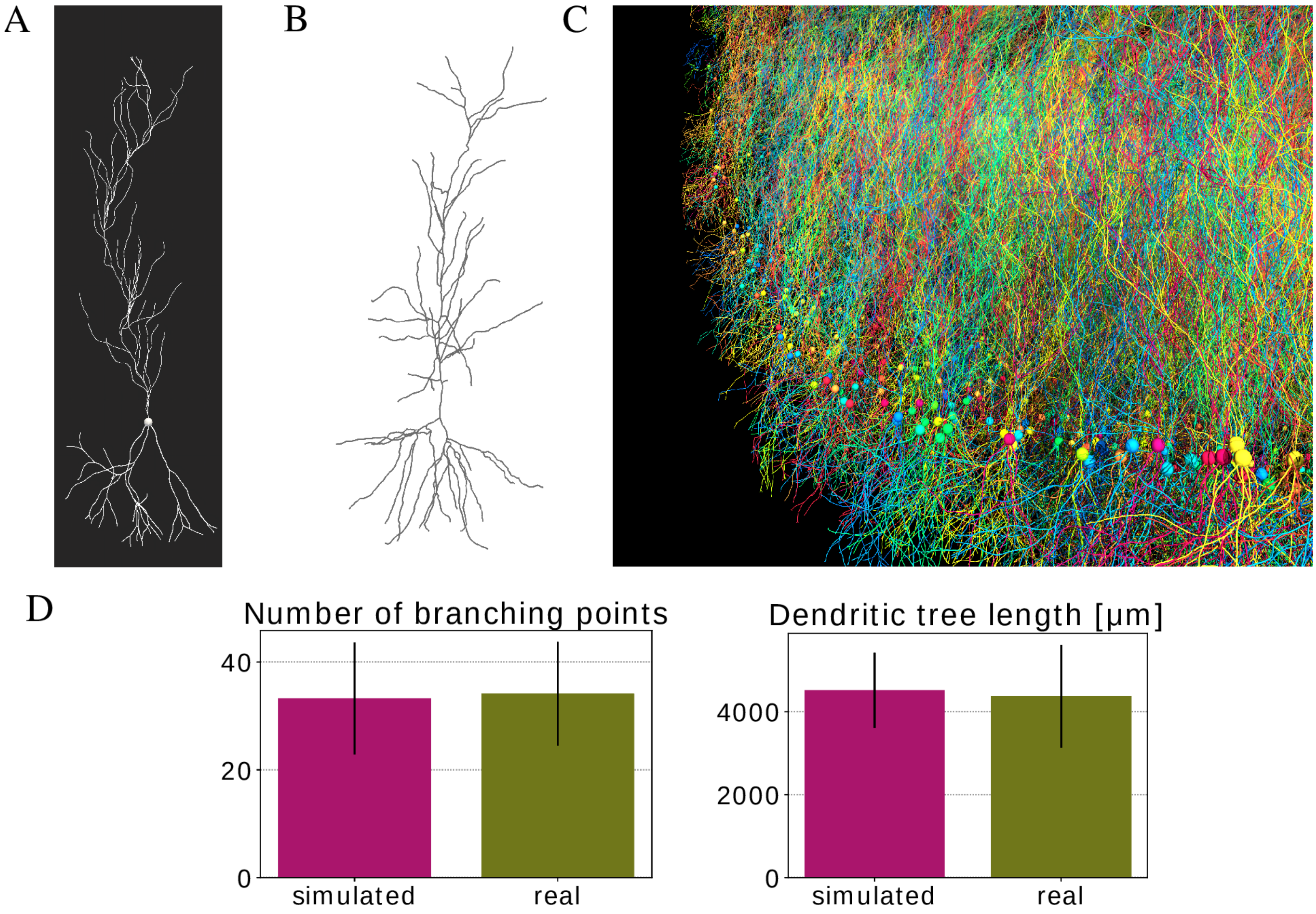


Neuroscience use case video

- Single pyramidal cell
  https://www.youtube.com/watch?v=taWMFs5D5Pg

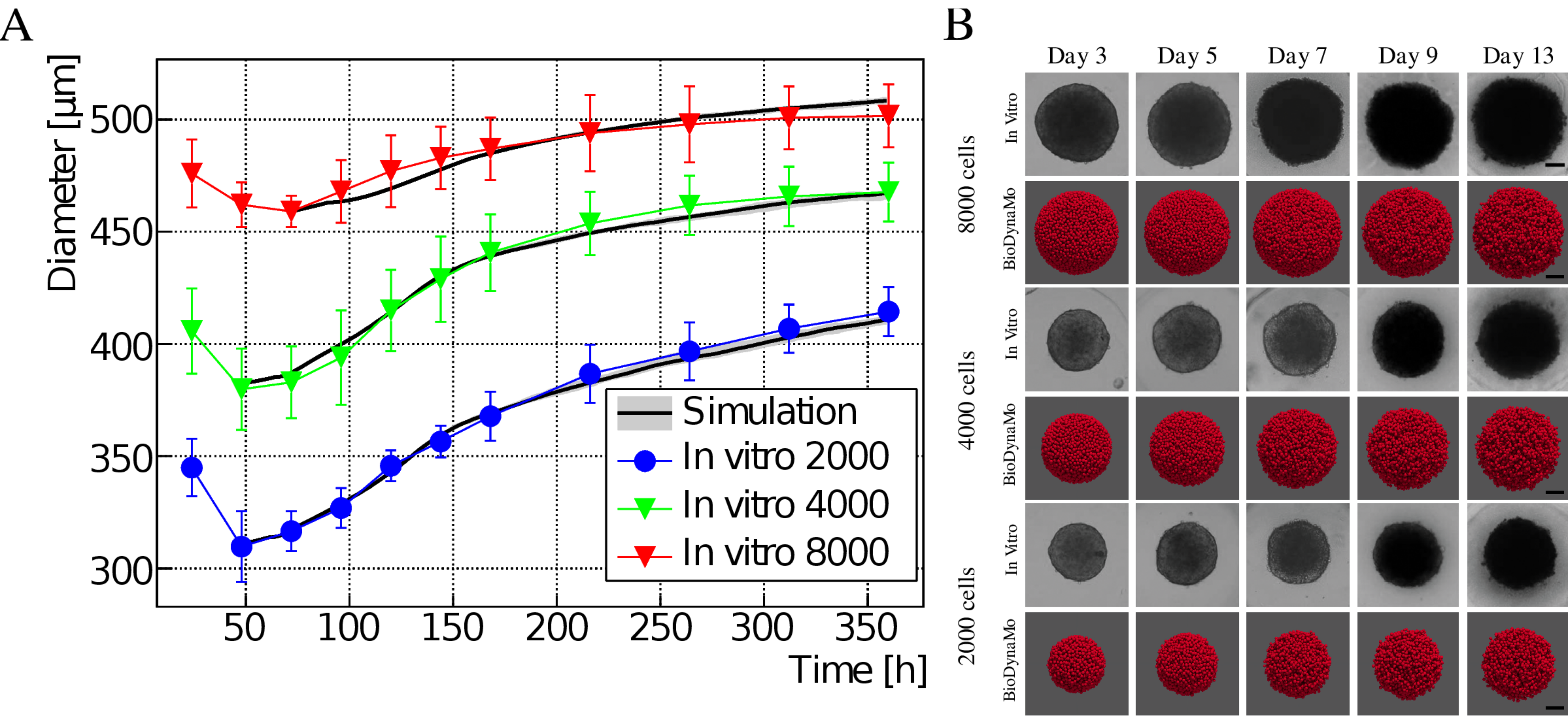
- Large-scale simulation
  https://www.youtube.com/watch?v=MA74wZbhO7w

Comparison with real neurons



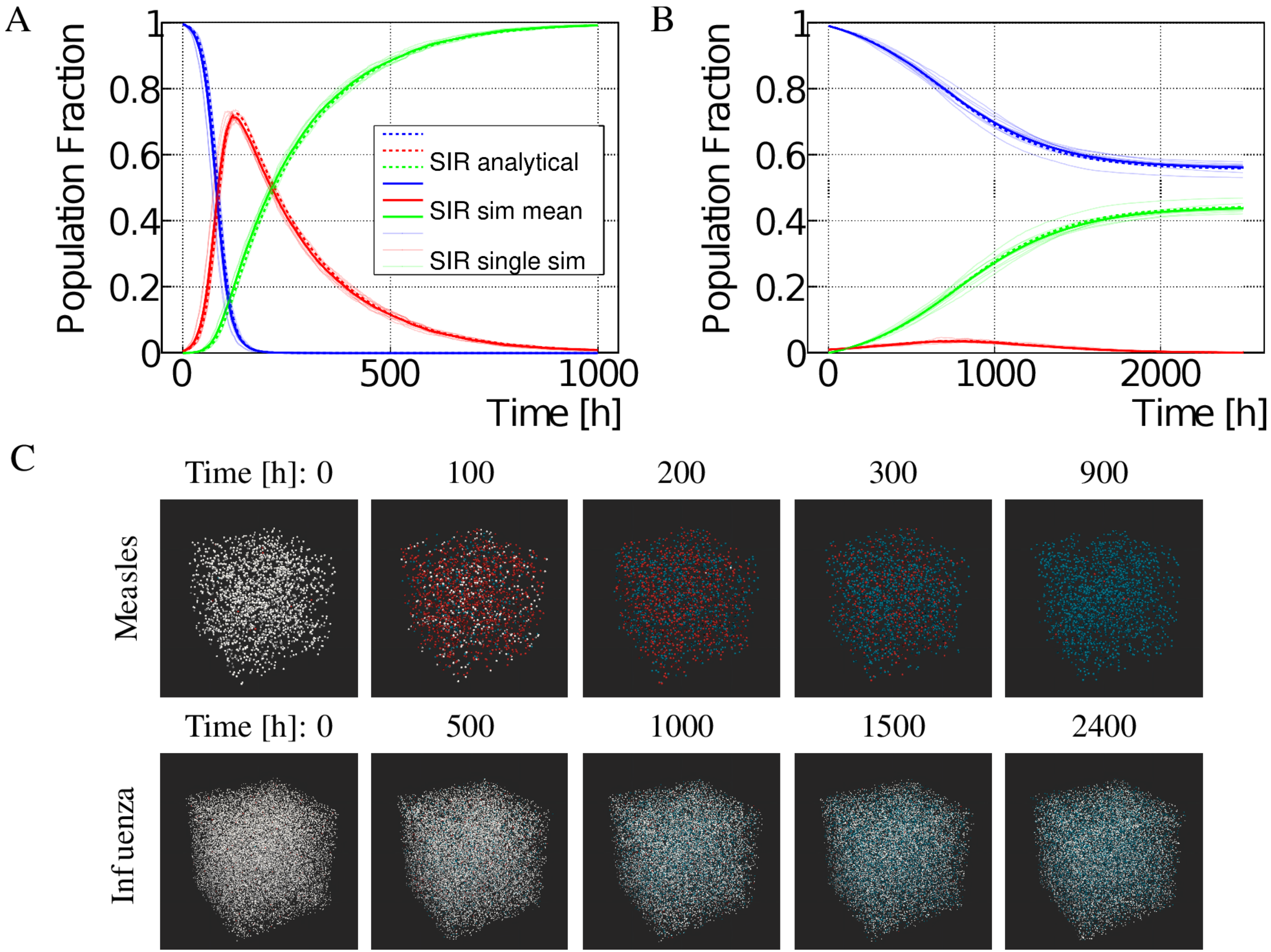
Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

Oncology Use Case



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

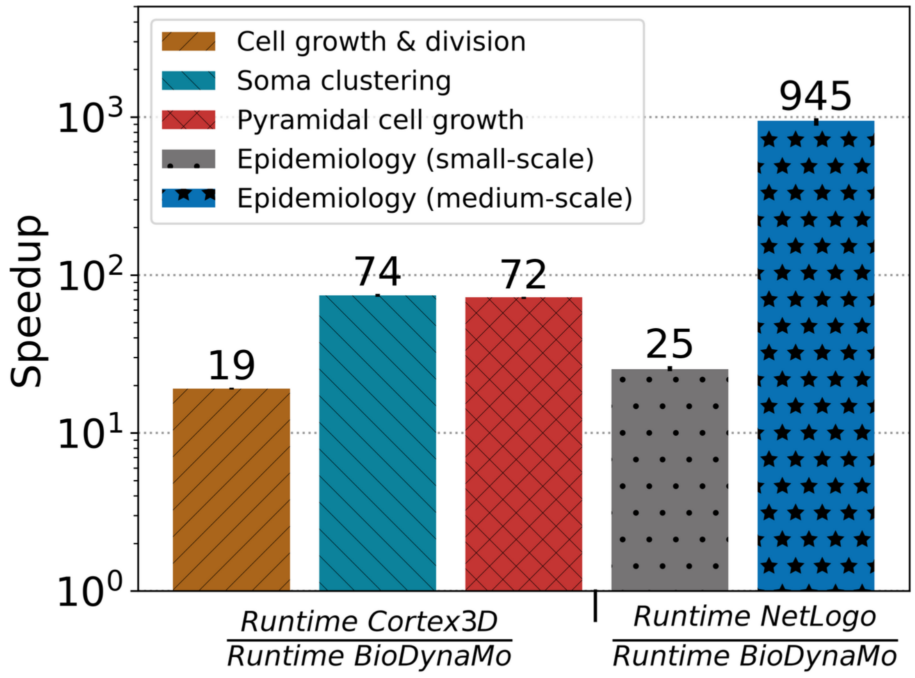
Epidemiology Use Case



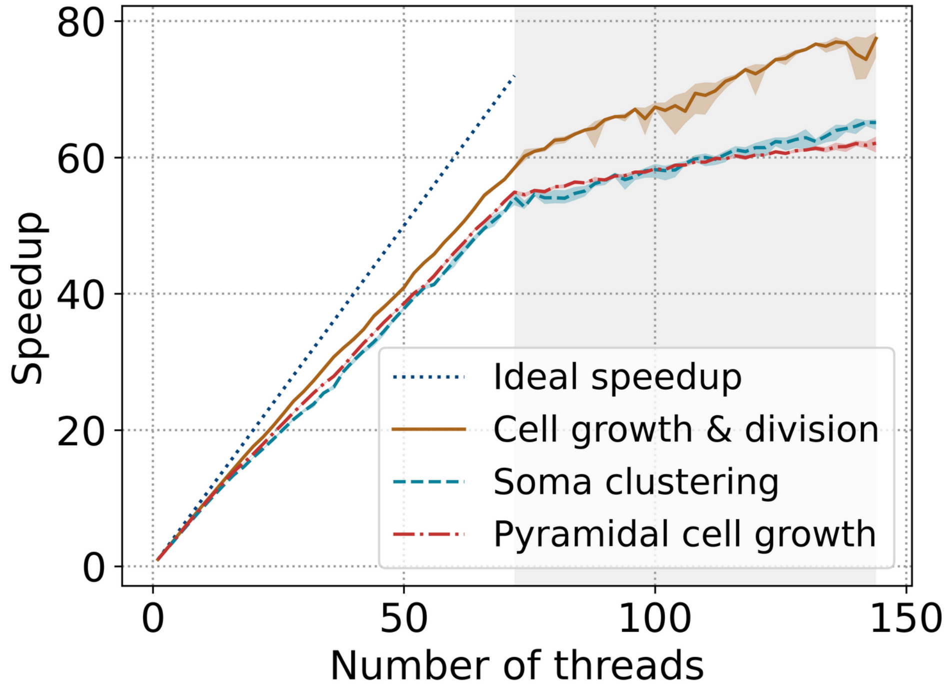
Source: Breitwieser et al. 2021, https://doi.org/10.1093/bioinformatics/btab649

BioDynaMo Performance 1/2

A



B



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

BioDynaMo Performance 2/2

Table 6. **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.

Simulation	Agents	Diffusion volumes	Iterations	System (Table 5)	Physical CPUs	Runtime	Memory
Neuroscience use case							
Single (Figure 4A in the main manuscript)	1 494	250	500	A	1	0.16 s	382 MB
				D	1	0.12 s	479 MB
Large-scale (Figure 4C in the main manuscript)	9 036 986	65 536	500	A	72	35 s	6.47 GB
				D	2	11 min 28 s	5.37 GB
Very-large-scale	1 018 644 154	5 606 442	500	B	72	1 h 24 min	438 GB
Oncology use case (Figure 5 in the main manuscript)							
2000 initial cells	4 177	0	312	A	1	1.05 s	382 MB
				D	1	0.832 s	480 MB
4000 initial cells	5 341	0	312	A	1	1.76 s	382 MB
				D	1	1.34 s	480 MB
8000 initial cells	7 861	0	288	A	1	3.27 s	384 MB
				D	1	2.60 s	482 MB
Large-scale	1 000 3925	0	288	A	72	1 min 42 s	7.42 GB
				D	2	43 min 56 s	5.84 GB
Very-large-scale	986 054 868	0	288	B	72	6 h 21 min	604 GB
Epidemiology use case (Figure 6C in the main manuscript)							
Measles	2 010	0	1000	A	1	0.53 s	381 MB
				D	1	0.42 s	479 MB
Seasonal Influenza	20 200	0	2500	A	1	16.41 s	383 MB
				D	1	16.40 s	479 GB
Medium-scale (measles)	100 500	0	1000	A	72	1.36 s	1 GB
Large-scale (measles)	10 050 000	0	1000	A	72	59.19 s	5.87 GB
				D	2	19 min 18 s	5.41 GB
Very-large-scale (measles)	1 005 000 000	0	1000	B	72	2 h 0 min	495 GB
Soma clustering (Figure 2)	32 000	1 240 000	6 000	A	72	12.91 s	1.02 GB
				D	2	2 min 7 s	522 MB

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

Questions?

https://forum.biodynamo.org

Lukas.Breitwieser@cern.ch
https://www.linkedin.com/in/lukasbreitwieser